Private Equilibrium Release, Large Games, and No-Regret Learning*

Michael Kearns† Mallesh M. Pai‡ Aaron Roth§ Jonathan Ullman¶

November 5, 2012

ABSTRACT

We give mechanisms in which each of n players in a game is given their component of an (approximate) equilibrium in a way that guarantees differential privacy — that is, the revelation of the equilibrium components does not reveal too much information about the utilities of other players. More precisely, we show how to compute an approximate correlated equilibrium (CE) under the constraint of differential privacy (DP), provided n is large and any player's action affects any other's payoff by at most a small amount. Our results draw interesting connections between noisy generalizations of classical convergence results for no-regret learning, and the noisy mechanisms developed for differential privacy. Our results imply the ability to truthfully implement good social-welfare solutions in many games, such as games with small Price of Anarchy, *even if the mechanism does not have the ability to enforce outcomes*. We give two different mechanisms for DP computation of approximate CE. The first is computationally efficient, but has a suboptimal dependence on the number of actions in the game; the second is computationally inefficient, but allows for games with exponentially many actions. We also give a matching lower bound, showing that our results are tight up to logarithmic factors.

*We gratefully acknowledge the support of NSF Grant CCF-1101389. We thank Nabil Al-Najjar, Eduardo Azevdeo, Tymofiy Mylovanov, Andy Postlewaite, Al Roth and Tim Roughgarden for helpful comments and discussions.

†Department of Computer and Information Science, University of Pennsylvania. Email:mkearns@cis.upenn.edu.

‡Department of Economics, University of Pennsylvania. Email:mallesh@econ.upenn.edu.

§Department of Computer and Information Science, University of Pennsylvania. Email:aaroth@cis.upenn.edu.

¶School of Engineering and Applied Sciences, Harvard University. Email:jullman@seas.harvard.edu.

1 Introduction

The field of mechanism design studies how to provide incentives to implement a desired outcome when agents have relevant private information. We revisit this with an additional desideratum motivated by privacy concerns — that no agent's information be revealed to any other agent, either directly or by the portion of the outcome that is revealed to any agent. This is relevant for mechanism design when the underlying private information is sensitive (e.g. healthcare) or agents presume privacy (e.g. agents' activity on social networks). Similarly, if there are (unmodeled) future interactions between the agents, privacy concerns are a reduced-form way to incorporate strategic concerns regarding the future. For a broad class of games, we derive a mechanism which (approximately) truthfully implements an (approximate) correlated equilibrium, where the approximation error quickly approaches zero as the size of the game grows. In particular, this gives us an *approximately truthful equilibrium selection mechanism* which does not need the power to enforce outcomes, but can instead simply make suggestions.

Consider a motivating example: imagine a city in which Google Navigation has become the dominant navigation service. Every morning, each person enters their starting point and destination into their Google device, receives a set of directions, and chooses his/her route according to those directions. In this setting our question reduces to the design of the navigation service such that: 1) Each agent should be incentivized to report his starting and end points truthfully, and then follow the driving directions provided. Both misreporting start and end points, and truthfully reporting start and end points, but then following a different (shorter) path should be ruled out for each agent. 2) Players are guaranteed the *privacy* of their starting and ending points, i.e. the mechanism should be such that other player or players cannot infer 'much' about a given player's source or destination based on the directions they received. As an alternative example with perhaps more realistic privacy concerns, consider a setting in which the members of a large population must contemplate revealing their status regarding an infectious disease (infected, uninfected, unknown) to a centralized mechanism that will recommend vaccinations based on social or physical proximity among members in a way that ensures best responses (equilibrium). Again in this setting, members would like to receive the best-response benefits of participation, but also want to strongly limit information leakage about their infection status.

Intuitively, our two desiderata are in conflict. In the commuting example above, if we are to guarantee that every player is incentivized to truthfully follow their suggested route, then we must compute an equilibrium of the game in question. On the other hand, to do so, our suggested route to some player i must *depend* on the reported location/destination pairs of each other player $j \neq i$. This tension seems also to pose a problem in terms of incentives: if we must compute an equilibrium of a game that is *defined* based on the reports of the players in step 1), an agent can potentially benefit by *misreporting* in the first step, causing us to compute an equilibrium of *the wrong game*. However, as we show, both of these problems can be alleviated by computing the equilibrium subject to the constraint of differential privacy.

Our mechanism is based on a combination of two ideas from the literature. The first of these ingredients is the use of 'no-regret methods' to compute approximate correlated equilibria (see, e.g. Foster & Vohra [16] and Hart & Mas-Colell [25]). A crucial feature of these methods is that equilibria can be computed *without the algorithms having direct access to the game matrix*. Instead, we "simulate play" of the game for only a small number of rounds, and need only feed each algorithm certain numeric values at each round: namely, the payoff that each simulated agent would have received, given the actions being played in the current round by the other simulated agents. We show that these

algorithms are extremely noise tolerant: that is, they work even if the reported payoffs are perturbed. This means that we can compute equilibria by accessing the game only via a small number of perturbed numeric valued queries. We may therefore apply the large body of literature in differential privacy addressing the question of how to privately answer numeric valued queries as accurately as possible.

We can view our algorithm as a mechanism that takes as input each player's report of (private) type; and outputs a suggested action to each player. It implements an approximate correlated equilibrium of the full information game given players' reports. Therefore, as in our example, the mechanism can also be used as a recommender mechanism for a game in which agents take actions directly. Further, this mechanism has both the desired incentive properties and preserves the privacy of each agent's information. Moreover, the approximations quickly become exact as the size of the game (i.e. the number of players) grows, so long as the game is "insensitive" in a sense in which we precisely define.

1.1 Overview of Model and Results

We consider a setting in which a centralized planner simultaneously receives type reports from each agent and proposes an action to each. We study the design of mechanisms that:

1. Propose an approximate equilibrium of the full information game given the reports. Our solution concept here is ϵ-correlated equilibrium.[1]

2. Make it an approximately-dominant strategy for agents to report their type truthfully.

3. Preserve the privacy of each agent's private information. Here we use a new variant of differential privacy defined in this paper. Informally speaking, we require that simultaneously for each player i, the joint distribution over actions reported to players $j \neq i$ be differentially private in the report of player i. This means, roughly, that although player i's suggested action can be highly sensitive in his own reported type, it must be insensitive to the reported type of any other player.

We note that this natural variant of differential privacy, which we call *joint differential privacy* is necessary in our setting, and may be of independent interest. In our case, it provides equally strong guarantees of privacy to each player i, even if all other agents $j \neq i$ collude and share their outputs, while allowing us to circumvent the impossibility of computing an equilibrium under the standard notion of differential privacy.

It is easy to see that the goal of computing an approximate equilibrium while preserving the privacy of the player's utility functions is hopeless in a 2-player game (or more generally a small number of players). Therefore we consider 'large' n-player games. We define these formally in Section 2, but roughly speaking, these are n-player games in which for all players $i \neq j$, i's choice of action can affect j's payoff by at most an additive $\pm\gamma$. We call γ the *sensitivity* of the game. In what follows, we discuss our results for games where γ is $O(1/n)$, but our results extend to other scales for γ. Examples of such games include atomic routing games as the player size decreases

[1] For certain classes of games, this can be extended to ϵ-Nash equilibria. The main constraint is our proof technique. We need that the solution concept must be computable by an appropriate distributed algorithm, to which we can add carefully calibrated noise. In certain special cases, these conditions are satisfied for Nash equilibrium, but in this paper we restrict our attention to correlated equilibrium so as to maintain generality.

(i.e. as the game approaches a non-atomic routing game), anonymous matching games, and more generally, any game in which a player's payoff depends in some Lipschitz-continuous manner only on the distribution of actions played (in aggregate) by his opponents.

We consider two equilibrium concepts: coarse correlated equilibrium (CCE), and correlated equilibrium (CE). For both solution concepts, we give a computationally efficient mechanism for privately computing α-approximate versions in games with k actions where α is $O(\text{poly}(k)/\sqrt{n})$. Holding the number of actions fixed, the approximation is $O(1/\sqrt{n})$, or to put it alternately, we get almost exact equilibria if the number of players n is large.

For games with a large number of actions (relative to n) the above algorithm is not useful, due to the $\text{poly}(k)$ term in the numerator. For example, in the routing game discussed earlier, the number of actions available to a player is the number of paths, which can be exponentially large relative to the size of the graph. For such settings with large numbers of actions, we show that positive results are still possible as long as the number of possible types for each player is bounded. Formally, we show that it is possible to privately compute an α-approximate equilibrium in a large k-action n-player game, with U types for each player, where α is $O(\log k \log^{3/2} |U|/\sqrt{n})$. However, the mechanism in this case is computationally inefficient.

We also show a matching lower bound: we give a family of n-player 2-action large games in which it is not possible to privately compute an α-approximate CCE (and therefore an α-approximate CE or an α-approximate Nash equilibrium) for $\alpha \ll 1/\sqrt{n}$, showing that even our efficient algorithm gives nearly the best possible approximation guarantees in the case that k is small (i.e. a fixed number independent of n). Our inefficient upper bound of course remains tight up to a factor of $\log k \log^{3/2} U$ for arbitrary k-action games with U feasible utility functions. Whether there is an *efficient* algorithm for privately computing α-approximate equilibria to error $\alpha = O(\text{polylog}(k, U)/\sqrt{n})$ is left as an open question.

What do these results mean in terms of incentive properties? It has been observed previously that differential privacy implies approximate strategy proofness (McSherry & Talwar [32]). Specifically, an ϵ differentially private mechanism is also ϵ-approximate dominant strategy truthful[2]. Since the actions proposed jointly constitute a α-approximate correlated equilibrium of the full information game defined by everyone's reports, it is a $(\epsilon + \alpha)$-approximate Nash equilibrium for everyone to follow the strategy "truthfully report type, then follow the recommended action".[3] Note that crucially this does not require the mechanism to have the power to enforce any outcome! We show a mechanism such that $(\epsilon + \alpha)$ quickly tends to zero in the size of the game. Therefore, as the size of the game grows large, truthfully reporting type and following the suggested action approaches an exact Nash equilibrium of the full information game.

Finally, note that for many games of interest (including the traffic routing game that serves as our running example), the *Price of Anarchy* over the set of (coarse) correlated equilibria is very small [3, 42]. Indeed, in any *smooth* game, the Price of Anarchy over this set of equilibria is no worse than the Price of Anarchy over pure strategy Nash equilibria [42]. Since our mechanism implements a correlated equilibrium, it has welfare guarantees that are at least as strong as the Price of Anarchy bound in the game of interest, which in many cases is extremely strong.

[2] In fact, *every* action is an ϵ-approximate dominant strategy in such a mechanism, which has been a criticism of privacy as a solution concept [35]. This objection does not apply to our setting, since the messages reported to each agent are *not* differentially private in their own reported type, but only in the reported types of others.

[3] It is always an ϵ-approximate *Dominant* strategy to truthfully report type. It is an $\epsilon + \alpha$ Nash equilibrium to follow both parts of the two-part strategy, of truthfully reporting, and then following the resulting suggested equilibrium action.

1.2 Related Work and Discussion

Market and Mechanism Design Our work is closely related to the large body of literature on mechanism/market design in 'large games'. This literature looks to exploit the large number of agents to provide mechanisms which have good incentive properties, even when the small market versions do not. It stretches back to Roberts & Postlewaite [36] who showed that market (Walrasian) equilibria are approximately strategy proof in large economies. More recently Immorlica and Mahdian [27] , Kojima and Pathak [29], Kojima, Pathak and Roth [30] have shown that various two-sided matching mechanisms are approximately strategy proof in large markets. There are similar results in the literature for one-sided matching markets, market economies, and double auctions. Azevedo and Budish [2] in a recent paper provide conditions for a mechanism to be 'strategy proof in the large', i.e. approximately strategyproof as the game grows large.

By comparison with these works, which study settings where the mechanism designer/principal can enforce outcomes (or take actions on behalf of participants), we study settings where the mechanism only suggests an action to participants. This leads to slightly weaker incentive properties (due to the possibility of 'double-deviations'). Indeed, if our mechanism could act on behalf of participants, it would be $(\epsilon + \alpha)$-approximately strategy proof when an α-approximate correlated equilibrium is computed while satisfying ϵ-differential privacy.[4]

On a related subject, there is literature suggesting that even if the mechanism can enforce outcomes rather than only suggest an action, other considerations may require the mechanism to select a 'equilibrium' outcome of the underlying game rather than an 'optimal' outcome. An influential body of work, starting with Roth and Xing [41] argues that in two-sided matching markets, centralized mechanisms that implement a stable outcome (a full information solution concept) are more resistant to *unraveling*, i.e. members of the market pre-empting the mechanism by contracting in advance.

Large Games Our results hold under two sufficient (and almost necessary) conditions: that the number of players be 'large', and the game be insensitive to $O(1/\sqrt{n})$, i.e. a player's action affects the payoff of all others by a small amount. These are closely related to the literature on large games, see e.g. Al-Najjar and Smorodinsky [1] or Kalai [28]. There has been recent work studying large games using tools from theoretical computer science (but in this case, studying robustness of equilibrium concepts)– see Gradwohl and Reingold [19, 20].

Differential Privacy Differential privacy is a formalization of privacy first defined by Dwork, McSherry, Nissim, and Smith [10] that has since become the standard privacy "solution concept" in the theoretical computer science literature. It is a quantification of the *worst-case harm* that can befall an individual as a result of his decision to allow his data to be used in some computation, as compared to if he did not provide his data.

There is by now a very large literature on differential privacy, which we will not attempt to survey. Instead, we mention here only the most relevant work. Interested readers can consult [8, 37] for a more thorough introduction to the field.

The most well studied problem in differential privacy is that of accurately answering numeric-valued queries on a data set. A basic result is that any single query that has sensitivity at most 1 (i.e. the addition or removal of a single individual from the data set can change the value of the query by at

[4] In fact, if the participants did not have the option of acting independently of the mechanism (i.e. still playing the game, but selecting an action without consulting the mechanism), then our mechanisms would be ϵ-strategyproof.

most 1) can be answered in a computationally efficient manner while preserving ϵ-differential privacy, and introducing error only $O(1/\epsilon)$ (Dwork et al [10]). Another fundamental result in differential privacy is that it composes gracefully: Any algorithm composed of T subroutines, each of which are $O(\epsilon/\sqrt{T})$-differentially private is itself ϵ-differentially private [9, 13]. Combined with the previous result, this gives an efficient algorithm for privately answering any T low sensitivity queries with error that grows only with $O(\sqrt{T})$, a result which we make use of.

Another line of work has shown that it is possible to privately answer queries much more accurately using computationally inefficient algorithms [4, 13, 21, 22, 24, 39]. Combining the results of [24, 39] yields an algorithm which can privately answer arbitrary low sensitivity queries, interactively as they arrive, with error that scales only logarithmically in the number of queries. We make use of this when we consider games with large action spaces but small type spaces.

There is also a line of work proving information theoretic lower bounds on the accuracy to which low sensitivity queries can be answered while preserving differential privacy [6, 7, 11, 14]. Our lower bounds for privately computing equilibria work by reducing the problem to privately answering queries: we design a game whose only equilibria encode answers to large numbers of queries about a database.

Finally, related to this paper, there is a recent literature on connections between differential privacy and game theory. McSherry and Talwar [32] were the first to observe that a differentially private algorithm is also approximately truthful. This line of work was extended by Nissim, Smorodinsky, and Tennenholtz [35] to give mechanisms in several special cases which are exactly truthful (although no longer privacy preserving) by combining private mechanisms with non-private mechanisms which explicitly punish non-truthful reporting. Huang and Kannan [26] showed that the mechanism used by Mcsherry and Talwar (the "exponential mechanism") is in fact maximal in distributional range, and so can be made exactly truthful with the addition of payments. We remark that the immediate connection between privacy and approximate incentive compatibility leveraged by these works only holds in settings in which the mechanism has the power to enforce its outcome or otherwise compel actions. The novelty in our work relative to this line is that our mechanisms implement approximate equilibria of the full information game. Therefore, truthful reporting and subsequently following the suggested equilibria actions remain approximate best responses *even if the players have the ability to act in the game, independently of the mechanism*.

Another interesting line of work considers the problem of designing truthful mechanisms for agents who explicitly experience a cost for privacy loss as part of their utility function [5, 34, 43]. The main challenge in this line of work is to formulate a reasonable model for how agents experience cost as a function of privacy. We remark that the approaches taken in the former two can also be adapted to work in our setting, for agents who explicitly value privacy. Gradwohl [18] studies the problem of implementation for various assumptions about players' preference for privacy and permissible game forms. A related line of work which also takes into account agent values for privacy considers the problem of designing markets by which analysts can procure private data from agents who explicitly experience costs for privacy loss [15, 17, 31, 40]. See Roth [38] for a survey.

2 Model & Preliminaries

There is a set of n players, $\{1, 2, \ldots, n\}$, the generic player is indexed i. Player i can take actions in a set A, $|A| = k$.[5] We denote a generic action by j and a generic action for player i by a_i. A tuple of actions, one for each player, will be denoted $\mathbf{a} = (a_1, a_2, \ldots a_n) \in A^n$.[6]

Player i's payoff function will be denoted $u_i \colon A^n \to \Re$. We will restrict attention to 'insensitive' games. Roughly speaking a game is γ-sensitive if a player's choice of action affects any other player's payoff by at most γ. Note that we do not constrain the effect of a player's *own* actions on his payoff. Formally:

DEFINITION 1 (γ-Sensitive). *A game is said to be γ-sensitive if for any two distinct players $i \neq i'$, any two actions a_i, a_i' for player i and any tuple of actions a_{-i} for everyone else:*

$$|u_{i'}(a_i, a_{-i}) - u_{i'}(a_i', a_{-i})| \leq \gamma. \tag{1}$$

Denote a distribution over A^n by π, the marginal distribution over the actions of player i by π_i, and the marginal distribution over the (joint tuple of) actions of every player but player i by π_{-i}. We now present (approximate versions of) two standard solution concepts—correlated and coarse correlated equilibrium.

DEFINITION 2 (Approximate Coarse Correlated Equilibrium). *Let $(u_1, u_2, \ldots u_n)$ be a tuple of utility functions, one for each player. Let π be a distribution over tuples of actions A^n. We say that π is an* α-approximate coarse correlated equilibrium *of the game defined by $(u_1, u_2, \ldots u_n)$ if for every player i, and any $a_i' \in A$:*

$$\mathop{\mathbb{E}}_{\pi}[u_i(\mathbf{a})] \geq \mathop{\mathbb{E}}_{\pi_{-i}}\left[u_i(a_i', a_{-i})\right] - \alpha$$

DEFINITION 3 (Approximate Correlated Equilibrium). *Let $(u_1, u_2, \ldots u_n)$ be a tuple of utility functions, one for each player. Let π be a distribution over tuples of actions A^n. We say that π is an* α-approximate correlated equilibrium *of the game defined by $(u_1, u_2, \ldots u_n)$ if for every player $i \in [N]$, and any function $f \colon A \to A$,*

$$\mathop{\mathbb{E}}_{\pi}[u_i(\mathbf{a})] \geq \mathop{\mathbb{E}}_{\pi}[u_i(f(a_i), a_{-i})] - \alpha$$

Let $\mathcal{U}$ be the set of all possible utility functions for the players,[7] with a generic profile of utilities $\mathbf{u} = (u_1, u_2, \ldots u_n) \in \mathcal{U}^n$. A mechanism is a function from a profile of utility functions to a probability distribution over $\mathcal{R}^n$, i.e. $\mathcal{M} \colon \mathcal{U}^n \to \Delta\mathcal{R}^n$. Here $\mathcal{R}$ is an appropriately defined range space.

First we recall the definition of differential privacy, both to provide a basis for our modified definition, and since it will be a technical building block in our algorithms. Roughly speaking, a mechanism is differentially private if for every $\mathbf{u}$ and every i, knowledge of the output $\mathcal{M}(\mathbf{u})$ as well as u_{-i} does not reveal 'much' about u_i.

[5] It is trivial to extend our results to the case where agents have different sets of actions, k will then be an upperbound on the number of actions across agents.

[6] In general, subscripts will refer indices i.e. players and periods, while superscripts will refer to components of vectors.

[7] It is trivial to extend our results to the case where agents have different sets of possible utility functions, $\mathcal{U}_i$. $\mathcal{U}$ will then be $\bigcup_{i=1}^{n} \mathcal{U}_i$.

DEFINITION 4 ((Standard) Differential Privacy). *A mechanism $\mathcal{M}$ satisfies (ε, δ)-differential privacy if for any player i, any two possibility utility functions for player i, u_i and u_i', and any tuple of utilities for every else u_{-i} and any $S \subseteq \mathcal{R}^n$,*

$$\mathop{\mathbb{P}}_{\mathcal{M}}\left[(\mathcal{M}(u_i; u_{-i})) \in S\right] \leq e^{\varepsilon} \mathop{\mathbb{P}}_{\mathcal{M}}\left[\left(\mathcal{M}(u_i'; u_{-i})\right) \in S\right] + \delta.$$

We would like something slightly different for our setting. We propose a relaxation of the above definition, motivated by the fact that the action recommended to a player is only observed by her. Roughly speaking, a mechanism is *jointly differentially private* if, for each player i, knowledge of the other $n-1$ recommendations (and submitted utility functions) does not reveal 'much' about player i's report. Note that this relaxation is necessary in our setting, since knowledge of player i's recommended action can reveal a lot of information about his utility function. It is still very strong- the privacy guarantee remains *even if* everyone else colludes against a given player i, so long as i does not himself make the component reported to him public. This relaxation also preserves the approximate truthfulness properties of private mechanisms.

DEFINITION 5 (Joint Differential Privacy). *A mechanism $\mathcal{M}$ satisfies (ε, δ)-joint differential privacy if for any player i, any two possible utility functions for player i, u_i and u_i', any tuple of utilities for everyone else u_{-i} and $S \subseteq \mathcal{R}^{n-1}$,*

$$\mathop{\mathbb{P}}_{\mathcal{M}}\left[(\mathcal{M}(u_i; u_{-i}))_{-i} \in S\right] \leq e^{\varepsilon} \mathop{\mathbb{P}}_{\mathcal{M}}\left[\left(\mathcal{M}(u_i'; u_{-i})\right)_{-i} \in S\right] + \delta.$$

An important result we will use is that differentially private mechanisms 'compose' nicely.

THEOREM 1 (Adaptive Composition [13]). *Let $\mathcal{A}\colon \mathcal{U} \to \mathcal{R}^T$ be a T-fold adaptive composition[8] of (ε, δ)-differentially private mechanisms. Then $\mathcal{A}$ 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 $\mathcal{A}$ is a T-fold adaptive composition of $(\varepsilon/\sqrt{8T\ln(1/\delta)}, 0)$-differentially privacy mechanisms, then $\mathcal{A}$ satisfies (ε, δ)-differential privacy.

Finally, differentially private mechanisms often involve adding Laplacian random noise. We will denote a (mean 0) and scale σ Laplacian random variable by $\mathrm{Lap}(\sigma)$. The following foundational result shows that adding Laplacian noise to a insensitive function makes it differentially private.

THEOREM 2 (Privacy of the Laplace Mechanism [10]). *Let $Q\colon \mathcal{U} \to \mathbb{R}$ be any γ-sensitive function. Define the mechanism $\mathcal{M}(\mathbf{u}) = Q(\mathbf{u}) + \mathrm{Lap}(\sigma)$. If $\sigma = \gamma/\varepsilon$, then $\mathcal{M}$ is $(\varepsilon, 0)$-differentially private.*

We state a known concentration inequality for Laplacian random variables that will be useful.

THEOREM 3 ([22]). *Suppose $\{Y_i\}_{i=1}^T$ are i.i.d. $\mathrm{Lap}(\sigma)$ random variables, and scalars $q_i \in [0, 1]$. Define $Y := \frac{1}{T}\sum_i q_i Y_i$. Then for any $\alpha \leq \sigma$,*

$$\Pr[Y \geq \alpha] \leq \exp\left(-\frac{\alpha^2 T}{6\sigma^2}\right).$$

[8] See [13] for further discussion

2.1 No-Regret Algorithms: Definitions and Basic Properties

Here we recall some of the basics about no-regret learning. See [33] for a text-book exposition.

Let $\{1, 2, \ldots, k\}$ be a finite set of k *actions*. Let $L = (l_1, \ldots, l_T) \in [0,1]^{T \times k}$ be a *loss matrix* consisting of T vectors of losses for each of the k actions. Let $\Pi = \left\{\pi \in [0,1]^k \mid \sum_{j=1}^k \pi^j = 1\right\}$ be the set of distributions over the k actions and let π_U be the uniform distribution. An *online learning algorithm* $\mathcal{A} \colon \Pi \times [0,1]^k \to \Pi$ takes a distribution over k actions and a vector of k losses, and produces a new distribution over the k actions. We use $\mathcal{A}_t(L)$ to denote the distribution produced by running $\mathcal{A}$ sequentially $t-1$ times using the loss vectors $l_1, \ldots, l_{t-1}$, and then running $\mathcal{A}$ on the resulting distribution and the loss vector l_t. That is:

$$\begin{aligned}\mathcal{A}_0(L) &= \pi_U, \\ \mathcal{A}_t(L) &= \mathcal{A}(\mathcal{A}_{t-1}(L), l_t).\end{aligned}$$

We use $\mathcal{A}(L) = (\mathcal{A}_0(L), \mathcal{A}_1(L), \ldots, \mathcal{A}_T(L))$ when T is clear from context.

Let $\pi_0, \ldots, \pi_T \in \Pi$ be a sequence of T distributions and let L be a T-row loss matrix. We define the quantities:

$$\begin{aligned}\lambda(\pi, l) &= \sum_{j=1}^k \pi^j l^j, \\ \lambda(\pi_0, \ldots, \pi_T, L) &= \frac{1}{T}\sum_{t=1}^T \lambda(\pi_t, l_t), \\ \lambda(\mathcal{A}(L'), L) &= \lambda(\mathcal{A}_0(L'), \mathcal{A}_1(L'), \ldots, \mathcal{A}_T(L'), L).\end{aligned}$$

Note that the notation retains the flexibility to run the algorithm $\mathcal{A}$ on one loss matrix, but measure the loss $\mathcal{A}$ incurs on a different loss matrix. This flexibility will be useful later.

Let $\mathcal{F}$ be a family of functions $f \colon \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$. For a function f and a distribution π, we define the distribution $f \circ \pi$ to be

$$(f \circ \pi)^j = \sum_{j' : f(j') = j} \pi^{j'}.$$

The distribution $f \circ \pi$ corresponds to the distribution on actions obtained by first choosing an action according to π, then applying the function f.

Now we define the following quantities:

$$\begin{aligned}\lambda(\pi_1, \ldots, \pi_T, L, f) &= \lambda(f \circ \pi_1, f \circ \pi_2, \ldots, f \circ \pi_T, L), \\ \rho(\mathcal{A}, L, f) &= \lambda(\mathcal{A}, L) - \lambda(\mathcal{A}, L, f), \\ \rho(\mathcal{A}, L, \mathcal{F}) &= \max_{f \in \mathcal{F}} \rho(\mathcal{A}, L, f).\end{aligned}$$

As a mnemonic, we offer the following. λ refers to expected loss, ρ refers to regret. Next, we define the families $\mathcal{F}_{\mathsf{fixed}}, \mathcal{F}_{\mathsf{swap}}$:

$$\begin{aligned}\mathcal{F}_{\mathsf{fixed}} &= \left\{f_j(j') = j, \text{ for all } j' \mid j \in \{1, 2, \ldots, k\}\right\} \\ \mathcal{F}_{\mathsf{swap}} &= \{f : \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}\}\end{aligned}$$

Looking ahead, we will need to be able to handle not just *a priori fixed* sequences of losses, but also adapted. To see why, note that for a game setting, a player's loss will depend on the distribution of actions played by everyone in that period, which will depend, in turn, on the losses everyone experienced in the previous period and how everyone's algorithms reacted to that.

DEFINITION 6 (Adapted Loss). *A loss function $\mathcal{L}$ is said to be adapted to an algorithm $\mathcal{A}$ if in each period t, the experienced losses $l_t \in [0,1]^k$ can be written as:*

$$l_t = \mathcal{L}(l_0, \mathcal{A}(l_0), l_1, \mathcal{A}(l_1), \ldots, l_{t-1}, \mathcal{A}(l_{t-1})).$$

We will make use of the following well-known results from the theory of no-regret algorithms, which show the existence of algorithms that guarantee low regret even against adapted losses (see e.g. [33]).

THEOREM 4. *There exists an algorithm $\mathcal{A}_{\mathsf{fixed}}$ such that for any adapted loss $\mathcal{L}$, $\rho(\mathcal{A}_{\mathsf{fixed}}, \mathcal{L}, \mathcal{F}_{\mathsf{fixed}}) \leq \sqrt{\frac{2\log k}{T}}$. There also exists an algorithm $\mathcal{A}_{\mathsf{swap}}$ such that $\rho(\mathcal{A}_{\mathsf{swap}}, \mathcal{L}, \mathcal{F}_{\mathsf{swap}}) \leq k\sqrt{\frac{2\log k}{T}}$.*

2.2 From No Regret to Equilibrium

Let $(u_1, \ldots, u_n)$ be utility functions for each of n players. Let $S = \{(\pi_{i,1}, \ldots, \pi_{i,T})\}_{i=1}^n$ be a collection of n sequences of distributions over k actions, one for each player. Let $\{(l_{i,1}, \ldots, l_{i,T})\}_{i=1}^n$ be a collection of n sequences of loss vectors $l \in [0,1]^k$ formed by the action distribution. More formally, for every j, $l_{i,t}^j = 1 - \mathbb{E}_{\pi_{-i,t}}[u_i(j, a_{-i})]$. Define the maximum regret that any player has to her losses

$$\rho_{\mathsf{max}}(S, L, \mathcal{F}) = \max_i \rho(S_i, L_i, \mathcal{F})$$

where $S_i = (\pi_{i,0}, \ldots, \pi_{i,T})$ and $L_i = (l_{i,1}, \ldots, l_{i,T})$.

Given the collection S, we define the correlated action distribution Π_S be the average distribution of play. That is, Π_S is the distribution over A^n defined by the following sampling procedure: Choose t uniformly at random from $\{1, 2, \ldots, T\}$, then, for each player i, choose a_i randomly according to the distribution $\pi_{i,t}$, independently of the other players.

The following well known theorem (see, e.g. [33]) relates ρ_{max} to the equilibrium concepts (Definitions 2 and 3):

THEOREM 5. *If the maximum regret with respect to $\mathcal{F}_{\mathsf{fixed}}$ is small, i.e. $\rho_{\mathsf{max}}(S, L, \mathcal{F}_{\mathsf{fixed}}) \leq \alpha$, then the correlated action distribution Π_S is an α-approximate coarse correlated equilibrium. Similarly, if $\rho_{\mathsf{max}}(S, L, \mathcal{F}_{\mathsf{swap}}) \leq \alpha$, then Π_S is an α-approximate correlated equilibrium.*

3 Noise Tolerance of No-Regret Algorithms

In this section we show that no-regret algorithms are tolerant to addition of 'some' noise, that is we still get good regret bounds with respect to the real losses if we run the no-regret algorithm on noisy losses (real losses plus low-magnitude noise).

Let $L \in [0,1]^{T \times k}$ be a loss matrix. Define $\overline{L} = \frac{L+1}{3}$ (entrywise) and note that $\overline{L} \in [\frac{1}{3}, \frac{2}{3}]^{T \times k}$. The following states that running $\mathcal{A}$ on $\overline{L}$ doesn't significantly increase the regret with respect to the real losses.

LEMMA 1. *For every algorithm $\mathcal{A}$, every family $\mathcal{F}$, and every loss matrix $L \in [0,1]^{T\times k}$,*

$$\rho(\mathcal{A}(\overline{L}), L, \mathcal{F}) \leq 3\rho(\mathcal{A}(\overline{L}), \overline{L}, \mathcal{F}).$$

In particular, for every $L \in [0,1]^{T\times k}$

$$\rho(\mathcal{A}_{\mathsf{fixed}}(\overline{L}), L, \mathcal{F}_{\mathsf{fixed}}) \leq \sqrt{\frac{18\log k}{T}} \;\; \textit{and} \;\; \rho(\mathcal{A}_{\mathsf{swap}}(\overline{L}), L, \mathcal{F}_{\mathsf{swap}}) \leq k\sqrt{\frac{18\log k}{T}}.$$

PROOF. Let $\pi_0, \ldots, \pi_T \in \Pi_k$ be any sequence of distributions and let $f\colon \{1,2,\ldots,k\} \to \{1,2,\ldots,k\}$ be any function. Then

$$\begin{aligned}
\rho(\pi_0, \ldots, \pi_T, L, f) &= \lambda(\pi_0, \ldots, \pi_T, L) - \lambda(f\circ\pi_0, \ldots, f\circ\pi_T, L) \\
&= 3\left(\lambda(\pi_0, \ldots, \pi_T, \overline{L}) - \lambda(f\circ\pi_0, \ldots, f\circ\pi_T, \overline{L})\right) \\
&= 3\left(\rho(\pi_0, \ldots, \pi_T, \overline{L}, f)\right).
\end{aligned}$$

The second equality follows from the definition of λ and from linearity of expectation. The Lemma now follows by setting $(\pi_0, \ldots, \pi_T) = \mathcal{A}_T(\overline{L})$, taking a maximum over $f \in \mathcal{F}$, and plugging in the guarantees of Theorem 4. □

In light of Lemma 1, for the rest of this section we will take L to be a loss matrix in $[\frac{1}{3}, \frac{2}{3}]^{T\times k}$. This rescaling will only incur an additional factor of 3 in the regret bounds we prove. Let $Z \in \mathbb{R}^{T\times k}$ be a real valued *noise matrix*. Let $\widehat{L} = L + Z$ (entrywise). In the next section we will consider the case where Z is an arbitrary matrix with bounded entries. Then we will prove a tighter bound for the case where Z consists of independent draws from a Laplace distribution.

3.1 General Noise

The next lemma states that when a no-regret algorithm is run on a noisy sequence of losses, it does not incur too much additional regret with respect to the real losses.

LEMMA 2 (Regret Bounds in the Presence of Bounded Noise). *Let $L \in [\frac{1}{3}, \frac{2}{3}]^{T\times k}$ be any loss matrix. Let $Z = (z_t^j) \in [-b, b]^{T\times k}$ be an arbitrary matrix with bounded entries, and let $\widehat{L} = L + Z$. Let $\mathcal{A}$ be an algorithm. Let $\mathcal{F}$ be any family of functions. Then*

$$\rho(\mathcal{A}(\widehat{L}), L, \mathcal{F}) \leq \rho(\mathcal{A}(\widehat{L}), \widehat{L}, \mathcal{F}) + 2b.$$

PROOF. Let $(\pi_0, \ldots, \pi_T)$ be any sequence of distributions and let $f\colon \{1,2,\ldots,k\} \to \{1,2,\ldots,k\}$

be any function. Then:

$$
\begin{aligned}
&\rho(\pi_0,\dots,\pi_T,L,f)-\rho(\pi_0,\dots,\pi_T,\widehat{L},f)\\
&=(\lambda(\pi_0,\dots,\pi_T,L)-\lambda(f\circ\pi_0,\dots,f\circ\pi_T,L))-(\lambda(\pi_0,\dots,\pi_T,\widehat{L})-\lambda(f\circ\pi_0,\dots,f\circ\pi_T,\widehat{L})).\\
&=(\lambda(\pi_0,\dots,\pi_T,L)-\lambda(\pi_0,\dots,\pi_T,\widehat{L}))+(\lambda(f\circ\pi_0,\dots,f\circ\pi_T,\widehat{L})-\lambda(f\circ\pi_0,\dots,f\circ\pi_T,\widehat{L}))\\
&=\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j(l_t^j-\widehat{l}_t^j)\right)+\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)^j(l_t^j-\widehat{l}_t^j)\right) && \text{(by definition of } \lambda)\\
&=\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K}\pi_t^j z_t^j\right)+\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)^j z_t^j\right) && \text{(by definition of } z) \qquad (2)\\
&\le b\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K}\pi_t^j\right)+b\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K}(f\circ\pi_t)^j\right) && (\forall j,t\ |z_t^j|\le b)\\
&=2b,
\end{aligned}
$$

where the final equality follows from the fact that $\pi_t, f\circ\pi_t$ are probability distributions. □

COROLLARY 1. *Let $L\in[\frac{1}{3},\frac{2}{3}]^{T\times k}$ be any loss matrix and let $Z\in\mathbb{R}^{T\times k}$ be a random matrix such that $\mathbb{P}_Z\left[Z\in[-b,b]^{T\times k}\right]\ge 1-\beta$ for some $b\in[0,\frac{1}{3}]$, and let $\widehat{L}=L+Z$. Then*

1. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}),L,\mathcal{F}_{\mathsf{fixed}})>\sqrt{\frac{2\log k}{T}}+2b\right]\le\beta$
2. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{swap}}(\widehat{L}),L,\mathcal{F}_{\mathsf{swap}})>k\sqrt{\frac{2\log k}{T}}+2b\right]\le\beta$

3.2 Laplacian Noise

Having handled the case of general noise, we will now prove a tighter bound on the additional regret in the case where the entries of Z are iid samples from a Laplace distribution.

LEMMA 3 (Regret Bounds for Laplace Noise). *Let $L\in[\frac{1}{3},\frac{2}{3}]^{T\times k}$ be any loss matrix. Let $Z=(z_t^j)\in\mathbb{R}^{T\times k}$ be a random matrix formed by taking each entry to be an independent sample from $\mathrm{Lap}(\sigma)$, and let $\widehat{L}=L+Z$. Let $\mathcal{A}$ be an algorithm. Let $\mathcal{F}$ be any family of functions. Then for any $\eta\le\sigma$.*

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}(\widehat{L}),L,\mathcal{F})-\rho(\mathcal{A}(\widehat{L}),\widehat{L},\mathcal{F})>\eta\right]\le 2|\mathcal{F}|e^{-\eta^2T/24\sigma^2}.$$

PROOF. Let $(\pi_0,\dots,\pi_T)$ be any sequence of distributions and let $f\colon\{1,2,\dots,k\}\to\{1,2,\dots,k\}$ be any function. Recall by (2),

$$\rho(\pi_0,\dots,\pi_T,L,f)-\rho(\pi_0,\dots,\pi_T,\widehat{L},f)=\left(\tfrac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j z_t^j\right)+\left(\tfrac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)_j z_t^j\right). \quad (3)$$

We wish to place a high probability bound on the quantities:

$$Y_{\pi_0,\ldots,\pi_T} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j z_t^j.$$

Changing the order of summation,

$$Y_{\pi_0,\ldots,\pi_T} = \sum_{a_1,\ldots,a_T\in A}\left(\prod_{t=1}^{T}\pi_t^{a_t}\right)\left(\frac{1}{T}\sum_{t=1}^{T}z_t^{a_t}\right),$$

the equality follows by considering the following two ways of sampling elements z_t^j. The first expression represents the expected value of z_t^j if t is chosen uniformly from $\{1,2,\ldots,T\}$ and then j is chosen according to π_t. The second expression represents the expected value of z_t^j if $(a_1,\ldots,a_T)$ are chosen independently from the product distribution $\pi_1\times\pi_2\times\cdots\times\pi_T$ and then a_t is chosen uniformly from $(a_1,\ldots,a_T)$. These two sampling procedures induce the same distribution, and thus have the same expectation. Thus we can write:

$$\mathop{\mathbb{P}}_{Z}\left[Y_{\pi_0,\ldots,\pi_T} > \eta\right] \le \max_{a_1,\ldots,a_T\in A}\mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T}z_t^{a_t} > \eta\right] \le \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T}z_t^{1} > \eta\right].$$

where the second inequality follows from the fact that the variables z_t^j are identically distributed. Applying Theorem 3, we have that for any $\eta < \sigma$,

$$\mathop{\mathbb{P}}_{Z}\left[Y_{\pi_0,\ldots,\pi_T} > \eta\right] \le e^{-\eta^2 T/6\sigma^2}. \tag{4}$$

Let $(\pi_0,\ldots,\pi_T) = \mathcal{A}(\widehat{L})$. By Equation (3) we have

$$\begin{aligned}
&\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}(\widehat{L}),L,f) - \rho(\mathcal{A}(\widehat{L}),\widehat{L},f) > \eta\right]\\
&\le \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j z_t^j > \eta/2\right] + \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)_j z_t^j > \eta/2\right],\\
&\le 2e^{-\eta^2 T/24\sigma^2},
\end{aligned}$$

where the last inequality follows from applying (4) to the sequences $(\pi_0,\ldots,\pi_T)$ and $(f\circ\pi_0,\ldots,f\circ\pi_T)$. The Lemma now follows by taking a union bound over $\mathcal{F}$. □

Finally, we obtain a tighter counterpart of Corollary 1 when the noise is independent Laplacian noise.

COROLLARY 2. *Let $L\in[\frac{1}{3},\frac{2}{3}]^{T\times k}$ be any loss matrix and let $Z\in\mathbb{R}^{T\times k}$ be a random matrix formed by taking each entry to be an independent sample from $\mathrm{Lap}(\sigma)$ for $\sigma < \frac{1}{6\log(4KT/\beta)}$ and let $\widehat{L} = L+Z$. Then*

1. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}),L,\mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24\log(4k/\beta)}{T}}\right] \le \beta,$
2. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{swap}}(\widehat{L}),L,\mathcal{F}_{\mathsf{swap}}) > k\sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24k\log(4k/\beta)}{T}}\right] \le \beta.$

4 Private Equilibrium Computation

Having demonstrated the noise tolerance of no-regret algorithms, we now argue that for appropriately chosen noise, the output of the algorithm constitutes a jointly-differentially private mechanism, in the sense of Definition 5. We prove two results of this type. First, in Section 4.1 we consider games with 'few' actions per player. While our algorithm for this case is conceptually more straightforward, it will not be sufficient in certain cases of interest. For example, in the routing games we described in the introduction, the set of actions available to a player consists of all routes between her starting point and her destination. Even if the graph (road network) is small, the number of feasible routes can be extremely large (exponential in the number of edges (roads)). However, in such games, the set of types (utility functions) is small (i.e. the set of all source-destination pairs). Motivated by this observation, in Section 4.2 we consider games with large action spaces, but bounded type spaces.

4.1 Games with Few Actions

To orient the reader at a high-level, our proof has two main steps. First, we construct a 'wrapper' NRLAPLACE$^{\mathcal{A}}$ which takes as input the parameters of the game, the reported tuple of utilities, and any no-regret algorithm $\mathcal{A}$. This wrapper runs the no-regret algorithm $\mathcal{A}$ in every period for each player on noisy losses, i.e. instead of reporting the true loss to $\mathcal{A}$, it reports the loss plus appropriately chosen Laplacian noise. In Theorem 6 we show that this constitutes a jointly differentially private mechanism. Then, in Theorem 7 and Corollary 3, we show that this wrapper converges to an approximate coarse correlated equilibrium when the input algorithm is $\mathcal{A}_{\mathsf{fixed}}$, and to an approximate correlated equilibrium when the input algorithm is $\mathcal{A}_{\mathsf{swap}}$.

4.1.1 Noisy No-Regret Algorithms are Differentially Private

NRLAPLACE$^{\mathcal{A}}(u_i, \dots u_n)$

PARAMS: $\varepsilon, \delta, \gamma \in (0, 1], n, k, T \in \mathbb{N}$

LET: $\pi_{1,1}, \dots, \pi_{n,1}$ each be the uniform distribution over $\{1, 2, \dots, k\}$.

LET: $\sigma = \frac{\gamma\sqrt{8nkT\ln(1/\delta)}}{\varepsilon}$

FOR: $t = 1, 2, \dots, T$

  LET: $l^j_{i,t} = 1 - \mathbb{E}_{\pi_{-i,t}}[u_i(j, a_{-i})]$ for every player i, action j.

  LET: $z^j_{i,t}$ be an i.i.d. draw from $\mathrm{Lap}(\sigma)$ for every player i, action j.

  LET: $\widehat{l}^j_{i,t} = l^j_{i,t} + z^j_{i,t}$ for every player i, action j.

  LET: $\pi_{i,t+1} = \mathcal{A}(\pi_{i,t}, \widehat{l}_{i,t})$ for every player i.

END FOR

OUTPUT: $(\pi_{i,1}, \dots, \pi_{i,T})$ to player i, for every i.

THEOREM 6 (Privacy of NRLAPLACE$^{\mathcal{A}}$). *For any $\mathcal{A}$, the algorithm* NRLAPLACE$^{\mathcal{A}}$ *satisfies (ε, δ)-joint differential privacy.*

We now sketch the proof. We'll fix a player i and utility functions u_{-i} and argue that the output to all other players is DP with respect to u_i. It will be easier to analyze a modified mechanism that outputs $(\widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,T})$. This output is sufficient to compute $(\pi_{-i,1}, \dots, \pi_{-i,T})$ just by running $\mathcal{A}$, so it is sufficient to prove that this output is DP.

Our goal will be to show that each element of the output, $l_{i',t}$ can be viewed as a γ-sensitive query on u_i. Since $\widehat{l}_{i',t}$ is $l_{i',t}$ plus Laplacian noise, it will satisfy differential privacy (for some suitable parameters). Notice that $l_{i',t}$ depends on the utility function u_i in two ways. The first is explicitly, through the action of player i. This can vary arbitrarily with u_i, but the loss is only γ-sensitive to this action. The second is indirect, in that player i's utility function affects the other players' losses, which will in turn affect the query we make. However, once we fix the noisy losses for the first $t-1$ rounds, we can compute player i's actions in each round, and then only have to worry about the first effect. Thus, we can view the output of our mechanism as T rounds of (possibly adaptively-chosen) low-sensitivity queries on u_i, and apply standard composition arguments in that setting.

PROOF. Fix any player i, any pair of utility functions for i, u_i, u_i', and a tuple of utility functions u_{-i} for everyone else. To show differential privacy, we need to analyze the change in the distribution of the joint output for all players other than i, $(\pi_{-i,1}, \ldots, \pi_{-i,T})$ when the input is (u_i, u_{-i}) as opposed to (u_i', u_{-i}).

It will be easier to analyze the privacy of a modified mechanism that outputs $(\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,T})$. Observe that this output is sufficient to compute $(\pi_{-i,1}, \ldots, \pi_{-i,T})$ just by running $\mathcal{A}$. Thus, if we can show the modified output satisfies differential privacy, then same must be true for the mechanism as written.

For every player $i' \neq i$, action $j \in \{1, 2, \ldots, k\}$, and $t \leq T$, we define the query $\mathcal{Q}^j_{i',t}(\cdot \mid \widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1})$ on the utility functions u_i, as well as u_{-i} the output of the mechanism in rounds $1, \ldots, t-1$.

Query $\mathcal{Q}^j_{i',t}(u_i, u_{-i} \mid \widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1})$

Using u_{-i}, u_i and $\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}$, compute $l^j_{i',t}$. Observe that this can be done in the following steps:

1. Using $\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}$, $\mathcal{A}$, and u_{-i}, compute $\pi_{-i,1}, \ldots, \pi_{-i,t-1}$.
2. Using $\pi_{-i,1}, \ldots, \pi_{-i,t-1}$, $\mathcal{A}$, and u_i, compute $\pi_{i,1}, \ldots, \pi_{i,t-1}$.
3. Using $\pi_{t-1} = (\pi_{i,t-1}, \pi_{-i,t-1})$, $\mathcal{A}$, and u_i, compute $l^j_{i',t}$.

Observe that the only step of the query computation that directly involves u_i is the second. Changing player i's utility function from u_i to u_i' can (potentially) affect $\pi_{i,t-1}$, and can (potentially) alter it to an arbitrary state $\overline{\pi}_{i,t-1}$. However, observe that

$$\mathcal{Q}^j_{i',t}(u_i \mid u_{-i}, \widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}) = 1 - \mathop{\mathbb{E}}_{\pi_{-i',t}}[u_{i'}(j, a_{-i'})]$$

$$= 1 - \mathop{\mathbb{E}}_{\pi_{-(i,i'),t}}\left[\mathop{\mathbb{E}}_{\pi_{i,t}}\left[u_{i'}(j, a_i, a_{-(i',i)})\right]\right] \leq$$

$$1 - \mathop{\mathbb{E}}_{\pi_{-(i',i),t}}\left[\mathop{\mathbb{E}}_{\overline{\pi}_{i,t}}\left[u_{i'}(j, a_i, a_{-(i',i)}) + \gamma\right]\right] =$$

$$Q^j_{i',t}(u_i', \mid u_{-i}, \widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}) + \gamma,$$

where the inequality comes from the fact that $u_{i'}$ is assumed to be γ-sensitive in the action of player i (Definition 1), and by linearity of expectation. A similar argument shows:

$$\mathcal{Q}^j_{i',t}(u_i \mid u_{-i}\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}) \geq Q^j_{i',t}(u'_i \mid u_{-i}\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}) - \gamma.$$

Note two facts about these queries: (1) The answer to $Q^j_{i',t}$ is exactly $l^j_{i',t}$, thus the noisy output to these queries (i.e. answer plus $\mathrm{Lap}(\sigma)$) is indeed equal to the output of the (modified) algorithm NR-LAPLACE$^{\mathcal{A}}$. (2) The noisy losses $\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}$ have already been computed when the mechanism reaches round t, thus the mechanism fits the definition of adaptive composition.

Thus, we have rephrased the output $(\widehat{l}_{i',1}, \ldots, \widehat{l}_{i',T})$ as computing the answers to nkT (adaptively chosen) queries on $(u_1, \ldots, u_n)$, each of which is γ-sensitive to the input u_i. Thus the Theorem follows from our choice of $\sigma = \gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)}$ and Theorems 1 and 2. □

4.1.2 Noisy No-Regret Algorithms Compute Approximate Equilibria

Therefore we have shown how that the this 'wrapper' algorithm is jointly differentially private in the sense of Definition 5. We now proceed to show that using this algorithm with $\mathcal{A}_{\mathsf{fixed}}$ will result in an approximate coarse correlated equilibrium (Theorem 7), and that using it with $\mathcal{A}_{\mathsf{swap}}$ will result in an approximate correlated equilibrium (Corollary 3).

THEOREM 7 (Computing CCE). *Let $\mathcal{A} = \mathcal{A}_{\mathsf{fixed}}$. Fix the environment, i.e. the number of players n, the number of actions k, the sensitivity of the game γ, the degree of privacy desired, (ϵ, δ), and the failure probability β. One can then select the number of rounds the algorithm must run, T, satisfying:*

$$\gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)} \leq \frac{1}{6\log(4nkT/\beta)}, \tag{5}$$

such that with probability at least $1-\beta$, the algorithm NRLAPLACE$^{\mathcal{A}_{\mathsf{fixed}}}$*, returns an α-approximate CCE for:*[9]

$$\alpha = \tilde{O}\left(\gamma\varepsilon^{-1}\sqrt{nk\log(1/\delta)}\log(1/\beta)\right). \tag{6}$$

Before we proceed to the proof, some discussion is appropriate. It is already well known that no-regret alogrithms converge 'quickly' to approximate equilibria– recall Theorems 4 and 5. In the previous section, we showed that adding noise still leads to low regret (and therefore to approximate equilibrium). The tradeoff therefore is this. To get a more 'exact' equilibrium, the algorithm has to be run for more rounds. By the arguments in Theorem 6, this will result in a less private outcome. The current theorem makes precise the tradeoff between the two. Fixing the various parameters, (5) tells us the number of rounds T the algorithm must run for. Then, (6) tell us that fixing the desired privacy and failure probability, one can compute an α-approximate CCE for $\alpha = \tilde{O}(\gamma\sqrt{nk})$.

This is a *strongly positive* result– in several large games of interest, e.g. anonymous matching games, $\gamma = O(n^{-1})$. Therefore, for games of this sort $\alpha = \tilde{O}(\sqrt{k}/\sqrt{n})$. If k is fixed, but n is large, therefore, a relatively exact equilibrium of the underlying game can be implemented, while still being jointly differentially private to the desired degree.

PROOF OF THEOREM 7. By our choice of the parameter σ, in the algorithm NRLAPLACE$^{\mathcal{A}_{\mathsf{all}}}$, which is

$$\sigma = \gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)},$$

[9] Here $\tilde{O}$ hides (lower order) $\mathrm{poly}(\log n, \log k, \log T, \log(1/\gamma), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ factors.

and by assumption of the theorem, (5), we have $\sigma \leq 1/6\log(4nkT/\beta)$. Applying Theorem 2 we obtain:

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\pi_{i,1},\ldots,\pi_{i,T},L_i,\mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + \sigma\frac{\sqrt{24\log(4nk/\beta)}}{T}\right] \leq \frac{\beta}{n}$$

for any player i, where L_i is the loss matrix derived from the given utility functions u_i and the distributions $\{\pi_{i,t}\}_{i\in[n],t\in[T]}$. Now we can take a union bound over all players i, yielding:

$$\mathop{\mathbb{P}}_{Z}\left[\max_i \rho(\pi_{i,1},\ldots,\pi_{i,T},L_i,\mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + \sigma\frac{\sqrt{24\log(4nk/\beta)}}{T}\right] \leq \beta,$$

$$\implies \mathop{\mathbb{P}}_{Z}\left[\rho_{\mathsf{max}}(\pi,L,\mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + \sigma\frac{\sqrt{24\log(4nk/\beta)}}{T}\right] \leq \beta.$$

By Theorem 5, therefore, the empirical distribution of play is a $\sqrt{\frac{2\log k}{T}} + \sigma\frac{\sqrt{24\log(4nk/\beta)}}{T}$-approximate coarse correlated equlibrium.

To finish, substitute $\sigma = \gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)}$ into the expression above. Therefore, with probability at least $1-\beta$, no player has regret larger than

$$\alpha = \sqrt{\frac{2\log k}{T}} + \frac{\gamma\sqrt{192nk\log(1/\delta)\log(4nk/\beta)}}{\varepsilon}$$

Since T is a parameter of the algorithm, we can choose T to minimize α. Since α is monotonically decreasing in T, we would like to choose T as large as possible. However, our argument requires (5), which (roughly) requires $\sqrt{T} \lesssim 1/\gamma\sqrt{nk}$, where we have suppressed dependence on some of the parameters. By choosing T so that $\sqrt{T} \sim 1/\gamma\sqrt{nk}$ we can make the first term of the error $\sim \gamma\sqrt{nk}$, which would make it be of a similar order to the second term. It is easy to verify that we can choose T is such a way that T satisfies the assumption and the resulting value of α satisfies the conclusion of the theorem. □

By considering $\mathcal{A}_{\mathsf{swap}}$ instead of $\mathcal{A}_{\mathsf{fixed}}$, we easily get similar results for approximate correlated equilibrium rather than coarse correlated equilibrium.

COROLLARY 3 (Computing CE). *Let $\mathcal{A} = \mathcal{A}_{\mathsf{swap}}$. Fix the environment, i.e. the number of players n, the number of actions k, the sensitivity of the game γ, and the degree of privacy desired, (ϵ,δ). One can then select the number of rounds the algorithm must run T, and two numbers α, β satisfying:*

$$\gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)} \leq \frac{1}{6\log(4nkT/\beta)}, \tag{7}$$

such that probability at least $1-\beta$, the algorithm NRLAPLACE$^{\mathcal{A}_{\mathsf{swap}}}$*, returns an α-approximate correlated equilibrium for:*[10]

$$\alpha = \tilde{O}\left(\frac{\gamma k^{3/2}\sqrt{n\log(1/\delta)}\log(1/\beta)}{\varepsilon}\right)$$

[10] Again $\tilde{O}$ hides lower order $\mathrm{poly}(\log N, \log K, \log T, \log(1/\Delta), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ factors.

PROOF. Following the same steps as the Proof of Theorem 7, but noting that we are using regret with respect to $\mathcal{F}_{\mathsf{swap}}$ rather than $\mathcal{F}_{\mathsf{fixed}}$, we find that NRLAPLACE$^{\mathcal{A}_{\mathsf{swap}}}$ will return, with probability at least $1-\beta$, an α-approximate correlated equlibrium where

$$\alpha = k\sqrt{\frac{2\log k}{T}} + \frac{\Delta k\sqrt{384n\log(1/\delta)\log(4kn/\beta)}}{\varepsilon}.$$

As in Theorem 7, we will choose $T \sim 1/\gamma\sqrt{nk}$ to complete the proof. □

4.2 Upper bounds for Games with Bounded Type Spaces

Recall that in the previous section, we showed that a private equilibrium can be computed with a $O(\sqrt{k}/\sqrt{n})$ approximate equilibrium. While these results are positive for some settings (e.g. anonymous matching games for large populations), they have no bite in settings where the number of actions is as large (or larger) than the number of players. The problem is roughly this– with large numbers of actions, the no-regret algorithm will have to be run 'many' times. This would require that we either sacrifice privacy, or introduce even more noise to ensure privacy, which in turn would give make the computed equilibrium a worse approximation.

4.2.1 The Median Mechanism

In order to get a better bound on the accuracy as a function of the number of queries, we will need a mechanism that is capable of answering a large number of queries accurately. One such mechanism is the so-called Median Mechanism of Roth and Roughgarden [39], paired with the privacy analysis of Hardt and Rothblum [24].[11]

THEOREM 8 (Median Mechanism For General Queries [39, 24]). *Consider the following R-round experiment between a mechanism $\mathcal{M}_M$, who holds a tuple $u_1, \ldots, u_N \in \mathcal{U}$, and a adaptive querier $\mathcal{B}$. For every round $r = 1, 2, \ldots, R$:*

1. *$\mathcal{B}(Q_1, a_1, \ldots, Q_{r-1}, a_{r-1}) = Q_r$, where Q_r is a γ-sensitive query.*
2. *$a_r \leftarrow_R \mathcal{M}_M(u_1, \ldots, u_n; Q_r)$.*

For every $\varepsilon, \delta, \gamma, \beta \in (0, 1], N, R, U \in \mathbb{N}$, there is a mechanism $\mathcal{M}_M$ such that for every $\mathcal{B}$

1. *The transcript $(Q_1, a_1, \ldots, Q_R, a_R)$ satisfies (ε, δ)-differential privacy.*
2. *With probability $1-\beta$ (over the randomizations of $\mathcal{M}_M$), $|a_r - Q_r(u_1, \ldots, u_N)| \leq \alpha_{\mathcal{M}_M}$ for every $r = 1, 2, \ldots, R$ and for*

$$\alpha_{\mathcal{M}_M} = 16\varepsilon^{-1}\gamma\sqrt{N\log U}\log(2R/\beta)\log(4/\delta).$$

[11] Originally, the median mechanism of [39] was only defined and analyzed for the case of linear queries. A 'folk' result, first observed by Hardt and Rothblum [23] is that the Median Mechanism (when instantiated with a net of all possible size n datasets) can be applied to arbitrary γ-sensitive queries, which immediately yields Theorem 8 when paired with the privacy analysis of [24]. The simple proof can be found in [12].

4.2.2 Noisy No-Regret via the Median Mechanism

We now define our algorithm for computing equilibria in games with exponentially many actions.

To keep notation straight, we will use $\mathbf{u} = (u_1, \ldots, u_N)$ to denote the utility functions specified by each of the n players, and $v \in \mathcal{U}$ to denote a utility function considered within the mechanism. Let $U = |\mathcal{U}|$, the size of the set of possible utility functions for any player.

First we sketch some intuition for how the mechanism works. In particular, why we cannot simply substitute the Median Mechanism for the Laplace mechanism and get a better error bound. Recall the queries we used in analyzing the Laplace-based algorithm $Q^j_{i,t}(\cdot \mid u_{-i}, \widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,T})$ in our previous analysis. We were able to argue that fixing u_{-i} and the previous noisy losses, the query was low-sensitivity as a function of its input u_i. This argument relied on the fact that we were effectively running *independent* copies of the Laplace mechanism, which guarantees that the answers given to each query do not explicitly depend on the previous queries that were asked (although the queries themselves may be correlated). However, in the mechanism we are about to define, the queries are all answered using a *single* instantiation of the Median mechanism. The Median mechanism correlates its answers across queries, and thus the answers to one query may depend on the previous queries that were made. This fact will be problematic, because the description of the queries $Q^j_{i,t}$ contains the utility functions u_{-i}. Thus, the queries we made to construct the output for players other than i will actually contain information about u_{-i}, and we cannot guarantee that this information does not leak into the answers given to other sets of players.

We address this problem by asking a larger set of queries whose description does not depend on any particular player's utility function. We will make the set of queries large enough that they will actually contain every query that we might possibly have asked in the Laplace-based algorithm, and each player can select from the larger set of answers only those which she needs to compute her losses. Since the queries do not depend on any utility function, we do not have to worry about leaking the description of the queries.

In order to specify the mechanism it will be easier to define the following family of queries first. Let i be any player, j any action, t any round of the algorithm, and v any utility function. The queries will be specified by these parameters and a sequence $\Lambda_1, \ldots, \Lambda_{t-1}$ where $\Lambda_{t'} \in \mathbb{R}^{n \times k \times \mathcal{U}}$ for every $1 \leq t' \leq t-1$. Intuitively, the query is given a description of the "state" of the mechanism in all previous rounds. Each state variable Λ_t encodes the losses that would be experienced by every possible player i and every action j and every utility function v, *given that the previous $t-1$ rounds of the mechanism were played using the real utility functions*. We will think of the variables $\Lambda_1, \ldots, \Lambda_{t-1}$ as having been previously sanitized, and thus we do not have to worry about the fact that these state variables encode information about the real utility functions.

$\mathcal{Q}^j_{i,t,v}(u_1, \ldots, u_N \mid \Lambda_1, \ldots, \Lambda_{t-1})$

Using $u_1, \ldots, u_N \mid \Lambda_1, \ldots, \Lambda_{t-1}$, compute $l^j_{i,t,v} = 1 - \mathbb{E}_{\pi_{-i,t}}[u_i(j, a_{-i})]$. This computation can be done in the following steps:

1. For every $i' \neq i$, use $\Lambda^j_{i',1,u_{i'}}, \ldots, \Lambda^j_{i',t-1,u_{i'}}$, $\mathcal{A}$, and $u_{i'}$ to compute $\pi_{i',1}, \ldots, \pi_{i',t-1}$.
2. Using $\pi_{-i,t-1}$, compute $l^j_{i,t,v}$.

Observe that $Q^j_{i,t,v}$ is γ-sensitive for every player i, step t, action j, and utility function v. To see

why, consider what happens when a specific player i' switches her input from $u_{i'}$ to $u'_{i'}$. In that case that $i = i'$, this has no effect on the query answer, because player i's utility is never used in computing $Q^i_{t,j,v}$. In the case that $i' \neq i$ then the utility function of player i' can (potentially) affect the computation of $\pi_{i',t-1}$, and can (potentially) change it to an arbitrary state $\overline{\pi}_{i',t-1}$. But then γ-sensitivity follows from the γ-sensitivity of u_i, the definition of $l^j_{i,t,v}$, and linearity of expectation. Notice that $u_{i'}$ does not, however, affect the state of any other players, who will use the losses $\Lambda_1, \ldots, \Lambda_{t-1}$ to generate their states, not the actual states of the other players.

Now that we have this family of queries in places, we can describe the algorithm. Our mechanism uses two steps. At a high level, there is an inner mechanism, NRMEDIAN-SHARED, that will use the Median Mechanism to answer each query $Q^j_{i,t,v}\left(\cdot \mid \widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_{t-1}\right)$, and will output a set of noisy losses $\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T$. The properties of the Median Mechanism will guarantee that these losses satisfy (ε, δ)-differential privacy (in the standard sense of Definition 4).

There is also an outer mechanism that takes these losses and, for each player, uses the losses corresponding to her utility function to run a no-regret algorithm. This is NRMEDIAN which takes the sequence $\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T$ and using the utility function u_i will compute the equilibrium strategy for player i. Since each player's output can be determined only from her own utility function and a set of losses that is (ε, δ)-differentially private with respect to every utility function, the entire mechanism will satisfy (ε, δ)-joint differential privacy.

NRMEDIAN-SHARED$^{\mathcal{A}}(u_1, \ldots u_N)$

PARAMS: $\varepsilon, \delta, \gamma \in (0, 1], n, k, T \in \mathbb{N}$

FOR: $t = 1, 2, \ldots, T$

LET: $\widehat{l}^j_{i,t,v} = \mathcal{M}_M\left(u_1, \ldots, u_N; Q^j_{i,t,v}(\cdot \mid \widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_{t-1})\right)$ for every i, j, v.

LET: $\widehat{\Lambda}^j(i, t, v) = \widehat{l}^j_{i,t,v}$ for every i, j, v.

END FOR

OUTPUT: $(\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T)$.

NRMEDIAN$^{\mathcal{A}}(u_1, \ldots u_N)$

PARAMS: $\varepsilon, \delta, \Delta \in (0, 1], n, k, T \in \mathbb{N}$

LET: $(\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T) =$ NRMEDIAN-SHARED$^{\mathcal{A}}(u_1, \ldots, u_N)$.

FOR: $i = 1, \ldots, N$

LET: $\pi_{i,1}$ be the uniform distribution over $\{1, 2, \ldots, k\}$.

FOR: $t = 1, \ldots, T$

LET: $\pi_{i,t} = \mathcal{A}\left(\pi_{i,t-1}, \widehat{\Lambda}_{i,t-1,u_i}\right)$

END FOR

OUTPUT TO PLAYER i: $(\pi_{i,1}, \ldots, \pi_{i,T})$.

END FOR

THEOREM 9 (Privacy of NRMEDIAN). *The algorithm* NRMEDIAN *satisfies* (ε, δ)*-joint differential privacy.*

PROOF. Observe that NRMEDIAN can be written as $h(\mathbf{u}) = (f_1(g(\mathbf{u})), \ldots, f_N(g(\mathbf{u})))$ where f_i depends only on u_i for every player i. (Here, g is NRMEDIAN-SHARED and f_i is the i-th iteration of

the main loop in NRMEDIAN). The privacy of the Median Mechanism (Theorem 8) directly implies that g is (ε, δ)-differentially private (in the standard sense).

Consider a player i and two profiles $\mathbf{u}, \mathbf{u}'$ that differ only in the input of player i, and consider the output $(f_{-i}(g(\mathbf{u})))$. Let $S \subseteq \text{Range}(f_{-i})$ and let $R(\mathbf{u}) = \left\{o \in \text{Range}(g) \mid f^{-i}(o) \in S\right\}$. Notice that f is deterministic, so R is well-defined. Also notice that R depends only on S and $\mathbf{u}_{-i}$ (in particular, not on u_i). Then we have

$$\begin{aligned}
\mathop{\mathbb{P}}_{h(\mathbf{u})}\left[h^{-i}(\mathbf{u}) \in S\right] &= \mathop{\mathbb{P}}_{g(\mathbf{u})}\left[g(\mathbf{u}) \in R(\mathbf{u}) = R(\mathbf{u}')\right] \\
&\leq e^{\varepsilon} \mathop{\mathbb{P}}_{g(\mathbf{u}')}\left[g(\mathbf{u}') \in R(\mathbf{u}) = R(\mathbf{u}')\right] + \delta \\
&\leq e^{\varepsilon} \mathop{\mathbb{P}}_{h(\mathbf{u})}\left[h^{-i}(\mathbf{u}') \in S\right] + \delta
\end{aligned}$$

where the first inequality follows from the (standard) (ε, δ)-differential privacy of g. Thus, NRMEDIAN satisfies (ε, δ)-joint differential privacy. □

4.2.3 Computing Approximate Equilibria

THEOREM 10 (Computing CCE). *Let $\mathcal{A}$ be $\mathcal{A}_{\text{fixed}}$. Fix the environment, i.e the number of players n, the number of actions k, number of possible utility functions U, sensitivity of the game γ and desired privacy (ϵ, δ). Suppose β and T are such that:*

$$16\varepsilon^{-1}\gamma\sqrt{n \log U}\log(2nkTU/\beta)\log(4/\delta) \leq \tfrac{1}{6} \tag{8}$$

Then with probability at least $1-\beta$ the algorithm $\text{NRMEDIAN}^{\mathcal{A}_{\text{fixed}}}$ *returns an α-approximate CCE for:*[12]

$$\alpha = \tilde{O}\left(\frac{\gamma\sqrt{N}\log^{3/2} U \log(k/\beta)\log(1/\delta)}{\varepsilon}\right).$$

Again, considering 'low sensitivity' games where γ is $O(1/n)$, the theorem says that fixing the desired degree of privacy, we can compute an α-approximate equilibrium for $\alpha = \tilde{O}\left(\frac{(\log U)^{\frac{3}{2}}\log k}{\sqrt{N}}\right)$. The tradeoff to the old results is in dependence on the number of actions. The results in the previous section had a $\sqrt{k}$ dependence on the number of actions k. This would have no bite if k grew even linearly in n. We show that positive results still exist if the number of possible private types is is bounded - the dependence on the number of actions and the number of types is now logarithmic. However this comes with two costs. First, we can only consider situations where the number of types any player could have is bounded, and grows sub-exponentially in n. Second, we lose computational tractability– the running time of the median mechanism is exponential in the number of players in the game.

PROOF. By the accuracy guarantees of the Median Mechanism:

$$\mathop{\mathbb{P}}_{\mathcal{M}_M}\left[\exists i, t, j, v \text{ s.t. } |\widehat{l}^{j}_{i,t,v} - l^{j}_{i,t,v}| > A_{\mathcal{M}_M}\right] \leq \beta$$

[12]Here, $\tilde{O}$ hides lower order $\text{poly}(\log n, \log\log k, \log T, \log\log U \log(1/\gamma), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ terms.

where

$$\alpha_{\mathcal{M}_M} = 16\gamma\varepsilon^{-1}\sqrt{n\log U}\log(2nkTU/\beta)\log(4/\delta)$$

By (8), $\alpha_{\mathcal{M}_M} \leq 1/6$. Therefore,

$$\mathop{\mathbb{P}}_{\mathcal{M}_M}\left[\exists i,j,t,v \text{ s.t. } |\widehat{l}^j_{i,t,v} - l^j_{i,t,v}| > \tfrac{1}{6}\right] \leq \beta$$

Applying Theorem 1 and substituting $A_{\mathcal{M}_M}$, we obtain:

$$\mathop{\mathbb{P}}_{Z}\left[\exists i \text{ s.t. } \rho(\pi_{i,1},\ldots,\pi_{i,T},L,\mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + 2\alpha_{\mathcal{M}_M}\right] \leq \beta$$

Now we can choose $\sqrt{T} = (\gamma\sqrt{n})^{-1}$ to conclude the proof. □

COROLLARY 4 (Computing CE). *Let $\mathcal{A}$ be $\mathcal{A}_{\mathsf{swap}}$. Fix the environment, i.e the number of players n, the number of actions k, number of possible utility functions U, sensitivity of the game γ, the desired privacy (ϵ,δ), and the failure probability β. Suppose T is such that:*

$$16\varepsilon^{-1}\gamma\sqrt{n\log U}\log(2nkTU/\beta)\log(4/\delta) \leq \tfrac{1}{6} \tag{9}$$

Then with probability at least $1-\beta$ the algorithm NRMEDIAN$^{\mathcal{A}_{\mathsf{swap}}}$ *returns an α-approximate CCE for:*[13]

$$\alpha = \tilde{O}\left(\frac{\gamma\sqrt{n}\log^{3/2}U\log(k/\beta)\log(1/\delta)}{\varepsilon}\right)$$

PROOF. By the accuracy guarantees of the Median Mechanism:

$$\mathop{\mathbb{P}}_{\mathcal{M}_M}\left[\exists i,t,j,v \text{ s.t. } |\widehat{l}^j_{i,t,v} - l^j_{i,t,v}| > A_{\mathcal{M}_M}\right] \leq \beta$$

where

$$\alpha_{\mathcal{M}_M} = 16\gamma\varepsilon^{-1}\sqrt{n\log U}\log(2nkTU/\beta)\log(4/\delta)$$

By (9), $\alpha_{\mathcal{M}_M} \leq 1/6$. Therefore,

$$\mathop{\mathbb{P}}_{\mathcal{M}_M}\left[\exists i,j,t,v \text{ s.t. } |\widehat{l}^j_{i,t,v} - l^j_{i,t,v}| > \tfrac{1}{6}\right] \leq \beta$$

Applying Theorem 1 and substituting $\alpha_{\mathcal{M}_M}$, we obtain:

$$\mathop{\mathbb{P}}_{Z}\left[\exists i \text{ s.t. } \rho(\pi_{i,1},\ldots,\pi_{i,T},L,\mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + 2\alpha_{\mathcal{M}_M}\right] \leq \beta$$

Now we can choose $\sqrt{T} = k(\gamma\sqrt{n})^{-1}$ to conclude the proof. □

[13]Here $\tilde{O}$ hides lower order $\mathrm{poly}(\log n, \log\log k, \log T, \log\log U\log(1/\gamma), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ terms.

4.3 A Lower Bound

In the case where $\gamma = O(1/n)$ and $k = O(1)$, both of our algorithms from the previous Section compute a differentially private, α-approximate equilibrium for $\alpha \sim 1/\sqrt{n}$ (ignoring all other parameters). It is natural to ask whether or not we can achieve significantly smaller values of α using some other algorithm. In this section we prove a lower bound showing that this is not the case. Specifically, we show that there is no algorithm that privately computes an α-approximate equilibrium of an arbitrary n-player 2-action game, for $\alpha \ll 1/\sqrt{n \log n}$. In other words, there cannot exist an algorithm that privately computes a 'signficantly' more exact equilibrium.

Our proof is by a reduction to the problem of differentially private *subset-sum query release*, for which strong information theoretic lower bounds are known [7, 14]. The problem is as follows: Consider a database $D \in \{0,1\}^n$, which we denote $(d_1, \ldots, d_n)$. A subset-sum query $q \subseteq [n]$ is defined by a subset of the n database entries and asks "What fraction of the entries in D are contained in q and are set to 1?" Formally, we define the query q as $q(D) = \frac{1}{n}\sum_{i \in q} d_i$. Given a set of subset-sum queries $\mathcal{Q} = \{q_1, \ldots, q_m\}$, we say that an algorithm $\mathcal{M}(D)$ *releases* $\mathcal{Q}$ *to accuracy* α if $\mathcal{M}(D) = (a_1, \ldots, a_m)$ such that $|a_j = q_j(D)| \leq \alpha$ for every $j \in [m]$.

Dinur and Nissim [7], showed that any differentially private algorithm that releases sufficiently many subset-sum queries must add a significant amount of noise. A quantitative improvement of their result is given by Dwork and Yekhanin [14]. They constructed a family $\mathcal{Q}_{\mathsf{DY}}$ of size $m = O(n)$ such that there is no differentially private algorithm that releases $\mathcal{Q}_{\mathsf{DY}}$ to accuracy $o(1/\sqrt{n})$. Thus, a natural approach to proving a lower bound is to show that an algorithm for computing approximate equilibrium in arbitrary games could also be used to release arbitrary sets of subset-sum queries accurately. The following theorem shows that a differentially private mechanism to compute approxmiate equilibrium implies a differentially private algorithm to compute subset-sums.

THEOREM 11. *For any $\alpha > 0$, if there is an (ε, δ)-jointly differentially private mechanism $\mathcal{M}$ that computes an α-approximate coarse correlated equilibria in $(n + m\log n)$-player, 2-action, $1/n$-sensitive games, then there is an (ε, δ)-differentially private mechanism $\mathcal{M}'$ that releases 36α-approximate answers to any m subset-sum queries on a database of size n.*

Applying the results of Dwork and Yekhanin [14], a lower bound on equilibrium computation follows easily.

COROLLARY 5. *Any $(\varepsilon = O(1), \delta = o(1))$-differentially private mechanism $\mathcal{M}$ that computes an α-approximate coarse correlated equilibria in n-player 2-action games with $O(1/n)$-sensitive utility functions must satisfy $\alpha = \Omega(\frac{1}{\sqrt{n \log n}})$.*

Here, we provide a sketch of the proof of Theorem 11. Let $D \in \{0,1\}^n$ be an n-bit database and $\mathcal{Q} = \{q_1, \ldots, q_m\}$ be a set of m subset-sum queries. For the sketch, assume that we have an algorithm that computes exact equilibria. We will split the $(n+m)$ players into n "data players" and m "query players." Roughly speaking, the data players will have utility functions that force them to play "0" or "1", so that their actions actually represent the database D. Each of the query players will represent a subset-sum query q, and we will try to set up their utility function in such a way that it forces them to take an action that corresponds to an approximate answer to $q(D)$. In order to do this, first assume there are $n+1$ possible actions, denoted $\left\{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\right\}$. We can set up the utility function so that for each action a, he receives a payoff that is maximized when an a fraction of the data players in q are playing 1. That is, when playing action a, his payoff is maximized when $q(D) = a$. Conversely, he will play the action a that is closest to the true answer $q(D)$. Thus, we can

read off the answer to q from his equilibrium action. Using each of the m query players to answer a different query, we can compute answers to m queries. Finally, notice that joint differential privacy says that all of the actions of the query players will satisfy (standard) differential privacy with respect to the inputs of the data players, thus the answers we read off will be differentially private (in the standard sense) with respect to the database.

This sketch does not address two important issues. The first is that we do not assume that the algorithm computes an exact equilibrium, only that it computes an approximate equilibrium. This relaxation means that the data players do not have to play the correct bit with probability 1, and the query players do not have to choose the answer that exactly maximizes their utility. In the proof we show that the error in the answers we read off is only a small factor larger than the error in the equilibrium computed.

The second is that we do not want to assume that the (query) players have $n+1$ available actions. Instead, we use $\log n$ players per query, and use each to compute roughly one bit of the answer, rather than the whole answer. However, if the query players' utility actually depends on a specific bit of the answer, then a single data player changing his action might result in a large change in utility. In the proof, we show how to compute bits of the answer using $1/n$-sensitive utility functions.

REMARK 12. We remark that we used $O(n)$ *linear* queries in proving our lower bound, for which a lower bound of $\Omega(1/\sqrt{n})$ is known for (ϵ, δ)-differentially private algorithms. Thus, our $\Omega(1/\sqrt{n \log n})$ lower bound also applies to games with *linear* utility functions. However, stronger lower bounds of $\Omega(1)$ are known for answering $O(n)$ low sensitivity *nonlinear* queries on a binary valued database [6] while preserving $(\epsilon, 0)$-differential privacy. We could equally well use the queries from the lower bound argument of [6] in our construction, to show that no $(\epsilon, 0)$-jointly differentially private algorithm can compute an α-approximate CCE to an n-player, 2-action, sensitivity $1/n$ game for any $\alpha < c$, where c is some fixed universal constant. This proves a strong separation between (ϵ, δ)-private equilibrium computation for $\delta > 0$, and $(\epsilon, 0)$-private equilibrium computation. In particular, with $(\epsilon, 0)$-privacy, it is not possible to compute an approximate equilibrium where the approximation factor tends to 0 with the number of players, and therefore not possible to get the "strategyproofness in the large" results that we are able to obtain when $\delta > 0$.

4.4 Incentive Properties

One of the things touched upon in our introduction was the incentive properties of our proposed mechanism. It is well understood that differentially private mechanisms are also approximately strategy proof (This point was initially made in McSherry and Talwar [32]). This will give us the desired incentive properties in our setting as well. The basic idea is as follows: Fix some player i considering changing his report. Joint differential privacy implies that fixing the reports of the other players, for any report of player i, the distribution over actions suggested to players $-i$ cannot change 'much'. Therefore player i's gain from misreporting must also be small. Formally, we have the following theorem:

THEOREM 13. *Consider a (ϵ, δ)-jointly differentially private mechanism $\mathcal{M}$ which computes a α-correlated equilibrium of the full information game induced by players' reports. Then:*

1. *If all players must follow their recommended actions, then it is a $(e^{\epsilon} - 1) + \delta$-approximate dominant strategy for each player to report their type truthfully.*

2. *It is a $(e^{\epsilon}-1)+\delta+\alpha$-approximate Nash Equilibrium for players to each play the following strategy– "truthfully report your type to the mechanism, then follow the suggested action".*

Part 1 follows easily from the definition of joint differential privacy and the fact that payoffs are bounded between 0 and 1. Part 2 follows since the mechanism suggests an α-approximate correlated equilibrium to the players.

It is easy to select ϵ, δ and α so that the incentive properties are also 'good' for large games. In particular recall that α is $O(\sqrt{\log(1/\delta)}/\epsilon\sqrt{n})$ (Corollary 3). Selecting e.g. ϵ of $O(n^{-1/4})$, and δ of $O(1/n)$, we have α is $\tilde{O}(n^{-1/4})$. Therefore for large n, the loss from privacy and approximation of equilibrium computed by this mechanism will asymptote to 0. Further it will be an almost exact equilibrium for all players to truthfully report their type and then follow the suggested action– the approximation is $\epsilon+\alpha+\delta=\tilde{O}(n^{-1/4})$.

References

[1] N.I. Al-Najjar and R. Smorodinsky. Pivotal players and the characterization of influence. *Journal of Economic Theory*, 92(2):318–342, 2000.

[2] E. Azevedo and E. Budish. Strategyproofness in the large as a desideratum for market design. Technical report, University of Chicago, 2011.

[3] A. Blum, M.T. Hajiaghayi, K. Ligett, and A. Roth. Regret minimization and the price of total anarchy. In *STOC*, pages 373–382. ACM, 2008.

[4] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In Cynthia Dwork, editor, *STOC*, pages 609–618. ACM, 2008.

[5] Yiling Chen, Stephen Chong, Ian A. Kash, Tal Moran, and Salil P. Vadhan. Truthful mechanisms for agents that value privacy. *CoRR*, abs/1111.5472, 2011.

[6] Anindya De. Lower bounds in differential privacy. In *TCC*, pages 321–338, 2012.

[7] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.

[8] C. Dwork. Differential privacy: A survey of results. *Theory and Applications of Models of Computation*, pages 1–19, 2008.

[9] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06*, pages 265–284, 2006.

[11] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of lp decoding. In *STOC*, pages 85–94, 2007.

[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. IEEE Computer Society, 2010.

[14] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *CRYPTO*, pages 469–480, 2008.

[15] Lisa Fleischer and Yu-Han Lyu. Approximately optimal auctions for selling privacy when costs are correlated with data. In *EC*, pages 568–585, 2012.

[16] D.P. Foster and R.V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2):40–55, 1997.

[17] Arpita Ghosh and Aaron Roth. Selling privacy at auction. In *EC*, pages 199–208, 2011.

[18] R. Gradwohl. Privacy in implementation. 2012.

[19] R. Gradwohl and O. Reingold. Fault tolerance in large games. In *EC*, pages 274–283. ACM, 2008.

[20] R. Gradwohl and O. Reingold. Partial exposure in large games. *Games and Economic Behavior*, 68(2):602–613, 2010.

[21] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *STOC '11*, pages 803–812, 2011.

[22] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *TCC*, pages 339–356, 2012.

[23] Moritz Hardt. Personal communication.

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

[25] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[26] Zhiyi Huang and Sampath Kannan. The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *FOCS*, 2012.

[27] N. Immorlica and M. Mahdian. Marriage, honesty, and stability. In *SODA*, pages 53–62. Society for Industrial and Applied Mathematics, 2005.

[28] E. Kalai. Large robust games. *Econometrica*, 72(6):1631–1665, 2004.

[29] F. Kojima and P.A. Pathak. Incentives and stability in large two-sided matching markets. *American Economic Review*, 99(3):608–627, 2009.

[30] F. Kojima, P.A. Pathak, and A.E. Roth. Matching with couples: Stability and incentives in large markets. Technical report, National Bureau of Economic Research, 2010.

[31] Katrina Ligett and Aaron Roth. Take it or leave it: Running a survey when privacy comes at a cost. *CoRR*, abs/1202.4741, 2012.

[32] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[33] N. Nisan. *Algorithmic game theory*. Cambridge Univ Pr, 2007.

[34] Kobbi Nissim, Claudio Orlandi, and Rann Smorodinsky. Privacy-aware mechanism design. In *EC*, pages 774–789, 2012.

[35] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. Approximately optimal mechanism design via differential privacy. In *ITCS*, pages 203–213, 2012.

[36] D.J. Roberts and A. Postlewaite. The incentives for price-taking behavior in large exchange economies. *Econometrica*, pages 115–127, 1976.

[37] Aaron Roth. The algorithmic foundations of data privacy, course notes. *http://www.cis.upenn.edu/ aaroth/courses/privacyF11.html*, 2011.

[38] Aaron Roth. Buying private data at auction: The sensitive surveyor's problem. *ACM SIGecom Exchanges*, pages 1–8, 2012.

[39] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *STOC '10*, pages 765–774, 2010.

[40] Aaron Roth and Grant Schoenebeck. Conducting truthful surveys, cheaply. In *EC*, pages 826–843, 2012.

[41] A.E. Roth and X. Xing. Jumping the gun: Imperfections and institutions related to the timing of market transactions. *American Economic Review*, pages 992–1044, 1994.

[42] T. Roughgarden. Intrinsic robustness of the price of anarchy. In *STOC*, pages 513–522. ACM, 2009.

[43] David Xiao. Is privacy compatible with truthfulness? *IACR Cryptology ePrint Archive*, 2011:5, 2011.

A Proofs

A.1 Proofs from Section 3

PROOF OF COROLLARY 1. We will prove only item 1, the proof for 2 is analogous. First, by the assumption of the theorem, we will have $\widehat{L} \in [0,1]^{T\times k}$ except with probability at most β. Therefore, by Theorem 4,

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}}\right] \leq \beta$$

Further, by Lemma 2, we know that $\widehat{L} \in [0,1]^{T\times k}$ implies

$$\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}) \leq \rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}) + 2b.$$

Combining, we have the desired result, i.e.

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + 2b\right] \leq \beta. \qquad \square$$

PROOF OF COROLLARY 2. First, we demonstrate that $\widehat{L} \in [0,1]^{T\times k}$ except with probability at most β, which will be necessary to apply the regret bounds of Theorem 4. Specifically:

$$\mathop{\mathbb{P}}_{Z}\left[\exists z_t^j \text{ s.t. } |z_t^j| > \frac{1}{3}\right] \leq Tk \mathop{\mathbb{P}}_{Z}\left[|z_1^1| > \frac{1}{3}\right] \leq 2Tke^{-1/6\sigma} \leq \beta/2, \tag{10}$$

where the first inequality follows from the union bound, the second from the definition of Laplacian r.v.'s and the last inequality follows from the assumption that $\sigma \leq 1/6\log(4Tk/\beta)$.

The Theorem now follows by conditoning on the event $\widehat{L} \in [0,1]^{T\times k}$ and combining the regret bounds of Theorem 4 with the guarantees of Lemma 3. For parsimony, we will only demonstate the first inequality, the second is analogous. Recall again by Theorem 4, we have that whenever $\widehat{l} \in [0,1]^{T\times k}$:

$$\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) \leq \sqrt{\frac{2\log k}{T}}.$$

Further, by Lemma 3, we know that:

$$\begin{aligned}\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) - \rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) > \eta\right] &\leq 2|\mathcal{F}_{\mathsf{fixed}}|e^{-\eta^2 T/24\sigma^2} \\ &= 2ke^{-\eta^2 T/24\sigma^2}.\end{aligned}$$

Substituting $\eta = \sigma\sqrt{\frac{24\log(4k/\beta)}{T}}$, we get:

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) - \rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) > \eta\right] \leq \beta/2. \tag{11}$$

The result follows by combining (10) and (11). □

A.2 Proofs from Section 4

A.3 Proof of Theorem 11

Given a database $D \in \{0,1\}^n$, $D = (d_1, \ldots, d_n)$ and m queries $\mathcal{Q} = \{q_1, \ldots, q_m\}$, we will construct the following $(N = n + m\log n)$-player 2-action game. We denote the set of actions for each player by $A = \{0,1\}$. We also use $\{(j,h)\}_{j\in[m],h\in[\log n]}$ to denote the $m\log n$ players $\{n+1, \ldots, n+m\log n\}$. For intuition, think of player (j,h) as computing the h-th bit of $q_j(D)$.

Each player $i \in [n]$ has the utility function

$$u_i(\mathbf{a}) = \begin{cases} 1 & \text{if } a_i = d_i \\ 0 & \text{otherwise} \end{cases}$$

That is, player i receives utility 1 if they play the action matching the i-th entry in D, and utility 0 otherwise. Clearly, these are 0-sensitive utility functions.

The specification of the utility functions for the query players (j,h) is somewhat more complicated. First, we define the functions $f_h, g_h \colon [0,1] \to [0,1]$ as

$$f_h(x) = 1 - \min_{r\in\left\{0,\ldots,2^{h-1}-1\right\}} \left|x - (2^{-(h+1)} + r2^{-(h-1)})\right|$$
$$g_h(x) = 1 - \min_{r\in\left\{0,\ldots,2^{h-1}-1\right\}} \left|x - (2^{-h} + 2^{-(h+1)} + r2^{-(h-1)})\right|$$

Each player (j,h) will have the utility function

$$u_{(j,h)}(a_{-(j,h)}, 0) = f_h(q_j(a_1, \ldots, a_n))$$
$$u_{(j,h)}(a_{-(j,h)}, 1) = g_h(q_j(a_1, \ldots, a_n))$$

Since $q(a_1, \ldots, a_n)$ is defined to be $1/n$-sensitive in the actions $a_1, \ldots, a_n$, and f_h, g_h are 1-Lipschitz in x, $u_{(j,h)}$ is also $1/n$-sensitive.

Also notice that since $\mathcal{Q}$ is part of the definition of the game, we can simply define the set of feasible utility functions to be all those we have given to the players. For the data players we only used 2 distinct utility functions, and each of the $m\log n$ query players may have a distinct utility function. Thus we only need the set $\mathcal{U}$ to be a particular set of utility functions of size $m\log n + 2$ in order to implement the reduction.

Now we can analyze the structure of α-approximate equilibrium in this game, and show how, given any equilibrium set of strategies for the query players, we can compute a set of $O(\alpha)$-approximate answers to the set of queries $\mathcal{Q}$.

We start by claiming that in any α-approximate CCE, every data player players the action d_i in most rounds. Specifically,

CLAIM 1. Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Then for every data player i,

$$\mathop{\mathbb{P}}_{\pi}[a_i \neq d_i] \leq \alpha.$$

PROOF.

$$\begin{aligned}
\mathop{\mathbb{P}}_{\pi}[a_i \neq d_i] &= 1 - \mathop{\mathbb{E}}_{\pi}[u_i(a_i, a_{-i})] \\
&\leq 1 - \left(\mathop{\mathbb{E}}_{\pi}[u_i(d_i, a_{-i})] - \alpha\right) && \text{(Definition of } \alpha\text{-approximate CCE)} \\
&= 1 - (1 - \alpha) = \alpha && \text{(Definition of } u_i\text{)}
\end{aligned}$$

□

The next claim asserts that if we view the actions of the data players, $a_1, \ldots, a_n$, as a database, then $q(a_1, \ldots, a_n)$ is close to $q(d_1, \ldots, d_n)$ on average.

CLAIM 2. Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Let $q \subseteq [n]$ be any subset-sum query. Then

$$\mathop{\mathbb{E}}_{\pi}[|q(d_1, \ldots, d_n) - q(a_1, \ldots, a_n)|] \leq \alpha.$$

PROOF.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\pi}[|q(d_1, \ldots, d_n) - q(a_1, \ldots, a_n)|] = \mathop{\mathbb{E}}_{\pi}\left[\frac{1}{n}\sum_{i \in q}(d_i - a_i)\right] \\
&\leq \frac{1}{n}\sum_{i \in q}\mathop{\mathbb{E}}_{\pi}[|d_i - a_i|] = \frac{1}{n}\sum_{i \in q}\mathop{\mathbb{P}}_{\pi}[a_i \neq d_i] \\
&\leq \frac{1}{n}\sum_{i \in q}\alpha \quad \leq \alpha && \text{(Claim 1, } q \subseteq [n]\text{)}
\end{aligned}$$

□

We now prove a useful lemma that relates the expected utility of an action (under any distribution) to the expected difference between $q_j(a_1, \ldots, a_n)$ and $q_j(D)$.

CLAIM 3. Let μ be any distribution over A^N. Then for any query player (j, h),

$$\begin{aligned}
\left|\mathop{\mathbb{E}}_{\mu}\left[u_{(j,h)}(0, a_{-(j,h)})\right] - f_h(q_j(D))\right| &\leq \mathop{\mathbb{E}}_{\mu}[|q_j(a_1, \ldots, a_n) - q_j(D)|], \text{ and} \\
\left|\mathop{\mathbb{E}}_{\mu}\left[u_{(j,h)}(1, a_{-(j,h)})\right] - g_h(q_j(D))\right| &\leq \mathop{\mathbb{E}}_{\mu}[|q_j(a_1, \ldots, a_n) - q_j(D)|].
\end{aligned}$$

PROOF. We prove the first assertion, the proof of the second is identical.

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{\mu}\left[u_{(j,h)}(0, a_{-i})\right] - f_h(q_j(D))\right| \\
&= \left|\mathop{\mathbb{E}}_{\mu}[f_h(q_j(a_1, \ldots, a_n)) - f_h(q_j(D))]\right| \\
&\leq \mathop{\mathbb{E}}_{\pi}[|q_j(a_1, \ldots, a_n) - q_j(D)|] && (f_h \text{ is 1-Lipschitz})
\end{aligned}$$

□

The next claim, which establishes a lower bound on the expected utility player (j,h) will obtain for playing a fixed action, is an easy consequence of Claims 2 and 3.

CLAIM 4. Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Then for every query player (j,h),

$$\left|\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(0,a_{-i})\right] - f_h(q_j(D))\right| \leq \alpha, \text{ and}$$
$$\left|\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(1,a_{-i})\right] - g_h(q_j(D))\right| \leq \alpha.$$

Now we state a simple fact about the functions f_h and g_h. Informally, this asserts that we can find alternating intervals of width nearly 2^{-h}, that nearly partition $[0,1]$, in which $f_h(x)$ is significantly larger than $g_h(x)$ or vice versa.

OBSERVATION 1. Let $\beta \leq 2^{-(h+1)}$. If

$$x \in \bigcup_{r \in \left\{0,1,\ldots,2^{h-1}-1\right\}} \left(r2^{-h}+\beta, (r+1)2^{-h}-\beta\right)$$

then $f_h(x) > g_h(x) + \beta$. We denote this region $F_{h,\beta}$. Similarly, if

$$x \in \bigcup_{r \in \left\{0,1,\ldots,2^{h-1}-1\right\}} \left((r+1)2^{-h}+\beta, (r+2)2^{-h}-\beta\right)$$

then $g_h(x) > f_h(x) + \beta$. We denote this region $G_{h,\beta}$

For example, when $h=3$, $F_{3,\beta} = [0, \frac{1}{8}-\beta] \cup [\frac{2}{8}+\beta, \frac{3}{8}-\beta] \cup [\frac{4}{8}+\beta, \frac{5}{8}-\beta] \cup [\frac{6}{8}+\beta, \frac{7}{8}-\beta]$.

By combining this fact, with Claim 4, we can show that if $q_j(D)$ falls in the region $F_{h,\alpha}$, then in an α-approximate CCE, player (j,h) must be playing action 0 'often'.

CLAIM 5. Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Let $j \in [m]$ and $2^{-h} \geq 10\alpha$. Then, if $q_j(D) \in F_{h,9\alpha}$, $\mathbb{P}_\pi[a_i = 0] \geq 2/3$. Similarly, if $q_j(D) \in G_{h,9\alpha}$, then $\mathbb{P}_\pi[a_i = 1] \geq 2/3$.

PROOF. We prove the first assertion. The proof of the second is identical. If player (j,h) plays the fixed action 0, then, by Claim 4,

$$\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(0,a_{-(j,h)})\right] \geq f_h(q_j(D)) - \alpha.$$

Thus, if π is an α-approximate CCE, player (j,h) must receive at least $f_h(q_j(D)) - 2\alpha$ under π. Assume towards a contradiction that $\mathbb{P}\left[a_{(j,h)} = 0\right] < 2/3$. We can bound player (j,h)'s expected

utility as follows:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{a}\xleftarrow{R}\pi}\left[u_{(j,h)}(\mathbf{a})\right] \\
&= \mathbb{P}\left[a_{(j,h)}=0\right]\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(0,a_{-(j,h)}) \mid a_{(j,h)}=0\right] \\
&\quad + \mathbb{P}\left[a_{(j,h)}=1\right]\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(1,a_{-(j,h)}) \mid a_{(j,h)}=1\right] \\
&\leq \mathbb{P}\left[a_{(j,h)}=0\right]\left(f_h(q_j(D)) + \mathop{\mathbb{E}}_{\mathbf{a}\xleftarrow{R}\pi}\left[|q_j(a_1,\ldots,a_n)-q_j(D)| \mid a_{(j,h)}=0\right]\right) \\
&\quad + \mathbb{P}\left[a_{(j,h)}=1\right]\left(g_h(q_j(D)) + \mathop{\mathbb{E}}_{\mathbf{a}\xleftarrow{R}\pi}\left[|q_j(a_1,\ldots,a_n)-q_j(D)| \mid a_{(j,h)}=1\right]\right) \qquad (12) \\
&= f_h(q_j(D)) + \mathop{\mathbb{E}}_{\mathbf{a}\xleftarrow{R}\pi}\left[|q_j(a_1,\ldots,a_n)-q_j(D)|\right] - \mathbb{P}\left[a_{(j,h)}=1\right]\left(f_h(q_j(D)) - g_h(q_j(D))\right) \\
&\leq f_h(q_j(D)) + \alpha - 9\alpha\,\mathbb{P}\left[a_{(j,h)}=1\right] \qquad (13) \\
&< f_h(q_j(D)) - 2\alpha \qquad (14)
\end{aligned}
$$

Line (12) follows from the Claim 3 (applied to the distributions $\pi \mid a_{(j,h)}=0$ and $\pi \mid a_{(j,h)}=1$). Line (13) follows from Claim 2 (applied to the expectation in the second term) and the fact that $q_j(D) \in F_{h,9\alpha}$ (applied to the difference in the final term). Line (14) follows from the assumption that $\mathbb{P}\left[a_{(j,h)}=0\right] < 2/3$. Thus we have established a contradiction to the fact that π is an α-approximate CCE. □

Given the previous claim, the rest of the proof is fairly straightforward. For each query j, we will start at $h=1$ and consider two cases: If player $(j,1)$ plays 0 and 1 with roughly equal probability, then we must have that $q_j(D) \notin F_{1,9\alpha} \cup G_{1,9\alpha}$. It is easy to see that this will confine $q_j(D)$ to an interval of width 18α, and we can stop. If player $(j,1)$ does play one action, say 0, a significant majority of the time, then we will know that $q_j(D) \in F_{1,9\alpha}$, which is an interval of width $1/2 - 9\alpha$. However, now we can consider $h=2$ and repeat the case analysis: Either $(j,2)$ does not significantly favor one action, in which case we know that $q_j(D) \notin F_{2,9\alpha} \cup G_{2,9\alpha}$, which confines $q_j(D)$ to the union of two intervals, each of width 18α. However, only one of these intervals will be contained in $F_{1,9\alpha}$, which we know contains $q_j(D)$. Thus, if we are in this case, we have learned $q_j(D)$ to within 18α and can stop. Otherwise, if player $(j,2)$ plays, say, 0 a significant majority of the time, then we know that $q_j(D) \in F_{1,9\alpha} \cap F_{2,9\alpha}$, which is an interval of width $1/4 - 9\alpha$. It is not too difficult to see that we can repeat this process as long as $2^{-h} \geq 18\alpha$, and we will terminate with an interval of width at most 36α that contains $q_j(D)$.