From External to Swap Regret 2.0: An Efficient Reduction for Large Action Spaces

Yuval Dagan
University of California
Berkeley, USA
yuvald@berkeley.edu

Constantinos Daskalakis
Massachusetts Institute of Technology
Cambridge, USA
costis@csail.mit.edu

Maxwell Fishelson
Massachusetts Institute of Technology
Cambridge, USA
maxfish@mit.edu

Noah Golowich
Massachusetts Institute of Technology
Cambridge, USA
nzg@mit.edu

ABSTRACT

We provide a novel reduction from *swap-regret* minimization to *external-regret* minimization, which improves upon the classical reductions of Blum-Mansour and Stoltz-Lugosi in that it does not require finiteness of the space of actions. We show that, whenever there exists a no-external-regret algorithm for some hypothesis class, there must also exist a no-swap-regret algorithm for that same class. For the problem of learning with expert advice, our result implies that it is possible to guarantee that the swap regret is bounded by ϵ after $(\log N)^{\tilde{O}(1/\epsilon)}$ rounds and with $O(N)$ per iteration complexity, where N is the number of experts, while the classical reductions of Blum-Mansour and Stoltz-Lugosi require at least $\Omega(N/\epsilon^2)$ rounds and at least $\Omega(N^3)$ total computational cost. Our result comes with an associated lower bound, which—in contrast to that of Blum-Mansour—holds for *oblivious* and ℓ_1-*constrained* adversaries and learners that can employ distributions over experts, showing that the number of rounds must be $\tilde{\Omega}(N/\epsilon^2)$ or exponential in $1/\epsilon$.

Our reduction implies that, if no-regret learning is possible in some game, then this game must have approximate *correlated equilibria*, of arbitrarily good approximation. This strengthens the folklore implication of no-regret learning that approximate *coarse* correlated equilibria exist. Importantly, it provides a sufficient condition for the existence of approximate correlated equilibrium which vastly extends the requirement that the action set is finite or the requirement that the action set is compact and the utility functions are continuous, allowing for games with finite Littlestone or finite sequential fat shattering dimension, thus answering a question left open in “Fast rates for nonparametric online learning: from realizability to learning in games” and “ Online learning and solving infinite games with an ERM oracle”. Moreover, it answers several outstanding questions about equilibrium computation and/or learning in games. In particular, for constant values of ϵ: (a) we show that ϵ-approximate correlated equilibria in *extensive-form games* can be computed efficiently, advancing a long-standing open problem for extensive-form games; see e.g. “ Extensive-form correlated equilibrium: Definition and computational complexity” and “ Polynomial-Time Linear-Swap Regret Minimization in Imperfect-Information Sequential Games”; (b) we show that the query and communication complexities of computing ϵ-approximate correlated equilibria in N-action normal-form games are $N \cdot \text{poly}\log(N)$ and $\text{poly}\log N$ respectively, advancing an open problem of “Informational Bounds on Equilibria”; (c) we show that ϵ-approximate correlated equilibria of sparsity $\text{poly}\log N$ can be computed efficiently, advancing an open problem of “Simple Approximate Equilibria in Large Games”; (d) finally, we show that in the adversarial bandit setting, sublinear swap regret can be achieved in only $\tilde{O}(N)$ rounds, advancing an open problem of “From External to Internal Regret” and “Tight Lower Bound and Efficient Reduction for Swap Regret”.

CCS CONCEPTS

• **Theory of computation → Convergence and learning in games**; **Online learning theory**.

KEYWORDS

swap regret, correlated equilibrium, large action space

ACM Reference Format:
Yuval Dagan, Constantinos Daskalakis, Max Fishelson, and Noah Golowich. 2024. From External to Swap Regret 2.0: An Efficient Reduction for Large Action Spaces. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3618260.3649681

1 INTRODUCTION

No-regret learning has been a central topic of study in game theory and online learning over the last several decades [10, 17, 20]. In view of the worst-case nature of the associated learning guarantee, no-regret learning has found myriad applications in a variety of settings, with varying degrees of restriction on the adversary's behavior. They are also particularly salient in game theory due to their connection with decentralized equilibrium computation. Indeed, it is well understood that, if players in a normal-form game iteratively update their strategies using a no-regret learning algorithm, then the empirical distribution of their strategies over time converges to a type of correlated equilibrium, depending on the notion of regret used.

STOC '24, June 24–28, 2024, Vancouver, BC, Canada
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649681

The most commonly studied type of regret, called *external regret*, measures the amount of extra utility that the agent could have gained if, instead of her realized sequence of strategies, she had instead played her best fixed action in hindsight. In a multi-agent interaction, if each agent uses a sublinear external regret learning algorithm to iteratively update her strategy, the empirical distribution of the agents' play converges to a *coarse correlated equilibrium* (CCE). A CCE is a correlated distribution over actions under which no player can improve her utility if, instead of playing according to the distribution, she unilaterally switches to playing any single fixed action. CCEs are a convex relaxation of *Nash equilibria*, which are computationally intractable even for normal-form games with a finite number of actions per player [11, 12]. While a plethora of efficient algorithms for minimizing external regret are known even when the size of the game is large (see e.g. [8, 10, 17]), the twin notions of external regret and coarse correlated equilibrium are too weak for many applications. In particular, the notion of CCE does not capture the fact that the action sampled from the CCE distribution for some player may leak information about what actions were sampled for the other players, which the player could potentially exploit to improve her utility.

Using the perspective of Bayesian rationality, Aumann introduced the concept of *correlated equilibrium* (CE), which corrects for this deficit [2]. A CE is a correlated distribution with the property that the action sampled for each player maximizes her expected utility against the distribution over actions sampled for the other players, *conditioning on the action sampled for this player.* Like CCE, the concept of CE is a convex relaxation of Nash equilibrium, and it can be reached in a decentralized manner by averaging the empirical play of algorithms which have sublinear *swap regret.* This measures the amount of extra utility that the agent could have gained, in hindsight, if she were to go back in time and transform the strategies that she played using the best, fixed *swap function* The stronger nature of swap regret leads it to have numerous applications, including in calibration and multicalibration [18, 23] and Bayesian games [26], amongst others.

1.1 Swap Regret: Challenges with Large Action Spaces

Despite the more appealing guarantees satisfied by swap regret minimization and its twin notion of CE, no-swap regret learning algorithms have not been as widely adopted as no-external regret ones. This is due in part to the substantially inferior quantitative guarantees offered by the best-known swap-regret-minimizing algorithms in terms of their dependence in the number of actions available to the learner. In particular, existing algorithms are inefficient in many settings of interest where the action space is exponentially large in the game's description complexity, or even infinite. To illustrate, we first consider the case of no-regret learning with a finite set of N actions, which is known as the "experts setting." Standard external-regret-minimizing algorithms, such as exponential weights [10], guarantee that the average external regret over T rounds is bounded by ϵ as long as $T \gtrsim \frac{\log N}{\epsilon^2}$.[1] In contrast, the best-known swap-regret-minimizing algorithms, which are all based on generic reductions from swap regret minimization to external regret minimization [6, 29], guarantee that the average swap regret over T rounds is ϵ as long as $T \gtrsim \frac{N \log N}{\epsilon^2}$. Thus, prior work left an *exponential gap* between the best-known algorithms for swap and external regret. It was explicitly asked by Blum and Mansour [6] if this gap could be improved. This gap is particularly noteworthy in light of many recent applications of no-regret learning, such as for solving games such as Poker [7] and Diplomacy [4], all of which have the property that N is moderate or large.

Prior work also left a polynomial-sized gap in the *bandit* setting, in which the learner must choose a single action each round and only receives the utility for that action. While it is known that $T \gtrsim \frac{N^2 \log N}{\epsilon^2}$ rounds suffice [21, 22] to ensure that swap regret is bounded by ϵ, the best known lower bound was that $\frac{N \log N}{\epsilon^2}$ rounds are necessary [6, 21]. The bandit setting is particularly useful due to its applications in reinforcement learning [22] and related areas.

Prior to the present work, the gap between swap regret and external regret was even larger in settings where the number of actions available to the learner is unbounded or infinite. For instance, suppose that each agent's action space is the set of parameters of a neural network: multi-agent interactions in which each agent chooses a neural network can be used to model tasks such as training generative adversarial networks [19], autonomous driving [28], or economic decision making [31]. In these cases, the number of possible networks is very large. In a more general setting, the action space is typically assumed to be constrained by a combinatorial complexity measure, such as the *Littlestone dimension* or *sequential fat shattering dimension.* In particular, if the learner's action space has Littlestone dimension L, then it was known [1, 5] that as long as the number T of rounds satisfies $T \geq \frac{L}{\epsilon^2}$, there is an algorithm which achieves at most ϵ external regret.[2] Since the reductions of [6, 29] for bounding swap regret assume that the number N of actions is bounded, prior to our work it was not known whether any class of finite Littlestone dimension has an algorithm with $o(T)$ swap regret, leaving open the possibility of an *infinite* gap between swap and external regrets for classes of finite Littlestone dimension.

Gaps in equilibrium computation. The above gaps between swap and external regret also manifest as gaps between the best known results for computing ϵ-approximate CE and CCE in various models of computation. We improve upon these gaps in the following settings:

- *Normal-form games with N actions.* We consider two computation problems. For simplicity we assume the number of players and ϵ are constants.
  - In the *communication complexity* model of computation, ϵ-CCE may be computed with $O(\log^2 N)$ bits of communication using no-external regret algorithms together with a sampling procedure. In contrast, prior to this work, the best known bound for ϵ-CE was exponentially worse, $O(N \log^2 N)$, using the swap regret algorithm of [6].
  - In the *query complexity* model of computation, ϵ-CCE may be computed using $O(N \log N)$ queries. Prior to this work,

[1] We consider normalized regret throughout the paper, i.e., we divide the cumulative regret by the number of rounds T.

[2] This bound is optimal; see [5].

the best known bound of $O(N^2 \log N)$ was quadratically worse for ϵ-CE.
  - Finally, ϵ-CCE which are poly $\log(N)$-sparse may be computed in polynomial time [3], whereas prior to this work, it was unknown how to efficiently compute ϵ-CE which are $o(N)$-sparse, marking another exponential gap (in the sparsity).
- In *infinite games* of Littlestone dimension $L < \infty$, for constant $\epsilon > 0$, ϵ-CCE may be found in a decentralized manner by running $O(L)$ rounds of no-external regret algorithms [13]. In contrast, prior to our work it was not known if ϵ-CE even *exist* in games of finite Littlestone dimension.
- Finally, in *extensive form games* with description length n denoting the size of the tree, for which the number of actions[3] typically scales as $N = \exp(\Theta(n))$, ϵ-CCE may be computed in poly(n) time (e.g., [15]). However, prior to this work, the best known algorithms for computing ϵ-CE took time exponential in n. Determining the complexity of ϵ-CE was a well-known open question in this field; see e.g. [16, 30].[4]

1.2 Main Results: Near-Optimal Upper and Lower Bounds for Swap Regret

Our main upper bound is a new reduction from swap regret to external regret: any no-external regret learning algorithm can be transformed into a no-distributional swap regret learner. We assume that a learner chooses, in each iteration $t \in [T]$, a distribution $\mathbf{x}^{(t)} \in \Delta_{\mathcal{X}}$ over a set of actions $\mathcal{X}$. After observing $\mathbf{x}^{(t)}$, an adversary selects a reward function $\mathbf{f}^{(t)} : \mathcal{X} \to \mathbb{R}$, and the learner receives the reward $\mathbf{f}^{(t)}(\mathbf{x}^{(t)}) = \mathbb{E}_{s^{(t)} \sim \mathbf{x}^{(t)}} \mathbf{f}^{(t)}[s^{(t)}]$. We assume the adversary's choices $\mathbf{f}^{(t)}$ are constrained to lie in some convex function class $\mathcal{F} \subset [0,1]^{\mathcal{X}}$.

Theorem 1.1. *Let $d, M \in \mathbb{N}$ be given, and suppose that there is a learner for some function class $\mathcal{F}$ which achieves external regret of ϵ after M iterations. Then there is a learner for $\mathcal{F}$ (*`TreeSwap`*; Algorithm 1) which achieves a swap regret of at most $\epsilon + \frac{1}{d}$ after $T = M^d$ iterations.*

If the per-iteration runtime complexity of the external-regret learner is C, then the swap regret learner `TreeSwap` *has a per-iteration amortized runtime complexity of $O(C)$.*

Notice that the swap regret of `TreeSwap` depends only on the external regret of the assumed learner, and is *independent of the number of actions* of the learner. In particular, it holds also for exponentially large or even infinite function classes.

Applications: concrete swap regret bounds. As applications of Theorem 1.1, in the setting of constant ϵ, we are able to close all of the gaps discussed for the regret minimization and equilibrium computation problems in Section 1.1. We begin with the case that the learner has N actions, also known as *learning with expert advice.* By applying Theorem 1.1 with action set $\mathcal{X} = [N]$, and reward class given by all $[0,1]$-bounded functions, i.e., $\mathcal{F} = [0,1]^{[N]}$, we obtain:

Corollary 1.2. *Fix $N \in \mathbb{N}$ and $\epsilon \in (0,1)$, and consider the setting of online learning with N actions. Then for any T satisfying $T \geq (\log(N)/\epsilon^2)^{\Omega(1/\epsilon)}$, there is an algorithm that, when faced with any adaptive adversary, has swap regret bounded above by ϵ. Further, the amortized per-iteration runtime of the algorithm is $O(N)$, its worst-iteration runtime is $O(N/\epsilon)$ and its space-complexity is $O(N/\epsilon)$.*

In the regime of constant ϵ, Corollary 1.2 improves on the previously best-known complexity of $T \geq \tilde{\Omega}(N/\epsilon^2)$, providing an exponential improvement in the dependence on N. We note that N/ϵ^2 is tight for all ϵ in the *non-distributional* setting, where the learner is allowed to randomize over her actions but has to play a concrete action rather than a probability distribution [6]. Thus, Theorem 1.2 shows that for a constant ϵ, a distributional swap regret of at most ϵ can be achieved with exponentially fewer rounds. Another advantage of our result is an improved total runtime of $\tilde{O}(N)$ for constant ϵ, compared to the previous $\Omega(N^3)$ runtime of [6], which answers an open question from that paper for constant ϵ.

Next, we apply Theorem 1.1 to an arbitrary function class $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ whose dual has finite Littlestone dimension. That is, the class of functions indexed by actions of the learner, which, via slight abuse of notation, we denote by $\mathcal{X} := \{f \mapsto f(s) \ : \ s \in \mathcal{X}\} \subset \{0,1\}^{\mathcal{F}}$. [5]

Corollary 1.3 (Swap regret for Littlestone classes)**.** *If the class $\mathcal{X}$ has Littlestone dimension at most L, then for any $T \geq (L/\epsilon^2)^{\Omega(1/\epsilon)}$, there is a learner whose swap regret is at most ϵ. In particular, games with finite Littlestone dimension admit no-swap regret learners and thus have ϵ-approximate CE for all $\epsilon > 0$.*

We remark that even the *existence* of approximate CEs in games of finite Littlestone dimension was previously unknown.

Finally, we prove an upper bound on the swap regret in the bandit setting that is tight up to poly $\log N$ factors when $\epsilon = O(1)$. While the result is not a direct consequence of Theorem 1.1, the overall structure of the algorithm and analysis are similar:

Theorem 1.4 (Bandit swap regret)**.** *Let $N \in \mathbb{N}, \epsilon \in (0,1)$ be given, and consider any $T \geq N \cdot (\log(N)/\epsilon)^{O(1/\epsilon)}$. Then there is an algorithm in the* adversarial bandit *setting with N actions which achieves swap regret bounded above by ϵ after T iterations.*

Concretely, for $\epsilon = O(1)$, Theorem 1.4 guarantees that $\tilde{O}(N)$ rounds suffice to achieve swap regret of at most ϵ. Interestingly, this implies that, for obtaining swap regret bounded by $\epsilon = O(1)$, there is only a polylogarithmic gap between the number of rounds needed in the adversarial bandit setting and the full-information non-distributional setting [6]. This is in stark contrast to the situation for *external regret*, for which there is an *exponential* gap between the full-information non-distributional setting (where $O(\log N)$ rounds suffice) and the adversarial bandit setting (where $\Omega(N)$ rounds are needed) [25]. Finally, we remark that our algorithm for the bandit setting is readily seen to be computationally efficient.

[3] An action is specified by a contingency plan, mapping each information set to an outgoing edge at that information set.

[4] To be clear, ϵ-CE here refers to the notion of ϵ-approximate *normal-form correlated equilibrium* (sometimes denoted ϵ-NFCE), as opposed to relaxations of this notion, such as extensive-form correlated equilibrium, which have been recently proposed, motivated in part by the apparent intractability of ϵ-NFCE [30].

[5] Technically, in order to ensure convexity of $\mathcal{F}$, we need to apply Theorem 1.1 to the *convex hull* of $\mathcal{F}$. Doing so does not materially affect the guarantees.

Applications: equilibrium computation. Next, we discuss implications of Corollary 1.2 for equilibrium computation. By considering the setting where players in a normal-form game run (a slight variant of) the algorithm of Corollary 1.2, we may obtain low query and communication protocols for learning in normal-form games.

Corollary 1.5 (Query and communication complexity upper bound)**.** *In normal-form games with a constant number of players and N actions per player, the communication complexity of computing an ϵ-approximate CE is $\log(N)^{\tilde{O}(1/\epsilon)}$ and the query complexity of computing an ϵ-approximate CE is $N \cdot \log(N)^{\tilde{O}(1/\epsilon)}$.*

Finally, we remark that our main reduction can be used to obtain efficient algorithms for computing ϵ-CE when N is exponentially large if there are nevertheless efficient external regret algorithms. This is the case in particular for the setting of extensive form games [14, 15, 24]:

Corollary 1.6 (Extensive form games)**.** *For any constant ϵ, there is an algorithm which computes an ϵ-approximate CE of any given extensive form game, with runtime polynomial in the representation of the game (i.e., polynomial in the number of nodes in the game tree and in the number of outgoing edges per node).*

Corollary 1.6 is an immediate consequence of Theorem 1.1 and the fact that there are efficient external regret minimization algorithms in extensive-form games. This is classically known as a consequence of the *counterfactual regret minimization* algorithm, i.e., Theorem 4 of [32], or improved recent results, such as Theorem 5.5 of [15], as well as [9, 14, 24].

Near-matching lower bounds. Theorem 1.1 and Corollary 1.2 require the number of rounds T to be exponential in $1/\epsilon$, where ϵ denotes the desired swap regret. The following lower bound shows this dependence is necessary, even facing an *oblivious* adversary that is constrained to choose reward vectors with *constant ℓ_1 norm*:

Theorem 1.7 (Lower bound for swap regret with oblivious adversary)**.** *Fix $N \in \mathbb{N}$, $\epsilon \in (0, 1)$, and let T be any number of rounds satisfying*

$$T \leq O(1) \cdot \min\left\{\exp(O(\epsilon^{-1/6})), \frac{N}{\log^{12}(N) \cdot \epsilon^2}\right\}. \tag{1}$$

Then, there exists an oblivious *adversary on the function class $\mathcal{F} = \left\{\mathbf{f} \in [0, 1]^N \middle| \|\mathbf{f}\|_1 \leq 1\right\}$ such that any learning algorithm run over T steps will incur swap regret at least ϵ.*[6]

Theorem 1.7 establishes:

- The first $\tilde{\Omega}\left(\min(1, \sqrt{N/T})\right)$ swap regret lower bound for *distributional swap regret.*
- The first $\tilde{\Omega}\left(\min(1, \sqrt{N/T})\right)$ swap regret lower bound achieved by an oblivious adversary. In particular, the adversary samples reward functions $\mathbf{f}^{(1:T)}$ from some fixed distribution before the first round of learning, independently of the actions of the learner. Moreover, this distribution is independent of the description of the learning algorithm.
- The first $\tilde{\Omega}\left(\min(1, \sqrt{N/T})\right)$ swap regret lower bound from an adversary that plays distributions over a function class of *constant* Littlestone dimension (namely, the class of point functions on $[N]$, which has Littlestone dimension 1).

Finally, while the lower bound of $\exp(\epsilon^{-1/6})$ rounds necessary (to ensure swap regret is bounded by ϵ) from Theorem 1.7 does not quite match the upper bound of $\exp(\epsilon^{-1})$ (from Corollary 1.2; ignoring $\log N$ factors), we can improve the lower bound somewhat if we allow the adversary to be adaptive. In particular, we give an *entirely different* (and somewhat simpler) construction which shows that $T \geq \exp(\Omega(\epsilon^{-3}))$ rounds are necessary to ensure that swap regret is bounded above by ϵ.

Concurrent work. We have been recently made aware of concurrent work by Peng and Rubinstein [27], which proves similar upper and lower bounds to Theorems 1.1 and 1.7. Moreover, they derive a similar set of applications for equilibrium computation problems.

1.3 Proof Sketch of the Upper Bound (Theorem 1.1)

We overview the proof of Theorem 1.1. Recall that we are given $M, d \in \mathbb{N}$, and will construct a swap regret learner for $T = M^d$ rounds. We assume access to a no-external regret learner ($\mathtt{Alg}_{\mathtt{Ext}}$) that, over M rounds, produces a sequence of distributions which has an external regret of at most $\epsilon \in (0, 1)$. We will show that there is an algorithm TreeSwap with swap regret at most $O(\epsilon + \frac{1}{d})$.

Algorithm 1 TreeSwap($\mathcal{F}, \mathcal{X}$, Alg, T, M, d)

```
Require: Action set $\mathcal{X}$, utility class $\mathcal{F}$, no-external regret algorithm Alg, time horizon $T$, parameters $M, d$ with $T \leq M^d$.
For each sequence $\sigma \in \bigcup_{h=1}^{d} \{0, 1, \ldots, M-1\}^{h-1}$, initialize an instance of Alg with time horizon $M$, denoted $\mathtt{Alg}_\sigma$.
for $1 \leq t \leq T$ do
    Let $\sigma = (\sigma_1, \ldots, \sigma_d)$ denote the base-$M$ rep of $t-1$.
    for $1 \leq h \leq d$ do
        if $\sigma_{h+1} = \cdots = \sigma_d = 0$ or $h = d$ then
            if $\sigma_h > 0$ then
                call $\mathtt{Alg}_{\sigma_{1:h-1}}$.update$\left(\frac{1}{M^{d-h}} \sum_{s=t-M^{d-h}}^{t-1} \mathbf{f}^{(s)}\right)$
            end if
            $\mathtt{Alg}_{\sigma_{1:h-1}}$.curAction $\leftarrow$ $\mathtt{Alg}_{\sigma_{1:h-1}}$.act().
        end if
    end for
    Output the uniform mixture
    $\mathbf{x}^{(t)} := \frac{1}{d} \sum_{h=1}^{d} \mathtt{Alg}_{\sigma_{1:h-1}}$.curAction, and observe $\mathbf{f}^{(t)}$.
end for
```

The algorithm simulates multiple instances of $\mathtt{Alg}_{\mathtt{Ext}}$ at *levels* $i = 0, 1, \ldots, d-1$, which are arranged as the nodes a depth-d M-ary tree. We traverse the $T = M^d$ leaves of the tree in order, one per round. At each round t, the TreeSwap algorithm outputs the uniform mixture over the d distributions produced by the $\mathtt{Alg}_{\mathtt{Ext}}$ instances on the root-to-leaf path for the current leaf.

Updating $\mathtt{Alg}_{\mathtt{Ext}}$ instances. Next we describe how the $\mathtt{Alg}_{\mathtt{Ext}}$ instances at each node of the tree are updated over the course of the T rounds. Notice that the M^i instances of $\mathtt{Alg}_{\mathtt{Ext}}$ at each

[6] If we replace the requirement that $\|\mathbf{f}\|_1 \leq 1$ with $\|\mathbf{f}\|_\infty \leq 1$, then the bounds are slightly improved, with $1/6$ replaced with $1/5$ and 12 with 10.

level i are used during a disjoint set of M^{d-i} consecutive rounds: the first algorithm in level i is used during rounds $1, \dots, M^{d-i}$, the second during rounds $M^{d-i}+1, \dots, 2M^{d-i}$, and so on. Each of these $\mathtt{Alg}_{\mathtt{Ext}}$ instances will be run in a *lazy* fashion, only producing M different distributions over the corresponding M^{d-i} rounds. The first algorithm in level i will be called to produce a distribution at round 1, and then play that distribution repeatedly for rounds $1, \dots, M^{d-i-1}$. At round M^{d-i-1}, we finally update the state of the algorithm based on the average reward over the previous M^{d-i-1} rounds. The algorithm then produces a new distribution, which it plays for rounds $M^{d-i-1}+1, \dots, 2M^{d-i-1}$, and so on. All algorithms in level i will be run in this way: updating every M^{d-i-1} rounds on an *average reward function* from the previous M^{d-i-1} rounds. According to the guarantee of our external regret algorithm, each of these instances will have external regret bounded above by ϵ relative to the M distributions it produces and the M average reward functions on which it updates.

Swap regret bound. To bound the swap regret of our algorithm, let us first denote by R_i the average reward of all the algorithms $\mathtt{Alg}_{\mathtt{Ext}}$ in level i over all T rounds. Further, for each $i = 0, \dots, d$, we define S_i in the following manner. For each block of rounds of size M^{d-i}, consider the average reward of the best fixed action in hindsight; then we define S_i to be the average of these best-in-hindsight rewards over all blocks at level i. By the external regret guarantee of $\mathtt{Alg}_{\mathtt{Ext}}$, we know that $S_i - R_i \le \epsilon$. This is due to the fact that each level-i algorithm is run during a block of M^{d-i} rounds, and therefore competes with the best fixed action over that block of rounds. Moreover, the contribution to the swap regret of `TreeSwap` from all algorithms at level i is at most $S_{i+1} - R_i$. This is due to the fact that these level-i algorithms repeatedly play actions for blocks of M^{d-i-1} rounds, and so the best swaps of these actions correspond to the best fixed actions over the blocks of that length. The total swap regret is then bounded by

$$\frac{1}{d}\sum_{i=0}^{d-1}(S_{i+1} - R_i) = \frac{1}{d}\sum_{i=0}^{d-1}(S_i - R_i) + \frac{S_d - S_0}{d} \le \epsilon + \frac{1}{d},$$

where we used that $S_i - R_i \le \epsilon$ and that $S_d - S_0 \le 1$ since the utilities are bounded between 0 and 1. This concludes the proof.

1.4 Proof Sketch of the Lower Bound (Theorem 1.7)

To prove Theorem 1.7, we consider two cases depending on the values of N, T (which correspond to which of the terms on the right-hand side of (1) is larger):

Case 1: $N \ge 4T$. As a warm-up, we present a strategy for the adversary that does not quite work. Then, we show how to fix it, describing a true strategy that achieves the desired lower bound. In both the warm-up and true strategies, we will consider an adversary that selects "point function" rewards at each time step t: one action $u^{(t)} \in [N]$ will receive a reward of 1, and all other actions 0. To describe these strategies, we will relate the actions to vertices in a full binary tree. Assume that $T = 2^D$ for some $D \in \mathbb{N}$. Consider a full binary tree of depth D, containing $2^{D+1} - 1$ vertices, and denote its vertex set by V. In our warm-up construction, each vertex will correspond to a single-action.[7] That is, in each round t, the learner plays a vertex $v^{(t)} \in V$ and the adversary plays a vertex $u^{(t)} \in V$. The reward of the learner is $\mathbf{1}[v^{(t)} = u^{(t)}]$. While our lower bound is valid also for the distributional setting, we analyze for simplicity the case where the learner has to play a concrete action in each round. However, the same proof goes through if they are allowed to output a distribution over vertices. Here is the strategy of the adversary: let us order the children of each internal node by 'left' and 'right'. This will create an ordering over the root-to-leaf paths in the tree: the first path goes left until reaching the leaf, the second path goes left except for the last step that is taken right, etc. Enumerate the paths by indices in $[T]$ according to this ordering, where path t is called P_t. For each time step t, the adversary will select at random a vertex, out of the $D+1$ vertices in path P_t, according to the following distribution: the probability of the vertex at depth i is $(i+1)/(1+2+\cdots+(D+1))$. The important property here, is that vertices get higher weight as we go down the tree.

Let us analyze the swap regret of any learner facing this adversary. Recall that this approach does not quite work for the adversary, but we will show how to fix it. At a high level, the goal of the adversary strategy is to increase swap regret every round t as follows.

- If the learner plays an internal node on P_t, they will incur swap regret to the node at depth one greater on P_t, which gets slightly more expected reward.
- If the learner plays the leaf node of P_t, there is a constant probability that the adversary will not play this leaf. In this case, the learner will incur a swap regret from that leaf.
- If the learner plays a node not on P_t, they will receive no reward and incur swap regret.

However, the problem is that the learner will later have a chance to undo this incurred swap regret. For an internal node v, let $[\underline{t}_v, \bar{t}_v]$ be the interval of time steps for which v is on P_t. Let's say the learner plays v heavily during the first half of these time steps $[\underline{t}_v, (\underline{t}_v + \bar{t}_v)/2]$: the times in which the left child of v is present on P_t. The learner will incur swap regret from v to its left child. However, let's say the learner continues to play v for much of the second half of the interval $[(\underline{t}_v + \bar{t}_v)/2, \bar{t}_v]$. During these times, the adversary never plays the left child of v, while continuing to play v with some probability, undoing the swap regret of v.

To account for this, the adversary actually plays the following "true" strategy instead. In this strategy, each node is associated with two actions: $v, \dot{v}$. During the first half $[\underline{t}_v, (\underline{t}_v + \bar{t}_v)/2]$, as before, the adversary will choose v with probability $(\mathrm{depth}(v)+1)/(1+\cdots+(D+1))$. However, with probability $1/2$, the adversary replaces v with $\dot{v}$ at the halfway point $t = (\underline{t}_v + \bar{t}_v)/2$. That means, with probability $1/2$, for the second half $[(\underline{t}_v + \bar{t}_v)/2, \bar{t}_v]$, the adversary will choose $\dot{v}$ with probability $(\mathrm{depth}(v)+1)/(1+\cdots+(D+1))$ and never choose v. On the other hand, with probability $1/2$ there is no replacement, and the adversary continues to select v with probability $(\mathrm{depth}(v)+1)/(1+\cdots+(D+1))$, not $\dot{v}$. This accomplishes the following. With probability $1/4$, v will get replaced at the halfway point $t = (\underline{t}_v + \bar{t}_v)/2$ but the left child of v will *not* get replaced at its halfway point $t = (3/4)\underline{t}_v + (1/4)\bar{t}_v$. In this event, which happens with constant

[7] Eventually, we will consider a construction wherein each vertex corresponds to two actions.

probability, both v and its left child will be played with non-zero probability over the interval $[\underline{t}_v, (\underline{t}_v + \bar{t}_v)/2]$ and at no other time steps. Thus, the learner will not have a chance to undo the swap regret.

The formal version of this argument breaks into case work. We lower bound the swap regret of action v by considering the reward of swapping v with the best of the 4 actions associated with its 2 children. In addition, we consider a swap from v to the root of the tree, in the event that v is played on many rounds outside of the interval $[\underline{t}_v, \bar{t}_v]$. Bounds for each case culminate in the following. Letting X_v be the total number of rounds the learner plays action v, we show that the best swap of action v increases expected total reward by $\tilde{\Omega}(X_v)$. Thus, the total swap regret of the learner would be $\tilde{\Omega}\left(\sum_v X_v\right) = \tilde{\Omega}(T)$, and her average swap regret would be $\tilde{\Omega}(1) = \Omega\left(\frac{1}{\text{polylog}(T)}\right)$, which is at least ϵ for $T \leq \exp(\text{poly } 1/\epsilon)$.

Case 2: $N < 4T$. This case is very similar to the first. In fact, we define the adversary strategy in a general way that avoids breaking into cases manually. The key difference in this case is that we don't have enough actions to associate 2 actions with each node of a full binary tree with T leaves. In this case, we consider a full binary tree with $N/4$ leaves. We again have an adversary that iterates through the root-to-leaf paths in DFS order. In this case though, each iteration corresponds to a batch in which the adversary plays a distribution over that root-to-leaf path repeatedly for $4T/N$ time steps.

The other key difference here is that we need to associate each leaf with two actions $\ell, \dot{\ell}$. As discussed before, there is a single coin flip for each of the internal nodes v that determines if it gets replaced at time $t = (\underline{t}_v + \bar{t}_v)/2$. On the other hand, for leaf nodes ℓ, we have a coin flip at every single time step in its batch, determining which of $\ell, \dot{\ell}$ will be played with non-zero probability. Letting X_ℓ be the total number of rounds the learner plays ℓ, due to the random deviation in the selection of the adversary, the expected swap regret to $\dot{\ell}$ in $\tilde{\Omega}\left(\sqrt{X_\ell}\right)$. Thus, the total swap regret of a learner that plays only leaf actions over all $N/4$ batches will be $\tilde{\Omega}\left(\sum_\ell \sqrt{T/N}\right) = \tilde{\Omega}\left(\sqrt{NT}\right)$ and her average swap regret will be $\tilde{\Omega}\left(\sqrt{N/T}\right)$, as desired.

1.5 Discussion

We next compare the guarantees of Corollary 1.2, Theorem 1.7, and our adaptive lower bound (recall that our adaptive lower bound yields a quantitatively stronger lower bound than Theorem 1.7 with the stronger notion of *adaptive* adversary). Let $\mathscr{M}(N, \epsilon)$ denote the smallest $T_0 \in \mathbb{N}$ so that, for all $T \geq T_0$, there is a learning algorithm whose action set is $[N]$ and for which the swap regret over T rounds is bounded above by ϵ. Then by Corollary 1.2, Theorem 1.7, and our adaptive lower bound,[8]

$$\Omega(1) \cdot \min\left\{\frac{\log N}{\epsilon^2} + 2^{\Omega(\epsilon^{-1/3})}, \frac{N}{\log^{10}(N) \cdot \epsilon^2}\right\}$$
$$\leq \mathscr{M}(N, \epsilon) \leq O(1) \cdot \min\left\{(\log(N)/\epsilon^2)^{O(1/\epsilon)}, \frac{N \log N}{\epsilon^2}\right\}. \quad (2)$$

The second terms in the upper and lower bounds in (2) differ by a poly $\log(N)$ factor, which is insignificant compared to N. The first terms differ in that (a) the $\log(N)$ is in the base of the exponent in the upper bound but not the lower bound, and (b) the exponent in the lower bound is $\epsilon^{-1/3}$ but is ϵ^{-1} in the upper bound. We remark that the term $2^{\Omega(\epsilon^{-1/3})}$ in the lower bound comes from our adaptive lower bound, which is stronger than the bound of $2^{\Omega(\epsilon^{-1/6})}$ from Theorem 1.7.

The swap regret bound of Corollary 1.2 improves upon those of of Stoltz-Lugosi and of Blum-Mansour [6, 29] when the accuracy parameter ϵ and number of actions N satisfy $\epsilon \gg \frac{\log \log N}{\log N}$. In particular, for $\epsilon = O(1)$, our reduction bounds swap regret above by ϵ via an efficient algorithm with $T \leq \text{poly} \log N$ rounds, whereas [6, 29] require $T \geq \tilde{\Omega}(N)$.

ACKNOWLEDGEMENTS

Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich are supported by NSF Awards CCF-1901292, DMS-2022448, and DMS2134108, a Simons Investigator Award, the Simons Collaboration on the Theory of Algorithmic Fairness, and a DSTA grant. Noah Golowich is supported by a Fannie & John Hertz Foundation Fellowship and an NSF Graduate Fellowship.

REFERENCES

[1] Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. 2021. Adversarial laws of large numbers and optimal regret in online classification. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. https://doi.org/10.48550/arXiv.2101.09054

[2] Robert Aumann. 1974. Subjectivity and Correlation in Randomized Strategies. *Journal of Mathematical Economics* 1 (1974), 67–96. https://doi.org/10.1016/0304-4068(74)90037-8

[3] Yakov Babichenko, Siddharth Barman, and Ron Peretz. 2014. Simple Approximate Equilibria in Large Games. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation* (Palo Alto, California, USA) *(EC '14)*. Association for Computing Machinery, New York, NY, USA, 753–770. https://doi.org/10.1145/2600057.2602873

[4] 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. 2022. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science* 378, 6624 (2022), 1067–1074. https://doi.org/10.1126/science.ade9097

[5] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. 2009. Agnostic Online Learning.. In *COLT*, Vol. 3. 1.

[6] Avrim Blum and Yishay Mansour. 2007. From External to Internal Regret. *J. Mach. Learn. Res.* 8 (2007), 1307–1324. https://doi.org/10.1007/11503415_42

[7] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890. https://doi.org/10.1126/science.aay2400

[8] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5, 1 (2012), 1–122. https://doi.org/10.48550/arXiv.1204.5721

[9] Andrea Celli, Alberto Marchesi, Tommaso Bianchi, and Nicola Gatti. 2019. Learning to Correlate in Multi-Player General-Sum Sequential Games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, Vol. 32. https://doi.org/10.48550/arXiv.1910.06228

[10] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games.* Cambridge university press. https://doi.org/10.1017/CBO9780511546921

[11] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. 2009. Settling the Complexity of Computing Two-Player Nash Equilibria. *J. ACM* (2009). https://doi.org/10.48550/arXiv.0704.1678

[12] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39, 1 (2009). https://doi.org/10.1137/070699652

[13] Constantinos Daskalakis and Noah Golowich. 2022. Fast rates for nonparametric online learning: from realizability to learning in games. In *Proceedings of the*

[8] The $\frac{\log N}{\epsilon^2}$ term in the lower bound of (2) comes the classic external regret lower bound. The second term in the minimum of the upper bound of (2) comes from the Blum-Mansour algorithm [6].

54th Annual ACM SIGACT Symposium on Theory of Computing. 846–859. https://doi.org/10.48550/arXiv.2111.08911

[14] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2021. Better Regularization for Sequential Decision Spaces: Fast Convergence Rates for Nash, Correlated, and Team Equilibria. In *Proceedings of the 22nd ACM Conference on Economics and Computation* (Budapest, Hungary) *(EC '21)*. Association for Computing Machinery, New York, NY, USA, 432. https://doi.org/10.48550/arXiv.2105.12954

[15] Gabriele Farina, Chung-Wei Lee, Haipeng Luo, and Christian Kroer. 2022. 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 (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 6337–6357. https://doi.org/10.48550/arXiv.2202.00237

[16] Gabriele Farina and Charilaos Pipis. 2023. Polynomial-Time Linear-Swap Regret Minimization in Imperfect-Information Sequential Games. In *Conference on Neural Information Processing Systems (NeurIPS)*. https://doi.org/10.48550/arXiv.2307.05448

[17] Drew Fudenberg and David Levine. 1998. *The Theory of Learning in Games*. MIT Press. https://doi.org/10.1016/s0898-1221%2898%2990111-0

[18] Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. 2023. Multicalibration as Boosting for Regression. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 460, 34 pages. https://doi.org/10.48550/arXiv.2301.13767

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014). https://doi.org/10.48550/arXiv.1406.2661

[20] James Hannan. 1957. Approximation to Bayes risk in repeated play. Contributions to the Theory of Games, Vol. III. Princeton University Press, 97–139. https://doi.org/10.1515/9781400882151-006

[21] Shinji Ito. 2020. A Tight Lower Bound and Efficient Reduction for Swap Regret. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 18550–18559. https://doi.org/10.5555/3495724.3497282

[22] Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. 2022. V-Learning – A Simple, Efficient, Decentralized Algorithm for Multiagent RL. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*. https://doi.org/10.48550/arXiv.2110.14555

[23] Bobby Kleinberg, Renato Paes Leme, Jon Schneider, and Yifeng Teng. 2023. U-Calibration: Forecasting for an Unknown Agent. In *Proceedings of Thirty Sixth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 195)*, Gergely Neu and Lorenzo Rosasco (Eds.). PMLR, 5143–5145. https://doi.org/10.48550/arXiv.2307.00168

[24] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. 2020. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming* (2020). https://doi.org/10.1007/s10107-018-1336-7

[25] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press. https://doi.org/10.1017/9781108571401

[26] Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. 2022. Strategizing against Learners in Bayesian Games. In *Proceedings of Thirty Fifth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 178)*, Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 5221–5252. https://doi.org/10.48550/arXiv.2205.08562

[27] Binghui Peng and Aviad Rubinstein. 2023. Fast swap regret minimization and applications to approximate correlated equilibria. https://doi.org/10.48550/arXiv.2310.19647

[28] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2016. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295* (2016). https://doi.org/10.48550/arXiv.1610.03295

[29] Gilles Stoltz and Gábor Lugosi. 2005. Internal regret in on-line portfolio selection. *Machine Learning* 59, 1-2 (2005), 125–159. https://doi.org/10.1007/s10994-005-0465-4

[30] Bernhard Von Stengel and Françoise Forges. 2008. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research* 33, 4 (2008), 1002–1022. https://doi.org/10.1287/moor.1080.0340

[31] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. 2020. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332* (2020). https://doi.org/10.48550/arXiv.2004.13332

[32] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. Regret Minimization in Games with Incomplete Information. In *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), Vol. 20. Curran Associates, Inc. https://doi.org/10.5555/2981562.2981779

Received 13-NOV-2023; accepted 2024-02-11