A Properties and separations for generalized equilibria

A.1 Proof of Proposition 1

Proof. The set of (Φ_A, Φ_B)-equilibria includes all strategy profile distributions in which both constraints are satisfied. If a player receives substantially more or less 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. The statement follows by observing that

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

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

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

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 Φ-regret and applying the given bounds on the Φ-regret yields the conclusion of the first direction. The reverse direction follows by reversing the steps. □

A.3 Proof of Proposition 4

Proof. Observe that under any strategy (α, b) where $b \in \mathsf{BR}(\alpha)$, player B cannot have any swap-regret, and so any Stackelberg equilibrium is also a $(\emptyset, \mathcal{I})$-equilibrium. Further, the marginal distributions over the optimal $(\emptyset, \mathcal{I})$-equilibrium for player A over each b_i cannot have distinct expected value for player A, as otherwise this would contradict optimality, and so an optimal $(\emptyset, \mathcal{I})$-equilibrium is either a single Stackelberg equilibrium or a mixture of Stackelberg equilibria with equal value. □

A.4 Proof of Proposition 3

Proof. By definition, the set of (Φ_A, Φ_B)-equilibria φ is a sub-polytope of $\Delta(\mathcal{A} \times \mathcal{B})$ defined via the following linear constraints:

- For each $f_A \in \Phi_A$, we have that

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

- For each $f_B \in \Phi_B$, we have that

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

The value $\mathsf{Val}_A(\Phi_A, \Phi_B)$ corresponds to the element φ of this polytope that maximizes $\sum_{i\in[M]}\sum_{j\in[N]}\varphi_{ij}\,u_A(a_i,b_j)$. Optimizing this linear function over the above polytope can be done in time $\mathrm{poly}(M, N, |\,\Phi_A\,|, |\,\Phi_B\,|)$ via any linear program solver. Computing $\mathsf{Val}_B(\Phi_A, \Phi_B)$ 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 $|\,\Phi_A\,| = \text{poly}(M)$; likewise, in all three of these cases $|\,\Phi_B\,| = \text{poly}(N)$ (and thus we can efficiently compute these values when $\Phi_A, \Phi_B \in \{\emptyset, \mathcal{E}, \mathcal{I}\}$). □

A.5 Reward separations

We show that with respect to optimal values, these equilibrium classes are often 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 4×4 game G with rewards in $\{0, 1, 2\}$ where:*

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

Proof. We prove both results by exhibiting a game with the desired chain of inequalities, which we found by searching random examples of 4×4 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. $\mathsf{Val}_A(\emptyset, \mathcal{E}) = 8/5$.
2. $\mathsf{Val}_A(\emptyset, \mathcal{I}) = 4/3$.
3. $\mathsf{Val}_A(\mathcal{E}, \mathcal{E}) = 1$.
4. $\mathsf{Val}_A(\mathcal{I}, \mathcal{I}) = 0$.

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

□

B Deviation to weaker regret classes

In Section 3, we show that if two players are playing no-swap-regret strategies against one another, it is often in the interest of each player to switch to playing their Stackelberg strategy (in particular, this is true whenever the game does not have a pure Nash equilibrium). However, as we later argue, learning ones Stackelberg strategy in such a game can be difficult. It is therefore natural to ask whether there are beneficial deviations to computationally efficient strategies. In particular, is it ever in a player's interest to weaken their regret benchmark, and e.g. switch from playing a no-swap-regret strategy to a no-external-regret strategy?

We give an example showing this is true in a fairly strong sense: we exhibit a game G where if player A switches from playing a no-swap-regret algorithm to *any* no-external-regret algorithm, their asymptotic utility never decreases and sometimes strictly increases – i.e., there is no downside to switching to an external regret algorithm (and potentially a high upside). We have the following theorem.

Theorem 9. *There exists a game G where* $\mathsf{MinVal}_A(\mathcal{E},\mathcal{I}) \geq \mathsf{Val}_A(\mathcal{I},\mathcal{I})$ *and* $\mathsf{Val}_A(\mathcal{E},\mathcal{I}) \geq \mathsf{Val}_A(\mathcal{I},\mathcal{I})$.

Proof. Consider the game G specified by the two payoff matrices

$$M_A := \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad M_B := \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}.$$

The corresponding values for this game are simple to compute:

1. $\mathsf{Val}_A(\mathcal{I},\mathcal{I}) = \mathsf{MinVal}_A(\mathcal{I},\mathcal{I}) = 0$.
2. $\mathsf{MinVal}_A(\mathcal{E},\mathcal{I}) = 0$.
3. $\mathsf{Val}_A(\mathcal{E},\mathcal{I}) = 1$.

□

C Proof of Theorem 1

Proof. Let φ be the joint distribution over action pairs corresponding to Ψ. 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 φ[2]. We will define $\mathcal{L}_A^*(\Psi)$ and $\mathcal{L}_B^*(\Psi)$ in two phases: in the first phase, A and B trust their opponent and play according to deterministic sequences corresponding to approximations of φ. 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 $c_i := \lfloor \mathsf{Length}(t) \cdot \varphi_i \rfloor$ example pairs (each of weight $1/\mathsf{Length}(t)$) to pure strategy pair $i \in [M \cdot N]$. This weighted distribution approximates φ 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

[2] φ 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.

ordered. Then, both players act according to the pure strategies in order, thereby (over the window) achieving an $(M \cdot N)/\,\mathsf{Length}(t)$ ℓ_1 approximation to φ:

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

This process repeats for every window.

The distrustful phase occurs if one of the players does not follow the agreed-upon instructions T_{distrust} times, where T_{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 $\mathcal{L}_A^*(\Psi)$ and $\mathcal{L}_B^*(\Psi)$, the play converges to φ, the joint distribution of play corresponding to Ψ. 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-Φ-regret with respect to any adversary. First we show no-Φ-regret for both players in the case where Player A plays $\mathcal{L}_A^*(\Psi)$ and Player B plays $\mathcal{L}_B^*(\Psi)$. Let $\hat{\varphi}_t$ be the approximation to φ implemented over the window corresponding to final step t, and suppose that $\|\varphi - \hat{\varphi}_t\|_1 < \varepsilon_t$. Recalling the proof of Theorem 1, for Player A (and analogously for Player B) we can bound

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

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

Then we can similarly upper bound $\mathbb{E}_{(a,b)\sim\hat{\varphi}_t}\left[u_A\left(f_A(a),b\right)\right] \le \mathbb{E}_{(a,b)\sim\varphi}\left[u_A\left(f_A(a),b\right)\right] + \varepsilon_t\cdot C\cdot M\sqrt{N}$ for any choice of $f_A\in\Phi_A$:

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

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

$$\frac{1}{\mathsf{Length}(t)}\,\mathsf{Reg}_\Phi(\hat{\varphi}_t,t) \le \frac{1}{\mathsf{Length}(t)}\,\mathsf{Reg}_\Phi(\varphi,t) + C_2\cdot\frac{1}{\mathsf{Length}(t)},$$

where C_2 is another constant depending on C, M, N and where we use the shorthand $\mathsf{Reg}_\Phi(\cdot,t)$ to denote the Φ-regret over the window ending in step t. Now call $\hat{\varphi}$ the strategy where the joint distribution $\hat{\varphi}_t$ as previously defined gets played in each window t. Now we can bound the total Φ-regret for $\hat{\varphi}$ by the sum of the Φ-regrets for each window (maximizing $f_A\in\Phi_A$ 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 Φ-regret, we have:

$$\mathsf{Reg}_\Phi(\hat{\varphi},T) \le \quad \mathsf{Reg}_\Phi(\varphi,T) + \mathsf{NumWindows}(T)\cdot C_2 \quad \le \quad \mathsf{Reg}_\Phi(\varphi,T) + o\left(T\right),$$

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 φ is no-Φ-regret and $\mathsf{Length}(T)$ is $o(T)$, we have proven that playing $\hat{\varphi}$ is no-Φ-regret in the case where Player A plays $\mathcal{L}_A^*(\Psi)$ and Player B plays $\mathcal{L}_B^*(\Psi)$.

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-Φ-regret and incurrs only $o(T)$ additional regret. Thus the theorem statement holds.

□

D Proof of Theorem 2

Proof. We begin with the first claim. To prove the forward direction, if there exists such a φ, then choose a pair of low-swap-regret algorithms $(\mathcal{L}_A, \mathcal{L}_B)$ such that the time-averaged trajectory over T rounds is guaranteed to asymptotically converge to φ (this is possible by either the results of (11), or our Theorem 1). That is, if the two players play strategy φ_t at round $t \in [T]$, then $\hat{\varphi} = \frac{1}{T}\sum_t \varphi_t$ satisfies $||\hat{\varphi} - \varphi||_\infty = o(1)$. It follows that $\sum_t u_A(\varphi_t) \geq T \cdot u_A(\varphi) - o(T) = T \cdot \mathsf{Stack}_A - o(T)$ and therefore player A has at most an $o(T)$ incentive to deviate (by (10), they can obtain at most $\mathsf{Stack}_A\, T + o(T)$ against $\mathcal{L}_B$). Symmetric logic holds for player B.

To prove the reverse direction, assume $\mathcal{L}_A$ and $\mathcal{L}_B$ are no-swap-regret algorithms such that $(\mathcal{L}_A, \mathcal{L}_B)$ 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 $\hat{\varphi}_T$; moreover, since $(\mathcal{L}_A, \mathcal{L}_B)$ is an $o(T)$-approximate Nash equilibrium, $\hat{\varphi}_T$ must have the property that $u_A(\hat{\varphi}_T) \geq \mathsf{Stack}_A - o(1)$ and $u_B(\hat{\varphi}_T) \geq \mathsf{Stack}_B - o(1)$. Taking the limit as $T \to \infty$ and selecting a convergent subsequence of the $\hat{\varphi}_T$, this shows there must exist a correlated equilibrium φ 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 $(\mathcal{L}_A, \mathcal{L}_B)$, and any correlated equilibrium φ they asymptotically converge to is guaranteed to have the property that $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$. In the reverse direction, since any correlated equilibrium is implementable by some pair of low-regret algorithms (again, by Theorem 1), the same logic shows that all correlated equilibria φ must satisfy $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$.

Finally, to see that these two conditions are efficiently checkable, note that: i. the two values Stack_A and Stack_B are efficiently computable given the game G, and ii. the set of correlated equilibria φ form a convex polytope defined by a small ($\mathrm{poly}(N, M)$) number of linear constraints (see Proposition 3). In particular, since $u_A(\varphi)$ and $u_B(\varphi)$ are simply linear functions of φ for a given game G, we can efficiently check whether there exists any point in this polytope where $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$. □

E Proof of Theorem 3

Proof. We will show that (for almost all games G) if there is a correlated equilibrium φ such that $u_A(\varphi) = \mathsf{Stack}_A$ and $u_B(\varphi) = \mathsf{Stack}_B$, then there exists a simultaneous unique Stackelberg equilibrium for both players in G, which must be a pure Nash equilibrium. Combined with Theorem 2, 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 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 α_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 α_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 (α_A, b_A) be the Stackelberg equilibrium for player A and let (a_B, β_B) be the Stackelberg equilibrium for player B. Now, consider the aforementioned correlated equilibrium $\varphi \in \Delta(A \times B)$. We will begin by decomposing it into its marginals based on its first coordinate; that is, we will write $\varphi = \sum_{i=1}^{M} \lambda_i (a_i, \beta_i)$ for some mixed strategies $\beta_i \in \Delta(\mathcal{B})$ and weights λ_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 $u_B(\varphi) = \mathsf{Stack}_B$, we must have that $\varphi = (a_B, \beta_B)$. By symmetry, we must also have that $\varphi = (\alpha_A, b_A)$. If both these are true, then φ 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). □

F Proof of Theorem 4

Proof. By Theorem 1, there is a pair of $\emptyset$-regret and $\mathcal{E}$-regret algorithms $\mathcal{L}_A^*$ and $\mathcal{L}_B^*$ which converge to a $(\emptyset, \mathcal{E})$-equilibrium for which player A obtains $\mathsf{Val}_A(\emptyset, \mathcal{E})$. By Proposition 1, this is optimal over all no-external-regret algorithms, as any adaptive strategy constitutes a no-$\emptyset$-regret algorithm. By Proposition 3 we can identify the optimal $(\emptyset, \mathcal{E})$-equilibrium in poly(M, N) time, which is sufficient to implement the algorithms $\mathcal{L}_A^*$ and $\mathcal{L}_B^*$ efficiently for any desired T. □

G Dominated-swapping external regret bounds for mean-based algorithms

For the following proof (of Theorem 10), we introduce the following notion of *dominated-swapping external regret*, a tighter upper bound on the behavior of mean-based algorithms than the standard no-external-regret guarantee.

Definition 8 (Dominated-swapping external regret)**.** *For a game G, let $D(G)$ be the set of dominated strategies for player B, i.e. $b_i \in D(G)$ if $b_i \notin \mathsf{BR}(\alpha)$ for all $\alpha \in \Delta(M)$. For $j, k \in [N]$ define $g_{jk}(b_i)$ as:*

$$g_{jk}(b_i) = \begin{cases} b_j & b_i \notin D(G) \\ b_k & b_i \in D(G) \end{cases}$$

i.e. $g_{jk}(b_i)$ swaps b_i to b_k if b_i is dominated and plays b_j otherwise. Let $\mathcal{E}_{D(G)} = \{g_{jk} : j, k \in [N]\}$ be the set of dominated-swapping external regret *comparators.*

This definition leads to the following tighter upper bound on what is achievable against a mean-based no-regret algorithm.

Theorem 10. *For any game G and any mean-based no-regret algorithm used by player B, there is no strategy for which the average reward of player A converges to $\mathsf{Val}_A(\emptyset, \mathcal{E}_{D(G)}) + \varepsilon$, for any $\varepsilon > 0$.*

Proof. First, we observe that mean-based algorithms will never play a dominated strategy $b_i \in D(G)$ in more than $o(T)$ rounds. As b_i is dominated, there is some $\delta > 0$ such that for every $\alpha \in \Delta(M)$, there is some b_j where $u_B(\alpha, b_j) \geq u_B(\alpha, b_i) + \delta$. Let α_t denote the empirical distribution of player A's actions up to time t. After some window of $O(\gamma T) = o(T)$ rounds we will have the cumulative rewards $\sigma_{i,t}$ and $\sigma_{j,t}$ satisfy $\sigma_{i,t} < \sigma_{j,t} - \delta t < \sigma_{j,t} - \gamma T$ under any α_t for some b_j in each subsequent round, and so b_i will never be played in more than $o(T)$ rounds.

We can also see that any such no-$\mathcal{E}$-regret algorithm is a no-$\mathcal{E}_{D(G)}$-regret algorithm. Suppose such an algorithm had $\mathcal{E}_{D(G)}$-regret ϵT, for $\epsilon > 0$; then, there is some g_{jk} for which $U_B(\alpha_T, g_{jk}(\beta_T)) \geq U_B(\alpha_T, \beta_T) + \epsilon$. By the $\mathcal{E}$-regret guarantee this cannot occur if $j = k$, as any such function g_{jj} is

equivalent to the fixed deviation rule for b_j. However, if this occurs for $j \neq k$, such an algorithm must have played dominated strategies in a total $\Omega(\epsilon T)$. This contradicts our assumption that no dominated strategy b_i is played in more than $o(T)$ rounds, and so any mean-based no-$\mathcal{E}$-regret algorithm is also a no-$\mathcal{E}_{D(G)}$-regret algorithm, against which player A cannot obtain average reward which converges to any amount higher than $\mathsf{Val}_A(\emptyset, \mathcal{E}_{D(G)}) + o(1)$. □

H Proof of Theorem 5

	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 $\mathsf{Val}_A(\emptyset, \mathcal{E}) > \mathsf{Val}_A(\emptyset, \mathcal{I}) = \mathsf{MBRew}_A$

Proof. Let 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 (by Theorem 10). 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 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:

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

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

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

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 $1 + o(1)$. □

I Proof of Theorem 6

Proof. We recall that the SU algorithm from (19) finds initial points $\alpha^*(b_i)$ in each best response region via random sampling, which takes takes $1/\operatorname{poly}(\varepsilon^{-1})$ queries in expectation. Then, upon calibrating for $O(\log(1/\varepsilon))$ bits of precision SU makes $\operatorname{poly}(M, N, \log(1/\varepsilon))$ queries, each of which can be taken to be a point on some grid of spacing $1/\operatorname{poly}(\varepsilon^{-1})$ within the simplex by the precision condition. The computed approximate Stackelberg strategy is then the optimal such point on the grid.

We first describe our strategy for simulating each query against an arbitrary anytime-no-regret learner; as $\mathcal{E} \subseteq \Phi$, we can restrict to considering only no-external-regret learners, as these regret constraints will always be satisfied. To implement a query q, greedily play the action whose historical frequency of play is the furthest below its target frequency in q. After $O(\operatorname{poly}(1/\varepsilon))$ rounds, the historical distribution will be within $1/\operatorname{poly}(\varepsilon^{-1})$ of q, and continuing the greedy selection strategy indefinitely will ensure that the history remains in a $1/\operatorname{poly}(\varepsilon^{-1})$-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 ball around q, provided that this ball is contained entirely inside some best response region R_j. For the sampling step, a taking sufficiently small grid (but still $1/\operatorname{poly}(\varepsilon^{-1})$) ensures that random sampling still suffices to find a point a point in each best response region even if our

queries may be adversarially perturbed to neighboring points on the grid, as each region is convex and has volume at least $1/\operatorname{poly}(\varepsilon^{-1})$. To address the issue for the line search steps, it suffices to take an additional step along each search conducted by SU before termination, where we then take each hyperplane boundary estimate to be one step inward along the grid from where our search terminates, maintaining a buffer between each hyperplane estimate in which all our points of uncertainty must lie. This adds at most a constant factor to our query complexity, and impacts our approximation by $1/\operatorname{poly}(\varepsilon^{-1})$, which then yields us a runtime of $\operatorname{poly}(1/\varepsilon)^Q$ rounds.

For the case of a no-adaptive-regret learner, suppose such an algorithm is calibrated for $T = O(Q^{C_1}(1/\varepsilon)^{C_2})$; then, over any window of length W its regret is at most $O\left((Q^{C_1}(1/\varepsilon)^{C_2})^c W^{-1}\right)$. Taking $W = \omega((Q^{C_1}(1/\varepsilon)^{C_2})^c)$ yields a per-round regret of at most $o(1)$ over the window, and so an algorithm must play a best response in $W - o(W)$ of the rounds. For sufficiently large C_1 and C_2, each W is large enough to yield the same precision we required for the anytime case, where now we greedily play the action whose frequency is furthest below its target *since our previous query terminated*, which allows us to again simulate the $O(Q)$ queries in $\operatorname{poly}(Q/\varepsilon)$ rounds (accounting for the robustness checks) while yielding $\Theta(WQ) = o(T)$. □

J 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 τ, 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 τ 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 τ 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 τ 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 τ 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 τ 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 3τ 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 9τ 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 (2) 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 $i := i + 1$.
end while

Lemma 6. *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 σ_{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 σ_{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 (5), 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 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_{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.

□