Electronic Companion for: Learning in Stackelberg Games with Non-myopic Agents

EC.1. Supplementary Material for our Reduction (Section 2)

EC.1.1. Reduction to robust learning with delays (proof of Proposition 1)

Proof of Proposition 1. Given a D-delayed policy $\mathcal{A}$, we show by contradiction that the policy $\mathcal{B}$ of a γ-discounting agent satisfies $\mathcal{B}(H_{t-1}, x_t) \in \mathsf{BR}^\tau(x_t)$ for any pair (H_{t-1}, x_t) that occurs with positive probability, where $\tau = \frac{1}{1-\gamma}\gamma^D$. Our choice of $D = \lceil T_\gamma \log(T_\gamma/\varepsilon) \rceil$ ensures that $\gamma^D \le \varepsilon/T_\gamma$, and so $\tau = T_\gamma \gamma^D \le \varepsilon$, as desired. If $\mathcal{B}(H_{t-1}, x_t) \notin \mathsf{BR}^\tau(x_t)$ for a pair (H_{t-1}, x_t) that occurs with positive probability, we can construct a modified agent policy $\mathcal{B}'$ with strictly higher expected payoff. Define $\mathcal{B}'$ so that $\mathcal{B}'(H_{t-1}, x_t) \in \mathsf{BR}(x_t)$ and $\mathcal{B}'(H', x') = \mathcal{B}(H', x')$ for all other pairs (H', x'). Conditioned on history H_{t-1}, a D-delayed policy $\mathcal{A}$ plays the same sequence of actions $x_{t+1}, \dots x_{t+D-1}$ under both $\mathcal{B}$ and $\mathcal{B}'$. Therefore, conditioned on playing history H_{t-1} and observing action x_t, the agent loses at most $\sum_{s=D+t}^{\infty} \gamma^s = \gamma^{D+t}/(1-\gamma)$ in discounted future payoff by switching to $\mathcal{B}'$ because the principal's policy is D-delayed. Moreover, the agent gains more than $\gamma^t \tau = \gamma^{D+t}/(1-\gamma)$ payoff at time t. Thus, switching from $\mathcal{B}$ to $\mathcal{B}'$ yields a strictly positive gain in expectation for the agent. □

EC.1.2. A batch-delay equivalence (proof of Proposition 2)

Before proving the result, we define a general framework for bandit problems that generalizes the Stackelberg setting. We consider bandit problems over T rounds. An *abstract bandit problem* is defined by a tuple $(\mathcal{X}, \mathcal{Y}, r)$, where $\mathcal{X}$ is the principal's action set, $\mathcal{Y}$ describes possible unknown states, and r is a regret function $r\colon \mathcal{H} \to \mathbb{R}$ mapping the set $\mathcal{H} \coloneqq \bigcup_{t \ge 0} (\mathcal{X} \times \mathcal{Y})^t$ of histories to regret values. We assume r is *subadditive*: if a history $H \in \mathcal{H}$ is partitioned into two complementary subsequences $H', H'' \in \mathcal{H}$, then $r(H) \le r(H') + r(H'')$. Subadditivity is satisfied by common notions of regret: in stochastic settings, regret is simply the sum of regrets over individual rounds; in adversarial settings, regret is subadditive. We further distinguish a subset $\mathcal{H}^* \subseteq \mathcal{H}$ of *feasible* histories,

and assume that any subsequence of a feasible history is also feasible. In our setting, feasible histories correspond to restrictions on the agent's behavior, e.g., the agent plays an ε-approximate best response at each round. The principal's policy is a map $\mathcal{A}\colon \mathcal{H} \to \mathcal{X}$. During the t-th round, the principal plays action $x_t = \mathcal{A}(H_{t-1})$ (where H_{t-1} is the history up to the start of round t) and observes y_t (which may be chosen randomly and adaptively based on x_t). We say that $\mathcal{A}$ satisfies the regret bound $R_{\mathcal{A}}(T)$ if, for each history $H \in \mathcal{H}^*$ of length T such that $x_t = \mathcal{A}(H_{t-1})$, then $r(H) \le R_{\mathcal{A}}(T)$.

Given any abstract bandit problem $(\mathcal{X}, \mathcal{Y}, r)$, we define learning with delayed feedback and batched queries as follows. As before, $\mathcal{A}$ is *D-delayed* if $\mathcal{A}(H_{t-1})$ depends only on the prefix H_{t-D}, and $\mathcal{A}$ is *B-batched* if $\mathcal{A}(H_{t-1})$ depends only on the prefix $H_{B\lfloor (t-1)/B \rfloor}$.

To cast our principal-agent learning setting as an abstract bandit problem, we let $\mathcal{X}$ be the set of principal actions, $\mathcal{Y}$ be the set of agent actions, and regret be Stackelberg regret. Note that Stackelberg regret (1) is subadditive because $\max_x(f(x)+g(x)) \le \max_x(f(x)) + \max_x(g(x))$. Finally, we take the set of feasible histories to be those where the agent policy belongs to a class $\mathfrak{B}$.

We now present a proof of the batch-delay equivalence (Proposition 2), which relies on the following lemma. It states that a 1-delayed policy $\mathcal{A}$ can be converted into a 2-delayed policy by instantiating two independent copies of $\mathcal{A}$ and following them on alternating rounds.

LEMMA EC.1. *Let $\mathcal{A}$ be a policy with regret bound $R_{\mathcal{A}}$. Consider the policy $\mathcal{A}'$ that instantiates two independent copies $\mathcal{A}_0$ and $\mathcal{A}_1$ of $\mathcal{A}$, and on round t plays $x_t = \mathcal{A}_r(((x_{t'}, y_{t'}))_{t' \equiv r \pmod 2, t' < t})$, where $t \equiv r \pmod 2$. Then $\mathcal{A}'$ is 2-delayed and satisfies a regret bound of $R_{\mathcal{A}'}(T) \le 2R_{\mathcal{A}}(T)$.*

Proof. To bound regret, note that $\mathcal{A}_0$ is run on the history $((x_{t'}, y_{t'}))_{t' \equiv 0 \pmod 2, t' \le T}$, which is by definition feasible. Thus, it incurs a total of $R_{\mathcal{A}}(\lceil T/2 \rceil)$ regret on this subsequence of the actual history. Likewise, $\mathcal{A}_1$ incurs at most $R_{\mathcal{A}}(\lceil T/2 \rceil)$ regret as well. Therefore, by the subadditivity axiom, the total regret is at most $2R_{\mathcal{A}}(\lceil T/2 \rceil) \le 2R_{\mathcal{A}}(T)$, since regret is monotonic in the delay length. The lemma now follows, since by definition, $\mathcal{A}'$ is 2-delayed. □

Proof of Proposition 2. The first claim follows from the definition of a batched algorithm, since $t - D \le D\lfloor (t-1)/D \rfloor$. The second claim follows from an application of Lemma EC.1 to the "B-batched" $(\mathcal{X}^B, \mathcal{Y}^B, r^B)$ bandit problem, where $\mathcal{X}^B$ and $\mathcal{Y}^B$ be are the B-fold products of $\mathcal{X}$ and $\mathcal{Y}$,

respectively, and r^B is given by evaluating r on the history given by the concatenations of actions $((x_1,\dots,x_B),(y_1,\dots,y_B))$. That is, in this new bandit problem, the principal simply chooses B-tuples of actions and receives feedback on these B-tuples at once, with regret measured according to the original $(\mathcal{X},\mathcal{Y},r)$ bandit problem. This new problem is by definition equivalent to our B-batched bandit problem defined above. By Lemma EC.1, an algorithm $\mathcal{A}$ for this equivalent problem can be converted into an algorithm $\mathcal{A}'$ for the 2-delayed version of this problem achieving $2R_{\mathcal{A}'}(T)$ regret. Forgetting about the batch structure, we see that this algorithm $\mathcal{A}'$ is B delayed, since any batch starting at time $t=kB+1$ depends on the history $H_{<k(B-1)+1}$. Hence the second claim is proven. □

EC.2. Supplementary Material for Theoretical Results on SSGs (Sections 3-4)

EC.2.1. Simplifying design & analysis of Clinch when $\mathcal{X}=\Delta_{n-1}$ (Remark 1)

When $\mathcal{X}=\Delta_{n-1}^{\le}\coloneqq\{x:\|\mathbf{x}\|_1\le 1\wedge x_y\ge 0\,\forall y\}$, we may as well restrict to $\mathcal{X}=\Delta_{n-1}$. Indeed, since each agent utility function v^y is continuous and strictly decreasing, we can increase coverage probabilities of any $\mathbf{x}\in\mathcal{X}\setminus\Delta_{n-1}$ to obtain $\mathbf{x}'\in\Delta_{n-1}$ with $\mathsf{BR}(\mathbf{x}')=\mathsf{BR}(\mathbf{x})$ and $v(\mathbf{x}',\mathsf{br}(\mathbf{x}'))<v(\mathbf{x}',\mathsf{br}(\mathbf{x}'))$. This argument also implies that the optimal stable strategy $\mathbf{x}^\star$ guaranteed by Proposition 3 belongs to Δ_{n-1}. From now on, we thus fix $\mathcal{X}=\Delta_{n-1}$.

For this setting, we present a simplified (simplexified) algorithm Clinch.Simplex that achieves the same query complexity as Theorem 1 but admits a simpler analysis. Similarly to Clinch, Clinch.Simplex maintains an (approximate) entry-wise lower bound $\underline{\mathbf{x}}$ for $\mathbf{x}^\star$ initialized to the 0 vector. This time, however, we envision the remaining mass $1-\|\underline{\mathbf{x}}\|_1$ as a potential that is decreased with each step. To ensure a significant reduction, we query $\mathbf{x}$ which distributes this remaining mass evenly across the coordinates of $\underline{\mathbf{x}}$ and update $\underline{x}_y\leftarrow x_y$ for the attacked target $y\in\mathcal{Y}$. Finally, we normalize $\underline{\mathbf{x}}$ so that it lies on the simplex and apply the same perturbation used by Clinch.

Proposition EC.1. *Fix $0<\lambda\le 1$. Then* Clinch.Simplex *returns a λ-approximate equilibrium strategy using $O(n\log\frac{C}{W\lambda})$ queries to an ε-approximate best response oracle with $\varepsilon\le\frac{W\lambda}{12C^3n}$.*

Algorithm 8: CLINCH.SIMPLEX: a robust algorithm for learning SSGs when $\mathcal{X} = \Delta_n$

input : target accuracy $\lambda \in (0, 1]$, best response oracle ORACLE with $\text{ORACLE}(x) \in \mathsf{BR}^{\varepsilon}(x)$
output: λ-approximate equilibrium strategy

$\underline{\mathbf{x}} \leftarrow (0, 0, \ldots, 0) \in \mathbb{R}^n, \delta \leftarrow \frac{W\lambda}{6C^2}$
for $i = 1, 2, \ldots, \lceil n \ln \frac{4}{\delta} \rceil$ **do**
  $\mathbf{x} \leftarrow \underline{\mathbf{x}} + (1, 1, \ldots, 1) \cdot \frac{1}{n}(1 - \|\underline{\mathbf{x}}\|_1)$
  $y \leftarrow \text{ORACLE}(\mathbf{x})$
  $\underline{x}_y \leftarrow x_y$
$\hat{\mathbf{x}} \leftarrow \underline{\mathbf{x}} / \|\underline{\mathbf{x}}\|_1$
$\hat{y} \leftarrow \arg\max_{y \in \mathcal{Y} : \hat{x}_y > W/2} u(\hat{x}, y)$
return $\hat{\mathbf{x}} - \frac{W\lambda}{2}\mathbf{x}_{\hat{y}}$

Proof. First, we note that $\underline{x}_y \le x^\star_y + C\varepsilon$ for each $y \in \mathcal{Y}$, by the same argument applied in the proof of Theorem 1. Next, we analyze convergence. Notice that the quantity $1 - \|\underline{\mathbf{x}}\|_1$ decreases by a factor of $1 - \frac{1}{n}$ after each iteration. Since $1 - \|\underline{\mathbf{x}}\|_1$ is initially 1, after $\lceil n \ln \frac{4}{\delta} \rceil$ iterations, it holds that

$$1 - \|\underline{\mathbf{x}}\|_1 \le \left(1 - \frac{1}{n}\right)^{n \ln \frac{4}{\delta}} \le \frac{\delta}{4}.$$

We may thus conclude, for $\hat{\mathbf{x}} \coloneqq \underline{\mathbf{x}} / \|\underline{\mathbf{x}}\|_1$, that

$$\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_\infty \le \|\hat{\mathbf{x}} - \mathbf{x}^\star\|_1 \le \left(\|\underline{\mathbf{x}}\|_1^{-1} - 1\right)\|\underline{\mathbf{x}}\|_1 + \|\underline{\mathbf{x}} - \mathbf{x}^\star\|_1 \le 1 - \|\underline{\mathbf{x}}\|_1 + \|\underline{\mathbf{x}} - \mathbf{x}^\star\|_1,$$

by the triangle inequality. By the entry-wise lower bound property of $\underline{x}$, we further note that

$$\|\underline{\mathbf{x}} - \mathbf{x}^\star\|_1 \le 1 - \|\underline{\mathbf{x}}\|_1 + Cn\varepsilon \le 1 - \|\underline{\mathbf{x}}\|_1 + \frac{\delta}{2},$$

and so $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_\infty \le \delta$. Finally, Lemma 4 gives that the returned point is a λ-approximate Stackelberg equilibrium strategy, as desired. □

EC.2.2. Minimizing agent best response utility (proof of Lemma 2)

To show that CLINCH makes continual progress, we require an approximate version of Grünbaum's inequality (Grünbaum 1960). First, we state the classic result.

LEMMA EC.2 **(Theorem 2 in Grünbaum (1960))**. *If K is a non-empty compact convex set in $\mathbb{R}^d$, then for any halfspace H containing its centroid $\mathbf{x} = \mathrm{E}_{\mathbf{z} \sim \mathrm{Unif}(K)}[\mathbf{z}]$, we have* $\mathrm{vol}_d(H \cap K) \ge \frac{1}{e}\mathrm{vol}_d(K)$.

For our approximate case, we use the following, which implies that each update to $\underline{\mathbf{x}}$ will sufficiently shrink the volume of the active search region S (so long as no target is removed from $\mathcal{R}$).

LEMMA EC.3. *Let $K \subseteq [0,1]^d$ be convex and downward closed with centroid $\mathbf{x} = \mathrm{E}_{\mathbf{z}\sim\mathrm{Unif}(K)}[\mathbf{z}]$, and write $\alpha = \sup_{\mathbf{z}\in K} z_1$. Then for $\beta \geq 0$, we have $\mathrm{vol}_d(\{\mathbf{z} \in K : z_1 \geq x_1 - \beta\}) < (1 + e\beta/\alpha)^d(1 - 1/e)\mathrm{vol}_d(K)$.*

Proof of Lemma EC.3. For convenience, assume that K is closed; this does not affect the volumes. Similarly replace β with $\min\{\beta, x_1\}$ so that $x_1 - \beta \geq 0$. Proving the result with this update implies the original result.

Now define the halfspace $H \coloneqq \{\mathbf{z} \in \mathbb{R}^d : z_1 \geq x_1 - \beta\}$. First, we observe that the related halfspace $H_0 = \{\mathbf{z} \in \mathbb{R}^d : z_1 \geq x_1\}$ satisfies $\frac{1}{e}\mathrm{vol}_d(K) \leq \mathrm{vol}_d(H_0 \cap K) < \left(1 - \frac{1}{e}\right)\mathrm{vol}_d(K)$ by Grünbaum's inequality (Lemma EC.2). By downward closure, we have $\alpha\mathbf{e}_1 \in K$ and deduce that $\mathrm{vol}_d(H_0 \cap K) \leq \left(1 - \frac{x_1}{\alpha}\right)\mathrm{vol}_d(K)$. This requires that $x_1 \leq \left(1 - \frac{1}{e}\right)\alpha$ to avoid violating the first inequality.

Next, consider the $(d-1)$-dimensional intersection of K with the hyperplane defining H_0, denoted by $L_0 \coloneqq \{\mathbf{z} \in K : z_1 = x_1\}$ for H_0. Convexity requires that $H_0 \cap K$ contain the convex hull of L_0 and $\alpha\mathbf{e}_1$, a cone we denote by A with $\mathrm{vol}_d(A) = \frac{\alpha - x_1}{d!}\mathrm{vol}_{d-1}(L_0)$. Moreover, every point in $H \cap K \setminus H_0$ is outside of A and connected to $\alpha\mathbf{e}_1$ by a line segment contained in H and passing through L_0. Thus, $H \cap K \setminus H_0$ is disjoint from the cone A but contained by the cone B obtained by intersecting H with the union of all rays emitted from $\alpha\mathbf{e}_1$ and passing through L_0. Similarly to A, we compute the volume of B to be $\frac{\alpha - x_1 + \beta}{d!}\left(\frac{\alpha - x_1 + \beta}{\alpha - x_1}\right)^{d-1}\mathrm{vol}_{d-1}(L_0)$. Consequently, we have

$$\begin{aligned}
\mathrm{vol}_d(H \cap K \setminus H_0) &\leq \mathrm{vol}_d(B) - \mathrm{vol}_d(A) \\
&= \left[(\alpha - x_1 + \beta)\left(\frac{\alpha - x_1 + \beta}{\alpha - x_1}\right)^{d-1} - (\alpha - x_1)\right]\frac{1}{d!}\mathrm{vol}_{d-1}(L_0) \\
&\leq \left[\left(\frac{\alpha - x_1 + \beta}{\alpha - x_1}\right)^{d} - 1\right]\mathrm{vol}_d(H_0 \cap K) \\
&= \left[\left(1 + \frac{\beta}{\alpha - x_1}\right)^{d} - 1\right]\mathrm{vol}_d(H_0 \cap K) \\
&\leq \left[\left(1 + \frac{e\beta}{\alpha}\right)^{d} - 1\right]\mathrm{vol}_d(H_0 \cap K).
\end{aligned}$$

Finally, we can bound

$$\begin{aligned}\mathrm{vol}_d(H \cap K) &= \mathrm{vol}_d(H_0 \cap K) + \mathrm{vol}_d(H \cap K \setminus H_0) \\ &\le \left(1 + \frac{e\beta}{\alpha}\right)^d \mathrm{vol}_d(H_0 \cap K) \\ &< \left(1 + \frac{e\beta}{\alpha}\right)^d \left(1 - \frac{1}{e}\right) \mathrm{vol}_d(K),\end{aligned}$$

as desired. □

We now prove the guarantee for the primary stage of CLINCH.

Proof of Lemma 2. First, we observe that $\underline{\mathbf{x}}$ always approximately lower bounds $\mathbf{x}^\star$ in each entry. Note that whenever $\underline{x}_y$ gets updated, we set $\underline{x}_y = x_y - C\varepsilon$ for some $\mathbf{x}$ such that $y \in \mathsf{BR}^\varepsilon(x)$. By monotonicity of v^y and our slope bound, this implies $x^\star_y \ge x_y - C\varepsilon = \underline{x}_y$, as desired.

Next, we will show that the termination condition at Step 2 is satisfied after at most $O(n \log \frac{n}{\delta})$ rounds, recalling that $0 < \delta \le 1$ is our desired accuracy. To start, we establish a bit of notation to keep track of variables between iterations. For each round $i = 1, 2, \dots$ before termination, we write $\mathcal{R}_i$ for the remaining targets and S_i for the active search region after Step 5, $\mathbf{x}^{(i)}$ for the queried point at Step 6, y_i for the oracle response at Step 7, and $\underline{\mathbf{x}}^{(i)}$ for the value of $\underline{\mathbf{x}}$ after Step 8. Set $n_i = \dim(S_i)$, defined as the minimum dimension of the subspace spanned by $S_i - \mathbf{w}$ over some $\mathbf{w} \in S_i$. Finally, write $\lambda = \frac{\delta}{4C^2}$ for the threshold used to flatten S. Now, we fix ourselves at some round i and consider two cases.

Case 1: $\mathcal{R}_{i+1} = \mathcal{R}_i$. In this case, no targets are removed from $\mathcal{R}_i$ and $n_{i+1} = n_i$. Since we selected $\mathbf{x}^{(i)}$ as the centroid of S_i, we can apply Lemma EC.3. Indeed, S_i is convex, and its translation $K = S_i - \underline{\mathbf{x}}^{(i-1)}$ is downward closed. Moreover, $\sup_{\mathbf{z} \in K} z_{y_i} = \sup_{\mathbf{z} \in S_i} z_{y_i} - \underline{x}^{(i-1)}_{y_i} \ge \lambda$ (otherwise, the target y_i would have been removed from $\mathcal{R}_i$ to obtain $\mathcal{R}_{i+1}$). Consequently, we have

$$\begin{aligned}\mathrm{vol}_{n_{i+1}}(S_{i+1}) = \mathrm{vol}_{n_i}(S_{i+1}) &= \mathrm{vol}_{n_i}\left(\left\{\mathbf{z} \in S_i : z_{y_i} \ge x^{(i)}_{y_i} - C\varepsilon\right\}\right) \\ &\le \left(1 + \frac{e \cdot C\varepsilon}{\lambda}\right)^n \left(1 - \frac{1}{e}\right) \mathrm{vol}_d(S_i) \\ &\le e^{1/3}\left(1 - \frac{1}{e}\right) \mathrm{vol}_d(S_i) < \frac{9}{10}\mathrm{vol}_d(S_i),\end{aligned}$$

where the penultimate inequality uses that $\varepsilon \le \frac{\lambda}{3Cen} = \frac{\delta}{12C^3 en}$.

Case 2: $\mathcal{R}_{i+1} \subset \mathcal{R}_i$. In this case, $n_{i+1} - n_i > 0$ targets are removed from $\mathcal{R}_i$ in the next step. Writing $K_i = \{\mathbf{x}' \in S_i : x'_z = \underline{x}_z^{(i-1)}\ \forall z \notin R_{i+1}\}$ for the region which enforces the locked coordinates for the next step — but not the updated lower envelope — we (loosely) bound

$$\mathrm{vol}_{n_{i+1}}(S_{i+1}) \le \mathrm{vol}_{n_{i+1}}(K_i) \le \left(\frac{n}{\lambda}\right)^{n_i - n_{i+1}} \mathrm{vol}_{n_i}(S_i).$$

The first inequality uses that $S_{i+1} \subseteq K_i$. For the second, convexity requires that S_i contains the convex hull of K_i and the points $\{\underline{x}^{(i-1)} + \lambda \mathbf{e}_z : z \in \mathcal{R}_i \setminus \mathcal{R}_{i+1}\}$, which has volume loosely bounded from below by $(\lambda/n)^{|n_{i+1}-n_i|}\mathrm{vol}_{n_{i+1}}(K_i)$.

Combining these cases inductively, we deduce that

$$\mathrm{vol}_{n_i}(S_i) < \left(\frac{n}{\lambda}\right)^n \left(\frac{9}{10}\right)^{i-n} \alpha^n. \tag{EC.1}$$

On the other hand, once $\mathrm{vol}_{n_i}(S_i) < \lambda^n/n!$, every coordinate must have slack less than λ, and the termination condition at Step 2 will be satisfied. Consequently, we compute that the outer loop must terminate after at most $15n \log \frac{2\alpha n}{\lambda} \le 15n \log \frac{8C^2\alpha n}{\delta}$ iterations. At this point, we have $y \in \mathsf{BR}^{\varepsilon}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{X}$ with $\mathbf{x} \ge \underline{\mathbf{x}}$ and either $\underline{\mathbf{x}} + \lambda \mathbf{e}_y \notin \mathcal{X}$ or $\underline{x}_y + \lambda > \overline{x}_y$. In the former case, downward closure of $\mathcal{X}$ implies $x^\star_y \le \underline{x}_y + \lambda \le x_y + \lambda$, and the same relations hold for the latter, since $x^\star_y \le \overline{x}_y$ at this point from the input guarantee. Hence, we obtain

$$\begin{aligned} v(\mathbf{x}^\star, \mathsf{br}(\mathbf{x}_\star)) \ge v^y(\mathbf{x}^\star_y) &\ge v^y(x_y + \lambda) \\ &\ge v^y(x_y) - C\lambda \\ &\ge v(\mathbf{x}, \mathsf{br}(\mathbf{x})) - 2C\lambda \\ &= v(\mathbf{x}, \mathsf{br}(\mathbf{x})) - \frac{\delta}{2C}. \quad \square \end{aligned}$$

EC.2.3. Mass conservation (proof of Lemma 3)

Proof of Lemma 3. Fixing $y \in \mathcal{Y}$, define the thresholds

$$\begin{aligned} r_y &= \sup\left\{p \in [\underline{x}_y, x_y] : \mathsf{BR}^{\lambda/C}(\mathbf{x} + [p - x_y]\mathbf{e}_y) = \{y\}\right\}, \\ s_y &= \sup\left\{p \in [\underline{x}_y, x_y] : \mathsf{BR}(\mathbf{x} + [p - x_y]\mathbf{e}_y) = \{y\}\right\}, \end{aligned}$$

$$t_y = \sup\left\{p \in [\underline{x}_y, x_y] : y \in \mathsf{BR}^{\lambda/C}(\mathbf{x} + [p - x_y]\mathbf{e}_y)\right\},$$

where we define each to be $\underline{x}_y$ if the corresponding set is empty. (Note that the set for s_y can contain at most one point by strict monotonicity of v^y.) By monotonicity of v^y and our slope bound, we have $s_y - \lambda \le r_y \le s_y \le t_y \le s_y + \lambda$. By our choice of binary search, either $\hat{x}_y > x_y - \lambda$, or $\hat{x}_y > m - \lambda$ at some iteration for which $\textsc{Oracle}(\mathbf{x} - [x_y - m]\mathbf{e}_y) \neq y$. In the latter case, monotonicity of v^y and the slope bound require that $\hat{x}_y \ge r_y$, while, for the former, we have $\hat{x}_y > r_y - \lambda$. Similarly, either $\hat{x}_y = \underline{x}_y$, or $\hat{x}_y < m$ for a search iteration during which $\textsc{Oracle}(\mathbf{x} - [x_y - m]\mathbf{e}_y) = y$. In the latter case, monotonicity of v^y and the slope bound require that $\hat{x}_y < t_y$, while, for the former, we have $\hat{x}_y \le t_y$. Combining, we have that $\hat{x}_y \in (s_y - 2\lambda, s_y + \lambda]$.

By definition of s_y, we must have $v^y(s_y) \le v(\mathbf{x}, \mathsf{br}(\mathbf{x}))$ (with equality unless $s_y = \underline{x}_y$). Hence, $v^y(\hat{x}_y) < v^y(s_y) + 2C\lambda \le v(\mathbf{x}, \mathsf{br}(\mathbf{x})) + 2C\lambda$. Since this holds for all $y \in \mathcal{Y}$, we have $v(\hat{\mathbf{x}}, \mathsf{br}(\hat{\mathbf{x}})) \le v(\mathbf{x}, \mathsf{br}(\mathbf{x})) + 2C\lambda$, proving the second part of the claim. Now, if $\hat{x}_y > \underline{x}_y$, we must have $t_y > \hat{x}_y > \underline{x}_y$, and so $y \in \mathsf{BR}^{\lambda/C}(\mathbf{x} + [\hat{x}_y - x_y]\mathbf{e}_y)$. The previous result then implies that $y \in \mathsf{BR}^{\lambda/C + 2C\lambda}(\mathbf{x} + [\hat{x}_y - x_y]\mathbf{e}_y)$. We bound $\lambda/C + 2C\delta \le 3C\lambda$ for conciseness, and note that each binary search use $O\big(\log \frac{\alpha}{\lambda}\big)$ queries. □

EC.2.4. Perturbing estimates of $\mathbf{x}^\star$ (proof of Lemma 4)

Proof of Lemma 4. The error bound implies that $\hat{x}_y > W/2$ only if $x^\star_y > 0$. On the other hand, if $x^\star_y > 0$, then our regularity width assumption requires that $x^\star_y \ge W$, and so $\hat{x}_y \ge W - \frac{W\lambda}{6C^2} > W/2$. As noted in the proof of Proposition 3, there are no $y \in \mathsf{BR}(\mathbf{x}^\star)$ with $x^\star_y = 0$, and so

$$\mathsf{BR}(\mathbf{x}^\star) = \{y \in \mathcal{Y} : x^\star_y > 0\} = \{y \in \mathcal{Y} : \hat{x}_y > W/2\}.$$

Now fix $\hat{y}$ as defined in Step 8, and consider the returned strategy $\tilde{\mathbf{x}} \coloneqq \hat{\mathbf{x}} - \frac{W\lambda}{2}\mathbf{e}_{\hat{y}}$. We know that $\tilde{\mathbf{x}} \in \mathcal{X}$ since $\hat{x}_y > W/2$ and $\mathcal{X}$ is downward closed. We claim that $\mathsf{BR}(\tilde{\mathbf{x}}) = \{\hat{y}\}$. Indeed, we have $\tilde{x}_{\hat{y}} < x^\star_{\hat{y}} - (\frac{W\lambda}{2} - \frac{W\lambda}{6C^2}) \le x^\star_{\hat{y}} - W\lambda/3$, and so our lower slope bound requires that

$$v^{\hat{y}}(\tilde{x}_{\hat{y}}) > v^{\hat{y}}(x^\star_{\hat{y}}) + \frac{W\lambda}{4C} = v(\mathbf{x}^\star, \mathsf{br}(\mathbf{x}^\star)) + \frac{W\lambda}{3C}.$$

For $y \neq \hat{y}$, we have $\tilde{x}_y > x^\star_y - \frac{W\lambda}{6C^2}$, and so our upper slope bound requires that

$$v^y(\tilde{x}_y) < v^y(x^\star_y) + C\frac{W\lambda}{6C^2} \leq v(\mathbf{x}^\star, \mathsf{br}(\mathbf{x}^\star)) + \frac{W\lambda}{6C} \leq v(\mathbf{x}^\star, \mathsf{br}(\mathbf{x}^\star)) + \frac{W\lambda}{3C} - \varepsilon.$$

Consequently, we have $\mathsf{BR}^\varepsilon(\tilde{\mathbf{x}}) = \{\hat{y}\}$. Finally, we compute

$$\begin{aligned} u(\tilde{\mathbf{x}}, \mathsf{br}(\tilde{\mathbf{x}})) &= u^{\hat{y}}(\tilde{x}_{\hat{y}}) \\ &\geq u^{\hat{y}}(x^\star_{\hat{y}}) - C|x^\star_{\hat{y}} - \tilde{x}_{\hat{y}}| \\ &\geq u(\mathbf{x}^\star, \mathsf{br}(\mathbf{x}^\star)) - \frac{W\lambda}{6C} \\ &> u(x^\star, \mathsf{br}(\mathbf{x}^\star)) - \lambda, \end{aligned}$$

verifying that $\tilde{\mathbf{x}}$ is indeed a λ-approximate Stackelberg equilibrium strategy for the principal. □

EC.2.5. Exact search with bounded bit precision (discussion in Section 3.4)

We now analyze Clinch imposing the additional regularity assumptions of Peng et al. (2019).

Assumption EC.1. *Agent utilities are linear with rational coefficients whose denominators are at most 2^L, and that each non-empty best response region has volume at least 2^{-nL}. Moreover, $\mathcal{X}$ is a polytope represented as the intersection of a finite set of half-spaces, each of the form $\{\mathbf{x} \in [0,1]^n : \mathbf{x}^\top a \leq b\}$ where $b \in \mathbb{R}$ and each entry of $a \in \mathbb{R}^n$ are rational with numerators and denominators at most 2^L.*

By the discussion of our regularity assumptions in Section 3.1, it suffices to take $C = 2^L$. With these settings, Theorem 1 states that Clinch terminates in $O(nL + n\log\frac{1}{\delta})$ oracle queries, and the returned strategy $\hat{\mathbf{x}} \in \mathcal{X}$ satisfies $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_\infty < \delta$. A more careful analysis can eliminate dependence on W—yielding query complexity $O(nL + n\log\frac{1}{\lambda})$—by avoiding the final perturbation step of Clinch. However this improvement will not impact our final result.

Next, we bound the bit complexity of $\mathbf{x}^\star$.

Lemma EC.4. *If agent utilities are linear with rational coefficients whose denominators are at most 2^L, then the entries of $\mathbf{x}^\star$ are rational with denominators at most 2^{8Ln}.*

Proof. Fix any $y \in \mathsf{BR}(x^\star)$, and write $v^y(t) = d_y - c_y t$, where $c_y, d_y \in (0,1]$ are rational with denominators at most 2^L. As noted in the proof of Proposition 3, we must have $x^\star_y > 0$. Writing $w^\star = v(\mathbf{x}^\star, \mathsf{br}(\mathbf{x}^\star)) = v^y(x^\star_y)$, we can solve for $x^\star_y = (d_y - w^\star)/c_y$. For $y \notin \mathsf{BR}(x^\star)$, we have $x^\star_y = 0$.

Now, since $\mathbf{x}^\star$ minimizes the agent's best response utility, we know that $w^\star$ is as small as possible so that the $\mathbf{x}^\star$ as determined above lies in $\mathcal{X}$. In other words, $\mathbf{x}^\star$ must lie on a face of $\mathcal{X}$, represented as $\{\mathbf{x} \in \mathbb{R}^n : a^\top \mathbf{x} = b\}$. Thus, we have $\sum_{y \in \mathsf{BR}(\mathbf{x}^\star)} a_y (d_y - w^\star)/c_y = b$ and can compute

$$w^\star = \frac{\sum_{y \in \mathsf{BR}(\mathbf{x}^\star)} a_y d_y / c_y - b}{\sum_{y \in \mathsf{BR}(\mathbf{x}^\star)} a_y / c_y}.$$

Our bit precision assumptions imply that $w^\star \in (0,1]$ is rational with denominator at most 2^{5Ln+L}, and so each $x^\star_y \in [0,1]$ must also be rational with denominator at most 2^{5Ln+3L}. □

Hence, rounding appropriately, we have the following.

PROPOSITION EC.2. *Under Assumption EC.1, running* CLINCH *with* $\delta = \frac{1}{3}2^{-8Ln}$ *and rounding each entry of the result to the nearest multiple of* 2^{-8Ln} *gives* $\mathbf{x}^\star$ *using* $O(n^2 L)$ *best response queries.*

EC.3. Full Details for Myopic Numerical Simulations (Section 3.4)

In Figure 3 of Section 3.4, we compare the performance of CLINCH to that of the previous state-of-the-art, SECURITYSEARCH (Peng et al. 2019), given queries to a best response oracle. Here, we provide further details on these experiments. Recall that code for the algorithm implementations and plots is available at `https://github.com/sbnietert/learning-stackelberg-games`.

EC.3.1. Implementation details

We implemented each algorithm in Python and NumPy according to their respective specifications. The best response oracle was straightforward to implement for the SSGs describe below, since their utilities are linear. We note that CLINCH is fully-specified without knowledge of C since the best response oracle is exact (i.e., $\varepsilon = 0$).

EC.3.2. Experimental setup

Figure 3 depicts the query complexity of Clinch and SecuritySearch on a sequence of SSGs with number of targets n ranging from 5 to 100. We examine two settings:

Setting 1. Given n, we consider the SSG where the defender's strategy space is the unit simplex Δ_{n-1} with payoffs such that the attacker (resp. defender) receives value 1 if they successfully attack (resp. defend) and value 0 otherwise. While it is clear that the optimal defender strategy is to mix uniformly over all n targets, this problem specification is unknown to the algorithms (and thus they must learn this from scratch).

In practice, we find that SecuritySearch suffers from severe numerical stability issues due to the symmetry between the targets of the problem described above. To alleviate stability problems of SecuritySearch and obtain a fairer comparison, we run this algorithm on a perturbed version of this problem where the payoff for successfully attacking or defending each target is slightly perturbed, by a uniformly random quantity between 0 and 0.0005.

Setting 2. Given n, we sample an SSG where the defender's strategy space is the unit simplex Δ_{n-1} with payoffs as follows: for each target, the attacker and defender have independent values for a successful attack or defense, each sampled independently and uniformly from $[0, 1]$; furthermore, each agent receives payoff 0 if they unsuccessfully attack or defend. Note that these games are non-zero sum, since the attacker and defender have independent valuations.

For each value of n, we sample 3 games as described above and report the averaged number of oracle queries across these three instantiations. In each setting, we run the algorithms until they solve for the equilibrium nearly exactly, up to an accuracy of 10^{-8} in each coordinate.

EC.3.3. Results

To illustrate the asymptotic scaling of sample complexity clearly, Figure 3 depicts our results on a log-log scale (n versus query count). In addition, to estimate the scaling rate, we plot a best linear fits of the log-transformed variables for each curve.

Figure 3 shows that, in Setting 1, CLINCH requires fewer than 100 samples when $n = 5$ and fewer than 2000 samples when $n = 100$, whereas SECURITYSEARCH requires over 10^4 samples when $n = 5$ and over 10^8 samples when $n = 100$. Figure 3 also shows that, in Setting 2, CLINCH requires fewer than 100 samples when $n = 5$ and fewer than 2000 samples when $n = 100$, whereas SECURITYSEARCH requires over 4000 samples when $n = 5$ and over $2 \cdot 10^6$ samples when $n = 100$.

EC.3.4. Discussion

We find CLINCH outperforms SECURITYSEARCH both in the constant factor hidden by the big-O and the asymptotic query complexity in n. The empirical complexities match theory, with the cost of SECURITYSEARCH scaling roughly as n^3 and the cost of CLINCH scaling roughly as n. (We note that the SECURITYSEARCH scales with exponent around n^2 on the random instantiations, suggesting better average- than worst-case performance; however, it is still outperformed by the linear scaling of CLINCH.) CLINCH is even efficient for small n, improving over SECURITYSEARCH by two orders of magnitude in the query complexity.

In summary, we find that CLINCH runs efficiently, being both asymptotically optimal and having a small constant factor in practice, while SECURITYSEARCH struggles even in these simple settings.

EC.4. Full Details for Non-myopic Numerical Simulations (Section 4.3)

In Figure 4 of Section 4.3, we compare the performance of multi-threaded and batched CLINCH against simulated non-myopic agents. Here, we provide further details on these experiments. Recall that code for the algorithm implementations and plots is available at `https://github.com/sbnietert/learning-stackelberg-games`.

EC.4.1. Implementation details

Both batched and multi-threaded versions of CLINCH were implemented using Python and NumPy. They were structured to advance their state one round of agent interaction at a time, so that the agent can copy their state and use it to simulate several potential future trajectories. The batched variant uses the naïve repetition approach described at the beginning of Section 4.1.

Given a batch size B, it sets accuracy $\lambda = nB/T$ and runs CLINCH with $\delta = \frac{W\lambda}{6C^2}$ and $\varepsilon = \frac{W\lambda}{200C^5 n}$ until some $\hat{x}$ is returned, naïvely repeating each query B times. Then $\tilde{x} = \text{PERTURB}(\hat{x}, \lambda)$ is played for the remaining rounds. Our multi-threaded algorithm runs $\log T$ threads in parallel, as in MULTITHREADEDCLINCH. However, the exploration phase for thread with delay B simply performs the batched search described above, instead of the full algorithm's series of searches. During each thread's exploit phase, we play the perturbed result of the highest-indexed thread which has entered the exploit phase. This variant is faster to simulate than the full algorithm and still achieves $\tilde{O}\big(nT_\gamma \log^{O(1)}(TC/W)\big)$ regret.

Our agent is defined by a discount function ν mapping delay τ to discount level $\nu(\tau)$, taken to be γ^τ for the geometric discounting plots in Figure 4. Then, at each round t, it selects target $y \in \mathcal{Y}$ maximizing $v(x_t, y) + \sum_{\tau=1}^{T} \nu(\tau) v(x_{t+\tau}, \mathsf{br}(x_{t+\tau}))$, where the future $x_{t+\tau}$ strategies are obtained by simulating the principal's algorithm forward with best responses.

EC.4.2. Experimental setup

Figure 4 uses a simplex SSG with random linear utilities and $n = 3$ targets, nearly as described in Setting 2 of Section EC.3.2. The only modification is that utilities are sampled randomly between 0.25 and 0.75 instead of 0 and 1, to ensure that the minimum width W is not too small. For the principal's algorithms, we take $W = 0.25/(0.25 + (n-1)\cdot 0.75)$ and $C = 1$. The slope bound is always valid since the coefficients are less than one, and the minimum width bound is valid when $n = 2$ and empirically worked well for larger n. Although W can feasible be much smaller for $n > 2$, we achieved strong performance with no additional tuning. Figure 4 depicts results for a single random SSG instance, though qualitatively similar results are observed for additional random instances.

EC.4.3. Additional results with geometric discounting

For Figure EC.1, we repeated the experiments for Figure 4 with four additional random SSG instances. Observe that the multi-threaded algorithm always achieves sublinear regret, while any fixed batch size performs poorly if the discount factor is too large. For each instance the set of

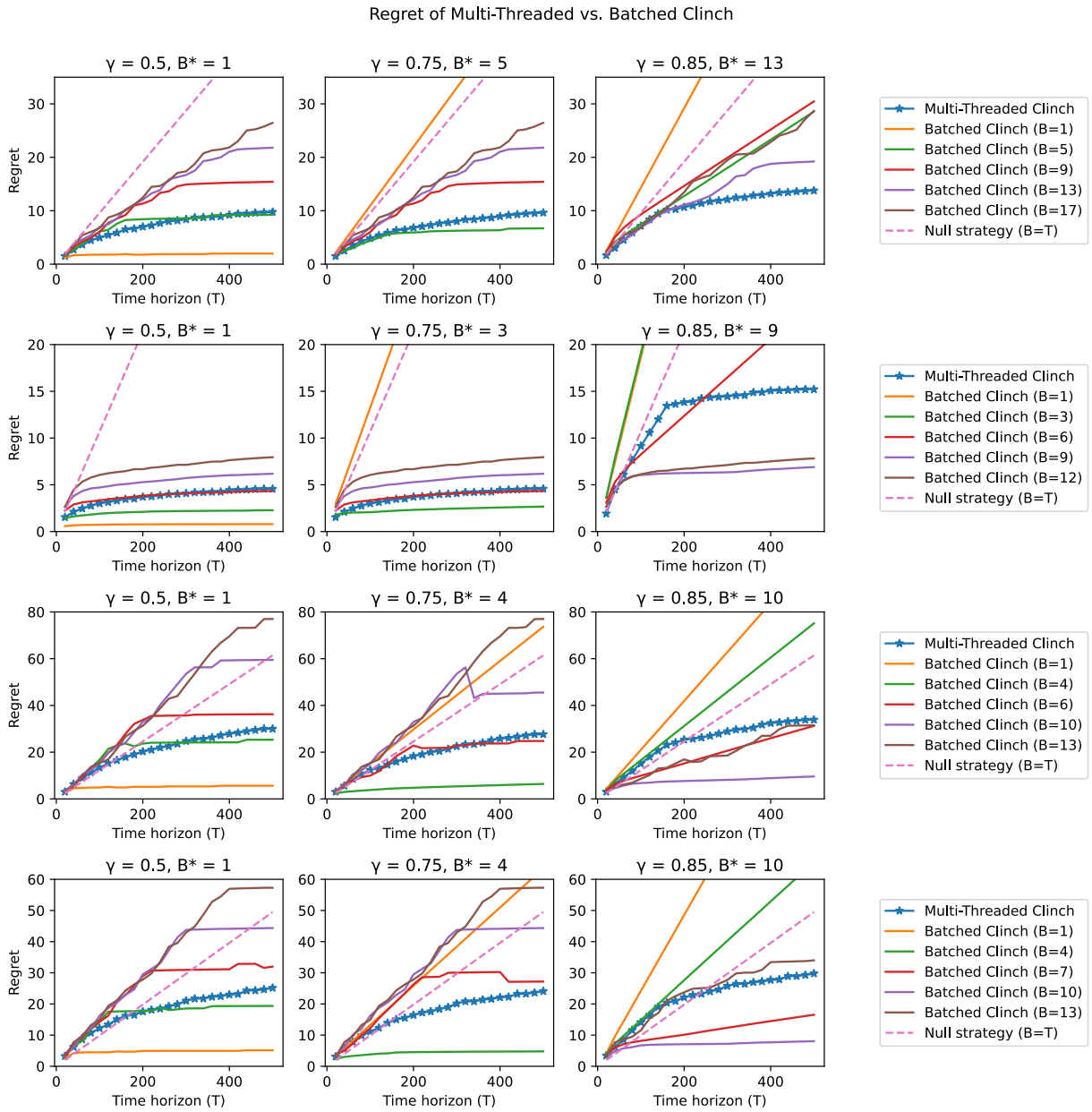

Figure EC.1 Regret achieved by batched and multi-threaded variants of CLINCH against a simulated γ-discounting agent on four random SSG instances with $n = 3$. For each instance and discount factor, we note the optimal batch size $B^\star$ at $T = 500$.

batch sizes displayed is selected to include the best batch size for each discount factor at $T = 500$ (computed via brute-force search), along with an intermediate and larger batch size.

Finally, in Figure EC.2, we present results for an extended set of experiments with $n = 10$. The maximum time horizon was extended from $T = 500$ to $T = 1000$, so that enough rounds pass by for

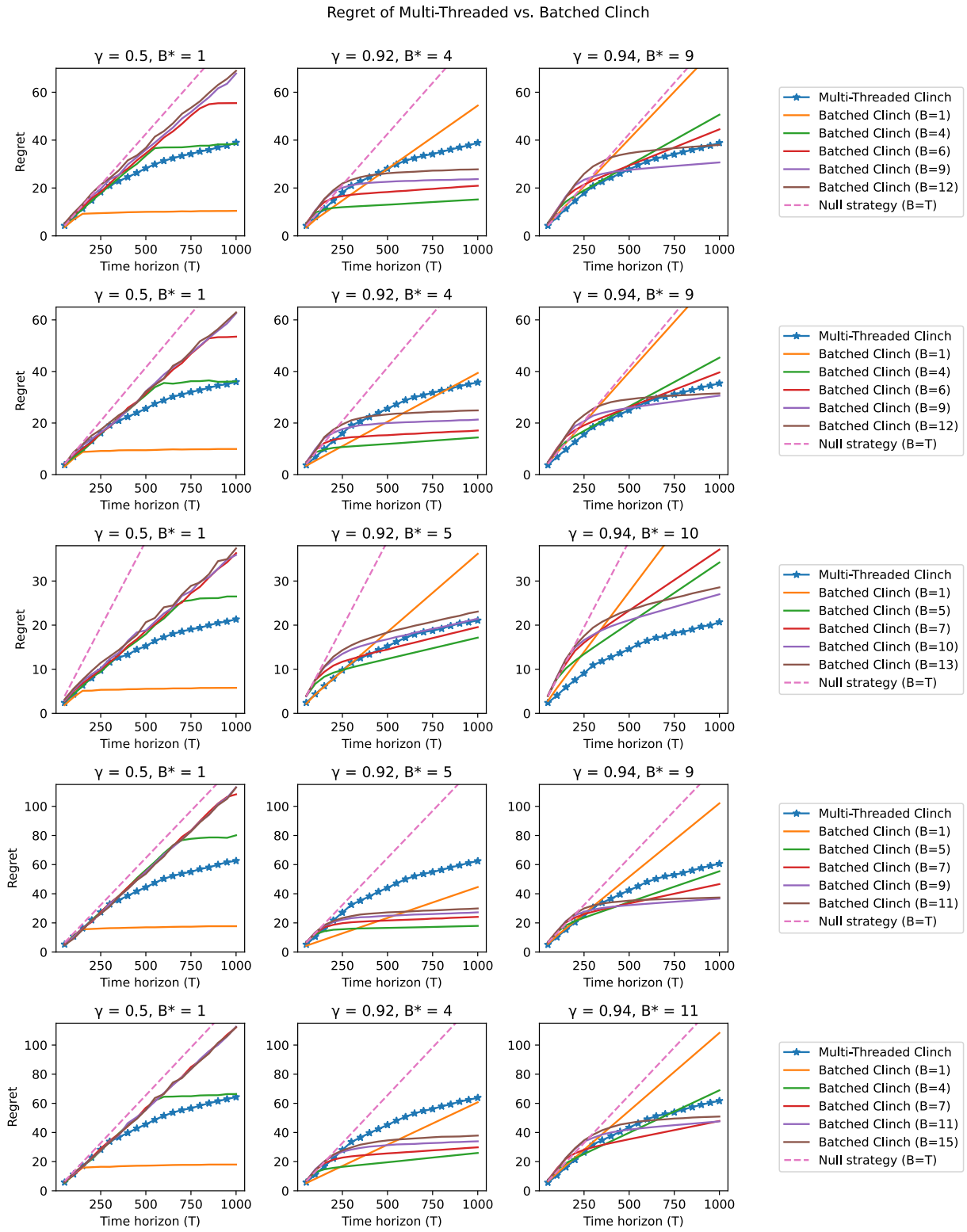

Figure EC.2 Regret achieved by batched and multi-threaded variants of CLINCH against a simulated γ-discounting agent on four random SSG instances with $n = 10$. For each instance and discount factor, we note the optimal batch size $B^\star$ at $T = 1000$.

searches to complete. Additionally, the set of discount factors was updated to $\{0.5, 0.92, 0.94\}$. The latter two were increased so that the simulated non-myopic agent has significant incentive to deviate from best-response behavior. We observe qualitatively similar results to Figure EC.1, although the overhead of multi-threading compared to selecting the optimal batch size, or slightly larger, is more pronounced in some regimes (particularly when $\gamma = 0.92$). Importantly, the regret of multi-threaded CLINCH is still sublinear, in contrast to the linear regret suffered when the batch size is too small.

EC.4.4. Additional results with hyperbolic discounting

In Figures EC.3 and EC.4, we repeat the experiments above for the alternative choice of hyperbolic discounting, taking $\nu(\tau) = 1/(1 + k\tau)$ for varied k. Here, the optimal batch size for fixed k is very sensitive to the time horizon T, so we use a fixed set of batch sizes for each choice of $n \in \{3, 10\}$. Interestingly, the multi-threaded algorithm occasionally outperforms all of batched algorithms. To understand why this is possible, note that the number of future rounds which the agent can impact with their current action is substantially fewer with multi-threading, since the principal never commits to any fixed strategy for very long. Moreover, it is natural that this phenomenon is more pronounced with the less-aggressive, hyperbolic discounting. Indeed, a collection of many future rounds can impact the agent's discounted utility far more than any single future round (whereas they are within a factor of T_γ under geometric discounting).

EC.5. Supplementary Material for Demand Learning (Section 5.1)

EC.5.1. Stochastic bandits with delays and perturbations (proof of Lemma 7)

Without loss of generality, we assume that each random interval $[\ell_t, u_t]$ is always contained within $[0, 1]$. For analysis, it will be convenient to define empirical counts, means, and confidence bounds for all arms i and rounds t as

$$n_i(t) = \max\left\{\sum_{\tau=1}^{t} \mathbb{1}\{i_\tau = i\}, 1\right\}, \quad \hat{\mu}_i(t) = \frac{1}{n_i(t)} \sum_{\tau=1}^{t} \mathbb{1}\{i_\tau = i\} r_\tau$$

$$\mathrm{LCB}_i(t) = \hat{\mu}_i(t) - \sqrt{2\log(T)/n_i(t)} - \delta, \quad \mathrm{UCB}_i(t) = \hat{\mu}_i(t) + \sqrt{2\log(T)/n_i(t)} + \delta$$

To start, we show that the confidence intervals are valid with high probability.

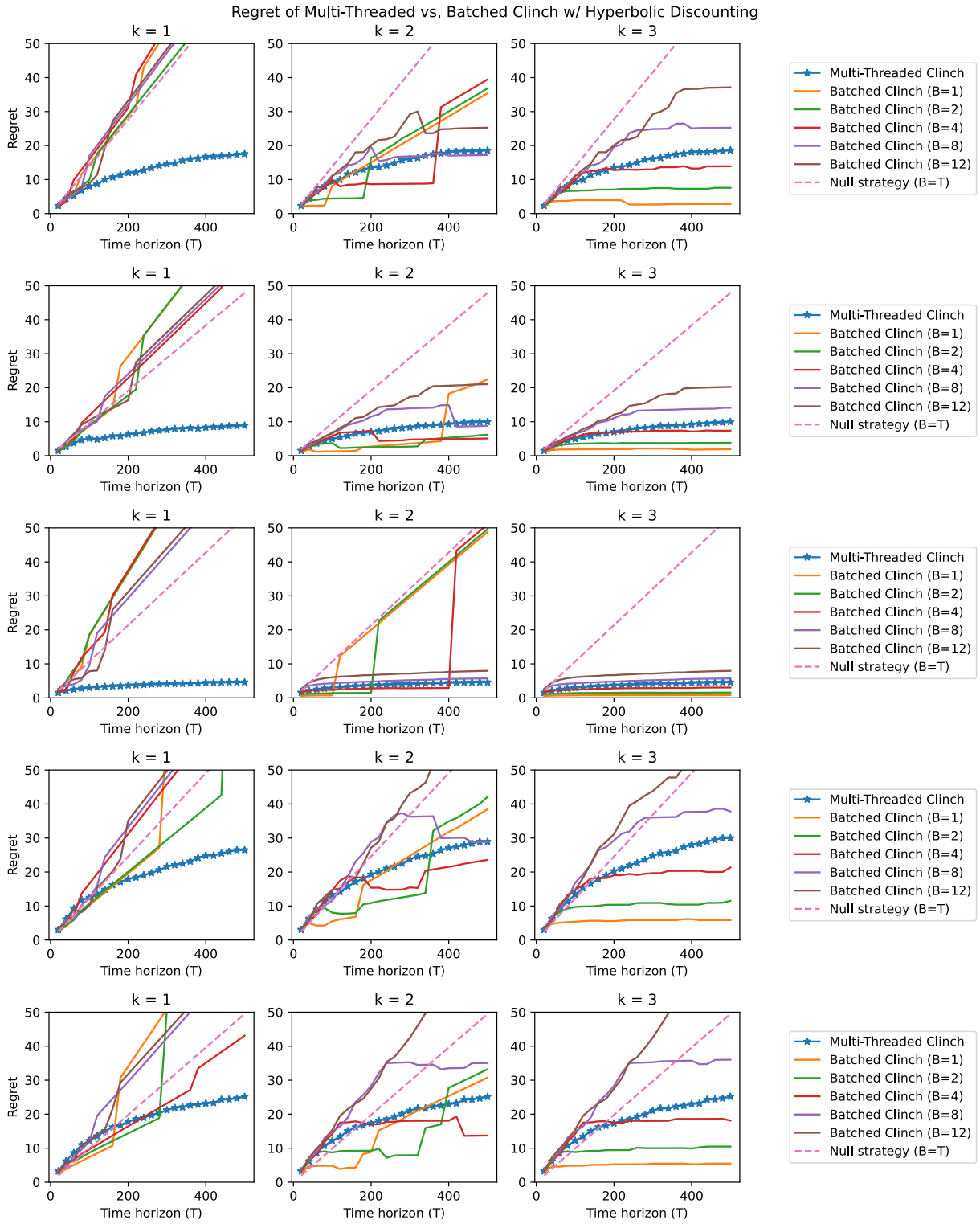

Figure EC.3 Regret achieved by batched and multi-threaded variants of Clinch against a simulated hyperbolic discounting agent on five random SSG instances with $n = 3$.

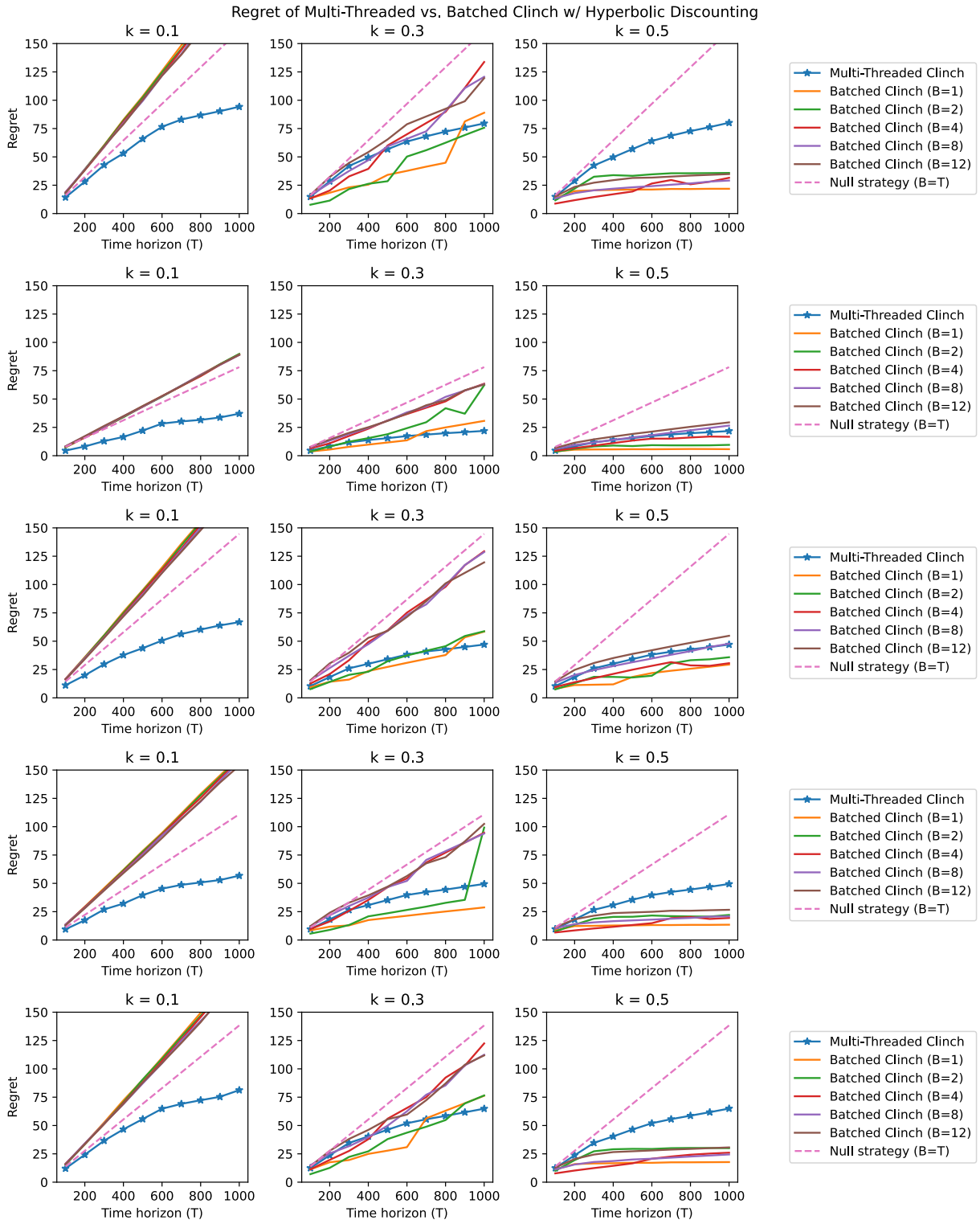

Figure EC.4 Regret achieved by batched and multi-threaded variants of Clinch against a simulated hyperbolic discounting agent on five random SSG instances with $n = 10$.

LEMMA EC.5. *With probability* $1-\frac{2}{T^3}$*, we have* $\mathrm{LCB}_i(t) \le \mu_i(t) \le \mathrm{UCB}_i(t)$ *for each arm* i *and round* t*.*

Proof. If arm i has not been pulled by time t, the confidence bound $[\mathrm{LCB}_i(t), \mathrm{UCB}_i(t)]$ is trivially valid. Otherwise, conditioning on any arm pulls $i_1, \dots, i_t$ and considering the intervals $[\ell_t, u_t]$ guaranteed by the perturbation bound, Hoeffding's inequality implies that the corresponding empirical mean $\hat{\mu}_i(t)$ satisfies

$$\hat{\mu}_i(t) = \frac{1}{n_i(t)} \sum_{\substack{\tau \le t \\ i_\tau = i}} r_\tau \le \frac{1}{n_i(t)} \sum_{\substack{\tau \le t \\ i_\tau = i}} u_t \le \mu_i + \delta + \sqrt{\frac{2\log T}{n_i(t)}}$$

with probability at least $1-\frac{1}{T^4}$. Likewise, we have $\hat{\mu}_i \ge \mu_i - \delta - \sqrt{2\log(T)/n_i(t)}$ with the same probability. Taking a union bound gives $\mu_i \in [\mathrm{LCB}_i(t), \mathrm{UCB}_i(t)]$ with probability at least $1-\frac{2}{T^4}$. Since one confidence interval is modified per round, a union bound over rounds gives the lemma. □

Next, we note an arm i can only contribute $O(\delta)$ regret in a given round if $\Delta_i = O(\delta)$. Hence, we call an arm *acceptable* if $\Delta_i < 8\delta$ and *unacceptable* otherwise. Conditioned on the "clean" event above, we show that the number of unacceptable arm pulls is bounded, extending the analysis from Theorem 2 of Lancewicki et al. (2021) to the perturbed setting.

LEMMA EC.6. *Conditioned on the event from Lemma EC.5, no unacceptable arm* i *is pulled more than* $128\log T/\Delta_i^2 + D/m + 2$ *times, where* m *is the number of remaining arms when it is pulled last.*

Proof. To simplify analysis, we split the history of the algorithm into epochs, where epoch $\ell = 1, 2, \dots$ denotes the ℓ-th iteration of SUCCELIMDELAYED's main while loop. With this convention, after ℓ epochs, the remaining arms in S have been pulled exactly ℓ times.

Now fix any unacceptable arm i, and consider the first epoch ℓ such that running UPDATEBOUNDS with the full (non-delayed) history through epoch ℓ would eliminate arm i after the corresponding update to S. Then i must be truly eliminated after $D+m$ additional rounds have passed, where m is the number of arms remaining when i is pulled for the last time. During these extra rounds, i

is pulled at most $D/m+1$ times due to the round-robin nature of arm pulls; this is the overhead from delayed feedback.

Now if $\ell \le 1$, we are done. Otherwise, fix S as the set of arms remaining after the final round t of epoch $\ell-1$, and let $\tilde{S} = \{j \in S : \mathrm{UCB}_j(t) \ge \mathrm{LCB}_k(t) \text{ for all } k \in S\}$ denote the hypothetical update to S based on non-delayed data. By the minimality of ℓ, we know that $i \in \tilde{S}$, and so

$$\hat{\mu}_i(t) \ge \max_{j \in S} \hat{\mu}_j(t) - 2\sqrt{\frac{2\log T}{\ell-1}} - 2\delta \ge \max_j \mu_j - 3\sqrt{\frac{2\log T}{\ell-1}} - 3\delta,$$

where the second inequality follows by conditioning (noting in particular that the optimal arm is not eliminated). On the other hand, we have

$$\hat{\mu}_i(t) \le \mu_i + \sqrt{\frac{2\log T}{\ell-1}} + \delta = \max_j \mu_j - \Delta_i + \sqrt{\frac{2\log T}{\ell-1}} + \delta.$$

Combining, we find that

$$\ell \le \frac{32\log T}{(\Delta_i - 4\delta)^2} + 1 \le \frac{128\log T}{\Delta_i^2} + 1.$$

Adding this upper bound to the overhead from delays gives the lemma. □

Now we are equipped to prove the main result.

Proof of Lemma 7. By Lemma EC.6, we control regret by

$$\sum_{\Delta_i \ge 8\delta} n_T(i)\Delta_i + 8\delta T \le 128 \sum_{\Delta_i > 0} \left(\frac{\log T}{\Delta_i} + \frac{D}{m_i} + 1 \right) + 8\delta T,$$

where m_i is the number of remaining arms when arm i is pulled last. Bounding $\sum_i \frac{1}{m_i} \le \sum_{i=1}^K \frac{1}{i} \le \log(K)+2$, we obtain a final bound of $128\sum_{\Delta_i>0} \frac{\log(3T)}{\Delta_i} + 128D\log(K) + 8\delta T$. □

EC.5.2. Perturbation bound for stochastic values (proof of Lemma 8)

Proof of Lemma 8. If $a=1$, then $v_t \ge p-\varepsilon$ and $a = \mathbb{1}\{v_t \ge p-\varepsilon\} = u$, while, if $a=0$, then $v_t \le p+\varepsilon$ and $a = \mathbb{1}\{v_t > p+\varepsilon\} = \ell$. Moreover, we have

$$p\mathrm{E}[u] = p\Pr(v_t \ge p-\varepsilon) \le pd(p) + pL\varepsilon \le f(p) + L\varepsilon,$$

using the definitions of d and f, the Lipschitz property of d, and that $p \in [0,1]$. Likewise, we bound

$$p\mathrm{E}[\ell] = p\Pr(v_t > p+\varepsilon) = p\Pr(v_t \ge p+\varepsilon) \ge f(p) - L\varepsilon. \quad \square$$

EC.6. Supplementary Material for Finite Stackelberg Games (Section 5.2)

This section provides full details and analysis for MultiThreadedRobustStack (Algorithm 10) to prove Theorem 6. We introduce this algorithm and its principal subroutine RobustStack (Algorithm 9) in Section EC.6.1. In Section EC.6.3, we state a search guarantee for RobustStack, Lemma EC.7, and use it to prove the theorem. In Section EC.6.3, we state three lemmas, pertaining to polytope conditioning bounds and robust convex optimization with membership queries, and use them to prove Lemma EC.7. We prove the remaining lemmas in Sections EC.6.4 and EC.6.5. Throughout, we make use of the constants r, Δ, and V defined in Section 5.2.

EC.6.1. Algorithm definitions and discussion

We first present RobustStack (Algorithm 9), a procedure for learning in finite games with ε-approximate best responses. Formally, the algorithm takes as input a desired search accuracy δ and an approximate best response oracle Oracle which, given query $\mathbf{x} \in \mathcal{X}$, returns $\textsc{Oracle}(\mathbf{x}) \in \mathsf{BR}^{\varepsilon}(\mathbf{x})$ for some $\varepsilon \geq 0$. For meaningful guarantees, we require $\varepsilon \leq \left(\frac{\delta r \Delta}{nm}\right)^{O(1)}$. This will later be implemented against discounting agents via delayed feedback. RobustStack outputs a δ-approximate Stackelberg equilibrium pair (see Section EC.6.2 for a formal statement).

The algorithm initially samples $\widetilde{O}(V^{-1})$ points from $\mathcal{X}$ uniformly at random. For each sampled point $\mathbf{x}$, we obtain an approximate best response y from Oracle and run ConservativeBestResponse (Algorithm 11 in Section EC.6.4). This subroutine tests whether $\mathbf{x}$ is robustly within K_y using multiple queries to Oracle in the neighborhood of $\mathbf{x}$. We are left with a collection $\{\mathbf{x}^{(y)}\}_{y \in \mathcal{Y}_0}$ of sampled points which passed this test. Under the regularity assumptions, we prove that $y^\star \in \mathcal{Y}_0$ and that each $\mathbf{x}^{(y)}$ is well centered within K_y with high probability.

Next, for each $y \in \mathcal{Y}_0$, we run an optimization procedure MembershipOpt (Algorithm 12 in Section EC.6.5) to find an δ-approximate maximizer $\hat{\mathbf{x}}^{(y)}$ for $u(\cdot, y)$ over K_y, starting at initial point $\mathbf{x}^{(y)}$. This subroutine applies convex optimization with membership queries, noting that $\mathbb{1}\{\textsc{Oracle}(\mathbf{x}) = y\} \approx \mathbb{1}\{\mathbf{x} \in K_y\}$. RobustStack then returns the strategy $\hat{\mathbf{x}}^{(y)}$ maximizing $u(\hat{\mathbf{x}}^{(y)}, y)$.

Algorithm 9: RobustStack: robust learning for finite Stackelberg games

input : search accuracy $\delta \geq 0$, approximate best response oracle Oracle
output: δ-optimal principal strategy $\hat{\mathbf{x}} \in \mathcal{X}$

$\mathcal{Y}_0 \leftarrow \emptyset$, $\eta \leftarrow \delta(3\lceil V^{-1} \log \frac{3}{\delta} \rceil)^{-1}$
for $i = 1, \dots, \lceil V^{-1} \log \frac{3}{\delta} \rceil$ **do**
  $y \leftarrow \text{Oracle}(\mathbf{x})$ for $\mathbf{x}$ sampled uniformly at random from $\mathcal{X}$
  if $y \notin \mathcal{Y}_0$ *and* ConservativeBestResponse$(y, \mathbf{x}, r/2, \eta, \text{Oracle}) = \text{True}$ **then**
    $\mathcal{Y}_0 \leftarrow \mathcal{Y}_0 \cup \{y\}$, $\mathbf{x}^{(y)} \leftarrow \mathbf{x}$
for $y \in \mathcal{Y}_0$ **do** $\hat{\mathbf{x}}^{(y)} \leftarrow \text{MembershipOpt}(y, \mathbf{x}^{(y)}, \frac{\delta}{3n}, \text{Oracle})$
return $\hat{\mathbf{x}}^{(\hat{y})}$ *for* $\hat{y} \in \arg\max_{y \in \mathcal{Y}_0} u(\hat{\mathbf{x}}^{(y)}, y)$

Finally, we present MultiThreadedRobustStack (Algorithm 10), a policy for the repeated game with γ-discounting agents (for unknown γ) that mirrors the layered approach of MultiThreadedClinch. As before, each of $O(\log T)$ parallel threads runs a separate instance of RobustStack, with thread k experiencing delay 2^k. Once a copy of RobustStack terminates, its thread always plays the strategy returned by the largest eligible thread, where thread k becomes eligible 2^k rounds after termination.

Algorithm 10: MultiThreadedRobustStack

for *thread* $k = 1, \dots, \lfloor \log T \rfloor + 1$ **do**
  Initialize copy $\mathcal{A}^{(k)}$ of RobustStack with $\delta = T^{-1}$
for *round* $t = 1, \dots, T$ **do**
  $k \leftarrow \arg\max\{\ell \in \mathbb{N}_{>0} : 2^{\ell-1} \text{ divides } t\}$ `// Identify current thread`
  if $\mathcal{A}^{(k)}$ *has not terminated* **then**
    Simulate oracle query/response for $\mathcal{A}^{(k)}$ using $\mathbf{x}^{(t)}, y_t$
    if $\mathcal{A}^{(k)}$ *terminates with output* $\hat{\mathbf{x}}$ **then** $\hat{\mathbf{x}}^{(k)} \leftarrow \hat{x}$
  else Play $\mathbf{x}^{(t)} \leftarrow \hat{\mathbf{x}}^{(\bar{k})}$, where $\bar{k} = \max\{\ell : \text{thread } \ell \text{ terminated by round } t - 2^\ell\}$

EC.6.2. Learning against γ-discounting agents (proof of Theorem 6)

We now provide a formal guarantee for RobustStack, deferring the proof to Section EC.6.3.

Lemma EC.7. *Fix $\delta \in (0,1)$, and let* Oracle *be an ε-approximate best response oracle for some $\varepsilon \geq 0$. Then* RobustStack(δ, Oracle) *terminates after at most* $100V^{-1}\sqrt{m}\log^2\left(\frac{3}{\delta}\right)\log V^{-1} + 10^7 m^{2.5} n \log^3\left(\frac{10mn}{\delta r}\right)$ *oracle calls. If $\varepsilon \leq \left(\frac{\delta r}{200nm}\right)^{20}\Delta$, then, with probability at least $1-\delta$, the returned strategy $\hat{\mathbf{x}} \in \mathcal{X}$ satisfies $u(\hat{\mathbf{x}}, y) \geq \max_{\mathbf{x} \in \mathcal{X}} u(\mathbf{x}, \mathsf{br}(\mathbf{x})) - \delta$ for all $y \in \mathsf{BR}^{\varepsilon}(\hat{\mathbf{x}})$.*

We are now equipped to prove the main theorem.

Proof of Theorem 6. Denote by $Q = 100V^{-1}\sqrt{m}\log^2(3T)\log V^{-1} + 10^7 m^{2.5} n \log^3(10mnT/r)$ and $\varepsilon = \left(\frac{r}{200nmT}\right)^{20}\Delta$ the query complexity and oracle accuracy required by Lemma EC.7 when

$\delta = T^{-1}$. As with MULTITHREADEDCLINCH, thread k runs on rounds $2^{k-1}(2\ell-1)$ for $\ell = 1, 2, \ldots$ and hence experiences delay 2^k while copy $\mathcal{A}^{(k)}$ of ROBUSTSTACK is running. For this copy's final oracle call, the selection rule at Step 8 ensures that the 2^k round delay is maintained. We term the rounds up to this point for thread k its "exploration phase," since it is learning from agent feedback. Once $\mathcal{A}^{(k)}$ has terminated, thread k enters an exploitation phase and ignores agent feedback (i.e., infinite feedback delay).

We now establish several consequences of Lemma EC.7, applied with our choice of $\delta = T^{-1}$. First, we can bound the total number of exploration rounds by $(\lfloor \log T \rfloor + 1)Q$. Next, let $k^\star = \log_2 \left\lceil T_\gamma \log \frac{T_\gamma}{\varepsilon} \right\rceil$ be the index of the first thread whose delay during exploration induces ε-approximate best responses by Proposition 1 (we can assume that $k^\star \le \lfloor \log T \rfloor + 1$ is a valid thread index; otherwise the regret bound holds trivially). We freely condition on the event that the search of $\mathcal{A}^{(k)}$ terminates successfully for all $k \ge k^\star$, since the complement has probability at most $O(T^{-1} \log T)$ by a union bound over threads. Starting at time $2^{k^\star}(Q+1)$, once copy $\mathcal{A}^{(k^\star)}$ has terminated and a delay of $2^{k^\star}$ has passed, the strategy $\hat{\mathbf{x}}^{(\bar{k})}$ played at Step 8 incurs regret at most T^{-1}, since $\bar{k} \ge k^\star$. Combining the above, we bound the regret by

$$\begin{aligned}
(\lfloor \log T \rfloor + 1)Q + 2^{k^\star}(Q+1) + 1 &= (\lfloor \log T \rfloor + 1)Q + \left\lceil T_\gamma \log \frac{T_\gamma}{\varepsilon} \right\rceil (Q+1) + 1 \\
&= \tilde{O}\left(\left(\log T + T_\gamma \log \frac{1}{r\Delta} \right) \left(V^{-1} \sqrt{m} \log^2 T + m^{2.5} n \log^3 \frac{T}{r} \right) \right) \\
&= O\left(T_\gamma \left(V^{-1} \sqrt{m} \log^3(T) \log \frac{1}{r\Delta} + m^{2.5} n \log^4 \frac{T}{r} \log \frac{1}{\Delta} \right) \right).
\end{aligned}$$

Substituting the given values for Q and ε gives the desired bound. □

EC.6.3. Robust search with ε-approximate best responses (proof of Lemma EC.7)

Our search guarantee for ROBUSTSTACK relies on several lemmas, which we state below after establishing notation. For ease of presentation, we assume in what follows that the principal has $|\mathcal{X}_0| = m+1$ actions, rather than m, so that $\mathcal{X}$ can be identified with its isometric embedding into the ball $B(\sqrt{2}) \subset \mathbb{R}^m$ (this poses no difficulties as the bulk of our analysis is coordinate-free). We

write $B(A,r) \subset \mathbb{R}^m$ for the Minkowski sum of a set or point A in $\mathbb{R}^m$ with the ℓ_2-ball of radius r, with $B(r) \coloneqq B(\mathbf{0}_m, r)$, and, for a set $A \subseteq \mathbb{R}^m$, write $B(A,-r) \coloneqq \{\mathbf{x} \in A : B(\mathbf{x},r) \subseteq A\}$.

Finally, for any (potentially negative) $\varepsilon \in \mathbb{R}$ and $\mathbf{x} \in \mathcal{X}$, we define $\mathsf{BR}^\varepsilon(\mathbf{x}) \coloneqq \{y \in \mathcal{Y} : v(\mathbf{x},y) \geq v(\mathbf{x},y') - \varepsilon \; \forall y' \in \mathcal{Y} \setminus \{y\}\}$. For each $y \in \mathcal{Y}$, let $K_y^\varepsilon \coloneqq \{\mathbf{x} \in \mathcal{X} : y \in \mathsf{BR}^\varepsilon(\mathbf{x})\} \subseteq \mathbb{R}^m$ and set $K_y \coloneqq K_y^0$. Negative values of ε are relevant because they control the extent to which neighboring $|\varepsilon|$-approximate best response regions can overlap with K_y. In particular, if $\varepsilon \geq 0$ and $x \in K_y^{-2\varepsilon}$, then $\mathsf{BR}^\varepsilon(\mathbf{x}) = \{y\}$ (with the constant of two taken to avoid reliance on tie-breaking).

We first provide a correctness guarantee for ConservativeBestResponse.

Lemma EC.8. *Fix $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, margin $\lambda \geq 0$, and failure probability $\delta \in (0,1)$. Let* Oracle *be an ε-approximate best response oracle for $\varepsilon \geq 0$. Then,* ConservativeBestResponse$(y, \mathbf{x}, \lambda, \delta,$ Oracle$)$ *terminates after $12\sqrt{m}\log\delta^{-1}$ oracle calls. If further $\varepsilon \leq \frac{\lambda\Delta}{6\sqrt{m}}$, then, with probability at least $1-\delta$, the subroutine returns* True *only if $\mathbf{x} \in B\big(K_y^{-2\varepsilon}, -\frac{\lambda}{2\sqrt{m}}\big)$ and* False *only if $\mathbf{x} \notin B(K_y^{-2\varepsilon}, -\lambda)$.*

Within the proof of Lemma EC.8, we show the following useful fact.

Lemma EC.9. *For each $y \in \mathcal{Y}$ and $\varepsilon \geq 0$, we have $B\big(K_y, -\varepsilon/\Delta\big) \subseteq K_y^{-\varepsilon}$.*

This result translates a distance margin of ε/Δ from the boundary of K_y to a utility margin of ε. Next, we give an optimization guarantee for MembershipOpt.

Lemma EC.10. *Fix $y \in \mathcal{Y}$, accuracy $\delta \in (0,1)$, initial point $\mathbf{x}_0 \in \mathcal{X}$, and radius $\rho > 0$. Let* Oracle *be an ε-approximate best response oracle for $\varepsilon \geq 0$. Then* MembershipOpt$(y, \delta, \mathbf{x}_0, \rho,$ Oracle$)$ *terminates after at most $10^5 m^{2.5}\log^3\big(\frac{120m}{\delta\rho}\big)$ oracle calls. If further $\varepsilon \leq \big(\frac{\delta\rho}{140m}\big)^{13}\Delta$ and $B(x_0,\rho) \subseteq K_y$, then, with probability at least $1-\delta$,* MembershipOpt *returns $\hat{\mathbf{x}} \in \mathcal{X}$ such that $\mathsf{BR}^\varepsilon(\hat{\mathbf{x}}) = \{y\}$ and $u(\hat{x},y) \geq \max_{\mathbf{x} \in K_y} u(\mathbf{x},y) - \delta$.*

Together, these suffice to prove the search guarantee.

Proof of Lemma EC.7. Since the initial sampling loop at Steps 2-5 calls ConservativeBestResponse $N = \lceil V^{-1}\log\frac{3}{\delta}\rceil$ times with margin $\lambda = r/2$, and $\varepsilon \leq \frac{\lambda\Delta}{6\sqrt{m}}$, the accuracy guarantee of

Lemma EC.8 holds for all calls within the loop with probability at least $1-\delta/3$. Conditioned on this event, we analyze the sampling loop. Let $S \subseteq K_{y^\star}$ denote the ball of radius $2r$ guaranteed by the regularity assumptions. The sample count N is taken sufficiently large such that some sampled point $\mathbf{x}$ will lie inside $B(S,-r) \subseteq B(K_{y^\star},-r)$ with probability at least $1-\delta/3$; indeed, the probability that any single point lies inside $B(S,-r)$ is V. Condition further on this event.

Since $\varepsilon \le r\Delta/4$, Lemma EC.9 implies that $\mathbf{x} \in B(K_{y^\star}^{-2\varepsilon}, -r/2)$, and so Oracle will return $y^\star$ when $\mathbf{x}$ is queried. Moreover, since ConservativeBestResponse is run with margin $r/2$, Lemma EC.8 implies that $\mathbf{x}$ will pass the check at Step 4 unless $\mathcal{Y}_0$ already contains $y^\star$. Thus, the final set $\mathcal{Y}_0$ will contain $y^\star$. Moreover, for each $y \in \mathcal{Y}_0$, Lemma EC.8 requires that the accepted strategy $\mathbf{x}^{(y)}$ have margin at least $\frac{r}{4\sqrt{m}}$ within K_y^ε. Again by Lemma EC.8, this sampling loop terminates within $N \cdot (1 + 12\sqrt{m}\log\frac{3N}{\delta})$ queries to Oracle.

Finally, we examine the search loop at Step 6, where each call to MembershipOpt is run with accuracy $\gamma = \frac{\delta}{3n}$ and radius $\rho = \frac{r}{4\sqrt{m}}$. Since $\varepsilon \le \left(\frac{\gamma\rho}{140m}\right)^{13}\Delta$, Lemma EC.10 that implies that the search over K_y will terminate with a γ-approximate maximizer $x^{(y)}$ within $10^5 m^{2.5}\log^3\left(\frac{120m}{\gamma\rho}\right)$ queries to Oracle, with total success probability over all regions at least $1-\delta/3$. Taking a union bound over the three high probability events, we see that RobustStack returns the desired maximizer with probability at least $1-\delta$. Indeed, conditioned on these good events, the returned strategy $\hat{\mathbf{x}}^{(\hat{y})}$, with $\hat{y} \in \arg\max_{y\in\mathcal{Y}_0} u(\mathbf{x}^{(y)}, y)$ satisfies

$$u(\hat{\mathbf{x}}^{(\hat{y})}, \hat{y}) = \max_{y\in\mathcal{Y}_0} u\big(\mathbf{x}^{(y)}, y\big) \ge \max_{\mathbf{x}\in K_{y^\star}} u(\mathbf{x}, y^\star) - \gamma = \max_{x\in\mathcal{X}} u(\mathbf{x}, \mathsf{br}(\mathbf{x})) - \gamma > \max_{x\in\mathcal{X}} u(\mathbf{x}, \mathsf{br}(\mathbf{x})) - \delta,$$

and $\mathsf{BR}^\varepsilon(\hat{\mathbf{x}}^{(\hat{y})}) = \{\hat{y}\}$, as desired.

For the final query complexity, we bound

$$\begin{aligned}
&N \cdot \left(1 + 12\sqrt{m}\log\frac{3N}{\delta}\right) + n \cdot 10^5 m^{2.5}\log^3\left(\frac{120m}{\gamma\rho}\right) \\
&\le 2V^{-1}\log\frac{3}{\delta} \cdot \left(1 + 12\sqrt{m}\log\frac{3V^{-1}\log\frac{3}{\delta}}{\delta}\right) + n \cdot 10^5 m^{2.5}\log^3\left(\frac{120\cdot 12m^{3.5}n}{\delta r}\right) \\
&< 100V^{-1}\sqrt{m}\log^2\left(\frac{3}{\delta}\right)\log V^{-1} + 10^7 m^{2.5} n \log^3\left(\frac{10mn}{\delta r}\right),
\end{aligned}$$

as desired. □

EC.6.4. Conservative best response data (proof of Lemma EC.8)

To ensure that the principal may safely commit to a strategy despite inexact best response feedback, we introduce ConservativeBestResponse (Algorithm 11) to determine whether a strategy $\mathbf{x}$ lies robustly within the best response polytope K_y. Specifically, given $\mathbf{x} \in \mathcal{X}$, we query the best response oracle at $O(\sqrt{m}\log\delta^{-1})$ small perturbations of $\mathbf{x}$, only returning True if the oracle always responds with the fixed action y. If K_y is sufficiently well-conditioned, a True output indicates that $\mathbf{x}$ lies firmly within the interior of K_y with high probability, despite inexact best responses.

Algorithm 11: ConservativeBestResponse

input : action $y \in \mathcal{Y}$, query $\mathbf{x} \in \mathbb{R}^m$, margin $\lambda \geq 0$, failure probability $\delta \in (0,1)$, approximate best response oracle Oracle
output: conservative estimate of $\mathbb{1}\{\mathbf{x} \in K_y\}$

for $i = 1$ *to* $\lceil 6\sqrt{m}\log\delta^{-1}\rceil$ **do**
  $\mathbf{w}_i \leftarrow \mathbf{x} + \lambda\mathbf{S}_i$, where $\mathbf{S}_i$ is sampled uniformly at random from $\mathbb{S}^{m-1}$
  if Oracle$(\mathbf{w}_i) \neq y$ *or* $\mathbf{w}_i \notin \mathcal{X}$ **then return** False
return True

Our accuracy guarantee for ConservativeBestResponse relies on the following lemma, which provides a certain conditioning bound on each best response polytope K_y^ε as ε varies.

Lemma EC.11. *If $y \in \mathcal{Y}$, $\lambda \geq 0$, and $\varepsilon_1, \varepsilon_2 \in \mathbb{R}$ with $\varepsilon_1 \leq \varepsilon_2$, then any $\mathbf{x} \in K_y^{\varepsilon_2}$ with margin $\lambda + (\varepsilon_2 - \varepsilon_1)/\Delta$ has margin λ within $K_y^{\varepsilon_1}$. That is, we have $B\big(K_y^{\varepsilon_2}, -(\lambda + (\varepsilon_2 - \varepsilon_1)/\Delta)\big) \subseteq B(K_y^{\varepsilon_1}, -\lambda)$.*

Proof. For this proof, we will view $\mathcal{X}$ and the best response polytopes as subsets of the affine subspace $A \coloneqq \{\mathbf{w} \in \mathbb{R}^{m+1} : \sum_{i=1}^{m+1} w_i = 1\}$ in the natural way (rather than their isometric embeddings into $\mathbb{R}^m$). With this change, the sets of the form $B(S, -r)$ in the statement should be updated to $B_A(S, r) \coloneqq \{\mathbf{x} \in S : B(\mathbf{x}, r) \cap A \subseteq S\}$. For readability, we also write $\bar{\mathbf{v}}_y = \bar{\mathbf{v}}^{(y)} \in \mathbb{R}^{m+1}$ for each $y \in \mathcal{Y}$, where these are the centered utility profiles defined in the regularity assumptions. We naturally extend to the uncentered utility profiles, defined for each $y \in \mathcal{Y}$ and $i \in [m+1]$ by $v_y(i) \coloneqq v_0(i, y)$. Writing $c_y \coloneqq \frac{1}{m+1}\sum_{i=1}^{m+1} v_0(i, y)$, we have $\bar{\mathbf{v}}_y = \mathbf{v}_y - c_y\mathbf{1}_{m+1}$ for each $y \in \mathcal{Y}$.

We shall prove the contrapositive. To start, fix any $\mathbf{x} \in A \setminus B_A(K_y^{\varepsilon_1}, -\lambda)$. That is, there exists $\mathbf{x}' \in A \setminus K_y^{\varepsilon_1}$ such that $\|\mathbf{x} - \mathbf{x}'\|_2 < \lambda$. Since $K_y^{\varepsilon_1}$ is a polytope, a hyperplane tangent to one of its faces must separate $\mathbf{x}$ and $\mathbf{x}'$. If this hyperplane corresponds to one of the non-negativity constraints defining $\mathcal{X} = \{\mathbf{w} \in A : w_i \geq 0 \,\forall i\}$, then $\mathbf{x}' \in A \setminus \mathcal{X}$, and so we trivially have $\mathbf{x} \in A \setminus B_A(K_y^{\varepsilon_2}, -\lambda) \subseteq A \setminus$

$B_A\big(K_y^{\varepsilon_2}, -(\lambda + (\varepsilon_2 - \varepsilon_1)/\Delta)\big)$. Otherwise, the hyperplane must take the form $\{\mathbf{w} \in \mathbb{R}^{m+1} : \mathbf{w}^\top(\mathbf{v}_{y'} - \mathbf{v}_y) = \varepsilon_1\}$ for some $y' \in \mathcal{Y} \setminus \{y\}$. In this case, the vector $\mathbf{x}'' \in A$ defined by $\mathbf{x}'' \coloneqq \mathbf{x}' + \frac{\varepsilon_2 - \varepsilon_1}{\Delta} \frac{\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y}{\|\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y\|_2}$ must satisfy

$$\begin{aligned}
(\mathbf{x}'')^\top(\mathbf{v}_{y'} - \mathbf{v}_y) &> \varepsilon_1 + \frac{\varepsilon_2 - \varepsilon_1}{\Delta} \frac{(\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y)^\top(\mathbf{v}_{y'} - \mathbf{v}_y)}{\|\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y\|_2} \\
&= \varepsilon_1 + \frac{\varepsilon_2 - \varepsilon_1}{\Delta} \frac{(\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y)^\top(\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y + (c_{y'} - c_y)\mathbf{1}_{m+1})}{\|\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y\|_2} \\
&= \varepsilon_1 + \frac{\varepsilon_2 - \varepsilon_1}{\Delta} \frac{(\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y)^\top(\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y)}{\|\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y\|_2} \\
&= \varepsilon_1 + \frac{\varepsilon_2 - \varepsilon_1}{\Delta} \|\bar{\mathbf{v}}_{y'} - \bar{\mathbf{v}}_y\|_2 > \varepsilon_2,
\end{aligned}$$

using our minimum distance assumption. Thus, $\mathbf{x}'' \in A \setminus K_y^{\varepsilon_2}$ and, since $\|\mathbf{x}'' - \mathbf{x}\|_2 < \lambda + \frac{\varepsilon_2 - \varepsilon_1}{\Delta}$, we obtain $\mathbf{x} \in A \setminus B_A\big(K_y^{\varepsilon_2}, -(\lambda + (\varepsilon_2 - \varepsilon_1)/\Delta)\big)$, as desired. □

As a simple consequence, we obtain Lemma EC.9.

Proof of Lemma EC.9. Setting $\lambda = \varepsilon_2 = 0$ and $\varepsilon_1 = -\varepsilon$, we find that $B\big(K_y, -\varepsilon/\Delta\big) \subseteq K_y^{-\varepsilon}$. □

Next, to analyze the sampling procedure of ConservativeBestResponse, we recall a standard lower bound for the volume of a spherical cap (see, e.g., Lemma 9 of Feige and Schechtman 2002). We provide a brief proof below to clarify the constant prefactor.

Lemma EC.12. *Let* $\mathbf{Z} \sim \mathrm{Unif}(\mathbb{S}^{m-1})$ *and* $t \geq 0$. *Then* $\Pr(Z_1 > t) \geq \frac{1}{\sqrt{2\pi m}}(1 - t^2)^{(m-1)/2}$.

Proof. The set $\{\mathbf{z} \in \mathbb{S}^{m-1} : z_1 > t\}$ is an open spherical cap, whose boundary is an $(m-1)$-dimensional sphere with radius $\sqrt{1-t^2}$. The surface area of the cap is bounded from below by the volume of the sphere, which is given by $(1-t^2)^{(m-1)/2} \frac{\pi^{(m-1)/2}}{\Gamma((m+1)/2)}$. Normalizing by the surface area of $\mathbb{S}^{m-1}$ gives

$$\Pr(Z_1 > t) \geq \frac{(1-t^2)^{(m-1)/2}\Gamma(m/2)}{\sqrt{\pi}\Gamma((m+1)/2)} \geq \frac{(1-t^2)^{(m-1)/2}}{\sqrt{2\pi m}},$$

as desired, using that $\Gamma(m/2)/\Gamma((m+1)/2) \geq 1/\sqrt{2m}$. □

Finally, we prove the desired guarantee for ConservativeBestResponse.

Proof of Lemma EC.8 Suppose that a query $\mathbf{x} \in \mathcal{X}$ does not lie robustly within K_y, in that $\mathbf{x} \notin B\big(K_y^{-2\varepsilon}, -\frac{\lambda}{2\sqrt{m}}\big)$. Then, applying Lemma EC.11 with $\varepsilon_1 = -2\varepsilon$, $\varepsilon_2 = \varepsilon$, and margin $\frac{\lambda}{2\sqrt{m}}$, we find that

$\mathbf{x} \notin B\big(K_y^\varepsilon, -\frac{\lambda}{2\sqrt{m}} - 3\varepsilon/\Delta\big)$. Assuming that $\varepsilon \le \frac{\lambda\Delta}{6\sqrt{m}}$, this implies that $\mathbf{x} \notin B\big(K_y^\varepsilon, -\lambda/\sqrt{m}\big)$. There thus exists an open half-space H tangent to (but disjoint from) K_y^ε, such that $d(\mathbf{x}, H) \le \lambda/\sqrt{m}$. Next, consider $\mathbf{w} = \mathbf{x} + \lambda\mathbf{S}$, for $\mathbf{S} \sim \mathrm{Unif}(\mathbb{S}^{m-1})$, as in Step 2. By Lemma EC.12, we must have

$$\begin{aligned}
\Pr(\mathbf{w} \notin K_y^\varepsilon) &\ge \Pr(\mathbf{w} \in H) \\
&= \Pr(\lambda S_1 > \lambda/\sqrt{m}) \\
&= \Pr(S_1 > 1/\sqrt{m}) \\
&\ge \frac{1}{\sqrt{2\pi m}(1 - \frac{1}{m})^{(m-1)/2}} \\
&\ge \frac{1}{2\sqrt{2\pi m}} \ge \frac{1}{6\sqrt{m}}.
\end{aligned}$$

Consequently, the probability that ConservativeBestResponse returns True is at most

$$\big(1 - \tfrac{1}{6}m^{-1/2}\big)^{6\sqrt{m}\log\delta^{-1}} \le \exp(\log\delta) = \delta.$$

On the other hand, if $\mathbf{x} \in B\big(K_y^{-2\varepsilon}, -\lambda\big)$, the algorithm will return True with probability 1, as desired. Regardless of whether ε is sufficiently small, the total number of calls to Oracle is at most $\lceil 6\sqrt{m}\log\delta^{-1}\rceil \le 12\sqrt{m}\log\delta^{-1}$. □

EC.6.5. Robust linear optimization with membership queries (proof of Lemma EC.10)

Our second subroutine, MembershipOpt (Algorithm 12), seeks to maximize the linear objective $u(\cdot, y)$ over a fixed best response region K_y, using queries to an ε-approximate best response oracle Oracle. Since the feedback $\mathbb{1}\{\textsc{Oracle}(\mathbf{x}) = y\}$ approximates $\mathbb{1}\{\mathbf{x} \in K_y\}$, our approach mirrors existing work for robust convex optimization with membership queries (c.f. Lee et al. 2018).

In particular, we introduce a method SimulatedSep (Algorithm 13) that simulates a conservative separation oracle for K_y. That is, unless a query $\mathbf{x} \in \mathbb{R}^m$ lies within $K_y^{-2\varepsilon}$, SimulatedSep returns a normal vector $\mathbf{w} \in \mathbb{S}^{m-1}$ for a half-space approximately separating $\mathbf{x}$ from K_y. To achieve this, we implement a conservative membership oracle for K_y using ConservativeBestResponse and apply a reduction from separation to membership due to Lee et al. (2018).

We then apply the standard center of gravity method (c.f. Section 2.1 of Bubeck 2015) to maximize our objective over K_y using separation queries. More precisely, at each round t, we query SimulatedSep at the centroid $\mathbf{x}_t$ of the current search space S_t and update S_{t+1} to incorporate the obtained feedback, either intersecting S_t with the returned half-space or eliminating all strategies with $u(\mathbf{x}, y) < u(\mathbf{x}_t, y)$. After a moderate number of queries, MembershipOpt returns a queried point which maximizes $u(\cdot, y)$, among those which SimulatedSep failed to separate from K_y. We note that Clinch from Section 3 has a similar flavor, since both are cutting-plane methods.

Algorithm 12: MembershipOpt: robust linear optimization via membership queries

input : action $y \in \mathcal{Y}$, optimization accuracy $\delta \geq 0$, initial point $\mathbf{x}_0 \in \mathcal{X}$, radius $\rho > 0$, approximate best response oracle Oracle
output: approximate minimizer $\hat{\mathbf{x}}$ for $\max_{\mathbf{x} \in K_y} u(\mathbf{x}, y)$, or "$\perp$"

$S_0 \leftarrow B(\mathbf{x}_0, \sqrt{2})$, $A \leftarrow \emptyset$, $t_f \leftarrow \lceil 3m \log \frac{16}{\delta\rho} \rceil$, $\alpha \leftarrow \min\{\frac{\delta}{t_f+1}, \frac{\delta\rho}{16\sqrt{2m}}\}$
for $t = 0, \ldots, t_f$ **do**
  if SimulatedSep$(y, \mathbf{x}_t, \rho, \alpha,$ Oracle$)$ *returns* $\mathbf{w} \in \mathbb{S}^{m-1}$ **then**
    $S_{t+1} \leftarrow \{\mathbf{x} \in S_t : \mathbf{w}^\top \mathbf{x} \geq \mathbf{w}^\top \mathbf{x}_t\}$
  else
    $S_{t+1} \leftarrow \{\mathbf{x} \in S_t : u(\mathbf{x}, y) \geq u(\mathbf{x}_t, y)\}$
    $A \leftarrow A \cup \{\mathbf{x}_t\}$
  $\mathbf{x}_{t+1} \leftarrow \mathrm{E}_{\mathbf{x} \sim \mathrm{Unif}(S_t)}[\mathbf{x}]$
if $A = \emptyset$ **then return** $\perp$ **else return** $\arg\max_{\mathbf{x} \in A} u(\mathbf{x}, y)$

Algorithm 13: SimulatedSep: simulation of separation oracle via best response queries

input : action $y \in \mathcal{Y}$, query $\mathbf{x} \in \mathcal{X}$, radius ρ, separation accuracy $\delta \in (0, 1)$, approximate best response oracle Oracle
output: normal vector $\mathbf{w} \in \mathbb{S}^{m-1}$ of half-space approximately separating $\mathbf{x}$ from K_y, or "$\perp$"

if $\mathbf{x} \notin \mathcal{X}$ **then return** any $\mathbf{w} \in \mathbb{S}^{m-1}$ such that $\mathbf{w}^\top(\mathbf{x} - \mathbf{z}) \leq 0$ for all $\mathbf{z} \in \mathcal{X}$
$\lambda \leftarrow \frac{\delta^6 \rho^6}{2^{36} m^{7/2}}$, $Q \leftarrow 2m\lceil \log_2(2/\lambda) \rceil + 1, \gamma \leftarrow \frac{\delta}{3Q}$
Define membership oracle Mem with domain $\mathbb{R}^m$ by
  Mem$(\mathbf{x}) \leftarrow$ ConservativeBestResponse$(y, \mathbf{x}, \lambda, \gamma,$ Oracle$)$
Run Algorithm 1 of Lee et al. (2018) with query access to Mem and parameters "n" $\leftarrow m$, "r" $\leftarrow \rho/2$, "R" $\leftarrow \sqrt{2}$, "ε" $\leftarrow \lambda$; terminate after Q queries to Mem
if *Algorithm 1 asserts that no cut exists or is terminated before completion* **return** $\perp$
else if *Algorithm 1 returns half-space defined by* $\tilde{\mathbf{g}}$ **then return** $-\tilde{\mathbf{g}}/\|\tilde{\mathbf{g}}\|_2$

We first bound the query complexity of SimulatedSep. While this method always terminates after a fixed number of queries, we only obtain meaningful performance guarantees if ε is sufficiently small and K_y contains a ball with radius bounded from below.

Lemma EC.13 **(Membership to separation).** *Fix $y \in \mathcal{Y}$, $\mathbf{x} \in \mathbb{R}^m$, radius $\rho > 0$, and accuracy $\delta \in (0, 1)$. Let* Oracle *be an ε-approximate best response oracle for some $\varepsilon \geq 0$. Then,* SimulatedSep$(y, \mathbf{x}, \rho, \delta,$ Oracle$)$ *terminates after at most $10^3 m^{1.5} \log^2\left(\frac{100m}{\delta\rho}\right)$ queries to* Oracle.

If further K_y contains a ball of radius ρ and $\varepsilon \le \frac{\delta^6\rho^6\Delta}{2^{39}m^4}$, then, with probability at least $1-\delta$, SimulatedSep *only returns "$\perp$" if $\mathbf{x} \in K_y^{-2\varepsilon}$, and, if,* SimulatedSep *returns $\mathbf{w} \in \mathbb{S}^{m-1}$, then $\mathbf{x}^\top \mathbf{w} \le \mathbf{z}^\top \mathbf{w} + \delta$ for every $\mathbf{z} \in K_y^{-2\varepsilon}$.*

Proof. We first address sample complexity. Due to the manual cutoff at Step 4, we make at most $Q = 2m\lceil \log_2(2/\lambda)\rceil + 1$ queries to the simulated oracle Mem. By Lemma EC.8, each query to Mem uses at most $12\sqrt{m}\log\frac{3Q}{\delta}$ queries to Oracle. Combining gives the stated query complexity bound of

$$\begin{aligned} Q \cdot 12\sqrt{m}\log\frac{3Q}{\delta} &\le 48m^{1.5}\log_2\left(\frac{2}{\lambda}\right)\left(\log\frac{12m\log_2\frac{2}{\lambda}}{\delta}\right) \\ &= 48m^{1.5}\log_2\left(\frac{2^{37}m^{7/2}}{\delta^6\rho^6}\right)\left(\log\frac{12m\log_2\frac{2^{37}m^{7/2}}{\delta^6\rho^6}}{\delta}\right) \\ &\le 10^3 m^{1.5}\log^2\left(\frac{100m}{\delta\rho}\right). \end{aligned}$$

For the remainder of the proof, we assume that $\varepsilon \le \frac{\delta^6\rho^6\Delta}{2^{39}m^4}$. This bound was taken to ensure that $\varepsilon \le \min\{\frac{\lambda\Delta}{6\sqrt{m}}, \frac{r\Delta}{4}\}$, and our choice of λ was taken to ensure that $3600m^{7/6}\lambda^{1/3}\rho^{-2}\delta^{-1} \le \delta$.

First, since $\varepsilon \le \frac{\lambda\Delta}{6\sqrt{m}}$, the accuracy guarantee of Lemma EC.8 holds for all queries to Mem with probability at least $1-\alpha/3$, by a union bound. Writing $K = K_y^{-2\varepsilon}$, we have by Lemma EC.9 that $B(K_y, -2\varepsilon/\Delta) \subseteq K$. Since K_y contains a ball of radius ρ and $\varepsilon \le \rho\Delta/4$, K must contain a ball of radius $\rho - \rho/2 = \rho/2$. Moreover, by Lemma EC.8, the simulated oracle Mem, defined at Step 3, is a λ-approximate, conservative membership oracle for the set K; that is, Mem only returns True for a query $\mathbf{z}$ if $\mathbf{z} \in K$ and only returns False if $\mathbf{z} \notin B(K, -\lambda)$. Of course, the lemma's guarantee is slightly stronger, but this relaxation suffices.

Our result now nearly follows from the proof of Theorem 14 in Lee et al. (2018), which provides an optimization guarantee for the Algorithm 1 which we apply at Step 4. We will slightly adapt their analysis to obtain explicit constants and to incorporate the conservative nature of our simulated membership oracle. First, we observe that the query cap of $Q = 2m\lceil \log_2(2/\lambda)\rceil + 1$ never goes into effect. Indeed, their Algorithm 1 calls Mem once at the beginning, and then at most $\lceil \log_2(2/\lambda)\rceil$ times within each of $2m$ binary searches performed by their Algorithm 2 subroutine.

Next, we verify our first guarantee, when SIMULATEDSEP fails to find a separating hyperplane and returns "$\perp$." Given a query $\mathbf{x} \in \mathbb{R}^m$, we only return "$\perp$" when their Algorithm 1 fails to return a half-space, which only occurs when MEM returns TRUE; in this case, we must have $\mathbf{x} \in K$, as desired. If $\mathbf{x} \notin \mathcal{X}$, then the returned half-space separates $\mathbf{x}$ from $\mathcal{X}$ (and thus K) with no error.

Otherwise, we must have $\mathbf{x} \in \mathcal{X} \setminus B(K, -\lambda) \subseteq B(0, \sqrt{2}) \setminus B(K, -\lambda)$. In this case, we perform the same error analysis appearing in their proof of Theorem 14, but explicitly state constants appearing due to an implicit use of Markov's inequality. With our notation, they start by proving (within their Lemma 13) that the returned vector $\tilde{\mathbf{g}} \in \mathbb{R}^m$ satisfies

$$\frac{600}{\delta} m^{7/6} \lambda^{1/3} \rho^{-1} \geq \tilde{\mathbf{g}}^\top (\mathbf{z} - \mathbf{x})$$

for all $\mathbf{z} \in K$, with probability at least $1 - \delta/3$. Then, they lower bound

$$\tilde{\mathbf{g}}^\top \mathbf{x} \geq \|\mathbf{x}\|_2 - \zeta \|\mathbf{x}\|_\infty - 64 m^{7/6} \lambda^{1/3} \rho^{-1},$$

where ζ is a non-negative random variable with $\mathrm{E}[\zeta] < 24 m^{7/6} \lambda^{1/3} \rho^{-1}$. Applying Markov's inequality, we find that

$$\tilde{\mathbf{g}}^\top \mathbf{x} \geq \frac{\rho}{2} - \lambda - \left(\frac{72}{\delta} - 64 \right) m^{7/6} \lambda^{1/3} \rho^{-1} \geq \frac{\rho}{2} - \lambda^{1/3} \left(\frac{72 m^{7/6}}{\delta \rho} \right)$$

with probability at least $1 - \delta/3$. Since $\lambda \leq \frac{\rho^6 \delta^3}{2^{25} m^{7/2}}$, this implies that $\tilde{\mathbf{g}}^\top \mathbf{x} \geq \rho/4$, in which case $\|\tilde{\mathbf{g}}\| \geq \frac{\rho}{4\|\mathbf{x}\|} > \rho/6$. Accounting for a normalization factor of $1/\|\tilde{\mathbf{g}}\|_2$ and taking a union bound, we obtain an error bound of $3600 m^{7/6} \lambda^{1/3} \rho^{-2} \delta^{-1} \leq \delta$ with cumulative error probability at most δ. $\square$

Next, we compare the maximum value of $u(\mathbf{x}, y)$ over $\mathbf{x} \in K_y$ to that over $\mathbf{x} \in K_y^{-2\varepsilon}$.

LEMMA EC.14. *Let $\varepsilon \geq 0$, and fix any $y \in \mathcal{Y}$ such that K_y contains a ℓ_2-ball of radius $\rho > 0$. We then have* $\max_{\mathbf{x} \in K_y^{-2\varepsilon}} u(\mathbf{x}, y) \geq \max_{\mathbf{x} \in K_y} u(\mathbf{x}, y) - \frac{2\sqrt{2m}\varepsilon}{\rho \Delta}$.

Proof. By Lemma EC.9, we have $B(K_y, -2\varepsilon/\Delta) \subseteq K_y^{-2\varepsilon}$. Moreover, since the principal utilities lie in $[0, 1]$, $u(\cdot, y)$ is $\sqrt{m}$-Lipschitz under the ℓ_2-norm. Fixing $\lambda = 2\varepsilon/\Delta$ and $K = K_y$, it suffices to show that, for each $\mathbf{x} \in K$, there exists $\mathbf{x}' \in B(K, -\lambda)$ with $\|\mathbf{x} - \mathbf{x}'\|_2 \leq \sqrt{2}\lambda/\rho$ (i.e., a Hausdorff

distance bound between the setes $B(K,-\lambda)$ and K). If $\lambda > \rho$, then this is trivially true by the diameter of K. Otherwise, fix any $\mathbf{x}_0 \in B(K,-\rho)$. Since K is convex and $B(\mathbf{x}_0,\rho) \subseteq K$, we have $\operatorname{conv}(\{\mathbf{x}\} \cup B(\mathbf{x}_0,\rho)) \subseteq K$. This convex hull contains the ball $B(\frac{\lambda}{\rho}\mathbf{x}_0 + (1-\frac{\lambda}{\rho})\mathbf{x}, \lambda)$, since $\lambda \le \rho$ and

$$B\left(\frac{\lambda}{\rho}\mathbf{x}_0 + \left(1-\frac{\lambda}{\rho}\right)\mathbf{x}, \lambda\right) = \left(1-\frac{\lambda}{\rho}\right)\mathbf{x} + \frac{\lambda}{\rho}B\left(\mathbf{x}_0,\rho\right).$$

Thus, we have $\mathbf{x}' = \frac{\lambda}{\rho}\mathbf{x}_0 + (1-\frac{\lambda}{\rho})\mathbf{x} \in B(K,-\lambda)$, with

$$\|\mathbf{x}-\mathbf{x}'\|_2 \le \frac{\lambda}{\rho}\|\mathbf{x}_0-\mathbf{x}\| \le \frac{\lambda\sqrt{2}}{r},$$

as desired. □

Finally, we are equipped to analyze MembershipOpt.

Proof of Lemma EC.10. We first address query complexity. Recall our parameter settings of $t_f = \lceil 3m \log \frac{16}{\delta\rho} \rceil$ and $\alpha = \min\{\frac{\delta}{t_f+1}, \frac{\delta\rho}{16\sqrt{2m}}\}$. By Lemma EC.13, we query Oracle at most

$$\begin{aligned}(t_f+1)\cdot 10^3 m^{1.5}\log^2\left(\frac{100m}{\alpha\rho}\right) &\le 6\cdot 10^3 m^{2.5}\log^2\left(\frac{100\cdot 16\sqrt{2}\cdot 6m^2\log\frac{16}{\delta\rho}}{\delta\rho^2}\right)\log\frac{16}{\delta\rho}\\ &\le 6\cdot 10^3 m^{2.5}\log^2\left(\frac{100\cdot 16\sqrt{2}\cdot 6m^2\log\frac{16}{\delta\rho}}{\delta\rho^2}\right)\log\frac{16}{\delta\rho}\\ &\le 10^5 m^{2.5}\log^3\left(\frac{120m}{\delta\rho}\right)\end{aligned}$$

times, as claimed.

From now on, we suppose that $\varepsilon \le \left(\frac{\delta\rho}{140m}\right)^{13}\Delta \le \frac{\alpha^6\rho^6\Delta}{2^{39}m^4}$ and that K_y contains a ball of radius ρ. Condition on the Lemma EC.13 guarantee for SimulatedSep holding for all of its t_f+1 calls. Since ε obeys the lemma's bound, this event has probability at least $1-\delta$ by a union bound. Under this event, SimulatedSep only returns "$\perp$" if a query $\mathbf{x}$ lies in $K = K_y^{-2\varepsilon}$, and only returns a normal vector $\mathbf{w} \in \mathbb{S}^{m-1}$ if $(\mathbf{x}-\mathbf{z})^\top\mathbf{w} \le \alpha$ for all $\mathbf{z} \in K$.

As in the proof of Lemma EC.13, we note that K must contain a ball of radius $\rho/2$ by Lemma EC.9. Write $V_m = \pi^{m/2}\Gamma(m/2+1)^{-1}$ for the volume of the unit ball in $\mathbb{R}^m$ and set $\tau = \frac{\delta}{4\sqrt{2m}}$. Now, we have $\operatorname{vol}_m(S_0) = 2^{m/2}V_m$, and, by Grünbaum's inequality (Lemma EC.2), $\operatorname{vol}_m(S_{t+1}) \le (1-1/e)\operatorname{vol}_m(S_t)$.

Fixing $\mathbf{x}^\star_\varepsilon \in \arg\max_{x\in K} u(x,y)$, we define $C = [(1-\tau)\mathbf{x}^\star_\varepsilon + \tau K] \cap B(K,-\alpha)$ and bound its volume from below by

$$\mathrm{vol}_m(C) \geq \mathrm{vol}_m\big(B(\tau K, -\alpha)\big) \geq (\tau\rho/2-\alpha)^m V_m \geq (\tau\rho/4)^m V_m,$$

using that $\alpha \leq \tau\rho/4$. Thus, after $t_f \geq m\log_2\big(\frac{\tau\rho}{4\sqrt{2}}\big)/\log_2\big(1-1/e\big)$ rounds, we cannot have $C \subseteq S_t$, and so there exists $r \in \{0,\dots,t_f\}$ for which we have some $\mathbf{x} \in C \cap S_r \setminus S_{r+1}$. This x may not be removed at Step 4, because $\mathbf{x} \in C \subseteq B(K,-\alpha)$. Thus, $\mathbf{x}$ must be removed at Step 6 with $u(\mathbf{x}_r, y) > u(\mathbf{x}, y)$, and $\mathbf{x}_r$ must be added to the set of candidate maximizers A.

In particular $A \neq \emptyset$, and so we do not return "$\perp$." Instead, for the returned strategy $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in A} u(\mathbf{x}, y)$, we have

$$\begin{aligned}
u(\hat{\mathbf{x}}, y) \geq u(\mathbf{x}_r, y) &> u(\mathbf{x}, y) \\
&\geq u(\mathbf{x}^\star_\varepsilon, y) - \|\mathbf{x}_r - \mathbf{x}^\star_\varepsilon\|_2\sqrt{m} \\
&\geq u(\mathbf{x}^\star_\varepsilon, y) - \mathrm{diam}(\tau K)\sqrt{m} \\
&\geq u(\mathbf{x}^\star_\varepsilon, y) - 2\sqrt{2m}\tau \\
&= u(\mathbf{x}^\star_\varepsilon, y) - \delta/2 \\
&\geq \max_{\mathbf{x}\in K_y} u(\mathbf{x}, y) - \delta,
\end{aligned}$$

as desired. The second inequality uses that $u(\cdot, y)$ is $\sqrt{m}$-Lipschitz under the ℓ_2-norm, and the final inequality uses Lemma EC.14. □

EC.7. Supplementary Material for Strategic Classification (Section 5.3)

First, we derive two lemmas from the regularity assumptions.

LEMMA EC.15. *At any strategic round t, the agent's payoff is bounded from above by $R^2(1+1/\alpha)$ and the best response $\mathsf{br}_t(\boldsymbol{\theta}_t)$ is unique with payoff at least $-R^2$.*

Proof. First, by the α-strong convexity assumption on f_t, v_{a_t} is α-strongly concave in $\boldsymbol{\theta}$, and so the best response $\mathsf{br}_t(\boldsymbol{\theta}_t)$ is unique. In the proof of Theorem 2 on page 8 of Dong et al. (2018),

the authors show that $v_{a_t}(\boldsymbol{\theta}_t, \mathsf{br}_t(\boldsymbol{\theta}_t)) \le \boldsymbol{\theta}_t^\top \mathsf{br}_t(\boldsymbol{\theta}_t) = \mathbf{x}_t^\top \boldsymbol{\theta}_t + 2f_t^*(\boldsymbol{\theta}_t)$, where f_t^* is the convex conjugate of f_t. In the proof of Claim 2 on page 20, they further show that $f_t^*(\boldsymbol{\theta}) = \sup_{\mathbf{v}\in\mathbb{S}^{d-1}} \frac{(\boldsymbol{\theta}^\top \mathbf{v})^2}{4f_t(\mathbf{v})}$. The numerator of this objective is bounded from above by R^2, while the denominator is bounded from below by 2α, since $\|\mathbf{v}\|_2 = 1$ and f_t is α-strongly convex with minimum of 0 the origin (due to homogeneity). Thus, $f_t^*(\boldsymbol{\theta}_t) \le \frac{R^2}{2\alpha}$, and so $v_{a_t}(\boldsymbol{\theta}_t, \hat{\mathbf{x}}_t) \le v_{a_t}(\boldsymbol{\theta}_t, \mathsf{br}_t(\boldsymbol{\theta}_t)) \le R^2(1 + \frac{1}{2\alpha})$. Finally, by playing $\hat{\mathbf{x}} = \mathbf{x}$, the agent obtains payoff $\boldsymbol{\theta}_t^\top \mathbf{x}_t \ge -R^2$. □

LEMMA EC.16. *Each map $\boldsymbol{\theta} \mapsto \ell(\boldsymbol{\theta}, \mathsf{br}_t(\boldsymbol{\theta}), y_t)$ is convex, $(R + 2R/\alpha)$-Lipschitz, and bounded in absolute value by $1 + R^2 + R^2/\alpha$. Moreover, each map $\hat{\mathbf{x}} \mapsto \ell(\boldsymbol{\theta}, \hat{\mathbf{x}}, y_t)$ is R-Lipschitz.*

Proof. Convexity of $\ell_t(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}, \mathsf{br}_t(\boldsymbol{\theta}), y_t)$ is implied by Theorem 2 of Dong et al. (2018). For Lipschitzness of ℓ_t, we turn to the proof of Theorem 7 on page 17 of Dong et al. (2018). The discussion there implies that $\ell_t(\boldsymbol{\theta})$ is Lipschitz with constant $\|\mathbf{x}_t\|_2 \le R$ plus twice the Lipschitz constant of f_t^*. To compute this, we bound

$$\left| \sup_{\mathbf{v}\in\mathbb{S}^{d-1}} \frac{(\boldsymbol{\theta}^\top \mathbf{v})^2}{4f_t(\mathbf{v})} - \sup_{\mathbf{v}\in\mathbb{S}^{d-1}} \frac{(\boldsymbol{\theta}'^\top \mathbf{v})^2}{4f_t(\mathbf{v})} \right| \le \sup_{\mathbf{v}\in\mathbb{S}^{d-1}} \left| \frac{(\boldsymbol{\theta}^\top \mathbf{v})^2}{4f_t(\mathbf{v})} - \frac{(\boldsymbol{\theta}'^\top \mathbf{v})^2}{4f_t(\mathbf{v})} \right| \le \frac{2R\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2}{2\alpha},$$

using the same lower bound on $f_t(\mathbf{v})$ as in the proof of Lemma EC.15. Combining gives a Lipschitz constant of $R + 2R/\alpha$ for ℓ_t. Finally, discussion on page 18 of Dong et al. (2018) implies that $|\ell_t(\boldsymbol{\theta})| \le 1 + |\boldsymbol{\theta}^\top \mathbf{x}_t\rangle| + 2f_t^*(\boldsymbol{\theta})$, which we bound by $1 + R^2(1 + \frac{1}{2\alpha})$ as in the proof of Lemma EC.15. Finally, the same discussion on page 17 implies that the map $\hat{\mathbf{x}} \mapsto \ell(\boldsymbol{\theta}, \hat{\mathbf{x}}, y_t)$ has Lipschitz constant bounded by that of the map $\hat{\mathbf{x}} \mapsto \boldsymbol{\theta}^\top \hat{\mathbf{x}}$, which is $\|\boldsymbol{\theta}\|_2 \le R$. □

Bandit convex optimization. By Lemma EC.16, each loss function $\ell_t(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}, \mathsf{br}_t(\boldsymbol{\theta}), y_t)$ in the myopic setting is convex, Lipschitz, and bounded. When $y_t = 1$, feedback $\hat{\mathbf{x}}_t$ is sufficient to determine the gradient $\nabla \ell_t(\boldsymbol{\theta}_t)$, suggesting regret minimization via online convex optimization (OCO). Although this is not the case when $y_t = -1$, since the agent's manipulation costs encoded by d_t are hidden, the regime of OCO with one-point function evaluations, or *bandit convex optimization*, is well-studied. Dong et al. (2018) employ the classic "gradient descent without a gradient" procedure GDwoG of Flaxman et al. (2005) (Algorithm 14) to obtain regret $O(\sqrt{d}T^{3/4})$ against myopic agents, computing unbiased gradient estimates from stochastic function evaluations.

Algorithm 14: Online **G**radient **D**escent **w**ith**o**ut a **g**radient (Flaxman et al. 2005)

input: domain $S \subset \mathbb{R}^d$ with $\mathbb{B} \subseteq S \subseteq R\mathbb{B}$, L-Lipshitz convex fn.s $c_1, \dots, c_T : S \to [-C, C]$

$\delta \leftarrow \sqrt{\frac{RdC}{3(L+C)}} T^{-1/4}$, $\eta \leftarrow \frac{R}{C\sqrt{T}}$, $v_1 \leftarrow (0, \dots, 0) \in \mathbb{R}^d$

for *round* $t = 1, \dots, T$ **do**

  Sample unit vector $\mathbf{s}_t \in \mathbb{S}^{d-1}$ uniformly at random

  $\mathbf{u}_t \leftarrow \mathbf{v}_t + \delta \mathbf{s}_t$

  $\mathbf{v}_{t+1} \leftarrow \Pi_{(1-\delta)S}(\mathbf{v}_t - \eta c_t(\mathbf{u}_t)\mathbf{s}_t)$ `// $\Pi_K$ is the Euclidean projection onto $K$`

LEMMA EC.17 **(Flaxman et al. (2005), Theorem 2)**. *If functions $c_1, \dots, c_T : S \to [-C, C]$ are convex and L-Lipschitz, and $S \subseteq \mathbb{R}^d$ is convex with $\mathbb{B} \subseteq S \subseteq R\mathbb{B}$, then the queries $\mathbf{u}_1, \dots, \mathbf{u}_T$ of* GDwoG *satisfy* $\mathrm{E}\left[\sum_{t=1}^T c_t(\mathbf{u}_t)\right] - \min_{\mathbf{u} \in S} \sum_{t=1}^T c_t(\mathbf{u}) \le 6T^{3/4}\sqrt{RdC(L+C)} + 5C(Rd)^2$.

Our extension to non-myopic agents. We now extend this approach to robust learning, due to an intrinsic robustness of GDwoG. First, we prove the relevant error bound.

LEMMA EC.18. *If the agent chooses $\hat{\mathbf{x}}_t \in \mathsf{BR}^\varepsilon(\boldsymbol{\theta}_t)$, then $|\ell(\boldsymbol{\theta}_t, \hat{\mathbf{x}}_t, y_t) - \ell(\boldsymbol{\theta}_t, \mathsf{br}_t(\boldsymbol{\theta}_t), y_t)| \le R\sqrt{2\varepsilon/\alpha}$.*

Proof. By the strong convexity assumption, we have $\|\hat{\mathbf{x}}_t - \mathsf{br}_t(\boldsymbol{\theta}_t)\|_2 \le \sqrt{2\varepsilon/\alpha}$, and so the result follows from the Lipschitz bound R from Lemma EC.16. □

We next provide an appropriate robust learning guarantee for GDwoG.

LEMMA EC.19. *Under the setting of Lemma EC.17,* GDwoG *achieves the same regret up to an additive factor of $\lambda RdT/\delta$ if each $c_t(\mathbf{u}_t)$ is substituted with an adversarial perturbation $\tilde{c}_t(\mathbf{u}_t) \in [c_t(\mathbf{u}_t) \pm \lambda]$.*

Proof sketch. The proof of Lemma EC.17 for the unperturbed setting goes through the smoothed functions $\bar{c}_t(\mathbf{u}) = \mathrm{E}[c_t(\mathbf{u} + \delta \mathbf{s}_t)]$. The key observations are that each $\bar{c}_t$ is convex and Lipschitz with $|\bar{c}_t(\mathbf{u}) - c_t(\mathbf{u})| \le L\delta$ and, crucially, $\mathrm{E}[\frac{d}{\delta} c_t(\mathbf{u}_t)\mathbf{s}_t] = \nabla \bar{c}_t(\mathbf{v}_t)$. With adversarial perturbations, we have

$$\left\|\mathrm{E}\left[\tfrac{d}{\delta}\tilde{c}_t(\mathbf{u}_t)\mathbf{s}_t\right] - \nabla \bar{c}_t(\mathbf{v}_t)\right\|_2 = \left\|\mathrm{E}\left[\tfrac{d}{\delta}\left(\tilde{c}_t(\mathbf{u}_t) - c_t(\mathbf{u}_t)\right)\mathbf{s}_t\right]\right\|_2 \le \frac{d\lambda}{\delta}.$$

At this point, the proof of Theorem 2 in Flaxman et al. (2005) for the unperturbed setting nearly applies, so long as their Lemma 2 is adjusted to our setting where gradient estimates $\mathbf{g}_t$ have bias $b = \frac{d\lambda}{\delta}$ rather than $b = 0$. Switching to their notation for the lemma, if we have $\mathrm{E}[\mathbf{g}_t | \mathbf{x}_t] = \nabla c_t(\mathbf{x}_t) + \boldsymbol{\xi}_t$ with $\|\boldsymbol{\xi}_t\|_2 \le b$, then the final chain of inequalities in their proof of Lemma 2 still holds, up to an

added term of $\sum_{t=1}^{n} \boldsymbol{\xi}_t^\top (\mathbf{x}_t - \mathbf{x}_\star) \le nbR$. Switching back to our notation and substituting our value for b, this gives the claimed regret overhead of $O(\frac{\lambda RTd}{\delta})$. $\square$

Finally, we provide our combined algorithm and prove its non-myopic regret bound.

Algorithm 15: Cycled gradient descent without a gradient (CGDwoG)

$\varepsilon \leftarrow \alpha(R^4 dT^{2.5})^{-1}$, $D \leftarrow \lceil T_\gamma \log(R^2(1+1/\alpha)T_\gamma/\varepsilon)\rceil$
Initialize copies $\mathcal{A}_1, \dots, \mathcal{A}_D$ of Algorithm 14 w/ $S = \Theta$, $C = 1 + R^2 + R^2/\alpha$, and $L = R + 2R/\alpha$
for *round* $t = 1, \dots, T$ **do**
    Write $t = D(k-1) + (r-1)$ for $k, r \in \mathbb{Z}_{>0}$
    Simulate query $\mathbf{u}_k$ and perturbed feedback $\tilde{c}_k(\mathbf{u}_k)$ for $\mathcal{A}_r$ using $\boldsymbol{\theta}_t$ and $\ell(\boldsymbol{\theta}_t, \hat{\mathbf{x}}_t, y_t)$

Proof of Theorem 7. Since Stackelberg regret is subadditive over disjoint sequences of rounds, we obtain regret $DR^{\varepsilon}_{\mathcal{A}_1}(\lceil T/D \rceil)$ against ε-approximate best-responding agents. Combining Lemmas EC.16, EC.18 and EC.19 and substituting our choices of constants, we bound the regret of any single copy by

$$\begin{aligned}
R^{\varepsilon}_{\mathcal{A}_1}(T) &\le \mathrm{E}\left[\sum_{t=1}^{T} \ell(\boldsymbol{\theta}_t, \hat{\mathbf{x}}_t, y_t) - \min_{\boldsymbol{\theta}\in\Theta} \sum_{t=1}^{T} \ell(\boldsymbol{\theta}, \mathsf{br}_t(\boldsymbol{\theta}), y_t)\right] \\
&\le \mathrm{E}\left[\sum_{t=1}^{T} \ell(\boldsymbol{\theta}_t, \mathsf{br}_t(\mathbf{x}_t), y_t) - \min_{\boldsymbol{\theta}\in\Theta} \sum_{t=1}^{T} \ell(\boldsymbol{\theta}, \mathsf{br}_t(\boldsymbol{\theta}), y_t)\right] + TR\sqrt{2\varepsilon/\alpha} \\
&\le 6T^{3/4}\sqrt{RdC(L+C)} + 5C(Rd)^2 + TR\sqrt{\frac{2\varepsilon}{\alpha}}(Rd/\delta + 1) \\
&= O\left(R^{5/2}\hat{\alpha}^{-1}\sqrt{d}T^{3/4} + R^4\hat{\alpha}^{-2}d^2\right),
\end{aligned}$$

where $\hat{\alpha} = \min\{\alpha, 1\}$. By Proposition 1 and Lemma EC.15, our feedback delay induces ε-approximate best responses, so we obtain a final regret bound of

$$O\left(R^{5/2}\hat{\alpha}^{-1}T_\gamma^{1/4}\sqrt{d}T^{3/4}\log^{1/4}(TRd/\alpha) + R^4\hat{\alpha}^{-2}d^2\right). \quad \square$$

Remark EC.1. As noted in (Dong et al. 2018), GDwoG can be replaced by more modern methods with regret $\widetilde{O}(\mathrm{poly}(d)\sqrt{T})$ (Bubeck et al. 2021), at the cost of substantial complexity and worse scaling with d. While beyond the scope of this paper, robustifying such algorithms would imply improved non-myopic regret bounds.