# Equilibrium Separations with No-Regret Agents

William Brown* Jon Schneider† Kiran Vodrahalli‡

February 12, 2023

**Abstract**

We consider a number of questions related to repeated gameplay between two agents, possibly using no-(swap)-regret algorithms, and introduce a notion of *generalized equilibrium* with asymmetric regret constraints which we use to analyze tradeoffs between the choices of learning algorithms and the rewards obtained by each player. We first show that there exists a wide class of games where a player can significantly increase their utility by switching from playing a no-swap-regret algorithm to their Stackelberg strategy (in fact, almost any game without pure Nash equilibria is of this form). We then show that there exist pairs of algorithms which converge to *any* desired generalized equilibrium while simultaneously maintaining regret guarantees against arbitrary opponents, which yields a tight characterization for the maximal reward obtainable against *some* no-regret learner, yet we also show that there are games in which this value is unobtainable against a large class of standard no-regret algorithms. Finally, we consider the question of identifying the Stackelberg strategy while playing against a no-regret agent when the game is initially unknown, and again show a number of separations depending on the agent's learning algorithm: the Stackelberg strategy is learnable in exponential time via repeated play with any no-regret agent (and in polynomial time with any no-*dynamic*-regret agent) for any game where it is learnable via queries, and there are games where this strategy is learnable in polynomial time against any no-swap-regret agent but requires exponential time against an arbitrary no-regret agent.

## 1 Introduction

How should two rational agents play a repeated, possibly unknown, game against one another? One natural answer – barring any knowledge of the game, or the capacity to compute potentially computationally intractible equilibria – is that they should employ some sort of learning algorithm to learn how to play over time. Indeed, there is a vast literature which studies what happens when all the players in a repeated game run (some specific type of) learning algorithms to select their actions. For example, when all players in a game simultaneously run no-(swap)-regret learning algorithms, it is known that the average strategy profile of the learners converge to a (coarse) correlated equilibrium moulin1978strategically, foster1998asymptotic, hart2000simple. More recent works have studied how to design algorithms that converge to these equilibria at faster rates daskalakis2011near, anagnostides2022near, daskalakis2021near, performance guarantees of such equilibria compared to the optimal possible welfare blum2008regret, hartline2015no, and the specific dynamics of such learning algorithms daskalakis2010learning, ligett2011beating, mertikopoulos2018cycles.

In contrast, relatively little attention has been devoted to whether it is actually *in the interest of these agents* to run these specific learning algorithms. For example, in a setting where all agents are running no-swap-regret learning algorithms, when can an agent significantly benefit by deviating and running a different type of algorithm? And if they can, what algorithm should the agent deviate to?

### 1.1 Our results

We explore the following questions (and others) in the case where two agents repeatedly play a normal-form, general-sum game for $T$ rounds.

*w.brown@columbia.edu
†jschnei@google.com
‡kirannv@google.com

**When is no-swap-regret play a stable equilibrium?** What should you do if you know that your opponent in a repeated game is running a no-swap-regret algorithm to select their actions? In [**?**], the authors show that one utility-optimizing response (up to additive $o(T)$ factors) is to play a static (mixed) strategy (your *Stackelberg strategy*) and obtain the *Stackelberg value* of the game.

However (as we discuss in further detail below), determining your Stackelberg strategy requires some knowledge of the game, and acquiring this knowledge from repeated play may be difficult. In comparison, it is relatively straightforward to also run a no-swap-regret learning algorithm. This begs the question: are there games where you obtain significantly less utility (i.e., at least $\Omega(T)$ less utility) by running a no-swap-regret learning algorithm instead of playing your Stackelberg strategy?

We show that the answer is *yes*, such games exist and are relatively common. In fact, we provide an efficient algorithmic characterization of the games $G$ for which both players playing a no-swap-regret learning algorithm is an ($o(T)$-)approximate Nash equilibrium of the entire repeated game. The exact characterization is presented in Section 3 and is somewhat subtle (e.g., there are slightly different characterizations depending on whether we insist all pairs of no-swap-regret algorithms lead to approximate equilibria or only one specific pair). One consequence of both characterizations, however, is that in order for it to be an approximate equilibrium for both players to play a low-swap regret strategy, the game $G$ must possess a *pure* Nash equilibrium. That is, in any game without a pure Nash equilibrium, it is possible for at least one of the parties to do significantly better by switching from no-swap-regret learning to playing their Stackelberg strategy.

**Playing against no-regret learners.** What if our opponent is not running a no-swap-regret algorithm, but simply a no-(external)-regret algorithm? In this case, it is still possible to obtain at least the Stackelberg value of the game by playing our Stackelberg strategy (no-regret algorithms are also guaranteed to eventually learn and play the best response to this strategy). However, unlike in the no-swap-regret setting, there exist specific games and no-regret algorithms where it is possible to obtain *significantly* ($\Omega(T)$) more than the Stackelberg value by playing a specific dynamic strategy.

This phenomenon was first observed in [**?**], where the authors gave an example of a specific game where it is possible to obtain $\Omega(T)$ more than Stackelberg when playing against any no-regret algorithm in the family of *mean-based learning algorithms* (including algorithms such as multiplicative weights and FTRL). However, many basic questions remain unanswered – for example, understanding in which games it is possible to outperform playing one's Stackelberg strategy, and by how much.

We present some answers to these questions for the case of generic no-regret algorithms. Specifically, we first show that if player $B$ is running (any) no-regret algorithm, the utility of player $A$ (regardless of what strategy they employ) is upper bounded by $\mathsf{Val}_A(\emptyset, \mathcal{E}) \cdot T + o(T)$, where $\mathsf{Val}_A(\emptyset, \mathcal{E})$ is what we call the the *unconstrained-external (equilibrium) value* of the game for player $A$, which is given by the solution to a linear program. We then show that this upper bound is asymptotically tight: there exists a no-regret algorithm $\mathcal{L}$ such that if player $B$ is playing according to $\mathcal{L}$, then the player $A$ can obtain utility at least $\mathsf{Val}_A(\emptyset, \mathcal{E}) \cdot T - o(T)$ by playing an appropriate strategy in response.

Note that this characterization does not completely resolve the question of [**?**] – it requires the construction of a fairly specific no-regret algorithm $\mathcal{A}$, and it is still open what is possible against specific (classes of) no-regret algorithms (e.g., multiplicative weights or mean-based algorithms). In fact, we show that there are games where it is impossible to obtain the unconstrained-external value against a mean-based learner (Theorem 4).

**Learning the Stackelberg strategy through repeated play.** Finally, we address the question of how hard is it to actually play the Stackelberg strategy in a game (e.g., in a setting where an agent is deviating from playing no-swap-regret). Given full knowledge of the game $G$, finding an agent's Stackelberg strategy is simply a computational problem which turns out to be efficiently solvable by solving several small LPs (see [**?**]). However, in a setting where the base game is unknown, the agent must learn their Stackelberg strategy over time.

There is some existing work on the problem of learning the Stackelberg strategy in a game (e.g., [**?**, **?**]), but this work generally assumes access to a best-response oracle for the game (i.e., for a specific mixed strategy, what is the best-response of our opponent and what utility do we obtain under this joint strategy profile?). In contrast, if our opponent is playing a specific no-regret learning algorithm, they may not immediately

best respond to the strategies we play! This raises the following two questions. First, when is it possible to learn the Stackelberg equilibrium of a game while playing against a learning opponent? Second, is it easier to learn this equilibrium when playing against certain classes of learning algorithms?

We begin by showing that it is indeed possible to convert any algorithm that learns the Stackelberg equilibrium via (approximate) best-response queries to an algorithm that can learn the Stackelberg equilibrium while playing against a generic no-regret learner, albeit at the cost of an exponential blow-up in the number of rounds (if the original algorithm required $Q$ best-response queries, the second algorithm requires $T = \exp(O(Q))$ rounds of play). We then show that, in some sense, this exponential blow-up is necessary. In particular, we give an example of a game with $M$ actions where it is possible to learn the Stackelberg equilibrium in $\text{poly}(M)$ rounds when playing against any no-swap-regret learning algorithm, but where it requires at least $\exp(\Omega(M))$ rounds to learn this equilibrium when playing against certain no-regret algorithms.

**Generalized equilibria (and separations).** In developing some of the above results, we find it useful to work with a generalization of correlated and coarse correlated equilibria that we call *generalized equilibria*. At a high level, given a class $\mathcal{F}_A$ of "deviation rules" (time independent rules for modifying an agent's strategy) for the first agent, and a class $\mathcal{F}_B$ of "deviation rules" for the second agent, we define an $(\mathcal{F}_A, \mathcal{F}_B)$-equilibrium to be a joint distribution over strategy profiles where neither player can increase their reward by applying one of the deviations in their class. These classes directly capture common regret notions: for example, when $\mathcal{F}_A$ is the class $\mathcal{E}$ of fixed action deviations, this corresponds to the first agent having a no-(external)-regret guarantee, and thus $(\mathcal{E}, \mathcal{E})$-equilibria are simply coarse correlated equilibria. Where the flexibility of this definition is useful is the ability to specify different classes for $\mathcal{F}_A$ and $\mathcal{F}_B$; for example, if we set $\mathcal{F}_A = \emptyset$ (no valid deviations except the identity) and $\mathcal{F}_B = \mathcal{E}$, then $(\emptyset, \mathcal{E})$-equilibria are exactly the unconstrained-external equilibria mentioned above.

Some of the above results extend to these generalized equilibria (for example, we prove an analogue of the above result on inducing the best unconstrained-external equilibrium for any generalized equilibrium notion). In addition, we characterize the maximal reward gaps between generalized equilibria for many of the explicit variants we consider (Theorem 8).

### 1.2 Related Work

The broader literature on no-regret learning in repeated games is substantial, covering many equilibrium convergence results varying assumptions. A recent line of work ([**?**, **?**, **?**]) considers problems related to optimizing one's reward when competing against a no-regret learner in a game. We extend these questions to consider the relationship and regret for an optimizer, as well as to settings where properties of the game are initially unknown, and give a series of separation results in terms of various notions of equilibrium. Also relevant is the literature on analysis of no-regret trajectory dynamics, and in particular [**?**] which shows a game in which no-regret dynamics outperform the reward of the Nash equilibrium. Additionally, there is also prior work considering regret minimization problems involving either best-responding or otherwise strategic agents (see e.g. [**?**, **?**], as well as work considering alternate notions of regret for repeated Stackelberg games (e.g. [**?**]).

## 2 Preliminaries

Throughout, we consider two-player bimatrix games $G = (A, B)$, where player $A$ ("the optimizer") has action set $\mathcal{A} = \{a_1, \ldots, a_M\}$ and player $B$ ("the learner") has action set $\mathcal{B} = \{b_1, \ldots, b_N\}$. When the optimizer plays action $a_i$ and the learner plays action $b_j$, the players receive rewards $u_A(a_i, b_j)$ and $u_B(a_i, b_j)$, respectively. We assume that the magnitude of each utility is bounded by a constant. The sets of mixed strategies for each player are denoted by $\Delta(\mathcal{A})$ and $\Delta(B)$, respectively; when the optimizer plays a mixed strategy $\alpha \in \Delta(\mathcal{A})$ and the learner plays $\beta \in \Delta(\mathcal{B})$, the expected reward for the optimizer is given by $u_A(\alpha, \beta) = \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_i \beta_j u_A(a_i, b_j)$, with $u_B(\alpha, \beta)$ defined analogously. An action $b \in \mathcal{B}$ is a *best response* to a strategy $\alpha \in \Delta(\mathcal{A})$ if $b \in \text{argmax}_{b' \in \mathcal{B}} u_B(\alpha, b')$. Let $\mathsf{BR}(\alpha)$ be the set of all such actions.

**Definition 1** (Stackelberg Equilibria)**.** *The* Stackelberg equilibrium *of a game $G$ is the pair of strategies $(\alpha, b)$ given by $\text{argmax}_{\alpha \in \Delta(\mathcal{A}),\ b \in \mathsf{BR}(\alpha)} u_A(\alpha, b)$, and the resulting expected utility for the optimizer is the*

Stackelberg value *of the game, denoted* $\mathsf{Stack}_A$.

Throughout, we assume that there is an (approximate) Stackelberg equilibrium with a unique best response; this is implied by several natural assumptions, such as that the best response regions have positive volume (see Section 5.1), or when the game has no weakly dominated strategies (see [**?**]).

Given a sequence of action pairs $(a_{i_1}b_{j_1}), \ldots, (a_{i_T}b_{j_T})$ for $T > 0$ and some set of functions $\mathcal{F}$, where each $f \in \mathcal{F}$ maps actions $\mathcal{A}$ to action profiles in $\Delta(\mathcal{A})$, we say that the *$\mathcal{F}$-regret* for the optimizer (and analogously for the learner) is

$$\mathsf{Reg}_{\mathcal{F}}(T) = \max_{f \in \mathcal{F}} \sum_{t=1}^{T} u_A(f(a_{i_t}), b_{j_t}) - u_A(a_{i_t}, b_{j_t}).$$

**Definition 2** (No-$\mathcal{F}$-Regret Learning)**.** *We say a learning algorithm $\mathcal{L}$ is a no-$\mathcal{F}$-regret algorithm if, for some constant $c < 1$, we have that $\mathsf{Reg}_{\mathcal{F}}(\mathcal{L}, T) = O(T^c)$, where $\mathsf{Reg}_{\mathcal{F}}(\mathcal{L}, T)$ is the $\mathcal{F}$-regret corresponding to the action sequence played by $\mathcal{L}$.*

Some notable sets of regret comparator functions $\mathcal{F}$ are the constant maps $\mathcal{E}$ (corresponding to *external regret*), where all input actions are mapped to the same output action, and the "swap functions" $\mathcal{I}$ (corresponding to *internal regret*), which contain all single swap maps $f_{ij} : [M] \to [M]$ where $f(i) = j$ and $f(i') = i'$ for $i' \neq i$. Imposing these constraints on players in a game results in a *(coarse) correlated equilibrium*, which are instances of our notion of *generalized equilibrium*.

**Definition 3** (Generalized Equilibria)**.** *A $(\mathcal{F}_A, \mathcal{F}_B)$-equilibrium $\sigma \in \Delta(\mathcal{A} \times \mathcal{B})$ in a two-player game is a joint distribution over action profiles $(a, b)$ such that player $A$ cannot increase their expected reward by deviating with some strategy in $\mathcal{F}_A$ and player $B$ cannot benefit by deviating with some strategy in $\mathcal{F}_B$.*

We consider an $\varepsilon$-approximate generalized equilibrium to be one where each constraint is satisfied up to additive error $\varepsilon$, and we say that the *value* of a game $G$ for player $A$ of a certain equilibrium class $(\mathcal{F}_A, \mathcal{F}_B)$, denoted $\mathsf{Val}_A(\mathcal{F}_A, \mathcal{F}_B)$ is the maximum reward obtainable by player $A$ at some $(\mathcal{F}_A, \mathcal{F}_B)$-equilibrium (with $\mathsf{Val}_B(\mathcal{F}_A, \mathcal{F}_B)$ defined analogously for player $B$).

While many equilibrium notions for two-player games impose symmetric regret constraints on each player (e.g. Nash, correlated, and coarse correlated equilibria), this need not always be the case (Stackelberg equilibria are a motivating example for considering more general notions of asymmetric equilibria). In particular, we focus the function classes $\mathcal{E}$ and $\mathcal{I}$ corresponding to external and internal regret as mentioned above, as the well empty set $\emptyset$ corresponding to unconstrained regret. For example, any joint distribution over action profiles where player $B$ has zero swap regret constitutes a $(\emptyset, \mathcal{I})$-equilibrium for a game, and the optimal value for such an equilibrium for player $A$ coincides with the Stackelberg value. Further, these sets of equilibria characterize the possible outcome spaces for pairs of learning algorithms, and each can be optimized over via a linear program.

For a repeated game over $T$ rounds where player $A$ uses a no-$\mathcal{F}_A$-regret algorithm and player $B$ uses a no-$\mathcal{F}_B$-regret algorithm, the average rewards obtained by each player are upper bounded by $\mathsf{Val}_A(\mathcal{F}_A, \mathcal{F}_B) + o(1)$ and $\mathsf{Val}_B(\mathcal{F}_A, \mathcal{F}_B) + o(1)$, respectively.

For a game $G$, if there is an approximate Stackelberg equilibrium $(\alpha, b)$ such that $u_A(\alpha, b) = \mathsf{Stack}_A - \varepsilon$ where $\mathsf{BR}(\alpha) = \{b\}$ for any $\varepsilon > 0$, then $\mathsf{Stack}_A = \mathsf{Val}_A(\emptyset, \mathcal{I})$.

For any game $G$ and constraints $(\mathcal{F}_A, \mathcal{F}_B)$, both $\mathsf{Val}_A(\mathcal{F}_A, \mathcal{F}_B)$ and $\mathsf{Val}_B(\mathcal{F}_A, \mathcal{F}_B)$ are computable via linear programs with $MN$ variables and $\mathrm{poly}(M, N, |\mathcal{F}_A|, |\mathcal{F}_B|)$ constraints. When $\mathcal{F}_A$ and $\mathcal{F}_B$ belong to $\{\emptyset, \mathcal{E}, \mathcal{I}\}$, the number of constraints is $\mathrm{poly}(M, N)$.

In general, these values for a player can differ when considering distinct notions of generalized equilibria. In Appendix A we give several examples of such separations, and discuss further characterizations of generalized equilibria and their corresponding values.

**Anytime regret-bounds and mean-based learning.** For some of our results, it will be most natural for us to assume that algorithms have *anytime* regret bounds, i.e. that regret over the first $t$ rounds is bounded by $O(t^c)$ for any $t \leq T$. Many standard no-regret algorithms obtain bounds of this form by appropriately decaying their learning rates or applying doubling methods. Another key property of many no-regret algorithms, identified by [**?**], is that they are *mean-based*.

**Definition 4** (Mean-Based Learning)**.** *Let $\sigma_{i,t}$ be the cumulative reward resulting from playing action $i$ for the first $t$ rounds. An algorithm $\mathcal{L}$ is $\gamma$-mean-based if, whenever $\sigma_{i,t} \leq \sigma_{j,t} - \gamma T$, the probability that the algorithm selects action $i$ in round $t+1$ is at most $\gamma$, for some $\gamma = o(1)$.*

# 3 Stability of no-swap-regret play

We begin by addressing the following question: when is it the case that for two players in a game, it is an approximate (Nash) equilibrium for both players to play no-swap-regret strategies? More specifically, imagine a "metagame" where at the beginning of this repeated game, both players simultaneously announce and commit to a specific adaptive (and possibly randomized) algorithm they intend to run to select actions to play in the repeated game $G$ for the next $T$ rounds. In this metagame, for which games $G$ is it an $o(T)$-approximate Nash equilibrium for both players to play a no-swap-regret learning algorithm?

Of course, the answer to this question might depend on *which* specific no-swap-regret learning algorithm the agents declare. We therefore attempt to understand the following two questions:

- **Necessity:** For which games $G$ is it the case that there exists *some* pair of no-swap-regret algorithms which form a $o(T)$-approximate Nash equilibrium? (Equivalently, when is it *never* the case that playing no-swap-regret algorithms forms an approximate Nash equilibrium).
- **Sufficiency:** For which games $G$ is it the case that *all* pairs of no-swap regret algorithms form $o(T)$-approximate Nash equilibria?

We begin by providing an efficient algorithmic procedure to answer both of these questions for a specific game $G$. To do this, recall that when two players both employ no-swap regret strategies, they asymptotically (time-average) converge to some correlated equilibrium. On the other hand, by defecting from playing a no-swap regret strategy (while the other player continues playing their no-swap regret strategy), a player can guarantee their Stackelberg value for the game. Moreover, as shown by [**?**], this is the *optimal* (up to $o(T)$ additive factors) best response to an opponent running a no-swap regret strategy.

It thus suffices to understand how the utility a player might receive under a correlated equilibrium compares to the utility they receive under their Stackelberg strategy. For a fixed game $G$, let $\mathsf{Stack}_A = \mathsf{Val}_A(\emptyset, \mathcal{I})$ be the Stackelberg value for the first player, $\mathsf{Stack}_B$ be the Stackelberg value for the second player. We have the following theorem.

**Theorem 1.** *Fix a game $G$. The following two statements hold:*

1. *There exists some pair of no-swap-regret algorithms that form an $o(T)$-approximate Nash equilibrium in the metagame iff there exists a correlated equilibrium $\sigma$ in $G$ such that $u_A(\sigma) = \mathsf{Stack}_A$ and $u_B(\sigma) = \mathsf{Stack}_B$.*
2. *Any pair of no-swap-regret algorithms form an $o(T)$-approximate Nash equilibrium in the metagame iff for all correlated equilibria $\sigma$ in $G$, $u_A(\sigma) = \mathsf{Stack}_A$ and $u_B(\sigma) = \mathsf{Stack}_B$.*

*Moreover, given a game $G$, it is possible to efficiently (in polynomial time in the size of $G$) check whether each of the above cases holds.*

The characterization in Theorem 1 is algorithmically useful, but sheds little direct light on in which games or how often we would expect playing no-swap-regret to be an approximate equilibrium. It turns out that for many games, playing no-swap-regret is *not* an equilibrium; below we will show that for almost all games, if $G$ does not have a pure Nash equilibrium, at least one player has an incentive to deviate to their Stackelberg strategy.

**Definition 5.** *A property $P$ of a game holds for* almost all *games if, given any game $G$, property $P$ holds with probability* 1 *for the game $G'$ formed by starting with $G$ and perturbing each of the entries $u_A(a_i, b_j)$ and $u_B(a_i, b_j)$ by independent uniform random variables in the range $[-\varepsilon, \varepsilon]$ (for any choice of $\varepsilon$). In other words, the property holds for almost all choices of the $2MN$ utility values that define a game (with respect to the standard measure on this space).*

**Theorem 2.** *For almost all games $G$, if $G$ does not have a pure Nash equilibrium, then there does not exist a pair of no-swap-regret algorithms which form a $o(T)$-approximate Nash equilibrium in the metagame for $G$.*

*Sketch.* We can show that if a correlated equilibrium has the same utility for a player as their Stackelberg value (a consequence of Theorem 1), then the correlated equilibrium must be a convex combination of valid Stackelberg equilibria. In almost all games, both players have unique Stackelberg equilibria, which implies that this correlated equilibrium must actually be the Stackelberg strategy for both players simultaneously. This implies that it is a pure Nash equilibrium (since one action in a generic Stackelberg equilibrium is always pure). □

Note that although Theorem 2 holds for almost all games, there are some important classes of games (most notably, zero-sum games) in the measure zero subset omitted by this theorem statement that both a. do not have pure Nash equilibria and b. have the property that playing no-swap-regret algorithms is an approximate equilibrium in the metagame (in particular, for zero-sum games, the Stackelberg value collapses to the value of the unique Nash equilibrium). Still, Theorem 2 shows that there are very wide classes of games for which playing no-swap-regret algorithms is not stable from the perspective of the agents.

# 4 Generalized Equilibria and No-Regret Learning

In this section, we assume the game is known by both players. Our goal is to establish whether or not it is possible to achieve a specific generalized equilibrium with no-$\mathcal{F}$-regret learners – of particular interest is the *optimal* such equilibrium. We prove that it is possible to "target" specified generalized equilibra (thus including various notions of optimal equilibrium) while maintaining no-$\mathcal{F}$-regret properties when given leeway to design the learning algorithms that both players use. However, even in very natural settings where one player is playing a mean-based no-regret algorithm, there exist games where a specific $(\emptyset, \mathcal{E})$-equilibrium cannot be reached regardless of the learning algorithm the second player uses.

Before we proceed, we formally state the regret-equilibrium connection between Definitions 2 and 3 in the following proposition: [Convergence of No-$\mathcal{F}$-Regret Dynamics to Generalized Equilibrium] Suppose after $T$ iterations of dynamics where player $A$ plays a no-$\mathcal{F}_A$-regret algorithm and player $B$ plays a no-$\mathcal{F}_B$-regret algorithm, player $A$ has average $\mathcal{F}_A$-regret $\leq \varepsilon$ and player $B$ has average $\mathcal{F}_B$-regret $\leq \varepsilon$. Let $\sigma^t := p_A^t \times p_B^t$ denote the joint distribution over both players' actions at time $t$ and $\sigma := \frac{1}{T}\sum_{t=1}^{T} \sigma^t$ denote the time-averaged history over joint player action distributions. Then we have that $\sigma$ is an $\varepsilon$-approximate $(\mathcal{F}_A, \mathcal{F}_B)$-equilibrium:

$$\mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_A\left(a,b\right)\right] \geq \mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_A\left(f_A(a),b\right)\right] - \varepsilon$$
$$\mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_B\left(a,b\right)\right] \geq \mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_B\left(a,f_B(b)\right)\right] - \varepsilon$$

for every possible deviation $f_A \in \mathcal{F}_A, f_B \in \mathcal{F}_B$. Likewise, if players $A$ and $B$ repeatedly play strategies corresponding to an $(\mathcal{F}_A, \mathcal{F}_B)$-equilibrium, then player $A$ is no-$\mathcal{F}_A$-regret and player $B$ is no-$\mathcal{F}_B$-regret.

We show that for any generalized equilibrium $E$ in a game, there exists a pair of algorithms which satisfy a "best-of-both-worlds" property: when played together, they converge to $E$, yet they simultaneously maintain the corresponding regret guarantees when played against arbitrary adversaries.

**Theorem 3.** *Consider any game $G$. Suppose there exists a no-$\mathcal{F}_A$-regret learning algorithm $\mathcal{L}_A$ and a no-$\mathcal{F}_B$-regret learning algorithm $\mathcal{L}_B$. For any particular $(\mathcal{F}_A, \mathcal{F}_B)$-equilibrium $E$ in a game $G$, there exists a pair of learning algorithms $(\mathcal{L}_A^*(E), \mathcal{L}_B^*(E))$ such that:*

- *The empirical sequence of play when Player A uses $\mathcal{L}_A^*(E)$ and Player B uses $\mathcal{L}_B^*(E)$ converges to $E$.*
- *$\mathcal{L}_A^*(E)$ and $\mathcal{L}_B^*(E)$ are no-$\mathcal{F}_A$-regret and no-$\mathcal{F}_B$-regret, respectively, against arbitrary adversaries.*

Our approach is for the algorithms to initially implement a schedule of strategies which converges to $E$. Yet, these algorithms also detect when their opponent disobeys the schedule by tracking their $\mathcal{F}$-regret with respect to $E$, and after $o(T)$ violations can deviate indefinitely to playing a standalone no-$\mathcal{F}$-algorithm for all remaining rounds. This result has some notable implications:

**Corollary 3.1.** *For any equilibrium scoring function $\phi : \Delta(\mathcal{A} \times \mathcal{B}) \to \mathbb{R}$ with a unique optimum computable in finite time, there exists a pair of learning algorithms $(\mathcal{L}_A^*, \mathcal{L}_B^*)$ such that:*

- *The empirical distribution when player $A$ uses $\mathcal{L}_A^*$ and player $B$ uses $\mathcal{L}_B^*$ converges to* $\operatorname{argmax}_E \phi(E)$.
- *$\mathcal{L}_A^*$ and $\mathcal{L}_B^*$ are no-$\mathcal{F}_A$-regret and no-$\mathcal{F}_B$-regret, respectively, against arbitrary adversaries.*

*Proof.* First optimize $\phi$ over $E$ in finite time to find the unique optimum; then apply Theorem 3 to the resulting desired equilibrium. □

Corollary 3.1 allows for optimizing for objectives such as total welfare or min-max utility for both players, and also for imposing further conditions on generalized equilibria beyond $\mathcal{F}$-regret constraints (e.g. product distribution constraints for Nash equilibria) by assigning arbitrarily low scores to invalid strategy profiles.

Furthermore, this result implies that $\mathsf{Val}_{(\emptyset,\mathcal{E})}(G)$ tightly characterizes the maximum reward attainable against a no-regret learner, i.e. for each game $G$, there exists a no-regret algorithm $\mathcal{L}_B$ for the learner and an efficiently implementable algorithm $\mathcal{L}_A$ for the optimizer (which may require positive regret) which, when played together, yield the maximum reward attainable with an adaptive strategy against a no-regret learner for the optimizer.

**Corollary 3.2.** *For any game $G$, there exists a no-regret algorithm $\mathcal{L}$ and a strategy for Player $A$ such that the empirical average reward of Player $A$ converges to $\mathsf{Val}_A(\emptyset, \mathcal{E})$ when Player $B$ uses $\mathcal{L}$.*

While this requires the learner to use a specific no-regret algorithm, we show that some such constraint is necessary to obtain $\mathsf{Val}_A(\emptyset, \mathcal{E})$: for the class of *mean-based* no-regret algorithms, there are instances where $\mathsf{Val}_A(\emptyset, \mathcal{E})$ and $\mathsf{Val}_A(\emptyset, \mathcal{I})$ are separated by a constant and where the maximal value obtainable against a mean-based learner is asymptotically equal to $\mathsf{Val}_A(\emptyset, \mathcal{I})$.

**Theorem 4.** *Against any mean-based no-regret algorithm for player $B$, there is a game where a $T$-round reward of $\mathsf{Val}_A(\emptyset, \mathcal{E}) \cdot T - o(T)$ cannot be reached by any adaptive strategy for player $A$.*

| | $b_1$ | $b_2$ | $b_3$ |
|---|---|---|---|
| $a_1$ | 1, 1 | 0, 0 | 3, 0 |
| $a_2$ | 0, 0 | 1, 1 | 0, 0 |

Figure 1: Game where $\mathsf{Val}_A(\emptyset, \mathcal{E})(G) > \mathsf{Val}_{(\emptyset,\mathcal{I})}(G) = \mathsf{MBRew}(G)$

*Proof.* Let $\mathsf{MBRew}_A$ denote the maximal reward obtainable by player $A$ when player $B$ uses a mean-based algorithm. Observe that $b_3$ is dominated for player $B$, and thus cannot be included in any $(\emptyset, \mathcal{I})$-equilibrium. Further, it will never be played by a mean-based learner for more than $o(T)$ rounds, as for any distribution over $a_1$ and $a_2$ the best response is either $b_1$ or $b_2$. As such, both $\mathsf{Val}_A(\emptyset, \mathcal{I})$ and $\mathsf{MBRew}_A$ are at most $1 + o(1)$; a reward of $1 - o(1)$ is obtainable by committing to either $a_1$ or $a_2$ for each round. However, we can see that the optimal $(\emptyset, \mathcal{E})$-equilibrium $p$ for player $A$ includes positive mass on $(a_1, b_3)$, and yields an average reward of $\mathsf{Val}_A(\emptyset, \mathcal{E}) = 2$ for player $A$. Let $p_1$ be the probability on $(a_1, b_1)$, let $p_2$ be the probability on $(a_2, b_2)$, let $p_3$ be the probability on $(a_1, b_3)$, and let $p_0$ be the remaining probability. The reward for player $A$ is given by:

$$\mathsf{Rew}_A(G, p) = p_1 + p_2 + 3p_3$$

and $p$ defines a $(\emptyset, \mathcal{E})$-equilibrium if

$$\mathsf{Rew}_B(G, p) \geq \mathsf{Rew}_B(G, p \to b_i)$$

for each $b_i$, which holds if:

$$\begin{aligned} p_1 + p_2 &\geq p_1 + p_3; \\ p_1 + p_2 &\geq p_2; \\ p_1 + p_2 &\geq 0. \end{aligned}$$

Only the first constraint is non-trivial, and so the optimal $(\emptyset, \mathcal{E})$-equilibrium for player $A$ occurs by maximizing $p_1 + p_2 + 3p_3$ subject to $p_2 \geq p_3$, which yields a probability of 0.5 for both $p_2$ and $p_3$ (and 0 for $p_1$ and $p_0$), as well as an average reward of 2. As such, player $A$ cannot obtain a reward approaching $\mathsf{Val}_A(\emptyset, \mathcal{E})$, as their per-round reward is at most $1 + o(1)$.

□

# 5 Learning Equilibria in Unknown Games

The previous results have considered settings where the optimizer has full information of the payoffs of the game $G$. In many cases, such as when a game is represented implicitly with rewards being returned by interaction with some oracle, this assumption is unrealistic. Here, we consider the question of strategizing against a no-regret learner when the game is initially unknown. In particular, we consider the goal of efficiently matching the Stackelberg value (which is always possible in the full-information case, and is the best conceivable outcome against some algorithms), and we show that the feasibility of this can depend on the details of the specific no-regret algorithm used by the agent. We introduce a method for *robustly* learning Stackelberg strategies via perturbed best response queries, which we extend to give a polynomial-round strategy learning against arbitrary no-dynamic-regret algorithms, as well an exponential-round strategy for arbitrary no-regret algorithms. We also give an example set of games for which learning the Stackelberg strategy against a class of *mean-based* algorithms requires exponential time, but where learning against an arbitrary no-swap-regret algorithm can be done in polynomial time.

## 5.1 Robust Query Algorithms

First, we consider the problem of *robustly* learning the Stackelberg strategy for a game via best response queries in which the input mixed strategy for any query may be perturbed in an arbitrary by some small amount. Algorithms for computing Stackelberg equilibria via queries typically assume the ability to specify a mixed strategy to arbitrary precision and observe the opponent's best response, as well one's reward [**?**, **?**], and aim to find the exact Stackelberg equilibrium. Instead, we will be interested in matching $\mathsf{Stack}_A$ to within $\text{poly}(\varepsilon)$, while tolerating a perturbation error of $\varepsilon$ to our queries. Given a strategy $\alpha$, an *$\varepsilon$-perturbed best response query* to $\alpha$ returns some action $b \in \mathsf{BR}(\alpha + p)$, where $p \in \mathbb{R}^M$ is an arbitrary vector satisfying $\|p\|_1 \leq \varepsilon$ and $\alpha + p \in \Delta(\mathcal{A})$. For a class of games $\mathcal{G}$, an algorithm $\mathcal{Q}_\varepsilon$ is an *$\varepsilon$-robust query algorithm* for learning an approximate Stackelberg equilibrium if, over a sequence of at most $Q$ $\varepsilon$-perturbed best response queries, it outputs a strategy $\alpha^*$ with *robust reward* $u_A(\alpha^* + p, b) \geq \mathsf{Stack}_A - O(\varepsilon^c)$ for some $c > 0$ and any $b \in \mathsf{BR}(\alpha^* + p)$, again for any $p \in \mathbb{R}^M$ satisfying $\|p\|_1 \leq \varepsilon$ and $\alpha + p \in \Delta(\mathcal{A})$.

The query complexity of existing algorithms often depends on structural properties of the game such as bit representation length, the number of extremal points of best response regions, and the size of the smallest best response region. This last condition poses inherent difficulties for finding initial points inside each *best response region* $R_j = \{\alpha \in \Delta(\mathcal{A}) : b_j = \mathsf{BR}(\alpha)\}$ in the robust setting: if a region $R_j$ does not contain a ball of radius $\varepsilon$, then adversarial perturbations may prevent *all* queries from observing $b_j$ as a best response. Here, we assume that regions are sufficiently large to preclude this, and further that an initial such point is known for each region.[1] Our robust query algorithm builds on the algorithm from [**?**], calibrated to precision $O(\log(1/\varepsilon))$, with an added contraction step to ensure that points in the estimated best response regions $\hat{R}_j$ are contained in the corresponding regions $R_j$ even after applying perturbations.

**Theorem 5.** *Suppose that for a game $G$, for each $b_j$ where $\mathsf{BR}(\alpha) = \{b_j\}$ for some $\alpha \in \Delta(\mathcal{A})$, we are given a strategy $\alpha^{(j)}$ such that $\mathsf{BR}(\alpha^{(j)} + p) = \{b_j\}$ for any $p$ with $\|p\|_1 \leq \varepsilon^{1/2}$ . Then, there is an $\varepsilon$-robust query algorithm which requires $Q = \text{poly}(M, N, \log(1/\varepsilon))$ queries, and which yields a strategy $\alpha^*$ with robust reward at least $\mathsf{Stack}_A - O(\varepsilon^{1/2})$.*

*Proof.* Note that when imposing an $\varepsilon$-grid line segment between any two points in (convex) feasible regions $R_j$ and $R_{j'}$, at most one point on this grid can result in multiple possible best responses due to perturbation. From this, we can run the SU algorithm from [**?**] starting from each $\alpha^{(j)}$, specified for $\log(1/\varepsilon)$ bits of

[1] The assumption that an initial point is known can be relaxed via sampling if each region $R_j$ has volume at least $\Omega(\text{poly}(\varepsilon))$, using the sampling approach from [**?**].

precision, with the small modification that the hyperplane boundary points for an estimated best response region $\hat{R}_j$ are chosen to be one step inward along the $\varepsilon$-grid used for binary searching line segments. Each such point necessarily belongs to $R_j$, and the set of linear programs for each $\hat{R}_j$ will output the optimal Stackelberg strategy associated with these contracted best response regions, up to precision $\log(1/\varepsilon)$.

To see that this impacts utility by at most $O(\varepsilon^{1/2})$, note that each $R_j$ contains a ball of radius $\varepsilon^{1/2}$ (here denoted $\varepsilon^{1/2}\,\mathbb{B}$). Let $R_j$ be parameterized such that $\alpha^{(j)}$ is the origin. Consider any halfspace defined by a hyperplane of $R_j$; any point the hyperplane's distance to the origin is decreased by a factor of at most $1-\varepsilon^{1/2}$, as the absolute distance along the normal vector is at least $\varepsilon^{1/2}$ and is decreased by $O(\varepsilon)$ when considering the parallel hyperplane to that of $R_j$; the estimated boundary of $\hat{R}_j$ lies between these hyperplanes. This is maintained upon taking the intersection of the halfspaces which define $\hat{R}_j$, and so we have that any point in $(1-O(\varepsilon^{1/2}))R_j$ is contained in $\hat{R}_j$. Further, we have that

$$(1-O(\varepsilon^{1/2}))\hat{R}_j + O(\varepsilon^{1/2})^2\,\mathbb{B} \subseteq (1-O(\varepsilon^{1/2}))\hat{R}_j + O(\varepsilon^{1/2})\hat{R}_j \subseteq R_j,$$

and so any point in $(1-O(\varepsilon^{1/2}))\hat{R}_j$ has a ball of radius $\varepsilon$ around it contained in $R_j$. As such, optimizing over $(1-O(\varepsilon^{1/2}))\hat{R}_j$ yields a strategy within $O(\varepsilon^{1/2})$, as rewards are bounded by a constant, and so upon taking the maximum over each $\hat{R}_j$ we obtain a reward at least $\mathsf{Stack}_A - O(\varepsilon^{1/2})$. □

## 5.2 From Robust Query Algorithms to Adaptive Strategies

We give a set of reductions from robust query algorithms for learning Stackelberg equilibria to *adaptive strategies* for playing no-regret or no-*dynamic*-regret learners. The notion of dynamic regret, which was originally formulated in [**?**] and extended to the online setting in [**?**], considers regret with respect to comparator sequences which may change over time. Here, we make use of a simplified notion which considers only comparator sequences which change once every $W$ rounds, for $W = o(T)$; note that this is implied for sufficiently large $W$ by the standard path-length formulation.

**Definition 6** (Dynamic Regret Learning)**.** *Consider the partition of a length-$T$ horizon into consecutive windows of length $W$. An algorithm is no-dynamic-regret for $W$ and $\mathcal{F}$ if, for some constant $c < 1$, we have that $\mathsf{Reg}_{\mathcal{F}}(\mathcal{L}, W) = O(W^c)$ for each window in the partition.*

Intuitively, the property that no-regret algorithms must frequently play the best response to the entire history of losses (or to a recent window of losses in the case of no-dynamic-regret algorithms) enables us to simulate queries by playing actions which align the relevant history with the strategy we intend to query. For arbitrary anytime no-regret algorithms, this approach may take exponentially many rounds, as subsequent queries may require "washing out" nearly the entire history of play, whereas the per-window regret bound of no-dynamic-regret algorithms allows query simulation in a polynomial number of rounds.

For simplicity, we assume that players know their own reward matrix but not their opponent's; this is straightforward to relax for the optimizer when the learner is no-regret, as the relevant history changes slowly enough inside each $R_j$ to enable playing each strategy $a_i$ and observing $u_A(a_i, b_j)$.

**Theorem 6.** *Given a robust query algorithm $\mathcal{Q}_\varepsilon$ for a class of games $\mathcal{G}$ and a learner using an anytime no-regret algorithm $\mathcal{L}$, there is an adaptive strategy for the optimizer which achieves an average reward at least $\mathsf{Stack}_A - O(\varepsilon^{1/2} + T^{c-1})$ after $T = O(\mathrm{poly}(1/\varepsilon)^Q)$ rounds. If $\mathcal{L}$ is a no-dynamic-regret algorithm for windows of size $W = \Omega(\mathrm{poly}(1/\varepsilon))$, there is an adaptive strategy which achieves an average reward at least $\mathsf{Stack}_A - O(\varepsilon^{1/2} + T^{c-1})$ after $T = O(WQ/\varepsilon^{1/2})$ rounds.*

*Proof.* First, consider an anytime no-regret learner with an $O(t^c)$ regret bound. To implement a query $q$, greedily play the action whose historical frequency of play is the furthest below its target frequency in $q$. After $O(\mathrm{poly}(1/\varepsilon))$ rounds, the historical distribution will be within $O(\varepsilon)$ of $q$, and continuing the greedy selection strategy indefinitely will ensure that the history remains in a $O(\varepsilon)$-ball around $q$. Let $t_q$ be the time at which this occurs. After maintaining the greedy strategy for $q$ for an additional $\omega(t_q^c)$ rounds, the anytime regret bound ensures that most frequently played item must indeed be the best response response to some point in the $O(\varepsilon)$-ball around $q$, provided that this ball is contained entirely inside some best response region $R_j$. Recall that our robust query algorithm $\mathcal{Q}_\varepsilon$ does not make use of the response for the point

which may lie on the boundary, and always steps inward by $\varepsilon$ towards $\alpha^{(j)}$, for which the corresponding ball will be entirely contained in $R_j$. For the $k$th subsequent query $q^{(k)}$, playing the greedy strategy will result in convergence of the history to with $O(\varepsilon)$ after $O(\text{poly}(1/\varepsilon)^i)$ rounds, as the entire history prior to implementing $q^{(i)}$ contributes at most $O(\varepsilon)$ of the current history. We repeat this process until all $Q$ queries have been answered, requiring $O(\text{poly}(1/\varepsilon)^Q)$ rounds, after which we can commit to playing the strategy $\alpha^*$ output by $\mathcal{Q}_\varepsilon$ rounds, which dominates the history after $t$ increases factor of $O(1/\varepsilon)$. The difference between our average utility and $\mathsf{Stack}_A$ is bounded by the $O(\varepsilon^{1/2})$ error resulting from $\alpha^*$, as well as the learner's average regret $T^{c-1}$ which bounds the fraction of rounds in which the best response to $\alpha^*$ is not played.

For the dynamic regret case, if $W$ is large enough to implement each query as above (ignoring prior history), we can simply allocate one window per query, requiring a total of $QW$ rounds to learn $\alpha^*$. After increasing $t$ by a factor of $\varepsilon^{-1/2}$, the average error contribution during our learning stage matches the error resulting from $\alpha^*$, yielding the bound.

□

### 5.3 Separations for Mean-Based and No-Swap-Regret Algorithms

While the previous result indicates that no-dynamic-regret algorithms are efficient to learn against in many cases, we show here that the exponential dependence for simpler *mean-based* algorithms is necessary: there exist games where learning the Stackelberg strategy requires exponentially many rounds. However, for the games we construct, we show that it is still possible to efficiently learn the Stackelberg strategy against a no-*swap*-regret learner.

**Theorem 7.** *There is a distribution over games $\mathcal{D}$ such that for a sampled game $G$:*

- *For any no-swap-regret learner used by the opponent, there is a strategy for the leader which yields an average reward of $\mathsf{Val}(G) - \varepsilon$ in $T = \text{poly}(M/\varepsilon)$ rounds.*
- *There is a mean-based no-regret algorithm such that, when used by the opponent, there is no strategy for the leader which yields an average reward of $\mathsf{Val}(G) - \varepsilon$ over $T$ rounds unless $T = \exp(\Omega(M))$.*

Our construction includes a set of actions for player $B$ which are best responses to pure actions from player $A$, and one such pure strategy pair will necessarily constitute the Stackelberg equilibrium; identifying each best response suffices for player $A$ to identify the Stackelberg strategy. The game also includes a number of *safety* actions for player $B$, which yield no reward for player $A$ with any strategy, yet allow player $B$ to "hedge" between multiple actions of player $A$. This poses a barrier to optimizing against a mean-based learner: the history must be heavily concentrated on a single action to observe the best response, and as such the history length must grow by a constant factor for each observation. However, against a no-swap-regret learner, it suffices for the optimizer to only play each action for a polynomially long window in order to identify the learner's best response, as we can track the accumulation of "swap-regret buffer" for any other action and show that it cannot be too large, limiting the number of rounds it can be played when it is not a current best response.

# A Properties and Separations for Generalized Equilibria

## A.1 Proof of Proposition 2

*Proof.* The set of $(\mathcal{F}_A, \mathcal{F}_B)$-equilibria includes all strategy profile distributions in which both constraints are satisfied. If a player receives substantially more than the corresponding value, this would imply a violation of the regret constraints for at least one of the players' learning algorithms. □

## A.2 Proof of Proposition 2

*Proof.* This follows from the proof of Theorem 4 in [**?**] upon noting that the assumption used there regarding no weakly dominated strategies can be replaced by any condition implying a unique best response to an approximate Stackelberg strategy. □

## A.3 Proof of Proposition 2

*Proof.* By definition, the set of $(\mathcal{F}_A, \mathcal{F}_B)$-equilbria $\sigma$ is a sub-polytope of $\Delta(\mathcal{A} \times \mathcal{B})$ defined via the following linear constraints:

- For each $f_A \in \mathcal{F}_A$, we have that

$$\sum_{i \in [M]} \sum_{j \in [N]} \sigma_{ij} u_A(a_i, b_j) \geq \sum_{i \in [M]} \sum_{j \in [N]} \sigma_{ij} u_A(a_{f(i)}, b_j).$$

- For each $f_B \in \mathcal{F}_B$, we have that

$$\sum_{i \in [M]} \sum_{j \in [N]} \sigma_{ij} u_B(a_i, b_j) \geq \sum_{i \in [M]} \sum_{j \in [N]} \sigma_{ij} u_B(a_{f(i)}, b_j).$$

The value $\mathsf{Val}_A(\mathcal{F}_A, \mathcal{F}_B)$ is the element $\sigma$ of this polytope that maximizes $\sum_{i \in [M]} \sum_{j \in [N]} \sigma_{ij} u_A(a_i, b_j)$. Optimizing this linear function over the above polytope can be done in time $\text{poly}(M, N, |\mathcal{F}_A|, |\mathcal{F}_B|)$ via any linear program solver. Computing $\mathsf{Val}_B(\mathcal{F}_A, \mathcal{F}_B)$ can be likewise done efficiently.

For player $A$, the regret comparator function sets $\emptyset$, $\mathcal{E}$, and $\mathcal{I}$ contain 0, $M$, and $M^2$ elements respectively. In all three of these cases $|\mathcal{F}_A| = \text{poly}(M)$; likewise, in all three of these cases $|\mathcal{F}_B| = \text{poly}(N)$ (and thus we can efficiently compute these values when $\mathcal{F}_A, \mathcal{F}_B \in \{\emptyset, \mathcal{E}, \mathcal{I}\}$). □

## A.4 Proof of Proposition 4

*Proof.* The statement follows by observing that

$$\mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_{\{A,B\}}(a,b)\right] = \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{(a,b)\sim\sigma^t}\left[u_{\{A,B\}}(a,b)\right]$$

$$\mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_A(f_A(a),b)\right] = \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{(a,b)\sim\sigma^t}\left[u_A(f_A(a),b)\right]$$

$$\mathop{\mathbb{E}}_{(a,b)\sim\sigma}\left[u_B(a,f_B(b))\right] = \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{(a,b)\sim\sigma^t}\left[u_B(a,f_B(b))\right]$$

which in turn are equivalent to the time-averaged utility of the play of players $A$ and $B$, the time-averaged utility for player $A$ under a deviation $f_A$, and the time-averaged utility for player $B$ under a deviation $f_B$. Applying the definition of average $\mathcal{F}$-regret and applying the given bounds on the $\mathcal{F}$-regret yields the conclusion of the first direction. The reverse direction follows by reversing the steps. □

## A.5 Reward Separations

We show that with respect to optimal values, these equilibrium classes are distinct, and there exist games where values do not collapse. The separations we show here consider the equilibrium cases either where both players have identical regret constraints, or where player $A$ is unconstrained. We note that while inspecting other cases, we identified similar examples for several other generalized equilibrium pairs, and we expect that strict separations exist between any distinct pair of generalized equilibria for the three regret notions we consider, in any direction not immediately precluded by the regret constraints. We are mostly interested in cases where $B$ is constrained, and $A$ may be constrained or unconstrained.

**Theorem 8.** *For each of the following, there exists a $4 \times 4$ game $G$ with rewards in $\{0, 1, 2\}$ where:*

1. $\mathsf{Val}_{(\emptyset,\mathcal{E})}(G) > \mathsf{Val}_{(\emptyset,\mathcal{I})}(G) > \mathsf{Val}_{(\mathcal{E},\mathcal{E})}(G) > \mathsf{Val}_{(\mathcal{I},\mathcal{I})}(G)$
2. $\mathsf{Val}_{(\emptyset,\mathcal{E})}(G) > \mathsf{Val}_{(\mathcal{E},\mathcal{E})}(G) > \mathsf{Val}_{(\emptyset,\mathcal{I})}(G) > \mathsf{Val}_{(\mathcal{I},\mathcal{I})}(G)$

*Proof.* We prove both results by exhibiting a game with the desired chain of inequalities, which we found by searching random examples of $4 \times 4$ games with values constrained in $\{0, 1, 2\}$ and computing the various values of the games with a linear programming library. The numerical values are easy to check with computation. The game $G_1 := (M_{A_1}, M_{B_1})$ satisfies the conditions for the first chain of inequalities, and the game $G_2 := (M_{A_2}, M_{B_2})$ satisfies the conditions for the second chain of inequalities. First we instantiate the game $G_1$:

$$M_{A_1} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 2 & 2 & 0 & 2 \\ 0 & 2 & 0 & 0 \end{bmatrix} \qquad M_{B_1} := \begin{bmatrix} 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The corresponding values for game $G_1$ are simple to check:

1. $\mathsf{Val}_{(\emptyset,\mathcal{E})}(G) = 8/5$.
2. $\mathsf{Val}_{(\emptyset,\mathcal{I})}(G) = 4/3$.
3. $\mathsf{Val}_{(\mathcal{E},\mathcal{E})}(G) = 1$.
4. $\mathsf{Val}_{(\mathcal{I},\mathcal{I})}(G) = 0$.

Then we instantiate the game $G_2$:

$$M_{A_2} := \begin{bmatrix} 2 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 \\ 0 & 2 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{bmatrix} \qquad M_{B_2} := \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 1 & 1 \end{bmatrix}$$

The corresponding values for game $G_2$ are simple to check:

1. $\mathsf{Val}_{(\emptyset,\mathcal{E})}(G) = 13/7$.
2. $\mathsf{Val}_{(\mathcal{E},\mathcal{E})}(G) = 12/7$.
3. $\mathsf{Val}_{(\emptyset,\mathcal{I})}(G) = 5/3$.
4. $\mathsf{Val}_{(\mathcal{I},\mathcal{I})}(G) = 4/3$.

□

# B Proof of Theorem 1

*Proof.* We begin with the first claim. To prove the forward direction, if there exists such a $\sigma$, then choose a pair of low-swap-regret algorithms $(\mathsf{Alg}_A, \mathsf{Alg}_B)$ such that the time-averaged trajectory over $T$ rounds is guaranteed to asymptotically converge to $\sigma$ (this is possible by either the results of [**?**] or Theorem 3, which we prove later in the paper). That is, if the two players play strategy $\sigma_t$ at round $t \in [T]$, then $\hat{\sigma} = \frac{1}{T}\sum_t \sigma_t$ satisfies $||\hat{\sigma} - \sigma||_\infty = o(1)$. It follows that $\sum_t u_A(\sigma_t) \geq T \cdot u_A(\sigma) - o(T) = T \cdot \mathsf{Stack}_A - o(T)$ and therefore player $A$ has at most an $o(T)$ incentive to deviate (by [**?**], they can obtain at most $\mathsf{Stack}_A\, T + o(T)$ against $\mathsf{Alg}_B$). Symmetric logic holds for player $B$.

To prove the reverse direction, assume $\mathsf{Alg}_A$ and $\mathsf{Alg}_B$ are no-swap-regret algorithms such that $(\mathsf{Alg}_A, \mathsf{Alg}_B)$ is an $o(T)$-approximate Nash equilibrium in the metagame. Since they are no-swap-regret, the time-averaged play of these two algorithms for $T$ rounds must converge to an $o(1)$-approximate correlated equilibrium $\hat{\sigma}_T$; moreover, since $(\mathsf{Alg}_A, \mathsf{Alg}_B)$ is an $o(T)$-approximate Nash equilibrium, $\hat{\sigma}_T$ must have the property that $u_A(\hat{\sigma}_T) \geq \mathsf{Stack}_A - o(1)$ and $u_B(\hat{\sigma}_T) \geq \mathsf{Stack}_B - o(1)$. Taking the limit as $T \to \infty$ and selecting a convergent subsequence of the $\hat{\sigma}_T$, this shows there must exist a correlated equilibrium $\sigma$ with the desired properties.

Likewise, similar logic proves the second claim with the following modifications. In the forward direction, we can now choose any pair of low-swap-regret algorithms $(\mathsf{Alg}_A, \mathsf{Alg}_B)$, and any correlated equilibrium $\sigma$ they asymptotically converge to is guaranteed to have the property that $u_A(\sigma) = \mathsf{Stack}_A$ and $u_B(\sigma) = \mathsf{Stack}_B$. In the reverse direction, since any correlated equilibrium is implementable by some pair of low-regret algorithms (again, by Theorem 3), the same logic shows that all correlated equilibria $\sigma$ must satisfy $u_A(\sigma) = \mathsf{Stack}_A$ and $u_B(\sigma) = \mathsf{Stack}_B$.

Finally, to see that these two conditions are efficiently checkable, note that: i. the two values $\mathsf{Stack}_A$ and $\mathsf{Stack}_B$ are efficiently computable given the game $G$, and ii. the set of correlated equilibria $\sigma$ form a convex polytope defined by a small ($\text{poly}(N, M)$) number of linear constraints (see Proposition 2). In particular, since $u_A(\sigma)$ and $u_B(\sigma)$ are simply linear functions of $\sigma$ for a given game $G$, we can efficiently check whether there exists any point in this polytope where $u_A(\sigma) = \mathsf{Stack}_A$ and $u_B(\sigma) = \mathsf{Stack}_B$. □

# C Proof of Theorem 2

*Proof.* We will show that (for almost all games $G$) if there is a correlated equilibrium $\sigma$ such that $u_A(\sigma) = \mathsf{Stack}_A$ and $u_B(\sigma) = \mathsf{Stack}_B$, then there exists a simultaneous unique Stackelberg equilibrium for both players in $G$, which must be a pure Nash equilibrium. Combined with Theorem 1, this implies the theorem statement.

We will rely on the following fact: in almost all games $G$, both players have a unique Stackelberg strategy. To see this, consider the following method for computing $A$'s Stackelberg strategy. For each pure strategy $b_j$ for player $B$, consider the convex set $A_j \subseteq \Delta(\mathcal{A})$ containing the mixed strategies for player $A$ which induce $b_j$ as a best response (i.e., $A_j = \{\alpha \in \Delta(\mathcal{A}) \mid b_j \in \mathsf{BR}(\alpha)\}$). Then, for each $j \in [N]$, compute the strategy $\alpha_j \in A_j$ which maximizes $u_A(\alpha_j, b_j)$. The Stackelberg value $\mathsf{Stack}_A$ is then given by $\max_j u_A(\alpha_j, b_j)$. In order for this to stem from a unique Stackelberg equilibrium, it is enough that: 1. the maximum utility is not attained by more than one $j$, and 2. for each $j$, the optimizer $\alpha_j \in A_j$ is unique.

These two properties are guaranteed to hold in almost all games. To see this, first note that the convex sets $A_j$ are determined entirely by the utilities $u_B$, so we will treat these as fixed. Now, given any convex set $A_j$, the extremal point in a randomly perturbed direction will be unique with probability 1 – but since $\alpha_j$ is simply the extremal point of $A_j$ in the direction specified by $u_A(\cdot, b_j)$ (which is a randomly perturbed direction), so $\alpha_j$ is unique in almost all games. Finally, if we perturb the magnitude of each of the utilities $u_A(\cdot, b_j)$ (keeping the direction the same), the maximizer $\max_j u_A(\alpha_j, b_j)$ will also be unique almost surely.

Let $(\alpha_A, b_A)$ be the Stackelberg equilibrium for player $A$ and let $(a_B, \beta_B)$ be the Stackelberg equilibrium for player $B$. Now, consider the aforementioned correlated equilibrium $\sigma \in \Delta(A \times B)$. We will begin by decomposing it into its marginals based on its first coordinate; that is, we will write $\sigma = \sum_{i=1}^{M} \lambda_i(a_i, \beta_i)$ for some mixed strategies $\beta_i \in \Delta(\mathcal{B})$ and weights $\lambda_i$ (with $\sum_i \lambda_i = 1$). By the definition of correlated equilibria, note that each $a_i$ belongs to $\mathsf{BR}(\beta_i)$. But this means that $u_B(a_i, \beta_i) \leq \mathsf{Stack}_B$, with equality holding iff $(a_i, \beta_i) = (a_B, \beta_B)$ (due to uniqueness of Stackelberg). Therefore, in order for $u_B(\sigma) = \mathsf{Stack}_B$, we must have that $\sigma = (a_B, \beta_B)$. By symmetry, we must also have that $\sigma = (\alpha_A, b_A)$. If both these are true, then $\sigma$

is a pure strategy correlated equilibrium of the game, and is hence a pure strategy Nash equilibrium (and moreover, is also the Stackelberg equilibrium for both $A$ and $B$). $\square$

# D Proof of Theorem 3

*Proof.* Let $\sigma$ be the joint distribution over action pairs corresponding to $E$. Let $T$ denote the total number of steps we run the algorithm for; we will use $t \leq T$ as a changing step size. Suppose both player $A$ and player $B$ know $\sigma$[2]. We will define $\mathcal{L}_A^*(E)$ and $\mathcal{L}_B^*(E)$ in two phases: in the first phase, $A$ and $B$ trust their opponent and play according to deterministic sequences corresponding to approximations of $\sigma$. If either player violates the other's trust $o(T)$ times, then the player defects to playing $\mathcal{L}_A$ or $\mathcal{L}_B$ respectively forever after.

First we elaborate upon the trusting phase. Both players consider windows of length $\mathsf{Length}(t)$ which is monotonically increasing in $t$ and also which grows sub-linearly in $t$. For concreteness, we pick a sub-linear monotonic increasing growth rate of $\mathcal{O}(\sqrt{t})$ and describe how to implement the schedule of window lengths. We can keep track of a real-valued variable $Z_t$ with $Z_1 = M \cdot N$, and after each window completes, update it by $Z_{t_{\text{next}}} = Z_t + \frac{1}{2\sqrt{t}}$ where $t$ is the step at the end of the window. To get an integral window length, we define $\mathsf{Length}(t) := \lfloor Z_t \rfloor$. Thus in this case, the $\mathsf{Length}(t)$ grows as $O(\sqrt{t})$, satisfying both conditions. Both players then compute a weighting instantiated with pairs of pure strategies by assigning $c_i := \lfloor \mathsf{Length}(t) \cdot \sigma_i \rfloor$ example pairs (each of weight $1/\mathsf{Length}(t)$) to pure strategy pair $i \in [M \cdot N]$. This weighted distribution approximates $\sigma$ given $\mathsf{Length}(t)$ samples. Note that the rounding approximation is feasible given only $\mathsf{Length}(t)$ samples since $\sum_{i=1}^{M \cdot N} c_i \leq \mathsf{Length}(t)$. These pure strategy pair samples are then lexicographically ordered. Then, both players act according to the pure strategies in order, thereby (over the window) achieving an $(M \cdot N)/\mathsf{Length}(t)$ $\ell_1$ approximation to $\sigma$:

$$\sum_{i=1}^{M \cdot N} \left| \sigma_i - \frac{c_i}{\mathsf{Length}(t)} \right| = \sum_{i=1}^{M \cdot N} \left| \sigma_i - \frac{\lfloor \mathsf{Length}(t) \cdot \sigma_i \rfloor}{\mathsf{Length}(t)} \right| \leq \frac{M \cdot N}{\mathsf{Length}(t)}.$$

This process repeats for every window.

The distrustful phase occurs if one of the players does not follow the agreed-upon instructions $T_{\text{distrust}}$ times, where $T_{\text{distrust}}$ is taken to be $o(T)$. After this many violations, Player $A$ defaults to playing $L_A$ and likewise Player $B$ defaults to playing $L_B$ ever after.

We now show that this algorithm satisfies both conditions in the theorem statement. First, if both players use $\mathcal{L}_A^*(E)$ and $\mathcal{L}_B^*(E)$, the play converges to $\sigma$, the joint distribution of play corresponding to $E$. This point is immediate to observe since $(M \cdot N)/\mathsf{Length}(t) \to 0$ as $t \to \infty$ as $\mathsf{Length}(t)$ is monotone increasing in $t$.

Now we prove that both players are no-$\mathcal{F}$-regret with respect to any adversary. First we show no-$\mathcal{F}$-regret for both players in the case where Player $A$ plays $\mathcal{L}_A^*(E)$ and Player $B$ plays $\mathcal{L}_B^*(E)$. Let $\hat{\sigma}_t$ be the approximation to $\sigma$ implemented over the window corresponding to final step $t$, and suppose that $\|\sigma - \hat{\sigma}_t\|_1 < \varepsilon_t$. Recalling the proof of Theorem 3, for Player $A$ (and analogously for Player $B$) we can bound

$$\begin{aligned} \left| \mathop{\mathbb{E}}_{(a,b)\sim\sigma} [u_A(a,b)] - \mathop{\mathbb{E}}_{(a,b)\sim\hat{\sigma}_t} [u_A(a,b)] \right| &= \left| (\sigma - \hat{\sigma}_t)^\top u_A \right| \\ &\leq \|\sigma - \hat{\sigma}_t\|_1 \cdot \|u_A\|_2 \leq \varepsilon_t \cdot C \cdot \sqrt{M \cdot N}, \end{aligned}$$

where here we interpret $\sigma, \hat{\sigma}_t, u_A, u_B \in \mathbb{R}^{M \times N}$ as vectors over the space of all action pairs. Thus for this particular window, the overall gap from the expected reward for $\sigma$ is $\varepsilon_t \cdot C \cdot \sqrt{M \cdot N}$.

Then we can similarly upper bound $\mathbb{E}_{(a,b)\sim\hat{\sigma}_t} [u_A(f_A(a), b)] \leq \mathbb{E}_{(a,b)\sim\sigma} [u_A(f_A(a), b)] + \varepsilon_t \cdot C \cdot M\sqrt{N}$ for

[2] $\sigma$ can be communicated from Player $A$ to Player $B$ during a burn-in phase of length $> M \cdot N$, the dimension of the discrete joint distribution over pure player strategy pairs.

any choice of $f_A \in \mathcal{F}_A$:

$$\begin{aligned}\left| \mathop{\mathbb{E}}_{(a,b)\sim\sigma} \left[u_A\left(f_A(a), b\right)\right] - \mathop{\mathbb{E}}_{(a,b)\sim\hat{\sigma}_t} \left[u_A\left(f_A(a), b\right)\right] \right| &= \left| \sum_{k=1}^{M}\sum_{j=1}^{N} \left(\hat{\sigma}_t(k,j) - \sigma(k,j)\right) \cdot \sum_{i=1}^{M} f_A(a_k)_i \cdot u(\cdot, b_j) \right| \\ &\leq \|\sigma - \hat{\sigma}_t\|_1 \cdot \| \left[f_A(a_1)^\top u_A(\cdot, b_1), \cdots, f_A(a_M)^\top u_A(\cdot, b_N)\right] \|_2 \\ &\leq \varepsilon_t \cdot \sqrt{M \cdot N} \cdot \max_{k,j} \|f_A(a_k)\|_2 \cdot \|u_A(\cdot, b_j)\|_2 \\ &\leq \varepsilon_t \cdot \sqrt{M \cdot N} \cdot 1 \cdot \sqrt{M \cdot C^2} \\ &= \varepsilon_t \cdot M \cdot \sqrt{N} \cdot C.\end{aligned}$$

Then recall that $\varepsilon_t \leq \frac{M \cdot N}{\mathsf{Length}(t)}$. Thus, overall, the average regret using due to the window is bounded by

$$\frac{1}{\mathsf{Length}(t)} \mathsf{Reg}_{\mathcal{F}}(\hat{\sigma}_t, t) \leq \frac{1}{\mathsf{Length}(t)} \mathsf{Reg}_{\mathcal{F}}(\sigma, t) + C_2 \cdot \frac{1}{\mathsf{Length}(t)},$$

where $C_2$ is another constant depending on $C, M, N$ and where we use the shorthand $\mathsf{Reg}_{\mathcal{F}}(\cdot, t)$ to denote the $\mathcal{F}$-regret over the window ending in step $t$. Now call $\hat{\sigma}$ the strategy where the joint distribution $\hat{\sigma}_t$ as previously defined gets played in each window $t$. Now we can bound the total $\mathcal{F}$-regret for $\hat{\sigma}$ by the sum of the $\mathcal{F}$-regrets for each window (maximizing $f_A \in \mathcal{F}_A$ over the steps in each window makes it more competitive than optimizing only one $f_A$ over the whole length $T$ sequence). Thus for total $\mathcal{F}$-regret, we have:

$$\mathsf{Reg}_{\mathcal{F}}(\hat{\sigma}, T) \leq \quad \mathsf{Reg}_{\mathcal{F}}(\sigma, T) + \mathsf{NumWindows}(T) \cdot C_2 \quad \leq \quad \mathsf{Reg}_{\mathcal{F}}(\sigma, T) + o\left(T\right),$$

where

$$\mathsf{NumWindows}(T) := \min_{\sum_{t=1}^{k} \mathsf{Length}(t) \geq T} k.$$

The last step follows since $\mathsf{NumWindows}(T) \leq o(T)$, because $\mathsf{Length}(T) \leq o(T)$.

Since we already know that the strategy $\sigma$ is no-$\mathcal{F}$-regret and $\mathsf{Length}(T)$ is $o(T)$, we have proven that playing $\hat{\sigma}$ is no-$\mathcal{F}$-regret in the case where Player $A$ plays $\mathcal{L}_A^*(E)$ and Player $B$ plays $\mathcal{L}_B^*(E)$.

The second case where the opposing player does not cooperate is easier: after at most $o(T)$ steps, the player switches to an algorithm $\mathcal{L}_A$ or $\mathcal{L}_B$ respectively which is no-$\mathcal{F}$-regret and incurrs only $o(T)$ additional regret. Thus the theorem statement holds.

□

# E Proof of Theorem 7

*Proof.* Our game consists of $M$ actions $\mathcal{A}$ for the optimizer, and $N = 2M + \binom{M}{2}$ actions for the learner, which are divided into $M$ *primary* actions $\mathcal{B}$, $M$ *secondary* actions $\mathcal{S}$, and $\binom{M}{2}$ *safety* actions $\mathcal{Y}$.

If we restrict the learner to only playing primary actions, the game somewhat resembles a coordination game, where each pure strategy pair $(a_j, b_j)$ is a Nash equilibrium. However, the set $\mathcal{B}$ is comprised of both *undominated* actions $\mathcal{B}_U$ and *dominated* actions $\mathcal{B}_D$, which are unknown to the optimizer, and where each $b_j \in \mathcal{B}_d$ is weakly dominated by the secondary action $s_j$. The optimizer receives reward 0 whenever the learner plays a secondary action, and so the challenge for the optimizer is to identify the pair $(a_j, b_j)$ which maximizes $u_A(a_j, b_j)$, for $b_j \in \mathcal{B}_D$, which will be the Stackelberg equilibrium. Further, the safety actions $y_{ij}$ essentially allow the learner to hedge between two actions; this does not pose substantial difficulty for the optimizer when the learner is no-swap-regret, yet creates an insurmountable barrier for learning the Stackelberg equilibrium in sub-exponential time against a mean-based learner.

An instance of a game $G \in \mathcal{G}$ is specified by the partition of $\mathcal{B}$ into $\mathcal{B}_U$ and $\mathcal{B}_D$. There is an action $s_j \in \mathcal{S}$ for each $j$, and for each pair $(i, j)$ with $i < j$ there is an action $y_{ij} \in \mathcal{Y}$. The rewards for a game $G$ are as follows. For any strategy pair, the optimizer's utility is given by:

- $u_A(a_j, b_j) = j/M$ for $b_j \in \mathcal{B}$;

- $u_A(a_i, b_j) = 0$ for $b_j \in \mathcal{B}$ and with $i \neq j$;
- $u_A(a_i, s_j) = 0$ for any $s_j \in \mathcal{S}$;
- $u_A(a_i, y_{jk}) = 0$ for any $y_{jk} \in \mathcal{Y}$;

and the learner's utility is given by:

- For $b_j \in \mathcal{B}_U$:
  - $u_B(a_j, b_j) = 1$;
  - $u_B(a_i, b_j) = 0$ for $i \neq j$;
- For $b_j \in \mathcal{B}_D$:
  - $u_B(a_i, b_j) = 0$ for any $i$;
- For $s_j \in \mathcal{S}$:
  - $u_B(a_j, s_j) = 1$ if $b_j \in \mathcal{B}_D$;
  - $u_B(a_j, s_j) = 0$ if $b_j \in \mathcal{B}_U$;
  - $u_B(a_i, s_j) = 0$ for $i \neq j$;
- For $y_{ij} \in \mathcal{Y}$:
  - $u_B(a_i, y_{ij}) = u_B(a_j, y_{ij}) = 2/3$;
  - $u_B(a_k, y_{ij}) = 0$ for $i, j \neq k$.

We assume that $\mathcal{B}_U$ is non-empty, and so there is some optimal pure Nash equilibrium $(a_i^*, b_i^*)$ which yields a reward of $i/M$; it is simple to check that this is also the Stackelberg equilibrium.

**Optimizing Against No-Swap Learners.** First, we give a method for matching the Stackelberg value against an arbitrary no-swap-regret learner, which corresponds to the pair $(a_j, b_j)$ for the largest value $j$ such that $b_j \in \mathcal{B}_U$. Consider a no-swap-regret learner which obtains a regret bound of $\tau = O(T^c)$ over $T$ rounds. Let $\mathsf{SR}_t(b, b')$ for any learner actions $b$ and $b'$ denote the $t$-round cumulative swap regret between $b$ and $b'$, i.e. the total change in reward which would have occurred if $b'$ was played instead for each of the first $t$ rounds in which $b$ was played. To model the behavior of an arbitrary no-swap-regret learner, we disallow the learner from taking any action which would increase $\mathsf{SR}_t(b, b')$ above $\tau$, given the loss function for the current round, and otherwise allow the action to be chosen adversarially. While our model is deterministic for simplicity, it is straightforward to extend to the analysis to algorithms whose regret bounds hold in only expectation, e.g. by considering a distribution over values of $\tau$ in accordance with Markov's inequality (as no algorithm can have negative expected regret against arbitrary adversaries) and considering our expected regret to the Stackelberg value.

Our strategy for the optimizer is:

- For each $i \in [M]$, play $a_i$ until either $b_i$ or $s_i$ is observed at least $t^* > \tau$ times;
- Return $a_i^*$ for the largest $i$ such that $b_i$ is observed $t^*$ times.

We show that this takes at most $O(T^c \cdot M^3)$ rounds. Once $a_i^*$ is identified, we can commit to playing it indefinitely, at which point the learner must play $b_i^*$ in all but at most $O(T^c \cdot \text{poly}(M))$ rounds, and so with $T = O(\text{poly}(M/\varepsilon))$ rounds we can increase the total fraction of rounds in which $(a_i^*, b_i^*)$ is played to $1 - \varepsilon$, which yields the desired average reward bound.

The key to analyzing the runtime of our strategy is to consider the "buffer" in regret between any pair of actions before the threshold of $\tau$ is reached, which enables us to the bound the number of rounds in which instantaneously suboptimal actions are played. Note that prior the start of window $i$ (where $a_i$ is played), both $b_i$ and $s_i$ obtain reward 0 in each round, and as such cannot decrease their expected regret relative to any other action, as all rewards in the game are non-negative. Further, for any previous window $j$, both $b_i$

and $s_i$ incur regret of 1 with respect to either $b_j$ or $s_j$, as well as between the suboptimal and optimal action in window $i$, and thus cannot be observed more than $\tau$ times in the window. As such, observing $b_i$ at least $t^*$ times in window $i$ indicates that $b_i \in \mathcal{B}_U$ (and likewise observing $b_i$ at least $t^*$ times indicates that $b_i \in \mathcal{B}_D$).

Any action $b \neq \mathsf{BR}(a_i)$ will incur positive swap regret with respect to $\mathsf{BR}(a_i)$, and cannot be played in window $i$ once $\mathsf{SR}_t(b, \mathsf{BR}(a_i)) \geq \tau$. Each action begins with $\mathsf{SR}_1(b, \mathsf{BR}(a_i)) = 0$ at time $t = 1$; for each of the learner's actions, we consider the rate at which its buffer decays, as well as instances in which swap regret can decrease:

- Previously optimal $b \in \mathcal{B} \cup \mathcal{S} \setminus \mathsf{BR}(a_i)$: actions in $\mathcal{B} \cup \mathcal{S}$ can only accumulate negative swap regret with respect to $\mathsf{BR}(a_i)$ during rounds in which they were previously optimal; any previous optimum $b = \mathsf{BR}(a_j)$ for $j < i$ was played at most $t^*$ times during window $j$, and so we have that $\mathsf{SR}_t(b, \mathsf{BR}(a_i)) \geq -t^*$.

- All $b \in \mathcal{B} \cup \mathcal{S} \setminus \mathsf{BR}(a_i)$: ignoring any previously accumulated regret buffer, each of these $2M - 1$ actions can be played at most $\tau$ rounds during window $i$ before exhausting their initial buffer. Accounting for possible previous optima with $\mathsf{SR}_t(b, \mathsf{BR}(a_i)) < 0$, the number of rounds during window $i$ in which some $b \in \mathcal{B} \cup \mathcal{S} \setminus \mathsf{BR}(a_i)$ is played is at most $Mt^* + (2M - 1)\tau$.

- Safety actions $y_{jk} \in \mathcal{Y}$: Suppose neither $a_j$ or $a_k$ have been played yet by the optimizer, including in the current window. As was the case for other actions which have never yielded positive instantaneous reward, $y_{jk}$ can be played at most $\tau$ times before $\mathsf{SR}_t(y_{jk}, \mathsf{BR}(a_i)) \geq \tau$. If $j = i$, i.e. this is the first window in which $y_{jk}$ obtains positive instantaneous reward, the per-round regret is $1/3$, and so at it can be played for most $3\tau$ rounds. Further, $y_{jk}$ a obtains a regret of $-2/3$ with respect to $\mathsf{BR}(a_k)$. If $k = i$ and the window for $a_j$ has already been completed, $y_{jk}$ can be played for at most $9\tau$ rounds, as initially we have that $\mathsf{SR}_t(y_{jk}, \mathsf{BR}(a_i)) \geq -2\tau$, which again increases by $1/3$ per round. We then have that the total amount of rounds with safety actions played during window $i$ is at most $(12M + M^2)\tau$, as there are fewer than $M^2$ total safety actions, and fewer than $M$ in each of the latter cases.

This yields a per-window runtime across all actions of at most $Mt^* + (M^2 + 10M - 1)\tau$, which is $O(T^c \cdot M^3)$ across all windows, and so we obtain the desired result for optimizing against arbitrary no-swap-regret learners.

**Optimizing Against Mean-Based Learners.** Here, we show that there are mean-based no-regret algorithms for which exponentially many rounds are required for an optimizer to approximate the Stackelberg value against a learner. When considering horizons which are superpolynomial in the parameters of the game, it is most natural to consider algorithms with regret bounds which are non-trivial for smaller horizons, as well as an anytime variant of the mean-based property. We define an extension of the classical Multiplicative Weight Updates algorithm (MWU; see [**?**] for a survey), called Rounded Mean-Based Doubling, which inherits both properties in the anytime setting.

**Algorithm 1** Rounded Mean-Based Doubling (RMBD)

Initialize and run MWU for $T_1 := 2$ rounds and $n$ actions.
Let $T_2 := 2T_1$ and $i := 2$.

**while** $T_i \leq T$ **do**
  Initialize MWU for $T_i$ rounds and $n$ actions.
  Simulate running MWU for $T_{i-1}$ rounds, using the average of the first $T_{i-1}$ rewards each round.
  For $T_{i-1}$ rounds, run MWU with action probabilities rounded to multiples of $4\gamma = \tilde{O}(T_i^{-1/2})$.
  Let $T_{i+1} = 2T_i$ and $i := i + 1$. $=0$

**Lemma 1.** *When running* RMBD *for $T$ rounds, the following hold at any round $t \leq T$:*

- RMBD *has cumulative regret $\tilde{O}(n\sqrt{t})$;*

- *If action $j$ has the highest cumulative reward and $\sigma_{i,t} \leq \sigma_{j,t} - \tilde{O}(\sqrt{t})$, then action $i$ is played with probability 0 at round $t$.*

*Proof.* Let $C\sqrt{t}$ bound the regret of MWU over $t$ rounds (where $C = O(\sqrt{\log n})$), and let $D = \sqrt{2}C + \tilde{O}(n)$. We can bound the regret of RMBD over $T_i$ rounds by $D\sqrt{T_i}$ via induction (which holds trivially at $T_1$). Suppose it holds for some $T_i$. Let $R(T_i)$ be the true reward obtained by RMBD over $T_i$ rounds, which is at least $\sigma_{j^*,T_i} - D\sqrt{T_i}$, where $\sigma_{j^*,T_i}$ is the cumulative reward of the best action over $T_i$ rounds. Consider our simulation of MWU over $T_i$ rounds using the average reward function. As the reward function is identical each round, and the cumulative reward for each action $j$ is equivalent under averaging, the measured reward $\hat{R}(T_i)$ from the simulated run is at most $\sigma_{j^*,T_i}$ after $T_i$ rounds. Upon continuing to run this instance of MWU for an additional $T_i$ rounds, the regret bound ensures that the total measured reward $\hat{R}(T_{i+1})$ is at least $\sigma_{j^*,2T_i} - C\sqrt{2T_i}$. Rounding probabilities contributes at most an additional $2n\gamma T_i$ to the regret; it suffices to implement rounding by reallocating probability mass from any $p_{i,t} < 2\gamma$ onto other actions arbitrarily, to avoid renormalization. The total reward of RMBD over $2T_i = T_{i+1}$ is given by its cumulative reward at $T_i$, as well as the additional reward obtained by the MWU instance over the next $T_i$ rounds, and so we have that

$$\begin{aligned} R(T_{i+1}) &= R(T_i) + \hat{R}(T_{i+1}) - \hat{R}(T_i) \\ &\geq \sigma_{j^*,T_{i+1}} - D\sqrt{T_i} - C\sqrt{2T_i} - 2n\gamma T_i \\ &\geq \sigma_{j^*,T_{i+1}} - D\sqrt{T_{i+1}}, \end{aligned}$$

which yields the bound for every $T_i$. We can extend this to any $t \in [T_i, T_{i+1}]$ with at most a factor 2 increase to cumulative regret.

To bound the selection frequency of actions with suboptimal cumulative reward, we recall the mean-based analysis of MWU given in Theorem D.1 from [**?**], which shows that the selection frequency $p_{k,t}$ for action $k$ at time $t$ is at most $\gamma = \frac{2\log(\sqrt{T\log n})}{\sqrt{T\log n}}$ if $\sigma_{k,t} \leq \sigma_{j,t} - \gamma T$ for the action $j$ with highest cumulative reward. As such, any action whose cumulative reward $\sigma_{k,t} \leq \sigma_{j,t} - \tilde{O}(\sqrt{t})$ will be played with probability 0. □

Suppose a learner plays the action with highest cumulative reward at each round for $t_{\text{burn}} = \tilde{\Omega}(M^2)$ rounds, then plays RMBD thereafter for a total of $T$ rounds. Note that this maintains the both properties of RMBD for all $t$. We show that at least $T = \exp(\Omega(M))$ rounds are required to identify the Stackelberg strategy. The optimizer must check the learner's pure best response to each $a_j$ for identification with certainty, and it is straightforward to construct a distribution in which any strategy which does not observe $\mathsf{BR}(a_j)$ for all $j$ will have linear regret to $\mathsf{Stack}_A$ in expectation (e.g. where $\mathcal{B}_U$ contains one action chosen uniformly at random). The difficulty in exploration of the best responses comes from the safety actions, as $a_j$ must have been played more frequently than any other action in order to not be dominated by some safety action. Let $\rho_{j,t}$ denote the number of rounds in which the optimizer has played $a_j$ out of the first $t$. Observe that by construction of the game and the properties of RMBD, an primary or secondary action $b_j$ or $s_j$ in $\mathsf{BR}(a_j)$ will only be played with positive probability when:

$$\begin{aligned} \rho_{j,t} &\geq \frac{2}{3}(\rho_{j,t} + \rho_{k,t}) - \tilde{O}(\sqrt{t}) \\ &= 2\rho_{k,t} - \tilde{O}(\sqrt{t}) \end{aligned}$$

for all $k$, which necessitates that $\rho_{j,t} \geq \frac{2t}{M} - \tilde{O}(\sqrt{t})$. Taking $t_{\text{burn}}$ sufficiently large, we have that $\rho_{j,t} \geq \frac{3}{2}\rho_{k,t}$ for any $t \geq t_{\text{burn}}$ and all $k$. For any subsequent observation $\mathsf{BR}(a_k)$ at $t'$, we must have that $\rho_{k,t'} \geq \frac{3}{2}\rho_{j,t}$, and so the number of rounds required to play an action before observing its best response grow at a rate of at least $(3/2)^M$, which completes the proof.

□