Learning Local Stackelberg Equilibria from Repeated Interactions with a Learning Agent

Nivasini Ananthakrishnan[1], Yuval Dagan[2], and Kunhe Yang[1]

[1]UC Berkeley, {nivasini,kunheyang}@berkeley.edu
[2]Tel Aviv University, ydagan@tauex.tau.ac.il

Abstract

Motivated by the question of how a principal can maximize its utility in repeated interactions with a learning agent, we study repeated games between an principal and an agent employing fictitious play. Prior work by Brown et al. (2024b) has shown that computing or even approximating the *global* Stackelberg value in similar settings can require an exponential number of rounds in the size of the agent's action space, making it computationally intractable. In contrast, we shift focus to the computation of *local* Stackelberg equilibria and introduce an algorithm that, within the smoothed analysis framework, constitutes a Polynomial Time Approximation Scheme (PTAS) for finding an ε-approximate local Stackelberg equilibrium. Notably, the algorithm's runtime is polynomial in the size of the agent's action space yet exponential in $(1/\varepsilon)$—a dependency we prove to be unavoidable.

1 Introduction

In repeated games, agents often have incomplete information about the game and resort to using a learning algorithm to optimize their utility over time. We consider a repeated game between a strategic player, referred to as the *principal*, and a player who follows a learning algorithm, referred to as the *agent*. When the principal anticipates the learning algorithm used by the agent, how should she adjust her strategy to maximize her utility over time? The answer depends on both the information available to the principal and the specific learning dynamics of the agent.

In this paper, we study the setting where the principal has no knowledge of the agent's utility function, and the agent employs a *mean-based* learning algorithm. A *mean-based* learner is a learning algorithm that selects its strategy based on the empirical distribution of the principal's past actions. Specifically, at each round, the learner plays an approximate best response to the time-averaged history of the principal's play. This class includes fundamental algorithms such as Fictitious Play, Follow the Regularized/Perturbed Leader, and Multiplicative Weights Update. In particular, we focus on Fictitious Play, where the agent *exactly* best responds to the empirical distribution of the principal's actions up to the current round.

Given that the agent is a mean-based learner, what should the principal aim to achieve? A natural goal is to devise a strategy that maximizes her cumulative utility over time, among all possible strategies. This objective has been considered in Mechanism Design (Braverman et al., 2018; Cai et al., 2023; Rubinstein and Zhao, 2024), Contract Design (Guruganesh et al., 2024), Information Design (Lin and Chen, 2024), and general games (Deng et al., 2019; Mansour et al., 2022; Brown et al., 2024b). However, even when the principal has full knowledge of the agent's utility function, computing the optimal long-term strategy is NP-hard in general games (Assos et al., 2024, 2025).

An alternative benchmark is the *Stackelberg equilibrium* of the one-shot game. If the principal knows the agent's utility function, she can compute a Stackelberg strategy in polynomial time and play it repeatedly. Since a mean-based learner best responds to the empirical distribution of the principal's past actions, this ensures that the agent selects the Stackelberg best response in each round, thereby approximately yielding the one-shot Stackelberg value to the principal in each round. When the principal does not have access to the agent's utility function, she could approximate the Stackelberg equilibrium using a *best-response oracle*—an oracle that, given a mixed strategy of the principal, returns the agent's best response Letchford et al. (2009); Peng et al. (2019); Blum et al. (2014). If the only information available about the agent comes from observing its decisions over time, the principal may attempt to influence the agent's learning process to induce best responses. This approach has been analyzed in various settings Haghtalab et al. (2022, 2024). However, for the fundamental class of mean-based learners, reducing the problem to learning via a best-response oracle introduces an exponential overhead. That is due to the fact that mean-based learners remember the whole history of play, hence they cannot be easily manipulated to produce best responses. In fact, Brown et al. (2024b) established that approximating the Stackelberg equilibrium in this setting requires exponential time.

1.1 Our contribution

The impossibility result of Brown et al. (2024b) raises the question of what can be learned from interactions with a mean-based learning agent when the optimizer lacks direct access to the agent's utility function. A key challenge arises from the fact that mean-based learners are slow to forget—they adjust their strategy based on the cumulative history of play rather than responding to individual queries. As a result, querying their best response to widely varying mixed strategies is infeasible, since shifting the empirical distribution of past plays requires many rounds. This makes traditional approaches based on best-response oracles impractical.

Given these constraints, the optimizer can only infer information from how the agent gradually adapts its strategy over time. Consequently, the optimizer's best option is to perform a local search by making small adjustments to its historical play—modifying the empirical distribution of past actions incrementally—and observing the agent's response. Through this process, the optimizer can progressively optimize its utility. A natural target for such an optimization process is an approximate *local equilibrium*, where the optimizer cannot significantly improve its utility through small deviations in its historical play. In this work, we study the problem of efficiently finding an ϵ-approximate local equilibrium, leveraging the structure of mean-based learning dynamics to design an algorithm for this setting.

We present a PTAS: an algorithm whose iteration complexity is exponential in $1/\epsilon$ but polynomial in the size of the game, with each iteration running in polynomial time. Furthermore, we prove that this exponential dependence on $1/\epsilon$ is unavoidable. To achieve our runtime guarantees, we impose a few natural assumptions to prevent adversarial instances and to ensures that the vectors in the learner's utility matrix are in general position.

Technical Contributions. Optimizing against a mean-based learner introduces several challenges:

- **Local versus Global Queries:** Prior work reduced the problem of finding an approximate Stackelberg equilibrium via interactions with a learner to a query-based approach: given an optimizer's mixed strategy $x \in \Delta(\mathcal{A})$, one can query the learner's (exact or approximate) best response $y \in \mathrm{BR}(x)$. However, for mean-based learners, queries are expensive if the mixed strategies differ substantially. To query $\mathrm{BR}(x)$ and then $\mathrm{BR}(x')$ when x and x' are far apart, the optimizer must play enough rounds to adjust the average history from one strategy to the other. Our key insight is to design an algorithm that only makes local changes—ensuring that successive queries remain close—thus circumventing the high cost associated with large shifts in the history.

- **Discontinuities in the Objective:** One might naturally attempt to apply standard local-search methods to our problem since our algorithm operates via local steps. However, the optimization target is not globally continuous: the learner's best response can change abruptly as the optimizer's mixed strategy crosses the boundaries between different regions of $\Delta(\mathcal{A})$. Within each such best-response region—which is always a polytope—the principal's utility is continuous, but the overall function is piecewise continuous. Our algorithm addresses this challenge by detecting the hyperplanes that separate these best-response polytopes and restricting the search to the regions that yield higher utility.
- **Searching across High-Dimensional Polytope Boundaries:** Our algorithm operates locally within polytopes, shifting from one to another upon reaching a local optimum within the current region. Once an $x \in \Delta(\mathcal{A})$ is found that locally maximizes the optimizer's utility in a best-response polytope, the algorithm must determine whether any neighboring polytopes offer a higher utility. Detecting all adjacent regions is challenging because these polytopes reside in $\mathbb{R}^{|\mathcal{A}|-1}$, and in high dimensions some may have volumes that are exponentially small relative to the dimension—even within a small neighborhood of x. We overcome this by sequentially detecting neighboring polytopes and, at each step, restricting the search to the intersection of all previously discovered regions. This gradual reduction in the search space's effective dimension avoids the pitfall of runtimes that scale inversely with the volume of the smallest polytope. In contrast to prior query-based algorithms for finding global Stackelberg equilibria—which may incur exponential runtimes due to this volume dependence—we demonstrate that focusing on local Stackelberg equilibria sidesteps this exponential barrier.

Related Work. Our work relates to many areas of research in learning the Stackelberg value and other game-theoretic benchmark in games. We provide a detailed discussion in Appendix A.

2 Model

In this paper, we consider a 2-player game between a *principal* and an *agent*. Let $\mathcal{A}$ be the action space of the principal and $\mathcal{B}$ be the action space of the agent. We assume that both action spaces are finite with size $|\mathcal{A}| = m$ and $|\mathcal{B}| = n$, respectively.

For a pair of pure strategies $(a, b) \in \mathcal{A} \times \mathcal{B}$, we use $U_1(a, b)$ and $U_2(a, b)$ to denote the utility of the principal and the agent, where we assume that all utilities are between $[0, 1]$. These utility functions can also be interpreted as matrices $U_1, U_2 \in [0, 1]^{m \times n}$.

When both players employ mixed strategies, i.e. distributions $\boldsymbol{x} \in \Delta(\mathcal{A})$ and $\boldsymbol{y} \in \Delta(\mathcal{B})$ over their action spaces, their expected utilities are given by

$$U_i(\boldsymbol{x}, \boldsymbol{y}) = \mathop{\mathbb{E}}_{a \sim \boldsymbol{x}, b \sim \boldsymbol{y}}[U_i(a, b)] = \boldsymbol{x}^{\mathsf{T}} U_i \ \boldsymbol{y}, \quad i \in \{1, 2\}.$$

Repeated Games. We consider repeated interactions between the principal and the agent, where the stage game (U_1, U_2) is repeated for T rounds. At the beginning of the interactions, both players know their own utility matrix U_i but do not know the utility matrix U_{-i} of their opponent.

At each round $t \in [T]$, the principal and agent simultaneously select strategies $\boldsymbol{x}^{(t)} \in \Delta(\mathcal{A})$ and $\boldsymbol{y}^{(t)} \in \Delta(\mathcal{B})$ respectively. Principal observes $\boldsymbol{y}^{(t)}$, and gains utility $U_1(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})$. Similarly, the agent observes $\boldsymbol{x}^{(t)}$, and gains utility $U_2(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})$

2.1 Principal's Benchmarks

To measure the principal's performance in repeated games, many previous works (e.g. (Haghtalab et al., 2022, 2024; Deng et al., 2019; Blum et al., 2014)) have focused on the Stackelberg value of the stage game, which we introduce below.

The Stackelberg Value. In the stage game (U_1, U_2), the Stackelberg value is a benchmark for the principal's optimal utility when she has full knowledge of the agent's utility function and the ability to commit to a strategy. Formally, it is defined as the solution to the following optimization problem:

$$\mathrm{StackVal} \triangleq \max_{\boldsymbol{x}^\star \in \Delta(\mathcal{A})} \max_{y^\star \in \mathsf{BR}(\boldsymbol{x}^\star)} U_1(\boldsymbol{x}^\star, y^\star),$$

where $\mathsf{BR} : \Delta(\mathcal{A}) \to 2^{\mathcal{B}}$ is the agent's best-response function that maps from a principal's mixed strategy to a set of pure strategies in $\mathcal{B}$ that maximizes the agent utility. Specifically, for all $\boldsymbol{x} \in \Delta(\mathcal{A})$,

$$\mathsf{BR}(\boldsymbol{x}) \triangleq \operatorname*{argmax}_{b \in \mathcal{B}} U_2(\boldsymbol{x}, y) = \{b \in \mathcal{B} : U_2(x, b) \geq U_2(x, y) \text{ for all } y \in \Delta(\mathcal{B})\}.$$

The Stackelberg value represents the global optimal utility that can be achieved against rational agents. However, computing it efficiently is intractable against mean-based agent, as shown by Brown et al. (2024b). In this paper, we study the *local Stackelberg equilibria*, which we show can be achieved efficiently.

Local Stackelberg Equilibria (LSE). A local Stackelberg strategy is one where no small local deviation can significantly improve the principal's utility when the agent best responds. This definition serves as the discrete analogue of the differential Stackelberg equilibria studied in (Fiez et al., 2020).

Definition 2.1 ((ε, δ)-Approximate Local Stackelberg Strategy)**.** *A principal's strategy $\boldsymbol{x} \in \Delta(\mathcal{A})$ is an (ε, δ)-Approximate Local Stackelberg Strategy if*

$$\forall \boldsymbol{x}' \in \mathbf{B}_1(\boldsymbol{x}; \delta), \qquad \sup_{y' \in \mathsf{BR}(\boldsymbol{x}')} U_1(\boldsymbol{x}', y') \leq \sup_{y \in \mathsf{BR}(\boldsymbol{x})} U_1(\boldsymbol{x}, y) + \varepsilon\delta,$$

where $\mathbf{B}_1(\boldsymbol{x}, \delta)$ denote the ℓ_1 ball of radius δ around $\boldsymbol{x}$, i.e., the set of strategies with $\|\boldsymbol{x}' - \boldsymbol{x}\|_1 \leq \delta$.

We remark that in settings with smooth utility functions, approximate local optima are often characterized by small gradients. However, when the principal's utility function is non-continuous due to the agent's best-response behavior, a gradient-based characterization is no longer appropriate. Instead, our definition provides a discrete analogue that captures the same intuition—ensuring that small perturbations in the principal's strategy do not lead to significantly higher payoffs.

We include a comparison of the local Stackelberg benchmark to other benchmarks in Appendix E. In general, the local Stackelberg is a weaker benchmark compared to the global Stackelberg, albeit a tractable one. However, there are special cases, where the local Stackelberg is equivalent to the the global Stackelberg in terms of the principal's utility. This is the case in a broad and commonly studied class of Stackelberg games — *Stackelberg security games* as we show in Proposition E.3.

2.2 Agent's algorithm

In this paper, we assume that the agent selects their actions according the a learning algorithm termed *Fictitious Play*, defined as follows:

Definition 2.2 (Fictitious play). *An agent follows the fictitious play algorithm if, at each round t, it chooses a strategy from the set of best responses to the principal's average strategy during the first $t-1$ rounds. Formally, let the principal's average past strategy be $\overline{\boldsymbol{x}}^{(t-1)} = \sum_{s=1}^{t-1} x^{(s)}/(t-1)$, then the agent's strategy at round t satisfies $y^{(t)} \in \mathsf{BR}(\overline{\boldsymbol{x}}^{(t-1)})$.*

In the above definition, the pure strategy $y^{(t)} \in \mathcal{B}$ can represent either a determinsitic action (where the agent plays a pure strategy), or a randomized action drawn from a mixed strategy $y^{(t)} \sim \boldsymbol{y}^{(t)} \in \Delta(\mathcal{B})$. In the latter case, the support of the mixed strategy $\boldsymbol{y}^{(t)}$ must be fully contained in the best response set $\mathsf{BR}(\overline{\boldsymbol{x}}^{(t-1)})$. We focus on fictitious play agents for the following two reasons:

- Fictitious play agents exhibit a property of slowly forgetting past interactions, as they update their strategy based on the average of the principal's past strategies. This introduces technical challenges that differ significantly from interactions with myopic best-responding agents who make decisions only based on recent rounds without any memory.
- Fictitious play serves as a building block for understanding the interactions with a broader class of *mean-based* agents (Braverman et al., 2018), which includes many widely-used learning algorithms, such as Multiplicative Weights Update, Follow the Regularized/Perturbed Leader and ε-greedy.

2.3 Geometric Interpretations of the Principal's Optimization Problem

In this section, we introduce some additional notations and revisit the geometric properties of the principal's optimization problem that will be useful in our algorithms.

For each of the agent's actions $b \in \mathcal{B}$, define the *best response polytope* to action b as the set of principal's mixed strategies that induce best response b, which we denote with $\mathrm{P}_b \triangleq \{\boldsymbol{x} \in \Delta(\mathcal{A}) \mid b \in \mathsf{BR}(\boldsymbol{x})\}$. These subsets are referred to as polytopes because they are characterized by the intersection of halfspaces:

$$\mathrm{P}_b = \left\{\boldsymbol{x} \in \Delta(\mathcal{A}) \mid \forall b' \in \mathcal{B},\ U_2(\boldsymbol{x}, b) - U_2(\boldsymbol{x}, b') = \langle \boldsymbol{u}_b - \boldsymbol{u}_{b'}, \boldsymbol{x}\rangle \geq 0\right\},$$

where $\boldsymbol{u}_b \in \mathbb{R}^m$ represents the agent's utility vector conditioned on action b, i.e., $\boldsymbol{u}_b = U_2(\cdot, b)$. We will also use $\boldsymbol{h}_{b,b'}$ to denote the hyperplane that separates polytopes P_b and $\mathrm{P}_{b'}$, i.e., $\boldsymbol{h}_{b,b'} \triangleq \boldsymbol{u}_b - \boldsymbol{u}_{b'}$.

Principal's Optimization Problem. Note that under full information about the polytope partition, the principal's Stackelberg value can be equivalently described as the optimal solution to the following piece-wise linear function: $\text{StackVal} = \max_{b\in\mathcal{B}} \max_{\boldsymbol{x}\in\mathrm{P}_b} U_1(\boldsymbol{x}, b)$, which involves maximization over all the polytopes P_b. On the other hand, in *local Stackelberg equilibria*, a strategy $\boldsymbol{x}$ is locally optimal if no surrounding polytopes achieve a higher principal utility. We define the notion of (approximately) surrounding polytopes below.

Surrounding Polytopes For a principal's strategy $\boldsymbol{x} \in \Delta(\mathcal{A})$, let $\mathcal{P}(\boldsymbol{x})$ denote the subset of polytopes that contains (surrounds) $\boldsymbol{x}$: $\mathcal{P}(\boldsymbol{x}) \triangleq \{\mathrm{P}_b \mid \boldsymbol{x} \in \mathrm{P}_b\}$. Note that $\mathcal{P}(\boldsymbol{x})$ equivalently contains all polytopes P_b for which $b \in \mathsf{BR}(\boldsymbol{x})$ is a best response.

For a radius $\varepsilon > 0$, let $\mathcal{P}_\varepsilon(\boldsymbol{x})$ be the polytopes that are within ℓ_2 distance ε to $\boldsymbol{x}$:

$$\mathcal{P}_\varepsilon(\boldsymbol{x}) \triangleq \{\mathrm{P}_b \mid \mathrm{dist}_2(\boldsymbol{x}, \mathrm{P}_b) \leq \varepsilon\}. \qquad (\varepsilon\text{-Surrounding Polytopes})$$

Here, the distance between vector $\boldsymbol{x}$ and set P_b is defined as $\mathrm{dist}_2(\boldsymbol{x}, \mathrm{P}_b) \triangleq \min_{\boldsymbol{x}' \in \mathrm{P}_b} \|\boldsymbol{x} - \boldsymbol{x}'\|_2$.

2.4 Our Assumptions

In this section, we briefly describe the assumptions we make to derive our results. We provide a more detailed discussion in Appendix F.

Our first assumption is a standard assumption in previous work on approximating the Stackelberg value Blum et al. (2014); Letchford et al. (2009); Haghtalab et al. (2022, 2024).

Assumption 2.3 (Distance from Polytope Boundaries)**.** *For every polytope* P_b*, there exists some strategy* $\boldsymbol{x} \in \mathrm{P}_b$ *such that all coordinates of* $\boldsymbol{x}$ *are lower bounded by* $R_{\min}$*, i.e.,* $\boldsymbol{x} \geq R_{\min} \cdot \mathbf{1}$*.*

Assumption F.3 is an arguably weaker condition than the assumption made in previous works as discussed in Remark F.4. One reason is previous works Blum et al. (2014); Letchford et al. (2009); Haghtalab et al. (2022, 2024) usually incur a dependency on the volume of the ball, which can be exponential in the dimension (i.e. the number of principal's actions). In contrast, our bound only has a polynomial dependence on $1/R_{\min}$.

The next assumption is that the polytopes and hyperplanes are sufficiently separated from each other. This assumption is also connected to an assumption on the minimum singular value of constraint matrices defining the polytopes (Lemma F.5), which we denote by $\underline{\sigma}$. The assumption and connection to $\underline{\sigma}$ is stated formally in Lemma F.5. Here we state it informally as follows:

Assumption 2.4 (Informally: Polytopes/Hyperplanes are Far Apart)**.**

1. *For any* $\boldsymbol{x} \in \Delta(\mathcal{A})$*, there are at most* m *polytopes that have distance at most* R_1 *to* $\boldsymbol{x}$*.*
2. *For all* $b \in \mathcal{B}$ *and all* $\boldsymbol{x} \in \mathrm{P}_b$*, there are at most* $m-1$ *hyperplanes from that polytope* P_b *that* $\boldsymbol{x}$ *can be close to.*

The second assumption above is satisfied with high probability under the smoothed analysis framework i.e., when the utility functions are perturbed by Gaussian noise. We show this in Appendix D.

3 Algorithm

In this section, we present our main algorithm (Algorithm 1) that finds an approximate Local Stackelberg equilibrium. Algorithm 1 alternates between the following two main subroutines:

- OptimizeWithinPolytope (Algorithm 2). At a high level, this subroutines finds an approximately optimal policy constrained to a given polytope.
- SearchForPolytopes (Algorithm 3). When the principal's search point is near a polytope boundary, this subroutine identifies all adjacent polytopes in the neighborhood.

Now we discuss the high-level ideas for the two subroutines respectively.

(1) Optimizing within polytopes (More details in Section 3.1). In this subroutine, the principal aims to compute an $\varepsilon\delta$-optimal strategy within the current polytope. This problem would be a linear program if the polytope were fully known. However, because the polytope depends on the agent's private utilities (which are unknown to the principal), the principal must learn this structure through interactions. To do so, the principal maintains an *approximate* version of the polytope, which is iteratively refined as constraints become known.

The approximation to the polytope is learned incrementally by adding new constraints as new polytopes are encountered. After each update of the constraints, the principal chooses the optimal strategy among strategies satisfying the current set of constraints. In doing so, the principal either improves utility or discovers a new constraint that refines the polytope approximation and prompts a re-optimization. (As shown in Figure 2a.)

(2) Searching for new polytopes (More details in Section 3.2). The SearchForPolytopes subrountine explores all polytopes within a certain radius around a given point. A key challenge in this search is that some surrounding polytopes may have volume that is exponentially small in the number of dimensions, requiring exponential in dimensions number of samples to identify through random sampling. To efficiently discover these polytopes, SearchForPolytopes employs an iterative dimension reduction approach. After a subset of polytopes are found, the algorithm restricts its search to the boundaries these polytopes. This restriction reduces the dimension of the search space and overcomes the challenge of searching in a high dimensional space. (See Figure 2b for an illustration.)

ALGORITHM 1: Local Stackelberg Equilibrium

```
Input: $x_{\text{start}}, \varepsilon, \delta, \alpha, \gamma$
Result: An $(\varepsilon, \delta)$-approximate Local Stackelberg equilibrium
// Iterating through the best-response polytopes
for $i = 1, \ldots, n$ do
    Current average strategy: $\overline{\boldsymbol{x}}$ ;
    Current agent best-response: $b_i \leftarrow \mathsf{BR}(\overline{\boldsymbol{x}})$ ;
    Update average strategy to $x_i^* \leftarrow \mathsf{OptimizeWithinPolytope}(\mathsf{BR}(\overline{\boldsymbol{x}}), \varepsilon, \delta, \alpha, \gamma)$;
    Move average strategy $\boldsymbol{x}_i^*$ to be $\delta$ close enough to the boundary ;
    Find all surrounding polytopes using $\mathsf{SearchForPolytopes}(x_i^*)$;
    if There is a surrounding polytope $\mathrm{P}_b$ with $U_1(x_i^*, b) \geq U_1(x_i^*, b_i) + \varepsilon_2$
     then
        Step into polytope $b$ and continue to the next iteration;
    end
    else
        Return $(x_i^*, \mathsf{BR}(x_i^*))$ is an $(\varepsilon, \delta)$-approximate LSE.
    end
end
```

Theorem 3.1 (Main theorem)**.** *Under Assumptions F.2 and F.3, with high probability, Algorithm 1 finds an* (ε, δ)*-approximate Local Stackelberg equilibrium within the following number of iterations:*

$$O\left(poly\left(m, n, \frac{1}{R_{\min}}, \log\frac{1}{\underline{\sigma}}\right) \cdot \exp\left(\frac{1}{\varepsilon\delta}\right)\right)$$

Although the number of iterations needed for Algorithm 1 to approximate the local Stackelberg equilibrium is exponential in $1/(\varepsilon\delta)$, we show in the following theorem that this exponential dependence is unavoidable. The proof of Theorem C.1 is deferred to Appendix J.5.

Theorem 3.2. *Any algorithm that finds an ϵ-approximate LSE requires $\Omega(\exp(1/\epsilon))$ steps in the worst case, even when the algorithm has full knowledge of the agent's utility function.*

Proof sketch of Theorem 3.1. In this proof sketch we will illustrate how we bound the number of rounds the algorithm takes. In the next sections we will analyze the subroutines of the algorithm in more detail and analyze their correctness. The correctness of the algorithm will follow from the correctness of these subroutines.

The main argument to bound the number of rounds our algorithm takes to find an (ε, δ)-approximate local Stackelberg strategy is that in each round where we improve utility, the principal's utility improves by at least some minimum amount. The amount of utility improvement has to necessarily depend on the round t since the maximum possible change in magnitude of utility at round t is $1/t$, as shown in Remark G.1. This follows from how the average strategy of the principal changes with the choice of strategies in each round. (We provide details on how our algorithm chooses strategies to change the average strategy in Appendix G.) We first suppose each improvement to the utility the algorithm makes has magnitude at least Δ/t, and show that our algorithm terminates within a bounded number of rounds leveraging the fact that the total possible improvement of utility is bounded. We will prove a lower bound on Δ at the end of the proof.

The argument is similar to the convergence analysis of algorithms such as gradient descent in continuous settings to local optima. However, a key challenge in our setting is that not every step is an improvement due to the discontinuous (piecewise) nature of the principal's utility functions. To deal with this discountinuity, our algorithm also takes non-improvement steps (such as the steps for constructing polytope boundaries and SearchForPolytopes).

Suppose the set of improvement rounds is T_I and the set of other rounds is T_N. We establish an upper bound on $|T_N|$ which allows us to still upper bound the iteration complexity of approximating the local Stackelberg strategy since each non-improvement step decreases utility by at most $\frac{1}{t}$. The total utility improvement is at least

$$\sum_{t \in T_{\mathrm{I}}} \frac{\varepsilon\delta}{t} - \sum_{t \in T_{\mathrm{N}}} \frac{1}{t} \geq \sum_{t=|T_{\mathrm{N}}|}^{|T_{\mathrm{N}}|+|T_{\mathrm{I}}|} \frac{\Delta}{t} - \sum_{t=1}^{|T_{\mathrm{N}}|} \frac{1}{t} \gtrsim \Delta \log \frac{|T_{\mathrm{N}}| + |T_{\mathrm{I}}|}{|T_{\mathrm{N}}|^2}.$$

Since utilities are bounded in $[0, 1]$,

$$\implies \Delta \log \tfrac{|T_{\mathrm{N}}|+|T_{\mathrm{I}}|}{|T_{\mathrm{N}}|^2} \leq 1, \qquad |T_{\mathrm{N}}| + |T_{\mathrm{I}}| \leq |T_{\mathrm{N}}|^2 + \exp\left(\frac{1}{\Delta}\right).$$

To complete the bound on the total number of rounds, it suffices to find a lower bound on the minimum utility improvement quantity Δ and an upper bound on the number of non-improving rounds $|T_{\mathrm{N}}|$. We will show in Section 3.1 that $\Delta \geq \varepsilon\delta$, in the analysis of OptimizeWithinPolytope (the subroutine where utility improvement steps are made).

Non-improvement rounds occur to construct boundary hyperplanes when an intended improvement step takes us outside the polytope. They occur at most n^2 times, once to construct easy hyperplane. And each round includes steps to get close to the boundary via BinarySearch and generating search vectors to explore nearby polytopes in SearchForPolytopes. The number of such steps $|T_{\mathrm{N}}|$ is analyzed in the analysis of the sub-routines and will turn out to be $\mathrm{poly}(m, n) \exp(1/(\varepsilon\delta))$. This results in the bound of the theorem. □

ALGORITHM 2: OptimizeWithinPolytope

```
Input: The current polytope $b$, starting point $\boldsymbol{x}_{\text{start}}$,
       approximation parameter $\alpha$
The estimated polytope $\hat{\mathrm{P}}_b \leftarrow \Delta(\mathcal{A})$;
while $\hat{\mathrm{P}}_b$ is not empty do
    $\boldsymbol{x}_{\text{target}} \leftarrow \operatorname{argmax}_{\boldsymbol{x} \in \hat{\mathrm{P}}_b} U_1(\boldsymbol{x}, b)$;
    if $U_1(\boldsymbol{x}_{target}, b) < U_1(\boldsymbol{x}_{start}, b) + \varepsilon\delta$ then
        Terminate and return $x_{\text{start}}$;
    end
    $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}_{\text{target}}$ // This is an improvement step
    if $y^{(t)} = b' \neq b$ then
        $\boldsymbol{x}, \boldsymbol{x}' \leftarrow$ BinarySearch between $\bar{\boldsymbol{x}}^{(t-1)} \in \mathrm{P}_b$ and
          $\bar{\boldsymbol{x}}^{(t)} \in \mathrm{P}_{b'}$ with accuracy $\alpha$;
        Use $\boldsymbol{x}, \boldsymbol{x}'$ to find approximate hyperplane $\hat{\boldsymbol{h}}_{b,b'}$ by
          calling SearchForPolytopes;
        $\hat{\mathrm{P}}_b \leftarrow \hat{\mathrm{P}}_b \cap \left\{ x : \hat{\boldsymbol{h}}_{b,b'}^{\mathsf{T}} \boldsymbol{x} \geq \alpha \right\}$;
        $x_{\text{start}} \leftarrow$ projection of $x_{\text{start}}$ onto $S$;
    end
end
```

ALGORITHM 3: SearchForPolytopes

```
Input: Starting point $\boldsymbol{x}^\star$,
       accuracy level $\alpha$
$\hat{\mathcal{S}} \leftarrow \{\boldsymbol{x} : \mathbf{1}^{\mathsf{T}} \boldsymbol{x} = 1\}$ (the
  constrained search space);
$L \leftarrow \{\}$ (the set of hyperplanes
  and polytopes discovered so far);
while $\hat{\mathcal{S}}$ is not empty do
    $\boldsymbol{h}, b \leftarrow$
      FindAHyperplane($\boldsymbol{x}^\star, \hat{\mathcal{S}}, \alpha$);
    if $\boldsymbol{h}, b$ is None then
        Return $L$
    end
    $\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \cap \{x : \langle \boldsymbol{h}, \boldsymbol{x} \rangle = \alpha\}$;
    $L \leftarrow L \cup \{(\boldsymbol{h}, b)\}$ ;
    $\boldsymbol{x}^\star \leftarrow$
      projection of $\boldsymbol{x}^\star$ onto $\hat{\mathcal{S}}$;
end
```

Figure 1: Key algorithms for local Stackelberg equilibria computation: (left) OptimizeWithinPolytope algorithm for optimizing principal's utility within a polytope; (right) SearchForPolytopes algorithm for efficiently discovering polytopes in high-dimensional spaces.

3.1 Analysis of OptimizeWithinPolytope

In this section we will analyze the correctness of the subroutine OptimizeWithinPolytope. Before we delve into the analysis of OptimizeWithinPolytope, let us describe its implementation in some more detail.

When optimizing within a polytope P_b corresponding to action $b \in \mathcal{B}$, OptimizeWithinPolytope maintains an estimate polytope $\hat{\mathrm{P}}_b$ of the true polytope P_b. $\hat{\mathrm{P}}_b$ starts of being the space of all strategies and gets refined as the principal learns more about P_b. The algorithm alters between *improvement* steps and *searching* steps:

- In *improvement* steps, we move the average strategy $\overline{\boldsymbol{x}}^{(t)}$ toward the optimizer $\boldsymbol{z} = \operatorname{argmax}_{\boldsymbol{x} \in \hat{\mathrm{P}}_b} U_1(\boldsymbol{x}, b)$, which is the optimizer in $\hat{\mathrm{P}}_b$. We move toward z only when utility improves significantly. This guarantees termination via the termination condition
$$U_1(\boldsymbol{z}, b) \leq U_1(\overline{\boldsymbol{x}}^{(t)}, b) + \varepsilon\delta. \qquad \text{(Termination condition)}$$
z might not actually be in P_b and may lead to an average strategy $\overline{\boldsymbol{x}}^{(t)}$ outside of P_b that can be detected by $y^{(t)} \neq b$ as shown in Figure 2a. When this happens, we switch to *searching* steps and identify a new direction of improvement if one exists as shown in Figure 2a.
- We enter a phase of *searching* steps with two consecutive points $\overline{\boldsymbol{x}}^{(t-1)}$ and $\overline{\boldsymbol{x}}^{(t)}$ that lie in opposite sides of some boundary $(\boldsymbol{h})$ of P_b. We approximate $\boldsymbol{h}$ by $\hat{\boldsymbol{h}}$ satisfying $\|\hat{\boldsymbol{h}} - \boldsymbol{h}\|_2 \leq \alpha$.

  We do this by backtracking from $\overline{\boldsymbol{x}}^{(t)}$ to $\overline{\boldsymbol{x}}^{(t-1)}$ and finding a point in between that is α close to the boundary. We search the polytopes surrounding this point via SearchForPolytopes to construct $\hat{\boldsymbol{h}}$.

This implementation of OptimizeWithinPolytope approximates the optimal strategy within the polytope of search as stated below.

Proposition 3.3 (Correctness of OptimizeWithinPolytope)**.** *Algorithm 2 takes an agent-action $b \in \mathcal{B}$ and a starting point in the best-response polytope P_b, and finds a strategy $\boldsymbol{x}_b^\star \in \mathrm{P}_b$ that achieves $\varepsilon\delta$-optimal principal's utility in the polytope, i.e., $U_1(\boldsymbol{x}_b^\star, b) \geq \max_{\boldsymbol{x}_b \in \mathrm{P}_b} U_1(\boldsymbol{x}_b, b) - \varepsilon\delta$.*

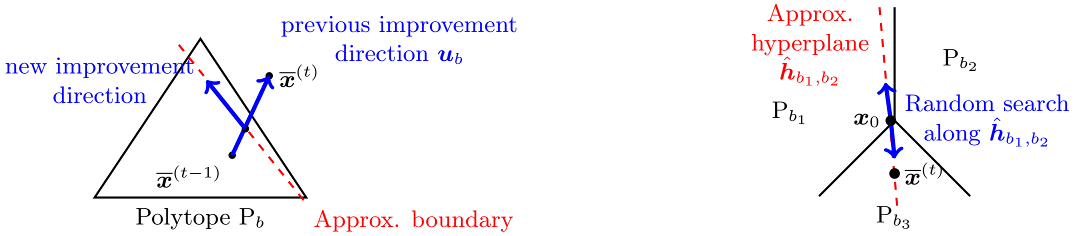
Proof sketch. The correctness of OptimizeWithinPolytope is due to the way we construct the estimated polytope $\hat{\mathrm{P}}_b$. The first property is that $\hat{\mathrm{P}}_b$ is defined by approximations to a subset of the constraints defining P_b. As a result, if no strategy in $\hat{\mathrm{P}}_b$ significantly improves utility (over $\varepsilon\delta$ amount), then there is also no strategy in $\hat{\mathrm{P}}_b$ improving utility beyond $\varepsilon\delta$. The second property is the accuracy to which we approximate the constraints of P_b that we discover. We achieve this accuracy by using BinarySearch to get close to the boundary of P_b and using SearchForPolytopes to discover all boundaries nearby. □

Beyond the correctness of OptimizeWithinPolytope, there are two other important properties that we discuss below. These properties are regarding the minimum improvement in every improvement step and the number of non-improvement steps. Both of these quantities are important for bounding the total number of rounds the main algorithm uses to find the approximate local Stackelberg equilibrium.

In each improvement step, utility increases by at least $\varepsilon\delta/t$ as shown in Lemma H.2. This is due to the choice of termination condition which implies that each improvement occurs in a direction with at least $\varepsilon\delta$ utility improvement possible.

All non-improvement steps occur in the searching phase are the rounds in sub-routines BinarySearch and SearchForPolytopes. These steps include back-tracking toward previous average strategies. This is made efficient by our algorithm's restriction of selecting strategies in $\Delta(\mathcal{A})$ that are at least at ℓ_1 distance of γ away from the boundary of the simplex. Given this restriction, changing the average strategy at round t from $\bar{\boldsymbol{x}}$ to $\bar{\boldsymbol{x}}'$ can be done in $\|\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}'\|_1 t/\gamma$ rounds.

3.2 Analyzing SearchForPolytopes Searching for polytopes


new improvement
direction
previous improvement
direction $\boldsymbol{u}_b$
$\overline{\boldsymbol{x}}^{(t)}$
$\overline{\boldsymbol{x}}^{(t-1)}$
Polytope P_b
Approx. boundary
Approx.
hyperplane
$\hat{\boldsymbol{h}}_{b_1,b_2}$
P_{b_2}
P_{b_1}
$\boldsymbol{x}_0$
Random search
along $\hat{\boldsymbol{h}}_{b_1,b_2}$
$\overline{\boldsymbol{x}}^{(t)}$
P_{b_3}

(a) Optimization within polytopes. The figure illustrates how improvement directions are updated in OptimizeWithinPolytope after adding new constraints. If $\overline{\boldsymbol{x}}^{(t)}$ falls outside of the true polytope P_b, we first find an approximate boundary (the dashed line), and update the improvement direction subject to the new constraint.

(b) Search for polytopes. The figure illustrates the SearchForPolytopes procedure. P_{b_1} and P_{b_2} are the discovered polytopes. When performing random search along the approximate hyperplane $\hat{\boldsymbol{h}}_{b_1,b_2}$, the search point $\overline{\boldsymbol{x}}^{(t)}$ falls in P_{b_3} with constant probability. When this happens, we discover a new polytope P_{b_3}.

Figure 2: Algorithmic components for computing local Stackelberg equilibria

The SearchForPolytopes algorithm finds all the polytopes that are within a distance of at most ρ from a given point $\boldsymbol{x} \in \Delta(\mathcal{A})$. For illustration, consider the example in Figure 2b. If the input point $\boldsymbol{x}$ is within ρ distance to the intersection $\boldsymbol{x}_0$, then SearchForPolytopes would return all the surrounding polytopes $\mathrm{P}_{b_1}, \mathrm{P}_{b_2}, \mathrm{P}_{b_3}$. This guarantee is formally stated in the following theorem.

Theorem 3.4 (Correctness of SearchForPolytopes)**.** *Starting from a point $\boldsymbol{x}^*$ in P_b, for any $\alpha \in (0, o(R_2\underline{\sigma}/m^3))$ and $\rho < \alpha$,* SearchForPolytopes *finds $\hat{\boldsymbol{h}}_{b,b'}$ such that $\|\hat{\boldsymbol{h}}_{b,b'} - h_{b,b'}\|_2 \leq \alpha$, for every $b' \in \mathcal{P}_\rho(\boldsymbol{x}^*)$.*

Remark 3.5. *Our algorithm applies* SearchForPolytopes *in two contexts. The first is to find separating hyperplanes in* OptimizeWithinPolytope *and the second is to find all polytopes within a δ radius of a point to certify it as a Local Stackelberg equilibrium or find evidence against this.*

In the first context, we require α-approximation of the separating hyperplane. Since the guarantee of the theorem holds for $\rho < \alpha$, we need to get within α close to the boundary so that the separating hyperplane lies in the region that SearchForPolytopes *explores. In the second context, we apply the theorem for the choice of $\rho = \delta$.*

As previously mentioned, the main challenge that SearchForPolytopes overcomes is searching all polytopes efficiently including ones with exponentially small volume are difficult to find through random search. SearchForPolytopes still performs random search, but within a restricted space of a lower dimension. The restricted space is the intersections of all the boundary hyperplanes discovered up to the point of the search.

That is, suppose boundary hyperplanes $\boldsymbol{h}_j$ for $j \in J$ are encountered and are approximated as $\hat{\boldsymbol{h}}_j$ for each $j \in [J]$. Then in the next round of search, SearchForPolytopes conducts a Gaussian random search within the subspace $\hat{\mathcal{S}}_J := \{\boldsymbol{x} \in \Delta(\mathcal{A}) \mid \hat{\boldsymbol{h}}_J \boldsymbol{x} = \boldsymbol{0}\}$, which has dimension $m - |J|$ as a consequence of Assumption F.2

Assumption 2.4 ensures that all the random directions generated for search do indeed fall in true surrounding polytopes and not in any other polytope. This is because by the assumption, all other polytopes are sufficiently far enough. Hence we don't falsely discover polytopes other than the surrounding polytopes.

References

Jacob D Abernethy, Rachel Cummings, Bhuvesh Kumar, Sam Taggart, and Jamie H Morgenstern. 2019. Learning auctions with robust incentive guarantees. *Advances in Neural Information Processing Systems* 32 (2019).

Saba Ahmadi, Avrim Blum, and Kunhe Yang. 2023. Fundamental Bounds on Online Strategic Classification. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*. 22–58.

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. 2013. Learning prices for repeated auctions with strategic buyers. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 26.

Bo An, David Kempe, Christopher Kiekintveld, Eric Shieh, Satinder Singh, Milind Tambe, and Yevgeniy Vorobeychik. 2012. Security games with limited surveillance. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 26. 1241–1248.

Nivasini Ananthakrishnan, Nika Haghtalab, Chara Podimata, and Kunhe Yang. 2024. Is Knowledge Power? On the (Im) possibility of Learning from Strategic Interaction. *arXiv preprint arXiv:2408.08272* (2024).

Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. 2024a. Learning to Play Against Unknown Opponents. *arXiv preprint arXiv:2412.18297* (2024).

Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. 2024b. Pareto-Optimal Algorithms for Learning in Games. *arXiv preprint arXiv:2402.09549* (2024).

Angelos Assos, Yuval Dagan, and Constantinos Costis Daskalakis. 2024. Maximizing utility in multi-agent environments by anticipating the behavior of other learners. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=0uGlKYS7a2

Angelos Assos, Yuval Dagan, and Nived Rajaraman. 2025. Computational Intractability of Strategizing against Online Learners. (2025).

Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. 2015. Commitment without regrets: Online learning in Stackelberg security games. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*. 61–78.

Omar Besbes and Assaf Zeevi. 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations research* 57, 6 (2009), 1407–1420.

Avrim Blum, Nika Haghtalab, and Ariel Procaccia. 2014. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 27. 1826–1834.

Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. 2018. Selling to a no-regret buyer. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*. 523–538.

William Brown, Christos Papadimitriou, and Tim Roughgarden. 2024a. Online Stackelberg Optimization via Nonlinear Control. In *The Thirty Seventh Annual Conference on Learning Theory*. PMLR, 697–749.

William Brown, Jon Schneider, and Kiran Vodrahalli. 2024b. Is Learning in Games Good for the Learners? *Advances in Neural Information Processing Systems (NeurIPS)* 36 (2024).

Linda Cai, S Matthew Weinberg, Evan Wildenhain, and Shirley Zhang. 2023. Selling to multiple no-regret buyers. In *International Conference on Web and Internet Economics*. Springer, 113–129.

Yiling Chen, Yang Liu, and Chara Podimata. 2020. Learning strategy-aware linear classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 15265–15276.

Yuan Deng, Jon Schneider, and Balasubramanian Sivan. 2019. Strategizing against no-regret learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32. 1579–1587.

Kate Donahue, Nicole Immorlica, Meena Jagadeesan, Brendan Lucier, and Aleksandrs Slivkins. 2024. Impact of Decentralized Learning on Player Utilities in Stackelberg Games. *arXiv preprint arXiv:2403.00188* (2024).

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. 2018. Strategic classification from revealed preferences. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*. 55–70.

Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. 2020. Implicit learning dynamics in Stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning (ICML)*. PMLR, 3133–3144.

Abraham D. Flaxman, Adam Tauman Kalai, and H. B. McMahan. 2004. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Alexander Gasnikov, Darina Dvinskikh, Pavel Dvurechensky, Eduard Gorbunov, Aleksandr Beznosikov, and Alexander Lobanov. 2023. Randomized gradient-free methods in convex optimization. In *Encyclopedia of Optimization*. Springer, 1–15.

Guru Guruganesh, Yoav Kolumbus, Jon Schneider, Inbal Talgam-Cohen, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Joshua R Wang, and S Matthew Weinberg. 2024. Contracting with a learning agent. *arXiv preprint arXiv:2401.16198* (2024).

Nika Haghtalab, Thodoris Lykouris, Sloan Nietert, and Alexander Wei. 2022. Learning in stackelberg games with non-myopic agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 917–918.

Nika Haghtalab, Chara Podimata, and Kunhe Yang. 2024. Calibrated stackelberg games: Learning optimal commitments against calibrated agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. 2009. Computing optimal randomized resource allocations for massive security games. (2009).

Robert Kleinberg and Tom Leighton. 2003. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 594–605.

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. 2009. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory*. Springer, 250–262.

Tao Lin and Yiling Chen. 2024. Persuading a learning agent. *arXiv preprint arXiv:2402.09721* (2024).

Jinyan Liu, Zhiyi Huang, and Xiangning Wang. 2018. Learning optimal reserve price against non-myopic bidders. *Advances in Neural Information Processing Systems* 31 (2018).

Chinmay Maheshwari and Eric Mazumdar. 2023. Convergent first-order methods for bi-level optimization and stackelberg games. *arXiv preprint arXiv:2302.01421* (2023).

Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. 2022. Strategizing against Learners in Bayesian Games. In *Conference on Learning Theory (COLT)*. PMLR, 5221–5252.

Mehryar Mohri and Andres Munoz. 2014. Optimal regret minimization in posted-price auctions with strategic buyers. *Advances in Neural Information Processing Systems* 27 (2014).

Yurii Nesterov and Vladimir Spokoiny. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics* 17, 2 (2017), 527–566.

Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. 2019. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 2149–2156.

Aviad Rubinstein and Junyao Zhao. 2024. Strategizing against No-Regret Learners in First-Price Auctions. *arXiv preprint arXiv:2402.08637* (2024).

Arvind Sankar, Daniel A Spielman, and Shang-Hua Teng. 2006. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM J. Matrix Anal. Appl.* 28, 2 (2006), 446–476.

Daniel A Spielman and Shang-Hua Teng. 2004. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)* 51, 3 (2004), 385–463.

Bernhard Von Stengel and Shmuel Zamir. 2010. Leadership games with convex strategy sets. *Games and Economic Behavior* 69, 2 (2010), 446–457.

Appendix Table of Contents

A Related Work

While the Stackelberg equilibrium can be computed in polynomial time given full information about the game via a reduction to linear programming, learning the Stackelberg equilibrium from a best-response oracle requires additional assumptions. This challenge has been analyzed in general games (Letchford et al., 2009) and in specific games such assecurity games (Letchford et al., 2009; Blum et al., 2014; Peng et al., 2019; Balcan et al., 2015), demand learning (Kleinberg and Leighton, 2003; Besbes and Zeevi, 2009) and strategic classification (Dong et al., 2018; Chen et al., 2020; Ahmadi et al., 2023).

Learning the Stackelberg equilibrium through interactions with a learning agent that can (relatively) quickly forget has been studied by reducing the problem to a query-based algorithm with access to an approximate best-response oracle. This approach has been applied to interactions with non-myopic agents in auction design (Amin et al., 2013; Mohri and Munoz, 2014; Liu et al., 2018; Abernethy et al., 2019) and in general games (Haghtalab et al., 2022) and to interactions with adaptively calibrated agents (Haghtalab et al., 2024). In contrast, when interacting with mean-based learners that do not forget quickly, both exponential lower and upper bounds have been established on the iteration complexity of approximating the Stackelberg value (Brown et al., 2024b).

Beyond the Stackelberg value, alternative benchmarks have been explored. Prior work has investigated the problem of finding a sequence of actions for the optimizer that maximizes its utility when interacting with a learning agent—potentially achieving a higher utility than that obtained by playing the Stackelberg equilibrium. This has been studied in both general game settings and in specific domains such as auction design, contract design, and information design (Braverman et al., 2018; Deng et al., 2019; Mansour et al., 2022; Cai et al., 2023; Rubinstein and Zhao, 2024; Guruganesh et al., 2024; Lin and Chen, 2024). However, in general games, even when the optimizer has full knowledge of the learner's utility function, the optimization task is known to be NP-hard (Assos et al., 2024, 2025).

Additional related work includes the impossibility of approximating the Stackelberg value when the learner is strategic or noisy (Ananthakrishnan et al., 2024; Donahue et al., 2024); designing learners that are Pareto-optimal against strategic agents (Arunachaleswaran et al., 2024b); learning with Bayesian knowledge about the opponent (Arunachaleswaran et al., 2024a); and computing the Stackelberg equilibrium in continuous games (Fiez et al., 2020; Maheshwari and Mazumdar, 2023; Brown et al., 2024a).

B Discussion and Limitations

In this paper, we propose an algorithm for learning a (ε, δ)-local Stackelberg equilibria of an unknown discrete game through repeated interactions with an agent employing the fictitious play algorithm. Our algorithm uses a number of iterations that is exponential in $\frac{1}{\varepsilon\delta}$ but polynomial in the size of the principal's and agent's action spaces. In particular, we avoid the exponential dependency on the action space size, which is unavoidable for learning the global Stackelberg equilibria as shown by Brown et al. (2024b).

When is our algorithm useful? Our main algorithm's round complexity grows exponentially with the accuracy parameters — a dependence that we show in Theorem C.1 to be unavoidable for all algorithms. Our method is therefore attractive when the action spaces are large and a moderate accuracy suffices: for example, when $m, n \gg \text{poly}(\frac{1}{\varepsilon\delta})$. By contrast, if the goal is to obtain very high accuracy in a small game, then existing approaches with exponential dependency on m, n (such as those in (Brown et al., 2024b) for learning the global Stackelberg equilibria) may be more practical.

Comparison between other benchmarks. Our polynomial-time guarantee is achieved by targeting a weaker benchmark: local Stackelberg equilibria. In general, the principal's utility in local Stackelberg equilibria may be lower than that in global Stackelberg equilibria. We provide a more detailed discussion of the comparison between various benchmarks in Appendix E. In particular, we show that in the class of Stackelberg security games, the local and global Stackelberg equilibria provides the same utility guarantee for the principal. As a corollary, our algorithm can efficiently approximately achieve utility equivalent to the global Stackelberg equilibria against fictitious play agents in unknown security games.

Future directions. In this paper, we focus on fictitious play agents as a first step towards studying learning agents who are slow to forget. A natural and pressing future direction is to extend our algorithms and analysis to *mean-based* agents — agents who *approximately* best respond to the average historical strategies. Addressing this would likely require a more robust SearchForPolytopes subroutine with more careful updates, as approximate best responses become more unstable near the polytope boundaries.

C Lower bound

Theorem C.1 (Lower Bound)**.** *Assume a repeated game between a learner, who employs Fictitious Play, and an optimizer who does not know the learner's utility. Let n be the number of actions for the learner, and let $\epsilon \in (0, 1/3)$. Assume that $n \geq 1/\epsilon^2$. Let m be the number of actions for the optimizer and assume that $m \geq 1/\epsilon^5$. Then, there exists a distribution over games such that for any randomized algorithm used by the optimizer, the expected number of iterations required to find an ϵ-approximate local Stackelberg equilibrium is at least $e^{C/\epsilon}$, where $C > 0$ is a universal constant.*

Further, this lower bound holds under smoothed analysis, when each entry in the learner's utility matrix is perturbed by a small constant and when Assumption F.3 is satisfied.

Proof Sketch. We present the intuition for the lower bound construction here and present the full proof in Appendix J.5. We construct a game that satisfies the following properties:

- There exists a unique local Stackelberg equilibrium, and the only ϵ-approximate local Stackelberg points are in its vicinity. Thus, the optimizer must locate this point.
- Until the optimizer reaches this local Stackelberg equilibrium, its best strategy is to perform a local search, iteratively moving from $\overline{\boldsymbol{x}}^{(t-1)}$ to a neighboring point $\overline{\boldsymbol{x}}^{(t)}$ with a higher utility.

Our goal is to show that in such a game, an optimizer following this local improvement strategy requires time exponential in $1/\epsilon$ to reach the equilibrium. This follows from the following properties of the game:

- With high probability, the optimizer's utility at $\overline{\boldsymbol{x}}^{(1)}$ is close to 0.
- At the local Stackelberg equilibrium, the optimizer's utility is 1.
- For most values of $\boldsymbol{x}$, the optimizer is only slightly shy of being an ϵ-approximate Stackelberg point, and the utility gradient at these points is on the order of ϵ.

These properties imply that the total distance traversed, $\sum_{t=1}^{T} \|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1$, must be at least $\Omega(1/\epsilon)$. Since each incremental change satisfies $\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1 \leq 1/t$ (due to the averaging effect of the history), it follows that the number of iterations T must be at least $\exp(\Omega(1/\epsilon))$.

To construct a game that exhibits this behavior, we define a partition of the simplex $\Delta(\mathcal{A})$ into best-response polytopes. The construction consists of the following key elements:

- A large best-response polytope that covers most of the simplex.

- A path consisting of approximately $1/\epsilon$ vertices, denoted $v_1, \dots, v_\ell$, where $\ell \approx 1/\epsilon$.
- For each vertex v_i, a small best-response polytope centered around it.
- For each edge $v_i v_{i+1}$ along the path, a corresponding polytope covering the edge of the simplex that connects these two vertices.
- No additional best-response polytopes beyond those described.

The game is designed so that the only local Stackelberg equilibrium occurs at v_ℓ. To enforce this, we construct the optimizer's utility within each polytope as follows:

- In the large polytope, the optimizer's utility increases monotonically as one moves toward v_1, with a gradient slightly greater than ϵ.
- Along the path, the optimizer's utility increases when transitioning from the polytope around v_i to the polytope of the edge $v_i v_{i+1}$, and again when moving from the edge polytope to the polytope around v_{i+1}. Further, within each polytope along the path, the optimizer's utility gradually increases.
- This ensures that the optimizer's utility strictly increases along the path, culminating at v_ℓ.

This structure guarantees that no point other than v_ℓ can serve as a local Stackelberg equilibrium. The formal proof provides the details of this construction.

Finally, we argue that the optimizer's best strategy is to follow the designated path. Since the number of vertices in $\Delta(\mathcal{A})$—equivalently, the number of optimizer actions—is large relative to $1/\epsilon$, and since the large polytope provides no information about the path beyond the location of v_1, the optimizer has no alternative but to go along it. Any attempt to shortcut the path would require searching through many vertices or edges of the simplex to locate one that lies on the path. This exhaustive search would take at least $\exp(\Omega(1/\epsilon))$ iterations, establishing our result.

D Smoothed analysis

In this section, we show that the singular value assumption (Assumption F.2) holds with high probability when the agent's utility matrix $\boldsymbol{U_2}$ is obtained by perturbing any given utility matrix (denote the original matrix with $\boldsymbol{U_2}$) with i.i.d. Gaussian noise.

More formally, let $\overline{\boldsymbol{U_2}} \in [0,1]^{m\times n}$ be the initial agent utility where each entry $U_2(a,b) \in [0,1]$ represents the agent's utility under action pair $(a,b) \in \mathcal{A} \times \mathcal{B}$. We will perturb this utility matrix by adding an independent Gaussian noise $W(a,b) \sim \mathcal{N}(0,\sigma^2)$ to each entry (a,b), i.e.,

$$\forall (a,b) \in \mathcal{A} \times \mathcal{B}, \quad U_2(a,b) = \overline{U}_2(a,b) + W(a,b).$$

In matrix form, this can be written as $\boldsymbol{U_2} = \overline{\boldsymbol{U_2}} + \boldsymbol{W}$, where $\boldsymbol{W} \overset{\text{iid}}{\sim} \mathcal{N}(0,\sigma^2)$ is the Gaussian perturbation matrix. Recall that as defined in Definition F.1, G_b denotes the augmented constraint matrix for any agent action $b \in \mathcal{B}$, and $\mathcal{S}_m(G_b)$ denotes the set of all $m \times m$ square sub-matrices of G_b.

Theorem D.1 (Lower bound on the minimum singular value)**.** *Let $\overline{\boldsymbol{U_2}} \in [0,1]^{m\times n}$ be an arbitrary utility matrix of the agent, and let $\boldsymbol{U_2}$ be a Gaussian perturbation of $\overline{\boldsymbol{U_2}}$ with variance σ^2. Then the resulting augmented constraint matrices of the perturbed utility matrix satisfies that for $\underline{\sigma} = \Theta\left(\frac{\sigma\delta}{m^{\frac{5}{2}}2^n}\right)$, Assumption F.2 holds with probability at least $1-\delta$.*

Remark D.2. *Although $\underline{\sigma}$ has an exponential dependency on m, in our final bound (Theorem 3.1), the dependence is only through* $\log(1/\underline{\sigma})$. *As a result, this introduces only a logarithmic dependence on m and a polynomial dependence on n.*

To establish Theorem D.1 under smoothed analysis, we will make use of the result on the minimum singular value of a Gaussian perturbed square matrix (Sankar et al., 2006). However, since the submatrix of the augmented constraint matrix G_b could contain rows from the identity matrix I_m or the all-one vector $\mathbf{1}_m$ (which are not perturbed by Gaussian noise), we need to perform some special treatments to these cases. We prove Theorem D.1 in Appendix J.6.

E Comparing local Stackelberg benchmark with other benchmarks

We will compare the local Stackelberg with other solution concepts in this section. We will compare solution concepts according to two criteria.

The first is the utility the principal achieves in a solution concept. Since there can be multiple solutions in a solution concept, we will compare the least principal utilities across solution concepts.

The second criterion is the computational feasibility of approximating that solution concept through interactions with a learning agent.

- **Stackelberg equilibrium:** The Stackelberg equilibrium is the local Stackelberg equilibrium with the highest utility for the principal. However, this can be intractable to approximate when interacting with a mean-based learner (Brown et al., 2024b).
- **Coarse Correlated Equilibrium (CCE):** A CCE be efficiently approximated against a mean-based learner. This can be done by simply employing a no-regret learning algorithm. There are cases where the minimum principal utility among CCEs is greater than the minimum principal utility among local Stackelberg equilibria and cases where the relation goes the other way. In particular, for Stackelberg security games, the local Stackelberg equilibrium achieve the same utility for the principal as the global Stackelberg equilibrium (see Proposition E.3 for more details). Hence, for this class of games, the local Stackelberg equilibrium yields a higher utility compared to any CCE (Von Stengel and Zamir, 2010).
- **Smoothed Local Stackelberg Equilibrium.** The smoothed local Stackelberg can be thought of the local Stackelberg equilibrium in the setting where there is some noise (from $\mathcal{N}(0, \eta^2 I)$) added to the principal's strategy.

  The noise added allows us to side-step the challenge of discontinuous utilities for the principal. However, the utilities of the smoothed local Stackelberg can be much worse than the utility of any local Stackelberg.

  The smoothed local Stackelberg is defined as follows:

  Definition E.1 ($(\varepsilon, \delta, \eta)$-Smoothed Local Stackelberg Equilibria)**.** *A principal's strategy $x \in \Delta(\mathcal{A})$ is an $(\varepsilon, \delta, \eta)$-Smoothed Local Stackelberg strategy if*

$$\forall \boldsymbol{x}' \in \mathbf{B}_1(\boldsymbol{x}; \delta), \qquad \mathop{\mathbb{E}}_{z' \sim \mathcal{N}(\boldsymbol{x}', \eta^2 I)} \sup_{y' \in \mathsf{BR}(\boldsymbol{z}')} U_1(\boldsymbol{z}', y') \leq \mathop{\mathbb{E}}_{z \sim \mathcal{N}(\boldsymbol{x}, \eta^2 I)} \sup_{y \in \mathsf{BR}(\boldsymbol{z})} U_1(\boldsymbol{z}, y) + \varepsilon\delta,$$

  where $\mathbf{B}_1(\boldsymbol{x}, \delta)$ denote the ℓ_1 ball of radius δ around $\boldsymbol{x}$, i.e., the set of strategies with $\|\boldsymbol{x}' - \boldsymbol{x}\|_1 \leq \delta$.

The random noise added to the principal's strategy smoothens the principal's utility and makes it continuous. As a result, gradient-free optimization methods (Maheshwari and Mazumdar, 2023; Fiez et al., 2020; Flaxman et al., 2004; Nesterov and Spokoiny, 2017; Gasnikov et al., 2023) can be used to compute a $(\varepsilon, \delta, \eta)$-Smoothed Local Stackelberg equilibrium.

We can however show that the worst smoothed local Stackelberg equilibrium can have principal's utility at least $\Omega(\varepsilon^{1/d})$ worse than the the worst local Stackelberg equilibrium.

We can show this by constructing an example as in Figure 3. In this example, there is a very thin polytope P_1 that still has a ball of radius R. Note that this example satisfies the assumptions we use for our results.

At the point of intersection of the three polytopes $\overline{x}$, the actions of polytopes P_1, P_2, P_3 yield the principal utilities in that order, with the utilities differing by a constant amount. $\overline{x}$ is the optimal strategy within P_2. Since the action of P_3 yields lesser utility and since P_1 is a very thin polytope, $\overline{x}$ is a smoothed LSE when $\eta \in o(R\varepsilon^{1/d})$.

$\overline{x}$ is however not an LSE. $\boldsymbol{x}^*$ is the only LSE and has at least $\Omega(R)$ utility larger than $\overline{x}$.

On the other hand, if $\eta \in \Omega(\varepsilon^{1/d})$, two points in the same best-response polytope at ℓ_1 distance η both form a smoothed Stackelberg equilibrium. By linearity of utilities within best-response polytopes, one of these points has principal utility less than $\Omega(\eta)$ compared to the other.

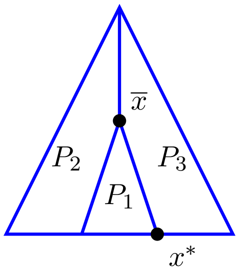


Figure 3: LSE vs Smoothed LSE : A game where $\overline{x}$ is a smoothed LSE for small enough η and x^* is the sole LSE. $\overline{x}$ is the optimal strategy within polytopes P_2 and P_3. And x^* is the optimal strategy within P_1. The strategy $\overline{x}$ with action of P_1 has utility 1 more than the strategy of $\overline{x}$ with actions of P_2 or P_3. Since P_1 is a very thin polytope, $\overline{x}$ is nevertheless a smoothed LSE. It is however not an LSE.

Stackelberg Security Games. In the remainder of this section, we consider a structured class of games: the Stackelberg Security Games (SSG) (Kiekintveld et al., 2009; An et al., 2012). We will show that under standard non-degeneracy assumptions (Assumption E.2), the value of any local Stackelberg equilibria is the same as the value of the global (strong) Stackelberg equilibria.

We set up the notations for the SSG model, following the formulation from (Haghtalab et al., 2022). In a SSG, there is a set of n targets that the principal aims to protect. The agent has action space $\mathcal{B} = [n]$, i.e., the agent can choose (potentially randomly) a target to attack. The principal's strategy space is a downward-closed subset $\mathcal{A} \subseteq [0, 1]^n$, where each coordinate x_i represents the amount of resource that the principal puts in protecting target $i \in [n]$. For $\boldsymbol{x} \in \mathcal{A}$ and $y \in \mathcal{B}$, the principal's and agent's utility functions are $U_1(\boldsymbol{x}, y) = u^y(x)$ and $U_2(\boldsymbol{x}, y) = v^y(x)$, where u^y and v^y are 1-dimensional functions that are strictly increasing and strictly decreasing respectively, and satisfy the slope bounds in Assumption E.2.

Assumption E.2 (Regularity (Haghtalab et al., 2022)). *There exists a constant $C \geq 1$ such that for all $0 \leq s < t \leq 1$ and $y \in \mathcal{B}$, the functions u^y and v^y satisfy*

$$\frac{1}{C} \leq \frac{v^y(s) - v^y(t)}{t - s} \leq C, \quad 0 < \frac{u^y(t) - u^y(s)}{t - s} \leq C.$$

Proposition E.3 (Equivalence of Local and Global Stackelberg Values). *For Stackelberg Security Games satisfying the regularity assumptions in Assumption E.2, every local Stackelberg equilibria $\tilde{\boldsymbol{x}}$ achieves the same principal utility as the global Stackelberg equilibria $\boldsymbol{x}^\star$, i.e.,*

$$U_1(\tilde{\boldsymbol{x}}, \mathsf{BR}(\tilde{\boldsymbol{x}})) = U_1(\boldsymbol{x}^\star, \mathsf{BR}(\boldsymbol{x}^\star)).$$

Proof of Proposition E.3. We prove this proposition using the characterization through *conservative strategies* proposed by Haghtalab et al. (2022). A principal's strategy $\boldsymbol{x} \in \mathcal{A}$ is called *conservative* if for all $y \in [n]$, $x_y > 0$ only if $y \in \mathsf{BR}(\boldsymbol{x})$, i.e., $\boldsymbol{x}$ only protect targets that belongs to the agent's best response set. Following (Haghtalab et al., 2022, Proposition 3.5), it is without loss of generality to assume that both $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}^\star$ are conservative strategies, as otherwise they can be easily transformed to conservative strategies without changing the principal's utility under best response.

Since both $\boldsymbol{x}^\star$ and $\tilde{\boldsymbol{x}}$ are conservative principal strategies, Proposition 3.5 from (Haghtalab et al., 2022) show that $\boldsymbol{x}^\star$ is the unique conservative maximizer of $U_1(\boldsymbol{x}^\star, \mathsf{BR}(\boldsymbol{x}^\star))$, which is also the unique conservative minimizer of $U_2(\boldsymbol{x}^\star, \mathsf{BR}(\boldsymbol{x}^\star))$. Therefore, if if $U_1(\tilde{\boldsymbol{x}}, \mathsf{BR}(\tilde{\boldsymbol{x}})) < U_1(\boldsymbol{x}^\star, \mathsf{BR}(\boldsymbol{x}^\star))$, it must be the case that $U_2(\tilde{\boldsymbol{x}}, \mathsf{BR}(\tilde{\boldsymbol{x}})) > U_2(\boldsymbol{x}^\star, \mathsf{BR}(\boldsymbol{x}^\star))$. Therefore, from Lemma 3.4 of (Haghtalab et al., 2022), we have $\mathsf{BR}(\tilde{\boldsymbol{x}}) \subseteq \mathsf{BR}(\boldsymbol{x}^\star)$, and $\tilde{\boldsymbol{x}}_y < \boldsymbol{x}^\star_y$ for all targets $y \in \mathsf{BR}(\boldsymbol{x}^\star)$. Since the principal's action space $\mathcal{A}$ is downward-closed, we can therefore infinitesimally increase the resource that the principal invests on every $y \in \mathsf{BR}(\tilde{\boldsymbol{x}})$ without changing the best response set. In other words, there exists a vector $\vec{\varepsilon} \leq \boldsymbol{x}^\star - \tilde{\boldsymbol{x}}$ such that $\varepsilon_y > 0$ for all $y \in \mathsf{BR}(\tilde{\boldsymbol{x}})$, such that the new strategy $\tilde{\boldsymbol{x}}'$ satisfies $\mathsf{BR}(\tilde{\boldsymbol{x}}') = \mathsf{BR}(\tilde{\boldsymbol{x}})$. Since u^y is strictly increasing, we have

$$U_1(\tilde{\boldsymbol{x}}', \mathsf{BR}(\tilde{\boldsymbol{x}}')) = \max_{y \in \mathsf{BR}(\tilde{\boldsymbol{x}}')} u^y(\tilde{\boldsymbol{x}}'_y) = \max_{y \in \mathsf{BR}(\tilde{\boldsymbol{x}})} u^y(\tilde{\boldsymbol{x}}'_y) > \max_{y \in \mathsf{BR}(\tilde{\boldsymbol{x}})} u^y(\tilde{\boldsymbol{x}}_y) = U_1(\tilde{\boldsymbol{x}}, \mathsf{BR}(\tilde{\boldsymbol{x}})),$$

which contradicts with the fact that $\tilde{\boldsymbol{x}}$ is a local Stackelberg equilibria! Therefore, any local Stackelberg equilibria must induce the same principal utility as the global Stackelberg equilibria. □

F Our Assumptions - Extended

In this section, we state our assumptions and their implications for the structures of the optimization problem.

Our first assumption involves the concept of constraint matrices.

Definition F.1 (Constraint Matrix). *For each of agent's action $b \in \mathcal{B}$, let H_b be the* constraint matrix *formed by the set of potential hyperplanes that separate P_b from all other polytopes—i.e., $H_b = \left[\boldsymbol{h}_{b,b'}^\mathsf{T}\right]_{b' \in \mathcal{B} \setminus \{b\}}$. Additionally, since the strategies all satisfy the simplex constraint (i.e., $\mathbf{1}^\mathsf{T}\boldsymbol{x} = 1$ and $\boldsymbol{x} \geq \mathbf{0}$), we define the* augmented constraint matrix *as the matrix obtained by augmenting H_b with an all-1 row vector ($\mathbf{1}_m$) and the identity matrix (I_m), and denote it with $G_b = \begin{bmatrix} H_b^\mathsf{T} & \mathbf{1}_m^\mathsf{T} & I_m \end{bmatrix}^\mathsf{T}$.*

Assumption F.2 (Minimum singular value of square submatrix of G_b). *Let $\mathcal{S}_m(G_b)$ denote all the $m \times m$ square submatrices of G_b. We assume that for all $b \in \mathcal{B}$ and all square submatrices $G'_b \in \mathcal{S}_m(G_b)$, the minimum singular value of G'_b satisfies $\sigma_{\min}(G'_b) \geq \underline{\sigma}$.*

We will justify Assumption F.2 in Appendix D, by showing that it holds with high probability under the smoothed analysis framework of Spielman and Teng (2004).

Assumption F.3 (Distance from Polytope Boundaries)**.** *For every polytope* P_b*, there exists some strategy* $\boldsymbol{x} \in \mathrm{P}_b$ *such that all coordinates of* $\boldsymbol{x}$ *are lower bounded by* $R_{\min}$*, i.e.,* $\boldsymbol{x} \geq R_{\min} \cdot \mathbf{1}$*.*

Remark F.4. *Assumption F.3 is an arguably weaker condition than the assumption made in previous works on computing an approximate Stackalberg Equilibrium from Best-Response oracle or from interactions with an agent, such as Blum et al. (2014); Letchford et al. (2009); Haghtalab et al. (2022). Their assumption states that any polytope* P_b *contains an* ℓ_2 *ball whose radius is lower bounded by some constant value* r_0*. Assumption F.3 is weaker for the following reasons:*

1. *Assumption F.3 is a direct implication of the ball assumption. If a polytope* P_b *contains a point* $\boldsymbol{x}_b$ *that satisfies the ball assumption, then the distance from* $\boldsymbol{x}_b$ *to all boundaries of* P_b *is at least* r_0*, which naturally implies that the minimum coordinate of* $\boldsymbol{x}_b$ *is also lower bounded by* r_0*.*

2. *Assumption F.3 only concerns the distance to the simplex boundaries, whereas the ball assumption requires the distance to all polytope boundaries to be lower bounded as well.*

3. *Previous works that build on the ball assumption (e.g., Blum et al. (2014); Letchford et al. (2009); Haghtalab et al. (2022, 2024)) usually incurs a dependency on the volume of the ball, which can be exponential in the dimension (i.e. the number of principal's actions). In contrast, our bound only has a polynomial dependence on* $1/R_{\min}$*.*

Structural properties implied by Assumption F.2 In the following lemma, we show that Assumption F.2 implies that the polytopes and hyperplanes are sufficiently separated from each other. We defer the proof of this lemma to Appendix J.1.

Lemma F.5 (Polytopes/Hyperplanes are Far Apart)**.** *Let* $\underline{\sigma}$ *be the lower bound on the minimum singular values defined in Assumption F.2. We have*

1. *Let* $R_1 = \underline{\sigma}/(2m^{3/2})$*. For all* $\boldsymbol{x} \in \Delta(\mathcal{A})$*,* $|\mathcal{P}_{R_1}(\boldsymbol{x})| \leq m$*, i.e., there are at most* m *polytopes that have distance at most* R_1 *to* $\boldsymbol{x}$*.*

2. *Let* $R_2 = \underline{\sigma}/(2m)$*. For all* $b \in \mathcal{B}$ *and all* $\boldsymbol{x} \in \mathrm{P}_b$*, there are at most* $(m-1)$ *rows* h *of* G_b *that satisfy* $0 \leq \langle x, h \rangle \leq R_2$*. This implies that there are at most* $m-1$ *hyperplanes from that polytope that* $\boldsymbol{x}$ *can be very close to.*

G Querying through Average Strategies

Recall that we use $\boldsymbol{x}^{(t)}$ to denote the principal's strategy at round t, and $\overline{\boldsymbol{x}}^{(t)} = \frac{1}{t}\sum_{s=1}^{t} \boldsymbol{x}^{(s)}$ to denote the average strategy during the first t rounds. Since a fictitious play agent best responds to the average strategy, we treat $\overline{\boldsymbol{x}}^{(t)}$ as our *effective search point* at round t. Through observing the best response action action from the agent, i.e., $y_t \in \mathsf{BR}(\overline{\boldsymbol{x}}^{(t)})$, we can infer which best response polytope the average strategies lie in. This best response further dictates which utility function $U_1(\cdot, y_t)$ the principal should be optimizing.

However, an inherent constraint of the *effective search points* is that we can not move too far between consecutive search points. Specifically, the maximum distance that we can travel in one step (i.e., $\|\overline{\boldsymbol{x}}^{(t+1)} - \overline{\boldsymbol{x}}^{(t)}\|$) shrinks at a rate of $O(1/t)$, and is often constrained by the previous search point's distance to the simplex boundaries. To capture this, we say that the principal takes *step size* $\eta^{(t)}$ at round t if the average strategy moves an ℓ_1 distance of $\eta^{(t)}$, i.e.,

$$\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1 = \eta^{(t)}/t. \tag{Step size $\eta^{(t)}$}$$

In Algorithm 4, we introduce a procedure for updating the average strategy with a pre-specified update direction and step size.

ALGORITHM 4: MoveOneStep

Input: Current round t, the current average principal strategy $\overline{\boldsymbol{x}}^{(t-1)} \in \mathcal{X}$, move direction $\mathbf{u}^{(t)} \in \mathcal{X}$,
step size $0 \leq \eta^{(t)} \leq \|\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1$.

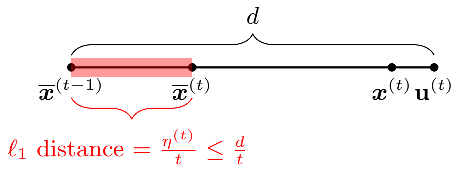
Output: The principal strategy $\boldsymbol{x}^{(t)}$, such that the average strategy $\overline{\boldsymbol{x}}^{(t)}$ moves by an ℓ_1 distance of $\eta^{(t)}$ in the direction of $\mathbf{u}^{(t)}$.

$$\boldsymbol{x}^{(t)} = \left(1 - \frac{\eta^{(t)}}{\|\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1}\right)\overline{\boldsymbol{x}}^{(t-1)} + \frac{\eta^{(t)}}{\|\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1}\mathbf{u}^{(t)}.$$

Algorithm 4 chooses a point $\boldsymbol{x}^{(t)}$ on the line segment between $\overline{\boldsymbol{x}}^{(t-1)}$ and $\mathbf{u}^{(t)}$. In other words, $\boldsymbol{x}^{(t)}$ is formed by moving the average strategy $\overline{\boldsymbol{x}}^{(t-1)}$ in the direction of $\mathbf{u}^{(t)}$. Note that $\boldsymbol{x}^{(t)}$ is a valid strategy in the simplex $\mathcal{X}$ because it is a linear combination of two valid strategies $\overline{\boldsymbol{x}}^{(t-1)}$ and $\mathbf{u}^{(t)}$. It has the following property:

$$\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)} = \frac{\boldsymbol{x}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}}{t} = \frac{\eta^{(t)}}{t} \cdot \frac{\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}}{\|\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1} \quad \Rightarrow \quad \|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1 = \frac{\eta^{(t)}}{t}.$$

Remark G.1 (Maximum step size in any direction)**.** *At round t, the maximum ℓ_1 that the principal can move its average strategy towards the direction $\mathbf{u}^{(t)}$ is $\|\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1/t$. In other words, the maximum feasible step size is $\|\mathbf{u}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1$.*



d
$\overline{\boldsymbol{x}}^{(t-1)}$
$\overline{\boldsymbol{x}}^{(t)}$
$\boldsymbol{x}^{(t)}$
$\mathbf{u}^{(t)}$
ℓ_1 distance $= \frac{\eta^{(t)}}{t} \leq \frac{d}{t}$

Figure 4: An illustration of how to move the average strategies in MoveOneStep. If $\|\overline{\boldsymbol{x}}^{(t-1)} - \mathbf{u}^{(t)}\|_1 = d$, then by choosing appropriate $\boldsymbol{x}^{(t)}$ along the line segment between $\overline{\boldsymbol{x}}^{(t-1)}$ and $\mathbf{u}^{(t)}$, the principal's average strategy can move ℓ_1 distance of $\|\overline{\boldsymbol{x}}^{(t-1)} - \overline{\boldsymbol{x}}^{(t)}\|_1 \in [0, \frac{d}{t}]$ (the shaded region is achievable).

H Supplementary materials for OptimizeWithinPolytope

H.1 Correctness of OptimizeWithinPolytope

In this section, we establish the correctness of OptimizeWithinPolytope. The OptimizeWithinPolytope subroutine invokes another subroutine SearchForPolytopes that to find the hyperplane separating two points in different polytopes. We will defer the analysis of the SearchForPolytopes subroutine to

Section 3.2 and simply assume that given $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ that lie in two different best response polytopes, the subroutine returns an α-approximate hyperplane to the true separating hyperplane. That is, if the true hyperplane separating $\boldsymbol{x}, \boldsymbol{x}'$ is $\boldsymbol{h}$, then, SearchForPolytopes returns a hyperplane $\hat{\boldsymbol{h}}$ that satisfies $\|\hat{\boldsymbol{h}} - \boldsymbol{h}\|_2 \leq \alpha$.

The main technical lemmas are listed below.

Lemma H.1 (Closeness of optimal values within true and estimated polytopes)**.** *Let $H, \hat{H} \in \mathbb{R}^{k \times m}$ with $k \leq n$ be the true and estimated constraints, which satisfy $\|H - \hat{H}\|_2 \leq \alpha\sqrt{m}$, where α is the estimation error that can be chosen sufficiently small. Consider polytopes P_b and $\hat{\mathrm{P}}_b$ defined as follows:*

$$\mathrm{P} = \{\boldsymbol{x} \in \mathbb{R}^m : Hx \geq 0, \mathbf{1}^\mathsf{T}\boldsymbol{x} = 1, \boldsymbol{x} \geq \mathbf{0}\}, \quad \hat{\mathrm{P}} = \{\boldsymbol{x} \in \mathbb{R}^m : \hat{H}x \geq \alpha, \mathbf{1}^\mathsf{T}\boldsymbol{x} = 1, \boldsymbol{x} \geq \gamma \cdot \mathbf{1}\}$$

Then, when maximizing the principal's utility $U_1(\cdot, b)$ over P_b and $\hat{\mathrm{P}}_b$, the corresponding optimal values satisfy

$$\max_{\boldsymbol{x} \in \hat{P}} U_1(\boldsymbol{x}, b) \leq \max_{\boldsymbol{x} \in P} U_1(\boldsymbol{x}, b) + \frac{2\alpha m}{\underline{\sigma} - \alpha\sqrt{m}} + 2\gamma/R_{\min},$$

where α is chosen to be at most $O\left(\frac{\sigma}{m^2 n}\right)$, $\underline{\sigma}$ is the lower bound on minimum singular values from Assumption F.2, $R_{\min}$ is the parameter from Assumption F.3, and $R_2 = \underline{\sigma}/m$ is from Lemma F.5.

Proof sketch (Full proof in Appendix J.2). The high level idea is that given a point $\boldsymbol{x}$ satisfying constraints according to the constraint matrix $H\boldsymbol{x} \geq 0$, we want to perturb $\boldsymbol{x}$ to the point $\boldsymbol{x} + \boldsymbol{z}$ satisfying the slightly perturbed constraint of $\hat{H}\boldsymbol{x} \geq \alpha$.

The constraints that $\boldsymbol{x}$ satisfies by a low margin are the most sensitive to being violated by perturbing $\boldsymbol{x}$. So we will focus on these constraints. Specifically, we will focus on the set of constraints indexed by J that are satisfied by a margin of less than R_2. By our Lemma F.5, we know that $|J| \leq m$.

Finding a $\boldsymbol{z}$ that satisfies the perturbed constraint with $\leq m$ constraints perturbed amounts to solving a system of equations with fewer equations than constraints. The norm of the minimum norm solution is provided by the inverse of the minimum singular value of H_J which we assume to be bounded below by $\underline{\sigma}$ (Assumption F.2).

We can show that we can find a $\boldsymbol{z}$ with norm small enough that the remaining contraints outside of J that were satisfied by a large margin remain satisfied even after perturbation by $\boldsymbol{z}$. □

H.2 Sample complexity of OptimizeWithinPolytope

Recall that the time steps that the principal spends in OptimizeWithinPolytope can be divided into two categories: the *improvement steps* and the *searching steps.* In this section, we establish guarantees for the two categories separately.

Analysis of the improvement steps. We will first analyze the *improvement steps* by showing that each step improves the principal's utility by at least $\varepsilon\delta/t$ (Lemma H.2). This lemma will be crucial for bounding the total number of different polytopes, as the cumulative utility improvement is at most 1.

Lemma H.2 (Utility Improvement)**.** *Suppose $\overline{\boldsymbol{x}}^{(t-1)} \to \overline{\boldsymbol{x}}^{(t)}$ is an improvement step, where both strategies are inside P_b. The principal's utility increases by at least $\frac{\varepsilon\delta}{t}$ during this improvement step.*

$$U_1(\overline{\boldsymbol{x}}^{(t)}, b) - \boldsymbol{u}(\overline{\boldsymbol{x}}^{(t-1)}, b) \geq \frac{\varepsilon\delta}{t}.$$

Proof of Lemma H.2. Since the OptimizeWithinPolytope algorithm does not terminate at round $t-1$, we know that there exists $\boldsymbol{z} \in \hat{\mathrm{P}}_b$ such that $U_1(\boldsymbol{z}, b) > U_1(\overline{\boldsymbol{x}}^{(t-1)}, b) + \varepsilon\delta$. Therefore, by choosing $\boldsymbol{x}^{(t)} = \boldsymbol{z}$, the principal's utility at $\overline{\boldsymbol{x}}^{(t)}$ can be calculated as

$$U_1(\overline{\boldsymbol{x}}^{(t)}, b) = \frac{t-1}{t} \cdot U_1(\overline{\boldsymbol{x}}^{(t-1)}, b) + \frac{1}{t} \cdot U_1(\boldsymbol{z}, b).$$

The improvement in utility therefore satisfies

$$U_1(\overline{\boldsymbol{x}}^{(t)}, b) - \boldsymbol{u}(\overline{\boldsymbol{x}}^{(t-1)}, b) = \frac{U_1(\boldsymbol{z}, b) - U_1(\overline{\boldsymbol{x}}^{(t-1)}, b)}{t} > \frac{\varepsilon\delta}{t}. \qquad \square$$

Analysis of the searching steps. The searching phases consists of two subroutines: BinarySearch (Algorithm 5) and SearchForPolytopes (Algorithm 3, which we analyze in Appendix I.1). We provide the guarantee of BinarySearch and defer its proof to Appendix J.3.

Lemma H.3 (Binary search to get close to the boundary)**.** *Let γ be the minimum distance from all search point to the boundary in Lemma H.1. Suppose the principal's average strategy moves from $\overline{\boldsymbol{x}}^{(t-1)}$ in polytope P_b to $\overline{\boldsymbol{x}}^{(t)}$ in a different polytope $\mathrm{P}_{b'}$, with ℓ_1 step size of $\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1 = \frac{\varepsilon}{t}$, where we assume $t \geq 10\frac{\varepsilon}{\gamma}$. Then the procedure* BinarySearch *(see Algorithm 5) takes at most $s \leq O\left(\frac{\varepsilon}{\gamma} + \log \frac{\varepsilon}{\alpha t}\right)$. steps and returns two consecutive average strategies $\overline{\boldsymbol{x}}^{(t+s-1)}$, $\overline{\boldsymbol{x}}^{(t+s)}$, such that they are on the opposite sides of the boundary (one is in P_b, the other is in $\mathrm{P}_{b'}$), and they are close in ℓ_1 distance, i.e., $\|\overline{\boldsymbol{x}}^{(t+s-1)} - \overline{\boldsymbol{x}}^{(t+s)}\|_1 \leq \alpha$.*

I Supplementary materials for SearchForPolytopes

I.1 Correctness of SearchForPolytopes

Theorem 3.4 (Correctness of SearchForPolytopes)**.** *Starting from a point $\boldsymbol{x}^*$ in P_b, for any $\alpha \in (0, o(R_2\underline{\sigma}/m^3))$ and $\rho < \alpha$,* SearchForPolytopes *finds $\hat{\boldsymbol{h}}_{b,b'}$ such that $\|\hat{\boldsymbol{h}}_{b,b'} - h_{b,b'}\|_2 \leq \alpha$, for every $b' \in \mathcal{P}_\rho(\boldsymbol{x}^*)$.*

In this section, we outline the main components of the proof of Theorem 3.4. We will present the full proof in Appendix J.4.

Let J of size j index the polytopes already discovered in previous iterations. We will analyze the iteration of the algorithm with the search space $\hat{\mathcal{S}}_J = \{\boldsymbol{x} \in \Delta(\mathcal{A}) : \hat{\boldsymbol{h}}_J\boldsymbol{x} \geq \alpha_j\}$, where $\hat{\boldsymbol{h}}_J$ is the matrix of approximations to the hyperplanes discovered thus far. Inductively we will show that if $\|\hat{\boldsymbol{h}}_{b,b_j} - \boldsymbol{h}_{b,b_j}\| \leq \alpha_j$ for every $j \in J$, the next constructed hyperplane satisfies $\|\hat{\boldsymbol{h}}_{b,b_{\mathrm{new}}} - h_{b,b_{\mathrm{new}}}\| \leq \alpha_j O(\alpha_j^2 m^2/\underline{\sigma}^2)$. By setting $\alpha_1 \leq \alpha(\underline{\sigma}/m)^m$, we ensure that all $\alpha_j \leq \alpha$. This is because the number of iterations of this algorithm is at most m as shown in the following lemma.

Lemma I.1 (Algorithm terminates after $m-1$ hyperplanes added)**.** *The* SearchForPolytopes *algorithm will terminate after at most $m-1$ hyperplanes are added.*

Now we will discuss how random search on the search space via Gaussian random vectors will discover new polytopes in $\mathcal{P}_\rho(\boldsymbol{x}^*)$ that are not yet discovered through sufficiently many search points landing in them.

Lemma I.2 (Search finds points close to a new boundary)**.** *With high probability, $\Omega(1/m)$ of the search points generated lie in a new, undiscovered polytope $\mathrm{P}_{b_{\mathrm{new}}}$ and lie within distance $O(\eta_j\sqrt{m} + \alpha_j m^3/\underline{\sigma})$ of $\boldsymbol{h}_{b,b_{\mathrm{new}}}$.*

Proof. First we will show that many search points fall in a new, previously undiscovered polytope. In the space $\{\boldsymbol{x} \in \Delta(\mathcal{A}) : H_J \leq \alpha_j\}$, the agent gets the same utility by playing any action in J or the action of the current polytope. Since our search space is an approximation of this space, the agent is approximately indifferent among all already discovered actions for strategies in the search space (Lemma J.4).

As long as there is a polytope yet to be discovered in $\mathcal{P}_\rho$, we show that random search generates a constant fraction of principal strategies where the action corresponding to undiscovered polytope yields a higher utility, indicating that the searched strategies lie in a previously undiscovered polytope (Lemma J.5).

Finally, due to the η scaling of the search vectors, the searched strategies are $\eta\sqrt{m}$ close to the point $\hat{\boldsymbol{x}}_J$ we search from which in turn is close to $\boldsymbol{x}^*$ (Lemma J.3) which lies on $\boldsymbol{h}_{b,b_{\text{new}}}$. This shows that the points discovered in $\mathrm{P}_{b_{\text{new}}}$ are close to the boundary hyperplane $\boldsymbol{h}_{b,b_{\text{new}}}$. □

The properties of the search vectors in the new polytope — a significant fraction of all search vectors and lying close to the boundary allow us to accurately reconstruct $\boldsymbol{h}_{b,b_{\text{new}}}$. Let Y be a matrix of the search points that land in the new polytope $\mathrm{P}_{b_{\text{new}}}$. We will construct a hyperplane solving $\hat{\boldsymbol{h}} = \operatorname{argmin}_{\boldsymbol{h}} \|\boldsymbol{h}Y^{\mathsf{T}}\|$. We will now show that $\hat{\boldsymbol{h}}$ is a good approximation to $\boldsymbol{h}_{b,b_{\text{new}}}$

Lemma I.3 (Approximating a hyperplane)**.** *Suppose* RandomSearch *in the search space* $\hat{\mathcal{S}}_J = \{\boldsymbol{x} : \hat{\boldsymbol{h}}_J\boldsymbol{x} = \alpha_j\}$ *generates* $d \in \Theta(m^2 \log m)$ *points. Let* b_{new} *be the undiscovered polytope with the maximum number of search points. Let* $Y \in \mathbb{R}^{k\times m}$ *be a matrix where the rows are all search points in* $\mathrm{P}_{b_{\text{new}}}$ *with distance from* $\hat{\boldsymbol{x}}_J \in O(\eta\sqrt{m})$.

Then $\hat{\boldsymbol{h}} = \operatorname{argmin}_{\boldsymbol{h}} \|hY^t\|_2$ *satisfies* $\|\hat{\boldsymbol{h}} - h_{b,b_{\text{new}}}\| \leq \alpha$.

Proof. The proof idea is to first provide a lower bound in the minimum singular value of Y. Because the rows in Y lie close to $\boldsymbol{h}_{b,b_{\text{new}}}$, $\|h_{b,b_{\text{new}}}Y^t\|$ is small. Since $\hat{\boldsymbol{h}}$ minimizes the norm $\operatorname{argmin}_h \|\hat{\boldsymbol{h}}Y^{\mathsf{T}}\|$, $\|h_{b,b_{\text{new}}}Y^t\|$ is also small. Due to the lower bound on the minimum singular value of Y, this will imply that $\|\hat{\boldsymbol{h}} - h_{b,b_{\text{new}}}\|$ is small. □

So far we argued that as long as some polytope in $\mathcal{P}_\rho(\boldsymbol{x}^*)$ is undiscovered, we will find new polytopes through random search. To complete the argument, we will argue that this means that *all* polytopes in $\mathcal{P}_\rho(\boldsymbol{x}^*)$ are discovered before the algorithm terminates. This is due to the property that no more than m polytopes surround a point and the only way the algorithm could terminate without finding some polytope in $\mathcal{P}_\rho(\boldsymbol{x}^*)$ is if there were more than m surrounding polytopes.

I.2 Complexity of SearchForPolytopes

Lemma I.4 (Number of rounds spent searching for polytopes)**.** *When* $\alpha \in O(\gamma\underline{\sigma}^2/(n^2m^7 \log m))$, *the number of rounds spent in calls of* SearchForPolytopes *is at most* $O(n^2m^2 \log m)$.

Proof. SearchForPolytopes is invoked at most n^2 times, once when encountering each hyperplane. The different types of movements are: 1) taking a random step of ℓ_2 distance at most $\eta\sqrt{m}$, 2) returning back to the search point, and 3) projecting the search point back to updated search spaces.

Each of these movements have a ℓ_1 length in $\Delta \in O(\alpha m^5/\underline{\sigma}^2)$. And the total number of these movements is $r = O(n^2m^2 \log m)$ over all the n^2 possible times SearchForPolytopes is invoked.

In rounds t with $\gamma/t \geq \Delta$, we can make the movement in a single round. So the number of rounds before $t_0 = \gamma/\Delta$ is at most $r = O(n^2 m^2 \log m)$.

For rounds after t_0, may need multiple steps to make the movement. In these rounds t, we will traverse exactly γ/t ℓ_1 distance.

The total distance traversed while making all these movements is at most $r\Delta$. Let us lower bound the total distance traversed by the total distance traveled in rounds after $t_0 = \gamma/\Delta$.

$$r\Delta \geq \sum_{\tau=t_0}^{t_0+K} \frac{\gamma}{\tau} \geq \gamma \log\left(1 + \frac{K}{t_0}\right) \quad \Longrightarrow K \leq t_0\left(\exp\left(\frac{r\Delta}{\gamma}\right) - 1\right)$$

Our assumption that $\alpha \in O(\gamma\underline{\sigma}^2/(n^2 m^7 \log m))$ implies that $r\Delta/\gamma \leq 1$. So, using $e^x - 1 \leq x$, $K \leq t_0\frac{r\Delta}{\gamma}$. Therefore the total number of rounds spent in $\mathsf{SearchForPolytopes}$ is at most the number of rounds before t_0 which is $\leq r \in O(n^2 m^2 \log m)$ and the number of rounds after t_0 which is also $\leq r \in O(n^2 m^2 \log m)$. □

J Full proofs of results

J.1 Proof of Lemma F.5

Proof of Lemma F.5, part (1). For the sake of contradiction, assume that there are exists $\boldsymbol{x} \in \Delta(\mathcal{A})$ such that $\mathcal{P}_{R_1}(\boldsymbol{x})$ contains at least $m+1$ polytopes. We denote them with $b_0, b_1, \ldots, b_m$, where we also assume without loss of generality that $\boldsymbol{x} \in \mathrm{P}_{b_0}$ (i.e., b_0 is a best response to strategy $\boldsymbol{x}$).

We will first show that for all $i \in [m]$, we have $0 \leq \boldsymbol{h}_{b_0,b_i}^{\mathsf{T}}\boldsymbol{x} \leq \sqrt{m}$. Since $\mathrm{dist}(\boldsymbol{x}, \mathrm{P}_{b_i}) \leq$, there exists $\boldsymbol{z}_i \in \mathrm{P}_{b_i}$ such that $\|\boldsymbol{x} - \boldsymbol{z}_i\|_2 \leq$. This implies

$$U_2(\boldsymbol{x}, b_0) - U_2(\boldsymbol{z}_i, b_0) = \langle \boldsymbol{u}_{b_0}, \boldsymbol{x} - \boldsymbol{z}_i \rangle \leq \|\boldsymbol{u}_{b_0}\|_2 \cdot \|\boldsymbol{x} - \boldsymbol{z}_i\|_2 \leq \sqrt{m}.$$

Therefore, we can bound the inner product of $\boldsymbol{x}$ and the hyperplane $\boldsymbol{h}_{b_0,b_i} = \boldsymbol{u}_{b_0} - \boldsymbol{u}_{b_i}$ as

$$\langle \boldsymbol{h}_{b_0,b_i}, \boldsymbol{x} \rangle = U_2(\boldsymbol{x}, b_0) - U_2(\boldsymbol{x}, b_i) = \underbrace{U_2(\boldsymbol{x}, b_0) - U_2(\boldsymbol{z}_i, b_0)}_{\leq \sqrt{m}} + \underbrace{U_2(\boldsymbol{z}_i, b_0) - U_2(\boldsymbol{z}_i, b_i)}_{\leq 0 \text{ since } b_i \in \mathsf{BR}(\boldsymbol{z}_i)} \leq \sqrt{m}.$$

We also know $\langle \boldsymbol{h}_{b_0,b_i}, \boldsymbol{x} \rangle \geq 0$ since $b_0 \in \mathsf{BR}(\boldsymbol{x})$. As a result, we have $|{\boldsymbol{h}_{b_0,b_i}}^{\mathsf{T}}\boldsymbol{x}| \leq \sqrt{m}$.

Finally, we establish a contradiction using the following matrix $K \in \mathbb{R}^{m \times m}$, which is a square sub-matrix of the augmented constraint matrix G_{b_0}.

$$K = \left[\boldsymbol{h}_{b_0,b_i}\right]_{i \in [m]}.$$

Combing the above bound on $|{\boldsymbol{h}_{b_0,b_i}}^{\mathsf{T}}\boldsymbol{x}|$ and the fact that $\|x\|_2 \geq \frac{\|\boldsymbol{x}\|_1}{\sqrt{m}} = 1/\sqrt{m}$, we have

$$\frac{\|K\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \leq \sqrt{m} \cdot \sqrt{\sum_{i=1}^{m} |{\boldsymbol{h}_{b_0,b_i}}^{\mathsf{T}}\boldsymbol{x}|^2} \leq \sqrt{m} \cdot \sqrt{\sum_{i=1}^{m} (\sqrt{m})^2} = m^{2/3} = \frac{\underline{\sigma}}{2},$$

which contradicts with the assumption that

$$\sigma_{\min}(K) = \min_{\boldsymbol{x} \neq 0} \frac{\|K\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \geq \underline{\sigma}.$$

Therefore, we conclude that there cannot be more than m polytopes in $\mathcal{P}(\boldsymbol{x})$ where $= \frac{\underline{\sigma}}{2m^{2/3}}$. □

Proof of Lemma F.5, part (2). The second claim follows from a very similar proof. Suppose for the sake of contradiction that there exists a submatrix $K \in \mathcal{S}_m(G_b)$ of m conditions, and an $\boldsymbol{x} \in \mathrm{P}_b \subseteq \Delta(\mathcal{A})$, such that $\boldsymbol{x}$ satisfies all constraints in K with margin at most R_1, then we have

$$\|K\boldsymbol{x}\|^2 \le \sqrt{m \cdot R_1^2} = \sqrt{m}R_1 \le mR_1\|\boldsymbol{x}\|_2,$$

which contradicts with $\sigma_{\min}(K) \ge \underline{\sigma}$ for $R_2 = \frac{\underline{\sigma}}{2m}$. Therefore, there can be at most $m-1$ constraints from G_b that are satisfied by a margin of at most R_2. □

J.2 Proof of Lemma H.1

Lemma H.1 (Closeness of optimal values within true and estimated polytopes)**.** *Let $H, \hat{H} \in \mathbb{R}^{k\times m}$ with $k \le n$ be the true and estimated constraints, which satisfy $\|H - \hat{H}\|_2 \le \alpha\sqrt{m}$, where α is the estimation error that can be chosen sufficiently small. Consider polytopes P_b and $\hat{\mathrm{P}}_b$ defined as follows:*

$$\mathrm{P} = \{\boldsymbol{x} \in \mathbb{R}^m : Hx \ge 0, \mathbf{1}^\mathsf{T}\boldsymbol{x} = 1, \boldsymbol{x} \ge \mathbf{0}\}, \quad \hat{\mathrm{P}} = \{\boldsymbol{x} \in \mathbb{R}^m : \hat{H}x \ge \alpha, \mathbf{1}^\mathsf{T}\boldsymbol{x} = 1, \boldsymbol{x} \ge \gamma \cdot \mathbf{1}\}$$

Then, when maximizing the principal's utility $U_1(\cdot, b)$ over P_b and $\hat{\mathrm{P}}_b$, the corresponding optimal values satisfy

$$\max_{\boldsymbol{x}\in\hat{P}} U_1(\boldsymbol{x}, b) \le \max_{\boldsymbol{x}\in P} U_1(\boldsymbol{x}, b) + \frac{2\alpha m}{\underline{\sigma} - \alpha\sqrt{m}} + 2\gamma/R_{\min},$$

where α is chosen to be at most $O\left(\frac{\underline{\sigma}}{m^2 n}\right)$, $\underline{\sigma}$ is the lower bound on minimum singular values from Assumption F.2, $R_{\min}$ is the parameter from Assumption F.3, and $R_2 = \underline{\sigma}/m$ is from Lemma F.5.

We prove this lemma by showing that the ℓ_1 Hausdorff distance between P and $\hat{\mathrm{P}}$ are close, i.e., for any $\boldsymbol{x} \in \mathrm{P}$, we will construct a $\boldsymbol{z} \in \mathbb{R}^m$ such that $\boldsymbol{x} + \boldsymbol{z} \in \hat{\mathrm{P}}$ and $\|\boldsymbol{z}\|_1$ is small.

This will automatically imply the statement of the Lemma since the principal's utilities are 1-Lipschitz in ℓ_1 distance within the polytope. The utility within a polytope due to a strategy $\boldsymbol{x}$ is $\langle \mathbf{u}_b, \boldsymbol{x}\rangle$. The absolute difference in utilities between two strategies is

$$|U_1(\boldsymbol{x}, b) - U_2(\boldsymbol{y}, b)| = |\langle \mathbf{u}_b, \boldsymbol{x}\rangle - \langle \mathbf{u}_b, \boldsymbol{y}\rangle| \le \|\mathbf{u}_b\|_\infty \|\boldsymbol{x} - \boldsymbol{y}\|_1 \le \|\boldsymbol{x} - \boldsymbol{y}\|_1$$

We bound the Hausdorff distance between P_b and $\hat{\mathrm{P}}_b$ by the sum of the Hausdorff distances between P_b and $\tilde{\mathrm{P}}_b$ and between $\tilde{\mathrm{P}}_b$ and $\hat{\mathrm{P}}_b$, where

$$\tilde{\mathrm{P}} = \{\boldsymbol{x} \in \mathbb{R}^m : H\boldsymbol{x} \ge 0, \mathbf{1}^\mathsf{T}\boldsymbol{x} = 1, \boldsymbol{x} \ge \gamma \cdot \mathbf{1}\}$$

Let us start by bounding the Hausdorff distance between P and $\tilde{\mathrm{P}}$.

Lemma J.1. *Assume $\gamma \le R_{\min}$ where $R_{\min}$ is the parameter from Assumption F.3. The Hausdorff distance between P and $\tilde{\mathrm{P}}$ is upper bounded by $2\gamma/R_{\min}$. In other words, for every $\boldsymbol{x} \in \mathrm{P}$, there is a $\boldsymbol{y} \in \tilde{\mathrm{P}}$ with $\|\boldsymbol{x} - \boldsymbol{y}\|_1 \le 2\gamma/R_{\min}$.*

Proof. By Assumption F.3, there is a point $\boldsymbol{x}_0 \in \mathrm{P}$ that satisfies $\boldsymbol{x} \ge R_{\min} \cdot \mathbf{1}$. Therefore, for any $\boldsymbol{x} \in \mathrm{P}$ and $\lambda \in [0, 1]$, we construct $\boldsymbol{y}_\lambda$ as follows: $\boldsymbol{y}_\lambda = \lambda\boldsymbol{x}_0 + (1 - \lambda)\boldsymbol{x}$. Clearly, $\boldsymbol{y}_\lambda$ lies in P by the convexity of polytopes. Moreover, since $\boldsymbol{x} \ge 0$, we have $\boldsymbol{y}_\lambda \ge \lambda\boldsymbol{x}_0 \ge (\lambda R_{\min}) \cdot \mathbf{1}$. We can thus choose $\lambda = \gamma/R_{\min}$ to ensure $\boldsymbol{y}_\lambda \in \tilde{\mathrm{P}}$. Finally, for the distance between $\boldsymbol{x}$ and $\boldsymbol{y}_\lambda$, we have $\|\boldsymbol{x} - \boldsymbol{y}_\lambda\|_1 = \lambda\|\boldsymbol{x} - \boldsymbol{x}_0\|_1 \le 2\lambda = 2\gamma/R_{\min}$. □

Next, let us bound the Hausdorff distance between $\tilde{\mathrm{P}}$ and $\hat{\mathrm{P}}$.

Lemma J.2. *When $\alpha \ll \frac{\underline{\sigma}}{m^2 n}$, the Hausdorff distance between $\tilde{\mathrm{P}}$ and $\hat{\mathrm{P}}$ is upper bounded by $2\alpha\underline{\sigma}$. In other words, for every $\boldsymbol{x} \in \tilde{\mathrm{P}}$, there is a $\boldsymbol{y} \in \hat{\mathrm{P}}$ with $\|\boldsymbol{x} - \boldsymbol{y}\|_1 \leq 4\alpha m/\underline{\sigma}$.*

Proof. Given any $\boldsymbol{x} \in \tilde{\mathrm{P}}$, we aim to construct $\boldsymbol{z}$ with small norm such that $\boldsymbol{x} + \boldsymbol{z} \in \hat{\mathrm{P}}$. To make sure that $\boldsymbol{x} + \boldsymbol{z} \in \hat{\mathrm{P}}$, we want $\boldsymbol{z}$ to satisfy the following two constraints: (1) $\hat{H}(\boldsymbol{x} + \boldsymbol{z}) \geq \alpha \cdot \mathbf{1}_k$; (2) $\mathbf{1}^\mathsf{T}(\boldsymbol{x} + \boldsymbol{z}) = 1$, i.e., $\mathbf{1}^\mathsf{T}\boldsymbol{z} = 0$.

Let $J \subseteq [k]$ be the index of rows in H that are satisfied by $\boldsymbol{x}$ by less than a R_2 margin, where R_2 is the parameter from Lemma F.5. Equivalently, H_J contains all rows $\boldsymbol{h}$ such that $0 \leq \boldsymbol{h}^\mathsf{T}\boldsymbol{x} \leq R_2$. Lemma F.5 guarantees that $|J| \leq m - 1$. Let $H_{\setminus J}$ denote all other rows. That is, $\mathbf{0} \leq H_J\boldsymbol{x} \leq R_2 \cdot \mathbf{1}$, and $H_{\setminus J}\boldsymbol{x} \geq R_2 \cdot \mathbf{1}$.

We will first find a z that satisfies constraint (2) and constraint (1), but only restricted to rows in J, i.e., $\hat{H}_J(\boldsymbol{x} + \boldsymbol{z}) \geq \alpha \cdot \mathbf{1}_{|J|}$. Defining $E_J = H_J - \hat{H}_J$ to be the error matrix of rows in J, and define $\hat{M} = \begin{pmatrix} \hat{H}_J \\ \mathbf{1}_m \end{pmatrix}$, we set $\boldsymbol{z}$ as follows:

$$\boldsymbol{z} = \hat{M}^\dagger \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix} = \hat{M}^\mathsf{T}(\hat{M}\hat{M}^\mathsf{T})^{-1} \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix}.$$

In the remainder of the proof, we will first show that $\boldsymbol{z}$ satisfies both constraints (1) and (2), then upper bound the norm $\|\boldsymbol{z}\|_2$.

$\boldsymbol{z}$ satisfies constraints (2) and (1) restricted to J. By our construction of $\boldsymbol{z}$, we have

$$\hat{M}\boldsymbol{z} = \hat{M}\hat{M}^\dagger \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix} = \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix}.$$

Therefore, the vector $\boldsymbol{x} + \boldsymbol{z}$ satisfies

$$\hat{M}(\boldsymbol{x} + \boldsymbol{z}) = \begin{pmatrix} \hat{H}_J(\boldsymbol{x} + \boldsymbol{z}) \\ \mathbf{1}^\mathsf{T}(\boldsymbol{x} + \boldsymbol{z}) \end{pmatrix} = \begin{pmatrix} \hat{H}_J\boldsymbol{x} \\ \mathbf{1}^\mathsf{T}\boldsymbol{x} \end{pmatrix} + \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix} = \begin{pmatrix} H_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 1 \end{pmatrix}$$

As a result, we have $\hat{H}(\boldsymbol{x} + \boldsymbol{z}) \geq \alpha\mathbf{1}$ (followed from $H_J\boldsymbol{x} \geq \mathbf{0}$), $\mathbf{1}^\mathsf{T}(\boldsymbol{x} + \boldsymbol{z}) = 1$, and $(\boldsymbol{x} + \boldsymbol{z}) \geq \gamma\mathbf{1}$.

$\boldsymbol{z}$ satisfies constraints (1) outside of J Now we will show that the $\boldsymbol{z}$ found above satisfies $\hat{H}_{\setminus J}(\boldsymbol{x} + \boldsymbol{z}) \geq \alpha$ using the fact that $\boldsymbol{x}$ satisfied $H_{\setminus J}\boldsymbol{x} \geq R_2 \cdot \mathbf{1}$.

$$\begin{aligned}
\hat{H}_{\setminus J}(\boldsymbol{x} + \boldsymbol{z}) &= (H_{\setminus J} + E_{\setminus J})(\boldsymbol{x} + \boldsymbol{z}) && \text{(where } E_{\setminus J} = \hat{H}_{\setminus J} - G_{\setminus J}\text{)} \\
&\geq R_2 \cdot \mathbf{1} + E_{\setminus J}(\boldsymbol{x} + \boldsymbol{z}) + H_{\setminus J}\boldsymbol{z} && (H_{\setminus J}\boldsymbol{z}\boldsymbol{x} \geq R_2\mathbf{1}) \\
&\geq \big(R_2 - \|E_{\setminus J}(\boldsymbol{x} + \boldsymbol{z})\|_\infty - \|H_{\setminus J}\boldsymbol{z}\|_\infty\big) \cdot \mathbf{1} \\
&\geq \big(R_2 - \|E_{\setminus J}\|_\infty\|\boldsymbol{x} + \boldsymbol{z}\|_1 - \sqrt{m+n}\|H_{\setminus J}\|_1\|\boldsymbol{z}\|_2\big) \cdot \mathbf{1} \\
&\geq \Big(R_2 - \alpha - \sqrt{mn(m+n)} \cdot \|\boldsymbol{z}\|_2\Big) \cdot \mathbf{1}
\end{aligned}$$

The constraints in $\hat{H}_{\setminus J}$ are satisfied when $\|\boldsymbol{z}\|_2 \leq \frac{R_2 - \alpha}{\sqrt{mn(m+n)}}$, which we will show next.

Upper bounding $\|\boldsymbol{z}\|_2$. Since $\boldsymbol{z} = \hat{M}^\dagger \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix}$, we have

$$\|\boldsymbol{z}\|_2 \leq \|\hat{M}^\dagger\|_2 \cdot \left\| \begin{pmatrix} E_J\boldsymbol{x} + \alpha\mathbf{1}_{|J|} \\ 0 \end{pmatrix} \right\|_2 \leq \frac{2\alpha\sqrt{m}}{\sigma_{\min}(\hat{M})} \leq \frac{2\alpha\sqrt{m}}{\underline{\sigma} - \alpha\sqrt{m}} \leq \frac{4\alpha\sqrt{m}}{\underline{\sigma}},$$

where the last step is due to the choice of sufficiently small α, and the second-last step is due to Weyl's inequality ($\sigma_{\min}(\hat{M}) \geq \sigma_{\min}(M) - \|M - \hat{M}\|_2$) and that $\sigma_{\min}(M) \geq \underline{\sigma}$, as M is a submatrix of G_b with at most m rows (Assumption F.2). This shows that we can choose α to be small so that

$$\frac{2\alpha\sqrt{m}}{\underline{\sigma} - \alpha\sqrt{m}} \leq \frac{R_2 - \alpha}{\sqrt{mn(m+n)}} \quad \Rightarrow \quad \hat{H}_{\setminus J}(\boldsymbol{x} + \boldsymbol{z}) \geq 0.$$

The above condition holds when $\alpha = o(\frac{\underline{\sigma}}{m^2 n})$. To conclude this part, we translate our ℓ_2 norm bound on $\boldsymbol{z}$ into an ℓ_1 norm bound of $\|z\|_1 \leq \sqrt{m}\|z\|_2 \leq 4\alpha m/\underline{\sigma}$. □

Combining both lemmas, we obtain an upper bound on the ℓ_1 Haussdorf distance between P and $\hat{\text{P}}$ which due to the 1-Lipschitzness of utility in ℓ_1 norm bounds the difference in optimal utility in P and $\hat{\text{P}}$. The proof is thus complete.

J.3 Proof of Lemma H.3

Lemma H.3 (Binary search to get close to the boundary)**.** *Let γ be the minimum distance from all search point to the boundary in Lemma H.1. Suppose the principal's average strategy moves from $\overline{\boldsymbol{x}}^{(t-1)}$ in polytope P_b to $\overline{\boldsymbol{x}}^{(t)}$ in a different polytope $\text{P}_{b'}$, with ℓ_1 step size of $\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1 = \frac{\varepsilon}{t}$, where we assume $t \geq 10\frac{\varepsilon}{\gamma}$. Then the procedure* BinarySearch *(see Algorithm 5) takes at most $s \leq O\Big(\frac{\varepsilon}{\gamma} + \log\frac{\varepsilon}{\alpha t}\Big)$. steps and returns two consecutive average strategies $\overline{\boldsymbol{x}}^{(t+s-1)}$, $\overline{\boldsymbol{x}}^{(t+s)}$, such that they are on the opposite sides of the boundary (one is in P_b, the other is in $\text{P}_{b'}$), and they are close in ℓ_1 distance, i.e., $\|\overline{\boldsymbol{x}}^{(t+s-1)} - \overline{\boldsymbol{x}}^{(t+s)}\|_1 \leq \alpha$.*

ALGORITHM 5: BinarySearch

```
Input: Left point $\boldsymbol{x}_L = \overline{\boldsymbol{x}}^{(t-1)} \in \text{P}_b$, Right point $\boldsymbol{x}_R = \overline{\boldsymbol{x}}^{(t)} \in \text{P}_{b'}$, Step-size parameters $\varepsilon, \gamma, \alpha$.
Output: Two consecutive points $\big(\overline{\boldsymbol{x}}^{(t+s-1)}, \overline{\boldsymbol{x}}^{(t+s)}\big)$ on opposite sides of the boundary, with
        $\ell_1$-distance at most $\alpha$.
// Perform a binary search along the line segment between $\boldsymbol{x}_L$ and $\boldsymbol{x}_R$.
$M \leftarrow \log_2(\varepsilon/(\alpha t))$;
for each binary search step $i \leq M$ do
    $\boldsymbol{z}^{(i)} \leftarrow$ target search point chosen by the binary search algorithm;
    If moving left, set $\beta \leftarrow \gamma$; if moving right, set $\beta \leftarrow \varepsilon$;
    Keep performing MoveOneStep with step size $(\beta + \frac{\varepsilon}{t2^i})$ towards $\boldsymbol{z}^{(i)}$;
end
```

Proof of Lemma H.3. Let us refer to $\overline{\boldsymbol{x}}^{(t)}$ as $\boldsymbol{x}_L$ and $\overline{\boldsymbol{x}}^{(t-1)}$ as $\boldsymbol{x}_R$. Let us also refer to the direction from $\boldsymbol{x}_L$ to $\boldsymbol{x}_R$ as *right* and the direction from $\boldsymbol{x}_R$ to $\boldsymbol{x}_L$ as *left.* Throughout the binary search process, we move along the line between $\boldsymbol{x}_L$ and $\boldsymbol{x}_R$. For notational simplicity denote $l = \|\boldsymbol{x}_L - \boldsymbol{x}_R\|_1 = \varepsilon/t$.

Let M be the number of iterations of binary search. Since each iteration halves the search space, we have $M \leq \log\left(\frac{\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1}{\alpha}\right) = O(\log(\frac{\varepsilon}{\alpha t}))$.

For $i \in [M]$, we denote the number of rounds spent in the i-th iteration as $[t_i, t_i + s_i]$, where we have $t_1 = t$ and $t_{i+1} = t_i + s_i + 1$. The total number of rounds is therefore $s = \sum_{i=1}^{M}(s_i + 1) = M + \sum_i s_i$. From the halving property of binary search, we know that in iteration i, the total distance moved is at most $\frac{\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t-1)}\|_1}{2^i}$, i.e.,

$$\|\overline{\boldsymbol{x}}^{(t_i)} - \overline{\boldsymbol{x}}^{(t_i+s_i)}\|_1 = \frac{l}{2^i}.$$

On the other hand, at each round $\tau \in [t_i, t_i + s_i)$, the average strategy $\overline{\boldsymbol{x}}^{(t)}$ has at least $l/2^i + \min\{\varepsilon, \gamma\}$ distance to the direction it is moving to (left or right). From Remark G.1 and the BinarySearch algorithm, the maximum ℓ_1 distance that the average strategy can travel from $\overline{\boldsymbol{x}}^{(\tau-1)}$ to $\overline{\boldsymbol{x}}^{(\tau)}$ satisfies

$$\|\overline{\boldsymbol{x}}^{(\tau)} - \overline{\boldsymbol{x}}^{(\tau-1)}\|_1 \leq \frac{\frac{l}{2^i} + \min\{\varepsilon, \gamma\}}{\tau}.$$

In fact, in each step $\tau < t_i + s_i$, the principal travels exactly this distance in order to minimize the total number of steps.

Therefore, the total movement in iteration i satisfies

$$\frac{l}{2^i} = \|\overline{\boldsymbol{x}}^{(t_i)} - \overline{\boldsymbol{x}}^{(t_i+s_i)}\|_1 \geq \sum_{\tau=t_i}^{t_i+s_i-1} \frac{\frac{l}{2^i} + \min\{\varepsilon, \gamma\}}{\tau} \geq \left(\frac{l}{2^i} + \min\{\varepsilon, \gamma\}\right) \cdot \log\left(\frac{t_i + s_i}{t_i}\right) \quad (t_i \leq t + s)$$

We can use the above inequality to upper bound s_i as follows:

$$\begin{aligned} s_i &\leq t_i \left(\exp\left(\frac{\frac{l}{2^i}}{\frac{l}{2^i} + \min\{\varepsilon, \gamma\}}\right) - 1\right) \\ &\leq t_i \cdot \frac{l}{2^{i-1} \cdot \min\{\varepsilon, \gamma\}} && (e^x - 1 \leq 2x \text{ when } x \in [0, 1]) \\ &\leq (t + s) \cdot \frac{l}{2^{i-1} \cdot \min\{\varepsilon, \gamma\}}. \end{aligned}$$

Recall that the total number of steps is $s = \sum_i s_i + M$. Summing the above inequality over $i \in [M]$ gives us

$$\begin{aligned} & s = \sum_i s_i + M \leq 2(t + s) \cdot \frac{l}{\min\{\varepsilon, \gamma\}} + M \\ \Rightarrow \quad & s \lesssim \frac{\varepsilon}{\gamma} + \log\left(\frac{\varepsilon}{\alpha t}\right). && \left(\text{From the assumption } t \geq 10\tfrac{\varepsilon}{\gamma}\right) \end{aligned}$$

The proof is thus complete. □

J.4 Proof of Theorem 3.4

Theorem 3.4 (Correctness of SearchForPolytopes)**.** *Starting from a point $\boldsymbol{x}^*$ in P_b, for any $\alpha \in (0, o(R_2\underline{\sigma}/m^3))$ and $\rho < \alpha$,* SearchForPolytopes *finds $\hat{\boldsymbol{h}}_{b,b'}$ such that $\|\hat{\boldsymbol{h}}_{b,b'} - h_{b,b'}\|_2 \leq \alpha$, for every $b' \in \mathcal{P}_\rho(\boldsymbol{x}^*)$.*

In this section, we present the full proof of Theorem 3.4 by expanding the proof sketch in Appendix I.1.

Let J of size j index the polytopes already discovered in previous iterations. We will analyze the iteration of the algorithm with the search space $\hat{\mathcal{S}}_J = \{\boldsymbol{x} \in \Delta(\mathcal{A}) : \hat{H}_J \boldsymbol{x} \geq \alpha_j\}$, where $\hat{H}_J$ is the matrix of approximations to the hyperplanes discovered thus far. Inductively we will show that if $\|\hat{h}_{b,b_j} - h_{b,b_j}\| \leq \alpha_j$ for every $j \in J$, the next constructed hyperplane satisfies $\|\hat{h}_{b,b_{\text{new}}} - h_{b,b_{\text{new}}}\| \leq \alpha_j O(\alpha_j^2 m^2/\underline{\sigma}^2)$. We will later show how to set each α_j so they all remain less than α.

Lemma I.1 (Algorithm terminates after $m-1$ hyperplanes added). *The* SearchForPolytopes *algorithm will terminate after at most $m-1$ hyperplanes are added.*

Proof of Lemma I.1. We will first argue that $|J| \leq m-1$. By Lemma F.5, the set $\{\boldsymbol{x} \in \Delta(\mathcal{A}) : \|H_J \boldsymbol{x}\|_\infty \leq R_2\}$ is empty if $|J| \geq m$. For any point in $\{\boldsymbol{x} \in \Delta(\mathcal{A}) : \hat{H}_J x = 0\}$, $\|H_J \boldsymbol{x}\|_\infty \leq \sqrt{m}\|\hat{H}_J x\|_2 + \sqrt{m}\|(H_J - \hat{H}_J)\|_2 \|x\|_2 \leq m\sqrt{m}\alpha$. As long as $m\sqrt{m}\alpha$, this implies that the set $\{\boldsymbol{x} \in \Delta(\mathcal{A}) : \hat{H}_J x = 0\}$ is empty for $|J| \geq m$. □

Lemma J.3 (Projection of search point is close to search point). *The search point $\hat{\boldsymbol{x}}_J$ of the iteration is close to the target search point $\boldsymbol{x}^*$, where $\hat{\boldsymbol{x}}_J$ is the projection of $\boldsymbol{x}^*$ on to the search space $\hat{\mathcal{S}}_J$.*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \in O\left(\frac{\alpha_j m^2}{\sigma_{\min}(H_J) - \alpha_j\sqrt{m}}\right).$$

Proof of Lemma J.3.

$$\begin{aligned}
\|\hat{\boldsymbol{x}}_J - \boldsymbol{x}^*\| &= \|\left(I - \hat{H}_J^t(\hat{H}_J\hat{H}_J^t)^{-1}\hat{H}_J\right)\boldsymbol{x}^* - \boldsymbol{x}^*\| \\
&= \|\hat{H}_J^t(\hat{H}_J\hat{H}_J^t)^{-1}\hat{H}_J\boldsymbol{x}^*\| \\
&= \|\hat{H}_J^t(\hat{H}_J\hat{H}_J^t)^{-1}H_J\boldsymbol{x}^* + \hat{H}_J^t(\hat{H}_J\hat{H}_J^t)^{-1}(\hat{H}_J - H_J)\boldsymbol{x}^*\| \\
&= \|\hat{H}_J^t(\hat{H}_J\hat{H}_J^t)^{-1}\| \left(\|H_J\boldsymbol{x}^*\| + \|(\hat{H}_J - H_J)\boldsymbol{x}^*\|\right) \\
&\leq m\frac{1}{\sigma_{\min}(H_J) - \alpha_j\sqrt{m}}\left(\alpha_j\sqrt{m} + \alpha m\right) \\
&\in O\left(\frac{\alpha_j m^2}{\underline{\sigma}}\right).
\end{aligned}$$

□

Lemma J.4. *For all points on $\hat{\mathcal{S}}_J$ the agent is approximately indifferent among agent actions in J. That is for every $i, j \in J$, for all $\boldsymbol{x} \in \hat{\mathcal{S}}_J$, $|U_2(\boldsymbol{x}, b_i) - U_2(\boldsymbol{x}, b_j)| \leq 4\alpha_j m$.*

Proof of Lemma J.4. Note that for all points on $\hat{\mathcal{S}}_J = \{\boldsymbol{x} \in \Delta(\mathcal{A}) : H_J \cdot \boldsymbol{x} = 0\}$, the principal is indifferent among follower actions in J since each row $j \in [J]$ equality states that the utility due to agent action b_j is the same as due to b.

Recall that for any $b \in \mathcal{B}$, $U_1(\boldsymbol{x}, b) = \langle \mathbf{u}_b, \boldsymbol{x}\rangle$. For any $\hat{\boldsymbol{x}} \in \{x : \hat{H}_J \cdot x = \alpha_j\}$, for any $i, j \in J$,

$$\begin{aligned}
|\left\langle \mathbf{u}_{b_i}, \hat{\boldsymbol{x}}\right\rangle - \left\langle \mathbf{u}_{b_j}, \hat{\boldsymbol{x}}\right\rangle| &\leq |h_i\hat{x} - h_j\hat{\boldsymbol{x}}| \\
&\leq 2\|H_J\hat{\boldsymbol{x}}\|_\infty \\
&\leq 2\|H_J\hat{\boldsymbol{x}}\|_2 \\
&\leq 2\|\hat{H}_J\hat{\boldsymbol{x}} + (H_J - \hat{H}_J)\hat{\boldsymbol{x}}\|_2 \\
&\leq 2\alpha_j(\sqrt{m} + m).
\end{aligned}$$

□

Now we will discuss how random search on the search space via Gaussian random vectors will discover new polytopes in $\mathcal{P}_\delta(\boldsymbol{x}^*)$ that are not yet discovered.

There are two components to this argument. The first is while there is an undiscovered polytope in $\mathcal{P}_\delta(\boldsymbol{x}^*)$, we will discover a new polytope. The second is that there are not many other polytopes that are discoverable. Together these two components ensure we discover all polytopes in $\mathcal{P}_\delta(\boldsymbol{x}^*)$.

The first component is shown in the following lemma.

Lemma J.5. *Let z be the random search vector in the space $\hat{\mathcal{S}}_J$ with step size η. That is, $z \sim \mathcal{N}(\mathbf{0}^m, \eta_j^2 \phi_J^t \phi_J)$. If $J \not\supseteq \mathcal{P}_\delta(\boldsymbol{x}^*)$ and $\eta_j \in \Theta(\alpha_j m^4/\underline{\sigma}^2)$, then with $\Omega(1)$ probability, $\hat{x}_J + z$ does not lie in P_b or in any polytope $\mathrm{P}_{b'}$ for $b' \in [J]$.*

Proof. The main idea of this proof is that if there is an undiscovered polytope $b' \in \mathcal{P}_\alpha(\boldsymbol{x}^*)$, then

$$\Pr_z \left[h_{b,b'}(\hat{x}_J + z) > \frac{\sigma_{\min}(H_J)}{\sqrt{m-1}} \cdot \eta - O\left(\frac{\alpha m^3}{\sigma_{\min}(H_J)}\right) - O(\delta\sqrt{m}) \right] \geq \Omega(1).$$

This is a lower bound on how much that the principal prefers action b' over b for $\hat{x}_J + z$. Since the principal is approximately indifferent between b and any b_j for $j \in J$, this is also an approximate lower bound on the principal's preference over the new action b' over any already discovered action. This means that $\hat{x}_J + z$ lies in a polytope that has not already be discovered.

We will lower bound $h_{b,b'}(\boldsymbol{x}^* + z)$ and the resulting bound follows from applying the lemma showing closeness of $\hat{\boldsymbol{x}}_J$ and $\boldsymbol{x}^*$.

Let $j = |J|$. Let $\phi = [v_1, \ldots, v_m]^\mathsf{T}$ be a set of orthonormal vectors such that $v_{1:j}$ is the orthonormal basis of $\mathrm{span}(H_J)$ and $v_{j+1:m}$ is the orthonormal basis of $\mathrm{null}(H_J)$. Let $\boldsymbol{z}$ be a Gaussian vector in the null space of H_J, i.e., $z \sim \mathcal{N}(\mathbf{0}^m, \phi_{j+1:d}^\mathsf{T} \phi_{j+1:d})$.

For any $i \notin J$, we can decompose h_i into $h_i = h_i^\parallel + h_i^\perp$, where $h_i^\parallel \in \mathrm{span}(H_J)$ and $h_i^\perp \in \mathrm{null}(H_J)$. This decomposition gives

$$\langle h_i, x + z\rangle = \langle h_i, z\rangle = \left\langle h_i^\perp, z \right\rangle = \|h_i^\perp\|_2 \cdot \left\langle h_i^\perp / \|h_i^\perp\|_2, z \right\rangle.$$

We will prove this lemma by showing that (1) $\|h_i^\perp\|_2 \geq \sigma_{\min}(H_J)$; and (2) with constant probability, we have $\cos(h_i^\perp, z) = \langle h_i^\perp/\|h_i^\perp\|_2, z/\|z\|_2 \rangle < -\frac{1}{\sqrt{d-j}}$.

For the first claim, we further decompose $h_i^\perp = y_1 + y_2$, where $y_1 \in \mathrm{null}(H_J) \cap \mathrm{span}(H_{[n]\setminus J \setminus \{i\}})$, and $y_2 \in \mathrm{null}(H_J) \cap \mathrm{null}(H_{[d]\setminus J \setminus \{i\}}) = \mathrm{null}(H_{[n]\setminus\{i\}})$. We have

$$\begin{aligned}
\sigma_{\min}(H_J) &= \min_{y \in \mathbb{R}^m : \|y\|_2 = 1} \|H_J y\|_2 \\
&\leq \frac{1}{\|y_2\|_2} \cdot \|H_J y_2\|_2 && \text{(choose } y = y_2/\|y_2\|_2) \\
&= \frac{1}{\|y_2\|_2} \cdot |\langle h_i, y_2\rangle| && (y_2 \in \mathrm{span}(H_{[d]\setminus\{i\}})) \\
&= \frac{1}{\|y_2\|_2} \cdot |\langle y_2, y_2\rangle| && (y_1 \perp y_2) \\
&= \|y_2\|_2 \leq \|h_i^\perp\|_2. && (\|h_i^\perp\|_2 = \sqrt{\|y_1\|_2^2 + \|y_2\|_2^2})
\end{aligned}$$

For the second claim, we can expand $h_i^\perp/\|h_i^\perp\|$ into another set of orthonormal basis $v'_{1:d-j}$ of $\mathrm{null}(H_J)$ and write $z = \sum_{l=1}^{d-j} v'_l z'_l$ where $z'_{1:d-j} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. Therefore, we have

$$\cos(h_i^\perp, z) = \left\langle h_i^\perp/\|h_i^\perp\|_2, z/\|z\|_2 \right\rangle = \frac{z'_1}{\sqrt{(z'_1)^2 + \|z'_{2:d-j}\|_2^2}}.$$

Let $\delta = \frac{1}{\sqrt{d-j}}$, we have

$$\begin{aligned}\cos(h_i^\perp, z) < -\delta &\iff z'_1 < -\frac{\delta}{\sqrt{1-\delta^2}}\|z'_{2:d-j}\|_2 \\ &\Longleftarrow z'_1 < -1, \text{ and } \|z'_{2:d-j}\|_2 < \frac{\sqrt{1-\delta^2}}{\delta}.\end{aligned}$$

Since $z'_{1:d-j}$ are iid Gaussian, the probability of both events are lower bounded by constants, so we have

$$\Pr[\cos(h_i^\perp, z) < -\delta] \geq \Pr[z'_1 < -1] \cdot \Pr\left[\|z'_{2:d-j}\|_2 < \frac{\sqrt{1-\delta^2}}{\delta}\right] \geq \Omega(1).$$

Combining the two claims proves the lemma. □

Claim J.6. *With probability $\Omega(1)$ a search direction point $\boldsymbol{z}$ generated from the distribution $\mathcal{N}(\mathbf{0^m}, \eta_j^2 \phi_J^\top \phi_J)$ has $\|z\|_2 \leq O(\eta\sqrt{m})$.*

From the lemma and claim above, we get can show that a fraction $\Omega(1/m)$ of the search points generated lie in a new, undiscovered polytope $\mathrm{P}_{b_{\text{new}}}$ and lie within distance $\eta_j\sqrt{m}$ close $\hat{\boldsymbol{x}}_J$ which in turn is $O(\alpha_j m^3/\underline{\sigma})$ close to $h_{b,b_{\text{new}}}$ (and in fact any boundary in $\mathrm{P}_\alpha(\boldsymbol{x}^*)$). So the points in Y lie at a distance $O(\eta_j\sqrt{m} + \alpha_j m^3/\underline{\sigma})$. From the choice of $\eta_j \in \Theta(\alpha_j m^4/\sigma^2)$, this is $O(\alpha_j m^{4.5}/\underline{\sigma}^2)$.

Using these properties, we will now show that with a large enough number of search points, we can approximate $h_{b,b_{\text{new}}}$ well. We construct the approximation in the following way. Let Y be a matrix of the search points obtained by $\hat{\boldsymbol{x}}_H + \boldsymbol{z}_J$, where $\boldsymbol{z}_J$ is a Gaussian random vector in the search space $\hat{\mathcal{S}}_J$. We will construct a hyperplane solving $\hat{h} = \mathrm{argmin}_h \|hY^\top\|$. We will now show that $\hat{h}$ is a good approximation to $h_{b,b_{\text{new}}}$.

Lemma J.7 (Approximating a hyperplane)**.** *Suppose* RandomSearch *in the search space $\hat{\mathcal{S}}_J = \{\boldsymbol{x} : \hat{H}_J\boldsymbol{x} = \alpha_j\}$ generates $d \in \Theta(m^2\log m)$ points. Let b_{new} be the undiscovered polytope with the maximum number of search points. Let $Y \in \mathbb{R}^{k\times m}$ be a matrix where the rows are all search points in $\mathrm{P}_{b_{\text{new}}}$ with distance from $\hat{\boldsymbol{x}}_J \in O(\eta\sqrt{m})$.*

Then $\hat{h} = \mathrm{argmin}_h \|hY^t\|_2$ satisfies $\|\hat{h} - h_{b,b_{\text{new}}}\| \leq \alpha$.

We will prove the accuracy of $\hat{h}$ in approximating $h_{b,b_{\text{new}}}$ by finding and using a lower bound on the minimum singular value of the matrix Y.

Let us denote the matrix of all search directions generated by random search by $Z \in \mathbb{R}^{d\times m}$. That is each row $z_i \sim \mathcal{N}(\mathbf{0}, \eta_j^2 \phi_J^t \phi_J)$. Let $W \in \mathbb{R}^{k\times m}$ denote the subset of rows of Z corresponding to search vectors with length at most $O(\eta_j\sqrt{m})$ and lying in $\mathrm{P}_{b_{\text{new}}}$. W has a $\Omega(1/m)$ fraction of the rows in Z. The matrix of close-to-boundary search points in $\mathrm{P}_{b_{\text{new}}}$ is Y where each row y_i is $\hat{\boldsymbol{x}}_J + w_i$.

The minimum singular value of $Z \geq \eta$. Since W has $\Omega(1/m)$ fraction of subsets of Z, we can show that the minimum singular value of W is also lower bounded. We show that $\sigma_{\min}(W) \in \Omega(\eta_j \sqrt{m/d})$ which is $\Omega(\eta_j \sqrt{1/(m \log m)})$ by our choice of d.

Each row of Z is an isotropic random vector in a dimension $p \leq m$. Fix some $u \in \mathbb{R}^p$. Then, each entry of Zu is normally distributed $N(0,1)$, i.i.d. With high probability, at least an $1/m$-fraction of the entries will be at least $1/(2m)$ in absolute value, from concentration. Now, we take a union bound over an ϵ-Net over the set of all unit vectors. With high probability, this will hold for any vector in the net. If this holds for any vector of the net, we can say that this holds for all unit vectors and the proof would be complete.

Next note that $\sigma_{d-1}(Y) \geq \sigma_d(W)$ since $Y - W$ is a matrix of rank 1.

Armed with the property that $\sigma_{\min}(Y) \geq \Omega(\eta_j \sqrt{1/m \log m})$, we will now show that $\hat{h} = \operatorname{argmax}_h \|hY^t\|_2$ is a good approximation for $h_{b,b_{\text{new}}}$.

The true boundary $h_{b,b_{\text{new}}}$ also has a small $\|h_{b,b_{\text{new}}} Y^t\|_2 \leq O(\alpha_j m^{5.5}/\underline{\sigma}^2)$ since points in Y are close to the boundary. Let v denote the singular vector of Y^t corresponding to the smallest singular value, where $\|v\|_2 = 1$. Write $\hat{h} = \cos(\theta)v + \sin(\theta)u$ for u perpendicular to v. Then,

$$\|\hat{h}Y^t\|^2 = \cos^2(\theta)\|vY^t\|^2 + \sin^2(\theta)\|uY^t\|^2 \geq \sin^2(\theta)\sigma_{d-1}(Y).$$

Since $\|\hat{h}Y^t\|^2 \leq O(\alpha_j^2 m^9/\underline{\sigma}^4)$ is small, this means that $\sin(\theta)$ is small. $\sin^2(\theta) \leq O(\alpha_j m^6/\underline{\sigma}^2)$. Hence $\|\hat{h} - v\| \precsim 2\sin\theta \in O(\sqrt{\alpha_j} m^3/\underline{\sigma})$ is small. Similarly, $\|h_{b,b_{\text{new}}} - v\|$ is small. Hence $\|h_{b,b_{\text{new}}} - \hat{h}\| \in O(\sqrt{\alpha_j} m^3/\underline{\sigma})$ is small.

We have inductively shown how to find approximations of accuracy α_j for the j^{th} discovered hyperplane where $\alpha_{j-1} \in O(\alpha m^3/\underline{\sigma})\alpha_j$. To ensure that all hyperplanes are approximated to within α level, we set α_1 so that $\alpha_1(m^3/\underline{\sigma})^m \leq \alpha$ or $\alpha_1 \in O(\alpha \cdot (\underline{\sigma}/m^3))^m$.

Guarantee that all polytopes in $\mathcal{P}_\delta$ are discovered. Lemma J.5 shows that the algorithm keeps finding a new polytope as long as the search space is non-empty and some polytope in $\mathcal{P}_\delta(\boldsymbol{x}^*)$ is not yet visited. The only way the algorithm can terminate without finding some polytope in $\mathcal{P}_\delta(\boldsymbol{x}^*)$ is if it finds m polytopes that are not a superset of $\mathcal{P}_\delta(\boldsymbol{x}^*)$.

All the polytopes discovered have points at distance $\alpha_j m^3/\underline{\sigma}$ from $\boldsymbol{x}^*$ and hence are polytopes in $\mathcal{P}_{\alpha_j m^3/\underline{\sigma}}(\boldsymbol{x}^*)$. By choosing α_j's so that each $\alpha_j m^3/\underline{\sigma} \in \omega(R_2)$, we have the property that all the discovered polytopes belong to the set $\mathcal{P}_{R_2}(x^*)$. By Lemma F.5, there are at most m polytopes in $\mathcal{P}_{R_2}(x^*)$.

J.5 Proof of Theorem C.1

Theorem C.1 (Lower Bound)**.** *Assume a repeated game between a learner, who employs Fictitious Play, and an optimizer who does not know the learner's utility. Let n be the number of actions for the learner, and let $\epsilon \in (0, 1/3)$. Assume that $n \geq 1/\epsilon^2$. Let m be the number of actions for the optimizer and assume that $m \geq 1/\epsilon^5$. Then, there exists a distribution over games such that for any randomized algorithm used by the optimizer, the expected number of iterations required to find an ϵ-approximate local Stackelberg equilibrium is at least $e^{C/\epsilon}$, where $C > 0$ is a universal constant.*

Further, this lower bound holds under smoothed analysis, when each entry in the learner's utility matrix is perturbed by a small constant and when Assumption F.3 is satisfied.

Proof of Theorem C.1. Let ℓ be the largest integer such that $1/(2\ell+1) \geq \epsilon$. We can assume that the optimizer has $n = \ell^2$ actions and the learner has $m = \ell^3 + 1$ actions: if they have more actions then it is easy to guarantee that they are not played, by constructing a suitable game. We will prove the version without the smoothed-analysis. However, it is clear from the proof that it still holds even when the utilities are perturbed as claimed above.

Assume that the optimizer has n actions. We want to define the optimizer's utility such that, in order to find a local optima, a lot of time has to be passed until the optimizer succeeds. We notice that the optimizer's utility is defined on the simplex Δ_n, and it is piecewise linear: the simplex is split into polytopes, defined by the learner's best responses. For intuition, we explain now how these polytopes look like. First, there's some path of length ℓ, where $\ell < n$, consisting of distinct vertices $v_1 - \cdots v_\ell$, where each $v_i \in [n]$, and $v_1 = 1$. The optimizer doesn't know the path, and the only local optima of the function would be if the optimizer plays deterministically the action v_ℓ. The optimizer would have to find v_ℓ in order to find a local optima, and this will take a long time. Here is how we define the polytopes such that the only local optima is v_ℓ. First, for each edge in the path, there's some polytope: namely, for each $i = 1, \dots, \ell - 1$, there's a polytope around the edge connecting v_i with v_{i+1}. We denote this polytope P_i and it is approximately defined as

$$P_i = \{x \in \Delta_n \colon x_{v_i} + x_{v_{i+1}} \geq \max\{0.9, x_{v_{i-1}} + x_{v_i}, x_{v_{i+1}} + x_{v_{i+2}}\}\ \}$$

In the remainder of the the simplex, there will be the polytope P_0 defined as

$$P_0 = \{x \in \Delta(n) \colon \forall i \in [\ell - 1],\ x_{v_i} + x_{v_{i+1}} \leq 0.9\}$$

The optimizer's utility is defined such that for $i < j \in \{0, \dots, \ell\}$ it is higher in P_j compared to P_i. Within the polytopes the optimizer's utility is as follows: in P_0 the utility is higher as x approaches $v_1 = 1$ (i.e. approaches the pure action $v_1 = 1$). For $i > 1$, the utility is higher as x approaches v_{i+1}. Concretely, the learner's utility is defined as follows: In $x \in P_i$ it equals $(2i + x_{v_{i+1}})/(2\ell + 1)$. It is trivial to see that there's not $1/(2\ell+1)$-approximate local Stackalberg except for x in the vicinity of v_ℓ. Indeed, within each P_i the local opt is only v_{i+1}. And further, for each $i < \ell$, v_i is not a local opt because v_i borders the polytope P_i whose local optima is v_{i+1}. Consequently, the only local opt is v_ℓ.

Now, we define the optimizer's actions and the utility functions such that the optimizer's utility is as defined above. Recall that the optimizer knows their own utility, however, they don't know the path. Consequently, for any potential edge $v_i - v_{i+1}$ there should be a potential polytope. Some of the potential polytopes will not exist, however, the optimizer would not know that in advance. Concretely, for each "potential polytope" there would be an action of the learner. First, for the big polytope P_0, there will be an action of the learner that we denote as 0. Further, for any $r \in [n]$, $s \in [n] \setminus \{r\}$ and $i = 1, \dots, \ell$, we define an action of the learner (r, s, i). We want to ensure the following:

- If $v_i = r$ and $v_{i+1} = s$ then the polytope P_i will correspond to the optimizer playing action (r, s, i). Otherwise, this action will correspond to no polytope.
- If this action does correspond to P_i then we want the optimizer's utility to be as defined above.

To achieve the following, we first define the learner's utility. For action 0 of the learner, and any action a of the optimizer, define

$$u_2(a, 0) = 0\ .$$

This corresond's to the "default" polytope P_0. For any (r, s, i) such that $v_i \neq r$ or $v_{i+1} \neq s$, we want to ensure that there's no polytope corresponding to this action, hence we define

$$u_2(a, (r, s, i)) = -1$$

for any action a of the optimizer, and this ensures that the learner will always prefer action 0 over (r, s, i). Next, we want to define the learner's utility on (r, s, i) such that $v_i = r$ and $v_{i+1} = s$ such that the region when the learner plays (r, s, i) corresponds to the polytope P_i. We define the learner's utility as

$$u_2(a, (r, s, i)) = \begin{cases} 1/9 & a \in \{r, s\} \\ -1 & a \notin \{r, s\} \end{cases}$$

This definition of v_L yields the polytopes P_i as defined above, namely, P_0 is the region where the learner plays action 0 and for $i > 0$, P_i is exactly the region where learner plays (v_i, v_{i+1}, i).

Now, we need to define the optimizer's utility to align with the definitions above. Recall that we want the optimizer's utility to be $(x_{v_{i+1}} + 2i)/(2\ell + 1)$ in each region P_i. Starting at P_0, we define:

$$u_1(a, 0) = \begin{cases} 1/(2\ell + 1) & a = v_1 \\ 0 & a \neq v_1 \end{cases}$$

For $i > 0$, define

$$u_1(a, (v_i, v_{i+1}, i)) = \begin{cases} (1 + 2i)/(2\ell + 1) & a = v_{i+1} \\ 2i/(2\ell + 1) & a \neq v_{i+1} \end{cases}$$

For all (r, s, i),

$$u_1(a, (r, s, i)) = \begin{cases} (1 + 2i)/(2\ell + 1) & a = s \\ 2i/(2\ell + 1) & a \neq s \end{cases}$$

It is easy to see that this yields the correct utility function.

Now, we want to prove that it takes a long time for the optimizer to get to v_ℓ. First, recall that $\overline{\boldsymbol{x}}^{(t)}$ is the average optimizer's history till time t. Notice that $\|\overline{\boldsymbol{x}}^{(t)} - \overline{\boldsymbol{x}}^{(t+1)}\|_1 \leq 1/(t + 1)$. Denote by $e_1, \ldots, e_k$ the indices of the polytopes visited throught the algorithm, except for polytope P_0: e_1 is the first visited polytope, e_2 is the second etc. Denote by t_i the first time that e_i was visited such that $t_1 \geq 1$. We want to argue that $t_{i+3} \geq 1.8 t_i$. Indeed, notice that each P_i has two neighboring polytopes (except for P_0): P_{i-1} and P_{i+1}. Consequently, one of $P_{e_{i+1}}, P_{e_{i+2}}, P_{e_{i+3}}$ does not neighbor P_{e_i}. Denote by $0 = t_0 < t_1 < \cdots < t_k$ the iterations such that for $i \geq 1$: t_i is the first $t > t_{i-1}$ such that $\overline{\boldsymbol{x}}^{(t)}$ has some coordinate that's is greater than 0.45 for the first time, namely, such that there's some j such that $(\overline{\boldsymbol{x}}^{(t)})_j \geq 0.45$ and $\overline{\boldsymbol{x}}_j^{(t')} < 0.45$ for all $t' < t$. We want to prove that $t_{i+2} \geq 1.35 t_i$. To show that, first, by definition, either the total variation between $\overline{\boldsymbol{x}}^{(t_i)}$ and $\overline{\boldsymbol{x}}^{(t_{i+1}}$ is at least 0.175 or between $\overline{\boldsymbol{x}}^{(t_i)}$ and $\overline{\boldsymbol{x}}^{(t_{i+2}}$: indeed, in $\overline{\boldsymbol{x}}^{(t_i)}$, $\overline{\boldsymbol{x}}^{(t_{i+1}}$ and $\overline{\boldsymbol{x}}^{(t_{i+2}}$ there are distinct coordinates that surpass 0.45 for the first time: for one of $i + 1$ and $i + 2$ this coordinate has to have a weight of at most $0.55/2 = 0.275$ at time $t = t_i$, which implies a total variation of at least $0.45 - 0.275 = 0.175$. Further, notice that the total variation between $\overline{\boldsymbol{x}}^{(t)}$ and $\overline{\boldsymbol{x}}^{(t+1)}$ is at most $1/(t + 1)$ by definition of $\overline{\boldsymbol{x}}^{(t)}$ (it is the average of everything up to time t. Consequently, it takes at least $0.175 t_i$ iterations until reaching some $\overline{\boldsymbol{x}}^{(t)}$ whose total variation is 0.175 from t_i, which means that $t_{i+2} \geq 1.175 t_i$ as claimed. This implies that the number of iterations is $e^{\Omega(k)}$ and we would like to lower bound k.

Denote by u_i the coordinate that's at least 0.45 for the first time at t_i. We assume that the algorithm stops when reaching close to a local optima which is at v_ℓ hence there must be some $i \leq k$ such that $u_i = v_\ell$. Denote by I the set of indices i such that u_i is not on the path, namely

$$I = \{i \colon u_i \notin \{v_1, \ldots, v_\ell\}\}$$

Further, denote by J the set of times i that $u_i \neq v_1$ is on the path but no neighbor of u_i was visited before:

$$J = \{i \colon \exists j > 1 \text{ s.t. } u_i = v_j \text{ and } \forall i' < i, u_{i'} \notin \{v_{j-1}, v_{j+1}\}\}$$

Now, we consider I and J as random variables. The randomness is both wrt the randomness of the optimizer's algorithm and wrt to a random choise of a path $v_1 - \cdots - v_\ell$ which is taken uniformly at random from all paths starting at $v_1 = 1$ whose all vertices are distinct and whose elements are $[n]$. Whenever $J = \emptyset$, then, we have that $k \geq \ell$: indeed, recall that $v_\ell \in \{u_1, \ldots, u_k\}$ and if $J = \emptyset$ this means that all vertices $v_1, \ldots, v_{\ell-1}$ have to appear w.p. 0.45 before v_ℓ. So, if $\Pr[|J| = \emptyset \geq 0.5]$ then we have that the expected number of iterations run by our algorithm is at least $e^{\Omega(k)} \geq e^{\Omega(\ell)}$.

Now assume otherwise, that $\Pr[|J| \neq \emptyset] \geq 0.5$. In this case, w.p. 0.5, there's some node u_i such that no neighbor has appeared before w.p. 0.45. Since the path is uniformly at random, $\mathbb{E}[|I|] \geq \Omega(n/\ell)\mathbb{E}[|J|]$: whenever that algorithm visits a node that none of its neighbors appeared on the path, the probability that this new node is on the path is at most $O(n/\ell)$, assuming that $n > 10\ell$. Hence, in case that $\mathbb{E}[|J|] \geq 0.5$ we have that $\mathbb{E}[|I|] \geq \Omega(n/\ell)$ and $\mathbb{E}[k] \geq \Omega(n/\ell)$. If we assume that $n \geq \ell^2$, this is $e^{\Omega(\ell)}$, as required.

□

J.6 Proof of Theorem D.1

Theorem D.1 (Lower bound on the minimum singular value). *Let $\overline{\boldsymbol{U}}_{\boldsymbol{2}} \in [0,1]^{m \times n}$ be an arbitrary utility matrix of the agent, and let $\boldsymbol{U}_{\boldsymbol{2}}$ be a Gaussian perturbation of $\overline{\boldsymbol{U}}_{\boldsymbol{2}}$ with variance σ^2. Then the resulting augmented constraint matrices of the perturbed utility matrix satisfies that for $\underline{\sigma} = \Theta\left(\frac{\sigma\delta}{m^{\frac{5}{2}}2^n}\right)$, Assumption F.2 holds with probability at least $1 - \delta$.*

Proof of Theorem D.1. We first show that it suffices to establish a lower bound on the minimum singular values of all $k \times k$ submatrices of the *un-augmented constraint matrices* H_b, where $k \leq m$. In particular, we will show that $\forall b \in \mathcal{B}$,

$$\Pr\left(\forall K \in \mathcal{S}_m(G_b),\ \sigma_{\min}(K) \geq \underline{\sigma}\right) \geq \Pr\left(\forall K' \in \mathcal{S}_{\leq m}(H_b),\ \sigma_{\min}(K') \geq m \cdot \underline{\sigma}\right) \tag{1}$$

Let $G_b \backslash = \begin{bmatrix} I_m \\ \mathbf{1}_m \end{bmatrix}$ be the augmented constraints to account for the simplex constraints $\boldsymbol{x} \geq \mathbf{0}$ and $\mathbf{1}^{\mathsf{T}}\boldsymbol{x} = 1$. We establish Equation (1) by proving the following two claims:

1. If a square submatrix $K \in \mathcal{S}_m(G_b)$ contains r rows from the nonnegativity constraints I_m, then there exists a $(m-r) \times (m-r)$ submatrix $K' \in \mathcal{S}_{m-r}(K)$, such that $\sigma_{\min}(K) \geq \frac{1}{2}\sigma_{\min}(K')$;
2. For a submatrix $K \in \mathcal{S}_m(G_b)$ that contains the row $\mathbf{1}_m$, its $\sigma_{\min}(K)$ can be viewed as a Gaussian perturbed matrix.

Proof of the first claim. Without loss of generality, we can assume the nonnegativity constraints are located in the first r rows and first r columns of K, i.e., K takes the following form,

$$K = \begin{bmatrix} I_{r \times r} & \mathbf{0}_{r \times (m-r)} \\ L & K' \end{bmatrix},$$

where $L \in \mathbb{R}^{(m-r) \times r}$ and $K' \in \mathbb{R}^{(m-r) \times (m-r)}$ is a square sub-matrix of K. We will show that $\sigma_{\min}(K) \geq \frac{1}{m}\sigma_{\min}(K')$ by proving that for all m-dimensional vector $\boldsymbol{x} \in \mathbb{R}^m$ where $\|\boldsymbol{x}\|_2 = 1$, we have $\|K\boldsymbol{x}\|_2 \geq \frac{1}{m}\sigma_{\min}(K')$.

We can write $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}$, where $\boldsymbol{y} \in \mathbb{R}^r$ and $\boldsymbol{z} \in \mathbb{R}^{n-r}$. We have $K\boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ L\boldsymbol{y} + K'\boldsymbol{z} \end{bmatrix}$. Consider the following two cases:

- If $\|\boldsymbol{y}\|_2 \geq \frac{1}{m} \cdot \sigma_{\min}(K')$, then $\|K\boldsymbol{x}\|_2 = \|\boldsymbol{y}\|_2 + \|L\boldsymbol{y} + K'\boldsymbol{z}\|_2 \geq \|\boldsymbol{y}\|_2 \geq \frac{1}{m}\sigma_{\min}(K')$, as desired.
- If $\|\boldsymbol{y}\|_2 < \frac{1}{m} \cdot \sigma_{\min}(K')$, we have $\|\boldsymbol{z}\|_2 \geq 1 - \frac{1}{m}$. In this case,

$$\begin{aligned} \|K\boldsymbol{x}\|_2 &= \|\boldsymbol{y}\|_2 + \|L\boldsymbol{y} + K'\boldsymbol{z}\|_2 \geq \|\boldsymbol{y}\|_2 + \|K'\boldsymbol{z}\|_2 - \|L\boldsymbol{y}\|_2 \\ &\geq \sigma_{\min}(K') \cdot \|\boldsymbol{z}\|_2 - (\|L\|_2 - 1) \cdot \|\boldsymbol{y}\|_2 \\ &\geq \sigma_{\min}(K') \cdot (1 - \|\boldsymbol{y}\|_2) - \left(\frac{m}{2} - 1\right) \cdot \|\boldsymbol{y}\|_2 && (\|L\|_2 \leq \sqrt{r(m-r)} \leq \tfrac{m}{2}) \\ &\geq \sigma_{\min}(K') \cdot \left(1 - \frac{\sigma_{\min}(K')}{m} - \left(\frac{m}{2} - 1\right) \cdot \frac{1}{m}\right) && (\|\boldsymbol{y}\|_2 \leq \sigma_{\min}(K')/m) \\ &= \sigma_{\min}(K') \cdot \left(\frac{1}{2} + \frac{1 - \sigma_{\min}(K')}{m}\right) \\ &\geq \frac{\sigma_{\min}(K')}{2} \geq \frac{\sigma_{\min}(K')}{m}. \end{aligned}$$

This finishes the proof of the first claim.

Proof of the second claim. Let $K = \begin{bmatrix} \mathbf{1}_m \\ K' \end{bmatrix}$ be a submatrix of G_b that contains the all-1 row vector. Since the Gaussian distribution is rotation invariant, we can rotate all the rows of $\boldsymbol{W}$ to $V \cdot \boldsymbol{W}$, where V is a rotation matrix. The resulting matrix $V\boldsymbol{W}$ satisfies that (1) the first row equals $c \cdot \mathbf{1}_m$, where $c\sqrt{m}$ is the length of the first row that follows the Chi distribution with m degrees of freedom; and (2) the distribution of the remaining $m-1$ rows does not change, i.e., their entries remains to be i.i.d. Gaussian random variables. We can therefore view $K = \begin{bmatrix} \mathbf{1}_m \\ K' \end{bmatrix}$ as some fixed matrix perturbed by a Gaussian matrix with variance $c^2\sigma^2$, where $c^2 \geq 1 - O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$ with probability at least $1-\delta$. As a result, the minimum singular value of K also follows from the characterization in Lemma J.8.

Union bound on all submatrices Finally, we are ready to bound the tail probability of the minimum singular value. From Equation (1), it suffices to consider all the square submatrices of G_b with size at most m, and show that they all have minimum singular value lower bounded by $\underline{\sigma}m$. From Lemma J.8, for a given $b \in \mathcal{B}$, $r \leq m$ and $K \in \mathcal{S}_r(H_b)$, we have

$$\Pr\left(\sigma_{\min}(K) \leq \underline{\sigma}m\right) \leq O\left(\frac{m^{3/2}}{\sigma} \cdot \underline{\sigma}\right).$$

Therefore, taking a union bound for all such submatrices, we have

$$\Pr\left(\exists b \in \mathcal{B}, r \leq m, K' \in \mathcal{S}_r(H_b),\ \sigma_{\min}(K') \geq m \cdot \underline{\sigma}\right) \leq O\left(\frac{m^{\frac{5}{2}}}{\sigma} \cdot \underline{\sigma} \cdot \binom{n}{\leq m}\right) \leq O\left(\frac{m^{\frac{5}{2}}2^n}{\sigma} \cdot \underline{\sigma}\right).$$

Finally, setting the right hand side probability to be δ, we have established that

$$\Pr\left(\forall b \in \mathcal{B}, \forall K \in \mathcal{S}_m(G_b),\ \sigma_{\min}(K) \geq \frac{\sigma\delta}{m^{\frac{5}{2}}2^n}\right) \geq 1 - \delta.$$

The proof is thus complete. □

Lemma J.8 (Theorem 3.3 of (Sankar et al., 2006))**.** *Let $\overline{\boldsymbol{A}} \in \mathbb{R}^{m \times m}$ be an arbitrary square matrix, and let $\boldsymbol{A}$ be a Gaussian perturbation of $\overline{\boldsymbol{A}}$ of variance σ^2. Then*

$$\Pr\left(\sigma_{\min}(\boldsymbol{A}) \leq x\right) \leq 2.35 \frac{\sqrt{m}}{\sigma} \cdot x.$$