Smooth Nash Equilibria: Algorithms and Complexity

Constantinos Daskalakis
MIT, Cambridge, MA, USA

Noah Golowich
MIT, Cambridge, MA, USA

Nika Haghtalab
University of California at Berkeley, CA, USA

Abhishek Shetty
University of California at Berkeley, CA, USA

Abstract

A fundamental shortcoming of the concept of Nash equilibrium is its computational intractability: approximating Nash equilibria in normal-form games is PPAD-hard. In this paper, inspired by the ideas of smoothed analysis, we introduce a relaxed variant of Nash equilibrium called *σ-smooth Nash equilibrium*, for a smoothness parameter σ. In a σ-smooth Nash equilibrium, players only need to achieve utility at least as high as their best deviation to a *σ-smooth strategy*, which is a distribution that does not put too much mass (as parametrized by σ) on any fixed action. We distinguish two variants of σ-smooth Nash equilibria: *strong* σ-smooth Nash equilibria, in which players are required to play σ-smooth strategies under equilibrium play, and *weak* σ-smooth Nash equilibria, where there is no such requirement.

We show that both weak and strong σ-smooth Nash equilibria have superior computational properties to Nash equilibria: when σ as well as an approximation parameter ϵ and the number of players are all constants, there is a *constant-time* randomized algorithm to find a weak ϵ-approximate σ-smooth Nash equilibrium in normal-form games. In the same parameter regime, there is a *polynomial-time* deterministic algorithm to find a strong ϵ-approximate σ-smooth Nash equilibrium in a normal-form game. These results stand in contrast to the optimal algorithm for computing ϵ-approximate Nash equilibria, which cannot run in faster than quasipolynomial-time, subject to complexity-theoretic assumptions. We complement our upper bounds by showing that when either σ or ϵ is an inverse polynomial, finding a weak ϵ-approximate σ-smooth Nash equilibria becomes computationally intractable.

Our results are the first to propose a variant of Nash equilibrium which is computationally tractable, allows players to act independently, and which, as we discuss, is justified by an extensive line of work on individual choice behavior in the economics literature.

2012 ACM Subject Classification Theory of computation → Exact and approximate computation of equilibria; Theory of computation → Algorithmic game theory

Keywords and phrases Nash equilibrium, smoothed analysis, PPAD

Digital Object Identifier 10.4230/LIPIcs.ITCS.2024.37

Related Version *Full Version*: `https://arxiv.org/abs/2309.12226`

Funding *Constantinos Daskalakis*: Supported by NSF Awards CCF-1901292, DMS-2022448, and DMS2134108, a Simons Investigator Award, the Simons Collaboration on the Theory of Algorithmic Fairness.
Noah Golowich: Supported by a Fannie & John Hertz Foundation Fellowship and an NSF Graduate Fellowship.
Nika Haghtalab: This work was supported in part by the National Science Foundation under grant CCF-2145898, a C3.AI Digital Transformation Institute grant, Google faculty scholar award, and Berkeley AI Research Commons grants.
Abhishek Shetty: Supported by an Apple AI/ML PhD Fellowship.

Acknowledgements This work was done in part while the authors were visiting the Learning and Games program at the Simons Institute for the Theory of Computing.

1 Introduction

The notion of *Nash equilibrium*, which is a strategy profile of a game in which each player acts independently of the other players in a way that best responds to them, has been a mainstay in the study of game theory and economics over the last several decades [52], with a wide range of applications in related areas. For instance, it provides a model that predicts agents' behavioral patterns [40], though with varying degrees of success, serves as a universal solution concept in multiagent learning settings [27, 65], and furnishes a rich set of problems for study of the computational aspects of equilibria [20, 5], among other applications.

Most known results pertaining to the computation of a Nash equilibrium in general-sum normal-form games are negative. In particular, it is known to be PPAD-hard to compute Nash equilibria, even approximately [21, 18, 20, 57], meaning that there is unlikely to be a polynomial-time algorithm. Similar hardness results abound in other natural models of computation, such as the *query complexity* model. In particular, even in 2-player normal-form games, one has to query a nearly constant fraction of the payoff matrices' entries to find an approximate Nash equilibrium [25, 26, 32]. The computational intractability of Nash equilibria casts doubt over whether it is actually a reasonable solution concept to model agents' play: real-world agents can only implement efficient algorithms, so should not be expected to play Nash equilibria in all games [21, 20, 38].[1] Moreover, for applications in which Nash equilibria do yield high-utility strategies for agents, its computation nevertheless presents significant challenges [47].

To address these issues, it is popular to consider relaxations of Nash equilibria. Given the competitive nature of many games, one fudamental property of Nash equilibria we would like such a relaxation to preserve is that players choose their actions independently. Second, of course, we would like such a relaxation to be computationally tractable. We remark that the well-known equilibrium concepts of *coarse correlated equilibria* [50] and *correlated equilibria* [4], which are relaxations of Nash equilibria, can be computed in polynomial time, but generally involve correlation in the actions of the players. In particular, implementing them requires a *correlation scheme*, which draws a random variable and sends each player a private signal depending on it. Such schemes with guaranteed privacy of the signals may not be available in certain applications, motivating our requirement that players' strategies be independent.

Smooth Nash equilibria

In this paper we introduce *smooth Nash equilibria*, which meet both of the above requirements. The inspiration for smooth Nash equilibria comes from smoothed analysis [63], which aims to circumvent computational hardness barriers by postulating that inputs to algorithms come from *smooth distributions*, which do not place too much mass on any single value. Alternatively, one may think of smooth distributions as being perturbations of arbitrary distributions. There are two ways to apply smoothed analysis in the context of game theory: first, one can assume the game itself (i.e., the payoff matrices) comes from a smooth

[1] This perspective is well summarized by a quote by Kamal Jain: "If your laptop can't find it then neither can the market".

distribution, and second, one could assume the game is worst-case but that the players choose their strategies according to smooth distributions. The former approach has been well-studied, but it fails to sidestep the PPAD-hardness barrier [18, 17]. We focus on the latter approach, which has attracted more attention in recent years, wherein players' strategies (usually the adversarial player's strategies) are smoothed [35, 36, 34, 13, 11, 15, 14, 16, 54].

In particular, given a normal-form game G where each player has n actions and a *smoothness parameter* $\sigma \in (0,1)$, we say that a strategy profile is a *σ-smooth Nash equilibrium* if each player cannot improve its utility by switching to any *σ-smooth* distribution over actions. A σ-smooth distribution is one which does not place too much mass, namely, more than $1/(\sigma n)$, on any single action. Further, given $\epsilon \in (0,1)$, an *ϵ-approximate σ-smooth Nash equilibrium* is a strategy profile in which each player cannot improve its utility by more than ϵ by switching to a σ-smooth distribution. We refer the reader to Section 3 for the formal definitions.

The computational properties of smooth Nash equilibrium (as well as its approximate counterpart) depend heavily on the smoothness parameter $\sigma \in (0,1)$. It is easy to see that when $\sigma = 1/n$, the definition of smooth Nash equilibrium coincides with that of Nash equilibrium, whereas when $\sigma = 1$, the concept is vacuous: the uniform distribution over $[n]$ is a smooth Nash equilibrium. Our focus is on the intermediate regime $1/n < \sigma < 1$. We also make a distinction between two versions of (approximate) smooth equilibrium: we say that a *strong* ϵ-approximate σ-smooth Nash equilibrium is one in which players are required to play σ-smooth distributions under equilibrium play; a *weak* ϵ-approximate σ-smooth Nash equilibrium does not have this additional requirement.

Our contributions

We study the complexity of computing ϵ-approximate σ-smooth Nash equilibria in m-player, n-action normal-form games, for various values of $\epsilon, \sigma \in (0,1)$. Our focus is on the setting when the number of players m is a constant, though our bounds apply more generally. In this setting, our main results are summarized below, where we let n denote the number of actions of each player:

- When ϵ, σ are constants, there is a $\mathrm{poly}(n)$-time deterministic algorithm which finds a *strong* ϵ-approximate σ-smooth Nash equilibrium (Theorem 15).
- When ϵ, σ are constants, there is a ***constant-time*** randomized algorithm which finds a *weak* ϵ-approximate σ-smooth Nash equilibrium (Theorem 14). Furthermore, the number of queries this algorithm makes to the payoff matrices is bounded above by $\mathrm{poly}(\epsilon^{-1}, \sigma^{-1})$ (Theorem 10).
- There is a constant σ_0 so that, for $\epsilon = 1/\mathrm{poly}(n)$, it is PPAD-hard to compute weak ϵ-approximate σ_0-smooth Nash equilibrium in 2-player games (Theorem 21). Moreover, there is a constant ϵ_0 so that, for $\sigma = 1/\mathrm{poly}(n)$, there is no algorithm running in time $n^{o(\log n)}$ which computes weak ϵ_0-approximate σ-smooth Nash equilibrium in 2-player games under the Exponential Time Hypothesis for PPAD (Corollary 18).

In the case that $\sigma = 1/n$, in which case weak and strong σ-smooth Nash equilibria are both equivalent to Nash equilibria, our results on time complexity recover well-known results on the complexity of Nash equilibria (assuming that m is constant). In particular, in this regime, our upper bound of $n^{O(\log(1/\sigma)/\epsilon^2)}$ becomes $n^{O(\log(n)/\epsilon^2)}$, which recovers the classical result of [44]. In fact, under ETH for PPAD, this is the best possible upper bound for computing approximate Nash equilibria in 2-player games [58]. Furthermore, any algorithm for computing approximate Nash equilibria in 2-player games must make $n^{2-o(1)}$ queries [32] which is in contrast with our *constant-time* randomized algorithm for finding weak σ-smooth Nash equilibria when σ is constant.

Thus, the parameters ϵ, σ act as dials which allow us to trade off computational efficiency for a stronger equilibrium notion. Our results are summarized in Table 1.

Table 1 Summary of our results. †: These upper bounds are obtained by randomized algorithms, and compute *weak* smooth Nash equilibria with probability $1-\delta$. ⋆: This upper bound is obtained by a deterministic algorithm, and computes *strong* smooth Nash equilibria. All lower bounds hold with respect to weak (and therefore also strong) Nash equilibria, and hold with respect to randomized algorithms.

Model	Upper Bound	Lower Bounds	
Query Complexity	$m \cdot \left(\frac{m \log^2(m/(\delta\sigma\epsilon))}{\epsilon^2\sigma}\right)^{m+1}$ Theorem 10†	$\Omega(\exp(m))$ Remark 12	$\epsilon = \Omega(1)$, $\sigma = \Omega(1)$
Time Complexity	$n^{O(m^4 \log(m/\sigma)/\epsilon^2)}$ Theorem 15⋆	PPAD-hard Theorem 21	$\epsilon = n^{-c}$, $\sigma = \Omega(1)$
Time Complexity	$n^{O(m^4 \log(m/\sigma)/\epsilon^2)}$ Theorem 15⋆	$n^{\log n}$ under ETH-PPAD Corollary 18	$\epsilon = \Omega(1)$, $\sigma = n^{-c}$
Time Complexity	$\left(\frac{m \log(1/\delta)}{\sigma\epsilon}\right)^{O\left(\frac{m^2 \log(m/\delta\sigma)}{\epsilon^2}\right)}$ Theorem 14†	$\Omega(\exp(m))$ Remark 12	$\epsilon = \Omega(1)$, $\sigma = \Omega(1)$

1.1 Further Motivation & Connections to Other Equilibrium Notions

Implicit in the definition of σ-smooth Nash equilibria is the assumption that agents only aim to compete with their best σ-smooth response. It is natural to question how reasonable this assumption is. In this section, we discuss an extensive line of work on individual choice behavior in the economics literature which has yielded choice models that help to justify σ-smooth Nash equilibria. In doing so, we also discuss connections between σ-smooth Nash equilibria and other variants of Nash equilibria which have been previously proposed.

Quantal Response Equilibria & Logit Equilibria

A fundamental difference between σ-smooth Nash equilibrium and the standard notion of Nash equilibrium is that in the former, agents do not place all mass of their strategy on their utility-maximizing action(s), given the strategies of others. One may wonder: *Is this realistic behavior?* The theory of *random utility models* [48] proposes a justification for this type of behavior. In particular, suppose that, for fixed strategies of other agents, each agent j's computation of its expected utilities for each action is perturbed by a random noise vector ε_j. This noise could represent calculation errors due to imperfect information an agent has about the game's payoffs or other agents' strategies. As a result of this noise, conditioned on other agents' strategies an agent will generally play some actions which are suboptimal. For a fixed distribution over the agents' noise vectors $(\varepsilon_1, \ldots, \varepsilon_m)$, a strategy profile in which each agent j's strategy is obtained as above by perturbing its (true) expected payoff by ε_j and then best-responding, is known as a *quantal response equilibrium* [49, 31]. Moreover, the mapping from an agent's expected payoff vector u to its strategy obtained by best-responding to the perturbed vector $u + \varepsilon_j$ is known as its *quantal response function*.

We remark that an agent's best σ-smooth response yields larger utility than its quantal response function for many choices of the perturbations ε_j. This observation provides further justification for the set of σ-smooth deviations as representing a reasonable deviation set with which to define equilibria. To illustrate, consider the common setting where the perturbations ε_j are drawn according to the extreme value distribution. In this case, given an expected payoff vector $u \in \mathbb{R}^n$, an agent's quantal response function is given by a softmax function of u. This class of response functions, known as the *logistic response functions*, has its origins in the seminal work of Luce [45], and the corresponding special case of quantal response equilibrium, called *logit equilibrium*, has been extensively studied since (e.g., [48, 2, 37, 30, 29]). We show in the full version [22, Proposition G.3] that, under mild conditions on σ, the utility obtained by the logistic response function is no better than that obtained by the best σ-smooth distribution.

In light of the above connections, one might wonder if σ-smooth Nash equilibria can in fact be obtained as a special case of quantal response equilibria, for some distribution of the noise vectors ε_j. As we discuss in [22, Appendix G], this is not the case: for any distribution of the noise vectors ε_j, there are games for which the set of quantal response equilibria is separated in total variation distance from the set of strong σ-smooth Nash equilibria.

Finally, we mention that a key property of σ-smooth Nash equilibrium which is essential to obtaining efficient algorithms is that there is a natural notion of ϵ-approximate smooth equilibrium (Definition 4). All of our efficient algorithms require $1/\epsilon$ to be bounded. There does not appear to be a correspondingly natural notion of approximation for quantal response equilibria.

Polyhedral games

The set of strong σ-smooth Nash equilibria for a given normal-form game may be viewed as the set of Nash equilibria in the *concave polyhedral game* in which each player's action set is the polyhedron consisting of all σ-smooth distributions, and the payoffs are defined by linearity. The study of (polyhedral) concave games was initiated by [55], and has since been studied in various contexts, including for analyzing no-regret learning algorithms [33] and modeling extensive-form games [24], amongst others [46]. Our results on strong σ-smooth equilibria can be viewed as showing that there are efficient algorithms for computation of approximate Nash equilibria in a certain natural class of polyhedral concave games.

Additional equilibrium notions

[56] introduced a notion of *bounded rationality equilibrium* in which the probability that each player plays an action is related linearly to the suboptimality of that action. The resulting equilibria have similar behavior to quantal response equilibria [49, 31]. [10] introduced *imperfect performance equilibria*, which are Nash equilibria in an augmented game in which each agent j intends to play a certain strategy $\tilde{x}_j$, but the strategy it actually plays, x_j, is the result of passing $\tilde{x}_j$ through a *performance mapping*. This performance mapping could simulate the fact that "agents" in a game can represent organizations, whose actions are the agglomeration of those of multiple individuals. As these individuals may be self-interested, they may not always act according to the organization's best interests (i.e., its best response).

Trembling-hand perfect equilibria, introduced by [60], are those which are obtained as limit points of a sequence of polyhedron games, parametrized by $\delta \searrow 0$, in which each player's polyhedron is the set of strategies which put mass at least δ on each action. Note that this polyhedron is distinct from the polyhedron of smooth strategies, which require that each action receives mass *no more than* $1/(\sigma n)$. Trembling-hand perfect equilibria are further refined by *proper equilibria* [51]. We remark that trembling-hand perfect equilibria are related to certain limit points of *control-cost equilibria* [64, Chapter 4], which are equilibria in which each player pays a cost for playing each action, which depends on the probability with which the action is played. The above notions of equilibria are inspired by various justifications as to why agents might play suboptimal actions with positive probability. However, unlike smooth Nash equilibria, none of them can be computed efficiently in general normal-form games. In fact, trembling-hand perfect equilibria and proper equilibria are refinements of Nash equilibria, so an efficient algorithm for computing them would imply an efficient algorithm for computing Nash equilibria.

Regularization and equilibrium solving

Quantal response equilibria can be interpreted as Nash equilibria in a modified game where each player's utility has an additional regularization term that depends on the entropy of its strategy. The idea of using entropy regularization, as well as generalizations of it, has been exploited in recent years [42, 8, 53, 23, 62] to instill desirable properties of solutions to multi-agent reinforcement learning problems. For instance, to find strategies which are "human-like", [42] does the following: for a fixed, known human strategy $\tilde{x}$, they introduce a regularizer which minimizes the KL divergence to $\tilde{x}$ (a generalization of entropy), thus ensuring that learned strategies will be similar to $\tilde{x}$, while being close to a best response in some sense. In a similar spirit, smooth Nash equilibria can be seen as equilibria induced by regularization with respect to the *min-entropy*. Moreover, appropriate generalizations of the min-entropy, as per Remark 6, could be used to model the task of finding equilibria which are close to arbitrary (non-uniform) distributions, such as ones played by humans. Whereas most of the prior work on regularization in this context (e.g., [62]) focuses on two-player zero-sum games, our concept of smooth Nash equilibrium leads to efficient algorithms in multi-player, general-sum games. It is an exciting question to understand if our ideas have implications in such general-sum versions of these multi-agent RL problems.

Boosting

The best-response condition of smooth Nash equilibrium is also motivated by a connection to boosting algorithms. The link between two-player zero-sum games and boosting is well-known [59] and can be seen as a fundamental application of equilibrium solving. One of the key aspects of the analysis of boosting algorithms (such as ADABoost) which allows for rates that do not depend on the number of samples is the fact that one needs to only compete against subsets of the dataset of size ηn where n is the size of the dataset and η is the desired accuracy [3, Sections 3.6 and 3.7]. This can be viewed as the best-response condition of weak η-smooth Nash equilibrium, where the "dataset" player who places weights on the sample set need only compete with smooth distributions. Furthermore, in several applications such as the construction of hard-core sets [41, 43, 9], noise resilient (smooth) boosting [61] and construction of private datasets [35], it is important that the distribution output by the "dataset" player does not concentrate too much mass on any single point (see [3, Section 3.7] for a further discussion). This can be viewed as the best-response condition of the strong smooth Nash equilibrium.

Organization

In Section 3, we formally define smooth Nash equilibria and present an important structural result in Section 3.1. In Section 4, we present our query-efficient algorithms for finding weak smooth equilibria. In Section 5, we discuss efficient algorithms for finding smooth equilibria, including both the weak (Section 5.1) and strong (Section 5.2) variants. Section 6 contains our hardness results, and Section 7 contains concluding remarks and open questions.

Due to page limits, all our formal proofs, as well as additional discussion, may be found in the full version of this paper [22].

2 Preliminaries

Given a tuple of elements $y = (y_1, \ldots, y_m)$, we will denote by y_{-i} the tuple $(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m)$. Further, for an element y_i', we will use the standard game theoretic notation of (y_i', y_{-i}) to refer to the tuple $(y_1, \ldots, y_{i-1}, y_i', y_i, \ldots, y_m)$. In particular, $y = (y_i, y_{-i})$. For $i \le i'$, we write $y_{i:i'} = (y_i, y_{i+1}, \ldots, y_{i'})$. For $n \in \mathbb{N}$, we let Δ^n denote the set of distributions over $[n]$. For $i \in [n]$, let $e_i \in \Delta^n$ denote the corresponding basis vector. For $y \in \Delta^n$, we let $\mathsf{supp}(y) := \{i : \; y_i > 0\}$.

▶ **Definition 1** (Normal Form Games). *Let n and m be positive integers. A normal form game with m players each with n actions is specified by a tuple of* payoff mappings $A_1, \ldots, A_m : [n]^m \to [0,1]$, *with the interpretation that player j receives utility $A_j(a_1, \ldots, a_m)$ when the players play the actions $a_1, \ldots, a_m$ respectively.*

A *mixed strategy* (or simply *strategy*) of player j is an element $x_j \in \Delta^n$. A *strategy profile* is a tuple $x = (x_1, \ldots, x_m) \in (\Delta^n)^m$. We will extend the notation $A_j(a_1, \ldots, a_m)$ by linearity: for a strategy profile $x = (x_1, \ldots, x_m)$, we write

$$A_j(x) = A_j(x_1, \ldots, x_m) = \mathbb{E}_{a_1 \sim x_1} \ldots \mathbb{E}_{a_m \sim x_m} \left[A_j(a_1, \ldots, a_m) \right].$$

For an action $a_j \in [n]$ and a strategy profile x, we will write $A_j(a_j, x_{-j})$ to denote $A_j(e_{a_j}, x_{-j})$, namely player j's expected utility when it plays a_j and others play according to x.

▶ **Definition 2** (Nash equilibrium). *Given $\epsilon \in (0,1)$ and a normal-form game specified by $A_1, \ldots, A_m$, a strategy profile $x = (x_1, \ldots, x_m) \in (\Delta^n)^m$ is an ϵ-*approximate Nash equilibrium *if for each $j \in [m]$,*

$$\max_{x_j' \in \Delta^n} A_j(x_j', x_{-j}) - A_j(x) \le \epsilon.$$

We will refer to a 0-approximate Nash equilibrium as simply a Nash equilibrium.

In the special case of 2-player games, we will denote the payoff functions of the game by matrices $A, B \in \mathbb{R}^{n \times n}$, and the players' respective strategy profiles by $x, y \in \Delta^n$.

3 Smooth Equilibria

First, we define the polytope of smooth distributions, which is the set of distributions that do not concentrate too much mass on any one coordinate.

▶ **Definition 3** (Smooth Distribution Polytope). *Suppose $n \in \mathbb{N}$ and $\sigma \in \mathbb{R}$ satisfy $1/n \le \sigma \le 1$. The* smooth distribution polytope $\mathcal{K}_{\sigma,n}$ *is the set*

$$\mathcal{K}_{\sigma,n} = \left\{ x \in \Delta^n : 0 \le x_i \le \frac{1}{n\sigma} \quad \forall i \in [n] \right\}. \tag{1}$$

We call the parameter σ the *smoothness parameter*; for larger σ, the constraints defining $\mathcal{K}_{\sigma,n}$ are more stringent, i.e., distributions in $\mathcal{K}_{\sigma,n}$ are "smoother". In particular, for $\sigma = 1/n$ we have $\mathcal{K}_{\sigma,n} = \Delta^n$ and for $\sigma = 1$ we have $\mathcal{K}_{\sigma,n} = \left\{ \frac{1}{n}\mathbf{1} \right\}$ where $\mathbf{1}$ is the all ones vector. With this definition in hand, we now define smooth Nash equilibria, the key concept that we study in this paper.

▶ **Definition 4** (Smooth Nash equilibria). *Fix an m-player normal-form game G with payoff mappings $A_1, \ldots, A_m : [n]^m \to [0,1]$, and $\epsilon, \sigma \in [0,1]$. A strategy profile $x = (x_1, \ldots, x_m) \in (\Delta^n)^m$ is an* weak ϵ-approximate σ-smooth Nash equilibrium *if, for each $j \in [m]$,*

$$\max_{x'_j \in \mathcal{K}_{\sigma,n}} A_j(x'_j, x_{-j}) - A_j(x) \leq \epsilon.$$

It is moreover a strong ϵ-approximate σ-smooth Nash equilibrium *if $x_j \in \mathcal{K}_{\sigma,n}$ for each $j \in [m]$.*

We refer to weak (resp., strong) 0-approximate σ-smooth Nash equilibria as simply weak (resp., strong) σ-smooth Nash equilibria. Intuitively, Definition 4 says that in σ-smooth Nash equilibria, no player can gain more than ϵ utility by deviating to a distribution that is σ-smooth. Furthermore, in strong σ-smooth Nash equilibria, the mixed strategies of each player are themselves σ-smooth.

The set of strong σ-smooth Nash equilibria of a normal-form game is equal to the set of Nash equilibria in the polyhedral concave game [55] where each player's action set is $\mathcal{K}_{\sigma,n}$ and the payoffs are given by $x \mapsto A_j(x)$, for $j \in [m]$, $x \in (\mathcal{K}_{\sigma,n})^m$. Since the sets $\mathcal{K}_{\sigma,n}$ are compact and convex, [55, Theorem 1] implies that strong σ-smooth Nash equilibria always exist.

▶ **Proposition 5** (Existence of Equilibria; [55]). *For any $\sigma \in (0,1)$, every normal-form game has a strong σ-smooth Nash equilibrium.*

▶ Remark 6 (Generalizing the smooth distribution polytope). The smooth polytope $\mathcal{K}_{\sigma,n}$ may alternatively be defined as the set of distributions on $[n]$ whose Radon-Nikodym derivative with repsect to the uniform distribution on $[n]$ is bounded by $1/\sigma$. In particular, $\mathcal{K}_{\sigma,n} = \left\{x \in \Delta^n : \left\|\frac{dx}{d\mu_n}\right\|_\infty \leq \frac{1}{\sigma}\right\}$, where μ_n is the uniform measure on $[n]$. As is common in smoothed analysis (e.g., [13, 15, 14, 16]), one may generalize the notion of smooth distributions and thereby smooth equilibria by allowing μ_n to be an arbitrary distribution on $[n]$ or even a more general probability space. Essentially all of our results remain unchanged under appropriate compactness assumptions.

3.1 Structure of Smooth Nash Equilibria

Next, we discuss a key structural result underlying our upper bounds, which states that there are weak approximate smooth Nash equilibria for which the players' strategy profiles are supported on a small number of actions (Corollary 9). In fact, we prove a slightly stronger statement, namely that a smooth Nash equilibrium can be approximated by sampling.

▶ **Lemma 7.** *Let $A_1, \ldots, A_m : [n]^m \to [0,1]$ denote the payoff matrices of an m-player normal-form game G. Let $\epsilon, \sigma \in (0,1)$ be fixed. Let $k = \frac{C_0 m \log(8m/\delta\sigma)}{\epsilon^2}$ where C_0 is a sufficiently large constant. Given a weak σ-smooth Nash equilibrium $x = (x_1, \ldots, x_m)$, let $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$ be the random strategy profile where $\hat{x}_j$ is the uniform average over k actions $B_{j,1}, \ldots, B_{j,k}$ sampled i.i.d. from x_j. Then, with probability $1-\delta$, we have, for all $j \in [m]$,*

$$|A_j(x_1, \ldots, x_m) - A_j(\hat{x}_1, \ldots, \hat{x}_m)| \leq \frac{\epsilon}{2}, \qquad \sup_{x'_j \in \mathcal{K}_{\sigma,n}} \left|A_j(x'_j, \hat{x}_{-j}) - A_j(x'_j, x_{-j})\right| \leq \frac{\epsilon}{2}, \tag{2}$$

and $\hat{x}$ satisfying Equation (2) is a weak ϵ-approximate σ-smooth Nash equilibrium.

The main technical lemma driving the sampling result of Lemma 7 is a generalization of Massart's finite class lemma to the case when the supremum is over the set of smooth distributions on $[n]$ (namely, [22, Lemma A.1]). At a high level, this lemma allows us to bound the sampling error in each player's deviation to a best-response smooth strategy without incurring a $\log(n)$ factor in the number k of samples.

We capture the sparsity of strategy profiles such as $\hat{x}$ produced by the sampling procedure in Lemma 7 with the following definition.

▶ **Definition 8.** *For $k \in \mathbb{N}$, we say that a distribution $y \in \Delta^n$ is* k-uniform *if for each $i \in [n]$, y_i is an integral multiple of $1/k$. We say that a strategy profile $x = (x_1, \ldots, x_m) \in (\Delta^n)^m$ is k-uniform if each of its constituent strategies x_j is.*

Thus, we see from Lemma 7 that k-uniform weak approximate smooth Nash equilibria exist for k that is independent of the number of actions of the game.

▶ **Corollary 9** (Existence of k-uniform weak smooth Nash)**.** *Let $m \in \mathbb{N}$, $\epsilon, \sigma \in (0, 1)$, be given and set $k = \frac{C_0 m \log(8m/\sigma)}{\epsilon^2}$. Then any m-player normal-form game G has a weak k-uniform ϵ-approximate σ-smooth Nash equilibrium.*

Corollary 9 generalizes the result of [44], as well as the follow-up of [39], which treat the case of approximate Nash equilibrium, corresponding to the case of $\sigma = 1/n$. The parameter k needed by [44, 39], which governs the sparsity of the equilibrium, thus grows logarithmically in n. As we show in Section 5, the fact that the sparsity is only logarithmic in $1/\sigma$ is key to getting a polynomial-time algorithm for computing smooth Nash equilibria when $1/\sigma = O(1)$.

4 Query Complexity of Smooth Nash Equilibria

In this section, we prove an upper bound on the randomized query complexity of computing weak smooth Nash equilibria. This result comprises the bulk of our proof in Section 5.1 that there is a randomized constant-time algorithm for computing weak ϵ-approximate σ-smooth Nash equilibria. Beside the application to computational complexity, query complexity of equilibria is of interest in its own right, as a tool to understand the amount of information that needs to be shared in order to find an equilibrium [5]. It is also closely related to the analysis of uncoupled dynamics converging to equilibrium [19, 38, 32].

We briefly review the query complexity model. The payoff mappings $A_1, \ldots, A_m : [n]^m \to [0, 1]$ are assumed to be unknown, but the algorithm can repeatedly query single entries $A_j(a_1, \ldots, a_m)$ (for $j \in [m], a_1, \ldots, a_m \in [n]$) of these matrices. Randomness is allowed in choosing the queries. After making at most Q queries, for $Q \in \mathbb{N}$ denoting the *query complexity*, the algorithm is required to output a strategy profile. Our main result in this model is Theorem 10 below, which states that we can find a smooth Nash equilibrium with a number of queries that is independent of the number of actions n.

▶ **Theorem 10.** *Let $A_1, \ldots, A_m : [n]^m \to [0, 1]$ denote the payoff matrices of an m-player normal-form game G. Let $\epsilon, \sigma, \delta \in (0, 1)$ be fixed. Then* `QueryEquilibrium`$((A_1, \ldots, A_m), \sigma, \epsilon, \delta)$ *(Algorithm 2) makes $O\left(m \cdot \left(\frac{m \log^2(m/(\delta\sigma\epsilon))}{\epsilon^2\sigma}\right)^{m+1}\right)$ queries to entries of payoff matrices of G. Moreover, it outputs a strategy profile $\hat{x}$ which is a weak ϵ-approximate σ-smooth Nash equilibrium of G with probability at least $1 - \delta$.*

Algorithm 1 `OptimizeDeviation`($\mathcal{R}, A, j, x, \sigma$): compute optimal smooth deviation using few queries.

1: Write $\mathcal{R} = \{r_{j,1}, \dots, r_{j,N}\}$, for $r_{j,1}, \dots, r_{j,N} \in [n]$.
2: **for** $i \in [N]$ **do**
3: Set $\hat{v}_i = A(r_{j,i}, x_{-j})$.
4: Let $\tau_1, \dots, \tau_N \in [N]$ denote a permutation of $[N]$ so that $\hat{v}_{\tau_1} \geq \dots \geq \hat{v}_{\tau_N}$.
5: **return** $\hat{v} := \frac{1}{\sigma N} \sum_{k=1}^{\sigma N} \hat{v}_{\tau_k}$.

Note that when $m, \sigma, \epsilon, \delta$ are constant, the query bound obtained by Theorem 10 is also a constant. This is in constrast to the query complexity of ϵ-approximate Nash equilibria, even for two players, which requires at least $n^{2-o(1)}$ queries even for constant ϵ and constant probability of error δ [25, 26, 32].

The main idea behind the algorithm `QueryEquilibrium` can be summarized in two steps. The first is the following strengthening of the existence of k-uniform approximate smooth Nash equilibria from Corollary 9. In particular, Lemma 11 below shows that not only do k-uniform approximate smooth Nash equilibria exist for small k, but also that they can found in small subsets that are uniformly sampled from the set of actions.

▶ **Lemma 11.** *Let $A_1, \dots, A_m : [n]^m \to [0, 1]$ denote the payoff matrices of an m-player normal-form game G. Let $\epsilon, \sigma \in (0, 1)$ be fixed, and $t = \frac{C_0 m \log(16m/\delta\sigma)}{\epsilon^2}, \ell = \frac{\log(2tm/\delta)}{\sigma}$ and set $k = t\ell$. For $j \in [m]$, let $\hat{X}^j \in \Delta^n$ denote the uniform measure over k uniformly random (with replacement) elements selected from $[n]$. Then, with probability $1 - \delta$, there is a t-uniform ϵ-approximate σ-smooth Nash equilibrium suported on* $\mathsf{supp}(\hat{X}^1) \times \dots \mathsf{supp}(\hat{X}^m)$.

Given Lemma 7, the main ingredient in the proof of Lemma 11 is the coupling lemma from [36] ([22, Lemma B.1]), which implies that, for any $k \in \mathbb{N}, \sigma \in (0, 1)$ and any σ-smooth distribution p, the following holds: there is a coupling between an i.i.d. sample S of size k from p and an i.i.d. sample S' of size roughly k/σ from the uniform distribution, so that with high probability under the coupling we have $S \subset S'$. To prove Lemma 11, this coupling lemma is applied for each player j, with the sample S corresponding to a sample from a σ-smooth Nash equilibrium as in Lemma 7, and the sample S' corresponding to $\hat{X}^j$.

Given Lemma 11, `QueryEquilibrium` proceeds in the natural way. It samples random k-uniform measures $\hat{X}^j$ as in Lemma 11 (Algorithm 2), and iterates over all t-uniform strategy profiles $\hat{x}$ supported on them (Algorithm 2). The key challenge, however, is to test whether each such strategy profile $\hat{x}$ is in fact an ϵ-approximate σ-smooth equilibrium, in a query-efficient way. In particular, for each player j, we must estimate the value of its best σ-smooth response to $\hat{x}_{-j}$. The naive way to compute this best response is too query-expensive, requiring more than $\Omega(n)$ queries: it would, for each player $j \in [m]$, iterate over all n actions of player j, and average the best σn. Instead, we settle for an approximation to the best σ-smooth deviation, computed as follows: we sample a subset $\mathcal{R}_j$ of sufficiently large size N (which is nevertheless *independent* of n) in Algorithm 2 uniformly at random. Then, we average estimates of player j's values for the best σN sampled actions in $\mathcal{R}_j$ (Algorithm 2, which calls `OptimizeDeviation`, Algorithm 1). This subset $\mathcal{R}_j$ is reused amongst all t-uniform strategy profiles $\hat{x}$ that are considered in `QueryEquilibrium`; reusing the samples $\mathcal{R}_j$ as such allows us to avoid having the query complexity scale exponentially in $\log(1/(\delta\sigma))/\epsilon^2$. If no player j can improve its utility by $\Omega(\epsilon)$, then the algorithm returns the strategy profile $\hat{x}$ (Algorithm 2).

Algorithm 2 `QueryEquilibrium`$((A_1, \ldots, A_m), \sigma, \epsilon, \delta)$: compute weak smooth Nash equilibrium using few queries.

Set $t = \frac{C_0 m \log(32m/\delta\sigma)}{(\epsilon/4)^2}$, $\ell = \frac{\log(4tm/\delta)}{\sigma}$, and $k = t\ell$, where C_0 is the constant of Lemma 7
Set $N := \frac{16 C_1 \cdot tm \log(k/\delta)}{\epsilon^2 \sigma^2}$, for a sufficiently large constant C_1.
For $j \in [m]$, initialize $\hat{A}_j, \tilde{A}_j : [n]^m \to [0,1]$ arbitrarily.
for $j \in [m]$ **do**
  Let $\hat{X}_j \in \Delta^n$ denote the uniform measure over k elements of $[n]$, chosen uniformly at random with replacement.
  Let $\mathcal{S}_j := \mathsf{supp}(\hat{X}_j) \subset [n]$.
for $j \in [m]$ **do**
  Let $\mathcal{R}_j$ be a set consisting of N uniformly random elements of $[n]$, chosen with replacement.
for $j \in [m]$ and each $(b_j, b_{-j}) \in \mathcal{R}_j \times \prod_{j' \neq j} \mathcal{S}_{j'}$ **do**
  Query $A_j(b_j, b_{-j})$ and let $\hat{A}_j(b_j, b_{-j})$ be the result of the query.
for $j \in [m]$ and $b \in \prod_{j' \in [m]} \mathcal{S}_{j'}$ **do**
  Query $A_j(b)$ and let $\tilde{A}_j(b)$ be the result of the query.
for each t-uniform strategy profile $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$ supported on $\mathcal{S}_1 \times \cdots \times \mathcal{S}_m$ **do**
  for $j \in [m]$ **do**
    Set $\hat{v}_j \leftarrow$ `OptimizeDeviation`$(\mathcal{R}_j, \hat{A}_j, \hat{x}, \sigma)$. ▷ *Algorithm 1*
    Set $\tilde{v}_j \leftarrow \tilde{A}_j(\hat{x})$.
  if $\max_{j \in [m]} \{\hat{v}_j - \tilde{v}_j\} \leq \epsilon/2$ **then**
    return the strategy profile $\hat{x}$.
return an arbitrary strategy profile. ▷ *We will show this happens w.p.* $\leq \delta$.

Strong smooth Nash equilibria

It is straightforward to see that an analogue of Theorem 10 cannot hold for strong σ-smooth Nash equilibria: even in the case $m = 1$, there is a constant $c > 0$ so that cn queries are needed to find a strong c-approximate $1/4$-smooth Nash equilibrium.[2] That said, sampling variants of a constant query strong Nash equilibria algorithm are not ruled out and we discuss some these questions in Section 7.

▶ Remark 12 (Query Complexity Lower Bounds). As mentioned earlier, the query and communication complexity of finding Nash equilibria in games is well-studied. In particular, it is known that finding an ϵ_0 approximate Nash equilibrium, for some constant ϵ_0, in a 2-player game requires $\Omega(n^{2-o(1)})$ queries [25, 26, 32]. Further, for m-player games with two actions per player the query complexity of finding an ϵ_0-approximate Nash equilibrium is known to be $2^{\Omega(m)}$ (see [7] and references therein). This implies that for constant σ and ϵ, the result of Theorem 10 is tight up to logarithmic factors in the exponent. Getting tighter lower bounds that fully elucidate the required dependence on σ, ϵ is an interesting avenue for future work.

[2] In particular, a 1-player game is described by a vector $A_1 \in \mathbb{R}^n$: suppose that A_1 is chosen randomly with each entry drawn independently from $\mathrm{Ber}(1/2)$. Consider any randomized algorithm which makes at most $n/8$ queries to A_1 and outputs a distribution $\hat{x} \in \mathcal{K}_{1/4,n}$. Conditioned on the set $\mathcal{Q} \subset [n]$ of entries queried by the algorithm, all values of $(A_1)_i$, for $i \notin \mathcal{Q}$, are uniform and independent bits. Thus, conditioned on any set $\mathcal{Q}$ with $|\mathcal{Q}| \leq n/8$, with at least constant probability, a constant fraction of the mass of $\hat{x}$ is on entries of A_1 which are 0, and thus the suboptimality of $\hat{x}$ with respect to the best $1/4$-smooth deviation (which has value 1 with high probability) is $\Omega(1)$.

5 Efficient Algorithms for Finding Smooth Nash Equilibria

In this section, we introduce algorithms to compute ϵ-approximate σ-smooth Nash equilibria. First, in Section 5.1, we show that, as a relatively straightforward consequence of the results of Sections 3.1 and 4, weak smooth Nash equilibria can be efficiently computed when the approximation and smoothness parameters are constants. Then, we show that a similar conclusion also applies to strong smooth Nash equilibria in Section 5.2, though the proof requires some new ideas.

5.1 Finding Weak Smooth Equilibria in Games

Recall that, in Corollary 9, we showed that, in any normal-form m-player game, there exists an ϵ-approximate σ-smooth Nash equilibrium which is k-uniform (per Definition 8) for $k = O\left(\frac{m \log(m/\sigma)}{\epsilon^2}\right)$, which is *constant* when ϵ, σ, m are constants. Since the number of k-uniform strategies of any player can be enumerated in time n^k, and since it can be efficiently tested whether a k-uniform strategy profile is an ϵ-approximate σ-smooth Nash equilibrium, it follows that we can compute such an equilibrium in time n^{mk}, which is $\text{poly}(n)$ when k, m, σ are constants.

▶ **Theorem 13** (Polynomial-time algorithm for weak smooth Nash)**.** *Let $A_1, \ldots, A_m : [n]^m \to [0, 1]$ denote the payoff matrices of an m-player normal-form game G. Then, for any $\sigma, \epsilon \in (0, 1)$, there is an algorithm running in time $n^{O\left(\frac{m^2 \log(m/\sigma)}{\epsilon^2}\right)}$ which finds a weak ϵ-approximate σ-smooth Nash equilibrium.*

The algorithm and proof for Theorem 13 are analogous to the well-known result of [44], where a *quasipolynomial*-time algorithm was established for computing ϵ-approximate Nash equilibrium in m-player games, when ϵ, m are constants. Theorem 13 improves this result to *polynomial-time* for $\Omega(1)$-smooth equilibria since the parameter k from Corollary 9 as discussed above is constant when $\sigma = \Omega(1)$, whereas for (non-smooth) equilibria, [44] requires $k = O(\log n)$.

5.1.1 Sublinear-time algorithms

Is it possible to beat polynomial-time, i.e., obtain *sublinear-time* algorithms for approximating σ-smooth equilibria? It is straightforward to see that if we restrict our attention to deterministic algorithms, this is not possible, even for $\epsilon = 1/2, \sigma = 1/4, m = 1$. To see this, note that in the case $m = 1$, the game is described simply by a vector $A_1 \in \mathbb{R}^n$. For any deterministic algorithm running in time at most $n/2$, it must make at most $n/2$ queries to A_1. Then there is a fixed sequence $i_1, i_2, \ldots, i_{n/2}$ of the $n/2$ indices at which it queries A_1 when all queries return 0. No matter which fixed distribution in Δ^n the algorithm outputs, there is a subset $\mathcal{S} \subset [n] \backslash \{i_1, \ldots, i_{n/2}\}$ of size $|\mathcal{S}| = \sigma n = n/4$, so that, setting $(A_1)_i = 1$ for $i \in \mathcal{S}$ (and $(A_1)_i = 0$ otherwise) ensures that the algorithm's output distribution yields utility at most $1/2$, whereas deviating to play uniformly on $\mathcal{S}$ yields utility of 1.

One may nevertheless wonder about randomized algorithms; surprisingly, the answer turns out to be vastly different. In particular, it follows from the techniques used to prove our query complexity upper bounds in Section 4 that, in the setting where m, σ, ϵ are constants, there is a randomized *constant time*[3] algorithm that computes ϵ-approximate σ-smooth equilibria in m-player games, with arbitrarily small constant failure probability.

[3] Note that here we work with the $O(\log(n/\epsilon))$-word RAM model. This model allows the algorithm to specify the indices of actions and to access entries of the payoffs in constant time. Note that the payoffs are only needed up to precision $\text{poly}(\epsilon)$ if our goal is to find ϵ-approximate smooth equilibria.

▶ **Theorem 14** (Constant-time randomized algorithm for weak smooth Nash). *Let $A_1, \ldots, A_m : [n]^m \to [0,1]$ denote the payoff matrices of an m-player normal-form game G. Then for any $\sigma, \epsilon, \delta \in (0,1)$, there is an algorithm running in time $\left(\frac{m \log(1/\delta)}{\sigma\epsilon}\right)^{O\left(\frac{m^2 \log(m/\delta\sigma)}{\epsilon^2}\right)}$ which outputs a strategy profile which is a weak ϵ-approximate σ-smooth equilibrium with probability at least $1-\delta$.*

The proof of Theorem 14 follows from Theorem 10 by bounding the running time of `QueryEquilibrium`.

5.2 Finding strong smooth Nash equilibria

The strategy profiles in the equilibria found by the search procedures used to establish Theorems 13 and 14 are k-sparse with $k = \text{poly}(m, \sigma^{-1}, \epsilon^{-1})$. For small values of m, ϵ, σ (which is the main regime of interest), these strategies are certainly not σ-smooth, and thus the equilibria found are not *strong* smooth equilibria. In order to find strong equilibria, we need to implement an additional "smoothening" step. As we shall see, we can do so using linear programming, leading to the following result.

▶ **Theorem 15** (Polynomial-time algorithm for strong smooth Nash, multi-player). *Let $A_1, \ldots, A_m : [n]^m \to [0,1]$ denote the payoff matrices of an m-player normal-form game G. Then for any $\sigma > 0$, there is an algorithm running in time $n^{O\left(\frac{m^4 \log(m/\sigma)}{\epsilon^2}\right)}$ which finds a **strong** ϵ-approximate σ-smooth Nash equilibrium.*

We first describe the proof of Theorem 15 in the case of $m = 2$ players, for which the claimed algorithm is given by `BimatrixStrongSmooth` (Algorithm 3). In this case, for ease of notation, we denote the payoff matrices by $A, B \in [0,1]^{n \times n}$ and the two players' strategies by $x, y \in \Delta^n$. Set $\epsilon_0 = \epsilon/4$, and $k = \frac{2C_0 \log(2/\sigma)}{\epsilon_0^2}$, where C_0 is the constant of Lemma 7. `BimatrixStrongSmooth` proceeds as follows. It iterates over all k-uniform strategy profiles $(\hat{x}, \hat{y})$. For each one which is a weak ϵ_0-approximate σ-smooth Nash equilibrium, it solves a linear program (namely, Equation (3)) which aims to find a σ-smooth strategy profile (x, y) which approximates $(\hat{x}, \hat{y})$ in terms of each player's utility with respect to all σ-smooth deviations of the other player. If such a program is feasible, realized by (x, y), then the program returns such (x, y). Intuitively, the constraints of this program ensure that the fact that the smooth Nash equilibrium constraints are satisfied for $(\hat{x}, \hat{y})$ implies that the smooth Nash equilibrium constraints are satisfied for (x, y).

To complete the proof of the theorem (for $m = 2$), we need to establish two facts. First, that for some k-uniform strategy profile $(\hat{x}, \hat{y})$ which is a weak smooth Nash equilibrium, the program Equation (3) will be feasible. Second, that any feasible solution of Equation (3) (given that $(\hat{x}, \hat{y})$ is a weak smooth Nash equilibrium) is in fact a strong smooth Nash equilibrium. The proof of the first fact uses Lemma 7 with input a *strong* σ-smooth Nash equilibrium (x, y) (which exists by Proposition 5). The conclusion Equation (2) of Lemma 7 can be used to show that (x, y) is a feasible solution to Equation (3) for an appropriate choice of $(\hat{x}, \hat{y})$. The proof of the second fact follows by using the constraints of Equation (3) to derive that any feasible solution must be an approximate strong smooth Nash equilibrium. Finally, we need to ensure that `BimatrixStrongSmooth` can be implemented in the claimed time. The most nontrivial part of this claim is ensuring that the ellipsoid algorithm can efficiently solve Equation (3), in light of the fact that Equation (3) has exponentially many constraints. We discuss this issue in detail in the appendix.

Algorithm 3 `BimatrixStrongSmooth`$((A,B),n,\sigma,\epsilon)$: compute strong smooth equilibria of 2-player games.

1: Set $\epsilon_0 = \epsilon/4$ and $k = \frac{2C_0 \log(2/\sigma)}{\epsilon_0^2}$, where C_0 is the constant of Lemma 7.
2: **for** Each k-uniform strategy profile $(\hat{x},\hat{y})$ **do**
3: **if** $(\hat{x},\hat{y})$ is a weak ϵ_0-approximate σ-smooth Nash equilibrium **then**
4: Solve the following feasibility LP for $x, y \in \mathbb{R}^n$, using the ellipsoid algorithm:

$$\begin{aligned}
\text{Find } x, y \in \mathbb{R}^n: \quad & x, y \in \mathcal{K}_{\sigma,n} && \text{(3a)}\\
& |x^\top A\hat{y} - \hat{x}^\top A\hat{y}| \le \epsilon_0 && \text{(3b)}\\
& |(x')^\top Ay - (x')^\top A\hat{y}| \le \epsilon_0 \quad \forall x' \in \mathcal{K}_{\sigma,n} && \text{(3c)}\\
& |x^\top By' - \hat{x}^\top By'| \le \epsilon_0 \quad \forall y' \in \mathcal{K}_{\sigma,n} && \text{(3d)}\\
& |\hat{x}^\top By - \hat{x}^\top B\hat{y}| \le \epsilon_0. && \text{(3e)}
\end{aligned}$$

5: **if** `SolveLPFeasibility` outputs that above LP is feasible, realized by (x,y) **then**
6: **return** (x,y).

Multiplayer games

The main challenge in extending the above arguments to the case of m-player games, for general $m > 2$, is generalizing the program Equation (3) to the m-player case. At first glance this may seem problematic if it turns out to be necessary to have, say, constraints of the form $|A_j(x'_j, x_{-j}) - A_j(x'_j, \hat{x}_{-j})| \le \epsilon_0$ for all $x'_j \in \mathcal{K}_{\sigma,n}$. Such a constraint is not linear in the program variables $x = (x_1,\dots,x_m)$ when $m > 2$, since $A_j(x'_j, x_{-j})$ is a polynomial of degree $(m-1)$ in x. Fortunately, such constraints are avoidable. We will use a hybrid argument to show, over the course of m steps, that satisfiability of a certain *linear* program yields a strong smooth equilibrium from a weak smooth equilibrium $\hat{x}$. To apply this hybrid argument, we will need a stronger version of Lemma 7, stated below as Lemma 16. In the lemma statement, we generalize the notation $\mathcal{K}_{\sigma,n}$. For any finite set $\mathcal{S}$ and $\sigma > 0$, we let $\mathcal{K}_{\sigma,\mathcal{S}}$ denote the set of distributions $P \in \Delta(\mathcal{S})$ so that $P(s) \le \frac{1}{\sigma|\mathcal{S}|}$ for all $s \in \mathcal{S}$. In particular, below we have $\mathcal{S} = [n]^\ell$ for $\ell \in \mathbb{N}$. We denote elements of $\mathcal{K}_{\sigma,[n]^\ell}$ by $x'_{1:\ell}$; note that such $x'_{1:\ell}$ is in general *not* a product distribution; nevertheless, we will slightly abuse notation by writing, for a fixed sequence $x_{\ell+1:m} = (x_{\ell+1},\dots,x_m) \in (\Delta^n)^{m-\ell}$,

$$A_j(x'_{1:\ell}, x_{\ell+1:m}) := \mathbb{E}_{(b_1,\dots,b_\ell)\sim x'_{1:\ell}} \mathbb{E}_{b_i \sim x_i \ \forall i \ge \ell+1}[A_j(b_1,\dots,b_m)].$$

Similarly, if $p \ge \ell+1$, we write $A_j(x'_{1:\ell,p}, x_{\ell+1:m,-p})$ to denote the corresponding expectation where the pth coordinate is included in the distribution $x'_{1:\ell,p} \in \Delta([n]^{\ell+1})$ and excluded from $x_{\ell+1:m}$.

▶ **Lemma 16.** *Let $A_1,\dots,A_m : [n]^m \to [0,1]$ denote the payoff matrices of an m-player normal-form game G. Let $\epsilon, \sigma \in (0,1)$ be fixed. Set $k = \frac{C_{16} m \log(m/\sigma)}{\epsilon^2}$, for a sufficiently large constant C_{16}. Given a strong 0-approximate σ-smooth Nash equilibrium $x = (x_1,\dots,x_m)$, there is a strategy profile $\hat{x} = (\hat{x}_1,\dots,\hat{x}_m)$ satisfying the following properties:*

1. *All entries of each $\hat{x}_j$, $j \in [m]$, are k-uniform;*
2. *$\hat{x}$ is a weak ϵ-approximate σ-smooth Nash equilibrium.*
3. *The following inequalities hold for each $j \in [m]$:*

$$\max_{x'_{1:\ell-1} \in \mathcal{K}_{\sigma^{\ell-1},[n]^{\ell-1}}} \left| A_j(x'_{1:\ell-1}, \hat{x}_\ell, \hat{x}_{\ell+1:m}) - A_j(x'_{1:\ell-1}, x_\ell, \hat{x}_{\ell+1:m}) \right| \le \epsilon \qquad \forall \ell \in [m] \quad (4)$$

$$\max_{x'_{1:\ell-1,j} \in \mathcal{K}_{\sigma^\ell,[n]^\ell}} \left| A_j(x'_{1:\ell-1,j}, \hat{x}_\ell, \hat{x}_{\ell+1:m,-j}) - A_j(x'_{1:\ell-1,j}, x_\ell, \hat{x}_{\ell+1:m,-j}) \right| \le \epsilon \qquad \forall \ell \in [j-1]. \quad (5)$$

The inequalities Equations (4) and (5) generalize Equation (2) in that a maximum is taken over all σ^ℓ-smooth distributions on $[n]^\ell$, for various $\ell \in [m]$. To prove Theorem 15 given Lemma 16, we use `GeneralStrongSmooth` [22, Algorithm 4], which is similar to `BimatrixStrongSmooth`, except that the program Equation (3) is replaced by [22, Eq. (22)], whose constraints mirror those in Equations (4) and (5). The proof uses these constraints together with a hybrid argument to show that a solution of [22, Eq. (22)] is an approximate smooth Nash equilibrium.

6 Hardness Results for Smooth Equilibria

In the previous sections, we showed that computing ϵ-approximate σ-smooth Nash equilibria is tractable when both ϵ, σ are constants. In this section, we show that this result is optimal in the sense that when either ϵ or σ is an inverse polynomial, then an efficient algorithm is ruled out by standard hardness assumptions for the complexity class PPAD. PPAD is a subclass of the class TFNP of total search problems consisting of those problems in TFNP which have a polynomial-time reduction to the `End-of-the-Line` problem.[4] PPAD-hardness of a problem is typically taken as strong evidence that a problem is intractable; this perspective is supported by the fact that under cryptographic assumptions, PPAD does not have polynomial-time algorithms (see [12, 28] and references therein).

Throughout the section, we restrict to 2-player, n-action games, and show PPAD hardness for various regimes of ϵ, σ. (Analogous lower bounds for games with a constant number m of players immediately follow, being a generalization of the former.)

6.1 Polynomially small smoothness parameter

First, we establish lower bounds for the setting that σ is an inverse polynomial in n and ϵ is a constant. The main ingredient for such lower bounds is the following lemma, which gives a polynomial-time reduction between the problems of finding ϵ-approximate σ-smooth Nash equilibria in 2-player games for differing values of ϵ, σ whose dependence on n differs by a polynomial.

▶ **Lemma 17.** *Let $\epsilon : \mathbb{N} \to (0,1), \sigma : \mathbb{N} \to (0,1)$ be non-increasing real-valued functions of natural numbers. Then for any $c \in (0,1)$, the problem of finding weak $\epsilon(n)$-approximate $\sigma(n)$-smooth Nash equilibrium in 2-player, n-action games has a polynomial-time reduction (in the sense of [22, Definition D.1]) to the problem of finding weak $\epsilon(n^c)$-approximate $\sigma(n^c)$-smooth Nash equilibrium in 2-player, n-action games.*

Moreover, the same conclusion holds if "weak" is replaced by "strong".

The proof of Lemma 17 proceeds via a padding argument: given an n-action normal-form game G, it constructs a game G' with $N := n^{1/c}$ actions which consists of $n^{(1-c)/c}$ copies of G. Given a $\sigma(N)$-smooth $\epsilon(N)$-approximate Nash equilibrium of G', the proof shows how to construct (roughly speaking) a $\sigma(2N)$-smooth $\epsilon(N)$-approximate Nash equilibrium of G. Recalling that $N = n^{1/c}$, we get the result after reparametrizing the functions $\epsilon(\cdot), \sigma(\cdot)$.

Lower bounds for constant ϵ under ETH for PPAD

We observe that $\mathcal{K}_{1/n,n} = \Delta^n$; thus, comparing Definitions 2 and 4, we see that weak (and strong) ϵ-approximate $1/n$-smooth Nash equilibria are equivalent to ϵ-approximate Nash

[4] Recall that *total* search problems are those for which a solution exists given any input.

equilibria. Moreover, we remark that under the exponential time hypothesis for PPAD, it is known [58] that, for some constant ϵ_0, there is no algorithm that computes ϵ_0-approximate Nash equilibria in 2-player n-action games in time $n^{\log^{1-\delta} n}$, for any constant $\delta > 0$. Lemma 17 tells us that computing ϵ_0-approximate Nash equilibria (or, equivalently, ϵ_0-approximate $1/n$-smooth Nash equilibria) is polynomial-time reducible to computing ϵ_0-approximate $1/n^c$-smooth Nash equilibria, for any constant $c \in (0, 1)$. Thus, we obtain the following as a corollary:

▶ **Corollary 18.** *For some $\epsilon_0 \in (0, 1)$, the following holds assuming the ETH for PPAD: for any $\delta, c \in (0, 1)$ there is no algorithm that computes weak ϵ_0-approximate n^{-c}-smooth Nash equilibrium in 2-player n-action games in time $n^{\log^{1-\delta} n}$.*

6.2 Constant smoothness parameter

Next, we establish lower bounds for the setting that σ is a constant and ϵ is an inverse polynomial in n. One might hope that the same technique used to establish Corollary 18 might apply in this setting, by using Lemma 17 for $c = 0$. However, this approach fails since in the limit $c \to 0$, the size of the game G' used to prove Lemma 17, which is $\Theta(n^{1/c})$, diverges. Indeed, to establish hardness of finding weak σ_0-approximate n^{-c}-smooth Nash equilibrium in 2-player, n-action games, for *constant* σ_0 (stated formally in Theorem 21 below), we require additional techniques.

Generalized matching pennies game

Our approach proceeds by showing that it is hard to find approximate smooth Nash equilibria in a certain class of games known as approximate generalized matching pennies games. To do so, we first show that all approximate smooth Nash equilibria in generalized matching pennies games have a certain structure (formalized in Lemma 20) that allows us to relate them to approximate Nash equilibria in such games. We will then apply a result of [18] which shows that it is PPAD-hard to compute approximate Nash equilibria in this subclass of games.

To define generalized matching pennies games, fix an integer $K \in \mathbb{N}$ and write $N = 2K$. Let $M := 2K^3$. Let $A^\star \in \mathbb{R}^{N \times N}$ denote the $K \times K$ block matrix where the 2×2 blocks along the main diagonal have all entries equal to M, and all other entries are 0. (Note that we have $A^\star = M \cdot I_K \otimes J_{2,2}$, where $J_{2,2}$ denotes the 2×2 matrix of 1s and $\otimes$ denotes the Kronecker product.) Let $B^\star = -A^\star$.

▶ **Definition 19.** *Given $K \in \mathbb{N}$ and $N := 2K$, a 2-player N-action game (A, B) is defined to be an* approximate K-generalized matching pennies (K-GMP) game *if all entries of $A - A^\star$ and of $B - B^\star$ are in $[0, 1]$, where $A^\star, B^\star \in \mathbb{R}^{N \times N}$ are defined in terms of K above.*

For ease of notation, we have departed from our convention that all payoffs of the game are in $[0, 1]$; this setting is equivalent to the former by rescaling. Moreover, we have suppressed the dependence of $A^\star, B^\star$ on K in our notation.

Let (A, B) denote an approximate K-GMP game and let $x, y \in \Delta^N$ be strategy vectors. We use the following convention: $\bar{x}, \bar{y} \in \Delta^K$ denote vectors defined by $\bar{x}_k = x_{2k-1} + x_{2k}$ and $\bar{y}_k = y_{2k-1} + y_{2k}$, for $k \in [K]$.

▶ **Lemma 20.** *Suppose $K, M \in \mathbb{N}$, $\sigma \in (0, 1)$, and $\epsilon \geq \frac{8\sigma K^2}{M}$ are given. If $G = (A, B)$ is a K approximate K-GMP game, then a weak 1-approximate σ-smooth Nash equilibrium (x, y) must satisfy the following for all $k \in [K]$:*

$$\bar{x}_k = 1/K \pm \epsilon, \qquad \bar{y}_k = 1/K \pm \epsilon.$$

As a result of Lemma 20, any ϵ-approximate σ-smooth Nash equilibrium (x, y) in an approximate K-GMP game must in fact be an ϵK-approximate Nash equilibrium. The intuition behind this implication is as follows: if some player (say the x-player) had a useful deviation from their strategy x to any fixed action i, they could instead deviate to the strategy whereby they play i with probability $1/K$ and x with the remaining probability. Since $\max_i x_i \leq \max_k \bar{x}_k \leq 1/K + \epsilon$ (by Lemma 20), as long as $2/K + \epsilon \leq \frac{1}{\sigma N} = \frac{1}{2\sigma K}$ (which will hold if $\sigma \leq 1/6$), the resulting strategy is σ-smooth. Moreover, the player would gain a $1/K$ fraction of their utility gain for deviating to i. By combining this observation and the fact that computing approximate Nash equilibria in K-GMP games is PPAD-hard (from [18]), we can show the following theorem.

▶ **Theorem 21** (PPAD-hardness for constant σ). *For any constant $c_1 \in (0, 1)$, the problem of computing weak n^{c_1}-approximate $1/6$-smooth Nash equilibria in 2-player n-action games is PPAD-hard.*

7 Discussions and Open Problems

In this paper we introduced the notion of smooth Nash equilibria and showed that they satisfy many desirable computational properties, namely that they have polynomial time and query algorithms. Our results open numerous avenues for future work, listed below.

- **Improving Dependence on the number of players:** Note that the dependence on m in the exponent of the running time of Theorem 15, namely $\tilde{O}(m^4)$, is worse than that in the corresponding result for finding *weak* smooth Nash equilibrium (Theorem 13), where the dependence is $\tilde{O}(m^2)$. It is an interesting question if this dependence can be improved. In fact, we expect even the $\tilde{O}(m^2)$ dependence can be improved to $O(m \log m)$ using the techniques of [6].
- **Sampling from strong smooth equilibrium:** In Section 4, we gave a constant time algorithm for computing a weak σ-smooth Nash equilibrium and argued that an analogous algorithm for strong σ-smooth Nash equilibrium could not run in sublinear time. One could ask if, instead of outputting a strong σ-smooth Nash equilibrium, we can output a sample from a strong σ-smooth Nash equilibrium in sublinear time.
- **General Notions of Smoothness:** In this paper, we focus on smoothness defined using the L_∞ norm of the Radon-Nikodym derivative (see Remark 6). However, one could consider other notions of smoothness corresponding to other notions of distance from a fixed measure, such as χ^2 divergence or KL divergence. We expect that our techniques can be extended to general settings with the appropriate changes (for example, [22, Lemma B.1] would need to be generalized as in [14]). Exploring this direction is an interesting avenue for future work.
- **General Complexity of Polyhedral and Concave Games:** One perspective on our result obtaining algorithms for smooth Nash equilibria in 2-player games is through the lens of covering numbers of matrices [1]. That is, the proof of Lemma 7 via [22, Lemma A.1] can be viewed as bounding the appropriate covering numbers of the polytope induced by the game matrix acting on the set of smooth distributions. A natural question is whether this notion of covering number can be used to prove a general characterization of complexity of finding Nash equilibria in polyhedral games. In particular, can the lower bound from [58] be generalized to provide a characterization of the complexity of finding Nash equilibria in general settings?

- **Sharp Lower Bounds:** A related question to the above is whether we can improve our lower bounds in Section 6 to match the upper bounds in the full regime of parameters (i.e., for all $\epsilon, \sigma \in (0, 1)$ and all $m \in \mathbb{N}$) both in the case of time complexity and query complexity. Understanding this question even for the case of $\sigma = 1/n$ and $m = 2$ seems to be a challenging open problem.
- **Extensions to Other Equilibrium Concepts:** Given that the notion of smooth Nash equilibria leads significant computational advantages relative to Nash equilibria, it is natural to ask if similar advantages can be obtained for other equilibrium concepts such as Arrow-Debreu market equilibria. Developing a notion of smoothness in these settings which simultaneously has a natural and economically meaningful interpretation while also leading to computational advantages would be ideal.
- **Applications:** As discussed earlier, one can view our notion of smooth Nash equilibria as a strengthening of quantal response equilibria. Given the extensive usage of this notion in modern applications of equilibrium solving to multi-agent reinforcement learning, it would be interesting to see if the notion of smooth Nash equilibria can be applied in multiagent settings to obtain computational advantages.

References

1 Noga Alon, Troy Lee, and Adi Shraibman. The cover number of a matrix and its algorithmic applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

2 Simon P. Anderson, Jacob K. Goeree, and Charles A. Holt. The logit equilibrium: A perspective on intuitive behavioral anomalies. *Southern Economic Journal*, 69(1):21–47, 2002.

3 Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012. doi:10.4086/toc.2012.v008a006.

4 Robert J. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, 1987.

5 Yakov Babichenko. Informational bounds on equilibria (a survey). *SIGecom Exch.*, 17(2):25–45, January 2020.

6 Yakov Babichenko, Siddharth Barman, and Ron Peretz. Empirical distribution of equilibrium play and its testing application. *Math. Oper. Res.*, 42(1):15–29, 2017. doi:10.1287/moor.2016.0794.

7 Yakov Babichenko and Aviad Rubinstein. Communication complexity of approximate nash equilibria. *Games Econ. Behav.*, 134:376–398, 2022. doi:10.1016/j.geb.2020.07.005.

8 Anton Bakhtin, David J. Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H. Miller, and Noam Brown. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/pdf?id=F61FwJTZhb.

9 Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 1193–1200. SIAM, 2009. doi:10.1137/1.9781611973068.129.

10 Avraham Beja. Imperfect equilibrium. *Games and Economic Behavior*, 4(1):18–36, 1992.

11 Alankrita Bhatt, Nika Haghtalab, and Abhishek Shetty. Smoothed analysis of sequential probability assignment, 2023. arXiv:2303.04845.

12 Nir Bitansky, Arka Rai Choudhuri, Justin Holmgren, Chethan Kamath, Alex Lombardi, Omer Paneth, and Ron D. Rothblum. PPAD is as hard as LWE and iterated squaring. In *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part II*, volume 13748 of *Lecture Notes in Computer Science*, pages 593–622. Springer, 2022. doi:10.1007/978-3-031-22365-5_21.

13 Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 1716–1786. PMLR, 2022. URL: https://proceedings.mlr.press/v178/block22a.html.

14 Adam Block and Yury Polyanskiy. The sample complexity of approximate rejection sampling with applications to smoothed online learning. In *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 228–273. PMLR, 2023. URL: https://proceedings.mlr.press/v195/block23a.html.

15 Adam Block, Max Simchowitz, and Alexander Rakhlin. Oracle-efficient smoothed online learning for piecewise continuous decision making. In *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 1618–1665. PMLR, 2023. URL: https://proceedings.mlr.press/v195/block23b.html.

16 Adam Block, Max Simchowitz, and Russ Tedrake. Smoothed online learning for prediction in piecewise affine systems. *CoRR*, abs/2301.11187, 2023. doi:10.48550/arXiv.2301.11187.

17 Shant Boodaghians, Joshua Brakensiek, Samuel B. Hopkins, and Aviad Rubinstein. Smoothed complexity of 2-player nash equilibria. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 271–282. IEEE, 2020. doi:10.1109/FOCS46700.2020.00034.

18 Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3):14:1–14:57, 2009. doi:10.1145/1516512.1516516.

19 Vincent Conitzer and Tuomas Sandholm. Communication complexity as a lower bound for learning in games. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004. doi:10.1145/1015330.1015351.

20 Constantinos Daskalakis. On the complexity of approximating a nash equilibrium. *ACM Trans. Algorithms*, 9(3):23:1–23:35, 2013. doi:10.1145/2483699.2483703.

21 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009. doi:10.1137/070699652.

22 Constantinos Daskalakis, Noah Golowich, Nika Haghtalab, and Abhishek Shetty. Smooth nash equilibria: Algorithms and complexity, 2023. arXiv:2309.12226.

23 Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. doi:10.1126/science.ade9097.

24 Gabriele Farina, Chung-Wei Lee, Haipeng Luo, and Christian Kroer. Kernelized multiplicative weights for 0/1-polyhedral games: Bridging the gap between learning in extensive-form and normal-form games. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6337–6357. PMLR, 17–23 July 2022.

25 John Fearnley, Martin Gairing, Paul Goldberg, and Rahul Savani. Learning equilibria of games via payoff queries, 2014. arXiv:1302.3116.

26 John Fearnley and Rahul Savani. Finding approximate nash equilibria of bimatrix games via payoff queries. *ACM Trans. Economics and Comput.*, 4(4):25:1–25:19, 2016. doi:10.1145/2956579.

27 Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes.* Springer, 1997.

28 Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 579–604. Springer, 2016. doi:10.1007/978-3-662-53008-5_20.

29 Jacob K. Goeree and Charles A. Holt. Asymmetric inequality aversion and noisy behavior in alternating-offer bargaining games. *European Economic Review*, 44(4):1079–1089, 2000.

30 Jacob K. Goeree, Charles A. Holt, and Thomas R. Palfrey. Quantal response equilibrium and overbidding in private-value auctions. *Journal of Economic Theory*, 104(1):247–272, 2002.

31 Jacob K. Goeree, Charles A. Holt, and Thomas R. Palfrey. *Quantal Response Equilibrium: A Stochastic Theory of Games.* Princeton University Press, 2016.

32 Mika Göös and Aviad Rubinstein. Near-optimal communication lower bounds for approximate nash equilibria. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 397–403. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00045.

33 Geoffrey J. Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 360–367, New York, NY, USA, 2008. Association for Computing Machinery.

34 Nika Haghtalab, Yanjun Han, Abhishek Shetty, and Kunhe Yang. Oracle-efficient online learning for smoothed adversaries. In *NeurIPS*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/1a04df6a405210aab4986994b873db9b-Abstract-Conference.html.

35 Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis of online and differentially private learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/685bfde03eb646c27ed565881917c71c-Abstract.html.

36 Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis with adaptive adversaries. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 942–953. IEEE, 2021. doi:10.1109/FOCS52979.2021.00095.

37 Philip A. Haile, Ali Hortaçsu, and Grigory Kosenok. On the empirical content of quantal response equilibrium. *American Economic Review*, 98(1):180–200, March 2008.

38 Sergiu Hart and Yishay Mansour. How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. *Games Econ. Behav.*, 69(1):107–126, 2010. doi:10.1016/j.geb.2007.12.002.

39 Sébastien Hémon, Michel de Rougemont, and Miklos Santha. Approximate nash equilibria for multi-player games. In *International Symposium on Algorithmic Game Theory*, pages 267–278. Springer, 2008.

40 Charles A. Holt and Alvin E. Roth. The nash equilibrium: A perspective. *Proceedings of the National Academy of Sciences*, 101(12):3999–4002, 2004. doi:10.1073/pnas.0308738101.

41 Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 538–545. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492584.

42 Athul Paul Jacob, David J. Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with kl-regularized search. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9695–9728. PMLR, 2022. URL: https://proceedings.mlr.press/v162/jacob22a.html.

43 Satyen Kale. Boosting and hard-core set constructions: a simplified approach. *Electron. Colloquium Comput. Complex.*, TR07-131, 2007. `arXiv:TR07-131`.

44 Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, EC '03, pages 36–41, New York, NY, USA, 2003. Association for Computing Machinery. `doi:10.1145/779928.779933`.

45 R. Duncan Luce. *Individual Choice Behavior: A Theoretical analysis.* Wiley, New York, NY, USA, 1959.

46 Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Strategizing against learners in bayesian games. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 5221–5252. PMLR, 02–05 July 2022.

47 Luke Marris, Ian Gemp, Thomas Anthony, Andrea Tacchetti, Siqi Liu, and Karl Tuyls. Turbocharging solution concepts: Solving nes, ces and cces with neural equilibrium solvers. In *NeurIPS*, 2022. URL: `http://papers.nips.cc/paper_files/paper/2022/hash/24f420aa4c99642dbb9aae18b166bbbc-Abstract-Conference.html`.

48 Daniel L. McFadden. Quantal Choice Analysis: A Survey. In *Annals of Economic and Social Measurement, Volume 5, number 4*, NBER Chapters, pages 363–390. National Bureau of Economic Research, Inc, October 1976. URL: `https://ideas.repec.org/h/nbr/nberch/10488.html`.

49 Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995.

50 Hervé Moulin and Jean-Philippe Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7:201–221, 1978.

51 R. B. Myerson. Refinements of the nash equilibrium concept. *Int. J. Game Theory*, 7(2):73–80, June 1978.

52 Roger B. Myerson. Nash equilibrium and the history of economic theory. *Journal of Economic Literature*, 37(3):1067–1082, September 1999. `doi:10.1257/jel.37.3.1067`.

53 Julien Pérolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas W. Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Rémi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of stratego with model-free multiagent reinforcement learning. *CoRR*, abs/2206.15378, 2022. `doi:10.48550/arXiv.2206.15378`.

54 Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Stochastic, constrained, and smoothed adversaries. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

55 J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965.

56 Robert Rosenthal. A bounded-rationality approach to the study of noncooperative games. *International Journal of Game Theory*, 18(3):273–91, 1989.

57 Aviad Rubinstein. Inapproximability of nash equilibrium. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 409–418, New York, NY, USA, 2015. Association for Computing Machinery.

58 Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. *SIGecom Exch.*, 15(2):45–49, 2017. `doi:10.1145/3055589.3055596`.

59 Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms.* The MIT Press, 2012.

60 Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.

61 Rocco A. Servedio. Smooth boosting and learning with malicious noise. In *Computational Learning Theory, 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 16-19, 2001, Proceedings*, volume 2111 of *Lecture Notes in Computer Science*, pages 473–489. Springer, 2001. doi:10.1007/3-540-44581-1_31.

62 Samuel Sokota, Ryan D'Orazio, J. Zico Kolter, Nicolas Loizou, Marc Lanctot, Ioannis Mitliagkas, Noam Brown, and Christian Kroer. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games, 2023. arXiv:2206.05825.

63 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM*, 52(10):76–84, 2009. doi:10.1145/1562764.1562785.

64 Eric van Damme. *Stability and Perfection of Nash Equilibria*. Springer-Verlag, Berlin, Heidelberg, 1987.

65 Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, 2019.