Communication Complexity of Approximate Nash Equilibria

Yakov Babichenko
Techinon, Israel Institute of Technology
Industrial Engineering and Management
Haifa 3200003, Israel

Aviad Rubinstein
University of California at Berkeley
Department of Electrical Engineering and Computer Sciences
Berkeley, CA 94720-1770, USA

ABSTRACT
For a constant \( \epsilon \), we prove a poly\((N)\) lower bound on the (randomized) communication complexity of \( \epsilon \)-Nash equilibrium \( N \times N \) games. For \( n \)-player binary-action games we prove an \( \exp(n) \) lower bound for the (randomized) communication complexity of \( (\epsilon, \epsilon) \)-weak approximate Nash equilibrium, which is a profile of mixed actions such that at least \( (1 - \epsilon) \)-fraction of the players are \( \epsilon \)-best replying.

CCS CONCEPTS
• Theory of computation → Communication complexity; Exact and approximate computation of equilibria; Convergence and learning in games;

KEYWORDS
Communication complexity, approximate Nash equilibria, convergence rate, uncoupled dynamics

ACM Reference format:
Yakov Babichenko and Aviad Rubinstein. 2017. Communication Complexity of Approximate Nash Equilibria. In Proceedings of 49th Annual ACM SIGACT Symposium on the Theory of Computing, Montreal, Canada, June 2017 (STOC’17), 13 pages.
DOI: 10.1145/3055399.3055407

1 INTRODUCTION
Complexity of equilibria has been studied in several complexity models. In particular, computational complexity, query complexity, and communication complexity. Due to recent developments in the field, the computational and the query complexity of approximate Nash equilibria are quite well understood, even for constant approximation value (see Section 3 for definitions):

- For constant \( \epsilon \), there exists a quasi-polynomial algorithm for \( \epsilon \)-Nash equilibrium in two-player \( N \times N \) games [LMM03]. Under the “Exponential Time Hypothesis for PPAD”, no better algorithm exists [Rub16]. For better approximation value of \( \epsilon = 1/N \) the problem becomes PPAD-complete [DGP09, CDT09].
- For constant \( \epsilon \), it is PPAD-hard to compute an \( \epsilon \)-Nash equilibrium in succinctly representable \( n \)-player games [Rub14], even for graphical polymatrix binary-action games [Rub15].
- The query complexity of \( \epsilon \)-Nash equilibrium in two-player \( N \times N \) games is poly\((N)\) [FGGS13]. The query complexity of \( \epsilon \)-Nash equilibrium in \( n \)-player binary-action games is \( \exp(n) \) [HN13, Bab14, CCT15, Rub16].

The main motivation for this line of research is the insight about the relevance of Nash equilibrium as a predictive solution concept: if specialized algorithms cannot compute an (approximate) equilibrium, it is unreasonable to expect selfish agents to “naturally” converge to one. In the famous words of Kamal Jain, “If your laptop can’t find it, then neither can the market”. (See also discussions in [DGP09, Nis09b, HM10].) Although extremely useful, lower bounds on computational and query complexity suffer from obvious caveats. Computational complexity lower bounds inherently rely on complexity assumptions (such as \( \text{NP} \neq \text{P} \) or \( \text{PPAD} \neq \text{P} \)); even though these assumptions are widely accepted by computer scientists, they make these theorems less accessible to game theorists and economists. Query complexity lower bounds hold only against a fairly restricted model of accessing data, where the algorithm must pay for querying the utility at each strategy profile; what if, for example, we instead give the algorithm access to a best-response oracle? In particular, no lower bounds were known for convergence to approximate Nash equilibrium via randomized uncoupled dynamics (see Subsection 1.1). It is thus of great interest to prove unconditional lower bounds on the much more general model of communication complexity, where each player has unrestricted access to information about her own utility.

While computational and query complexity of approximate Nash equilibria are quite well understood, for communication complexity, only results on pure Nash equilibria or exact Nash equilibria were known: The communication complexity of pure Nash equilibrium in two-player \( N \times N \) game is poly\((N)\) [CS04], and in \( n \)-player games it is \( \exp(n) \) [HM10]. The communication complexity of exact Nash equilibrium in \( n \)-player games is also \( \exp(n) \) [HM10]. No communication complexity lower bounds were known for approximate Nash equilibria. In fact, even for an approximate equilibrium for approximation of value \( \epsilon = 1/\text{poly}(N) \), no bounds were known, see [Nis09a]. In this paper we prove the hardness of approximate Nash equilibria in the randomized^2 communication complexity model.

^The unique Nash equilibrium in the game considered by [HM10] requires probabilities that are doubly-exponentially small. Hence their lower bound is exponential in the number of players, but only polynomial in the size of the description of the equilibrium; see [Nis09a].

^In an earlier version of this paper we proved hardness for deterministic communication. We were able to extend it to randomized communication thanks to a helpful suggestion by Mika Goos [Gio16].
Theorem (Main Theorem, informal). There exists a constant $\epsilon > 0$, such that:

2-player $\epsilon$-Nash equilibrium in two-player $N \times N$ games requires $\text{poly}(N)$ communication.

$n$-player $\epsilon$-Nash equilibrium in $n$-player binary-action games require $2^{\Omega(n)}$ communication. In fact, we prove the exponential lower bound even for a weaker notion of $(\epsilon, \epsilon)$-weak approximate Nash equilibrium, where it is allowed that $\epsilon$-fraction of the players will play an arbitrary action (not necessarily an $\epsilon$-best-reply).

1.1 Uncoupled Dynamics

An underlying assumption of the Nash equilibrium solution is that players predict correctly the (mixed) action of their opponents (or alternatively predict correctly their expected payoff at each action). One justification for this problematic assumption, which appears in the seminal work of John Nash [Nas51], is that in some scenarios players may learn the behaviour of their opponents in cases where the game is played repeatedly. This idea led to an extensive study of learning dynamics and their convergence to Nash equilibrium, see e.g. [You04, HMC13, KL93]. One natural, and general, class of adaptive dynamics is that of uncoupled dynamics [HMC03, HMC06] where it is assumed that players do not know the utilities of their opponents (but observe their past behaviour). The question on the existence of uncoupled dynamics that lead to Nash equilibrium is quite well understood [FY06, HMC06, GL07, Bab12]. Several uncoupled dynamics that converge to approximate Nash equilibrium (or pure Nash equilibrium [You09]) are known. All these dynamics are based on an exhaustive search principle, where at the moment a player realizes she is acting sub-optimally she updates her action to a random one (rather than to an optimal one or a better one). One criticism of these dynamics is that convergence to equilibrium may take an unreasonably long time in large games where the exhaustive search is done over a large space. This led to the study of the rate of convergence of uncoupled dynamics. As pointed out by [CS04] for every solution concept (in particular equilibria solutions), the (randomized) communication complexity of a solution is identical (up to a logarithmic factor) to the rate of convergence by any (randomized) uncoupled dynamics to the solution. This observation initiated the communication complexity study in games. As was mentioned above, the communication complexity, and thus also the rate of convergence of uncoupled dynamics, was known only for exact or pure Nash equilibrium. The question on the rate of convergence of uncoupled dynamics to approximate Nash equilibrium was an open question. Given the fact that all known positive results introduce dynamics that converge to approximate Nash equilibrium, this question is central. Our Main Theorem resolves this open question, yielding the following negative result:

Corollary 1 (Uncoupled Dynamics). There exists a constant $\epsilon > 0$ such that any uncoupled dynamics require:

2-player at least $\text{poly}(N)$ rounds to converge to an $\epsilon$-Nash equilibrium in two-player $N \times N$ games.

1.2 Techniques

Proving communication complexity lower bounds for Nash equilibrium is notoriously challenging for two reasons. The first reason, as is common in hardness of Nash equilibrium in other models, is totality: there always exists at least one (exact) equilibrium, and the proof of existence induces a non-trivial (yet inefficient) algorithm for finding it. In order to construct hard instances we must carefully hide the equilibrium (we can’t just remove it), and make sure that the above algorithm is indeed inefficient for our instances.

Another reason for the communication complexity of approximate equilibrium being an open question for a long time is the fact that there exist efficient non-deterministic communication protocols $\text{polylog}(N)$ for two-player, $\text{poly}(n)$ for $n$-player: verification of equilibrium (exact or approximate) requires only constant communication, and small-representation approximate equilibria always exist (e.g., by [LMM03]). Therefore, the communication complexity lower bounds for approximate equilibria, as we prove in the present paper, show an exponential gap between the non-deterministic and randomized (or even deterministic) communication complexity of a total search problem. We are certainly not the first to show such separations, see e.g. [RW90, KRW95, RM99]. But such separations are still not very common in communication complexity, and for a good reason: for decision problems, they are impossible! The deterministic communication complexity is upper-bounded by the product of the non-deterministic and co-non-deterministic communication complexities [AUY83].

In this work, we overcome both obstacles by combining techniques from hardness of Nash equilibrium in other models [HPV89, MT05, Shm12, Bab14, Rub16] together with the recent simulation theorem for randomized communication complexity [AGJ+17, GPW17]. We note that even given all those techniques, several challenges must be overcome, as is evident by [RW16].

The main steps in our proofs are as follows. First, we prove a randomized query complexity hardness result for the problem of finding the end of a line in a particular constant-degree graph. Then we use a simulation theorem of [AGJ+17, GPW17] to "lift" this query complexity hardness result to randomized communication complexity hardness. We use a construction of [HPV89, Rub16] to embed this line as a continuous Lipschitz function $f : [0, 1]^n \to [0, 1]^n$. Finally, we build on ideas from [MT05, Shm12, Bab14] to construct a two-player (respectively $n$-player) “imitation game” that simulates both the short communication protocol for the computation of $f$, as well as a fixed-point verification. In particular, every (approximate) Nash equilibrium of the game corresponds to an approximate fixed-point of $f$, which in turn corresponds to an end of a line. Proof overview appears in Section 2.1. The formal proofs appear in Section 4.

It is interesting to remark that our result is arguably the first example of a natural problem which exhibits such a gap. To the best of our knowledge, approximate Nash equilibrium is the first problem that is not defined in order to exhibit a gap, but rather happens to have one.
1.3 Additional Related Literature

For two-player $N \times N$ games and $\epsilon \approx 0.382$, [CDF⁺15] show that polylog$(N)$ communication is sufficient for computing an $\epsilon$-approximate Nash equilibrium (improving over a protocol for $\epsilon \approx 0.438$ due to [GP14]).

For the related notion of correlated equilibrium, in $n$-player games with a constant number of actions, it is known that even exact correlated equilibrium can be computed using only poly$(n)$ communication, see [HM10, PR08, JL15]. Interestingly, for exact correlated equilibria, there is an exponential gap between the above communication protocol and the query complexity lower bound of [HN13, BB15]. Further discussion on correlated equilibria appears in Section 5.

For the related problem of finding a fixed point, [RW16] study the communication complexity of approximate fixed point of the decomposition. Namely, Alice holds a Lipschitz function $f : A \rightarrow B$ and Bob holds a Lipschitz function $g : B \rightarrow A$, where $A$ and $B$ are compact convex sets, and their goal is to compute a fixed point of the decomposition $f \circ g$. [RW16] prove that the following version of this problem is communicationally hard: find an approximate fixed point of $f \circ g$ on a grid of $A$, when it is promised that such an approximate fixed point on the grid exists (the problem is not total).

As discussed earlier, the main motivation for studying the (communication) complexity of Nash equilibrium is understanding its relevance as a predictive solution concept. This is a good place to mention a recent work of Roughgarden [Rou14], which highlights another important motivation for studying the complexity of Nash equilibrium: understanding the quality of equilibria. The Price of Anarchy (POA) of a game is the ratio between the social welfare (sum of players' utilities) in an optimum strategy profile, and the social welfare in the worst Nash equilibrium of that game. Roughgarden [Rou14] provides the following widely applicable recipe for lower bounds on POA: if a Nash equilibrium can be found efficiently (in particular, via the non-deterministic protocol due to [LMM03]), but approximating the optimal social welfare requires a higher communication complexity (even for non-deterministic protocols, e.g. by reduction from set disjointness), then clearly not all Nash equilibria yield high social welfare.

2 RESULTS AND PROOF OVERVIEW

For two-player games the communication complexity of $\epsilon$-Nash equilibria is defined to be the problem of finding an $\epsilon$-Nash equilibrium, when Alice holds the utility function of player 1 and Bob holds the utility function of player 2. We prove the following result:

**Theorem 1.** There exists a constant $\epsilon > 0$ such that the randomized communication complexity (RCC) of $\epsilon$-Nash equilibria in two-player $N \times N$ games is at least $N^\epsilon$.

For $n$-player games, we consider a two-party communication problem where the set of players $[n]$ is divided into two disjoint subsets $[n] = A \cup B$. Alice holds the utilities of the players in $A^A$, and Bob holds the utilities of the players in $B^B$. In particular, this communication problem is easier than the $n$-parties communication problem where each player holds his own utility function. Our negative result holds for the notion of weak approximate Nash equilibrium [BPR16], which in particular implies the same negative result for the standard notion of approximate Nash equilibrium (see also Definition 3).

**Theorem 2.** There exists a constant $\epsilon > 0$ such that the randomized communication complexity (RCC) of $(\epsilon, \epsilon)$-weak approximate Nash equilibria in $n$-player binary-action games is at least $2^{n^\epsilon}$.

The formal proofs appear in Section 4. Below we present the main ideas of the proof.

2.1 Overview of the Proofs

As mentioned in the Introduction, the proof consists of four main steps. Below we present the ideas of each step.

**Query Complexity of End-of-any-Line.** Our proof starts with the following query complexity hardness result (Lemma 4.1): There exists a constant degree graph $G = (V, E)$ with $2^{\Theta(n)}$ vertices, such that finding the end of a line in $G$ requires $2^{\Omega(n)}$ queries. In fact, we prove the hardness result for directed graph $G$ where each vertex has outgoing and incoming degree 2. Therefore, the successor and predecessor of each vertex are binary variables. In particular, for each $v \in V$, the information about its role in the line can be represented using only three bits, which we denote $I(v) = (T(v), P(v), S(v)) \in \{0, 1\}^3$:

- (a) Whether the line goes through $v$, which is denoted by $T(v)$,
- (b) Who is the successor of $v$ (if $v$ is on the line), which is denoted by $S(v)$,
- (c) Who is the predecessor of $v$ (if $v$ is in the line), which is denoted by $P(v)$.

**Lemma (QUERY-EOL; informal).** Finding an end of a line with high probability, requires $2^{\Omega(n)}$ queries to $I$.

Proving hardness for randomized query complexity is quite easy (see Lemma 4.1). This can be "lifted" to a lower bound on the randomized communication complexity via the simulation theorem of [AG⁺17, GPW17].

**From Query Complexity to Communication Complexity.** We use a recent simulation theorem to "lift" our randomized query complexity lower bound to a randomized communication complexity bound.

The simulated communicationally hard problem has the following form. For each $v \in V$, Alice holds a triplet of vectors $\alpha_{T,v}, \alpha_{S,v}, \alpha_{P,v} \in \{0, 1\}^M$ where $M = 2^{\Theta(n)}$, and Bob holds a reasonably small input which is just a triplet of indexes $\beta_{T,v}, \beta_{S,v}, \beta_{P,v} \in [M]$. $T(v)$ is given by the decomposition $T(v) = \alpha_{T,v}(\beta_{T,v})$ (similarly for the successor and predecessor). The simulation theorem of [GPW17, AG⁺17] now implies:

**Corollary (CC(SIM-EOL); informal).** Finding an end of a line requires $2^{\Omega(n)}$ bits of communication.

**Embedding as a Continuous Function.** Our next step is to reduce the problem of finding an end of a line to that of finding a Brouwer fixed point. Here, we use a recent construction of a hard function by [Rub16], which improved over the classic construction of Hirsch et al [HPV89].
We embed the points of the discrete graph $G$ in the continuous space $[-1, 2]^{\Theta(n)}$. Specifically, we embed each vertex $v$ of $G$ into a point $x_v$ in $[-1, 2]^{\Theta(n)}$ and each edge $(v, w)$ in $G$ into a (continuous) path in $[-1, 2]^{\Theta(n)}$ that connects the corresponding points $x_v$ and $x_w$. Roughly speaking, [Rub16] shows that we can map a collection of lines in $G$ into a Lipschitz function $f : [-1, 2]^{\Theta(n)} \to [-1, 2]^{\Theta(n)}$ such that:

1. The computation of $f$ can be done using local information about $I$. Namely, for points that are close to $x_v$, it is sufficient to know $I(v)$ to compute $f$. For points that are close to a path that corresponds to the edge $(v, w)$ but far from the points $x_v, x_w$, it is sufficient to know whether $(v, w)$ is an edge in the line (in particular, knowing either $I(u)$ or $I(v)$ suffices). For points that are far from all paths $(v, w)$, $f$ does not depend on $I$ at all (thus can be computed without any communication).

2. Any (normalized) $\| \cdot \|_2$-approximate fixed point of $f$ can be mapped (efficiently) back to an end of some line in $I$.

Property 1 induces the following efficient communication protocol for computing $f(x)$: Bob finds $v$ such that $x$ is close to $x_v$, and sends $\beta_T(v, \beta_S(v, \beta_T(v))$; Alice replies with $I(v) = (\alpha_T(v, \beta_T(v), \alpha_T(v, \beta_T(v)), \alpha_T(v, \beta_T(v)))$, and they each use $I(v)$ to locally compute $f(x)$. (Similarly, if $x$ is close to the path corresponding to edge $(v, w)$, they use a similar protocol to compute $I(v)$ and $I(w)$.)

By Property 2, we have:

**Corollary (CC[Brouwer]; informal).** Finding a (normalized) $\| \cdot \|_2$-approximate fixed point of $f$ requires $2^\Theta(n)$ bits of communication.

**Two-Player Game.** Naively thinking, we would like to design a game where Alice chooses a point $x \in [-1, 2]^{\Theta(n)}$ (on the $\epsilon$-grid) and Bob chooses a point $y \in [-1, 2]^{\Theta(n)}$ (on the $\epsilon$-grid). Alice’s utility will be given by $-\|x - y\|_2^2$, and Bob’s utility will be given by $\|y - f(x)\|_2^2$. Then, by applying similar arguments to those in [MT05, Shm12, Bab14, Rub16] we can deduce that every approximate Nash equilibrium corresponds to an approximate fixed point, and thus also to an end of a line.

However, the above idea is obviously incorrect because Bob’s utility depends on $f$, whereas in the communication problem his utility should depend on the $\beta$s only. Our key idea is to use the fact that $f$ can be computed locally to design a somewhat similar game where similar phenomena to those in the “naive” game will occur in approximate equilibria.

Bob doesn’t know $f$, but to compute $f(x)$ he should only know the local information about the vertex (or vertices) that correspond to $x$. We incentivize Alice and Bob to combine their private information about the corresponding vertex (or vertices) by the following utilities structure.

- Alice’s first component of utility is given by $-\|x - y\|_2^2$. As in the “naive” game, in any approximate Nash equilibrium Alice will play points in the $\epsilon$-cube of the $\epsilon$-grid that contains $\|y\|$ with probability close to one.

Bob tries to guess the vertex $v$ (or the vertices $u, w$) that correspond to the point $x$. Since $x$ (almost always) belongs to the same $\epsilon$-cube, in any (approximate) Nash equilibrium, his guess should be correct (with high probability). In addition, Bob should announce the $\beta$ indexes $\beta_T, \beta_S$ and $\beta_P(v)$ of $(v, w)$. Again, we incentivize him to do so by defining that he should “guess” also these $\beta$ indexes, and in an (approximate) equilibrium his guess should be correct (w.h.p).

- We want Alice to announce $I(v)$ (similarly for $w$ in case of two vertices). Thus, we ask her to guess the decomposition $\alpha_T(v, \beta_T(v)$ where $\alpha_T(v, \beta_T(v)$ are the announced $v$ and $\beta$ by Bob. In (approximate) equilibrium, since Bob announces the correct $v$ and $\beta$ (w.h.p), this incentivizes her to announce the correct $I(v)$ (w.h.p).

Now Bob uses the local information of $I(v)$ (and $I(w)$) to compute $f$. Namely, his utility component is defined by $\|y - f_I(x, I(w)))\|_2^2$, where $f_I(x, I(w))$ is Bob’s “estimation” of $f$ under the relevant local information announced by Alice. In (approximate) equilibrium Alice announces the correct local information (w.h.p), thus Bob computes $f$ correctly (w.h.p).

Summarizing, the (approximate) equilibrium analysis of the presented game is similar to the analysis of the naive game, because in (approximate) equilibrium $f$ is computed correctly (w.h.p). But unlike the naive game, here Alice’s utility depends only on the $\alpha$s and Bob’s utility only on the $\beta$s.

**n-Player Game: $\epsilon$-WSNE.** The $n$-player game reduction is based on the same ideas as the two-player reduction. For clarity, we present first the idea of a reduction that proves the following weaker result:

There exists a constant $\epsilon > 0$ such that the communication complexity of $\epsilon$-well supported approximate Nash equilibrium in $n$-player games with constant number of actions for each player is at least $2^{\Theta(n)}$ for some constant $\epsilon$.

After that, we explain how we can strengthen this result in two aspects: first to improve the constant-number-of-action to binary-action, second to improve the $\epsilon$-well supported Nash equilibrium to $(\epsilon, \epsilon)$-weak approximate equilibrium.

The idea is to replace a single player- Alice- who chooses $x$ in the $\epsilon$-grid of $[-1, 2]^{\Theta(n)}$ by a population of $\Theta(n)$ players $\{p_{xi}\}_{i \leq \Theta(n)}$: each player $p_{xi}$ in the population is responsible for the $i$th coordinate of $x$. The payoff of player $p_{xi}$ is given by $-|x_i - y_i|^2$. This incentivizes player $p_{xi}$ to play either a single, or two adjacent actions, in the $\epsilon$-grid of the segment $[-1, 2]$ (in every WSNE). By looking at the action profile of all $p_{xi}$ players we get the same phenomenon as in the two-player case: every point $x$ in the support of Alice’s players belongs to the same $\epsilon$-cube of the $\epsilon$-grid.

Now, we replace the guess of $v \in [0, 1]^{\Theta(n)}$, that is done by Bob, by population of size $\Theta(n)$ where again each player is responsible to a single coordinate of $v$. Again in a WSNE all players will guess correctly.
Similarly for the guess of \( \beta \): we think of \( \beta \in [M]^{3} \) as an element of \( \{0,1\}^{3\log M} \) and we construct a population of \( 3 \log M \) players, each controls a single bit.

Similarly for Alice’s guesses of \( I^A(v) \) and \( I^A(v) \): we construct 6 players, each chooses a bit.

Finally, we again replace the choice of \( y \in [-1,2]^{\Theta(n)} \) by a population of \( \Theta(n) \) players \( p_{yi} \). Each is responsible to a single coordinate. The payoff of player \( p_{yi} \) is given by \(-|y_i - \left[ f_{I^A(v)}(v_i) - f_{I^B(v)}(x_i) \right]|^2\).

The analysis of this game is very similar to the two-player game analysis.

### 3 PRELIMINARIES

**Notation.** We use \( 0_n \) (respectively \( 1_n \)) to denote the length-\( n \) vectors whose value is 0 (1) in every coordinate. For vectors \( x, y \in \mathbb{R}^n \), we let

\[
\|x - y\|_2^\delta = \sqrt{\frac{1}{n} \sum_{i \in [n]} (x_i - y_i)^2}
\]

denote the normalized 2-norm. Unless specified otherwise, when we say that \( x \) and \( y \) are \( \delta \)-close (or \( \delta \)-far), we mean \( \delta \)-close in normalized 2-norm.

### 3.1 Different Notions of Approximate Nash Equilibrium

A mixed strategy of player \( i \) is a distribution \( x_i \) over \( i \)'s set of actions, \( A_i \). We say that a vector of mixed strategies \( x \in x_i A_i \) is a Nash equilibrium if every strategy \( a_i \) in the support of every \( x_i \) is a best response to the actions of the mixed strategies of the rest of the players, \( x_{-i} \). Formally, for every \( a_i \in \text{supp}(x_i) \)

\[
E_{a_{-i} \sim x_{-i}} \left[ u_i (a_i, a_{-i}) \right] = \max_{a_i' \in A_i} E_{a_{-i} \sim x_{-i}} \left[ u_i (a_i', a_{-i}) \right].
\]

Equivalently, \( x \) is a Nash equilibrium if each mixed strategy \( x_i \) is a best response to \( x_{-i} \):

\[
E_{a \sim x} \left[ u_i (a) \right] = \max_{a_i' \in A_i} E_{a_{-i} \sim x_{-i}} \left[ u_i (a) \right] - \epsilon.
\]

On the other hand, we generalize the first definition of Nash equilibrium in the following stricter definition:

**Definition 2 (\( \epsilon \)-Well-Supported Nash Equilibrium).** \( x \) is a \( \epsilon \)-Well-Supported Nash Equilibrium (\( \epsilon \)-WNE) if every \( a_i \) in the support of \( x_i \) is an \( \epsilon \)-best response to \( x_{-i} \) for every \( a_i \in \text{supp}(x_i) \)

\[
E_{a_{-i} \sim x_{-i}} \left[ u_i (a_i, a_{-i}) \right] \geq \max_{a_i' \in A_i} E_{a_{-i} \sim x_{-i}} \left[ u_i (a_i') \right] - \epsilon.
\]

We can further relax the (already more lenient) notion of \( \epsilon \)-ANE by requiring that the \( \epsilon \)-best response condition only hold for most of the players (rather than all of them).

**Definition 3 ((\( \epsilon, \delta \))-WeakNash [BPR16]).** We say that \( x \) is an \((\epsilon, \delta)\)-WeakNash if for a \((1 - \delta)\)-fraction of \( i \)'s, \( x_i \) is an \( \epsilon \)-best mixed response to \( x_{-i} \):

\[
\Pr_i \left[ E_{a \sim x} \left[ u_i (a) \right] \geq \max_{a_i' \in A_i} E_{a_{-i} \sim x_{-i}} \left[ u_i (a_i') \right] - \epsilon \right] \geq 1 - \delta.
\]

### 3.2 Simulation Theorem for BPP

Let \( D : \{0,1\}^N \rightarrow \{0,1\} \) be a decision problem. We consider the following query complexity models (called also decision tree).

Each query is an index \( k \in [N] \) and the answer is the \( k \)-th bit of the input. The randomized query complexity of \( D \), denoted by \( \text{BPP}_\epsilon^\delta(D) \) where \( \delta \) is the allowed probability of error.

We also consider the following communication complexity models. Here, for every \( k \in [N] \) Alice holds a vector \( a_k \in \{0,1\}^M \) and Bob holds an index \( \beta_k \in [M] \). For some \( M = \text{poly}(N) \). Their goal is to compute \( D \) for the input \((a_1(\beta_1), \ldots, a_N(\beta_N))\). The standard bounded error two-party probabilistic communication complexity of the simulated problem \( D \), denoted by \( \text{BPP}_\epsilon^\delta(\text{Sim-D}) \).

To “lift” from query complexity hardness to communication complexity, we use the following recent simulation theorem for BPP, due to Goos et al [GPW17], and independently due to Anshu et al [AGJ+17].
Theorem 3 (BPP Simulation Theorem, [GPW17, AGJ17, Theorem 2]). There exists $M = \text{poly}(N)$ such that for any constants $0 < \delta < 1/2$,
\[
\text{BPP}^S_{\text{Sim-D}} = \Omega \left( \text{BPP}^S_{\text{D}}(\log N) \right).
\]

4 PROOFS

In Section 4.1 we prove a randomized query lower bound for the end-of-the-line problem. In Section 4.2 we show how the lower bounds of Sections 4.1 can be "lifted" to get a hard problem in the randomized communication complexity models. In Sections 4.3, 4.4, and 4.5 we reduce the communicationally hard end-of-any-line problem to the approximate Nash equilibrium problem.

4.1 A Randomized Query Complexity Lower Bound

Let $G$ be a directed graph with the vertices $V = \{0,1\}^n \times [n+1]$. Each vertex $v = (v_1,v_2,k)$, where $v_1,v_2 \in \{0,1\}^n$ and $k \in [n]$, has two outgoing edges to the vertices $(v_1,v_2^{0+1}(0),k+1)$ and $(v_1,v_2^{0+1}(1),k+1)$, where $v_2^{0+1} = (v_2,\ldots,v_2,0,0)$. We call $v_2^{0+1}(0)$ the 0-predecessor of $v$, and $v_2^{0+1}(1)$ the 1-successor of $v$.

Each vertex $v = (v_1,v_2,n+1)$ has a single outgoing edge to the vertex $v_2,v_2,0)$. Note that the incoming degree of each vertex $u = (v_1,v_2,k) \in V$ is at most two. For $k = 1$ there is a single incoming edge from $(v_2,v_1,1+n+1)$. For $k > 1$ there are two incoming edges from $(v_1,v_2^{0+1}(0),k-1)$ and from $(v_1,v_2^{k+1}(1),k-1)$. We call $(v_1,v_2^{0+1}(0),k-1)$ the 0-predecessor of $v$, and $(v_1,v_2^{k+1}(1),k-1)$ the 1-predecessor of $v$.

We define the Query End-of-the-Line (QUERY-EOTL) to be the problem of finding the end of a line in $G$ that starts at the point $0_{2n-1}$. More formally, we represent a line $L$ in $G$ by a triple $I(v) = (T(v),S(v),P(v))$ where $T(v) \in \{0,1\}$ indicates whether the line goes through $v$, $S(v) \in \{0,1\}$ indicates who is the successor of $v$, and $P(v) \in \{0,1\}$ indicates who is the predecessor of $v$ (here we use the fact that each vertex has outgoing and incoming degree of at most two). Throughout the paper, we slightly abuse notation and use $S(v)/P(v)$ to refer both to the bits, and to the corresponding vertices (i.e. the $S(v)/P(v)$-successor/predecessor of $v$). The end of the line is the vertex $v^* = T(v^*) = 1$ but $T(S(v^*)) = 0$.

Definition 4. The problem Query-EOTL is given by

**Input**: A line $I = (T,S,P)$ over the graph $G$ that starts at the point $0_{2n+1}$.

**Output**: The first bit $[v^*]_1$ of the end of the line vertex.

**Queries**: Each query is a vertex $v \in V$. The answer is the triplet of bits $I(v) = (T(v),S(v),P(v)) \in \{0,1\}^3$.

Lemma 4.1 (Randomized query complexity). For every constant $\delta < 1/2$,
\[
\text{BPP}^S_{\text{Sim-D}} = \Omega(2^n).
\]

Proof. By Yao’s Minimax Theorem it is sufficient to introduce a distribution over paths such every deterministic query algorithm requires $\Omega(2^n)$ queries to determine the first bit of the end of line vertex with probability of at least $1-\delta$. We choose a permutation $\pi$ over $\{0,1\}^n \times \{0,1\}^n$ uniformly at random, and set $\pi(0) = 0_n$. $\pi$ induces a line of length $\Theta(2^n \cdot n)$ over $G$, starting at $0_{2n+1}$, ending at $(\pi(2^n-1),\pi(2^n-1),0)$, and where two consecutive vertices $v = \pi(i)$ and $w = \pi(i+1)$ are mapped to the following line of $n+1$ edges:

\[
T(v,w) \rightarrow \cdots \rightarrow (\pi_1,\pi_2) \rightarrow \cdots \rightarrow (\pi_1,\pi_2,0_{2n+1}) \rightarrow (\pi_1,\pi_2,0_{2n+1}).
\]

Here $(\pi_1,\pi_2)$ denotes the vertex with first $k$ coordinates as in $v$ and the last $n-k$ coordinates as in $w$.

The information of a single query of QUERY-EOTL (for the above class of lines) can be extracted from $\pi(i-1),\pi(i)$ and $\pi(i+1)$. Therefore QUERY-EOTL is at least as hard as the problem of finding the first bit of the last element in a random permutation, where each query returns the previous, the current, and the next vertices. Conditioning on the answers to $k$ queries $\pi(q_1-1),\pi(q_1),\pi(q_1+1),\ldots,\pi(q_k),\pi(q_k+1)$, the last element of the permutation is still uniformly random across all vertices that are not $\pi(q_1),\ldots,\pi(q_k),\pi(q_1-1),\ldots,\pi(q_k-1),\pi(q_k+1),\ldots,\pi(q_k+1)$. This proves that the latter problem requires $\Omega(2^n)$ queries. \hfill $\square$


4.2 Communicationally Hard, Discrete End-of-the-Line Problem

In order to use a simulation theorem (Theorem 3) for randomized communication complexity, we define the following simulation variant of QUERY-EOTL:

Definition 5 (SIM-EOTL). Let $N = 2^n \cdot 2^n \cdot (n+1) \cdot 3$.

**Input**: For each $v \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\}$, Alice receives three vectors $\alpha_S^v,\alpha_P^v,\alpha^v \in \{0,1\}^M$, and Bob receives three indices $\beta_S^v,\beta_P^v,\beta^v \in [M]$.

We define
\[
T(v) = \alpha_S^v(\beta_S^v), S(v) = \alpha_P^v(\beta_P^v), P(v) = \alpha^v(\beta^v). \tag{1}
\]

We simulate the problem QUERY-EOTL, therefore we restrict attention to inputs such that $(T,S,P)$ that are defined in (1) meet all the requirements of QUERY-EOTL.

**Output**: The first bit $[v^*]_1$ of a non-trivial end or start of a line $(v',v^*,0) \neq 0_{2n+1}$.

Applying the randomized Simulation Theorem (Theorem 3) to the query complexity lower bound (Lemma 4.1) gives a lower bound on the randomized communication complexity of a discrete end of line problem SIM-EOTL.

Corollary 2. \[ \text{BPP}^S_{\text{Sim-D}}(\text{SIM-EOTL}) = \Omega(2^n). \]

4.3 Embedding a Line as a Local Lipschitz Function

It will be more convenient to define $G$ as a graph over $\{0,1\}^{2n+\log(n+1)}$. Let $m = \Theta(2n + \log(n+1)) = \Theta(n)$ and let $E: \{0,1\}^{2n+\log(n+1)} \rightarrow \{0,1\}^m$ denote the encoding function of a good binary error correcting code. We embed the discrete graph $G$ into the continuous cube $[-1,2]^{4m}$. The vertex $v$ is embedded to the point $(E(v),E(w),0_m,0_m) \in [-1,1]^{2^{4m}}$, which is called the embedded vertex.

For two vertices $v,w \in V$ such that $(v,w)$ is an edge in the graph $G$, we define five vertices:
\[ x^1(v, w) \triangleq (E(v), E(v), 0_m, 0_m) \]
\[ x^2(v, w) \triangleq (E(v), E(v), 1_m, 0_m) \]
\[ x^3(v, w) \triangleq (E(v), E(w), 1_m, 0_m) \]
\[ x^4(v, w) \triangleq (E(v), E(w), 0_m, 0_m) \]
\[ x^5(v, w) \triangleq (E(w), E(w), 0_m, 0_m). \]

Note that \( x^1(v, w) \) is the embedded vertex \( v \), \( x^5(v, w) \) is the embedded vertex \( w \).

The line that connects the points \( x^i(v, w) \) and \( x^{i+1}(v, w) \) is called a Brouwer line segment. The union of these four Brouwer line segments is called the embedded edge \((v, w)\). It is not hard to check that non-consecutive Brouwer line segments are \( \Omega(1) \)-far one from the other, and in particular it implies that non-consecutive embedded edges are sufficiently far one from the other.

The following Proposition shows that the end-of-any-line problem can be reduced to the problem of finding an approximate fixed point of a continuous Lipschitz function, when the function is "local" in the following sense: every edge in \( G \) is embedded as a path in the continuous hypercube (as described above). For points close to the embedding of an edge, \( f \) depends only on the "local behaviour" of the lines \( f \) at the endpoints of this edge; for all other points, \( f \) is independent of the lines \( f \).

**Proposition 1** (Essentially [Rub16]). There exist constants \( \delta, \epsilon > 0 \) such that given a line \( I = (T, S, P) \over G \) there exists a function \( f = f(I) = [-1, 2]^{4m} \rightarrow [-1, 2]^{4m} \) with the following properties:

1. \( \|f(x) - x\|_2 > \delta \) for every \( x \) that is not \( \epsilon \)-close to the embedded edge of any non-trivial end or start of a line (i.e., the embedding of the edge \((P(x^*), P(x^*))\) such that \( T(P(x^*)) = 1 \) but \( T(S(x^*)) = 0 \) or the edge \((P(x^*), S(x^*))\) for \( x^* \) such that \( T(P(x^*)) = 0, T(S(x^*)) = 1 \) and \( x^* \neq 0_{2+1} \).

2. \( f \) is \( O(1) \)-Lipschitz in \( \| \cdot \|_2 \) norm.

3. \( f \) is local in the sense that it can be defined as an interpolation between a few (in fact, 64) functions, \( \{ f_{l_1, l_2} : [-1, 2]^{4m} \rightarrow [-1, 2]^{4m} \}_{l_1 \in \{0,1\}^3} \), that do not depend on the lines \( l \) and such that:
   a. If the first \( m \)-tuple of coordinates of \( x \) is \( \epsilon \)-close to the embedded vertex \( E(v) \), but the second \( m \)-tuple of coordinates of \( x \) is \( \epsilon \)-far from any encoded vertex \( E(w) \) then \( f_{l_1, l_2}(x) = f(x) \) for every \( l_2 \in \{0,1\}^3 \).
   b. If the second \( m \)-tuple of coordinates of \( x \) is \( \epsilon \)-close to the encoded vertex \( E(v) \), but the first \( m \)-tuple of coordinates of \( x \) is \( \epsilon \)-far from any encoded vertex \( E(v) \) then \( f_{l_1, l_2}(x) = f(x) \) for every \( l_1 \in \{0,1\}^3 \).

4. If none of the above conditions are satisfied, then \( f_{l_1, l_2}(x) = f(x) \) for every \( l_1, l_2 \in \{0,1\}^3 \).

A very similar proposition was recently proved in [Rub16]. Property (3) in Proposition 1 differs a little from the way "locality" is formalized in [Rub16], but it is an immediate consequence of the construction. The proof of Proposition 1 appears in the Appendix of the full version of the paper [BR16].

### 4.4 Two-Player Game

**Theorem (Theorem 1, restated).** There exists a constant \( \epsilon > 0 \) such that the communication complexity of \( \epsilon \)-Nash equilibrium in two-player \( N \times N \) games is at least \( N^\epsilon \).

We construct a two-player game between Alice and Bob of size \( N_A \times N_B \) for

\[ N_A \triangleq (3/\epsilon)^{4m} - 3 = 2^{\Omega(n)} \]
\[ N_B \triangleq (3/\epsilon)^{4m} \cdot (2^{2m+\log(n+1)})^2 \cdot M^3 = 2^{\Omega(n)} \]

such that Alice’s utility depends on \( \{ A_i^T, A_i^S, A_i^P \} \) only, Bob’s utility depends on \( \{ B_i^T, B_i^S, B_i^P \} \) only, and all \( \epsilon^4 \)-approximate Nash equilibria of the game correspond to a \( \delta \)-fixed point of \( f \) from Proposition 1. By property 1 in Proposition 1, any fixed point of \( f \) corresponds to a non-trivial end or start of a line in \( I \).

#### 4.4.1 The game

In this subsection we construct our reduction from Sim-EoAL to the problem of finding an \( \epsilon \)-WSNE.

**Strategies.** Recall that \( \delta \) is the desired approximation parameter for Brouwer fixed point in the construction of Proposition 1. We let \( \epsilon \) be a sufficiently small constant; in particular, \( \epsilon = O(\delta) \) (this will be important later for Inequality (10)).

Each of Alice’s actions corresponds to an ordered tuple \( (x, I^A, I^B) \), where:

- \( x \in [-1, 2]^{4m} \), where the interval \([-1,2] \) is discretized into \([-1, -1 + \epsilon, \ldots, 2 - \epsilon, 2] \);
- \( I^A = \{ I^A_{1}, I^A_{2}, I^A_{3} \} \) and \( I^B = \{ I^B_{1}, I^B_{2}, I^B_{3} \} \).

Each of Bob’s actions corresponds to an ordered tuple \( (y, v^B, w^B, \rho^B, \beta^B) \), where:

- \( y \in [-1, 2]^{4m} \), where the interval \([-1,2] \) is discretized into \([-1, -1 + \epsilon, \ldots, 2 - \epsilon, 2] \);
- \( v^B, w^B \in \{0,1\}^{2m+\log(n+1)} \) are vertices in the graph \( G \).
- \( \rho^B = \{ \rho^B_{y}, \rho^B_{x}, \rho^B_{z} \} \in \{0,1\}^3 \) and \( \beta^B = \{ \beta^B_{y}, \beta^B_{z} \} \in \{0,1\}^2 \) are triples of indexes.

**Utilities.** Alice’s and Bob’s utilities decompose as

\[ U^A = U^A_{\text{imitation}} + U^A_{\text{guess}} + U^A_{\text{guessW}} \]
\[ U^B = U^B_{\text{Brouwer}} + U^B_{\text{guess}} + U^B_{\text{guessW}}. \]

The first component of Alice’s utility depends only on the first components of her and Bob’s strategies; it is given by:

\[ U^A_{\text{imitation}}(x, y) = -\|x - y\|_2^2. \]

Given the first component \( x \in [-1, 2]^{4m} \) of Alice’s strategy, we define two decoding functions \( D_A, D_B : [-1, 2]^{4m} \rightarrow \{0, 1\}^n \) as follows. Let \( R(x) \in \{0, 1\}^m \) be the rounding of the first \( m \)-tuple of coordinates of \( x \) to \{0, 1\}; let \( D_A(x) = E^{-1}(R_1(x)) \in \{0, 1\}^{2m+\log(n+1)} \), where \( E^{-1} \) denote the decoding of the error correcting code from Section 4.3. We define \( D_B(x) \in \{0, 1\}^{2m+\log(n+1)} \) analogously with respect to the second \( m \)-tuple of coordinates of \( x \). The second components of Bob’s
utility is now given by $U_{\text{GUESSV}}^B = 1$ iff he guesses correctly the vertex $D_v(x)$, and the corresponding $\beta$ operation on this vertex. Namely, $U_{\text{GUESSV}}^B(v^B, \beta_v^B; x) = 1$ if $v^B = D_v(x)$ and $\beta_v^B = (\beta_1^B D_v(x), \ldots, \beta_v^B D_v(x))$, and $U_{\text{GUESSV}}^B(v^B, \beta_v^B; x) = 0$ otherwise. Similarly we define Bob’s third component $U_{\text{GUESSW}}^B$ with respect to $D_w(x)$.

Note that Bob knows the indexes $\beta_1^B, \ldots, \beta_v^B, \beta_w^B$ (for every $v$), thus to achieve $U_{\text{GUESS}}^B = 1$ Bob needs to guess correctly only the vertices $D_v(x), D_w(x)$ and announce the corresponding triplet of $\beta$ indexes.

Going back to Alice, the second component of her utility is given by $U_{\text{GUESSV}}^A = 1$ iff she guesses correctly the triplet $I_v^A = (T(v^B), S(v^B), P(v^B))$ when the calculation of $T, S, P$ is done by the decomposition of $\alpha(\beta)$. Namely, $U_{\text{GUESSV}}^A(I_v^A; v^B, \beta_v^B) = 1$ if $I_v^A = (\alpha_{T,v}^B(\beta_v^B), \alpha_{S,v}^B(\beta_v^B), \alpha_{P,v}^B(\beta_v^B))$, and $U_{\text{GUESSV}}^A(I_v^A; v^B, \beta_v^B) = 0$ otherwise. Similarly we define Alice’s third component $U_{\text{GUESSW}}^A$.

Finally, the first component of Bob’s utility is given by:

$$U_{\text{BROUWER}}^B(y; x, \alpha^A) = \|f_{\alpha^A, t_\alpha^A}(y) - x\|^2_2,$$

where the function $f_{\alpha, t}$ is defined in Proposition 1.

### 4.4.4 Analysis of the Game

In this subsection, we prove the reduction from Sim-EoAl to finding an $e^4$-ANE. The proof proceeds via a sequence of lemmas that establish the structure of any $e^4$-ANE.

**Lemma 4.4.** In every $e^4$-ANE $(A; B)$, it holds that $x - E_{x\to y} y = O(e^2)$ with probability of at least $1 - e^2$ (where the probability is taken over A).

**Proof.** We denote $E_i(B) = E_{y\to B}[y_i]$, $E(B) = (E_1(B), \ldots, E_n(B))$ is the vector of expectations, and $\text{Var}(B) = (\text{Var}_{y\to B}[y_1], \ldots, \text{Var}_{y\to B}[y_n])$ is the vector of variances. For every $x$ we can rewrite

$$U_{\text{IMITATION}}^A(x; B) = -E_{y\to B}[x - y]^2_2 = \frac{1}{4m} \sum_{i\in[4m]} E_{y\to B}(x_i - y_i)^2 = \frac{1}{4m} \sum_{i\in[4m]} (x_i - y_i(B))^2 + \text{Var}_{y\to B}[y_i]$$

$$= -\|x - E(B)\|^2_2 - \|\text{Var}(B)\|^2_2. \tag{2}$$

Since the variance of the $y_i$’s, as well as $U_{\text{GUESSV}}^A$ and $U_{\text{GUESSW}}^A$ do not depend on $x$, Alice’s best response to $B$ is

$$x^* = \{E_i(B)\}_{i=1}^n$$

where $\cdot \cdot \cdot$ denotes the rounding to the closest $\epsilon$ integer multiplication. $x^*$ yields a payoff of at least

$$U_{\text{IMITATION}}^A(x^*, B) \geq \frac{e^2}{4} - \|\text{Var}(B)\|^2_2.$$

Note that in every $e^4$-ANE Alice assigns a probability of at most $1 - e^2$ to actions that are $e^2$-far from optimal. By Equation (2) this implies that the probability of Alice to choose a vector $x$ that satisfies $\|x - E(B)\|^2_2 \geq e^2/4 + e^2/4$ is at most $e^2$.

**Lemma 4.5.** In every $e^4$-ANE $(A; B)$, if the first $m$-tuple of coordinates of $E_{y\to B}[y]$ is $\epsilon$-close to the binary encoding $E(v)$ of a vertex $v$, then

$$v^B = v, \quad \beta_v^B = (\beta_v^B, \beta_v^S, \beta_v^P) \tag{3}$$

with probability of at least $1 - O(e^4)$ (where the probability is taken over $B$).

**Proof.** By Lemma 4.2 and the triangle inequality, with probability of at least $1 - e^2$, the first $m$-tuple of $x$ is $O(h)$-close to $E(v)$. Rounding to $R_v(x) \in \{0, 1\}^m$ can at most double the distance to $E(v)$ in each coordinate. Therefore, the Hamming distance of $R_v(x)$ and $E(v)$ is $O(h)$. Hence $R_v(x)$ is correctly decoded as $D_v(x) = v$, with probability of at least $1 - e^2$.

Since $v^B, \beta_v^B$ do not affect $U_{\text{BROUWER}} + U_{\text{GUESSW}}$, Bob’s utility from guessing $v^B = v$, and $\beta_v^B = (\beta_v^B, \beta_v^S, \beta_v^P)$ is at least $1 - e^2$. Whereas his utility from guessing any other guess is at most $e^2$.

Therefore, Bob assigns probability at least $1 - e^4/(1 - 2e^2)$ to actions that satisfy (3).

Similarly, for the second $m$-tuple of $x$ and the vertex $w$, we have:

**Lemma 4.6.** In every $e^4$-ANE $(A; B)$, if the second $m$-tuple of coordinates of $E_{y\to B}[y]$ is $\epsilon$-close to the binary encoding $E(v)$ of a vertex $w$, then

$$w^B = w, \quad \beta_w^B = (\beta_w^B, \beta_w^S, \beta_w^P) \tag{4}$$

with probability of at least $1 - O(e^4)$ (where the probability is taken over $B$).

Since Alice receives the correct $v^B$ and $\beta_v^B$, we also have:

**Lemma 4.7.** In every $e^4$-ANE $(A; B)$, if the second $m$-tuple of coordinates of $E_{y\to B}[y]$ is $\epsilon$-close to the binary encoding $E(v)$ of a vertex $v$, then

$$I_v^A = (\alpha_{T,v}^T(\beta_v^T), \alpha_{S,v}^S(\beta_v^S), \alpha_{P,v}^P(\beta_v^P)) \tag{5}$$

with probability $1 - O(e^4)$ (where the probability is over $A$ and $B$).

**Proof.** Follows immediately from Lemma 4.3 and the fact that $I_v^A$ does not affect $U_{\text{IMITATION}} + U_{\text{GUESSW}}$.

A similar lemma holds for the second $m$-tuple of $x$ and the vertex $w$.

**Lemma 4.8.** In every $e^4$-ANE $(A; B)$, if the second $m$-tuple of coordinates of $E_{y\to B}[y]$ is $\epsilon$-close to the binary encoding $E(v)$ of a vertex $w$, then

$$I_w^A = (\alpha_{T,w}^T(\beta_w^T), \alpha_{S,w}^S(\beta_w^S), \alpha_{P,w}^P(\beta_w^P)) \tag{6}$$

with probability $1 - O(e^4)$ (where the probability is taken over $A$ and $B$).

**Proof.** Follows immediately from Lemmas 4.5 and 4.6 and the “locality” condition in Proposition 1.
The following corollary completes the analysis of two-player games.

**Corollary 3.** For every $\epsilon^4$-ANE $(A; B)$ we have
\[
\|E_{x\rightarrow A}[x] - f(E_{y\rightarrow B}[y])\|_2^2 < \delta.
\]

**Proof.** We recall that in Lemma 4.2 we have proved that
\[
\|x - E_{y\rightarrow B}[y]\|_2^2 = O(\epsilon^2)
\]
with probability $1 - O(\epsilon^2)$, which also implies that $x$ is, with high probability, close to its expectation:
\[
\left|\left|\frac{x - E_{x\rightarrow A}[x]}{\epsilon}\right|^2 \leq \left(\|x - E_{y\rightarrow B}[y]\|_2^2 + \|E_{x\rightarrow A}[x] - E_{y\rightarrow B}[y]\|_2^2\right)
\]
\[
\leq 2\|x - E_{y\rightarrow B}[y]\|_2^2 + 2\|E_{x\rightarrow A}[x] - E_{y\rightarrow B}[y]\|_2^2
\]
\[
= O(\epsilon^2),
\]
with probability $1 - O(\epsilon^2)$. Where the first inequality follows from the Triangle inequality, the second follows from the Arithmetic-Mean Geometric-Mean inequality AM-GM inequality, the third follows from convexity of $\|\cdot\|_2^2$ and the last follows from Lemma 4.2.

Using that $f$ is $O(1)$-Lipschitz together with Equation (5), we get that
\[
\|x - E_{x\rightarrow A}[x]\|_2^2 = O(\epsilon^2)
\]
with probability $1 - O(\epsilon^2)$.

By Lemma 4.7 we know that $f_{\epsilon A, I_{4A}}(x) = f(x)$ with probability $1 - O(\epsilon^2)$, which implies that
\[
\left|\left|\frac{E_{x\rightarrow A}[f_{\epsilon A, I_{4A}}(x)] - E_{x\rightarrow A}[f(x)]}{\epsilon}\right|^2 = O(\epsilon^2).
\]
Using similar arguments to those of Lemma 4.2 we deduce that
\[
\|y - E_{x\rightarrow B}[y]\|_2^2 = O(\epsilon^2)
\]
with probability $1 - O(\epsilon^2)$. As in the derivation of Equation (5), this implies:
\[
\|y - E_{x\rightarrow B}[y]\|_2^2 = O(\epsilon^2)
\]
with probability $1 - O(\epsilon^2)$.

With probability $1 - O(\epsilon^2)$ Inequalities (5),(4),(9),(8),(7),(6) hold simultaneously. In such a case, by the triangle inequality and by applying the inequalities in the exact above order, we have
\[
\|E_{x\rightarrow A}[x] - f(E_{x\rightarrow A}[x])\|_2^2 = O(\epsilon^2 < \delta^2).
\]

**Proof of Theorem 1.** Any communication protocol that solves the $\epsilon^4$-Nash equilibrium problem in games of size $N \times N$ for $N = 2^{O(n)}$ induces a communication protocol for the problem Sim-EOAL: Alice constructs her utility in the above presented game using her private information of the $a$s. Bob constructs his utility the $b$s. They implement the communication protocol to find an $\epsilon^4$-Nash equilibrium. Then both of them know $E_{x\rightarrow A}[x]$ which is a $\delta$-approximate fixed point of $f$ (by Corollary 3). Using $D_4$, they decode the vertex $v^*$ and they know the first coordinate of $v^*$. By Corollary 2, the communication complexity of $\epsilon^4$-Nash equilibrium in games of size $2^{O(n)} \times 2^{O(n)}$ is at least $2^{\Omega(n)}$.

**4.5 n-Player Game**

**Theorem (Theorem 2, restated).** There exists a constant $\epsilon > 0$ such that the communication complexity of $(\epsilon, \epsilon)$-weak Nash equilibrium in $n$-player binary-action games is at least $2^n$.

The proof follows similar lines to those in the proof of Theorem 1. Rather than two players whose actions correspond to $\Theta(n)$ long vectors, we have a player for each bit of the encoded vectors. We construct a game with $8m'$ players for $m' = \Theta(n)$ such that Alice holds the utility function of the first $3m'$ players, Bob holds the utilities of the last $3m'$ players. Alice’s players utilities depend only on the $a$s, Bob’s players utilities depend only on the $b$s, and every $(\epsilon^4/\delta, \epsilon^3/\delta)$-weak approximate Nash equilibrium corresponds to a $\delta$-fixed point of the function $f$ from Proposition 1.

**Players and Actions.** In section 4.4 we have used error correcting code to encode vectors that are deduced from $x$ and $y$. Here, since we consider weak approximate equilibrium, we should add additional encodings for $I_{4A}, I_{4B}, ^\sigma u^5, w^5, ^\sigma w^5, b_{x, y}^B, b_{u, v}^B$. Since we want to use the same number of players for each of the above objects, it will be convenient to encode them in the same space $\{0, 1\}^m$. We let the following be encoding functions of binary error correcting codes with constant (relative) distance:

- $E_1 : \{0, 1\}^3 \rightarrow \{0, 1\}^m$.
- $E_u : \{0, 1\}^{2n + \log(n + 1)} \rightarrow \{0, 1\}^m$.
- $E_{\beta} : \{0, 1\}^{3\log M} \rightarrow \{0, 1\}^m$ (note that $3\log M = \Theta(n)$).

Let $E$ and $m$ denote encoding function and block length of the error correcting code from Section 4.3, i.e.:

- $E : \{0, 1\}^{2n + \log(n + 1)} \rightarrow \{0, 1\}^m$.

For vectors $x, y \in \{-1, 2\}^4m$, we use $(3/\epsilon - 1)$ bits to encode each continuous coordinate (up to precision $\epsilon$) in unary encoding. We choose $m'$ such that $m' = 4(3/\epsilon - 1)m$, so the encoding of each of $x, y$ also takes $m'$ bits. For $E_{\beta}$, we must also have $m' > 3\log M$. Here and henceforth, $\epsilon$ is a sufficiently small constant, satisfying $\epsilon = \Theta(\delta)$.

Instead of having a single player, Alice, with actions

\[
(x, I_A, I_{24}^A) \in \{-1, \ldots, -1 + \epsilon, \ldots, -2 - \epsilon, 2 + \epsilon\}^m \times \{0, 1\}^3 \times \{0, 1\}^3
\]

we replace her by $3m'$ players with binary actions. We have three types of Alice players:

- $x$-type players. Player $x_i$ chooses one of the actions $a_i^j \in \{j + \epsilon\}$ for every $i \in [4m]$ and $j \in \{-1, 1 + \epsilon, 2 - 2\epsilon, 2 - \epsilon\}$. Note that the total number of $x$-type players is $4m(3/\epsilon - 1) = m'$.
- $I^c$-type and $I^m$-type players. Player $I_{i, x}^c$ chooses a bit 0 or 1 for every $i \in [m']$. Similarly for $I_{i, x}^m$-type players.

In the communication problem, we assume that Alice knows the utilities of all the above players.
Instead of having a single player, Bob, with actions
\[(y, v, v^B, v^W, \beta^S_1, \beta^S_2) \in \{(-1, -1 + \epsilon, \ldots, 2 - \epsilon, 2)\}^m \times \{0, 1\}^{2n + \log(n + 1)} \times \{0, 1\}^{2n + \log(n + 1)} \times [M]^3 \times [M]^3 \]
we replace him by 5m' players with binary actions. We have five types of players:

- \(y\)-type players. Player \(y_1\) chooses one of the actions \(b^y_{ij} \in \{j, j + \epsilon\}\) for every \(i \in [4m]\) and \(j \in \{-1, -1 + \epsilon, \ldots, 2 - 2\epsilon, 2\}\).
- \(v\)-type players. Player \(v_i\) chooses a bit 0 or 1 for every \(i \in [m']\). Similarly for \(w\)-type players.
- \(\beta^S\)-type players. Player \(\beta^S_{ij}\) chooses a bit 0 or 1 for every \(i \in [m']\). Similarly for \(\beta^W\)-type players.

In the communication problem, we assume that Bob knows the utilities of all the above players.

**Utilities.** Before getting to the description of the utilities we define the notions of realized number and realized point by a set of players. For every \(i \in [m']\), for simplicity of notations we add a dummy player \(x^i_2\) who has a single action \(a^2 = 2\). Given an action profile \(a = (a^y, a^v, a^w)\) of the players \(x_1, \ldots, x_m\), the realized number \(r(a)\) is \([0, 1]^{m'}\) defined to be the minimal \(j\) such that \(a^2_j = 2\). Note that \(r(a)\) is well defined because the last player \(x^i_2\) plays 2. Given an action profile \(a = (a^j, a^y, a^v, a^w)\) of all \(x\)-type players we denote by \(r(a) = \langle (r(a^j)), \langle -1, 2 \rangle^m \rangle\) the realized point. Similarly we define the realized point of \(y\)-type players.

The utilities are defined similarly to the two-player case with the following differences:

1. For \(x\)-type/\(y\)-type players’ utilities are defined with respect to the realized points of the opponents. In addition, player that is responsible to the \(i\)-th coordinate of the point pays the distance from the \(i\)-th coordinate of the opponent’s point/the \(i\)-th coordinate of the \(f\)-operation of the opponent’s point.

2. For all other types, the \(i\)-th player chooses the value of the \(i\)-th bit in the (alleged) codeword in \([0, 1]^{m'}\).

Formally the payoffs are defined as follows:

- For \(x\)-type players, \(U^x_i(a^y_i, b^i) \triangleq -|a^y_j - r(b^i)|^2\) (we recall that player \(x^i_1\) is allowed to choose only \(a^y_i = j \) or \(a^y_i = j + \epsilon\), and \(b^i\) is the profile of action played by players \(x_2^i\)).
- For a \(v\)-type player \(v_i\), we define \(U^v_i(v_i, a) = 1\) iff he announces the bit \(E_\nu(v)(r(a))\) (where the decoding function \(E_\nu\) is defined in Section 4.4.1). Otherwise, \(U^v_i(v_i, a) = 0\). Namely, the \(i\)-th player tries to guess the \(i\)-th coordinate of the encoded vector \(E_\nu(v) \in [0, 1]^{m'}\), were \(v\) is computed using the decoding operation \(D_{\nu}\) on the realized point \(r(a) \in [-1, 2]^{m'}\). Similarly we define the utility of a \(w\)-type player.
- For a \(\beta^S\)-type player \(\beta^S_{ij}\), we define \(U^y_i(\beta^S_{ij}; a) = 1\) iff he announces the bit \(E_\beta(\beta^S_{ij})(r(a))\). Namely, the \(i\)-th player tries to guess the \(i\)-th coordinate of the encoded vector \(E_\beta(\beta^S_{ij}), \) when in the previous bullet, is computed using decoding. Similarly we define the utilities of \(\beta^W\)-type players.
- For a \(I^S\)-type player \(I^S\), we define \(U^S_i(I^S; \beta^S_{ij}) = 1\) iff he announces the bit \([E_\nu(\beta^S_{ij})(r(a))\] where \(\beta\) is the decoded vector announced by \(v\)-type players and \(\beta\) is the decoded vector of indexes announced by \(\beta^U\)-type players. Similarly we define the utilities of \(I^W\)-type players.

4.5.1 Analysis of the Game. We analyse \((\tau, \tau)\)-weak approximate equilibria for \(\tau = \epsilon^2/82\). The analysis of the game follows the same sequence of Lemmas as the analysis in the two-player case (Section 4.4.2). The analogue of Lemma 4.2 is the following.

**Lemma 4.8.** In every \((\tau, \tau)\)-weak approximate equilibrium \((A, B)\), the realized point by the \(x\)-type players \(r(a)\) satisfies
\[|r(a) - E_{\beta^S_B}(r(b^i))|^2 \leq \epsilon^2\]

with high probability\(^6\) (the probability is over the mixed strategy of the \(x\)-type players).

**Proof.** We say that player \(x^i_1\)’s action \(j\) is wrong if \(E_{\beta^S_B}(r(b^i)) \geq j + \epsilon\); similarly, we say that action \(j + \epsilon\) is wrong when \(E_{\beta^S_B}(r(b^i)) \leq j\). Note that if for some coordinate \(i\), no player \(x^i_1\) plays a wrong action, then the realized number \(r_i(a)\) is close to \(E_{\beta^S_B}(r(b^i))\). We show that instead of an \((\tau, \tau)\)-weak approximate equilibrium we will have such many coordinates \(i\).

Recall that player \(x^i_1\)’s utility when she plays \(j + \epsilon\) her utility is given by
\[u(j + \epsilon) \triangleq E_{\beta^S_B}[U^y_i(j^i + \epsilon; b^i)] = E_{\beta^S_B}[-|j + \epsilon - r(b^i)|^2] = -|j + \epsilon - E_{\beta^S_B}(r(b^i))|^2 - Var_{\beta^S_B}(r(b^i))].\]

Similarly, when she plays \(j + \epsilon\) her utility is given by
\[u(j + \epsilon) = E_{\beta^S_B}[U^y_i(j + \epsilon; b^i) = E_{\beta^S_B}[-|j + \epsilon - r(b^i)|^2] = -|j + \epsilon - E_{\beta^S_B}(r(b^i))|^2 - Var_{\beta^S_B}(r(b^i))].\]

When \(j\) is wrong (i.e., \(E_{\beta^S_B}(r(b^i)) \geq j + \epsilon\)) the difference in the utilities \(u(j + \epsilon) - u(j)\) is given by
\[u(j + \epsilon) - u(j) = -((E_{\beta^S_B}(r(b^i)) - j - \epsilon)^2 + (E_{\beta^S_B}(r(b^i)) - j)^2 = (2E_{\beta^S_B}(r(b^i)) - 2j - \epsilon)^2 \geq \epsilon^2\]

\(^6\) Here and throughout this section, we use “with high probability” to mean with probability approaching 1 as \(n\) grows (in fact, with an exponential dependence); in particular, the probability is approaching 1 faster than any polynomial in \(\epsilon\).
When \( j + \epsilon \) is wrong (i.e., \( E_{b_1 - B}[r(b')] \leq j \)) the difference in the utilities \( u(j) - u(j + \epsilon) \) is given by
\[
u(j) - u(j + \epsilon) = -(j - E_{b_1 - B}[r(b')])^2 + (j + \epsilon - E_{b_1 - B}[r(b')])^2
= (2j - 2E_{b_1 - B}[r(b')] + \epsilon) \epsilon \geq \epsilon^2.
\]
Therefore, player \( x_1 \) can always increase her payoff by at least \( \epsilon^2 \) by deviating from a wrong action. Note that if player \( x_1 \) is \( \tau \)-best replying, she assigns a probability of at most \( \tau/\epsilon^2 \) to a wrong action. In addition, the fraction of \( x \)-type players that are not \( \tau \)-best replying is at most \( 8\tau \) (because we have 8 types of players of equal cardinality). Therefore, in the expected fraction of \( x \)-type players playing a wrong action is at most \( 8\tau/\epsilon^2 \), for example (e.g., by Chernoff bound). Therefore the fraction of coordinates \( i \in [4m] \) where at least one player \( x_i \) plays a wrong action is at most \( 9\tau/\epsilon^2 \) (because we have 3\tau players in each coordinate). So in \((1 - \eta)/\epsilon^2\) fraction of coordinates we have \( |r(a_i) - E_{b_1 - B}[r(b')]| \leq \epsilon \), which implies
\[
|r(a) - E_{b_1 - B}[r(b')]|^2 \leq \frac{1}{4m} \sum_{j=1}^{m} |a_i - E_{b_1 - B}[r(b')]|^2
\leq (1 - \eta)\epsilon^2 + \frac{9\epsilon^2}{\epsilon^2} \leq \frac{8\epsilon^2}{\epsilon^2} = \epsilon.
\]

The analogue of Lemma 4.3 is the following.

**Lemma 4.9.** In every \((\tau, \epsilon)\)-weak approximate equilibrium \((A, B)\), if the first \( m \)-tuple of coordinates of \( E_{b_1 - B}[r(b')] \) is \( \delta \)-close to the binary encoding \( E(v) \) of a vertex \( v \), then

1. The decoding of the action profile of the \( v \)-type players is \( v \) with probability \( 1 - o(\epsilon) \).
2. The decoding of the action profile of the \( \beta^\epsilon \)-type players is \( \beta^\epsilon \) with probability \( 1 - o(\epsilon) \).

**Proof.** Whenever (11) holds, \( D_{r}(r(a)) = v \). In particular, for each \( i \in [m^*] \), \( E_{[D(r(a))]}[E_{v}(u_i)] = E_{v}(u_i) \) with high probability. Therefore, by playing the action \( E_{v}(u_i) \), player \( v_i \) has expected utility of \( 1 - o(\epsilon) \) whereas by playing the action \( 1 - E_v(u_i) \), his expected utility is \( o(\epsilon) \).

Every player that is \( \tau \)-best replying, assigns probability of at least \( 1 - O(\tau) \) to the correct bit. In addition, we have at most \( 8\tau \) fraction of \( v \)-type players who are not \( \tau \)-best replying (because we have 8 types of players of equal cardinality). Therefore the expected fraction of \( v \)-type players who play the wrong bit is \( O(\tau) \). By Chernoff bound, it also holds that with high probability at most an \( O(\tau) \)-fraction of \( v \)-type players play the wrong bit. Whenever this is the case, \( v \) is indeed decoded correctly.

Similarly we prove the second claim in the lemma for \( \beta^\epsilon \)-type players.

**Corollary 4.** In every \((\tau, \epsilon)\)-weak approximate equilibrium \((A, B)\), the expectation of the realized point \( E_{a \sim A}[r(a)] \) is a \( \delta \)-approximate equilibrium of \( f \); i.e.,
\[
\|E_{a \sim A}[r(a)] - f(E_{a \sim A}[r(a)])\|_2 \leq \delta.
\]

**Proof.** The proof is similar to the proof of Corollary 3. We recall that in Lemma 4.8 we have proved that
\[
\|r(a) - E_{b_1 - B}[r(b')]\|_2 \leq \epsilon^2
\]
with high probability. This, in particular, implies that \( r(a) \) is, with high probability, close to its expectation:
\[
\|r(a) - E_{a \sim A}[f_{\tau, \epsilon}(r(a'))]\|_2 \leq \delta
\]
(13)

Using the \( O(1) \)-Lipschitzness of \( f \) we deduce that
\[
\|f(r(a)) - f(E_{a \sim A}[f_{\tau, \epsilon}(r(a'))])\|_2 = O(\epsilon^2)
\]
(15)

with high probability.

Using similar arguments to those of Lemma 4.8 we deduce that
\[
\|r(b) - E_{a \sim A}[f_{\tau, \epsilon}(r(a'))]\|_2 = O(\epsilon^2)
\]
(16)

with high probability, where we recall that \( f^v, f^w \) denote the decoded information of the action profile played by the \( i^v, i^w \)-types players. By an analogous argument to (13),
\[
\|r(b) - E_{b_1 - B}[r(b')]\|_2 = O(\epsilon^2)
\]
(17)

with high probability. By Lemma 4.10,
\[
\|E_{a \sim A}[f_{\tau, \epsilon}(r(a'))] - E_{a \sim A}[f_{\tau, \epsilon}(r(a'))]\|_2 = O(\epsilon^2).
\]
(18)

By Equations (13),(12),(17),(16),(18),(15) (applied exactly in this order) and the triangle inequality we get
\[
\|E_{a \sim A}[r(a')] - f(E_{a \sim A}[r(a')])\|_2 = O(\epsilon^2) < \delta^2.
\]
(19)

**Proof of Theorem 2.** Any communication protocol that solves the \((\epsilon/82, \epsilon/82)\)-weak approximate Nash equilibrium problem in \( \Theta(n) \)-player games with binary actions induces a communication protocol for the problem Sim-EoAL: Alice constructs the utilities of her players using her private information of the \( a_s \). Bob constructs his utility using the \( \beta_s \). They implement the communication protocol to find an \((\epsilon^2/82, \epsilon^2/82)\)-weak approximate Nash equilibrium, and then both of them know \( E_{a \sim A}[r(a)] \) which is a \( \delta \)-approximate fixed point of \( f \) (by Corollary 4). Finally, they round and decode the approximate fixed point to get an end or start of a line.

Using Corollary 2 we deduce that the communication complexity of \((\epsilon/82, \epsilon^2/82)\)-weak approximate Nash equilibrium problem in \( \Theta(n) \)-player games with binary actions is at least \( 2^{O(n)} \).
5 AN OPEN PROBLEM: CORRELATED EQUILIBRIA IN 2-PLAYER GAMES

As mentioned in Section 1.3, it is known that for $n$-player, $O(1)$-action games, even exact correlated equilibrium can be found with poly($n$) deterministic communication complexity (see [HM10, PR08, JLB15]).

In two-player $N \times N$ games, for approximate correlated equilibrium with constant value of approximation, to the best of our knowledge, no non-trivial results are known (neither positive nor negative). Does a polylog($N$) communication protocol for approximate correlated equilibrium exist? Is there a poly($N$) communication lower bound? For small values of approximation, recently [GS17] have shown that $1/N$-correlated equilibrium requires poly($N$) communication.

ACKNOWLEDGEMENTS

We are very grateful for Mika Goos, for very helpful comments on simulation theories. We also thank Noam Nisan for inspiring discussions and suggestions.

Yakov Babichenko gratefully acknowledges the support of the Israeli Science Foundation grant ISF-2021296. Most of this work was done while Aviad Rubinstein was an intern at Microsoft Research, Hertzliyah; he was also partly supported by Microsoft Research PhD Fellowship, NSF grant CCF1408635, and Templeton Foundation grant 3966.

REFERENCES

[AG17] Anurag Anshi, Naresh B Goud, Rahul Jain, Srijita Kundu, and Priyanka Mukhopadhyay. Lifting randomized query complexity to randomized communication complexity. arXiv preprint arXiv:1703.07521, 2017.

[AUY83] Alfred V. Aho, Jeffrey D. Ullman, and Mihalis Yannakakis. On notions of information transfer in VLSI circuits. In Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA, pages 133–139, 1983.

[Bab12] Yakov Babichenko. Completely uncoupled dynamics and nash equilibria. Games and Economic Behavior, 76(1):11–14, 2012.

[Bab14] Yakov Babichenko. Query complexity of approximate nash equilibria. In Proceedings of the 6th Annual ACM Symposium on Theory of Computing, pages 535–544, ACM, 2014.

[BB15] Yakov Babichenko and Siddharth Barman. Query complexity of correlated equilibrium. ACM Transactions on Economics and Computation, 3(4):22, 2015.

[BR16] Yakov Babichenko and Aviad Rubinstein. Communication complexity of approximate nash equilibria. arXiv preprint arXiv:1608.06580, 2016.

[CTT15] Xi Chen, Yu Cheng, and Bo Tang. Well-supported versus approximate nash equilibria: Query complexity of large games. arXiv preprint arXiv:1511.00785, 2015.

[CDF15] Artur Czumaj, Argyrios Deligkas, Michael Fasoulakis, John Fearnley, Marcin Jurdzinski, and Rahul Savani. Distributed methods for computing approximate equilibria. CoRR, abs:1512.03515, 2015.

[CDF09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. J. ACM, 56(3), 2009.

[CS04] Vincent Conitzer and Tuomas Sandholm. Communication complexity as a lower bound for learning in games. In Proceedings of the twenty-first international conference on Machine learning, page 24. ACM, 2004.

[DGPW09] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. SIAM Journal on Computing, 38(1):195–259, 2009.

[FGGS13] John Fearnley, Martin Gairing, Paul W Goldberg, and Rahul Savani. Learning equilibria of games via payoff queries. In EC, pages 397–414, 2013.

[FY06] Dean P. Foster and H. Peyton Young. Regret testing: learning to play Nash equilibrium without knowing you have an opponent. Theoretical Economics, 1(3):341–367, September 2006.

[GL07] Fabrizio Germano and Gabor Lugosi. Global Nash convergence of foster and young’s regret testing. Games and Economic Behavior, 60(1):135–154, 2007.

[Göös16] Mika Göös. August 2016. Private Communication.

[GP14] Paul W Goldberg and Arnoldas Pastink. On the communication complexity of approximate nash equilibria. Games and Economic Behavior, 85:19–31, 2014.

[GS17] Anat Ganor and Karthik C. Srikanta. Communication complexity of correlated equilibrium in two-player games. arXiv preprint arXiv:1704.01044, 2017.

[HM10] Sergiu Hart and Yishay Mansour. How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. Games and Economic Behavior, 69(1):107–126, 2011.

[HMC03] Sergiu Hart and Andreu Mas-Colell. Uncoupled dynamics do not lead to nash equilibrium. American Economic Review, 93(5):1830–1836, 2003.

[HMC06] Sergiu Hart and Andreu Mas-Colell. Stochastic uncoupled dynamics and nash equilibrium. Games and Economic Behavior, 57(2):286–303, 2006.

[HMC13] Sergiu Hart and Andreu Mas-Colell. Simple adaptive strategies: from regret-matching to uncoupled dynamics, volume 4. World Scientific, 2013.

[HN13] Sergiu Hart and Noam Nisan. The query complexity of correlated equilibrium. arXiv preprint arXiv:1305.4874, 2013.

[HPV99] Michael D. Hirsch, Christos H. Papadimitriou, and Stephen A. Vavasis. Exponential lower bounds for finding brouwer fix points. J. Complexity, 5(4):379–416, 1989.

[JLB15] Albert Xin Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. Games and Economic Behavior, 91:347–359, 2015.

[KL93] Elad Kalai and Ehud Lehrer. Rational learning leads to nash equilibrium. Econometrica: Journal of the Econometric Society, pages 1019–1040, 1993.

[KRW05] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. Computational Complexity, 5(3-4):191–204, 1995.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Ananyu Mehta. Playing large games using simple strategies. In Proceedings of the 4th ACM conference on Electronic commerce, pages 36–41. ACM, 2003.

[MT06] Andrew McLennan and Rabee Tourky. From imitation games to kakutani. Manuscript, available at http://www.econ.umn.edu/~mclennan/Papers/papers.html, 2005.

[Nas51] John Nash. Non-cooperative games. Annals of mathematics, pages 286–295, 1951.

[Nis09a] Noam Nisan. Communication complexity of mixed-nash equilibria. https://agtb.wordpress.com/2009/09/28/communication-complexity-of-mixed-nash-equilibria/, 2009.

[Nis09b] Noam Nisan. Economists and complexity. https://agtb.wordpress.com/2009/11/27/economists-and-complexity/, 2009.

[PR08] Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. Journal of the ACM (JACM), 55(3):1–14, 2008.

[RMM99] Ran Raz and Pierre McKenzie. Separation of the monotone nc hierarchy. volume 19, pages 403–435, 1999.

[Rou14] Tim Roughgarden. Barriers to near-optimal equilibria. pages 71–80, 2014.

[Rou15] Aviad Rubinstein. Computational complexity of approximate nash equilibrium in large games. arXiv preprint arXiv:1405.0254, 2014.

[Rub15] Aviad Rubinstein. Inapproximability of nash equilibrium. In Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, pages 409–418, ACM, 2015.

[Rub16] Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. In To appear in FOCS, 2016.

[RW90] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. In Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pages 1019–1040, 1990.

[RW14] Tim Roughgarden and Omri Weinstein. Monotone circuits for matching require linear depth. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, pages 1019–1040, 2013.

[W97] Fabrizio Germano and Gabor Lugosi. Global Nash convergence of Foster and Young’s regret testing. Games and Economic Behavior, 60(1):135–154, 2007.

[You09] H Peyton Young. Strategic learning and its limits. OUP Oxford, 2004.

[You10] H Peyton Young. Learning by trial and error. Games and economic behavior, 65(2):626–643, 2009.