No-Regret Learning in Bayesian Stackelberg Games with Unknown Follower Types

author names withheld

Under Review for NExT-Game 2026

Abstract

We study online learning in *Bayesian Stackelberg games*, where a leader repeatedly interacts with a follower whose *unknown private type* is independently drawn at each round from an *unknown* probability distribution. The goal is to design algorithms that minimize the leader's *regret* with respect to always playing an optimal commitment computed with knowledge of the game. We consider, for the first time to the best of our knowledge, the most realistic case in which the leader does *not* know anything about the follower's types, *i.e.*, the possible follower payoffs. This raises considerable additional challenges compared to the commonly studied case in which the payoffs of follower types are known. First, we prove a strong negative result: no-regret is unattainable under *action feedback*, *i.e.*, when the leader only observes the follower's best response at the end of each round. Thus, we focus on the easier *type feedback* model, where the follower's type is also revealed. In such a setting, we propose a no-regret algorithm that achieves a regret of $\widetilde{O}(\sqrt{T})$, when ignoring the dependence on other parameters.

1. Introduction

Stackelberg games (SGs) [16] are foundational economic models that capture asymmetric strategic interactions among rational agents. In an SG, a leader publicly commits to a strategy beforehand, and a follower then best responds to this commitment. This simple form of interaction underlies several more complex economic models, such as Bayesian persuasion [9], contracts [8], auctions [12], and security games [15].

The problem of learning an optimal strategy to commit to in SGs has recently received growing attention [3, 6, 7, 10, 13]. In particular, in this paper we study online learning in Bayesian SGs (BSGs), where a leader repeatedly interacts with a follower over T rounds, with the follower having a (different) unknown private type at each round. The follower's type determines the follower's payoffs and, consequently, the best response they play. At each round t, the leader commits to a strategy prescribed by a learning algorithm, and the follower then best responds to it. The goal of the leader is to minimize their (Stackelberg) regret, which measures how much utility they lose over the T rounds compared to always committing to an optimal strategy in hindsight. Ideally, one would like learning algorithms that are no-regret, meaning that their regret grows sublinearly in the number of rounds T. Online learning in BSGs has already been investigated in the literature, both when the follower's types are selected adversarially [4, 5] and when they are independently drawn at each round according to some *unknown* probability distribution [14]. However, all these works rely on the very stringent assumption that *the leader knows the payoffs associated with every possible follower type*. This assumption is rather unreasonable in practice. For instance, in security

games it would amount to assuming that the defender knows the target preferences of every possible malicious attacker profile. We consider, for the first time to the best of our knowledge, online learning in BSGs where the leader does not know anything about the game, including the payoffs of possible follower types and the probability distribution according to which such types are drawn at each round. Our main result is the first no-regret learning algorithm for BSGs that does not require any assumptions on the leader's knowledge.

Original Contributions. We start by proving a strong negative result for the action feedback model, where, at the end of each round, the leader only observes the best-response action played by the follower. Specifically, we show that there exist BSGs in which any learning algorithm must incur regret that grows exponentially in the number of bits needed to represent the follower's payoffs. Thus, even in instances where only a few bits are sufficient to represent the follower's payoffs, the regret can be prohibitively large.

In the remainder of the paper, we focus on the easier type feedback model, where, at the end of each round, the leader not only observes the follower's best response but also their type. In such a setting, we provide a no-regret learning algorithm for the leader that does not require any knowledge of the follower's payoffs in order to operate.

Our no-regret learning algorithm works by splitting the time horizon into epochs. At each epoch h, the algorithm learns the follower's best-response regions in order to identify the leader's commitments that are at most ϵ_h-suboptimal. Then, in the following epoch $h+1$, the algorithm restricts the leader's decision space to such commitments and halves the suboptimality level ϵ_h. This allows the algorithm to use commitments that are not overly suboptimal in the next epoch and thus keep the regret under control.

At each epoch h, our algorithm performs three main steps.

1. First, the algorithm uses $\mathcal{O}(1/\epsilon_h^2)$ rounds of interaction with the follower to build a suitable estimator of the probability distribution over types μ. At the same time, it identifies a subset of types whose probability is at least ϵ_h.

2. The second main step is to learn the polytopes defining the best-response regions for the follower types identified in the previous step.

3. The third and final step of the epoch is to use the information on best-response regions collected so far to compute an ϵ_h-suboptimal commitment to be used in the subsequent epoch.

Our no-regret algorithm achieves regret of order $\tilde{O}(\sqrt{T})$ in T and depends polynomially on the size of the BSG instance when the number of leader actions m is fixed.

2. Preliminaries

A Bayesian Stackelberg game (BSG) is characterized by a finite set $\mathcal{A}_L := \{a_i\}_{i=1}^m$ of m leader's actions and a finite set of different follower's types Θ, with $K := |\Theta|$. Without loss of generality, we assume that all follower's types share the same action set of size n, denoted by $\mathcal{A}_F := \{a_j\}_{j=1}^n$. The leader's payoffs are encoded by the utility function $u^L : \mathcal{A}_L \times \mathcal{A}_F \to [0,1]$, while each follower's type $\theta \in \Theta$ has a payoff function $u_\theta^F : \mathcal{A}_L \times \mathcal{A}_F \to [0,1]$. We assume that the entries of these utilities function are encoded by L bits.

In a BSG, the leader commits in advance to a mixed strategy, which is a probability distribution $x \in \Delta(\mathcal{A}_L)$ over leader's actions, where each $x_i \in [0,1]$ denotes the probability of selecting action $a_i \in \mathcal{A}_L$. Then, a follower's type $\theta \in \Theta$ is drawn according to a probability distribution $\mu \in \Delta(\Theta)$, i.e., $\theta \sim \mu$. We assume that the follower, after observing the leader's commitment $x \in \Delta(\mathcal{A}_L)$, plays an action deterministically. Specifically, the follower plays a best response, which is an action maximizing their expected utility given the commitment $x \in \Delta(\mathcal{A}_L)$. For every follower's type $\theta \in \Theta$, the set of follower's best responses is

$$A^F_\theta(x) := \arg\max_{a_j \in \mathcal{A}_F} \sum_{i \in [m]} x_i u^F_\theta(a_i, a_j).$$

As is customary in the literature, we assume that, when multiple best responses are available, the follower breaks ties in favor of the leader by choosing an action that maximizes the leader's expected utility. Formally, a follower of type $\theta \in \Theta$ selects a best-response action $a^\star_\theta(x) \in A^F_\theta(x)$ such that

$$a^\star_\theta(x) \in \arg\max_{a_j \in A^F_\theta(x)} \sum_{i \in [m]} x_i u^L(a_i, a_j).$$

With a slight abuse of notation, given a commitment $x \in \mathbb{R}^m$ and a follower's action $a_j \in \mathcal{A}_F$, we denote with $u^L(x, a_j) := \sum_{i \in [m]} x_i u^L(a_i, a_j)$ the expected leader utility and with $u^F_\theta(x, a_j) := \sum_{i \in [m]} x_i u^F_\theta(a_i, a_j)$ the expected utility of a follower of type $\theta \in \Theta$. The leader's goal is to find an optimal commitment, which is a mixed strategy $x \in \Delta(\mathcal{A}_L)$ that maximizes the leader's expected utility, assuming that the follower always responds by selecting a best-response action. We denote this value as $\text{OPT} := \max_{x \in \Delta(\mathcal{A}_L)} u^L(x)$, where $u^L(x) := \sum_{\theta \in \Theta} \mu_\theta u^L(x, a^\star_\theta(x))$.

Follower's Best-Response Regions. For every follower's type $\theta \in \Theta$ and action $a_j \in \mathcal{A}_F$, we define the best-response region $\mathcal{P}_\theta(a_j) \subseteq \Delta(\mathcal{A}_L)$ as the set of leader's commitments under which the utility of type θ is maximized by playing action a_j. Formally:

$$\mathcal{P}_\theta(a_j) := \Delta(\mathcal{A}_L) \cap \bigcap_{a_k \in \mathcal{A}_F \setminus \{a_j\}} \mathcal{H}^\theta_{jk},$$

where $\mathcal{H}^\theta_{jk} := \{x \in \mathbb{R}^m \mid u^F_\theta(x, a_j) \geq u^F_\theta(x, a_k)\}$ is the separating hyperplane between actions a_i and a_j for type θ. We observe that the best-response region $\mathcal{P}_\theta(a_j)$ is a polytope. Given an action profile $\boldsymbol{a} = (\boldsymbol{a}_\theta)_{\theta \in \Theta'}$ for some subset of types $\Theta' \subseteq \Theta$, we will also denote with $\mathcal{P}(\boldsymbol{a}) := \bigcap_{\theta \in \Theta'} \mathcal{P}_\theta(\boldsymbol{a}_\theta)$ the best-respones region of $\boldsymbol{a}$.

Online Learning in Bayesian Stackelberg Games. We study settings in which the leader repeatedly interacts with the follower over multiple rounds. We assume that the leader has no knowledge of either the distribution over types μ or the utility function u^F_θ of each type $\theta \in \Theta$. At each round $t \in [T]$, the leader-follower interaction unfolds as follows:

1. The leader commits to a mixed strategy $x_t \in \Delta(\mathcal{A}_L)$.
2. A follower's type $\theta_t \in \Theta$ is sampled according to μ, and the follower plays action $a^\star_{\theta_t}(x_t)$.
3. Under type feedback, the leader observes both $a^\star_{\theta_t}(x_t)$ and θ_t, while under action feedback, the leader only observes the best response $a^\star_{\theta_t}(x_t)$.

The performance of an algorithm is evaluated in terms of the cumulative regret

$$R_T := T \cdot \mathrm{OPT} - \mathbb{E}\left[\sum_{t\in[T]} u^L(x_t, a^\star_{\theta_t}(x_t))\right].$$

3. A Negative Result With Action Feedback

We start with a negative result for the action-feedback setting. Specifically, we show that no algorithm can achieve regret polynomial in the bit-complexity of the follower's payoffs.

Theorem 1 *Let $L \in \mathbb{N}$ be the bit-complexity of the follower's payoffs. Under action feedback, for any learning algorithm, there exists a BSG instance with constant-sized leader/follower action sets and set of follower types, and $T = \Theta(2^{2L})$ rounds, such that*

$$R_T \geq \Omega(2^{2L}).$$

Intuitively, in order to prove the lower bound we consider $\Theta(2^{2L})$ instances. Each instance is characterized by a subregion in which all the follower's types play the unique action that provides positive utility to the leader. These regions are designed to be disjoint across instances, and the leader receives identical feedback whenever it selects a commitment outside them. As a result, in the worst case, the leader is required to enumerate an exponential number of regions before identifying the one that yields positive utility, which translates into an exponential lower bound on the regret.

Theorem 1 has strong practical implications. Indeed, even under commonly used 32-bit integer representations, the lower bound on the regret suffered by any algorithm is prohibitively large. This highlights that action feedback is insufficient to achieve meaningful regret guarantees. Let us also observe that, as a corollary of Theorem 1, for every time horizon T and any learning algorithm, there is an instance represented by only $\mathcal{O}(\log(T))$ bits where the algorithm incurs in $\Theta(T)$ regret.

4. No-Regret With Type Feedback

In this section, we present a no-regret algorithm that operates in the type-feedback setting. At a high level, the algorithm splits the T rounds into epochs indexed by $h \in \mathbb{N}$. In each epoch h, it executes three different procedures, namely `Find-Types`, `Find-Partition`, and `Prune`.

The algorithm initializes the leader's decision space by setting $\mathcal{X}_1 = \Delta(\mathcal{A}_L)$ and the parameter ϵ_h by setting $\epsilon_1 = 1/K$. The parameter ϵ_h plays a crucial role in balancing the length of the different epochs against the optimality of the commitments chosen by the leader. The decision space $\mathcal{X}_h$ shrinks as the algorithm advances to different epochs, concentrating around an optimal commitment. At every epoch the algorithm focuses on a subset of types $\widetilde{\Theta}_h \subseteq \Theta$. Denoting by $\mathcal{A}_F(\widetilde{\Theta}_h)$ the set of action profiles of these types, the algorithm knows the region $\mathcal{P}(\boldsymbol{a}_h) \cap \mathcal{X}_h$ for every $\boldsymbol{a}_h \in \mathcal{A}_F(\widetilde{\Theta}_h)$ at the end of epoch h.

We now give an overview of the three main operations performed by the Algorithm at epoch h. Details are referred to the appendix.

FIND-TYPES. This procedure takes as inputs the set $\mathcal{X}_h$ and the parameter ϵ_h.The goal of this procedure is to build an empirical estimator $\hat{\mu}_h$ of the distribution μ and to identify a subset of types $\widetilde{\Theta}_h \subseteq \Theta$ to be "explored" in the current iteration. This is done by querying any commitment $x \in \mathcal{X}_h$ for $\mathcal{O}(1/\epsilon_h^2)$ rounds. The empirical distribution of the types observed in these rounds is used to compute $\hat{\mu}_h$ and the set $\widetilde{\Theta}_h$. The algorithm provides the following guarantees.

Lemma 2 *Let $\epsilon_h, \delta_1 \in (0,1)$ and let $\mathcal{X}_h \subseteq \Delta(\mathcal{A}_L)$ be non-empty. Then, with probability at least $1-\delta_1$, the* `Find-Types` *procedure computes:*

(i) an estimator $\hat{\mu}_h \in \Delta(\Theta)$ such that $\|\mu - \hat{\mu}_h\|_\infty \leq \epsilon_h$;

(ii) a set of types $\widetilde{\Theta}_h \subseteq \Theta$ such that $\mu_\theta \geq \epsilon_h$ for every $\theta \in \widetilde{\Theta}_h$ and $\mu_\theta \leq 3\epsilon_h$ for every other type;

by using $\mathcal{O}\left(\frac{1}{\epsilon_h^2}\log\left(\frac{K}{\delta_1}\right)\right)$ rounds.

FIND-PARTITION. The goal of this procedure is to identify the mapping $\mathcal{Y}_h(\boldsymbol{a}_h) \coloneqq \mathcal{P}(\boldsymbol{a}_h) \cap \mathcal{X}_h$ for every action profile $\boldsymbol{a}_h \in \mathcal{A}_F(\widetilde{\Theta}_h)$. This is done by leveraging a procedure developed to learn the best-response regions for non-Bayesian Stackelberg games [2]. Intuitively, for each $\theta \in \widetilde{\Theta}_h$ we employ this procedure to learn the best response regions $\mathcal{P}_\theta(a_j) \cap \mathcal{X}_h$ of this type, then for each $\boldsymbol{a}_h$ we intersect a suitable set of these regions to obtain $\mathcal{Y}(\boldsymbol{a}_h)$. When executing the procedure by [2] on type θ, we play each commitment prescribed by this procedure until a follower of type θ best responds. Notice that we need to repeat every query only $\mathcal{O}(1/\epsilon_h)$ times to observe θ with high probability. We state here the main guarantees of this procedure, and we refer the reader to the appendix for additional details.

Lemma 3 (Informal version of Lemma 9) *If $\mu_\theta \geq \epsilon_h$ for every θ in the current set of relevant types $\widetilde{\Theta}_h$, then* `Find-Partition` *computes $\mathcal{Y}_h$ correctly with probability at least $1-\delta_2$, using at most*

$$\mathcal{O}\left(\textit{poly}(K, \frac{1}{\epsilon}, n, L)\log\left(\frac{1}{\delta_2}\right)\right)$$

rounds when the number of leader's actions m is constant.

Notice that this procedure uses a number of rounds exponential in m, which is unavoidable due to an existing lower bound [13].

PRUNE. The goal of this procedure is to exploit the estimator $\hat{\mu}_h$ and the mapping $\mathcal{Y}_h$ to refine the leader's decision space, thereby obtaining a new decision space $\mathcal{X}_{h+1}$ for the subsequent epoch $h+1$ and ensuring that all leader commitments selected at that epoch are at most $\mathcal{O}(\epsilon_h)$-suboptimal. This operation can be performed since we can estimate the utility $u^{\text{L}}(x)$ provided by the types in $\widetilde{\Theta}_h$ for every $x \in \mathcal{X}_h$, as we know the best responses of these types and an estimation $\widehat{\mu}_h$ of their probabilities. Other types can be instead ignored, as their contribution to the utility is negligible. Therefore, the procedure can compute a lower bound on the value of an optimal commitment and prune the subset of $\mathcal{X}_h$ that is guaranteed to yield low utility.

Lemma 4 (Informal version of Lemma 17) *Suppose that* `Find-Types` *and* `Find-Partition` *were executed successfully up to epoch h. Then* `Prune` *computes a search space $\mathcal{X}_{h+1}$ containing an optimal commitment. Furthermore, it holds $u^L(x) \geq \text{OPT} - 14K\epsilon_h$ for every $x \in \mathcal{X}_{h+1}$.*

5. Theoretical Guarantees

Before presenting the regret guarantees of the algorithm, let us observe that our algorithm executes at $H \leq \log_4(5T)$ epochs. This result can be proved exploiting the properties of geometric series, and the fact that the algorithm uses at least $(1/\epsilon_h)$ rounds at each epoch h and T rounds in total. We can now state the main guarantee.

Theorem 5 *For every $\delta \in (0, 1)$, the regret of the algorithm is*

$$R_T \leq \tilde{O}\left(K^2 \log\left(\frac{1}{\delta}\right)\sqrt{T} + \beta \log^2(T)\right), \tag{1}$$

with probability at least $1 - \delta$, where $\beta = \text{poly}(n, K, L, \log(1/\delta))$ when m is constant.

Proof We now provide a proof sketch of the above theorem. For the sake of presentation, in the following we omit the dependence on logarithmic terms and we assume m to be a constant. We observe that the number of rounds T_h required to execute a generic epoch h can be bounded as follows:

$$T_h \leq \mathcal{O}\left(\frac{1}{\epsilon_h^2} + \frac{1}{\epsilon_h}\text{poly}(n, K, L)\right),$$

according to the guarantees provided by Lemma 2 and Lemma 3, together with the fact that `Prune` does not require any leader-follower interactions. Therefore, the regret suffered by the algorithm can be upper bounded as follows:

$$R_T \leq \mathcal{O}\left(\sum_{h=1}^{H} T_h K \epsilon_{h-1}\right) \leq \mathcal{O}\left(K^2\sqrt{T} + \text{poly}(n, K, L)\log T\right).$$

The first inequality follows since each commitment selected at epoch h is $\mathcal{O}(K\epsilon_{h-1})$-optimal, while the second one can be proved observing that $\epsilon_h = 2^{1-h}/K$ and using the geometric series. ■

Thanks to the theorem above, the algorithm achieves a $\tilde{O}(\sqrt{T})$ regret upper bound when the number of leader's actions m is a fixed constant. Conversely, when m is not fixed, the theorem exhibits an exponential dependence on m in the regret suffered by the algorithm. However, even in the case of a single follower type, such an exponential dependence is unavoidable [13]. The algorithm can also be executed in polynomial time when m is constant. Indeed, the main bottleneck of our algorithm is the computation of the lower bound on the optimum in the `Prune` procedure, which is similar to the computation of the optimum in the offline problem – a problem that is already NP-Hard when m is not constant [11].

6. Conclusion

We studied online learning in Bayesian Stackelberg games in the realistic setting in which the leader does not know anything about the follower's types, namely the possible follower payoffs, nor the probability distribution according to which such types are drawn. We proved that no-regret is unattainable under action feedback. We then focused on the type-feedback setting and provided a no-regret algorithm that achieves regret $\tilde{O}(\sqrt{T})$ ignoring the dependence on the other parameters.

References

[1] Francesco Bacchiocchi, Matteo Bollini, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Online bayesian persuasion without a clue. *Advances in Neural Information Processing Systems*, 37:76404–76449, 2024.

[2] Francesco Bacchiocchi, Matteo Bollini, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. The sample complexity of stackelberg games. In *International Conference on Artificial Intelligence and Statistics*, pages 2053–2061. PMLR, 2025.

[3] Yu Bai, Chi Jin, Huan Wang, and Caiming Xiong. Sample-efficient learning of stackelberg equilibria in general-sum games. *Advances in Neural Information Processing Systems*, 34: 25799–25811, 2021.

[4] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015.

[5] Maria-Florina Balcan, Martino Bernasconi, Matteo Castiglioni, Andrea Celli, Keegan Harris, and Zhiwei Steven Wu. Nearly-optimal bandit learning in stackelberg games with side information. *arXiv preprint arXiv:2502.00204*, 2025.

[6] Maria-Florina Balcan, Kiriaki Fragkia, and Keegan Harris. Learning in structured stackelberg games. *arXiv preprint arXiv:2504.09006*, 2025.

[7] Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International conference on machine learning*, pages 3133–3144. PMLR, 2020.

[8] Sanford J Grossman and Oliver D Hart. An analysis of the principal-agent problem. In *Foundations of Insurance Economics: Readings in Economics and Finance*, pages 302–340. Springer, 1992.

[9] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[10] Niklas Lauffer, Mahsa Ghasemi, Abolfazl Hashemi, Yagiz Savas, and Ufuk Topcu. No-regret learning in dynamic stackelberg games. *IEEE Transactions on Automatic Control*, 69(3): 1418–1431, 2023.

[11] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory: Second International Symposium, SAGT 2009*, pages 250–262, 2009.

[12] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[13] Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. Learning optimal strategies to commit to. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 2149–2156, 2019.

[14] Gerson Personnat, Tao Lin, Safwan Hossain, and David C Parkes. Learning to play multi-follower bayesian stackelberg games. *arXiv preprint arXiv:2510.01387*, 2025.

[15] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

[16] Heinrich Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

Appendix A. Proof of Theorem 1

Theorem 1 *Let $L \in \mathbb{N}$ be the bit-complexity of the follower's payoffs. Under action feedback, for any learning algorithm, there exists a BSG instance with constant-sized leader/follower action sets and set of follower types, and $T = \Theta(2^{2L})$ rounds, such that*

$$R_T \geq \Omega(2^{2L}).$$

Proof Building the instances We consider a set of instances $\mathcal{I}$ in which the leader's action set is $\mathcal{A}_{\text{L}} \coloneqq \{a_1, a_2, a_3\}$ and the follower's action set is $\mathcal{A}_{\text{F}} \coloneqq \{a_1, a_2, a_3, a^\star\}$, and the leader's utility function is defined as follows.

$$u^{\text{L}}(a_i, a_j) = \begin{cases} 1 & a_j = a^\star, \\ 0 & a_j \neq a^\star. \end{cases}$$

The set of types is $\Theta \coloneqq \{\theta_1, \theta_2, \theta_3\}$, with uniform prior distribution μ. Each instance $\text{I} \in \mathcal{I}$ is parametrized by a subsimplex $\mathcal{S}^{\text{I}} \subseteq \Delta(\mathcal{A}_{\text{F}})$, corresponding to the set of optimal leader strategies in that specific instance.The subsimplex

$$\mathcal{S}^{\text{I}} \coloneqq \Delta(\mathcal{A}_{\text{L}}) \cap \{\mathcal{H}^{\text{I}}_j\}_{j \in [m]}$$

is defined as the intersection of $\Delta(\mathcal{A}_{\text{L}})$ with m half-spaces $\mathcal{H}^{\text{I}}_j$, for $j \in [m]$, defined as follows:

$$\mathcal{H}^{\text{I}}_j = \{x \in \mathbb{R}^m \mid \sum_{i \in [m]} w^{\text{I}}_{j,i} x_i \geq 0\}, \tag{2}$$

where $w^{\text{I}}_{j,i} \in [-1/2, 1/2]$ has bit complexity bounded by CB, for some $B \in \mathbb{N}$ and a universal constant $C > 0$. We refer the reader to the next paragraph for the formal definition of these subsimplices.

Given an instance $\text{I} \in \mathcal{I}$ and a corresponding subsimplex $\mathcal{S}^{\text{I}}$, we show how to define the follower's utility function so that every commitment $x \in \mathcal{S}^{\text{I}}$ is optimal, while all other commitments are suboptimal. To do so, we need to ensure that $\mathcal{P}_\theta(a^\star) = \mathcal{S}^{\text{I}}$ for every type $\theta \in \Theta$. In this way, the leader achieves utility $u^{\text{L}}(x) = 1$ when $x \in \mathcal{S}^{\text{I}}$, and utility $u^{\text{L}}(x) = 0$ when $x \notin \mathcal{S}^{\text{I}}$.

We start by considering the first type $\theta_1 \in \Theta$, and devise the utility function $u^{\text{F,I}}_{\theta_1} : \mathcal{A}_{\text{L}} \times \mathcal{A}_{\text{F}} \to [0, 1]$ in such a way that $\mathcal{P}_{\theta_1}(a^\star) = \mathcal{S}^{\text{I}}$. In order to do so, the follower's utilities must satisfy the following:

$$\begin{cases} u^{\text{F,I}}_{\theta_1}(a_i, a^\star) - u^{\text{F,I}}_{\theta_1}(a_i, a_j) = w^{\text{I}}_{j,i} & \forall a_j \in \mathcal{A}_{\text{L}} \setminus \{a^\star\}, \\ 0 \leq u^{\text{F,I}}_{\theta_1}(a_i, a_j) \leq 1 & \forall a_j \in \mathcal{A}_{\text{F}}, a_i \in \mathcal{A}_{\text{L}}. \end{cases} \tag{3}$$

Then, the follower's utility function for type $\theta_1 \in \Theta$ in instance $I \in \mathcal{I}$ is defined as follows:

$$\begin{cases} u^{\text{F,I}}_{\theta_1}(a_i, a^\star) = \frac{1}{2} & \forall a_i \in \mathcal{A}_{\text{L}} \\ u^{\text{F,I}}_{\theta_1}(a_i, a_j) = \frac{1}{2} - w^{\text{I}}_{i,j} & \forall a_i \in \mathcal{A}_{\text{L}},\ a_j \in \mathcal{A}_{\text{F}} \setminus \{a^\star\}. \end{cases}$$

With a simple calculation, it is easy to verify that the above definition of the follower's utility satisfies Equation (3). We complete the instance by defining the utility functions for types $\{\theta_2, \theta_3\}$. In particular, for $k \in \{2, 3\}$ and $a_i \in \mathcal{A}_{\text{L}}$, we let:

$$\begin{cases} u^{\text{F,I}}_{\theta_k}(a_i, a^\star) = u^{\text{F,I}}_{\theta_1}(a_i, a^\star), \\ u^{\text{F,I}}_{\theta_k}(a_i, a_j) = u^{\text{F,I}}_{\theta_1}(a_i, a_{f(j,k)}) & \forall a_j \in \mathcal{A}_{\text{F}} \setminus \{a^\star\}, \end{cases}$$

where $f(j,k) \coloneqq 1 + ((j+k+1) \mod 3)$.

It easy to see that $\mathcal{P}_{\theta_k}(a^\star) = \mathcal{S}^{\mathrm{I}}$ for every $k \in [3]$. Therefore, $a^\star_\theta(x) = a^\star$ for every $x \in \mathcal{S}^{\mathrm{I}}$ and every $\theta \in \Theta$. Instead, for every $x \notin \mathcal{S}^{\mathrm{I}}$ we have that $a^\star_\theta(x) \neq a^\star$ for every $\theta \in \Theta$. Thus, if the leader plays $x \in \mathcal{S}^{\mathrm{I}}$, they observe action $a^\star$ with probability one, otherwise they observe an action a drawn uniformly at random from $\{a_1, a_2, a_3\}$ (as μ is uniform over the types). Finally, we observe that the bit complexity of the follower's utility is bounded by $C'B$ for some constant $C' > C$.

Building the optimal regions Let $\epsilon = 2^{-B}$ for some $B > 0$ defined in the following and let $\mathcal{T}_\varepsilon$ be the regular triangulation of the simplex into subsimplices of side $\epsilon > 0$. Each instance $\mathrm{I} \in \mathcal{I}$ is associated with a subsimplex $S^{\mathrm{I}} \in \mathcal{T}_\varepsilon$.

Given an instance I, the subsimplex S^{I} has boundaries given by the hyperplanes defining the half-spaces $\{\mathcal{H}^{\mathrm{I}}_j\}_{j\in[m]}$, which we denote by $\{H^{\mathrm{I}}_j\}_{j\in[m]}$. Each H^{I}_j passes through the origin and two points on the boundary of the simplex, which for ease of exposition we take to be $(k\epsilon, 1-k\epsilon, 0)$ and $(k\epsilon, 0, 1-k\epsilon)$ for some $k \in \{0, 1, \ldots, 2^B\}$. The coefficients w^{I}_j of its algebraic representation satisfy the following:

$$\begin{aligned} k\epsilon w^{\mathrm{I}}_{j,1} + (1-k\epsilon) w^{\mathrm{I}}_{j,2} + 0 w_3 &= 0 \\ k\epsilon w^{\mathrm{I}}_{j,1} + 0 w^{\mathrm{I}}_{j,3} + (1-k\epsilon) w^{\mathrm{I}}_{j,3} &= 0 \\ -1/2 \leq w^{\mathrm{I}}_{j,i} &\leq 1/2 \quad \forall i \in [m]. \end{aligned}$$

By an argument similar to the one provided in Lemma D.2 by [2], the bit complexity of each $w^{\mathrm{I}}_{j,i}$ is at most CB for some constant $C > 0$.Thus, the bit complexity of the follower's utility is bounded by $C'B$ for some $C' > C$.

Lower bound on the regret We observe that the number of instances satisfies $|\mathcal{T}_\epsilon| = \Theta(2^{2B})$. In each instance $\mathrm{I} \in \mathcal{I}$, the leader obtains utility equal to one and observes action $a^\star$ if they play $x \in \mathcal{S}^{\mathrm{I}}$ (which is an optimal commitment). Otherwise, they collect zero utility and observe an action sampled uniformly at random from $\{a_1, a_2, a_3\}$. The behavior of an optimal deterministic algorithm is to play commitments belonging to the vertices of $\mathcal{T}_\epsilon$ according to some fixed order, and whenever it observes the optimal action $a^\star$, it starts playing it, since any commitment outside the optimal region does not provide any useful information to any algorithm. The reason why an optimal algorithm should select vertices of the subsimplices is that, if it picks a vertex and does not observe action $a^\star$, it can exclude (at most) six instances as non-optimal, since the same vertex is a vertex of (at most) six subsimplices and the follower breaks ties in favor of the leader.

Thus, given T rounds, any deterministic algorithm can check at most $6T$ instances to determine whether they are optimal or not. Let $T = \frac{(1-3/4)}{6}|\mathcal{T}_\epsilon|$ be the number of rounds. Then, the probability that the algorithm never selects an optimal commitment over T rounds is at least

$$1 - \frac{6T}{|\mathcal{T}_\epsilon|} = 1 - \frac{(1-3/4)|\mathcal{T}_\epsilon|}{|\mathcal{T}_\epsilon|} = \frac{3}{4}.$$

Therefore, the regret suffered by any deterministic algorithm is $\Theta(|\mathcal{T}_\epsilon|)$. By Yao's minimax principle, this implies that for any (possibly) randomized algorithm there exists an instance on which it suffers $\Theta(|\mathcal{T}_\epsilon|)$ regret. To conclude the proof, we set $B \coloneqq \frac{L}{C'}$, so that $\Theta(|\mathcal{T}_\epsilon|) = \Theta(2^{2B}) = \Theta(2^{2L})$ and $T = \Theta(2^{2L})$. ∎

Appendix B. Details on the Main Algorithm and `Find-Types`

In this section, we provide the details of our algorithm and the `Find-Types` procedure. The pseudocode of the main algorithm is provided in Algorithm 1. Notice that we ensure that $\widetilde{\Theta}_{h-1} \subseteq \widetilde{\Theta}_h$ for every epoch h (Line 5). We also implicitly assume that the algorithm automatically terminates after T rounds.

Algorithm 1 `No-Regret-Bayesian-Stackelberg`

Require: $T \in \mathbb{N}, \delta \in (0,1)$

1: $\epsilon_1 \leftarrow 1/K$, $\mathcal{X}_1 \leftarrow \Delta(\mathcal{A}_{\text{L}})$, $\widetilde{\Theta}_0 \leftarrow \varnothing$
2: $\delta_1 \leftarrow \frac{\delta}{2\lceil \log_4(5T) \rceil}$, $\delta_2 \leftarrow \delta_1$
3: **for** $h = 1, 2, \ldots$ **do**
4: $\quad \overline{\Theta}_h, \widehat{\mu}_h \leftarrow$ `Find-Types`$(\mathcal{X}_h, \epsilon_h, \delta_1)$
5: $\quad \widetilde{\Theta}_h \leftarrow \widetilde{\Theta}_{h-1} \cup \overline{\Theta}_h$
6: $\quad \mathcal{Y}_h \leftarrow$ `Find-Partition`$(\mathcal{X}_h, \epsilon_h, \widetilde{\Theta}_h, \widetilde{\Theta}_{h-1}, \delta_2)$
7: $\quad \mathcal{X}_{h+1} \leftarrow$ `Prune`$(\mathcal{Y}_h, \epsilon_h, \widetilde{\Theta}_h, \widetilde{\Theta}_{h-1}, \widehat{\mu}_h)$
8: $\quad \epsilon_{h+1} \leftarrow \epsilon_h/2$
9: **end for**

Structure of the Leader's Decision Space It is crucial to notice that the leader's decision space $\mathcal{X}_h$ is computed as the union, over action profiles $\boldsymbol{a}_{h-1} \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_{h-1})$, of polytopes $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1})$ with pairwise zero-volume intersections. Formally, we have:

$$\mathcal{X}_h \coloneqq \bigcup_{\boldsymbol{a}_{h-1} \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_{h-1})} \mathcal{X}_h(\boldsymbol{a}_{h-1}). \tag{4}$$

For each $\boldsymbol{a}_{h-1}$, the set $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ is the subset of $\mathcal{P}(\boldsymbol{a}_{h-1})$ in which the leader's expected utility is at most $\mathcal{O}(\epsilon_{h-1})$-suboptimal.[1] Furthermore, in order to guarantee that Equation (4) is well defined for every epoch $h \geq 1$ including the first one, we let $\widetilde{\Theta}_0 \coloneqq \varnothing$, $\mathcal{A}_{\text{F}}(\widetilde{\Theta}_0) \coloneqq \{\perp\}$, and $\mathcal{X}_0(\perp) \coloneqq \Delta(\mathcal{A}_{\text{L}})$.

Details of the `Find-Types` procedure We now discuss the first procedure executed by Algorithm 1 at each epoch h, namely `Find-Types`, whose pseudocode is presented in Algorithm 2. The goal of this procedure is to compute an estimator $\widehat{\mu}_h$ of the prior distribution μ such that $\|\widehat{\mu}_h - \mu\|_\infty \leq \epsilon_h$, together with a subset of types $\overline{\Theta}_h \subseteq \Theta$ satisfying $\mu_\theta \geq \epsilon_h$ for all $\theta \in \overline{\Theta}_h$. In order to achieve its goal, Algorithm 2 selects an arbitrary strategy in the leader's decision space $\mathcal{X}_h$ and commits to it for $T_{h,1} = O(1/\epsilon_h^2)$ rounds. By using the realized types observed during these $T_{h,1}$ rounds of leader-follower interaction, the algorithm constructs the estimator $\widehat{\mu}_h$. Based on this estimator, it then defines $\overline{\Theta}_h$ as the set of all follower types θ such that $\widehat{\mu}_{h,\theta} \geq 2\epsilon_h$.

1. With a slight abuse of notation, we denote by $\boldsymbol{a}_h$ an element of $\mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$ so as to distinguish it from action profiles associated with other epochs.

Algorithm 2 `Find-Types`

Require: $\epsilon_h > 0, \varnothing \neq \mathcal{X}_h \subseteq \Delta(\mathcal{A}_{\mathrm{L}}), \delta_1 \in (0,1)$

1: $T_{h,1} \leftarrow \left\lceil \frac{1}{2\epsilon_h^2} \log\left(\frac{2K}{\delta_1}\right) \right\rceil$
2: Play any $x \in \mathcal{X}_h$ for $T_{h,1}$ rounds and observe $\theta_t \sim \mu$
3: Compute $\widehat{\mu}_h$ with the observed feedback
4: $\overline{\Theta} \leftarrow \{\theta \in \Theta \mid \widehat{\mu}_{h,\theta} \geq 2\epsilon_h\}$
5: **Return** $\widehat{\mu}_h, \overline{\Theta}$

Lemma 6 *Let $\epsilon_h, \delta_1 \in (0,1)$ and let $\mathcal{X}_h \subseteq \Delta(\mathcal{A}_L)$ be non-empty. Then, with probability at least $1-\delta_1$, the `Find-Types` procedure computes:*

(i) an estimator $\hat{\mu}_h \in \Delta(\Theta)$ such that $\|\mu - \hat{\mu}_h\|_\infty \leq \epsilon_h$;

(ii) a set of types $\widetilde{\Theta}_h \subseteq \Theta$ such that $\mu_\theta \geq \epsilon_h$ for every $\theta \in \widetilde{\Theta}_h$ and $\mu_\theta \leq 3\epsilon_h$ for every other type;

by using $\mathcal{O}\left(\frac{1}{\epsilon_h^2} \log\left(\frac{K}{\delta_1}\right)\right)$ rounds.

Proof Algorithm 2 computes the empirical estimator $\widehat{\mu}_h$ of μ using

$$T_1 \coloneqq \left\lceil \frac{1}{2\epsilon_h^2} \log\left(\frac{2K}{\delta_1}\right) \right\rceil \tag{5}$$

samples. Subsequently it computes the set $\widetilde{\Theta} \coloneqq \{\theta \in \Theta \mid \widehat{\mu}_{h,\theta} \geq 2\epsilon_h\}$.

Applying a union bound and Hoeffding bound we get that $\|\widehat{\mu}_{h,\theta} - \mu\|_\infty \leq \epsilon_h$ with probability at least $1-\delta_1$. Therefore, with probability at least $1-\delta_1$, when $\widehat{\mu}_{h,\theta} \geq 2\epsilon$ the true probability μ_θ satisfies $\mu_\theta \geq \epsilon_h$, while when $\widehat{\mu}_{h,\theta} \leq 2\epsilon_h$, we have $\mu_\theta \leq 3\epsilon_h$. ■

Appendix C. Details on the `Find-Partition` procedure

C.1. Details of the `Find-Partition` procedure

The pseudocode of `Find-Partition` is provided in Algorithm 3. The procedure computes the mapping $\mathcal{Y}_h$ defined as follows:

$$\mathcal{Y}_h(\boldsymbol{a}) \coloneqq \begin{cases} \mathcal{P}(\boldsymbol{a}) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1}) & \text{if } \mathrm{vol}(\mathcal{P}(\boldsymbol{a}) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1})) > 0 \\ \emptyset & \text{otherwise.} \end{cases} \tag{6}$$

Observe that, since $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1})$, the region $\mathcal{Y}_h(\boldsymbol{a})$ is equivalent to $\mathcal{X}_h \cap \mathcal{P}(\boldsymbol{a})$ up to zero-volume regions.

In order to compute the mapping $\mathcal{Y}_h$, Algorithm 3 must be executed satisfying the following condition.

Condition 7 *Algorithm 3 is called with the following inputs: parameters $\epsilon_h, \delta_2 \in (0,1)$, sets of types $\widetilde{\Theta}_{h-1} \subseteq \widetilde{\Theta}_h \subseteq \Theta$ such that $\mu_\theta \geq \epsilon_h$ for every $\theta \in \widetilde{\Theta}_h \setminus \widetilde{\Theta}_{h-1}$, search space*

$$\mathcal{X}_h = \bigcup_{\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})} \mathcal{X}_h(\boldsymbol{a}_{h-1}),$$

where each $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1})$ is empty or a non-zero-volume polytope.

Intuitively, Condition 7 requires that Algorithm 2 and Algorithm 3 have been correctly executed every previous epoch. In Appendix E we will show that this condition is always verified with high probability.

Algorithm 3 builds on a procedure developed by Bacchiocchi et al. [2], which is designed to learn the best-response regions $\mathcal{P}_\theta(a)$ in the single-typed follower setting, *i.e.*, when Θ is a singleton. This procedure requires access to an oracle that maps each leader's commitment $x \in \Delta(\mathcal{A}_\text{L})$ to the corresponding agent's best response $a^\star_\theta(x)$. In our setting, such an oracle can be implemented by repeatedly letting the leader commit to the same mixed strategy x until a follower of type θ is sampled. In such a case, we say that the oracle *queries* the strategy x. Let us also observe that the procedure in [2] was originally designed to operate over the simplex $\Delta(\mathcal{A}_\text{L})$. However, it can be easily adapted to work over a generic polytope $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ (see, *e.g.*, [1]). Thus, the following holds.

Lemma 8 *[Restate [2]] Let $S \subseteq \Delta(\mathcal{A}_\text{L})$ be a polytope with $\text{vol}(S) > 0$ defined as the intersections of N hyperplanes whose coefficients have bit complexity bounded by some $B \geq L$. Then, given any $\zeta \in (0, 1)$ and $\theta \in \Theta$, under the event that each query terminates in a finite number rounds, there exists an algorithm, denoted by* `Stackelberg`*, that computes the polytopes $\mathcal{P}_\theta(a \mid S)$ according to Equation (7) for each $a \in \mathcal{A}_\text{F}$ by using at most*

$$\widetilde{\mathcal{O}}\left(n^2\left(m^7 B \log\left(\frac{1}{\zeta}\right) + \binom{N+n}{m}\right)\right)$$

queries with probability at least $1 - \zeta$.

We will say that an execution of `Stackelberg` is *successful* when it computes the polytopes $\mathcal{P}_\theta(a|S)$ according to Equation (7) in the number of rounds specified in the lemma above. These polytopes $\mathcal{P}_\theta(a|S)$ are equivalent to $\mathcal{P}_\theta(a) \cap \mathcal{S}$ up to zero-volume regions, and are formally defined as:

$$\mathcal{P}_\theta(a \mid S) \coloneqq \begin{cases} \mathcal{P}_\theta(a) \cap S & \text{if } \text{vol}(\mathcal{P}_\theta(a) \cap S) > 0 \\ \emptyset & \text{otherwise.} \end{cases} \tag{7}$$

Next, we describe Algorithm 3. At a high level, it consists of three macro blocks.

1. In the *first* macro block (Lines 3–7), Algorithm 3 computes the regions $\mathcal{P}_\theta(\cdot \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ for each type in $\widetilde{\Theta}_{h-1}$ and each action profile in $\mathcal{A}_\text{F}(\widetilde{\Theta}_{h-1})$. We notice that Algorithm 3 does not require any leader-follower interactions here, as it simply manipulates the follower's best-response regions associated with follower types in $\widetilde{\Theta}_{h-1}$, which have been computed in previous epochs.

2. In the *second* macro block (Lines 8–11), Algorithm 3 invokes the `Stackelberg` procedure for each type in $\widetilde{\Theta}_h \setminus \widetilde{\Theta}_{h-1}$ and each action profile in $\mathcal{A}_\text{F}(\widetilde{\Theta}_{h-1})$. By Lemma 8, the `Stackelberg` procedure is guaranteed to learn the polytope $R(a) = \mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ for every $a \in \mathcal{A}_\text{F}$. Thus, at Line 10, Algorithm 3 sets $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ equal to $R(a)$, for every $a \in \mathcal{A}_\text{F}$.

3. Finally, in the *third* macro block, Algorithm 3 computes $\mathcal{Y}_h(\boldsymbol{a}_h)$ for every $\boldsymbol{a}_h$ belonging to $\mathcal{A}_\text{F}(\widetilde{\Theta}_h)$. by intersection suitable polytopes.

Algorithm 3 provides the following guarantees.

Algorithm 3 `Find-Partition`

Require $\mathcal{X}_h$, $\widetilde{\Theta}_{h-1} \subseteq \widetilde{\Theta}_h \subseteq \Theta$, $\epsilon_h, \delta_2 \in (0,1)$
$\zeta \leftarrow \delta_2/2$
for $\theta \in \widetilde{\Theta}_{h-1}, \boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$ **do**
  $a \leftarrow \boldsymbol{a}_{h-1,\theta}$
  $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) \leftarrow \mathcal{X}_h(\boldsymbol{a}_{h-1})$
  $\mathcal{P}_\theta(a' \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) \leftarrow \varnothing \quad \forall a' \neq a$
end for
for $\theta \in \widetilde{\Theta}_h \setminus \widetilde{\Theta}_{h-1}, \boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$ **do**
  $R \leftarrow$ `Stackelberg`$(\mathcal{X}_h(\boldsymbol{a}_{h-1}), \theta, \zeta)$
  $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) \leftarrow R(a) \quad \forall a \in \mathcal{A}_{\mathrm{F}}$
end for
for $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ **do**
  $\boldsymbol{a}_{h-1} \leftarrow \boldsymbol{a}_h | \widetilde{\Theta}_{h-1}$
  if $\mathrm{vol}\left(\bigcap_{\theta \in \widetilde{\Theta}_h} \mathcal{P}_\theta(\boldsymbol{a}_{h,\theta} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))\right) > 0$ **then**
    $\mathcal{Y}_h(\boldsymbol{a}_h) \leftarrow \bigcap_{\theta \in \widetilde{\Theta}_h} \mathcal{P}_\theta(\boldsymbol{a}_{h,\theta} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$
  else
    $\mathcal{Y}_h(\boldsymbol{a}_h) \leftarrow \emptyset$
  end if
end for
Return $\mathcal{Y}_h$

Lemma 9 *[Formal version of Lemma 3] Suppose that Condition 7 is satisfied and each polytope $\mathcal{X}_h(\boldsymbol{a})$ composing $\mathcal{X}_h$ is defined as the intersections of N hyperplanes whose coefficients have bit complexity bounded by some $B \geq L$. Then Algorithm 3 computes $\mathcal{Y}_h$ according to Equation (6) with probability at least $1 - \delta_2$, using at most:*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h} K^{m+1} n^{2m+2} \left(m^7 B \log^2\left(\frac{1}{\delta_2}\right) + \binom{N+n}{m}\right)\right)$$

rounds.

We observe that, in principle, the number of action profiles $\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$ is exponential in K, and Algorithm 3 would therefore require a number of rounds exponential in K. However, this is *not* the case, and Algorithm 3 exhibits an exponential dependence only on m. Intuitively, this is because most regions $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ are empty, and thus executing the `Stackelberg` procedure over them requires no leader-follower interactions. More formally, it is possible to show that the number of non-empty best-response regions $\mathcal{P}(\boldsymbol{a}_{h-1})$ is bounded by $K^m n^{2m}$ (see [14, Lemma 3.2]). Thus, since $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1})$, we can show that the `Stackelberg` procedure is actually executed at most $\mathcal{O}(K^m n^{2m})$ times. Furthermore, in Section E we bound B and N, thus recovering the guarantees of Lemma 3.

C.2. Intermediate lemmas and Proof of Lemma 3

To prove Lemma 9 (formal version of Lemma 3), we first provide two intermediate lemmas. These two results show that, when the `Stackelberg` subprocedure is always executed correctly, Algo-

rithm 3 computes $\mathcal{Y}_h$ according to Equation (6). The proof of Lemma 9 follows by bounding the number of samples required by such a procedure and the probability of correctly executing it.

Lemma 10 *Suppose that Condition 7 is satisfied and every execution of* `Stackelberg` *is successful, Algorithm 3 computes correctly* $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ *according to Equation 7 for every* $\theta \in \widetilde{\Theta}_h$, $a \in \mathcal{A}_{\mathrm{F}}$ *and* $\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$.

Proof For the types $\theta \in \widetilde{\Theta}_h \setminus \widetilde{\Theta}_{h-1}$, the algorithm computes $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ by means of the `Stackelberg` subprocedure, which computes it according to Equation (6) when successful. There remains to be considered the types in $\widetilde{\Theta}_{h-1}$. Let $\theta \in \widetilde{\Theta}_{h-1}$, $a \in \mathcal{A}_{\mathrm{F}}$ and $\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$. In the following we let $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ be defined according to Equation (7) and $\mathcal{P}'_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ be the region computed by Algorithm 3 at Lines 5 and 6.

Suppose that $a = \boldsymbol{a}_{h-1,\theta}$. Then Algorithm 3 computes $\mathcal{P}'_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) = \mathcal{X}_h(\boldsymbol{a}_{h-1})$. Condition 7 guarantees that $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1})$. As a result, we have that:

$$\mathcal{P}'_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) = \mathcal{X}_h(\boldsymbol{a}_{h-1}) = \mathcal{X}_h(\boldsymbol{a}_{h-1}) \cap \mathcal{P}(\boldsymbol{a}_{h-1})$$

Observe that such a polytope $\mathcal{X}_h(\boldsymbol{a}_{h-1}) = \mathcal{X}_h(\boldsymbol{a}_{h-1}) \cap \mathcal{P}(\boldsymbol{a}_{h-1})$ cannot have zero volume while being non-empty. Indeed, $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ is either empty or has strictly positive volume, as required by Condition 7. Therefore, $\mathcal{P}'_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) = \mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ when $a = \boldsymbol{a}_{h-1,\theta}$.

Suppose instead that $a \neq \boldsymbol{a}_{h-1,\theta}$. By construction, Algorithm 3 computes $\mathcal{P}'_\theta(a|\mathcal{X}_h(\boldsymbol{a}_{h-1})) = \emptyset$. At the same time, we have that $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}_\theta(\boldsymbol{a}_{h-1,\theta})$ and $\mathrm{vol}(\mathcal{P}_\theta(\boldsymbol{a}_{h-1,\theta}) \cap \mathcal{P}_\theta(a)) = 0$, as $\boldsymbol{a}_{h-1,\theta} \neq a$. Consequently, $\mathrm{vol}(\mathcal{X}_h(\boldsymbol{a}_{h-1}) \cap \mathcal{P}_\theta(a)) = 0$, and by Equation (6), we have:

$$\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) = \emptyset = \mathcal{P}'_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})).$$

As a result, Algorithm 3 computes $\mathcal{P}_\theta(\cdot|\mathcal{X}_h(\cdot))$ correctly for every $\theta \in \widetilde{\Theta}_h$, concluding the proof. ■

Lemma 11 *If Condition 7 is satisfied and every execution of* `Stackelberg` *is successful, Algorithm 3 correctly computes* $\mathcal{Y}_h$ *according to Equation* (6).

Proof We let $\mathcal{Y}_h$ be computed according to Equation (6), and $\mathcal{Y}'_h$ be the one computed by Algorithm 3 at Line 15.

Formally, for every $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ we let $\boldsymbol{a}_{h-1}$ be the restriction of $\boldsymbol{a}_h$ to $\mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$ and:

$$\mathcal{Y}_h(\boldsymbol{a}_h) \coloneqq \begin{cases} \mathcal{P}(\boldsymbol{a}_h) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1}) & \text{if its volume is greater than zero} \\ \emptyset & \text{otherwise,} \end{cases}$$

according to Equation (6). As for Lemma 10, Algorithm 3 correctly computes the regions $\mathcal{P}_\theta(a \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$, therefore Line 15 Algorithm 3 computes:

$$\mathcal{Y}'_h(\boldsymbol{a}_h) = \begin{cases} \bigcap_{\theta \in \widetilde{\Theta}_h} \mathcal{P}_\theta(\boldsymbol{a}_{h,\theta} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) & \text{if its volume is greater than zero} \\ \emptyset & \text{otherwise.} \end{cases} \tag{8}$$

The two regions $\mathcal{Y}_h(\boldsymbol{a}_h)$ and $\mathcal{Y}'_h(\boldsymbol{a}_h)$ are both empty when $\mathcal{X}_h(\boldsymbol{a}_{h-1}) = \emptyset$. We now show that the two coincide even when $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \neq \emptyset$. Observe that Condition 7 requires that $\mathrm{vol}(\mathcal{X}_h(\boldsymbol{a}_{h-1})) > 0$ when the polytope is not empty.

Suppose that $\mathcal{P}_\theta(\boldsymbol{a}_{h,\theta} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ is not empty for every $\theta \in \widetilde{\Theta}_h$. Then, by Equation (7), $\mathcal{P}_\theta(\boldsymbol{a}_{h,\theta} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})) = \mathcal{P}_\theta(\boldsymbol{a}_{h,\theta}) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1})$. As a result:

$$\mathcal{P}(\boldsymbol{a}_h) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1}) = \bigcap_{\theta \in \widetilde{\Theta}_h} \mathcal{P}_\theta(\boldsymbol{a}_{h,\theta}) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1}) = \bigcap_{\theta \in \widetilde{\Theta}_h} \mathcal{P}_\theta(\boldsymbol{a}_{h,\theta} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1})),$$

where the first equality uses the definition of $\mathcal{P}(\boldsymbol{a}_h)$, and the second Equation (7). Therefore, by applying Equation (6) and Equation (8), we have that $\mathcal{Y}_h(\boldsymbol{a})$ and $\mathcal{Y}'_h(\boldsymbol{a})$ coincide.

Suppose now that there exists some $\bar{\theta} \in \widetilde{\Theta}_h$ such that $\mathcal{P}_{\bar{\theta}}(\boldsymbol{a}_{h,\bar{\theta}} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ is empty. Then according to Equation (8) $\mathcal{Y}'_h(\boldsymbol{a}_h)$ is also empty. Furthermore, by Equation (7), for $\mathcal{P}_{\bar{\theta}}(\boldsymbol{a}_{h,\bar{\theta}} \mid \mathcal{X}_h(\boldsymbol{a}_{h-1}))$ to be empty, we need $\mathcal{P}_{\bar{\theta}}(\boldsymbol{a}_{h,\bar{\theta}})$ to have null volume (possibly to be empty). This implies that $\mathcal{P}(\boldsymbol{a}_h) \cap \mathcal{X}_h(\boldsymbol{a}_{h-1})$ has zero volume too, as it is a subset of $\mathcal{P}_{\bar{\theta}}(\boldsymbol{a}_{h,\bar{\theta}})$. As a result, $\mathcal{Y}_h(\boldsymbol{a}_h)$ is empty and coincides with $\mathcal{Y}'_h(\boldsymbol{a}_h)$, concluding the proof. ■

Lemma 12 *[Formal version of Lemma 3] Suppose that Condition 7 is satisfied and each polytope $\mathcal{X}_h(\boldsymbol{a})$ composing $\mathcal{X}_h$ is defined as the intersections of N hyperplanes whose coefficients have bit complexity bounded by some $B \geq L$. Then Algorithm 3 computes $\mathcal{Y}_h$ according to Equation (6) with probability at least $1 - \delta_2$, using at most:*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h} K^{m+1} n^{2m+2} \left(m^7 B \log^2\left(\frac{1}{\delta_2}\right) + \binom{N+n}{m}\right)\right)$$

rounds.

Proof As of Lemma 11, Algorithm 3 computes $\mathcal{Y}_h$ correctly if every execution of `Stackelberg` is successful. We therefore have to bound the number of rounds required by these procedures and the probability that they correct terminate.

The `Stackelberg` subprocedure is executed once for every type $\theta \in \widetilde{\Theta}_h \setminus \widetilde{\Theta}_{h-1}$ and every action profile $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$. However, it takes exactly zero rounds when it is executed over an empty region $\mathcal{X}_h(\boldsymbol{a}_{h-1})$. By [14] Lemma 3.2 the number of non-empty regions $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ is at most $K^m n^{2m}$. As a result, the number of actual calls to the subrpocedure is bounded by $K^{m+1} n^{2m}$, which accounts for at most K calls for each non-empty region.

According to Lemma 8, under the event that each query ends in a finite number of rounds, each execution of `Stackelberg` correctly terminates with probability at least ζ and employs

$$\widetilde{\mathcal{O}}\left(n^2 \left(m^7 B \log \frac{1}{\zeta} + \binom{N+n}{m}\right)\right) = \mathcal{O}\left(n^2 \left(m^7 B \log \frac{1}{\delta_2} + \binom{N+n}{m}\right)\right)$$

queries, where $\zeta \coloneqq \delta/2$ is define at Line 2 Algorithm 3. Therefore, the number of queries C performed by Algorithm 3 is:

$$C \leq \widetilde{\mathcal{O}}\left(K^{m+1} n^{2m} n^2 \left(m^7 B \log \frac{1}{\delta_2} + \binom{N+n}{m}\right)\right)$$

with probability at least $1 - \zeta = 1 - \delta_2/2$.

In order to conclude the proof, we bound the number of samples required by the algorithm to terminate correctly with probability at least $1 - \delta$. We observe that every query is performed over

a type θ such that $\mu_\theta \geq \epsilon_h$. A simple probabilistic argument (see *e.g.*, Lemma 2 in [1]) shows that given any $\rho \in (0,1)$, a query ends in at most $T_{\mathrm{q}}(\rho) \coloneqq \lceil 1/\epsilon \log(1/\rho) \rceil$ with probability at least $1-\rho$, *i.e.*, with probability at least $1-\rho$ any given type appears in $T_{\mathrm{q}}(\rho)$ rounds. By considering $\rho = \zeta/2C$ and employing an union bound over the event that Algorithm 3 executes C queries, we have that with probability at least:

$$1 - \zeta - C\rho = 1 - \frac{\delta_2}{2} - C\frac{\zeta}{2C} = 1 - \delta_2$$

Algorithm 3 terminates correctly in $T_{\mathrm{P}} \coloneqq CT_{\mathrm{q}}(\rho)$ rounds. In order to conclude the proof, we observe that:

$$\begin{aligned} T_{\mathrm{P}} &= CT_{\mathrm{q}} \\ &= \widetilde{\mathcal{O}}\left(\frac{C}{\epsilon_h}\log\left(\frac{C}{\delta_2}\right)\right) \\ &= \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h}K^{m+1}n^{2m+2}\left(m^7 B \log^2\frac{1}{\delta_2} + \binom{N+n}{m}\right)\right), \end{aligned}$$

proving the statement. ■

Appendix D. Details of the `Prune` procedure

In this section we provide the details of the procedure `Prune` (Algorithm 4). This procedure informally requires that Algorithm 2 and Algorithm 3 were executed "successfully", which happens with high probability. Formally, `Prune` must be executed under the following condition.

Condition 13 *Algorithm 3 is called with the following inputs:*

1. *a parameter* $\epsilon_h \in (0,1)$.
2. *two sets of types* $\widetilde{\Theta}_{h-1} \subseteq \widetilde{\Theta}_h \subseteq \Theta$ *such that* $\widetilde{\Theta}_h \neq \emptyset$ *and* $\mu_\theta \leq 3\epsilon_h$ *for every* $\theta \in \Theta \setminus \widetilde{\Theta}_h$.
3. *a prior estimator* $\widehat{\mu}_h$ *such that* $\|\mu - \widehat{\mu}_h\|_\infty \leq \epsilon_h$.
4. *a search space* $\mathcal{X}_h = \bigcup_{\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})} \mathcal{X}_h(\boldsymbol{a}_{h-1})$ *where each* $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \subseteq \mathcal{P}(\boldsymbol{a}_{h-1})$ *is either empty or a polytope with volume greater than zero.*
5. *a mapping* $\mathcal{Y}_h$ *satisfies Equation (6) and such that an optimal commitment* $x^\star$ *belongs to some* $\mathcal{Y}_h(\boldsymbol{a}_h)$*, with* $\boldsymbol{a}_{h,\theta} = a^\star_\theta(x^\star)$ *for every* $\theta \in \widetilde{\Theta}_h$.

During its execution, Algorithm 4 relies on empirical estimates of the leader's expected utility. Specifically, for every $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ and $x \in \mathcal{Y}_h(\boldsymbol{a}_h)$, we define

$$\widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) \coloneqq \sum_{\theta \in \widetilde{\Theta}_h} \widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x, \boldsymbol{a}_{h,\theta}). \tag{9}$$

We notice that these estimates ignore types $\theta \notin \widetilde{\Theta}_h$ and rely on the empirical estimator $\widehat{\mu}_h$, incurring an approximation error of at most $\mathcal{O}(K\epsilon_h)$ thanks to Lemma 2.

Algorithm 4 `Prune`

Require: $\mathcal{Y}_h, \widetilde{\Theta}_h, \epsilon_h, \widehat{\mu}_h$.
1: $C_1 \leftarrow 3, C_2 \leftarrow 6$
2: $\underline{\text{OPT}}_h \leftarrow$ Equation (10)
3: **for** $\boldsymbol{a} \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$ **do**
4: $\quad \mathcal{H}_h(\boldsymbol{a}) \leftarrow$ Equation (11)
5: $\quad \mathcal{X}_{h+1}(\boldsymbol{a}) \leftarrow \mathcal{Y}_h(\boldsymbol{a}) \cap \mathcal{H}_h(\boldsymbol{a})$
6: $\quad$ **if** $\text{vol}(\mathcal{X}_{h+1}(\boldsymbol{a})) = 0$ **then**
7: $\quad\quad \mathcal{X}_{h+1}(\boldsymbol{a}) \leftarrow \varnothing$
8: $\quad$ **end if**
9: **end for**
10: **Return** $\mathcal{X}_{h+1} = \bigcup_{\boldsymbol{a} \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)} \mathcal{X}_{h+1}(\boldsymbol{a})$

We now describe how Algorithm 4 works. As a first step, it computes a lower bound $\underline{\text{OPT}}_h$ on the value of an optimal commitment, defined as:

$$\underline{\text{OPT}}_h \coloneqq \max_{\substack{\boldsymbol{a} \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h) \\ x \in \mathcal{Y}_h(\boldsymbol{a})}} \widehat{u}_h^{\text{L}}(x, \boldsymbol{a}) - C_2 K \epsilon_h, \tag{10}$$

where C_2 is a constant defined at Line 1. The procedure then computes the polytope $\mathcal{X}_{h+1}(\boldsymbol{a}_h)$, for every $\boldsymbol{a}_h \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$, by *pruning* the subset of $\mathcal{Y}_h(\boldsymbol{a}_h)$ that is guaranteed to yield low utility. To this end, it defines the half-spaces $\mathcal{H}_h(\boldsymbol{a}_h) \subseteq \mathbb{R}^m$ as follows:

$$\mathcal{H}_h(\boldsymbol{a}_h) \coloneqq \{x \in \mathbb{R}^m \mid \widehat{u}_h^{\text{L}}(x, \boldsymbol{a}_h) + C_1 K \epsilon_h \geq \underline{\text{OPT}}_h\}, \tag{11}$$

where C_1 is a constant defined at Line 1. Then, for every $\boldsymbol{a}_h \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$, the polytope $\mathcal{X}_{h+1}(\boldsymbol{a}_h)$ is defined as the intersection of the half-space $\mathcal{H}_h(\boldsymbol{a}_h)$ and $\mathcal{Y}_h(\boldsymbol{a}_h)$,

$$\mathcal{X}_{h+1}(\boldsymbol{a}_h) \coloneqq \mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h),$$

as implemented at Line 5.

The half-spaces $\mathcal{H}_h(\boldsymbol{a}_h)$ defined in Equation (11) guarantee two crucial properties: (i) an optimal commitment belongs to at least one such $\mathcal{H}_h(\boldsymbol{a}_h)$ and therefore to the new search space $\mathcal{X}_{h+1}$, and (ii) all commitments belonging to such half-spaces are $\mathcal{O}(K\epsilon_h)$-optimal.

The remaining part of the appendix is organized as follows. First, we provide two additional lemmas, namely Lemma 14 and Lemma 15. The first provide upper and lower bounds on the quantity $\sum_{\theta \in \widetilde{\Theta}_h} \mu_\theta u^{\text{L}}(x, \boldsymbol{a}_\theta)$, which is the leader utility in $x \in \Delta(\mathcal{A}_{\text{L}})$ assuming that only the types in $\widetilde{\Theta}_h$ are drawn from μ, and that they respond according to the action profile $\boldsymbol{a}_\theta \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$. Lemma 15 provides instead upper and lower bounds on the actual utility $u^{\text{L}}(x)$ in the terms of the estimated utility $\widehat{u}_h^{\text{L}}(x, \boldsymbol{a}_h)$, for any $\boldsymbol{a}_h \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$ and $x \in \mathcal{Y}_h(\boldsymbol{a}_h)$. Together, these two lemmas are employed to prove Lemma 17, which is the formal version of Lemma 4. Finally, we conclude this appendix with Lemma 18 and its proof, which will be instrumental to upper bound the number of facets of each polytope composing $\mathcal{X}_h$ (see Lemma 25).

Lemma 14 *If Condition 13 is satisfied, then for every $\boldsymbol{a} \in \mathcal{A}_{\text{F}}(\widetilde{\Theta}_h)$ and $x \in \Delta(\mathcal{A}_{\text{L}})$, we have:*

$$\widehat{u}_h^{\text{L}}(x, \boldsymbol{a}) - K\epsilon \leq \sum_{\theta \in \widetilde{\Theta}_h} \mu_\theta u^{\text{L}}(x, \boldsymbol{a}_\theta) \leq \widehat{u}_h^{\text{L}}(x, \boldsymbol{a}) + K\epsilon.$$

Proof We observe that:

$$\left|\widehat{u}_h^{\mathrm{L}}(x,\boldsymbol{a}) - \sum_{\theta\in\widetilde{\Theta}_h}\mu_\theta u^{\mathrm{L}}(x,\boldsymbol{a}_\theta)\right| = \left|\sum_{\theta\in\widetilde{\Theta}_h}(\widehat{\mu}_{h\theta}-\mu_\theta)u^{\mathrm{L}}(x,\boldsymbol{a}_\theta)\right| \leq K\epsilon_h$$

where the inequality holds because, under Condition 13, $\|\widehat{\mu}_h - \mu\|_\infty \leq \epsilon$ and $|\widetilde{\Theta}_h| \leq K$. The statement follows by unraveling the absolute value. ■

Lemma 15 *If Condition 13 is satisfied, then for every $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ and $x \in \mathcal{Y}_h(\boldsymbol{a}_h)$:*

$$u^{\mathrm{L}}(x) \geq \widehat{u}_h^{\mathrm{L}}(x,\boldsymbol{a}_h) - \epsilon|\widetilde{\Theta}_h|.$$

Furthermore, if $\boldsymbol{a}_{h,\theta} = a^\star_\theta(x)$ for every $\theta \in \widetilde{\Theta}_h$, it holds that:

$$u^{\mathrm{L}}(x) \leq \widehat{u}_h^{\mathrm{L}}(x,\boldsymbol{a}_h) + 4K\epsilon_h.$$

Proof Consider any $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ and let $\boldsymbol{a}_{h-1}$ be its restriction to $\mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$. Thanks to Equation (6), we have that $\mathcal{Y}_h(\boldsymbol{a}_h) \subseteq \mathcal{P}(\boldsymbol{a}_h)$. Now take any $x \in \mathcal{Y}_h(\boldsymbol{a}_h) \subseteq \mathcal{P}(\boldsymbol{a}_h)$. We observe that for every $\theta \in \widetilde{\Theta}_h$, a follower of type θ is indifferent in x between action $\boldsymbol{a}_{h,\theta}$ and $a^\star_\theta(x)$.[2] As the follower breaks ties in favor of the leader, we have:

$$u^{\mathrm{L}}(x, a^\star_\theta(x)) \geq u^{\mathrm{L}}(x,\boldsymbol{a}_{h,\theta}). \tag{12}$$

By employing this inequality, we lower bound the leader's utility in x as follows:

$$\begin{aligned}
u^{\mathrm{L}}(x) &= \sum_{\theta\in\Theta}\mu_\theta u^{\mathrm{L}}(x,a^\star_\theta(x)) \\
&= \sum_{\theta\in\widetilde{\Theta}_h}\mu_\theta u^{\mathrm{L}}(x,a^\star_\theta(x)) + \sum_{\theta\notin\widetilde{\Theta}_h}\mu_\theta u^{\mathrm{L}}(x,a^\star_\theta(x)) \\
&\geq \sum_{\theta\in\widetilde{\Theta}_h}\mu_\theta u^{\mathrm{L}}(x,\boldsymbol{a}_{h,\theta}) + \sum_{\theta\notin\widetilde{\Theta}_h}\mu_\theta u^{\mathrm{L}}(x,a^\star_\theta(x)) \\
&\geq \sum_{\theta\in\widetilde{\Theta}_h}\mu_\theta u^{\mathrm{L}}(x,\boldsymbol{a}_{h,\theta}) \\
&= \sum_{\theta\in\widetilde{\Theta}_h}\widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x,\boldsymbol{a}_{h,\theta}) + \sum_{\theta\in\widetilde{\Theta}_h}(\mu_\theta - \widehat{\mu}_{h,\theta})u^{\mathrm{L}}(x,\boldsymbol{a}_\theta) \\
&\geq \sum_{\theta\in\widetilde{\Theta}_h}\widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x,\boldsymbol{a}_{h,\theta}) - \epsilon_h|\widetilde{\Theta}_h| = \widehat{u}_h^{\mathrm{L}}(x,\boldsymbol{a}_h) - \epsilon_h|\widetilde{\Theta}_h|
\end{aligned}$$

where the first inequality follows from Equation (12), the second removes a non-negative quantity, and the last one follows from Condition 13, in particular that $\|\mu - \widehat{\mu}_h\|_\infty \leq \epsilon_h$ and that the utility is bounded in $[0,1]$.

2. Notice that the two actions are different when x is on the boundary between two best-response regions.

Now consider an action profile $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ a commitment $x \in \mathcal{Y}_h(\boldsymbol{a}_h)$ such that $\boldsymbol{a}_{h,\theta} = a^\star_\theta(x)$ for every $\theta \in \widetilde{\Theta}_h$. We can upper bound $u^{\mathrm{L}}(x)$ as follows:

$$\begin{aligned}
u^{\mathrm{L}}(x) &= \sum_{\theta\in\Theta} \mu_\theta u^{\mathrm{L}}(x, a^\star_\theta(x)) \\
&= \sum_{\theta\in\widetilde{\Theta}_h} \mu_\theta u^{\mathrm{L}}(x, \boldsymbol{a}_{h,\theta}) + \sum_{\theta\in\Theta\setminus\widetilde{\Theta}_h} \mu_\theta u^{\mathrm{L}}(x, a^\star_\theta(x)) \\
&\leq \sum_{\theta\in\widetilde{\Theta}_h} \mu_\theta u^{\mathrm{L}}(x, \boldsymbol{a}_{h,\theta}) + 3K\epsilon_h \\
&= \sum_{\theta\in\widetilde{\Theta}_h} \widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x, \boldsymbol{a}_{h,\theta}) + \sum_{\theta\in\widetilde{\Theta}_h} (\mu_\theta - \widehat{\mu}_{h,\theta}) u^{\mathrm{L}}(x, \boldsymbol{a}_{h,\theta}) + 3K\epsilon_h \\
&\leq \sum_{\theta\in\widetilde{\Theta}_h} \widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x, \boldsymbol{a}_{h,\theta}) + K\epsilon_h + 3K\epsilon_h = \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) + 4K\epsilon_h.
\end{aligned}$$

The first inequality follows from the fact that $\mu_\theta \leq 3\epsilon_h$ for $\theta \notin \widetilde{\Theta}_h$, while the second one applies $\|\mu - \widehat{\mu}_h\|_\infty \leq \epsilon_h$. Both properties are guaranteed by Condition 13. ■

Lemma 16 *If Condition 13 is satisfied, then Algorithm 4 computes* $\underline{\mathrm{OPT}}_h$ *such that:*

$$\mathrm{OPT} - K\epsilon_h(4 + C_2) \leq \underline{\mathrm{OPT}}_h \leq \mathrm{OPT} - K\epsilon(C_2 - 1),$$

where C_1 *and* C_2 *are computed at Line 1 in Algorithm 3.*

Proof Let $x^\star \in \mathcal{X}_h$ be an optimal commitment which, thanks to Condition 13, belongs to some $\mathcal{Y}_h(\boldsymbol{a}^\star), \boldsymbol{a}^\star \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ such that $\boldsymbol{a}^\star_\theta = a^\star_\theta(x^\star)$ for every $\theta \in \widetilde{\Theta}_h$. Then we lower bound $\underline{\mathrm{OPT}}_h$ as follows:

$$\begin{aligned}
\underline{\mathrm{OPT}}_h &= \max_{\substack{\boldsymbol{a}\in\mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h) \\ x\in\mathcal{Y}_h(\boldsymbol{a})}} \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}) - C_2 K\epsilon_h \\
&\geq \widehat{u}^{\mathrm{L}}_h(x^\star, \boldsymbol{a}^\star) - C_2 K\epsilon_h \\
&\geq u^{\mathrm{L}}(x^\star) - K\epsilon_h(4 + C_2),
\end{aligned}$$

where the first inequality holds by the $\max$ operator, and the last inequality by Lemma 15.

To provide an upper bound, let:

$$x^\circ, \boldsymbol{a}^\circ \in \operatorname*{arg\,max}_{\substack{\boldsymbol{a}\in\mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h) \\ x\in\mathcal{Y}_h(\boldsymbol{a})}} \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}).$$

Then:

$$\begin{aligned}
\underline{\mathrm{OPT}}_h &= \max_{\substack{\boldsymbol{a}\in\mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h) \\ x\in\mathcal{Y}_h(\boldsymbol{a})}} \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}) - C_2 K\epsilon_h \\
&= \widehat{u}^{\mathrm{L}}_h(x^\circ, \boldsymbol{a}^\circ) - C_2 K\epsilon_h
\end{aligned}$$

$$\leq u^{\mathrm{L}}(x^\circ) + K\epsilon_h - C_2 K\epsilon_h$$
$$\leq \mathrm{OPT} - K\epsilon_h(C_2 - 1),$$

where the first inequality comes from Lemma 15 and the last one by the optimality of OPT. ■

Lemma 17 *[Formal version of Lemma 4] If Condition 13 is satisfied, then Algorithm 4 computes a union of polytopes* $\mathcal{X}_{h+1} = \bigcup_{\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)} \mathcal{X}_{h+1}(\boldsymbol{a})$ *such that* $\mathcal{X}_{h+1}(\boldsymbol{a}) \subseteq \mathcal{P}(\boldsymbol{a})$ *and* $u^{\mathrm{L}}(x) \geq \mathrm{OPT} - K\epsilon_h(5 + C_1 + C_2)$ *for every* $x \in \mathcal{X}_{h+1}$*, where* C_1 *and* C_2 *are defined at Line 1 in Algorithm 4. Furthermore, there exists an optimal commitment* $x^\star$ *and an action profile* $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ *such that* $x^\star \in \mathcal{X}_{h+1}(\boldsymbol{a}_h)$ *and* $a^\star_\theta(x^\star) = \boldsymbol{a}_{h,\theta}$ *for every* $\theta \in \widetilde{\Theta}_h$*.*

Proof We observe that the regions $\mathcal{Y}_h(\boldsymbol{a})$ are subsets of $\mathcal{P}(\boldsymbol{a})$ by Equation (6), which is verified according to Condition 13. Therefore, the regions $\mathcal{X}_{h+1}(\boldsymbol{a}) \subseteq \mathcal{Y}_h(\boldsymbol{a})$ computed at Line 5 are subsets of $\mathcal{P}(\boldsymbol{a})$ themselves. To conclude the proof, we analyze the utility of the leader in the commitments $x \in \mathcal{X}_{h+1}$.

Consider a commitment $x \in \mathcal{X}_{h+1}(\boldsymbol{a}_h)$ for some $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$. As it is not empty, by construction $\mathcal{X}_{h+1}(\boldsymbol{a}_h) = \mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$, where $\mathcal{H}_h(\boldsymbol{a}_h)$ is the half-space defined by:

$$\widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) + C_1\epsilon \geq \underline{\mathrm{OPT}}_h.$$

Since $\underline{\mathrm{OPT}}_h \geq \mathrm{OPT} - K\epsilon_h(4 + C_2)$ by Lemma 16, we have:

$$\widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) \geq \mathrm{OPT} - K\epsilon_h(4 + C_1 + C_2).$$

Finally, we apply Lemma 15 to get:

$$u^{\mathrm{L}}(x) \geq \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) - K\epsilon_h \geq \mathrm{OPT} - K\epsilon_h(5 + C_1 + C_2),$$

proving the lower bound on the utility of every commitment $x \in \mathcal{X}_{h+1}$.

According to Condition 13, there exists an optimal commitment $x^\star$ belonging to some $\mathcal{Y}_h(\boldsymbol{a}_h)$, with $a^\star_\theta(x^\star) = \boldsymbol{a}_{h,\theta}$ for every $\theta \in \widetilde{\Theta}_h$. To conclude the proof, we show that $x^\star \in \mathcal{X}_{h+1}(\boldsymbol{a}_h)$ and that $\mathrm{vol}(\mathcal{X}_{h+1}(\boldsymbol{a}_h)) > 0$. By construction, Algorithm 4 computes:

$$\mathcal{X}_{h+1}(\boldsymbol{a}_h) = \begin{cases} \mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h) & \text{if its volume is greater than zero} \\ \emptyset & \text{otherwise,} \end{cases}$$

where $\mathcal{H}_h(\boldsymbol{a}_h)$ is the half-space computed at Line 4. We therefore have to show that $x^\star \in \mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$, *i.e.*, $x^\star \in \mathcal{H}_h(\boldsymbol{a}_h)$, and that $\mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$ has volume larger than zero. Let us observe that by the definition of $\mathcal{H}_h(\boldsymbol{a}_h)$ (Line 4), a point $x \in \mathcal{Y}_h(\boldsymbol{a}_h)$ belongs to $\mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$ if

$$\widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) + C_1\epsilon_h \geq \underline{\mathrm{OPT}}_h,$$

where $\underline{\mathrm{OPT}}_h \leq \mathrm{OPT} - K\epsilon_h(C_2 - 1)$ according to Lemma 16. Therefore, for x to belong to $\mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$ is sufficient that:

$$\widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}_h) \geq \mathrm{OPT} - K\epsilon_h(C_1 + C_2 - 1) \tag{13}$$

We first show that $x^\star \in \mathcal{H}_h(\boldsymbol{a}_h)$. By Lemma 15, we have $\widehat{u}^{\mathrm{L}}(x^\star, \boldsymbol{a}_h) \geq \mathrm{OPT} - 4K\epsilon_h$. Therefore, Equation (13) is satisfied and $x^\star \in \mathcal{H}_h(\boldsymbol{a}_h)$ as long as $C_1 + C_2 \geq 5$.

In order to complete the proof, we have to show that $\mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$ has volume larger than zero. To do so, by Lemma 28 it is sufficient to find a commitment $x^\circ \in \mathcal{H}_h(\boldsymbol{a}_h) \cap \mathrm{int}(\mathcal{Y}_h(\boldsymbol{a}_h))$.[3] If $x^\star \in \mathrm{int}(\mathcal{Y}_h(\boldsymbol{a}_h))$, this is satisfied. Suppose instead that $x^\star \in \partial\mathcal{Y}_h(\boldsymbol{a}_h)$. Recall that since $\mathcal{Y}_h(\boldsymbol{a}_h)$ is non-empty, it also has non-zero volume (Equation (6) holds by Condition 13). Therefore, there exists some $x' \in \mathrm{int}(\mathcal{Y}_h(\boldsymbol{a}_h))$. Let

$$y := \max(\widehat{u}^{\mathrm{L}}_h(x', \boldsymbol{a}_h), \mathrm{OPT} - 4K\epsilon_h).$$

By Lemma 27, there exists a point x° in the segment between $x^\star$ and x' with estimated utility $\widehat{u}^{\mathrm{L}}(x^\circ, \boldsymbol{a}_h) = y$. This point is also different from $x^\star$ (if $\widehat{u}^{\mathrm{L}}(x^\star, \boldsymbol{a}_h) = y$, by Lemma 27 we can take any other point on the segment). As a result, it must be an interior point of $\mathcal{Y}_h(\boldsymbol{a}_h)$. At the same time, x° satisfies Equation (13), and thus belongs to $\mathcal{H}_h(\boldsymbol{a}_h)$. By Lemma 28 the polytope $\mathcal{Y}_h(\boldsymbol{a}_h) \cap \mathcal{H}_h(\boldsymbol{a}_h)$ has non-zero volume, concluding the proof. ■

Lemma 18 *Suppose that Condition 7 is satisfied for two successive epochs $h-1, h$, and that $\widetilde{\Theta}_h = \widetilde{\Theta}_{h-1} =: \widetilde{\Theta}$. Let $\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta})$ and $\mathcal{H}_{h-1}(\boldsymbol{a}), \mathcal{H}_h(\boldsymbol{a})$ be the half-spaces computed at Line 4 when Algorithm 4 is executed at epochs $h-1$ and h, respectively. Then $\mathcal{H}_h(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}}) \subseteq \mathcal{H}_{h-1}(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}})$.*

Proof Fix some $\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta})$ such that $\mathcal{X}_{h+1}(\boldsymbol{a}) \neq \emptyset$. Observe that we drop the subscript h from $\boldsymbol{a}$, as $\widetilde{\Theta} = \widetilde{\Theta}_h = \widetilde{\Theta}_{h+1}$.

In order to prove the statement, we show that:

$$\Delta(\mathcal{A}_{\mathrm{L}}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}) \cap \mathcal{H}_h(\boldsymbol{a}) \supseteq \Delta(\mathcal{A}_{\mathrm{L}}) \cap \mathcal{H}_h(\boldsymbol{a}). \tag{14}$$

Take any $x \in \Delta(\mathcal{A}_{\mathrm{L}}) \cap \mathcal{H}_h(\boldsymbol{a})$. To prove Equation (14), we need to show that $x \in \mathcal{H}_{h-1}(\boldsymbol{a})$, that is:

$$U_{h-1} := \widehat{u}^{\mathrm{L}}_{h-1}(x, \boldsymbol{a}) + 2C_1\epsilon_h \geq \underline{\mathrm{OPT}}_{h-1},$$

where we considered that $\epsilon_{h-1} = 2\epsilon_h$. As $x \in \mathcal{H}_h(\boldsymbol{a})$, it holds that:

$$U_h := \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}) + C_1\epsilon_h \geq \underline{\mathrm{OPT}}_h.$$

Therefore, we can prove Equation (14) by showing that $U_h \geq U_{h-1}$ and $\underline{\mathrm{OPT}}_h \leq \underline{\mathrm{OPT}}_{h-1}$. Employing Lemma 16 to epochs h and $h-1$ we get:

$$\underline{\mathrm{OPT}}_h \geq \mathrm{OPT} - K\epsilon_h(4 + C_2)$$
$$\underline{\mathrm{OPT}}_{h-1} \leq \mathrm{OPT} - K\epsilon_h(2C_2 - 2).$$

By taking $C_2 \geq 6$ we get $\underline{\mathrm{OPT}}_h \leq \underline{\mathrm{OPT}}_{h-1}$. At the same time, we can bound U_h and U_{h-1} by means of Lemma 14. Observe that this lemma holds for every $x \in \Delta(\mathcal{A}_{\mathrm{L}})$. Let $\alpha := \sum_{\theta \in \widetilde{\Theta}} \mu_\theta u^{\mathrm{L}}(x, \boldsymbol{a}_\theta) \geq 0$. We have:

$$U_h = \widehat{u}^{\mathrm{L}}_h(x, \boldsymbol{a}) + C_1\epsilon_h \leq \alpha + K\epsilon_h + C_1\epsilon_h = \alpha + K\epsilon_h(C_1 + 1)$$

3. We let $\mathrm{int}(\mathcal{P}) := \mathcal{P} \setminus \partial\mathcal{P}$ be the interior of any given polytope $\mathcal{P}$.

$$U_{h-1} = \widehat{u}^{\mathrm{L}}_{h-1}(x, \boldsymbol{a}) + 2C_1\epsilon_h \geq \alpha - 2K\epsilon_h + 2C_1\epsilon_h = \alpha + K\epsilon_h(2C_1 - 2)$$

where we leverage the fact that $\epsilon_{h-1} = 2\epsilon_h$. By taking $C_1 \geq 3$, we have $U_h \leq U_{h-1}$. As a result, Equation (14) holds when $C_1 \geq 3$ and $C_2 \geq 6$, proving the statement. ■

Appendix E. Proofs of the regret bound

We first define the following event.

Definition 19 *We let $\mathcal{E}_h$ be the event under which, for every epoch $h' \leq h$, Condition 7 is verified when Algorithm 3 is executed, and Condition 13 is verified when Algorithm 4 is executed. Furthermore, Algorithm 3 is executed in the number of rounds specified in Lemma 9.*

This event corresponds to succesffull execution of Algoirthm 1. In the following we introduce some intermediate lemmas to bound the probability of even $\mathcal{E}_H$ (Lemma 20 and Lemma 21), the number of epochs (Lemma 22), the number of facets of each polytope $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ during the execution of the algorithm (Lemma 23 and Lemma 24), and finally the bit complexity required to represent these facets (Lemma 25).

Subsequently, we prove Theorem 26, which is the formal version of Theorem 5 with the explicit upper bound on the regret.

Lemma 20 *The probability of event $\mathcal{E}_1$ is at least $1 - \delta_1 - \delta_2$.*

Proof We first observe that

$$\Delta(\mathcal{A}_{\mathrm{L}}) = \mathcal{X}_1 = \bigcup_{\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_0)} \mathcal{X}_1(\boldsymbol{a}),$$

where $\mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_0) \coloneqq \{\perp\}$ and $\mathcal{X}_1(\perp) = \mathcal{P}(\perp) = \Delta(\mathcal{A}_{\mathrm{L}})$. Consequently, the search space satisfies the requirements of Condition 7 and Condition 13. The set $\widetilde{\Theta}_1 \subseteq \Theta$ and the estimator $\widehat{\mu}_1$ are computed by Algorithm 2. As of Lemma 2, with probability at least $1 - \delta_1$ both $\widetilde{\Theta}_1$ and $\widehat{\mu}_1$ are computed according to Condition 7 and Condition 13. Observe that $\widetilde{\Theta}_1 \neq \emptyset$, as at least one type appears with probability at least $\epsilon_1 = 1/K$. Putting all together, Condition 7 holds with probability at least $1 - \delta_1$.

We can now apply Lemma 9 and a union bound, proving that $\mathcal{Y}_1$ satisfies Equation (6) and all the properties above hold. To conclude the proof, we need to show that there exists some $\boldsymbol{a}_1 \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_1)$ such that $x^\star \in \mathcal{Y}_1(\boldsymbol{a}_1)$ and $a^\star_\theta(x^\star) = \boldsymbol{a}_{1,\theta}$ for every $\theta \in \widetilde{\Theta}_1$, which implies that Condition 13 is satisfied. Let $\boldsymbol{a}^\star \coloneqq (a^\star_\theta(x^\star))_{\theta \in \widetilde{\Theta}_1}$ and $\boldsymbol{a}_1 \coloneqq \boldsymbol{a}^\star | \widetilde{\Theta}_1$. It suffices to show that $x^\star \in \mathcal{Y}_1(\boldsymbol{a}_1)$. We recall that $\mathrm{vol}(\mathcal{P}(\boldsymbol{a}^\star)) > 0$. Therefore, $\mathrm{vol}(\mathcal{P}(\boldsymbol{a}_1)) > 0$ and $x^\star \in \mathcal{P}(\boldsymbol{a}_1)$, as $\mathcal{P}(\boldsymbol{a}_1) \supseteq \mathcal{P}(\boldsymbol{a}^\star)$. By Equation (6), we have:

$$\mathcal{Y}_1(\boldsymbol{a}_1) = \mathcal{P}(\boldsymbol{a}_1) \cap \mathcal{X}_1(\boldsymbol{a}_1 | \widetilde{\Theta}_0) = \mathcal{P}(\boldsymbol{a}_1) \cap \Delta(\mathcal{A}_{\mathrm{L}}) = \mathcal{P}(\boldsymbol{a}_1).$$

As a result, $x^\star \in \mathcal{Y}_1(\boldsymbol{a}_1)$, concluding the proof. ■

Lemma 21 *With probability at least $1 - H(\delta_1 + \delta_2)$, the event $\mathcal{E}_H$ holds, where H is the number of epochs of Algorithm 1.*

Proof We prove by induction that for every epoch $h \in \{1, \dots, H\}$, it holds

$$\mathbb{P}(\mathcal{E}_h \mid \mathcal{E}_{h-1}) \geq 1 - \delta_1 - \delta_2,$$

where we define $\mathbb{P}(\mathcal{E}_0) \coloneqq 0$.

The base step $h = 1$ is proved by Lemma 20. Now suppose that we are under the event $\mathcal{E}_{h-1}$ for any $2 \leq h \leq H$. For the sake of explanation, suppose that the epoch is completed without reaching T rounds.

We now show that with probability at least $1 - \delta_1$, the set $\widetilde{\Theta}_h$ and the estimator $\widehat{\mu}_h$ satisfy the constraints imposed by Condition 7 and Condition 13. By Lemma 2, the estimator satisfies $\|\mu - \widehat{\mu}_h\|_\infty \leq \epsilon_h$, and the set of types $\bar{\Theta}_h$ is such that $\mu_\theta \geq \epsilon_h$ for each $\theta \in \bar{\Theta}_h$ and $\mu_\theta \leq 3\epsilon_h$ for each $\theta \in \Theta \setminus \bar{\Theta}_h$. Algorithm 1 computes $\widetilde{\Theta}_h \coloneqq \widetilde{\Theta}_{h-1} \cap \bar{\Theta}_h$. By the inductive hypothesis, $\mu_\theta \geq \epsilon_{h-1} \geq \epsilon_h$ for every $\theta \in \widetilde{\Theta}_{h-1}$, hence $\mu_\theta \geq \epsilon_h$ for every $\theta \in \widetilde{\Theta}_h$. Moreover, every $\theta \notin \widetilde{\Theta}_h$ appears with probability at most $3\epsilon_h$, as $\theta \notin \bar{\Theta}_h$ by construction. Finally, $\widetilde{\Theta}_h \supset \widetilde{\Theta}_{h-1}$ is non-empty by the inductive hypothesis. Therefore, both the set of types and the estimator are computed correctly with probability at least $1 - \delta_1$. By Lemma 9 and a union bound, we also have that $\mathcal{Y}$ satisfies Equation (6) with probability at lest $1 - \delta_1 - \delta_2$.

We now observe that by Lemma 17 applied to the previous epoch, the search space $\mathcal{X}_h$ is a union of polytopes as required by Condition 7 and Condition 13. Furthermore, combining Equation (6) with the result provided by Lemma 17, one can verify that an optimal commitment belongs $\mathcal{X}_h$ as required by Condition 13. As a result, with probability $1 - \delta_1 - \delta_2$ both properties hold before the respective algorithms are executed. Hence, $\mathbb{P}(\mathcal{E}_h \mid \mathcal{E}_{h-1}) \geq 1 - \delta_1 - \delta_2$ for every epoch $h \in \{1, \dots, H\}$. A recursive argument completes the proof. ■

Lemma 22 *The largest epoch index $H \in \mathbb{N}$ executed by Algorithm 1 satisfies $H \leq H' \coloneqq \log_4(5T)$.*

Proof We observe that to complete epoch $h \in \{1, \dots, H\}$, Algorithm 1 employs at least $1/\epsilon_h^2$ rounds (see Algorithm 2). Since $\epsilon_h = 1/(K2^{h-1})$, we have that:

$$\sum_{h=1}^{H-1} \frac{1}{\epsilon_h^2} = \sum_{h=1}^{H-1} (K2^{h-1})^2 \leq T,$$

as the rounds to complete $H - 1$ epochs cannot exceed T. We do not count epoch H as that the algorithm may We then observe that:

$$T \geq \sum_{h=1}^{H-1} (K2^{h-1})^2 = K^2 \sum_{h=1}^{H-1} 4^{h-1} = K^2 \sum_{h=0}^{H-2} 4^h = K^2 \frac{1 - 4^{H-1}}{1 - 4} = K^2 \frac{4^{H-1} - 1}{3}.$$

As a result, we have:

$$H \leq \log_4 \left(\frac{3T}{K^2} + 1 \right) + 1,$$

concluding the proof. ■

To bound the regret of Algorithm, we need to upper bound the number of facets of the polytopes $\mathcal{X}_h(\boldsymbol{a}_{h-1})$, in order to apply Lemma 9. We define:

$$\Psi := \{\bar{h} \in [H] \mid \widetilde{\Theta}_{\bar{h}} \neq \widetilde{\Theta}_{\bar{h}-1}\} \cup \{H+1\},$$

where H is the number of epochs. In the following, for the sake of the analysis, we will assume that the last epoch is completed (otherwise it is sufficient to consider a fictitious $\mathcal{X}_{H+1}$ computed as if the algorithm did not terminate after T rounds). We also observe (see Algorithm 4 Line 5) that under the event $\mathcal{E}_H$, for every $h \in [H]$ and $\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$ such that $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \neq \emptyset$, we have:

$$\mathcal{X}_h(\boldsymbol{a}_{h-1}) = \mathcal{Y}_{h-1}(\boldsymbol{a}_{h-1}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_{h-1}), \tag{15}$$

where $\mathcal{H}_{h-1}(\boldsymbol{a}_{h-1})$ is a half-space.

Lemma 23 *Let $\bar{h}, \bar{h}'$ be two successive values in Ψ. For every $h \in \{\bar{h}+1, \dots, \bar{h}'\}$ and $\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{\bar{h}}) = \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$, it holds that $\mathcal{X}_h(\boldsymbol{a}) = \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a})$, unless it is empty.*

Proof The statement is trivially satisfied when $\bar{h}+1 = \bar{h}' = h$ (see Line 5 Algorithm 4). We therefore assume that

$$\bar{h}' \geq \bar{h} + 2.$$

Let $\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{\bar{h}})$. For every $h \in \{\bar{h}+1, \dots, \bar{h}'\}$ such that $\mathcal{X}_h(\boldsymbol{a}) \neq \emptyset$, Algorithm 4 at Line 5 computes:

$$\mathcal{X}_h(\boldsymbol{a}) = \mathcal{Y}_{h-1}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}). \tag{16}$$

Furthermore, for every $h \in \{\bar{h}+1, \dots, \bar{h}'-1\}$, we can employ Lemma 18 as follows:

$$\mathcal{H}_h(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}}) \subseteq \mathcal{H}_{h-1}(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}}). \tag{17}$$

If $\mathcal{X}_h(\boldsymbol{a}) \neq \emptyset$, we can leverage Equation (6) to get:

$$\mathcal{Y}_h(\boldsymbol{a}) = \mathcal{P}(\boldsymbol{a}) \cap \mathcal{X}_h(\boldsymbol{a}|\widetilde{\Theta}_{h-1}) = \mathcal{P}(\boldsymbol{a}) \cap \mathcal{X}_h(\boldsymbol{a}) = \mathcal{X}_h(\boldsymbol{a}), \tag{18}$$

where the second equality exploit the fact that $\widetilde{\Theta}_{h-1} = \widetilde{\Theta}_h$, and the last equality holds because $\mathcal{X}_h(\boldsymbol{a}) \subseteq \mathcal{P}(\boldsymbol{a})$ under $\mathcal{E}_h$.

We prove by induction that for every $h \in \{\bar{h}+1, \dots, \bar{h}'\}$ and $\boldsymbol{a} \in \widetilde{\Theta}_{\bar{h}}$ such that $\mathcal{X}_h(\boldsymbol{a}) \neq \emptyset$, it holds that $\mathcal{X}_h(\boldsymbol{a}) = \mathcal{Y}_{\bar{h}} \cap \mathcal{H}_{h-1}(\boldsymbol{a})$. The base step is $h = \bar{h}+1$. Here we have:

$$\mathcal{X}_h(\boldsymbol{a}) = \mathcal{Y}_{h-1}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}) = \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}),$$

where we employed Equation (16) and the equality $h-1 = \bar{h}+1-1 = \bar{h}$. Now consider any $h \in \{\bar{h}+2, \dots, \bar{h}'\}$, and assume that $\mathcal{X}_{h-1}(\boldsymbol{a}) = \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap \mathcal{H}_{h-2}(\boldsymbol{a})$. We have:

$$\begin{aligned}
\mathcal{X}_h(\boldsymbol{a}) &= \mathcal{Y}_{h-1}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}) \\
&= \mathcal{X}_{h-1}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}) \\
&= \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap \mathcal{H}_{h-2}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}) \\
&= \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap (\mathcal{H}_{h-2}(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}})) \cap (\mathcal{H}_{h-1}(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}})) \\
&= \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}) \cap \Delta(\mathcal{A}_{\mathrm{L}}) = \mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}),
\end{aligned}$$

where we employed Equation (16), Equation (18), the inductive step, the identity $\mathcal{Y}_{\bar{h}}(\boldsymbol{a}) \subseteq \Delta(\mathcal{A}_{\mathrm{L}})$, and Equation (17). Therefore the inductive step holds for h, concluding the proof. ■

Lemma 24 *Let $h \in \{2, \dots, H\}$ and $\bar{h} = \max\{h' \in \Psi \mid h' < h\}$. Under the event $\mathcal{E}_H$, every non-empty polytope $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ composing $\mathcal{X}_h$ is defined as:*

$$\mathcal{X}_h(\boldsymbol{a}_{h-1}) = \mathcal{P}(\boldsymbol{a}_{\bar{h}}) \cap \bigcap_{\bar{h}' \in \Psi : 1 < \bar{h}' < \bar{h}} \mathcal{H}_{\bar{h}'-1}(\boldsymbol{a}_{\bar{h}'}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_{h-1}),$$

where $\boldsymbol{a}_{\bar{h}'} = \boldsymbol{a}_{h-1} | \widetilde{\Theta}_{\bar{h}'}$.

Proof In the following we let $\bar{h}^\diamond := \min \Psi \setminus \{1\}$.

Let $\bar{h} \in \Psi \setminus \{1, H+1\}$ and $\boldsymbol{a}_{\bar{h}} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{\bar{h}})$ such that $\mathcal{X}_{\bar{h}}(\boldsymbol{a}_{\bar{h}}) \neq \emptyset$. We also let $\bar{h}'$ the previous value in Ψ. Thanks to Equation (6) and Lemma 23, we have:

$$\mathcal{Y}_{\bar{h}}(\boldsymbol{a}_{\bar{h}}) = \mathcal{P}(\boldsymbol{a}_{\bar{h}}) \cap \mathcal{X}_{\bar{h}}(\boldsymbol{a}_{\bar{h}-1}) = \mathcal{P}(\boldsymbol{a}_{\bar{h}}) \cap \mathcal{Y}_{\bar{h}'}(\boldsymbol{a}_{\bar{h}'}) \cap \mathcal{H}_{\bar{h}-1}(\boldsymbol{a}_{\bar{h}'}).$$

Recursively applying this step to $\mathcal{Y}_{\bar{h}'}(\boldsymbol{a}_{\bar{h}'})$ till reaching epoch 1 and considering that $\mathcal{P}(\boldsymbol{a}_{\bar{h}''}) \subseteq \mathcal{P}(\boldsymbol{a}_{\bar{h}})$ for $\bar{h}'' \leq \bar{h}$ and $\mathcal{Y}_1(\boldsymbol{a}_1) = \mathcal{P}(\boldsymbol{a}_1)$, we get:

$$\mathcal{Y}_{\bar{h}}(\boldsymbol{a}_{\bar{h}}) = \mathcal{P}(\boldsymbol{a}_{\bar{h}}) \cap \bigcap_{\bar{h}' \in \Psi : 1 < \bar{h}' < \bar{h}} \mathcal{H}_{\bar{h}'-1}(a_{\bar{h}'}). \tag{19}$$

Now consider an epoch $h \in \{\bar{h}^\diamond + 1, \dots, H\}$ and an action profile $\boldsymbol{a}_{h-1} \in \widetilde{\Theta}_{h-1}$ such that $\mathcal{X}_h(\boldsymbol{a}_{h-1}) \neq \emptyset$. There exists $\bar{h} = \max\{\bar{h}' \in \Psi \mid \bar{h}' < h\}$ different from one. We can therefore employ Lemma 23 and Equation (19) to get:

$$\begin{aligned}\mathcal{X}_h(\boldsymbol{a}_{h-1}) &= \mathcal{Y}_{\bar{h}}(\boldsymbol{a}_{\bar{h}}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_{h-1}) \\ &= \mathcal{P}(\boldsymbol{a}_{\bar{h}}) \cap \bigcap_{\bar{h}' \in \Psi : 1 < \bar{h}' < \bar{h}} \mathcal{H}_{\bar{h}'-1}(\boldsymbol{a}_{\bar{h}'}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_h).\end{aligned}$$

Consider instead an epoch $h \in \{2, \dots, \bar{h}^\diamond\}$ and let $\bar{h} = 1$. By employing Lemma 23 and Equation (6) we get:

$$\begin{aligned}\mathcal{X}_h(\boldsymbol{a}_{h-1}) &= \mathcal{Y}_1(\boldsymbol{a}_{h-1}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_{h-1}) \\ &= \mathcal{P}(\boldsymbol{a}_{h-1}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_{h-1}),\end{aligned}$$

concluding the proof. ■

Lemma 25 *Under the event $\mathcal{E}_H$, whenever Algorithm 3 is executed, every $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ has at most $N \leq Kn + m + K$ facets. Furthermore, the coefficients of the hyperplanes defining these facets can be encoded by at most $\mathcal{O}(L + \log(T) + \log(K) + B_\delta)$ bits, where B_δ is the bit complexity of the parameter δ given in input to Algorithm 1.*

Proof For the sake of the analysis, we will assume that the last epoch is completed (otherwise it is sufficient to consider a fictitious $\mathcal{X}_{H+1}$). Given any $\boldsymbol{a}_h \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)$ for some epoch h, we let $\boldsymbol{a}_{h'} = \boldsymbol{a}_h | \widetilde{\Theta}_{h'}$ for every $0 \leq h' < h$.

We let N_h^{X} and N_h^{Y} be the maximum number of facets of any region $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ and $\mathcal{Y}_h(\boldsymbol{a}_h)$ at epoch h, respectively. To bound the number of facets of a polytope, we will bound the number of

half-spaces defining it. Notice that every polytope that we consider is a subset of the hyperplane containing $\Delta(\mathcal{A}_{\mathrm{L}})$. This hyperplane will not be considered, as it does not define a facet.

Let $h \in \{2, \dots, H\}$ and $\bar{h} = \max\{h' \in \Psi \mid h' < h\}$. Consider a non-empty polytope $\mathcal{X}_h(\boldsymbol{a}_{h-1})$ with $\boldsymbol{a}_{h-1} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_{h-1})$. By Lemma 24 we have:

$$\mathcal{X}_h(\boldsymbol{a}_{h-1}) = \mathcal{P}(\boldsymbol{a}_{\bar{h}}) \cap \bigcap_{\bar{h}' \in \Psi : 1 < \bar{h}' < \bar{h}} \mathcal{H}_{\bar{h}'-1}(\boldsymbol{a}_{\bar{h}'}) \cap \mathcal{H}_{h-1}(\boldsymbol{a}_{h-1}).$$

We observe that $\mathcal{P}(\boldsymbol{a}_{\bar{h}})$ has at most $|\widetilde{\Theta}_{\bar{h}}| n + m$ facets. Furthermore, $|\Psi \setminus \{1, H+1\}| \leq K - 1$. The single epoch we did not consider was the first one. Since $\mathcal{X}_1 = \Delta(\mathcal{A}_{\mathrm{L}})$ has m facets, the number of facets is at most $Kn + m + K$ for every epoch $h \in [1, H]$.

To conclude the proof, we need to upper bound the number B of bits required to encode the coefficients of the hyperplanes defying the facets of any $\mathcal{X}_h(\boldsymbol{a}_h)$. For the facets of the region $\mathcal{P}(\boldsymbol{a}_{\bar{h}})$, these coefficients have at most L bits. The hyperplanes $\mathcal{H}_h(\boldsymbol{a}_h)$ are instead defined by the inequality:

$$\sum_{\theta \in \widetilde{\Theta}_h} \widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x, \boldsymbol{a}_\theta) + C_1 K \epsilon_h \geq \max_{\boldsymbol{a} \in \mathcal{A}_{\mathrm{F}}(\widetilde{\Theta}_h)} \max_{x \in \mathcal{Y}_h(\boldsymbol{a})} \sum_{\theta \in \widetilde{\Theta}_h} \widehat{\mu}_{h,\theta} u^{\mathrm{L}}(x, \boldsymbol{a}_\theta) - C_2 K \epsilon_h,$$

where $\epsilon_h = 1/(K 2^{h-1})$ and $h \leq H \leq \log_4(5T)$ as of Lemma 22. Therefore, ϵ_h can be represented by

$$\mathcal{O}(\log(T) + \log(K))$$

bits. Similarly, $CK\epsilon_h$ can be represented by $\mathcal{O}(\log(T) + \log(K))$ bits for any constant C. The prior estimator $\widehat{\mu}_h$ has instead been computed by Algorithm 2 as the empirical estimator of μ using $\mathcal{O}(1/\epsilon_h^2 \log(K/\delta_1))$ samples, where δ_1 defined at Line 2 Algorithm 1 has bit complexity bounded by $\mathcal{O}(\log(\log(T)) + B_\delta)$. Therefore, each component of $\widehat{\mu}_h$ can be represented by at most

$$\mathcal{O}(\log(T) + \log(K) + B_\delta).$$

Overall, each coefficient of the hyperplane can be encoded by at most $\mathcal{O}(L + \log(T) + \log(K) + B_\delta)$, accounting for L bits to represent the leader utility. As a result, $B \leq \mathcal{O}(L + \log(T) + \log(K) + B_\delta)$. ■

Theorem 26 *With probability at least $1 - \delta$, the regret of Algorithm 1 is:*

$$R_T \leq \widetilde{\mathcal{O}}(K^2 \log\left(\frac{K}{\delta}\right) \sqrt{T} + \beta \log^2(T)),$$

where:

$$\beta \coloneqq K^{m+2} n^{2m+2} \left(m^7 (L + B_\delta) \log\left(\frac{1}{\delta}\right) + (Kn + m)^m \right)$$

is a time-independent function of the instance size and δ.

Proof By Lemma 21 the event the event $\mathcal{E}_H$ happens with probability at least $1 - H(\delta_1 + \delta_2)$. We also have $H \leq \log_4(3T/K^2 + 1) + 1 \leq \log_4(5T)$ by Lemma 22. By taking

$$\delta_1 = \delta_2 = \frac{\delta}{2 \lceil \log_4(5T) \rceil},$$

the clean event holds with probability at least $1 - \delta$. Let us observe that:

$$\frac{1}{\delta'} = \mathcal{O}\left(\frac{1}{\delta}\log(T)\right).$$

Now we bound the regret of Algorithm 1 under the event $\mathcal{E}_H$. We will let $B \coloneqq L + B_\delta$ and $N \coloneqq Kn + m + K$. We also define the following two quantities:

$$\alpha_1 \coloneqq \log\left(\frac{K}{\delta}\right)$$
$$\alpha_2 \coloneqq K^{m+1}n^{2m+2}\left(m^7 B \log\left(\frac{1}{\delta}\right) + \binom{N+n}{m}\right).$$

Consider an epoch $h \in \{1, \dots, H\}$. We let R_h the regret accumulated during epoch h. The number of rounds of epoch h is bounded by $T_h \coloneqq T_{h,1} + T_{h,2}$, where $T_{h,1}$ and $T_{h,2}$ are the number of rounds to execute Algorithm 2 and Algorithm 3, respectively. By Lemma 2 we have:

$$T_{h,1} \leq \frac{1}{\epsilon_h^2}\log\left(\frac{K}{\delta'}\right) = \mathcal{O}\left(\frac{1}{\epsilon_h^2}\alpha_1 \log(T)\right)$$

while by Lemma 9 and Lemma 25 we have:

$$\begin{aligned}
T_{h,2} &\leq \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h}K^{m+1}n^{2m+2}\left(m^7(B+\log(T))\log\frac{1}{\delta'} + \binom{N+n}{m}\right)\right) \\
&= \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h}K^{m+1}n^{2m+2}\log(T)\left(m^7(B+\log(T))\log\frac{1}{\delta} + \binom{N+n}{m}\right)\right) \\
&= \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h}K^{m+1}n^{2m+2}\log^2(T)\left(m^7 B\log\frac{1}{\delta} + \binom{N+n}{m}\right)\right) \\
&= \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_h}\alpha_2\log^2(T)\right).
\end{aligned}$$

We will divide the epochs in three intervals. First, we consider the single epoch $h = 1$. Then, we will consider the epochs from $h = 2$ to $h^\diamond := \min\{H, \lceil \frac{1}{2}\log(T)\rceil\}$. Finally, the epochs from $h = h^\diamond + 1$ to H.

During the first epoch, the regret at each round is at most one. We can thus bound R_1 as:

$$\begin{aligned}
R_1 &\leq R_{1,1} + R_{1,2} \\
&\leq \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon_1^2}\alpha_1\log(T) + \frac{1}{\epsilon_1}\alpha_2\log^2(T)\right) \\
&= \widetilde{\mathcal{O}}\left(K^2\alpha_1\log(T) + K\alpha_2\log^2(T)\right).
\end{aligned}$$

Consider now an epoch $h \in \{2, \dots, h^\diamond\}$. By Lemma 17 applied to the previous epoch, the regret of each round during this epoch is at most $14K\epsilon_{h-1} = 28K\epsilon_h$, as $\epsilon_h = \frac{1}{K2^{h-1}}$. Therefore:

$$\sum_{h=2}^{h^\diamond} R_h \leq \mathcal{O}\left(K\epsilon_h(T_{h,1} + T_{h,2})\right)$$

$$
\begin{aligned}
&\leq \widetilde{\mathcal{O}}\left(\sum_{h=2}^{h^\diamond} K\epsilon_h \left(\frac{1}{\epsilon_h^2}\alpha_1 + \frac{1}{\epsilon_h}\alpha_2\right)\log^2(T)\right)\\
&= \widetilde{\mathcal{O}}\left(\sum_{h=2}^{h^\diamond} K \left(\frac{1}{\epsilon_h}\alpha_1 + \alpha_2\right)\log^2(T)\right)\\
&= \widetilde{\mathcal{O}}\left(\sum_{h=2}^{h^\diamond} K \left(K2^{h-1}\alpha_1 + \alpha_2\right)\log^2(T)\right)\\
&= \widetilde{\mathcal{O}}\left(K^2\log^2(T)\alpha_1\sum_{h=2}^{h^\diamond} 2^{h-1} + \alpha_2\log^2(T)\right)\\
&= \widetilde{\mathcal{O}}\left(\log^2(T)K^2\alpha_1 2^{h^\diamond} + \alpha_2\log^2(T)\right)\\
&= \widetilde{\mathcal{O}}\left(K^2\alpha_1\sqrt{T} + \alpha_2\log^2(T)\right).
\end{aligned}
$$

Finally, we consider an epoch $h \in \{h^\diamond + 1, \dots, H\}$. We can again employ Lemma 17 to the previous epoch. This provides us an upper on the per-round regret of:

$$
14K\epsilon_{h-1} = \frac{28K}{K2^h} \leq \frac{28}{2^{h^\diamond}} = \frac{28}{2^{\frac{\log(T)}{2}}} = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right).
$$

By considering that the epochs from $h^\diamond + 1$ to H can take at most T rounds, we get:

$$
\sum_{h=h^\diamond+1}^{H} R_h \leq T \cdot \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) = \mathcal{O}(\sqrt{T}).
$$

Putting all together, with probability at least $1-\delta$ the regret of Algorithm 1 is:

$$
\begin{aligned}
R_T &\leq \widetilde{\mathcal{O}}\left(K^2\alpha_1\log(T) + K\alpha_2\log^2(T) + K^2\alpha_1\sqrt{T} + \alpha_2\log^2(T) + \sqrt{T}\right)\\
&= \widetilde{\mathcal{O}}(K^2\alpha_1\sqrt{T} + K\alpha_2\log^2(T)).
\end{aligned}
$$

The proof os concluded by observing that $\binom{N+n}{m} \leq (N+n)^m$ and performing simple computations. ■

Appendix F. Technical Lemmas

Lemma 27 *Let $\mathcal{P} \subseteq \Delta(\mathcal{A}_{\text{L}})$ be a polytope and $x_1, x_2 \in \mathcal{P}$. Let also $u : \mathcal{P} \to [0,1]$ be an affine linear function, with $u(x_1) \leq u(x_2)$. Then for every $y \in [u(x_1), u(x_2)]$ there exists some $x_y \in \mathcal{P}$ belonging to the segment between x_1 and x_2 such that $u(x_y) = y$. When $u(x_1) = u(x_2)$, then $u(x') = u(x_1)$ for every x' in the segment between x_1 and x_2.*

Proof Suppose $u(x_1) < u(x_2)$ and consider a generic point x_λ belonging to the segment between x_1 and x_2 and parametrized by $\lambda \in [0,1]$ as:

$$
x_\lambda \coloneqq x_2 + (x_1 - x_2)\lambda.
$$

This point has utility $u(x_\lambda) = y$ when:

$$u(x_\lambda) = u(x_2) + \lambda(u(x_1) - u(x_2)) = y,$$

that is:

$$\lambda = \frac{y - u(x_2)}{u(x_1) - u(x_2)}.$$

It easy to say that when $y \in [u(x_1), u(x_2)]$, λ belongs to $[0, 1]$. By convexity, it also holds that $x_\lambda \in \mathcal{P}$.

Suppose now $u(x_1) = u(x_2)$. Then $u(x_\lambda) = u(x_1) + \lambda(u(x_1) - u(x_1)) = u(x_1)$ for every $\lambda \in [0, 1]$, concluding the proof. ■

Lemma 28 *Let $\mathcal{P} \subseteq \Delta(\mathcal{A}_{\mathrm{L}})$ be a polytope with $\mathrm{vol}(\mathcal{P}) > 0$, and let $x \in \mathrm{int}(\mathcal{P})$, where volume and interior are relative to the hyperplane containing $\Delta(\mathcal{A}_{\mathrm{L}})$. Then, if $x \in \mathcal{H}$ for some half-space $\mathcal{H}$, it holds that $\mathrm{vol}(\mathcal{P} \cap \mathcal{H}) > 0$.*

Proof Let H_Δ be the hyperplane containing $\Delta(\mathcal{A}_{\mathrm{L}})$. We observe that if $H_\Delta \subseteq \mathcal{H}$, then the statement is trivially satisfied, as $\mathcal{P} \cap \mathcal{H} = \mathcal{P}$ has non-zero volume. We thus suppose that $H_\Delta \not\subseteq \mathcal{H}$. Since $x \in \mathrm{int}(\mathcal{P})$, there exists some sphere

$$\mathcal{B}_\epsilon(x) = \{x' \in \mathbb{R}^m \mid \|x' - x\|_2 \leq \epsilon\}$$

of radius $\epsilon > 0$ such that $\mathcal{B}_\epsilon(x) \cap H_\Delta \subseteq \mathcal{P}$. There must exist some point x° at distance $0 < \epsilon_{\text{small}} \leq \epsilon$ from x that belongs to both H_Δ and $\mathrm{int}(\mathcal{H})$. Now take any $x' \in \mathbb{R}^m$ such that $\|x^\circ - x'\|_2 \leq \epsilon^\circ$, with $0 < \epsilon^\circ < \epsilon - \epsilon_{\text{small}}$, then:

$$\|x' - x\|_2 \leq \|x' - x^\circ\|_2 + \|x^\circ - x\|_2 \leq \epsilon^\circ + \epsilon_{\text{small}} \leq \epsilon.$$

Therefore, we have $\mathcal{B}_{\epsilon^\circ}(x^\circ) \subseteq \mathcal{B}_\epsilon(x)$. As a result:

$$\mathcal{B}_{\epsilon^\circ}(x^\circ) \cap H_\Delta \subseteq \mathcal{B}_\epsilon(x) \cap H_\Delta \subseteq \mathcal{P}.$$

It follows that $x^\circ \in \mathrm{int}(\mathcal{P})$. To conclude the proof, we observe that since $x^\circ \in \mathrm{int}(\mathcal{H})$, $\mathrm{int}(\mathcal{P} \cap \mathcal{H})$ is non-empty. ■