Contextual Search in the Presence of Irrational Agents

Akshay Krishnamurthy* Thodoris Lykouris† Chara Podimata‡ Robert Schapire§

First version: February 2020
Current version: November 2020¶

Abstract

We study contextual search, a generalization of binary search in higher dimensions, which captures settings such as feature-based dynamic pricing. Standard game-theoretic formulations of this problem assume that agents act in accordance with a specific behavioral model. In practice however, some agents may not prescribe to the dominant behavioral model or may act in ways that are seemingly *arbitrarily irrational*. Existing algorithms heavily depend on the behavioral model being (approximately) accurate for all agents and have poor performance in the presence of even a few such arbitrarily irrational agents.

We initiate the study of contextual search when some of the agents can behave in ways inconsistent with the underlying behavioral model. In particular, we provide two algorithms, one built on robustifying multidimensional binary search methods and one on translating the setting to a proxy setting appropriate for gradient descent. Our techniques draw inspiration from learning theory, game theory, high-dimensional geometry, and convex analysis.

*Microsoft Research NYC, akshaykr@microsoft.com

†Microsoft Research NYC, thlykour@microsoft.com

‡Harvard University, podimata@g.harvard.edu. Research was initiated while the author was an intern at Microsoft Research NYC. The author is supported in part under grant No. CCF-1718549 of the National Science Foundation.

§Microsoft Research NYC, schapire@microsoft.com

¶Compared to the first version titled *Corrupted Multidimensional Binary Search: Learning in the Presence of Irrational Agents*, this version provides a broader scope of behavioral models of irrationality, specifies how the results apply to different loss functions, and discusses the power and limitations of additional algorithmic approaches.

1 Introduction

We study *contextual search*, a fundamental algorithmic problem that extends classical binary search to higher dimensions and has direct applications, for instance, to pricing and personalized medicine [CLPL19, BB20, LPLV18]. In this problem, at every round t, some *context* $\mathbf{x}_t \in \mathbb{R}^d$ arrives. Associated with this context is an unknown *true value* $v_t \in \mathbb{R}$, which we here assume is a linear function of the context so that $v_t = \langle \boldsymbol{\theta}^\star, \mathbf{x}_t \rangle$ for some unknown vector $\boldsymbol{\theta}^\star \in \mathbb{R}^d$, called the *ground truth*. Based on the observed context $\mathbf{x}_t$, the decision-maker or *learner* selects a *query* $\omega_t \in \mathbb{R}$ with the goal of minimizing some *loss* that depends on the query as well as the true value; examples include the *absolute loss*, $|v_t - \omega_t|$, and the *ε-ball loss*, $\mathbb{1}\{|v_t - \omega_t| > \varepsilon\}$, both of which measure discrepency between ω_t and v_t. Finally, the learner observes whether or not $v_t \geq \omega_t$, but importantly, the true value v_t is never revealed, nor is the loss that was suffered.

For example, in feature-based dynamic pricing [CLPL19, LPLV18, PLS18, LPLS21], say, of Airbnb apartments, each context $\mathbf{x}_t$ describes a particular apartment with components, or features, providing the apartment's location, cleanliness, and so on. The true value v_t is the price an incoming customer or *agent* is willing to pay, which is assumed to be a linear function (defined by $\boldsymbol{\theta}^\star$) of $\mathbf{x}_t$. Based on $\mathbf{x}_t$, the host decides on a price ω_t. If this price is not more than the customer's value v_t, then the customer makes a reservation, yielding revenue ω_t; otherwise, the customer passes, generating no revenue. The host observes whether the reservation occurred (that is, if $\omega_t \leq v_t$). The natural loss in this setting is called the *pricing loss*, which captures how much revenue was lost relative to the maximum price that the customer was willing to pay.

A key challenge in contextual search is that the learner only observes *binary feedback*, i.e., whether or not $\omega_t \leq v_t$. This contrasts with classical machine learning where the learner observes either the entire loss function (full feedback) or only the loss itself for just the incurred query (bandit feedback).

The above model makes the strong assumption that the feedback is always consistent with the ground truth, typically called *full rationality* in behavioral economics. This is not always realistic. The assumption that the agent's value is described by a linear model may be violated to some degree. Moreover, some agents may behave in ways not prescribed by the dominant behavioral model, or may deviate from it for idiosyncratic reasons. Although some prior works allow for some benign, stochastic noise (see related work below), these are generally not robust to any interference that is adversarial, and therefore cannot extend to settings where some agents may act in ways that are *irrational* with respect to the underlying ground truth.

In this paper, we present the first contextual search algorithms that can handle adversarial noise models. In particular, we allow some agents to behave in ways that are arbitrarily inconsistent with the ground truth and are thereby seemingly *irrational*. Inspired by the recent line of work on stochastic learning with adversarial corruptions [LMPL18], we do not impose any stochastic assumption on the order in which the irrational agents arrive and aim for guarantees that gracefully degrade with how many such agents exist while attaining near-optimal guarantees when all agents are fully rational.

1.1 Our contributions

We first provide a unifying framework encompassing disparate behavioral models determining agent responses and various loss functions (Section 2). In particular, we assume that the agent behaves according to a *perceived value* $\widetilde{v}$ and the precise behavioral model determines the transformation

from true to perceived value. The loss functions can depend on either the true value (to capture parameter estimation objectives) or the perceived value (for pricing objectives). This formulation allows us to formalize arbitrarily irrational agents, a setting not studied in prior work. Our model also captures stochastic noise settings studied in prior work such as *bounded rationality* (Section 5).

Our first main result is an algorithm (Section 3) that works for all of the aforementioned loss functions (ε-ball, absolute, pricing) and applies to a variety of behavioral models. We prove that, with probability $1-\delta$, this algorithm enjoys a loss in performance or *regret* of $\mathcal{O}(C \cdot d^3 \cdot \operatorname{poly}\log(T/\delta))$ when employed against an unknown number C of irrational agents, where T is the number of rounds or *time-horizon*. Our guarantee is logarithmic in the time-horizon, as is typical in binary search methods when $C \approx 0$, and degrades gracefully as C becomes larger. Our algorithm builds on the ProjectedVolume algorithm [LPLV18] which is optimal for the ε-ball loss in contextual search.

Our main technical advance is a method for maintaining a set of candidates for $\boldsymbol{\theta}^\star$, successively removing candidates by hyperplane cuts while ensuring that $\boldsymbol{\theta}^\star$ is never removed. When $C = 0$, this is done via ProjectedVolume which removes all parameters $\boldsymbol{\theta}$ that are inconsistent with the query response in a way that each *costly* query guarantees enough volumetric progress in the size of the set of remaining parameters. However, when some responses are corrupted, such an aggressive elimination method runs the risk of removing the ground truth $\boldsymbol{\theta}^\star$ from the parameter space.

To deal with this key challenge, we run the algorithm in epochs, each corresponding to one query of ProjectedVolume, and only proceed to the next epoch if we can find a halfspace that both makes volumetric progress and does not eliminate $\boldsymbol{\theta}^\star$. We start from an easier setting where we assume access to a known upper bound $\bar{c}$ on the corrupted responses ($C \leq \bar{c}$) and only move to the next epoch when we can find a halfspace with enough volumetric progress that includes all parameters that are misclassified by at most $\bar{c}$ queries (thereby also $\boldsymbol{\theta}^\star$ as the latter needs to be consistent with all non-corrupted responses).

To avoid being fooled by corrupted responses, we face several challenges. First, even after arbitrarily many queries, we cannot guarantee that one of them induces such a halfspace (Section 4.4). Interestingly, when we do not restrict ourselves to the halfspaces associated with one particular query but allow for *improper* cuts, we can use ideas from convex analysis (in particular, the Carathéodory theorem) to show that one such query exists after collecting $\mathcal{O}(d^2\bar{c})$ queries (Section 4.1). In particular, we identify a point in the parameter space that is outside of a convex body including all the *protected parameters* (the ones with misclassification at most $\bar{c}$) and the separating hyperplane theorem implies the existence of the desired cut.

A second challenge is that the above argument is only existential and does not suggest a direct way to compute the halfspace. To deal with this, we use geometric techniques (in particular, volume cap arguments) to provide a sampling process that, with significant probability, identifies a point $\mathbf{q}$ with big enough margin from the aforementioned separating hyperplane. Subsequently, we use the classical learning-theoretic perceptron algorithm to either compute this hyperplane or resample a new point if its mistake bound is violated (Section 4.2).

There are two remaining, intertwined challenges. On the one hand, the running time of perceptron depends on the number of subregions created by removing all possible combinations of $\bar{c}$ queries which is exponential in $\bar{c}$. On the other hand, our algorithm needs to be agnostic to the number C. We deal with both of these via a multi-layering approach introduced in [LMPL18] that runs multiple parallel versions of the aforementioned algorithm with only $\bar{c} \approx \log T$ (Section 4.3). This results in a final algorithm that is quasipolynomial in the time horizon and does not assume knowledge of C.

Our second algorithm is based on gradient descent (Section 6) and has a guarantee of $\mathcal{O}(\sqrt{T} + C)$ for the absolute loss. This algorithm is simpler and has better running time but does not provide logarithmic guarantees when $C \approx 0$ and does not extend to non-Lipschitz loss functions such as the pricing loss. The key idea in its analysis lies in identifying a simple proxy loss function based on which we can run gradient descent and apply directly its corresponding regret guarantee.

1.2 Related work

Our work is closely related to the problem of *dynamic pricing* when facing an agent with *unknown* demand curve (see [dB15] for a survey). The single-dimensional case has been studied both from a parametric model standpoint ([LG08, BR12, dBZ14, dB14, KZ14]), and a non-parametric one with the goal of regret minimization. Our work falls in the second category, where Kleinberg and Leighton [KL03] first formulated the problem. Recently, Cesa-Bianchi, Cesari, and Perchet [CBCP19] studied a variant, where there are various possible demand curves, rather than only one.

Moving from the single- to the higher-dimensional case, there have been mainly two families of algorithms to deal with it. The more classical approach involves statistical methods based, for example, on linear regression and the central limit theorem [ARS14, QB16, JN19, BK17, NSLW19, BB20, GJL19, GJM19, SJB19]. This line of work does not deal with adversarial contexts and tends to obtain performance loss of the order of $\sqrt{T}$, but allows for some stochastic noise in the behavior of the agents (*bounded rationality*). Closer to our work are approaches based on multidimensional binary search. This line of work was introduced by Cohen, Lobel, and Paes Leme [CLPL19] who studied contextual pricing with adversarial contexts. Their results were improved for the ε-ball loss [LPLV18] and the pricing loss [PLS18, LPLS21]. Mao, Paes Leme, and Schneider [MPLS18] studied a variant of the standard contextual pricing problem, where the buyers' utilities are Lipschitz, rather than linear in the contexts. With respect to stochastic noise models, [CLPL19] extend their results to a stochastic noise model which we also discuss in Section 5. [1] A concurrent work by Liu, Paes Leme, and Schneider [LPLS21] also extends results for the absolute loss to a stochastic noise model where the feedback is flipped. However, all of the aforementioned works, both the statistical and the binary search approaches, do not extend to adversarial noise models such as the one we consider.

The single-dimensional version of the problem we study bears a lot of similarities with *Ulam's game* ([Ula91]), where one wants to make the least number of queries to an opponent, in order to identify a number within a known range. The opponent can only inform the player that the queried point is larger/smaller than the target one, and has a fixed number of "lies" that you do not know when they will occur ([Spe92], see also [Pel02] for a survey). Rivest, Meyer, Kleitman, Winklmann, and Spencer [RMK+80] provide the optimal solution for this problem. This question is also related with works in noisy binary search [KK07]. Although Ulam's game can be interpreted as the single dimensional version of the contextual search problem that we are solving, the techniques presented in [RMK+80] require exponential computation time for higher dimensions. To the best of our knowledge, adversarial noise models have not been studied for binary search in higher dimensions.

Our work also draws intuitions from the recent literature in bandit learning with adversarial corruptions; indeed, we use "corruptions" to model our adversarially irrational agents. Learning in the presence of adversarial corruptions was introduced by Lykouris, Mirrokni, and Paes Leme [LMPL18] for stochastic multi-armed bandits and their results have been subsequently strengthened by Gupta, Koren, and Talwar [GKT19] and Zimmert and Seldin [ZS19]. This line of work has been extended

[1] Models of stochastic noise have also been considered for the problem of binary search in graphs in [EZKS16].

to several other settings [AAK+20, CKW19, LLS19, BKS20, LSSS19]. Our work differs from these in that we use adversarial corruptions as a *modeling tool* to capture arbitrarily irrational agent behavior in game-theoretic settings.[2] On a technical level, our setting involves a continuous action space with requires new analytical tools, while all prior results involve discrete (potentially large) action spaces.

2 Model

In this section, we provide a general framework (Section 2.1) that allows us to study contextual search under different behavioral models (Section 2.2) and different loss functions (Section 2.3).

2.1 Protocol

We consider the following repeated interaction between the learner and nature. Following classical works in contextual search [CLPL19, LPLV18, PLS18] we assume that the learner has access to a parameter space $\mathcal{K} = \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\|_2 \leq 1\}$ and a context space $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1\}$. We denote by $\Omega = \mathbb{R}$ the decision space of the learner and by $\mathcal{V} = \mathbb{R}$ a value space; in the pricing setting, Ω can be thought as the set of possible prices available to the learner and $\mathcal{V}$ as a set of values associated with incoming agents. Domain $\mathcal{V}$ helps express both the true value of the agents and the perceived value driving their decisions. Finally, we consider a behavioral model determining the transformation from the agent's *true value* to a *perceived value* that drives his decision at each round. All of the above are known to the learner throughout the learning process.

The setting proceeds for T rounds. Before the first round, nature chooses a ground truth $\boldsymbol{\theta}^\star \in \mathcal{K}$; this is fixed across rounds and is *not* known to the learner. This ground truth determines both the agent's *true* value function $v : \mathcal{X} \to \mathcal{V}$ and the learner's loss function $\ell : \Omega \times \mathcal{V} \times \mathcal{V} \to [0, 1]$. We note that both value and loss functions are also functions of the ground truth $\boldsymbol{\theta}^\star$; given that $\boldsymbol{\theta}^\star$ is fixed throughout this process, we drop the dependence on $\boldsymbol{\theta}^\star$ to ease notation. The functional form of both $v(\cdot)$ and $\ell(\cdot)$ as a function of the ground truth $\boldsymbol{\theta}^\star$ is known to the learner but the learner does not know $\boldsymbol{\theta}^\star$. For each round $t = 1, \ldots, T$:

1. Nature chooses (potentially adaptively and adversarially) and reveals context $\mathbf{x}_t \in \mathcal{X}$.
2. Nature chooses but *does not* reveal a perceived value $\widetilde{v}_t \in \mathcal{V}$ based on the behavioral model.
3. Learner selects query point $\omega_t \in \Omega$ (in a randomized manner) and observes $y_t = \operatorname{sgn}(\widetilde{v}_t - \omega_t)$. [3]
4. Learner incurs (but does *not* observe) loss: $\ell(\omega_t, v(\mathbf{x}_t), \widetilde{v}_t) \in [0, 1]$.

Nature is an adaptive adversary (subject to the behavioral model), i.e., it knows the learner's algorithm along with the realization of all randomness up to and including round $t-1$ (i.e, it knows all $\omega_\tau, \forall \tau \leq t-1$), does not know the learner's randomness at the current round t. Moreover, the learner only observes the context $\mathbf{x}_t$ and the *binary* variable y_t as described in Steps 1 and 3 of the protocol, and has access to neither the perceived value $\widetilde{v}_t$ nor the loss $\ell(\omega_t, v(\mathbf{x}_t), \widetilde{v}_t)$. Finally, in the pricing setting, y_t corresponds to whether the agent associated with round t made a purchase or not.

The above protocol unifies contextual search under general behavioral models and loss functions. In Section 2.2, we discuss the different behavioral models we consider that determine the agents'

[2] [CKW19] also consider a behavioral setting, but do not make a connection to irrationality.

[3] $\operatorname{sgn}(\cdot)$ is the sign function, i.e., $\operatorname{sgn}(x) = 1$ if $x \geq 0$ and -1 otherwise

perceived values (Step 2 in the protocol). In Section 2.3, we discuss the main loss functions for this setting (Step 4 in the protocol) as well as the corresponding performance metrics.

2.2 Behavioral models

We assume that the agents' true value function is: $v(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\theta}^\star \rangle$ for any $\mathbf{x} \in \mathcal{X}$ (i.e., independent of the agent's behavioral model). The behavioral model affects the perceived value $\widetilde{v}$ at round t, which then affects both the loss incurred and the feedback observed by the learner. The behavioral model that is mostly studied in contextual search works is *full rationality*. This assumes that agents always behave according to their true value, i.e., $\widetilde{v}_t = v(\mathbf{x}_t) = \langle \mathbf{x}_t, \boldsymbol{\theta}^\star \rangle$. In learning-theoretic terms, this consistency with respect to a ground truth is typically referred to as *realizability*.

Our main focus in this work is the study of a behavioral model that we call *adversarial irrationality*. There, nature selects the rounds where the irrational agents arrive ($c_t = 1$ if irrational agents arrive, else $c_t = 0$), together with an upper bound C on this number of rounds (i.e., $\sum_{t \in [T]} c_t \leq C$). Neither the sequence $\{c_t\}_{t \in [T]}$ nor the number C are ever revealed to the learner. If $c_t = 0$, then nature is constrained to $\widetilde{v}_t = v(\mathbf{x}_t)$, but can select adaptively and adversarially $\widetilde{v}_t$ if $c_t = 1$. This model is inspired by the model of adversarial corruptions in stochastic bandit learning [LMPL18].

Our results extend to *bounded rationality* which posits that the perceived value is the true value plus some noise parameter. The noise parameter is drawn from a σ-subgaussian distribution $\texttt{subG}(\sigma)$, *fixed* across rounds and *known* to the learner, i.e., nature selects it before the first round and reveals it. At every round t a realized noise $\xi_t \sim \texttt{subG}(\sigma)$ is drawn, but ξ_t is never revealed to the learner. The agent's perceived value is then $\widetilde{v}_t = v(\mathbf{x}_t) + \xi_t$. This stochastic noise model has been studied in the contextual search literature as a way to incorporate idiosyncratic market shocks [CLPL19].

2.3 Loss functions and objective

We study three variants for the learner's loss function: the ε-ball, the absolute, and the pricing loss. Abstracting away from t subscripts and dependencies on contexts $\mathbf{x}$, the loss $\ell(\omega, v, \widetilde{v})$ evaluates the loss of a query ω when the true value is v and the perceived value is $\widetilde{v}$.

The first class of loss functions includes parameter estimation objectives that estimate the value of $\boldsymbol{\theta}^\star$. One such function is the *ε-ball loss* which is defined with respect to an accuracy parameter $\varepsilon > 0$. The ε-ball is 1 if the difference between the query point ω and the true value v is larger than ε and 0 otherwise. Formally, $\ell(\omega, v, \widetilde{v}) = \mathbb{1}\{|v - \omega| \geq \varepsilon\}$. Another parameter estimation loss function is the *absolute* or *symmetric* loss that captures the absolute difference between the query point and the true value, i.e., $\ell(\omega, v, \widetilde{v}) = |v - \omega|$. The aforementioned loss functions are unobservable to the learner as the true value v is latent; this demonstrates that binary feedback does not offer strictly more information than the bandit feedback as the latter reveals the loss of the selected query.

Another important objective in pricing is the revenue collected which is the price ω in the event that the purchase occurred, i.e., $\widetilde{v} \geq \omega$. This can be expressed based on observable information by setting a reward equal to ω when $\widetilde{v} \geq \omega$ and 0 otherwise. However, this expression does not convey that querying $\omega = \widetilde{v}$ is optimal and does not enable logarithmic performance guarantees that are typical in binary search. A loss function exploiting this structure is the *pricing loss* which is defined as the difference between the highest revenue that the learner could have achieved at this round (the agent's *perceived* value $\widetilde{v}$) and the revenue that the learner currently receives, i.e., ω if a purchase happens, and 0 otherwise. The outcome of whether a purchase happens or not is tied to whether ω is higher or smaller than the perceived value $\widetilde{v}$. Putting everything together:

$\ell(\omega, v, \widetilde{v}) = \widetilde{v} - \omega \cdot \mathbb{1}\{\omega \le \widetilde{v}\}$.

We remark that the ε-ball and the absolute loss depend only on the true value v (and not the perceived value $\widetilde{v}$); indeed, when these losses are considered $\widetilde{v}$ affects only the feedback that the learner receives. That said, we define $\ell(\cdot,\cdot,\cdot)$ with three arguments for unification purposes, since the pricing loss does depend on the feedback that the learner receives (and hence, on $\widetilde{v}$).

The learner's goal is to minimize a notion of regret. For adversarially irrational agents, the loss of the best-fixed action in hindsight is *at least* 0 and *at most* C. Hence, to simplify exposition for the case of adversarially irrational agents we slightly abuse notation and conflate the loss and the regret:

$$R(T) = \sum_{t\in[T]} \ell(\omega_t, v(\mathbf{x}_t), \widetilde{v}_t) \tag{1}$$

This is no longer possible for bounded rational agents. We defer the more careful definition of the regret of this cases to Section 5.

3 Corrupted Projected Volume: the algorithm and main guarantee

In this section, we provide an algorithmic scheme that handles all the aforementioned behavioral models and loss functions. The main result of this and the next section is an algorithm (Algorithm 4) for the adversarial irrationality behavioral model when there is an *unknown* upper bound C on the number of irrational agents. The regret of this algorithm is upper bounded by the following theorem.

Theorem 3.1. Run with accuracy $\varepsilon > 0$ and an unknown corruption level C, CorPV.AI incurs regret $\mathcal{O}(d^3 \cdot \log(T/\beta) \cdot \log(d/\varepsilon) \cdot \log(1/\beta) \cdot (\log T + C))$ with probability at least $1-\beta$ for the ε-ball loss. Run with $\varepsilon = 1/T$, its regret for the pricing and absolute loss is $\mathcal{O}(d^3 \log(dT) \log(T) \cdot (\log T + C) \log(1/\beta))$ with probability at least $1-\beta$. The expected runtime of the algorithm is quasi-polynomial; in particular, it is $\mathcal{O}((d^2 \log T)^{\text{poly}\log T} \cdot \text{poly}(d, \log T))$.

We first present the algorithm in this section and prove the stated theorem in Section 4.3. A useful intermediate setting is the case where we know an upper bound $\bar{c}$ on the number of adversarial agents, i.e. $C \le \bar{c}$; we refer to this as the *$\bar{c}$-known-corruption* setting (Section 3.1). This setting allows us to introduce our key ideas and serves as a building block to extend to both the setting where C is unknown (Section 3.2) as well as the bounded rationality behavioral model (Section 5).

3.1 Algorithm for the known-corruption setting

Our algorithm CorPV.Known (Algorithm 1) builds on the ProjectedVolume algorithm of Lobel, Paes Leme, and Vladu [LPLV18] which is optimal in terms of regret for the ε-ball loss when $\bar{c} = 0$ (see Appendix A.1). The main idea in ProjectedVolume is to maintain a *knowledge set* $\mathcal{K}_t$ which includes all candidate parameters $\boldsymbol{\theta}$ that are *consistent* with what has been observed so far. The true parameter $\boldsymbol{\theta}^\star$ is always consistent and therefore is never eliminated from the knowledge set. Further the volume of the knowledge set is intuitively a measure of progress for the algorithm.

Given a context $\mathbf{x}_t$, there are two scenarios. Before we describe them, we define the *width* of a body $\mathcal{K}$ on direction $\mathbf{x}$ as $w(\mathcal{K}, \mathbf{x}) = \sup_{\boldsymbol{\theta},\boldsymbol{\theta}'\in\mathcal{K}} \langle \boldsymbol{\theta} - \boldsymbol{\theta}', \mathbf{x}\rangle$. If the width of the knowledge set in the direction of $\mathbf{x}_t$ is $w(\mathcal{K}, \mathbf{x}_t) \le \varepsilon$, the the algorithm can make an *exploit* query $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta}_t\rangle$ for *any* point $\boldsymbol{\theta}_t \in \mathcal{K}_\phi$ thereby guaranteeing an ε-ball loss equal to 0. Otherwise, if $w(\mathcal{K}, \mathbf{x}_t) > \varepsilon$, the

algorithm queries the point $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\kappa}_t \rangle$, where $\boldsymbol{\kappa}_t$ is the (approximate) centroid of $\mathcal{K}_t$. [4] This is called an *explore* query. By querying this point, the algorithm learns that $\boldsymbol{\theta}^\star$ lies in one of the two halfspaces passing through $\boldsymbol{\kappa}_t$ with normal vector $\mathbf{x}_t$, so it can update the knowledge set by taking intersection with this halfspace. We use $(\mathbf{h}, \omega)$, $\mathbf{H}^+(\mathbf{h}, \omega)$, and $\mathbf{H}^-(\mathbf{h}, \omega)$ to denote the hyperplane with normal vector $\mathbf{h} \in \mathbb{R}^d$ and intercept ω, and the positive and negative halfspaces it creates with intercept ω, i.e., $\{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{h}, \mathbf{x} \rangle \geq \omega\}$ and $\{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{h}, \mathbf{x} \rangle \leq \omega\}$, respectively. By properties of $\boldsymbol{\kappa}_t$, the volume of the updated knowledge set is a constant factor of the initial volume, leading to geometric volume progress. For technical reasons, ProjectedVolume keeps a set S_t of dimensions with small width and works with the so-called *cylindrification* $\texttt{Cyl}(\mathcal{K}_t, S_t)$ rather than the knowledge set $\mathcal{K}_t$. Although we do the same to build on their analysis in a black-box manner, the distinction between $\mathcal{K}_t$ and $\texttt{Cyl}(\mathcal{K}_t, S_t)$ is not important for understanding our algorithmic ideas and the corresponding definitions are deferred to Section 3.3.

Having described ProjectedVolume that works when there are no corruptions, we turn to our algorithm. The problem is that, even when there are few corruptions, ProjectedVolume may quickly eliminate $\boldsymbol{\theta}^\star$ from $\mathcal{K}_t$ (see Appendix A.2). To deal with this, we run the algorithm in epochs consisting of multiple queries. The goal of the epochs is to ensure that $\boldsymbol{\theta}^\star$ is never eliminated from the knowledge set and that, at the end of the epoch, we make enough volumetric progress on the latter's size. We face three important design decisions discussed separately below: what occurs inside an epoch, when to stop an epoch, and how to initialize the next one.

What occurs within an epoch? CorPV.Known (Algorithm 1) formalizes what happens within an epoch ϕ. The knowledge set is updated only at its end; this means that all rounds t in epoch ϕ have the same knowledge set $\mathcal{K}_\phi$ (and hence, the same centroid $\boldsymbol{\kappa}_\phi$). If the width of the knowledge set in the direction of $\mathbf{x}_t$ is smaller than ε, then, as in ProjectedVolume, we make an *exploit* query precisely described in Section 3.3. Otherwise, we make an *explore* query $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\kappa}_\phi \rangle$, described below. The epoch keeps track of all explore queries that occur within its duration in a set $\mathcal{A}_\phi$. When it ends ($\phi' = \phi + 1$), the knowledge set $\mathcal{K}_{\phi+1}$ of the new epoch is initialized. In this subsection, $\texttt{Cyl}(\mathcal{K}_\phi, S_\phi)$ can be thought as the knowledge set $\mathcal{K}_\phi$ and the sets S_ϕ and L_ϕ can be ignored; these quantities are needed for technical reasons and are discussed in Section 3.3.

ALGORITHM 1: CorruptedProjectedVolume-Known (CorPV.Known)

Global parameters: Budget $\bar{c}$, accuracy ε
Initialize $\phi = 1, \mathcal{K}_\phi \leftarrow \mathcal{K}, S_\phi \leftarrow \emptyset$, $\boldsymbol{\kappa}_\phi \leftarrow$ apx-centroid$(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi))$, $L_\phi \leftarrow$ orthonorm-basis$(\mathbb{R}^d)$, $\mathcal{A}_\phi \leftarrow \emptyset$

for $t \in [T]$ **do**
  Observe context $\mathbf{x}_t$.
  if $w(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \leq \varepsilon$ *or* $L_\phi = \emptyset$ **then**
    Select query point $\omega_t =$ CorPV.Exploit$(\mathbf{x}_t, \mathcal{K}_\phi)$
  else
    $(\phi', \mathcal{A}_\phi) \leftarrow$ CorPV.Explore$(\mathbf{x}_t, \phi, \boldsymbol{\kappa}_\phi, L_\phi, \mathcal{A}_\phi)$
    if $\phi' = \phi + 1$ **then** ▷ epoch changed
      Compute separating cut: $(\widetilde{\mathbf{h}}, \widetilde{\omega}) \leftarrow$ CorPV.SeparatingCut$(\boldsymbol{\kappa}_\phi, S_\phi, L_\phi, \mathcal{A}_\phi)$
      Make updates $(\mathcal{K}_{\phi+1}, S_{\phi+1}, L_{\phi+1}) \leftarrow$ CorPV.EpochUpdates$(\mathcal{K}_\phi, S_\phi, L_\phi, \widetilde{\mathbf{h}}, \widetilde{\omega})$
      Initialize next epoch: $\phi \leftarrow \phi'$, $\boldsymbol{\kappa}_\phi \leftarrow$ apx-centroid$(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi))$, and and $\mathcal{A}_\phi \leftarrow \emptyset$.

CorPV.Explore (Algorithm 2) describes how we handle an explore query. When $\bar{c} = 0$, we can

[4] For a convex body $\mathcal{K}$, the centroid is defined as $\boldsymbol{\kappa}^\star = \frac{1}{\texttt{vol}(\mathcal{K})} \int_{\mathcal{K}} \mathbf{u} d\mathbf{u}$, where $\texttt{vol}(\cdot)$ denotes the volume of a set. Although computing the exact centroid of a convex set is #P-hard, one can efficiently approximate it [LPLV18].

eliminate the halfspace that lies in the opposite direction of the feedback y_t after each explore query. However, when $\bar{c} > 0$, this may eliminate $\boldsymbol{\theta}^\star$. Instead, we keep all explore queries that occurred in epoch ϕ as well as the halfspace consistent with the observed feedback in $\mathcal{A}_\phi$ and wait until we have enough data to identify a halfspace of the knowledge set that includes $\boldsymbol{\theta}^\star$ and makes sufficient volumetric progress; we refer to this as a *separating cut*. We then move to epoch $\phi' = \phi + 1$.

ALGORITHM 2: CorPV.Explore

Parameters: $\mathbf{x}_t$, ϕ, $\boldsymbol{\kappa}_\phi$, L_ϕ, $\mathcal{A}_\phi$
Select query point $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\kappa}_\phi \rangle$ and observe feedback y_t.
Update explore queries: **if** $y_t = +1$: $\mathcal{A}_\phi \leftarrow \mathcal{A}_\phi \bigcup \mathbf{H}^+ \left(\Pi_{L_\phi} \mathbf{x}_t, \omega_t\right)$ **else** $\mathcal{A}_\phi \leftarrow \mathcal{A}_\phi \bigcup \mathbf{H}^- \left(\Pi_{L_\phi} \mathbf{x}_t, \omega_t\right)$.
if $|\mathcal{A}_\phi| \geq \tau$ **then** ▷ $\tau := 2d \cdot \bar{c} \cdot (d+1) + 1$
    Move to next epoch $\phi' \leftarrow \phi + 1$.
else Stay in the same epoch $\phi' \leftarrow \phi$.
return $(\phi', \mathcal{A}_\phi)$

When does the epoch end? It turns out that $\tau = 2d \cdot \bar{c} \cdot (d+1) + 1$ explore queries suffice to identify such a separating cut. In particular, it suffices to ensure that all candidate parameters $\boldsymbol{\theta}$ on its negative halfspace are misclassified by at least $\bar{c} + 1$ explore queries; since there are at most $\bar{c}$ corruptions, this guarantees that $\boldsymbol{\theta}^\star$ is not incorrectly eliminated. In other words, if the set of all candidate parameters $\boldsymbol{\theta}$ that are misclassified by at most $\bar{c}$ explore queries are on the non-eliminated halfspace of the hyperplane, then this hyperplane can serve as separating cut. We refer to the set of these parameters as the *protected region* as we aim to ensure that they are not eliminated.

At first glance, one might think that, after a few explore queries, we can directly use one of them as a separating cut. Interestingly, although this is indeed the case for $d = 2$, we show that for $d = 3$, if we are restricted to separating cuts on the direction of existing explore queries, even arbitrarily many such queries do not suffice (see Section 4.4). One key technical component in our analysis is to show that when we remove this restriction and allow for *improper cuts* then, after τ explore queries, there exists a point $\mathbf{p}^\star$ close to $\boldsymbol{\kappa}_\phi$ that is outside the convex hull of the protected region. A cut that separates the point $\mathbf{p}^\star$ from the convex hull of the protected region exists because of the separating hyperplane theorem and implies enough volumetric progress as explained in Section 4.2. The proof of the existence argument relies on the Carathéodory theorem and is described in Section 4.1.

How do we initialize the next epoch? Since the existence of the separating cut is established, if we were able to compute this cut, we would be able to compute the knowledge set of the next epoch by taking its intersection with the positive halfspace of the cut. However, the separating hyperplane theorem provides only an existential argument and no direct way to compute the separating cut. To deal with this, recall that the separating cut should have $\mathbf{p}^\star$ on its negative halfspace and the whole protected region in its positive halfspace. To compute it, we use the Perceptron algorithm [Ros58], which is typically used to provide a linear classifier for a set of (positive and negative) points in the *realizable* setting (i.e., when there exists a hyperplane that correctly classifies these points). Perceptron proceeds by iterating across the points and suggesting a classifier. Every time that a point is misclassified, Perceptron makes an update. If the entire protected region is classified as positive and $\mathbf{p}^\star$ as negative by Perceptron, then we return its hyperplane as the separating cut; otherwise we feed one point that violates the intended labeling to Perceptron. Perceptron makes a mistake and updates its classifier. The main guarantee of Perceptron is that, if there exists a classifier with margin of $\gamma > 0$ (i.e., smallest distance to any data point is γ), the number of mistakes that Perceptron does is at most of the order of $1/\gamma^2$ (precisely, the bound is given in Lemma B.10).

The problem is that we do not know $\mathbf{p}^\star$ and, even if we deal with this, $\mathbf{p}^\star$ does not necessarily have a big enough margin from the protected region. To overcome this, we provide a sampling process that with big enough probability identifies a different point $\mathbf{q}$, *in the vicinity* of $\mathbf{p}^\star$, whose margin to the protected region is lower bounded by γ. If $\mathbf{q}$ does have the desired margin, the mistake bound of Perceptron controls the running time needed to identify the separating hyperplane. Otherwise, we proceed with a new random point. This takes care of the margin-problem of $\mathbf{p}^\star$.

In order to pin down $\mathbf{p}^\star$, we construct a set of points Λ_ϕ, which we call *landmarks*, such that at least one of them is outside of the convex hull of the protected region. We run multiple versions of Perceptron, each with a random $\mathbf{p}^\star \in \Lambda_\phi$ and a point $\mathbf{q}$ randomly selected in a ball around $\mathbf{p}^\star$ of an appropriately defined radius ζ, which we denote by $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$. [5] If $\mathbf{q}$ has a big-enough margin then the mistake bound of Perceptron ensures that CorPV.SeparatingCut (Algorithm 3) returns the separating cut. Volume cap arguments detailed in Section 4.2 show that point $\mathbf{q}$ has the required margin with big enough probability, which bounds the number of the outer *while* loops and thereby also the running time.

ALGORITHM 3: CorPV.SeparatingCut

1 **Parameters:** $\boldsymbol{\kappa}_\phi, S_\phi, L_\phi, \mathcal{A}_\phi$ ▷ size of small dimensions $\delta := \frac{\varepsilon}{4(d+\sqrt{d})}$
2 Fix landmarks $\Lambda_\phi = \{\boldsymbol{\kappa}_\phi \pm \bar{\nu} \cdot e_i, \forall e_i \in E_\phi\}$ where E_ϕ is an orthonormal basis on L_ϕ and $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$
3 **while** `true` **do**
4 Initialize Perceptron hyperplane to $(\widetilde{\mathbf{h}}, \widetilde{\omega})$ and mistake counter to $M = 0$.
5 Sample a random point $\mathbf{q}$ from ball $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$ with radius $\zeta = \bar{\nu}$ around random $\mathbf{p}^\star \in \Lambda_\phi$.
6 **while** $M < \frac{d-1}{\zeta^2 \cdot \ln^2(3/2)}$ **do** ▷ Perceptron mistake bound
7 Set $m \leftarrow 0$.
8 **if** $\mathbf{q} \in \mathbf{H}^-(\widetilde{\mathbf{h}}, \widetilde{\omega})$ **then** make Perceptron update of $(\widetilde{\mathbf{h}}, \widetilde{\omega})$ on $\mathbf{q}$ with label $-$; set $m \leftarrow m + 1$.
9 **if** $\boldsymbol{\kappa}_\phi \in \mathbf{H}^+(\widetilde{\mathbf{h}}, \widetilde{\omega})$ **then** make Perceptron update of $(\widetilde{\mathbf{h}}, \widetilde{\omega})$ on $\boldsymbol{\kappa}_\phi$ with label $+$; set $m \leftarrow m + 1$.
10 **for** *subsets* $D_\phi \subseteq \mathcal{A}_\phi$ *such that* $|D_\phi| = \bar{c}$ **do**
11 Let P be the polytope created by halfspaces of $\mathcal{A}_\phi \setminus D_\phi$ and $\mathbf{H}^-(\widetilde{\mathbf{h}}, \widetilde{\omega})$.
12 **if** $P \neq \emptyset$ **then** make Perceptron update of $(\widetilde{\mathbf{h}}, \widetilde{\omega})$ on $\mathbf{z} \in P$ with label $+$; set $m \leftarrow m + 1$.
13 **if** $m \neq 0$ **then** increase mistake counter $M \leftarrow M + m$.
14 **else return** $(\widetilde{\mathbf{h}}, \widetilde{\omega})$

We discuss next the computational complexity of our algorithm. As written in lines 10-12 of Algorithm 3, checking whether the protected region is contained in the positive halfspace of the Perceptron hyperplane requires going over all $\binom{|\mathcal{A}_\phi|}{\bar{c}}$ ways to remove $\bar{c}$ hyperplanes and checking whether the resulting region intersects the negative halfspace (if this happens, then points with misclassification of at most $\bar{c}$ may be misclassified). This suggests a running time that is exponential in $\bar{c}$. Fortunately, as detailed in Section 3.2, to handle the unknown corruption or the other intricacies in our actual behavioral model beyond the $\bar{c}$-known-corruption, we only run this algorithm with $\bar{c} \approx \log(T)$. As a result, the final running time of our algorithms is quasi-polynomial in T.

3.2 Adapting to an unknown corruption level

We now provide the algorithm when the corruption level C is unknown (Algorithm 4). The places where CorPV.AI differs from CorPV.Known are in lines 4, 7-8, 15-19 of Algorithm 4. This section extends ideas from [LMPL18] for multi-armed bandits to contextual search which poses an

[5] Such a point can be computed efficiently by normalizing $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$ to a unit ball and then using the techniques presented in [BHK16, Section 2.5].

additional difficulty as the search space is continuous. This is not as straightforward as a doubling trick for the unknown C, as both the loss and the corruption c_t are unobservable; doubling tricks require identifying a proxy for the quantity under question and doubling once a threshold is reached.

The basic idea is to maintain multiple copies of CorPV.Known, which we refer to as *layers*. At every round, we decide which copy to play probabilistically. Each copy j keeps its own environment with its corresponding epoch $\phi(j)$ and knowledge set $\mathcal{K}_{j,\phi(j)}$. Smaller values j for the copies are *less robust* to corruption and we impose a monotonicity property among them by ensuring that the knowledge sets are nested, i.e., $\mathcal{K}_{j,\phi(j)} \subseteq \mathcal{K}_{j',\phi(j')}$ for $j \leq j'$. This allows more robust layers to correct mistakes of less robust layers that may inadvertently eliminate $\boldsymbol{\theta}^\star$ from their knowledge set.

More formally, we run $\log T$ parallel versions of the $\bar{c}$-known-corruption algorithm with a corruption level of $\bar{c} \approx \log(T)$. At the beginning of each round t, the algorithm randomly selects layer j with probability 2^{-j} (line 4) and executes the layer's algorithm for this round. Since the adversary does not know the randomness in the algorithm, this makes layers j with $C \leq 2^j$ robust to corruption level of C. The reason is that the expected number of corruptions occurring at layer j is at most 1 and, with high probability, less than $\log T$ which is accounted by the $\bar{c} = \log T$ upper bound on corruption based on which we run CorPV.Known on this layer.

However, there is a problem: all layers with $C > 2^j$ are not robust to corruption of C so they may eliminate $\boldsymbol{\theta}^\star$ and, to make things worse, the algorithm follows the recommendation of these layers with large probability. As a result, we need a way to supervise their decisions by more robust layers. To achieve that, we use nested active sets; when the layer j_t selected at round t proceeds with a separating cut on its knowledge set, we also make the same cut on all less robust layers $j' < j_t$ (lines 15-16). This allows non-robust layers that have eliminated $\boldsymbol{\theta}^\star$ from their knowledge set to correct their mistakes by removing the incorrect parameters of their version space that they had converged to from their knowledge sets.

ALGORITHM 4: CorPV.AI (Adversarial Irrationality version)

1 **Global parameters:** Failure probability β, budget $\bar{c} := 2\log(T/\beta)$, accuracy ε
2 Initialize layer-specific quantities for all layers $j \in [\log T]$: $\phi(j) = 1$, $\mathcal{K}_{j,\phi(j)} \leftarrow \mathcal{K}$, $S_{j,\phi(j)} \leftarrow \emptyset, \boldsymbol{\kappa}_{j,\phi(j)} \leftarrow$ apx-centroid($\texttt{Cyl}(\mathcal{K}_{j,\phi(j)}, S_{j,\phi(j)})$), $L_{j,\phi(j)} \leftarrow$ orthonorm-basis($\mathbb{R}^d$), $\mathcal{A}_{j,\phi(j)} \leftarrow \emptyset$
3 **for** $t \in [T]$ **do**
4 Sample layer $j_t \in [\log T]$: $j_t = j$ with probability 2^{-j}; with remaining probability, $j_t = 1$.
5 Observe context $\mathbf{x}_t$.
6 **if** $w(\texttt{Cyl}(\mathcal{K}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)}) \leq \varepsilon$ *or* $L_{j_t,\phi(j_t)} \neq \emptyset$ **then**
7 Find smallest more robust layer $j \geq j_t$ with small width: $j = \min_{j' \geq j_t} w(\mathcal{K}_{j',\phi(j')}, \mathbf{x}_t) \leq \varepsilon$..
8 Compute *exploit* query point for this layer: $\omega_t =$ CorPV.Exploit($\mathbf{x}_t, \mathcal{K}_{j,\phi(j)}$) .
9 **else**
10 $(\phi', \mathbf{p}^\star, \mathcal{A}_{j_t,\phi(j_t)}) \leftarrow$ CorPV.Explore($\mathbf{x}_t, \phi(j_t), \boldsymbol{\kappa}_{j_t,\phi(j_t)}, L_{j_t,\phi(j_t)}, \mathcal{A}_{j_t,\phi(j_t)}$)
11 **if** $\phi' = \phi(j_t) + 1$ **then**
12 $(\widetilde{\mathbf{h}}, \widetilde{\omega}) \leftarrow$ CorPV.SeparatingCut($\boldsymbol{\kappa}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)}, L_{j_t,\phi(j_t)}, \mathcal{A}_{j_t,\phi(j_t)}$)
13 $(\mathcal{K}_{j_t,\phi'}, S_{j_t,\phi'}, L_{j_t,\phi'}) \leftarrow$ CorPV.EpochUpdates($\mathcal{K}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)}, L_{j_t,\phi(j_t)}, \widetilde{\mathbf{h}}, \widetilde{\omega}$)
14 $\phi(j_t) \leftarrow \phi'$, $\boldsymbol{\kappa}_{j_t,\phi(j_t)} \leftarrow$ apx-centroid($\texttt{Cyl}(\mathcal{K}_{j_t,\phi(j_t)}, S_{j_t,\phi(j_t)})$), and $\mathcal{A}_{j_t,\phi(j_t)} \leftarrow \emptyset$.
15 **for** $j' \leq j_t$ **do** ▷ Make less robust layers consistent with j_t
16 $(\mathcal{K}_{j',\phi(j')}, S', L') \leftarrow$ CorPV.EpochUpdates($\mathcal{K}_{j',\phi(j')}, S_{j',\phi(j')}, L_{j',\phi(j')}, \widetilde{\mathbf{h}}, \widetilde{\omega}$)
17 **if** $\boldsymbol{\kappa}_{j',\phi(j')} \notin \mathcal{K}_{j',\phi(j')}$ ***or*** $S' \neq S_{j',\phi(j')}$ **then**
18 $\phi(j') \leftarrow \phi(j') + 1$, $(S_{j',\phi(j')+1}, L_{j',\phi(j')+1}) \leftarrow (S', L')$, $\mathcal{A}_{j',\phi(j')} \leftarrow \emptyset$
19 $\boldsymbol{\kappa}_{j',\phi(j')} \leftarrow$ apx-centroid($\texttt{Cyl}(\mathcal{K}_{j',\phi(j')}, S_{j',\phi(j')})$).

There are two additional points that arise in the contextual search setting. First, the aforementioned cut may not make enough volumetric progress in the knowledge sets of layers $j' < j_t$. As a result, as described in lines 17-19, we only move to the next epoch for layer j' if its centroid is removed from the knowledge set or another change discussed in Section 3.3 is triggered. Second, with respect to exploit queries, we want to make sure that we do not keep confidence on non-robust layers. As a result, we follow the exploit recommendation of the largest layer $j \geq j_t$ that has converged to exploit recommendation in this direction, i.e., $w(\mathcal{K}_{j,\phi(j)}) \leq \varepsilon$ (lines 7- 8). This eventually allows us to bound the regret from all non-robust layers by the smallest robust layer $\lceil \log C \rceil$ (see Section 4.3).

3.3 Remaining components of the algorithm.

The presentation of the algorithm until this point has disregarded some technical parts. We now discuss each of them so that the algorithm is fully defined.

Cylindrification, small, and large dimensions. To facilitate relating the volume progress to a bound on the explore queries, similar to [LPLV18], we keep two sets of vectors/dimensions S_ϕ and L_ϕ whose union creates an orthonormal basis. The set S_ϕ has *small dimensions* $\mathbf{s} \in S_\phi$ with width $w(\mathcal{K}_\phi, \mathbf{s}) \leq \delta$ for $\delta := \frac{\varepsilon}{4(d+\sqrt{d})}$. The set L_ϕ is any basis for the subspace orthogonal to S_ϕ, with the property that $\forall \mathbf{l} \in L_\phi : w(\mathcal{K}_\phi, \mathbf{l}) > \delta$. The set L_ϕ completes an orthonormal basis maintaining that $\mathbf{l} \in L_\phi : w(\mathcal{K}_\phi, \mathbf{l}) > \delta$. When an epoch ends, sets S_ϕ and L_ϕ are updated together with the knowledge set $\mathcal{K}_\phi$ as described in CorPV.EpochUpdates (Algorithm 5): if the new direction $\widetilde{\mathbf{h}}$ of the separating cut projected to the large dimensions has width $w(\Pi_{L_\phi}\mathcal{K}_{\phi+1}, \widetilde{\mathbf{h}}) \leq \delta$, we add it to $S_{\phi+1}$ and we update $L_{\phi+1}$ to keep the invariant that no large dimension has width larger than δ.

ALGORITHM 5: CorPV.EpochUpdates

Parameters: $\mathcal{K}_\phi, S_\phi, L_\phi, \widetilde{\mathbf{h}}, \widetilde{\omega}$
$\mathcal{K}_{\phi+1} \leftarrow \mathcal{K}_\phi \bigcap \mathbf{H}^+(\widetilde{\mathbf{h}}, \widetilde{\omega})$
Save temporary sets $S' \leftarrow S_\phi$ and $L' \leftarrow L_\phi$.
if $w\left(\Pi_{L_\phi}\mathcal{K}_{\phi+1}, \widetilde{\mathbf{h}}\right) \leq \delta$ **then** ▷ size of small dimensions $\delta := \frac{\varepsilon}{4(d+\sqrt{d})}$
    Add hyperplane to small dimensions $S' \leftarrow S_\phi \bigcup \left\{\widetilde{\mathbf{h}}\right\}$.
    Compute orthonormal basis for new large dimensions L' (without S').
Update $L_{\phi+1} \leftarrow L' \setminus \{e_i \in L' : w(\mathcal{K}_{\phi+1}, e_i) \leq \delta\}$ and $S_{\phi+1} \leftarrow S' \bigcup (L' \setminus L_{\phi+1})$.
return $(\mathcal{K}_{\phi+1}, S_{\phi+1}, L_{\phi+1})$

Overall, the potential function we use to make sure that we make progress depends on the projected volume of the knowledge set on the large dimensions L_ϕ, as well as the number of small dimensions S_ϕ. This is why in lines 7- 8 of Algorithm 4, we update the epoch of less robust layers when one of these two measures of progress is triggered. Sets S_ϕ and L_ϕ serve in explaining which dimensions are identified well enough so that we can focus our attention on making progress in the remaining dimensions. For this to happen, an important notion is that of *Cylindrification* which creates a box covering the knowledge set and removes the significance of the small dimensions.

Definition 3.2 (Cylindrification, Definition 4.1 of [LPLV18]). *Given a set of orthonormal vectors* $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, *let* $L = \{\mathbf{u} | \langle \mathbf{u}, \mathbf{s} \rangle = 0; \forall \mathbf{s} \in S\}$ *be a subspace orthogonal to* $\texttt{span}(S)$ *and* $\Pi_L \mathcal{K}$ *be*

the projection of convex set $\mathcal{K} \subseteq \mathbb{R}^d$ onto L. We define:

$$\texttt{Cyl}(\mathcal{K}, S) := \left\{ \mathbf{z} + \sum_{i=1}^{n} b_i \mathbf{s}_i \,\middle|\, \mathbf{z} \in \Pi_L \mathcal{K} \text{ and } \min_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, \mathbf{s}_i \rangle \leq b_i \leq \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, \mathbf{s}_i \rangle \right\}.$$

By working with the Cylindrification $\texttt{Cyl}(\mathcal{K}_\phi, S_\phi)$ (Definition 3.2) rather than the original set of small dimensions S_ϕ, we can ensure that we make queries that make volumetric progress with respect to the large dimensions, that have been less well understood. This is also the reason why the landmark $\mathbf{p}^\star$ that we identify lives in the large dimension while also being close to the centroid $\boldsymbol{\kappa}_\phi$ (line 2).

Exploit queries for different loss functions. When the width of the knowledge set on the direction of the incoming context is small, i.e., $w(\mathcal{K}_\phi, \mathbf{x}_t) \leq \varepsilon$, we proceed with an exploit query. This module evaluates the loss of each query ω with respect to any parameter that is consistent with the knowledge set, i.e., $\boldsymbol{\theta}^\star \in \mathcal{K}_\phi$. It then employs a min-max approach by selecting the query ω_t that has the minimum loss for the worst-case selection of $\boldsymbol{\theta} \in \mathcal{K}_\phi$.

ALGORITHM 6: CorPV.Exploit

1 **Parameters:** $\mathbf{x}_t$, $\mathcal{K}_\phi$
2 Compute query point
$\omega_t = \min_{\omega \in \Omega} \max_{\boldsymbol{\theta} \in \mathcal{K}_\phi} \ell(\omega, \langle \boldsymbol{\theta}, \mathbf{x}_t \rangle, \langle \boldsymbol{\theta}, \mathbf{x}_t \rangle)$
3 **return** ω_t

For the ε-ball loss, any query point $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta}' \rangle$ with $\boldsymbol{\theta}' \in \mathcal{K}_\phi$ results to loss equal to 0; this is what ProjectedVolume also does to achieve optimal regret for the ε-ball loss function.

Moving to the pricing loss and assuming that the query point is $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta} \rangle$ for some $\boldsymbol{\theta} \in \mathcal{K}_\phi$, although the distance of $\boldsymbol{\theta}^\star$ to hyperplane $(\mathbf{x}_t, \omega_t \rangle)$ is less than ε, there is a big difference based on which side of the hyperplane $\boldsymbol{\theta}^\star$ lies in (i.e., whether $\boldsymbol{\theta}^\star \in \mathbf{H}^+(\mathbf{x}_t, \omega_t)$ or not). Specifically, if $\omega_t > \langle \mathbf{x}_t, \boldsymbol{\theta}_t^\star \rangle$ then a fully rational agent does not buy and we get zero revenue, thereby incurring a loss of $\langle \mathbf{x}_t, \boldsymbol{\theta}^\star \rangle$. On the other hand, querying $\omega^\star = \langle \mathbf{x}_t, \boldsymbol{\theta}^\star \rangle$ would lead to a purchase from a fully rational agent, and hence, to a pricing loss of 0. [6]

One way to deal with this discontinuity is by querying point ω_t with $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\theta}^\star \rangle - \varepsilon$, as the value of the fully rational agent is certainly above this price. As we formalize in Section 5, for the behavioral model of bounded rationality, even if we know $\boldsymbol{\theta}^\star$, for the pricing loss, we should select a slightly smaller price to account for the noise and the definition of ω_t takes into account the distributional information about the noise of the behavioral model.

4 Analysis

In this section we provide the analysis of the algorithm introduced in Section 3.

4.1 Existence of a separating hyperplane at the end of any epoch

In this subsection, we prove that after $\tau = 2d \cdot \bar{c}(d+1) + 1$ rounds, there exist $\mathbf{h}_\phi^\star \in \mathbb{R}^d$ and $\omega_\phi^\star \in \mathbb{R}$ such that the hyperplane $\left(\mathbf{h}_\phi^\star, \omega_\phi^\star\right)$ is a separating cut, i.e., it passes close to the approximate centroid $\boldsymbol{\kappa}_\phi$ (and therefore also to the centroid $\boldsymbol{\kappa}_\phi^\star$), and it has undesirability at least $\bar{c} + 1$ in the entirety of one of its halfspaces. Formally, this is stated in Lemma 4.1. The results of this subsection hold for *any* scalar $\delta < \frac{\varepsilon}{2\sqrt{d}+4d}$.

[6] This discontinuity in pricing loss poses further complications in extending other algorithms to the contextual search problem as we discuss in Section 6.

Easing notation. For the analysis, we make three simplifications (all without loss of generality) in an effort to ease the notation. First, we assume that $y_t = +1, \forall t \in [T]$. This is indeed without loss of generality since the algorithm can always negate the received context $\mathbf{x}_t$ and the chosen query ω_t to force $y_t = +1$ (Step 3 of Algorithm 2). Second, for rounds where the nature's answer is arbitrary, we assume that the perceived value is $\widetilde{v}_t = \langle \mathbf{x}_t, \boldsymbol{\theta}_t \rangle$, where $\boldsymbol{\theta}_t \in \mathcal{K}_0$ and it can change from round to round. For all other rounds $\boldsymbol{\theta}_t = \boldsymbol{\theta}^\star$. Third, we assume that all hyperplanes have unit ℓ_2 norm.

Lemma 4.1. For any epoch ϕ, scalar $\delta \in \left(0, \frac{\varepsilon}{2\sqrt{d}+4d}\right)$, and scalar $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$, after $\tau = 2d \cdot \bar{c}(d+1)+1$ rounds, there exists a hyperplane $(\mathbf{h}_\phi^\star, \omega_\phi^\star)$ *orthogonal to all small dimensions* S_ϕ such that the resulting halfspace $\mathbf{H}^+(\mathbf{h}_\phi^\star, \omega_\phi^\star)$ always contains $\boldsymbol{\theta}^\star$ and $\texttt{dist}(\boldsymbol{\kappa}_\phi, (\mathbf{h}_\phi^\star, \omega_\phi^\star)) \leq \bar{\nu}$, where by $\texttt{dist}(\boldsymbol{\kappa}, (\mathbf{h}, \omega))$ we denote the distance of point $\boldsymbol{\kappa}$ from hyperplane $(\mathbf{h}, \omega)$, i.e., $\texttt{dist}(\boldsymbol{\kappa}, (\mathbf{h}, \omega)) = \frac{|\langle \boldsymbol{\kappa}, \mathbf{h}\rangle - \omega|}{\|\mathbf{h}\|}$.

At a high level, the tuning of $\bar{\nu}$ depends on two factors. First, in order to make sure that we make enough progress in terms of volume elimination, despite the fact that we do not make a cut through $\boldsymbol{\kappa}_\phi$, we need $\bar{\nu}$ to be close enough to $\boldsymbol{\kappa}_\phi$ (Lemma B.5). Second, we need to guarantee that there exists at least one point with a very high undesirability level (Lemma 4.8). For the analysis, we define the ν-margin projected undesirability levels, which we later use for some fixed $\nu < \bar{\nu}$:

Definition 4.2 (ν-Margin Projected Undesirability Level)**.** *Consider an epoch ϕ, a scalar ν, and a point $\mathbf{p}$ in $\mathcal{K}_\phi$. Given the set $\mathcal{A}_\phi = \{(\Pi_{L_\phi}\mathbf{x}_t, \omega_t)\}_{t\in[\tau]}$, we define $\mathbf{p}$'s* ν-margin projected undesirability level*, denoted by $u_\phi(\mathbf{p}, \nu)$, as the number of rounds within epoch ϕ, for which*

$$u_\phi(\mathbf{p}, \nu) = \sum_{t\in[\tau]} \mathbb{1}\left\{\left(\left\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t \right\rangle + \nu\right) < 0\right\}$$

Intuitively, $u_\phi(\mathbf{p}, \nu)$ gives penalty to a point $\mathbf{p}$ if it is far (more than ν) from the negative halfspace of the query (when projected to the large dimensions L_ϕ).

Using that $y_t = +1, \forall t \in [T]$, we obtain the next lemma, which at a high level states that the undesirability level of a point $\mathbf{p}$ during an epoch ϕ corresponds to the number of times during epoch ϕ that $\mathbf{p}$ and $\boldsymbol{\theta}_t$ were at opposite sides of hyperplane $(\Pi_{L_\phi}\mathbf{x}_t, \nu + \omega_t)$ for any $\nu > \bar{\nu}$.

Lemma 4.3. If $\nu > \underline{\nu}$, where $\underline{\nu} = \sqrt{d}\delta$, for any point $\mathbf{p}$, we have that:

$$u_\phi(\mathbf{p}, \nu) = \sum_{t\in[\tau]} \mathbb{1}\left\{\left(\left\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t \right\rangle + \nu\right) \cdot \left(\left\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t \right\rangle + \nu\right) < 0\right\}$$

Proof. In order to prove the lemma, we argue that $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \nu > 0$. Recall that the feedback in Step 3 of the protocol is defined as $\text{sgn}(\widetilde{v}_t - \omega_t)$, and that we set $y_t = +1$ and $\widetilde{v}_t = \langle \mathbf{x}_t, \theta_t \rangle$. As a result, $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \mathbf{x}_t\rangle \geq 0$ and expanding:

$$\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{S_\phi}\mathbf{x}_t\rangle \geq 0 \tag{2}$$

We proceed by upper bounding the quantity $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{S_\phi}\mathbf{x}_t\rangle$. Let S be a matrix with columns

corresponding to the basis of vectors in S_ϕ, so that $\Pi_{S_\phi} = SS^\top$. Then, we obtain:

$$\begin{aligned}
\langle \Pi_{S_\phi}\mathbf{x}_t, \mathbf{p} - \boldsymbol{\kappa}_\phi\rangle &= \langle \Pi_{S_\phi}\mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\rangle \leq |\langle \Pi_{S_\phi}\mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\rangle| \\
&\leq \|\Pi_{S_\phi}\mathbf{x}_t\|_2 \cdot \|\Pi_{S_\phi}(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 && \text{(Cauchy-Schwarz inequality)} \\
&= \|\Pi_{S_\phi}\mathbf{x}_t\|_2 \cdot \|S^\top(\mathbf{p} - \boldsymbol{\kappa}_\phi)\|_2 \\
&\leq \|\mathbf{x}_t\|_2 \cdot \sqrt{d} \cdot \|S^\top(\mathbf{p} - \kappa_\phi)\|_\infty && (\|\mathbf{z}\|_2 = \sqrt{\textstyle\sum_{i\in[d]} \mathbf{z}_i^2} \leq \sqrt{d \cdot \|\mathbf{z}\|_\infty^2}) \\
&\leq 1 \cdot \delta\sqrt{d}. && (\|\mathbf{x}_t\|_2 = 1 \text{ and } w(\mathcal{K}_\phi, \mathbf{s}) \leq \delta, \forall \mathbf{s} \in S_\phi)
\end{aligned}$$

Using this to relax Equation (2) along with $\underline{\nu} = \sqrt{d}\delta$ we get that: $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle \geq -\underline{\nu}$. Since $\nu > \underline{\nu}$, it follows that $\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \nu \geq -\underline{\nu} + \nu > 0$. Combined with Definition 4.2, this concludes the lemma. ■

We define the $\bar{c}$-protected region in large dimensions, $\mathcal{P}(\bar{c}, \nu)$, which is the set of points in $\mathcal{K}_\phi$ with ν-margin projected undesirability level at most $\bar{c}$. Mathematically:

$$\mathcal{P}(\bar{c}, \nu) = \{\mathbf{p} \in \mathcal{K}_\phi : u_\phi(\mathbf{p}, \nu) \leq \bar{c}\}$$

The next lemma establishes that if we keep set $\mathcal{P}(\bar{c}, \nu)$ intact in the convex body formed for the next epoch $\mathcal{K}_{\phi+1}$, then we are guaranteed to not eliminate point $\boldsymbol{\theta}^\star$.

Lemma 4.4. If $\nu > \underline{\nu}$, where $\underline{\nu} = \sqrt{d}\delta$, the ground truth $\boldsymbol{\theta}^\star$ belongs in set $\mathcal{P}(\bar{c}, \nu)$.

Proof. By Lemma 4.3, $u_\phi(\mathbf{p}, \nu) = \sum_{t\in[\tau]} \mathbb{1}\left\{\left(\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \nu\right) \cdot \left(\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \nu\right) < 0\right\}$. For the uncorrupted rounds $\boldsymbol{\theta}^\star = \mathbf{p} = \boldsymbol{\theta}_t$; as a result, the corresponding summands are non-negative: $\left(\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \nu\right) \cdot \left(\langle \boldsymbol{\theta}_t - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\rangle + \nu\right) \geq 0$. Hence, the only rounds for which $\boldsymbol{\theta}^\star$ can incur undesirability are the corrupted rounds, of which there are at most $\bar{c}$. As a result, $u_\phi(\boldsymbol{\theta}^\star, \nu) \leq \bar{c}$ and $\boldsymbol{\theta}^\star \in \mathcal{P}(\bar{c}, \nu)$ by the definition of region $\mathcal{P}(\bar{c}, \nu)$. ■

We next show that there exists a hyperplane cut, that is orthogonal to all small dimensions in a way that guarantees that set $\mathcal{P}(\bar{c}, \nu)$ is *preserved* in $\mathcal{K}_{\phi+1}$ (i.e., $\mathcal{P}(\bar{c}, \nu) \subseteq \mathcal{K}_{\phi+1}$). Note that due to Lemma 4.4, it is enough to guarantee that we have $\boldsymbol{\theta}^\star \in \mathcal{K}_{\phi+1}$. However, $\mathcal{P}(\bar{c}, \nu)$ is generally non-convex and it is not easy to directly make claims about it. Instead, we focus on its convex hull, denoted by $\texttt{conv}(\mathcal{P}(\bar{c}, \nu))$; for any point in $\texttt{conv}(\mathcal{P}(\bar{c}, \nu))$ we can upper bound its undesirability by applying Carathéodory's Theorem, which says that any point in the convex hull of a (possibly non-convex) set can be written as a convex combination of at most $d + 1$ points of that set. Using this result, we can bound the ν-margin projected undesirability levels of all the points in $\texttt{conv}(\mathcal{P}(\bar{c}, \nu))$.

Lemma 4.5. For any scalar ν, epoch ϕ and any point $\mathbf{p} \in \texttt{conv}(\mathcal{P}(\bar{c}, \nu))$, its ν-margin projected undesirability level is *at most* $\bar{c} \cdot (d + 1)$, i.e., $u_\phi(\mathbf{p}, \nu) \leq \bar{c} \cdot (d + 1)$.

Proof. From Carathéodory's Theorem, since $\mathbf{p} \in \mathbb{R}^d$ and is inside $\texttt{conv}(\mathcal{P}(\bar{c}, \nu))$, it can be written as the convex combination of *at most* $d+1$ points in $\mathcal{P}(\bar{c}, \nu)$. Denoting these points by $\{\mathbf{z}_1, \ldots, \mathbf{z}_{d+1}\}$ such that $\mathbf{z}_i \in \mathcal{P}(\bar{c}, \nu), \forall i \in [d + 1]$, $\mathbf{p}$ can be written as $\mathbf{p} = \sum_{i=1}^{d+1} a_i\mathbf{z}_i$ where $a_i \geq 0, \forall i \in [d + 1]$

and $\sum_{i=1}^{d+1} a_i = 1$. Hence, the ν-margin projected undesirability level of $\mathbf{p}$ in epoch ϕ is:

$$
\begin{aligned}
u_\phi(\mathbf{p}, \nu) &= \sum_{t\in[\tau]} \mathbb{1}\left\{\left(\left\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\right\rangle + \nu\right) < 0\right\} && \text{(Definition 4.2)}\\
&= \sum_{t\in[\tau]} \mathbb{1}\left\{ \sum_{i\in[d+1]} a_i \underbrace{\left(\left\langle \mathbf{z}_i - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\right\rangle + \nu\right)}_{Q_i} < 0 \right\} && \text{(Carathéodory's Theorem)}\\
&\leq \sum_{t\in[\tau]} \sum_{i\in[d+1]} \mathbb{1}\left\{\left(\left\langle \mathbf{z}_i - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\right\rangle + \nu\right) < 0\right\}\\
&\leq \sum_{i\in[d+1]} u_\phi(\mathbf{z}_i, \nu) \leq \bar{c}\cdot(d+1) && (\mathbf{z}_i \in \mathcal{P}(\bar{c},\nu) \text{ and definition of } \mathcal{P}(\bar{c},\nu))
\end{aligned}
$$

where the first inequality comes from the fact that if $Q_i \geq 0$ for *all* $\mathbf{z}_i, i \in [d+1]$, then the corresponding summand contributes 0 undesirability points to $u_\phi(\mathbf{p}, \nu)$, since $a_i \geq 0$ as this is a convex combination. As a result, each undesirability point on the left hand side of the latter inequality can be attributed to at least one $\mathbf{z}_i$ from the right hand side. ■

Next, we prove that there exists some point $\mathbf{q} \in \mathcal{K}_\phi$ such that $u_\phi(\mathbf{q}, \nu) \geq \bar{c}\cdot(d+1)+1$. Note that by the previous lemma, we know that $\mathbf{q} \notin \texttt{conv}(\mathcal{P}(\bar{c},\nu))$. As a result, *any* hyperplane separating $\mathbf{q}$ from $\texttt{conv}(\mathcal{P}(\bar{c},\nu))$ preserves $\mathcal{P}(\bar{c},\nu)$ (and as a result, $\boldsymbol{\theta}^\star$) for $\mathcal{K}_{\phi+1}$. To make sure that we also make progress in terms of volume elimination, we show below that there exists a separating hyperplane in the space of large dimensions (i.e., orthogonal to all small dimensions). For our analysis, we introduce the notion of *landmarks*.

Definition 4.6 (Landmarks)**.** *Let basis* $E_\phi = \{e_1, \ldots, e_{d-|S_\phi|}\}$ *such that* E_ϕ *is orthogonal to* S_ϕ*, any scalar* $\delta \in \left(0, \frac{\varepsilon}{2\sqrt{d}+4d}\right)$*, and a scalar* $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$*. We define the* $2(d-|S_\phi|)$ landmarks *to be the points such that* $\Lambda_\phi = \{\boldsymbol{\kappa}_\phi \pm \bar{\nu}\cdot e_i, \forall e_i \in E_\phi\}$.

Landmarks possess the convenient property that at every round where the observed context $\mathbf{x}_t$ was such that $w(\mathcal{K}_\phi, \mathbf{x}_t) \geq \varepsilon$, at least one of them gets a ν-margin projected undesirability point, when $\nu < \bar{\nu}$. Before proving this result (formally in Lemma 4.8) we need the following technical lemma, whose proof follows ideas from [LPLV18] and is provided in Appendix B.1 for completeness.

Lemma 4.7. Let basis $E_\phi = \{e_1, \ldots, e_{d-|S_\phi|}\}$ orthogonal to S_ϕ. For all $\{(\mathbf{x}_t, \omega_t)\}_{t\in[\tau]}$ such that $w(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \geq \varepsilon$, there exists i such that: $|\langle e_i, \mathbf{x}_t\rangle| \geq \bar{\nu}$, where $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$.

The tuning of $\bar{\nu}$ explains the constraint imposed on δ, i.e., $\delta < \frac{\varepsilon}{2\sqrt{d}+4\delta}$. This constraint is due to the fact that since $\nu > \underline{\nu}$ and $\nu < \bar{\nu}$, then it must be the case that $\underline{\nu} < \bar{\nu}$, where $\underline{\nu} = \sqrt{d}\delta$ and $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}}{4\sqrt{d}}$.

Lemma 4.8. For every round $t \in [\tau]$, any scalar $\delta \in (0, \frac{\varepsilon}{2\sqrt{d}+4d})$, any scalar $\nu < \bar{\nu}$, at least one of the landmarks in Λ_ϕ gets one ν-margin projected undesirability point, i.e.,

$$\exists \mathbf{p} \in \Lambda_\phi : \left(\left\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi}\mathbf{x}_t\right\rangle + \nu\right) < 0.$$

Proof. By Lemma 4.7, there exists a direction $e_i \in E$ such that $|\langle e_i, \mathbf{x}_t \rangle| \geq \bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$. The proof then follows by showing that for $\nu < \bar{\nu}$ landmark points $\mathbf{q}_+ = \boldsymbol{\kappa}_\phi + \nu \cdot e_i$ and $\mathbf{q}_- = \boldsymbol{\kappa}_\phi - \nu \cdot e_i$ get different signs in the undesirability point definition. This is shown by the following derivation:

$$\begin{aligned}
&\left(\left\langle \mathbf{q}_+ - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \right\rangle + \nu\right) \cdot \left(\left\langle \mathbf{q}_- - \boldsymbol{\kappa}_\phi, \Pi_{L_\phi} \mathbf{x}_t \right\rangle + \nu\right) \\
&= \left(\left\langle \bar{\nu} \cdot e_i, \Pi_{L_\phi} \mathbf{x}_t \right\rangle + \nu\right) \cdot \left(\left\langle -\bar{\nu} \cdot e_i, \Pi_{L_\phi} \mathbf{x}_t \right\rangle + \nu\right) \\
&= \nu^2 - \left(\bar{\nu} \cdot |\langle e_i, \mathbf{x}_t \rangle|\right)^2 \leq \nu^2 - \bar{\nu}^2 < 0
\end{aligned}$$

where the last inequality comes from the fact that $\nu \in (\underline{\nu}, \bar{\nu})$. As a result there exists $\mathbf{p} \in \{\mathbf{q}_+, \mathbf{q}_-\} \subseteq \mathcal{L}_\phi$ satisfying the condition in the lemma statement. ■

Since Lemma 4.8 establishes that at every round at least one of the landmarks gets a ν-margin projected undesirability point, then if we make τ sufficiently large, then, by the *pigeonhole principle*, at least one of the landmarks has ν-margin projected undesirability at least $\bar{c} \cdot (d+1) + 1$, which allows us to distinguish it from points in $\texttt{conv}(\mathcal{P}(\bar{c}))$. Formally:

Lemma 4.9. For scalar $\nu \in (\underline{\nu}, \bar{\nu})$, after $\tau = 2d \cdot \bar{c} \cdot (d+1) + 1$ rounds in epoch ϕ, there exists a landmark $\mathbf{p}^\star \in \Lambda_\phi$ such that $\mathbf{p}^\star \notin \texttt{conv}(\mathcal{P}(\bar{c}, \nu))$.

Proof. At each of the τ explore rounds, at least one of the landmarks gets a ν-margin projected undesirability point (Lemma 4.8). Since there are *at most* $2d$ landmarks, by the pigeonhole principle after τ rounds, there exists at least one of them with ν-margin projected undesirability $u_\phi(\mathbf{p}^\star, \nu) \geq \bar{c} \cdot (d+1) + 1$. Since all points $\mathbf{q}$ inside $\texttt{conv}(\mathcal{P}(\bar{c}, \nu))$ have $u_\phi(\mathbf{q}, \nu) \leq \bar{c} \cdot (d+1)$, then $\mathbf{p} \notin \texttt{conv}(\mathcal{P}(\bar{c}, \nu))$. ■

We are now ready to prove the main lemma of this subsection. We note that during the computation of $\mathbf{h}^\star_\phi$, the nature does not provide any new context $\mathbf{x}$, and hence, we incur no additional regret.

Proof of Lemma 4.1. By Lemma 4.9, for $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$ and $\delta \in \left(0, \frac{\varepsilon}{2\sqrt{d}+4d}\right)$, there exists a landmark $\mathbf{p}^\star \in \Lambda_\phi$ that lies outside of $\texttt{conv}(\mathcal{P}(\bar{c}, \nu))$. As a result, there exists a hyperplane separating $\mathbf{p}^\star$ from the convex hull. We denote this hyperplane by $(\mathbf{h}^\star_\phi, \omega^\star_\phi)$. Recall that since $\mathbf{p}^\star \in \Lambda_\phi$ then by definition $\|\boldsymbol{\kappa}_\phi - \mathbf{p}^\star\| = \bar{\nu}$. As the hyperplane separates $\boldsymbol{\kappa}_\phi$ from $\mathbf{p}^\star$, we immediately have that $\texttt{dist}(\boldsymbol{\kappa}_\phi, (\mathbf{h}^\star_\phi, \omega^\star_\phi)) \leq \bar{\nu}$ as well. The fact that $\boldsymbol{\theta}^\star$ is always in the preserved halfspace $\mathbf{H}_\phi(\mathbf{h}^\star_\phi, \omega^\star_\phi)$ follows directly from Lemma 4.4. ■

4.2 Key proposition for the known-corruption setting

In this subsection, we analyze the result for the intermediate $\bar{c}$-known corruption setting. Formally:

Proposition 4.10. For the $\bar{c}$-known-corruption setting, the regret of CorPV.Known for the ε-ball loss is $\mathcal{O}\left((d^2\bar{c}+1)d\log\left(d/\varepsilon\right)\right)$. When run with parameter $\varepsilon = 1/T$, its guarantee for the absolute and pricing loss is $\mathcal{O}\left((d^2\bar{c}+1)d\log\left(dT\right)\right)$. The expected runtime is $\mathcal{O}((d^2\bar{c})^{\bar{c}} \cdot \text{poly}(d\log(d/\varepsilon), \bar{c}))$

Before delving into the details, we make two remarks. First, the regret guarantee of Proposition 4.10 is *deterministic*; only the runtime is randomized. Second, although the expected runtime is exponential in $\bar{c}$, in Section 4.3 the algorithm is run with $\bar{c} \approx \log(T)$, which renders it quasipolynomial.

The first step in proving Proposition 4.10 is to analyze CorPV.SeparatingCut (Algorithm 3), which is formally stated in Lemma 4.11. For what follows, let $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$ be the ball of radius ζ

around $\mathbf{p}^\star$ in the space of large dimensions, where $\mathbf{p}^\star \in \Lambda_\phi$ is the landmark such that $u_\phi(\mathbf{p}^\star, \nu) = \bar{c} \cdot (d+1) + 1$. Recall that we proved the existence of a landmark $\mathbf{p}^\star \in \Lambda_\phi$ with this property in Lemma 4.9.

Lemma 4.11. For any epoch ϕ, scalar $\delta \in (0, \frac{\varepsilon}{2\sqrt{d}+4d})$, and scalar $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$, after $\tau = 2d \cdot \bar{c}(d+1)+1$ rounds, algorithm CorPV.SeparatingCut computes hyperplane $(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ *orthogonal to all small dimensions* S_ϕ such that $\texttt{dist}(\boldsymbol{\kappa}^\star_\phi, (\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)) \leq 3\bar{\nu}$, and the resulting halfspace $\mathbf{H}^+(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ always contains $\boldsymbol{\theta}^\star$. With probability *at least* $(40d \cdot \sqrt{d-1})^{-1}$ the complexity of this computation is:

$$\mathcal{O}\left(\frac{(d-1)}{\bar{\nu}^2} \cdot \left(d^2 \cdot \bar{c}\right)^{\bar{c}} \cdot O(\mathrm{CP}(d, \bar{c}\cdot(2d(d+1)-1)+1))\right)$$

where $O(\mathrm{CP}(n,m))$ denotes the computational complexity of solving a Convex Program (CP) with n variables and m constraints.

Proof. From Lemma 4.1, there exists a hyperplane $(\mathbf{h}^\star_\phi, \omega^\star_\phi)$ with distance at most $\bar{\nu}$ from $\boldsymbol{\kappa}_\phi$ that has all of $\mathcal{P}(\bar{c}, \nu)$ inside $\mathbf{H}^+(\mathbf{h}^\star_\phi, \omega^\star_\phi)$. By arguing about the volume contained in any specified "cap" of a multi-dimensional ball, we prove that with probability at least $(20\sqrt{d-1})^{-1}$ we can identify a point $\mathbf{q}$ lying on the halfspace

$$\mathbf{H}^+\left(\mathbf{h}^\star_\phi, \left\langle \mathbf{h}^\star_\phi, \mathbf{p}^\star \right\rangle + \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}}\right).$$

This is formally stated and proved in Appendix B.2. In each iteration of CorPV.SeparatingCut using point $\mathbf{q}$ (i.e., lines 6–13 in Algorithm 3), Perceptron can identify a hyperplane $(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ separating $\mathbf{q}$ and $\mathcal{P}(\bar{c}, \nu)$ after $\frac{d-1}{\zeta^2 \cdot \ln^2(3/2)}$ samples. The number of samples depends on the Perceptron mistake bound, as mentioned in Section 3.1. Since $\mathbf{q} \in \mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$, then,

$$\begin{aligned}
\|\mathbf{q} - \boldsymbol{\kappa}^\star_\phi\| &= \|\mathbf{q} - \boldsymbol{\kappa}_\phi + \boldsymbol{\kappa}_\phi - \boldsymbol{\kappa}^\star_\phi\| \\
&\leq \|\mathbf{q} - \boldsymbol{\kappa}_\phi\| + \|\boldsymbol{\kappa}_\phi - \boldsymbol{\kappa}^\star_\phi\| && \text{(triangle inequality)} \\
&\leq \|\mathbf{q} - \boldsymbol{\kappa}_\phi\| + \bar{\nu} && \text{(approximation of } \boldsymbol{\kappa}^\star_\phi \text{ in polynomial time)} \\
&= \|\mathbf{q} - \mathbf{p}^\star + \mathbf{p}^\star - \boldsymbol{\kappa}_\phi\| + \bar{\nu} \\
&\leq \|\mathbf{q} - \mathbf{p}^\star\| + \|\mathbf{p}^\star - \boldsymbol{\kappa}_\phi\| + \bar{\nu} && \text{(triangle inequality)} \\
&\leq \bar{\nu} + \bar{\nu} + \bar{\nu} && (\mathbf{q} \in \mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta) \text{ and Definition 4.6})
\end{aligned}$$

Hence, $\texttt{dist}(\boldsymbol{\kappa}^\star_\phi, (\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)) \leq 3\bar{\nu}$.

Assume for now that the random landmark $\mathbf{p}^\star \in \Lambda_\phi$ chosen at Step 5 of Algorithm 3 is the landmark such that $u_\phi(\mathbf{p}^\star, \nu) \geq \bar{c} \cdot (d+1) + 1$. Because $\mathbf{p}^\star$ is chosen uniformly at random from set Λ_ϕ, then the probability of it being the target landmark is $(2d)^{-1}$, and is independent with all other random variables of Algorithm 3.

At every iteration of the inner while-loop of Algorithm 3, the algorithm checks whether $\mathbf{q}$ and $\boldsymbol{\kappa}_\phi$ are on the "correct" side of $\mathbf{h}$ and possibly updates the Perceptron (this is done in time $\mathcal{O}(1)$): the centroid $\boldsymbol{\kappa}_\phi$ needs to be part of the $\mathcal{P}(\bar{c}, \nu)$, while $\mathbf{q}$ needs to be separated from $\mathcal{P}(\bar{c}, \nu)$. Thus, they must belong to $\mathbf{H}^+(\widetilde{\mathbf{h}}, \widetilde{\omega})$ and $\mathbf{H}^-(\widetilde{\mathbf{h}}, \widetilde{\omega})$ respectively. This is done in Steps 8–9 of Algorithm 3. The

most computationally demanding part of Algorithm 3 is Steps 9–12, where we check whether the hyperplane cuts some part of $\mathcal{P}(\bar{c}, \nu)$. For that, we check all the possible

$$\binom{|\mathcal{A}_\phi|}{|D_\phi|} = \binom{|\mathcal{A}_\phi|}{\bar{c}} \leq (2d \cdot \bar{c}(d+1)+1)^{\bar{c}} = \Theta\left(\left(d^2\bar{c}\right)^{\bar{c}}\right)$$

combinations of which $\bar{c}$ hyperplanes to disregard as corrupted and solve a mathematical program which we refer to as CP [7] for each such combination. We remark that this computation serves the purpose of identifying which $\bar{c}$ hyperplanes in $\mathcal{A}_\phi$ were corrupted, thus giving erroneous feedback regarding where $\boldsymbol{\theta}^\star$ lies. These CPs have at most d variables (since $\mathcal{K}_\phi \subseteq \mathbb{R}^d$) and $|\mathcal{A}_\phi| - \bar{c} = \bar{c} \cdot (2d(d+1)-1)+1)$ constraints. Denoting the complexity of solving an LP with n variables and m constraints as $O(\mathrm{LP}(n,m))$ we therefore obtain that Steps 9–12 have computational complexity

$$\mathcal{O}\left(O(\mathrm{LP}(d, \bar{c} \cdot (2d(d+1)-1)+1))) \cdot \left(d^2\bar{c}\right)^{\bar{c}}\right).$$

Putting everything together, and given that the event that $\mathbf{p}^\star$ is the target landmark is independent from the event that $\widetilde{\mathbf{q}}$ is found at the desired halfspace we have that with probability at least

$$\frac{1}{2d} \cdot \frac{1}{20\sqrt{d-1}}$$

the computational complexity of Algorithm 3 is:

$$\mathcal{O}\left(\frac{(d-1)}{\bar{\nu}^2} \cdot \left(d^2 \cdot \bar{c}\right)^{\bar{c}} \cdot O(\mathrm{CP}(d, \bar{c} \cdot (2d(d+1)-1)+1))\right)$$

This concludes our proof. ■

The second step in proving Proposition 4.10 is to establish that we make enough volumetric progress when using $(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ as our separating cut for epoch ϕ. We remark that in the analysis of [LPLV18], when ProjectedVolume observes a context $\mathbf{x}_t$ such that $w(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \leq \varepsilon$, then it can directly discard it, since $\mathbf{x}_t$ does not contribute to the regret with respect to the ε-ball loss function. This is because $\mathbf{x}_t$'s are used in order to make the separating cuts. In our epoch-based setting, the separating cuts are different than the observed contexts, as we have argued. Importantly, if $w(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi), \widetilde{\mathbf{h}}_\phi) \leq \varepsilon$, we cannot relate this information to the regret of epoch ϕ, because for all rounds comprising the epoch, the width of $\mathcal{K}_\phi$ in the direction of the observed context was greater than ε (Step 7 of Algorithm 1). The auxiliary lemmas that we use can be found in Appendix B.3.

Lemma 4.12. After at most $\Phi = O(d \log(d/\varepsilon))$ epochs, CorPV.Known (Algorithm 1) has reached a knowledge set $\mathcal{K}_\Phi$ with width at most ε in *every* direction $\mathbf{u}$.

Proof. To prove this lemma, we construct a potential function argument, similar to the one of [LPLV18]; we highlight, however, the places where our analysis differs from theirs.

We use $\Gamma_\phi = \texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_\phi\right)$ as our potential function. Its lower bound is: $\Gamma_\phi \geq \Omega\left(\frac{\delta}{d}\right)^{2d}$. To see this, note that Step 7 of CorPV.EpochUpdates (Algorithm 5) ensures that for all $\mathbf{u} \in L_\phi$ it holds that $w\left(\Pi_{L_\phi}\mathcal{K}_\phi, \mathbf{u}\right) \geq \delta$. It is known (see [LPLV18, Lemma 6.3] or Lemma B.3) that if $\mathcal{K} \subseteq \mathbb{R}^d$ is a

[7] Technically this program is convex and not linear as we also take intersection with the $\mathcal{K}_\phi$ which is a convex body.

convex body such that $w(\mathcal{K}, \mathbf{u}) \geq \delta$ for every unit vector $\mathbf{u}$, then $\mathcal{K}$ contains a ball of diameter δ/d. This means that $\Pi_{L_\phi}\mathcal{K}_\phi$ contains a ball of radius $\frac{\delta}{|L_\phi|}$, so

$$\texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_\phi\right) \geq V(|L_\phi|)\left(\frac{\delta}{|L_\phi|}\right)^{|L_\phi|},$$

where by $V(|L_\phi|)$ we denote the volume of the $|L_\phi|$-dimensional unit ball. Using that $|L_\phi| \leq d$ and $V(d) \geq \Omega\left(\frac{1}{d}\right)^d$, the latter can be lower bounded by $\Omega\left(\frac{\delta}{d}\right)^{2d}$. Hence, $\Gamma_\phi = \texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_\phi\right) \geq \Omega\left(\frac{\delta}{d}\right)^{2d}$.

We split our analysis of the upper bound of Γ_ϕ in two parts. In the first part, we study the potential function between epochs where the set of large dimensions L_ϕ does not change. In the second part, we study the potential function between where the set of large dimensions L_ϕ becomes smaller. For both cases, we prove the following useful result (Lemma B.9) which relates the volume of $\Pi_{L_\phi}\mathcal{K}_{\phi+1}$ with the volume of $\Pi_{L_\phi}\mathcal{K}_\phi$ when $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$:

$$\texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_{\phi+1}\right) \leq \left(1 - \frac{1}{2e^2}\right)\texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_\phi\right) \tag{3}$$

When the set L_ϕ does not change between epochs $\phi, \phi+1$ (i.e., $L_\phi = L_{\phi+1}$) Equation (3) becomes:

$$\texttt{vol}\left(\Pi_{L_{\phi+1}}\mathcal{K}_{\phi+1}\right) \leq \left(1 - \frac{1}{2e^2}\right)\texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_\phi\right) \tag{4}$$

When L_ϕ does change, then the set of small dimensions increases from S_ϕ to $S_{\phi+1}$. In order to correlate $\texttt{vol}(\Pi_{L_{\phi+1}}\mathcal{K}_{\phi+1})$ with $\texttt{vol}(\Pi_{L_\phi}\mathcal{K}_\phi)$ we make use of the following known inequality ([LPLV18, Lemma 6.1] or Lemma B.8) that for a convex body $\mathcal{K} \subseteq \mathbb{R}^d$ if $w(\mathcal{K}, \mathbf{u}) \geq \delta'$ (for some scalar $\delta' > 0$, for *every* unit vector $\mathbf{u}$), then, for every $(d-1)-$dimensional subspace L it holds that:

$$\texttt{vol}\left(\Pi_L\mathcal{K}\right) \leq \frac{d(d+1)}{\delta'}\texttt{vol}(\mathcal{K}) \tag{5}$$

We are going to apply Equation (5) for $\mathcal{K} := \Pi_{L_\phi}\mathcal{K}_{\phi+1}$ and $L := L_{\phi+1}$. For that, we need to find δ' for which $w(\Pi_{L_\phi}\mathcal{K}) \geq \delta'$.

We make use of the following lemma ([LPLV18, Theorem 5.3] or Lemma B.4) which relates the width of the following convex body: $\mathcal{K}_+ = \mathcal{K} \bigcap \{\mathbf{x} | \langle \mathbf{u}, \mathbf{x} - \boldsymbol{\kappa} \rangle\}$ (where $\boldsymbol{\kappa}$ is the centroid of $\mathcal{K}$ and $\mathbf{u}$ is any unit vector) with the width of $\mathcal{K}$ in the direction of any unit vector $\mathbf{v}$ as follows:

$$\frac{1}{d+1}w(\mathcal{K}, \mathbf{v}) \leq w(\mathcal{K}_+, \mathbf{v}) \leq w(\mathcal{K}, \mathbf{v}) \tag{6}$$

By the definition of large dimensions, $w(\mathcal{K}_\phi, \mathbf{u}) \geq \delta, \forall \mathbf{u} \in L_\phi$. So, if we were to cut $\mathcal{K}_\phi$ with a hyperplane that passes precisely from the centroid $\boldsymbol{\kappa}^\star_\phi$ then, from Equation (6): $w(\mathcal{K}_+, \mathbf{u}) \geq \frac{\delta}{d+1}$. Since, however, we make sure that we cut $\mathcal{K}_\phi$ through the approximate centroid $\boldsymbol{\kappa}_\phi$, then $w(\mathcal{K}_+, \mathbf{u}) \geq \frac{\delta}{d+1}$. This is due to the fact that the halfspace $\mathbf{H}^+(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ returned from CorPV.SeparatingCut (Algorithm 3) always contains $\boldsymbol{\kappa}_\phi$. Since $\|\boldsymbol{\kappa}_\phi - \boldsymbol{\kappa}^\star_\phi\| \leq \bar{\nu}$ then $\boldsymbol{\kappa}^\star_\phi \in \mathbf{H}^+(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$. As a result, from Equation (5) we get:

$$\texttt{vol}\left(\Pi_{L_{\phi+1}}\mathcal{K}_{\phi+1}\right) \leq \frac{d(d+1)^2}{\delta}\texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_{\phi+1}\right) \tag{7}$$

We have almost obtained the target Equation (3). To complete the argument we need to correlate $\texttt{vol}(\Pi_{L_\phi}\mathcal{K}_{\phi+1})$ with $\texttt{vol}(\Pi_{L_\phi}\mathcal{K}_\phi)$. We prove the following inequality between the two (Lemma B.9):

$$\texttt{vol}\left(\Pi_{L_\phi}\mathcal{K}_{\phi+1}\right) \leq \left(1 - \frac{1}{2e^2}\right)\texttt{vol}(\Pi_{L_\phi}\mathcal{K}_\phi) \tag{8}$$

Combining Equations (7) and (8) and using the fact that we add at most d new directions to S_ϕ, that the volume of $\mathcal{K}_0$ is upper bounded by $O(1)$, and the lower bound for Γ_ϕ we computed, we have:

$$\Omega\left(\frac{\delta}{d}\right)^{2d} \leq \Gamma_\Phi = \texttt{vol}\left(\Pi_{L_\Phi}\mathcal{K}_\Phi\right) \leq O(1)\cdot\left(\frac{d(d+1)^2}{\delta}\right)^d \cdot \left(1-\frac{1}{2e^2}\right)^\Phi$$

where by Φ we denote the total number of epochs. Solving the above in terms of Φ and substituting δ we obtain: $\Phi \leq O(d\log\frac{d}{\varepsilon})$. ■

Proof of Proposition 4.10. We start by analyzing the regret for the ε-ball loss function. Lemma 4.12 establishes that after $\Phi = O(d\log(d/\varepsilon))$ epochs, the set of large dimensions L_ϕ is empty. When $L_\phi = \emptyset$, then S_ϕ must be an orthonormal basis for which $w(\mathcal{K}_\phi, \mathbf{s}) \leq \delta, \forall \mathbf{s} \in S_\phi$. For any received context $\mathbf{x}_t$ after $L_\phi = \emptyset$, we have the following. First, $\mathbf{x}_t = \sum_{i\in[|S_\phi|]} a_i \cdot s_i$, and for any vector $\mathbf{a}$ it holds that: $\|\mathbf{a}\|_1 \leq \sqrt{d}\cdot\|\mathbf{a}\|_2$. So, the width of $\mathcal{K}_\phi$ in the direction of $\mathbf{x}_t$ when $|S_\phi| = d$ is:

$$\begin{aligned}
w\left(\mathcal{K}_\phi, \mathbf{x}_t\right) &= \max_{\mathbf{p},\mathbf{q}\in\mathcal{K}_\phi} \langle \mathbf{x}_t, \mathbf{p}-\mathbf{q}\rangle && \text{(definition of width)}\\
&\leq \sum_{i\in[|S_\phi|]} a_i \cdot \max_{\mathbf{p},\mathbf{q}\in K_\phi} \langle s_i, \mathbf{p}-\mathbf{q}\rangle && \left(\mathbf{x}_t = \textstyle\sum_{i\in[|S_\phi|]} a_i\mathbf{s}_i \text{ and properties of } \max(\cdot)\right)\\
&\leq \sum_{i\in[|S_\phi|]} a_i \cdot \delta && \text{(definition of small dimensions)}\\
&\leq \|\mathbf{a}\|_1 \cdot \delta \leq \sqrt{d}\cdot\delta\cdot\|\mathbf{a}\|_2 && \left(\|\mathbf{a}\|_1 \leq \sqrt{d}\cdot\|\mathbf{a}\|_2\right)
\end{aligned}$$

Substituting $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$ the latter becomes: $w(\mathcal{K}_\phi, \mathbf{x}_t) \leq \frac{\varepsilon}{4(\sqrt{d}+1)} \leq \varepsilon$. Therefore, when $L_\phi = \emptyset$, then CorPV.Known incurs no additional regret for any context it receives in the future, if we are interested in the ε-ball loss. Using the fact that each epoch contains $\tau = 2d\cdot\bar{c}(d+1)+1$ rounds during which we can incur a loss of at most 1 we have that the regret for the ε-ball loss is equal to:

$$R_{\varepsilon-\text{ball}}(T) = O\left(d\log\frac{d}{\varepsilon}\right)\cdot(2d\cdot\bar{c}\cdot(d+1)+1) = \mathcal{O}\left((d^2\bar{c}+1)d\log\left(\frac{d}{\varepsilon}\right)\right)$$

For the absolute and the pricing loss, for every round after the set L_ϕ has become empty, the queried point incurs a loss of at most ε. As a result the regret for both cases is *at most*

$$R_{\varepsilon-\text{ball}}(T) + \varepsilon\cdot T$$

Tuning $\varepsilon = 1/T$ we get the result for these two loss functions. ■

4.3 Proof of the main result

In this subsection, we prove the regret guarantee for algorithm CorPV.AI for the case of adversarially irrational agents. Any auxiliary lemmas used can be found in Appendix B.4.

Proof of Theorem 3.1. We present the proof for the ε-ball loss. Tuning $\varepsilon = 1/T$ afterwards gives the stated result for the absolute and pricing loss.

We separate the layers into two categories: layers $j \geq \log C$ are *corruption-tolerant*, and layers $j < \log C$ are *corruption-intolerant*. Every layer j, if it were to run in isolation, would spend Φ_j epochs until converging to a knowledge set with width at most ε in all the directions. However, in CorPV.AI layer j's epoch potentially gets increased every time that a layer $j' \geq j$ changes epoch.

Since there are at most $\log T$ layers, this results in an added $\log T$ multiplicative overhead for the epochs of each layer. This overhead is suffered by the corruption-tolerant layers.

We first study the performance of the corruption-tolerant layers. Let $\beta_j > 0$ denote the failure probability for layer j such that $\beta_j \leq \frac{\beta}{\log T+1}$. From Lemma B.11 we have that with probability at least $1-\beta_j$, the actual corruption experienced by the tolerant layers is at most

$$\widetilde{C} = \ln\left(\frac{1}{\beta_j}\right) + 3 \leq \log\left(\frac{T}{\beta}\right). \tag{9}$$

From the regret guarantee of Proposition 4.10 for all rounds that this corruption-tolerant layer was sampled, the regret incurred by each of the tolerant layers j, denoted by $R_{\texttt{tol},j}$, is upper bounded by:

$$R_{\texttt{tol},j} \leq \mathcal{O}\left(\left(d^2\widetilde{C}+1\right) d \log\left(\frac{d}{\varepsilon}\right)\right) \tag{10}$$

Since there are at most $\log T$ tolerant layers, then with probability at least $1-\bar{\beta}$ (Lemma B.12), where $\bar{\beta} = \sum_{j\in[\log T]} \beta_j = \frac{\log T}{\log T+1}\beta$) the regret incurred by all the corruption-tolerant layers is:

$$R_{\text{tolerant}} \leq \sum_{j=1}^{\log T} R_{\texttt{tol},j} \tag{11}$$

We now move to the analysis of the corruption-intolerant layers. Let $j^\star$ denote the smallest corruption-tolerant layer, i.e., $j^\star = \min_j \{j \geq \log C\}$. Observe that each layer $j \leq j^\star$ is played until layer $j^\star$ identifies the target knowledge set having width at most ε in every direction. If $j^\star$ was run in isolation, from Equation (10) it would incur regret $R_{\texttt{tol},j^\star}$. When a context is not costly for $j^\star$, it is also not costly for layers $j < j^\star$. This follows because we have consistent knowledge sets and sets of small dimensions across the layers. As a result, whenever a context causes regret for a corruption-intolerant layer, with probability $1/C$, $j^\star$ is selected and it makes progress towards identifying the target. Using standard arguments for the binomial distribution (see Lemma B.13) we can show that for any scalar $\widetilde{\beta} > 0$ with probability at least $1-\widetilde{\beta}$, layer $j^\star$ is played *at least once* every $N = C\log(1/\widetilde{\beta})$ rounds. Set $\widetilde{\beta}$ to be $\widetilde{\beta} \leq \beta/(\log T + 1)$. Hence, the total regret from corruption-intolerant layers can be bounded by the total regret incurred by the first corruption-tolerant layer times N. Mathematically:

$$\begin{aligned} R_{\text{intolerant}} &\leq N \cdot R_{j^\star} \\ &= \mathcal{O}\left(N \cdot (2d(d+1)\log C + 1) d \log\left(\frac{d}{\varepsilon}\right)\right) \\ &= \mathcal{O}\left(C \cdot (2d(d+1)\log C + 1) d \log\left(\frac{d}{\varepsilon}\right) \log\left(\frac{1}{\widetilde{\beta}}\right)\right) \end{aligned} \tag{12}$$

until the appropriately small knowledge set is constructed for $j^\star$; subsequently this knowledge set dictates the behavior of the intolerant layers.

Putting everything together, and using the union bound again, we have that with probability at

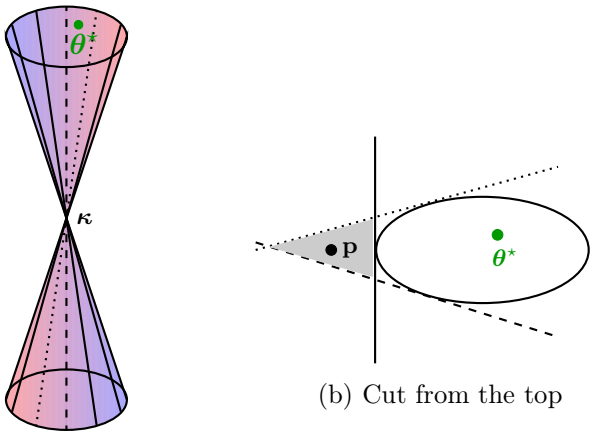
least $1 - \sum_{j\in[n]} \beta_j - \widetilde{\beta} = 1 - \beta$ the regret of CorPV.AI is:

$$\begin{aligned} R &= R_{\text{tolerant}} + R_{\text{intolerant}} && \text{(Equations (10) and (12))} \\ &\leq \mathcal{O}\left((\log T + C)\cdot\left(d^2\widetilde{C}+1\right) d\log\left(\frac{d}{\varepsilon}\right)\cdot\log\left(\frac{1}{\beta}\right)\right) \\ &\leq \mathcal{O}\left(d^3\cdot\log\left(\min\left\{T,\frac{d}{\varepsilon}\right\}\cdot\frac{1}{\beta}\right)\cdot\log\left(\frac{d}{\varepsilon}\right)\cdot\log\left(\frac{1}{\beta}\right)\cdot(\log(T)+C)\right) \end{aligned}$$

We finally discuss the computational complexity of CorPV.AI. Note that the complexity is dictated by the choice of $\widetilde{C} = \operatorname{poly}\log(T)$). As a result, from Lemma 4.11, substituting C with $\log T$, we get that CorPV.AI has expected runtime: $\tilde{\mathcal{O}}\left(\left(d^2\log T\right)^{\operatorname{poly}\log T}\cdot\operatorname{poly}\left(d\log\frac{d}{\varepsilon},\log T\right)\right)$. ■

4.4 Improper cuts in Corrupted Projected Volume.

Before we conclude our analysis section, we revisit the decision to make *improper* separating cuts. For the purposes of simplification, assume that contexts $\{\mathbf{x}_t\}_{t\in[\tau]}$ lead only to *explore* queries and that we are still interested in the simpler $\bar{c}$-known corruption setting with $\bar{c} = 1$. Ideally, if we could identify one context $\mathbf{x} \in \{\mathbf{x}_t\}_{t\in[\tau]}$ such that the $\bar{c}$-protected region $\mathcal{P}(\bar{c},\nu)$ is inside the halfspace $\mathbf{H}^+(\mathbf{x},\boldsymbol{\kappa}_\phi)$, i.e., $\mathcal{P}(\bar{c},\nu)\subseteq\mathbf{H}^+(\mathbf{x},\boldsymbol{\kappa}_\phi)$, then we could update the knowledge set as $\mathcal{K}_\phi\bigcap\mathbf{H}^+(\mathbf{x},\boldsymbol{\kappa}_\phi)$. As we have explained, these properties ensure that we make enough volume progress. In $d = 2$, there indeed exists one of the contexts among $\{\mathbf{x}_t\}_{t\in[\tau]}$ with the aforementioned property. This is due to a monotonicity argument that we describe in Appendix A.2.



(a) 3D view of the hyperplane "cuts" created by contexts

(b) Cut from the top

Figure 1: Reason why proper cuts do not suffice.

However, this is no longer true in $d = 3$, even if one sees arbitrarily many contexts in an epoch. To see this, take for example Figure 1 and assume that all rounds are *uncorrupted*. In Figure 1a each straight line corresponds to a context $\mathbf{x}_t$ and the shaded region corresponds to the halfspace with feedback $y_t = +1$, forming this "cone." In Figure 1b we take a *cut* from the knowledge set shown in Figure 1a and zoom in on only 3 of the contexts around $\boldsymbol{\theta}^\star$; the dotted, the dashed and the solid.

We are going to reason about the undesirability level of points like $\mathbf{p}$, all lying in the shaded area of Figure 1b. Both the dashed and the dotted hyperplane have $\mathbf{p}$ and $\boldsymbol{\theta}^\star$ as lying on the "same side", hence they do not contribute any undesirability points to $\mathbf{p}$. The solid line does contribute an undesirability point to $\mathbf{p}$ (and all the points in the shaded region).

Recall that since $\bar{c} = 1$, we need a hyperplane with undesirability at least 2 in the *entirety* of one of its halfspaces. But no matter how many contexts we have, if they have formed the sketch in Figure 1a, then there always exists a similar shaded region and the undesirability of the points it contains is 1.

That said, there exists an improper separating cut with undesirability at least $\bar{c}+1$. This is the cut that separates the upper part of the cone in Figure 1a (and which contains $\boldsymbol{\theta}^\star$) and the lower part of the cone.

5 Extension to bounded rationality

In this section, we extend the aforementioned algorithm and analysis to the bounded rationality behavioral model. We first recap the behavioral model. There is a noise parameter ξ_t drawn from a σ-subgaussian distribution $\texttt{subG}(\sigma)$, *fixed* across rounds and *known* to the learner, i.e., nature selects it before the first round and reveals it. At every round t a realized noise $\xi_t \sim \texttt{subG}(\sigma)$ is drawn, but ξ_t is never revealed to the learner. The agent's perceived value is then $\widetilde{v}_t = v(\mathbf{x}_t) + \xi_t$.

We focus on a *pseudo-regret* definition that compares to a benchmark that has access to $\boldsymbol{\theta}^\star$ and $\texttt{subG}(\sigma)$ but does not have access to the realization ξ_t. The resulting benchmark is:

$$L^\star_{\boldsymbol{\theta}^\star}(\mathbf{x}) = \min_{\omega^\star} \mathbb{E}_{\xi' \in \texttt{subG}(\sigma)} \left[\ell(\omega^\star, \langle \mathbf{x}, \boldsymbol{\theta}^\star \rangle, \langle \mathbf{x}, \boldsymbol{\theta}^\star \rangle + \xi') \right]. \tag{13}$$

and the corresponding regret is $R(T) = \sum_{t\in[T]} \left[\ell(\omega_t, v(\mathbf{x}_t), \widetilde{v}_t) - L^\star_{\boldsymbol{\theta}^\star}(\mathbf{x}_t) \right]$.

We remark that $\omega^\star$ should be thought of as the optimal query that the learner could have issued had we known $\boldsymbol{\theta}^\star$ but not the realization of ξ'. To develop more intuition regarding the benchmark stated assume for example that ξ' comes from a normal distribution. Then, the optimal $\omega^\star$ in expectation for the ε-ball and the absolute loss is equal to $\langle \mathbf{x}, \boldsymbol{\theta}^\star \rangle$. However, $\omega^\star$ should be strictly *lower* than $\langle \mathbf{x}, \boldsymbol{\theta}^\star \rangle$ when interested in the pricing loss, due to its discontinuity.

Our algorithm only differs from the one described in Section 3 in the ExploIT module (Algorithm 6) as ω_t is defined in a similar way with the benchmark. More formally, we again consider the worst-case selection of $\boldsymbol{\theta}$ consistent with the knowledge set and select the query that minimizes our loss with respect to that, i.e., $\omega_t = \min_\omega \max_{\boldsymbol{\theta}\in\mathcal{K}_\phi} \mathbb{E}_{\xi' \in \texttt{subG}(\sigma)} \left[\ell(\omega, \langle \mathbf{x}_t, \boldsymbol{\theta}^\star \rangle, \langle \mathbf{x}_t, \boldsymbol{\theta}^\star, \mathbf{x}_t \rangle + \xi') \right]$. The algorithm also doubles the corruption budget it should be robust to ($\bar{c}$ in Algorithm 4) and takes care of the additional noise by treating its tail as corruption and upper bounding it by $\bar{c}$.

Theorem 5.1. Let ε be as in Theorem 3.1. With probability at least $1-2\beta$, its guarantee extends to when rounds with fully rational agents are replaced by boundedly rational with $\sigma \leq \frac{\varepsilon}{8\sqrt{2d}(\sqrt{d}+1)\ln T}$.

Proof. We first show that under the low-noise regime stated above, the noise is bounded by $\Xi = \sqrt{2}\sigma \ln T$ with high probability at every round. Indeed, by Hoeffding's inequality we have that $\mathbb{P}\left[|\xi_t| > \Xi\right] \leq e^{-\ln^2 T}$. Using the union bound we have: $\mathbb{P}\left[|\xi_t| > \Xi, \text{ for any } t \in [T]\right] \leq \beta' = \beta/T$, and so $\mathbb{P}\left[|\xi_t| \leq \Xi, \forall t \in [T]\right] \geq 1-\beta$, which contributes the additional β in the high-probability argument.

We next show that when $\sigma \leq \frac{\varepsilon}{8\sqrt{2d}(\sqrt{d}+1)\ln T}$, then our algorithm maintains $\boldsymbol{\theta}^\star$ in $\mathcal{K}_\phi$. This is enough to ensure that the regret guarantee remains order unchanged. Since the perceived value of BR agents is $\widetilde{v}_t = v(\mathbf{x}_t) + \xi_t$, then, in order to "protect" $\boldsymbol{\theta}^\star$ (i.e., make sure that $\boldsymbol{\theta}^\star \in \mathcal{K}_{\phi+1}$) we need the hyperplanes that we feed to CorPV.SeparatingCut to have a margin of Ξ (since $\xi_t \leq \Xi$). To do so, it suffices to slightly change the lower bound of ν for the ν-margin projected undesirability levels that we use throughout the proof such that the new lower bound is $\underline{\nu}' = \underline{\nu} + \Xi = \sqrt{d}\cdot\delta + \Xi$. Since ν is such that $\underline{\nu}' \leq \nu \leq \bar{\nu}$, then it must the case that $\underline{\nu}' = \sqrt{d}\delta + \Xi \leq \bar{\nu} = \frac{\varepsilon(2\sqrt{d}+1)}{8\sqrt{d}(\sqrt{d}+1)}$. Solving for Ξ we obtain the result. This concludes our proof. ∎

Remark 1. *Corollary 1 in [CLPL19] has a regret of $\mathcal{O}(d^2 \log T)$ for pricing loss with $\sigma \approx \frac{d}{T \log T}$. For pricing loss, $\varepsilon = 1/T$ and our bound is weaker by a factor of d on the regret and a factor d^2 on the subgaussian variance σ, but it allows for the simultaneous presence of adversarially irrational agents.*

6 Gradient descent algorithm

In this section, we describe our analysis for the variant of gradient descent that we create for this problem. This algorithm is significantly simpler than the algorithms that are based on multidimensional binary search methods and has a better running time. On the other hand, it does not provide the logarithmic guarantees that we can obtain in previous sections when $C \approx 0$ and it does not extend to pricing loss.

ALGORITHM 7: ContextualSearch.GD

Initialize $\mathbf{z}_0 \in \mathcal{K}_0$ and $\gamma_0 = 1/2$.
for *rounds* $t \in [T]$ **do**
    For observed context $\mathbf{x}_t$, query $\omega_t = \langle \mathbf{x}_t, \mathbf{z}_t \rangle$.
    Receive feedback: $y_t = \operatorname{sgn}(\omega_t - \langle \boldsymbol{\theta}^\star, \mathbf{x}_t \rangle)$.
    Choose $\mathbf{z}_{t+1} = \Pi_{\mathcal{K}_0}\left(\mathbf{z}_t - \gamma_t \nabla f_t(\mathbf{z}_t)\right)$, where $\gamma_t = \sqrt{2/t}$ and $f_t(\mathbf{z}) = -y_t \cdot \langle \mathbf{z}, \mathbf{x}_t \rangle$.

Let us restrict our attention to the absolute loss, and recall that our goal is to minimize it using only binary feedback. At a high level, the algorithm tries to optimize a *proxy* function $f_t(\mathbf{z}) : \mathcal{K}_0 \to \mathbb{R}^d$, which is Lipschitz. Specifically, denoting the binary feedback received by $y_t = \operatorname{sgn}(\omega_t - \langle \mathbf{x}_t, \boldsymbol{\theta}^\star \rangle)$, then the proxy function to be optimized is $f_t(\mathbf{z}) = -y_t \cdot \langle \mathbf{x}_t, \mathbf{z} \rangle$. In other words, the query point at the next round $t+1$ is $\omega_{t+1} = \langle \mathbf{x}_{t+1}, \mathbf{z}_{t+1} \rangle$. This proxy function is convenient because on the one hand it is Lipschitz and on the other, its regret is an *upper bound* on the regret incurred by any algorithm optimizing the absolute loss for the same problem. Additionally, when faced with adversarially irrational agents the *same* algorithm suffers regret $\mathcal{O}(\sqrt{T} + C)$; this is due to the fact that adversarially irrational agents merely add an extra set of C erroneous rounds, from which the algorithm can certainly "recover" since there is no notion of a shrinking knowledge set.

Proposition 6.1. For an unknown corruption level C, ContextualSearch.GD incurs, in expectation, regret $\mathcal{O}(\sqrt{T} + C)$ for the absolute loss and $\mathcal{O}(\sqrt{T}/\varepsilon + C/\varepsilon)$ for the ε-ball loss.

Proof. Function $f_t(\mathbf{z}) = -y_t \cdot \langle \mathbf{z}, \mathbf{x}_t \rangle$ is Lipschitz in $\mathbf{z}$. So, using the known guarantees for ContextualSearch.GD and denoting by $\mathbf{z}^\star = \arg\min_{\mathbf{z}} \sum_{t \in [T]} f_t(\mathbf{z})$ we know that:

$$\sum_{t \in [T]} f_t(\mathbf{z}_t) - \sum_{t \in [T]} f_t(\mathbf{z}^\star) = \mathcal{O}\left(\sqrt{T}\right) \tag{14}$$

Due to the definition of $\mathbf{z}^\star$ we can relax the left-hand side of Equation (14) and get:

$$\sum_{t \in [T]} f_t(\mathbf{z}_t) - \sum_{t \in [T]} f_t(\boldsymbol{\theta}^\star) \leq \mathcal{O}\left(\sqrt{T}\right) \tag{15}$$

We now analyze the quantity on the left-hand side of Equation (15) as follows:

$$\sum_{t \in [T]} f_t(\mathbf{z}_t) - \sum_{t \in [T]} f_t(\boldsymbol{\theta}^\star) = \sum_{t \in [T]} y_t \cdot \left(\langle \boldsymbol{\theta}^\star, \mathbf{x}_t \rangle - \langle \mathbf{z}_t, \mathbf{x}_t \rangle\right) = \sum_{t \in [T]} \left|\langle \boldsymbol{\theta}^\star, \mathbf{x}_t \rangle - \omega_t\right| \tag{16}$$

which is the quantity that we wish to minimize when we are trying to minimize the absolute loss given binary feedback when the round is not corrupted. Given the fact that y_t is arbitrary for at most C rounds, we get that the regret incurred by ContextualSearch.GD is at most $\mathcal{O}(\sqrt{T}+C)$.

For the proof of the ε-ball loss, we show how the latter compares with the absolute loss. Indeed:

$$\sum_{t\in[T]} |\langle \boldsymbol{\theta}^\star, \mathbf{x}_t\rangle - \omega_t| \geq \varepsilon \sum_{t\in[T]} \mathbb{1}\left\{|\langle \boldsymbol{\theta}^\star, \mathbf{x}_t\rangle - \omega_t| \geq \varepsilon\right\}$$

Combining the above with Equation (16) we get that ContextualSearch.GD for the ε-ball loss incurs regret $\mathcal{O}(\sqrt{T}/\varepsilon + C/\varepsilon)$ ■

7 Conclusion

In this paper, we initiate the study of contextual search under adversarial noise models, motivated by pricing settings where some agents may act *irrationally*, i.e., in ways that are inconsistent with respect to the underlying ground truth. Although classical algorithms may be prone to even a few such agents, we show two algorithms that achieve regret guarantees that attain the logarithmic (uncorrupted) guarantees, while degrading gracefully with the number C of irrational agents.

Our work opens up two fruitful avenues for future research. First, the regret in both of our algorithms is sublinear when $C = o(T)$ but becomes linear when $C = \Theta(T)$. Designing algorithms that can provide sublinear regret against the ex-post best linear model, in the latter regime, is an exciting direction of future research and our model offers a concrete formulation of this problem. Second, our algorithm that attains the logarithmic guarantee has a regret of the order of $Cd^3 \operatorname{poly}\log T$ and improving the dependence on d is an interesting open direction. After a sequence of papers, this dependence is now optimized when all agents are fully rational [CLPL19, LPLV18, PLS18, LPLS21].

A Contextual search with fully rational agents for ε-ball loss [LPLV18]

A.1 ProjectedVolume algorithm and intuition

In this subsection, we describe the ProjectedVolume algorithm of [LPLV18], which is the algorithm that CorPV.Known builds on. ProjectedVolume minimizes the ε-ball loss for fully rational agents by approximately estimating $\boldsymbol{\theta}^\star$. At all rounds $t \in [T]$ ProjectedVolume maintains a convex body, called the *knowledge set* and denoted by $\mathcal{K}_t \in \mathbb{R}^d$, which corresponds to all values $\boldsymbol{\theta}$ that are not ruled out based on the information until round t. It also maintains a set of orthonormal vectors $S_t = \{\mathbf{s}_1, \dots, \mathbf{s}_{|S_t|}\}$ such that $\mathcal{K}_t$ has small width along these directions, i.e., $\forall \mathbf{s} \in S_t : w(\mathcal{K}_t, \mathbf{s}) \leq \delta'$. The algorithm "ignores" a dimension of $\mathcal{K}_t$, once it becomes small, and focuses on the projection of $\mathcal{K}_t$ onto a set L_t of dimensions that are orthogonal to S_t and have larger width, i.e., $\forall \mathbf{l} \in L_t : w(\mathcal{K}_t, \mathbf{l}) \geq \delta'$.

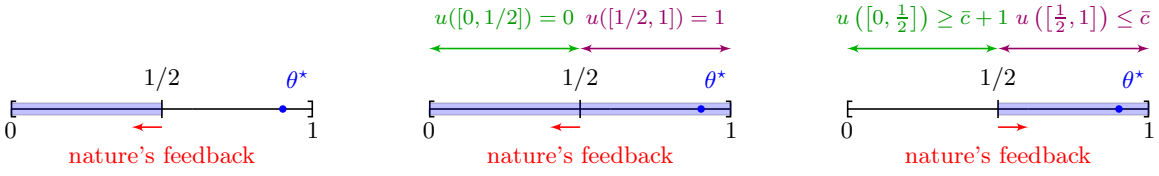
ALGORITHM 8: ProjectedVolume [LPLV18]

Initialize $S_0 \leftarrow \emptyset, \mathcal{K}_0 \leftarrow \mathcal{K}$.
for $t \in [T]$ **do**
  context $\mathbf{x}_t$, chosen by nature.
  Query point $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\kappa}_t \rangle$, where $\boldsymbol{\kappa}_t \leftarrow$ approx-centroid($\texttt{Cyl}(\mathcal{K}_t, S_t)$).
  Observe feedback y_t and set $\mathcal{K}_{t+1} \leftarrow \mathcal{K}_t \bigcap \mathbf{H}^+(\mathbf{x}_t, \omega_t)$ if $y_t = +1$ or $\mathcal{K}_{t+1} \leftarrow \mathcal{K}_t \bigcap \mathbf{H}^-(\mathbf{x}_t, \omega_t)$ if $y_t = -1$.
  Add all directions $\mathbf{u}$ orthogonal to S_t with $w(\mathcal{K}_{t+1}, \mathbf{u}) \leq \delta' = \frac{\varepsilon^2}{16d(d+1)^2}$ to S_t.
  Set $S_{t+1} = S_t$.

At round t, after observing $\mathbf{x}_t$, the algorithm queries point $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\kappa}_t \rangle$ where $\boldsymbol{\kappa}_t$ is the *approximate* centroid of knowledge set $\mathcal{K}_t$. Based on the feedback, y_t, the algorithm eliminates one of $\mathbf{H}^+(\mathbf{x}_t, \omega_t)$ or $\mathbf{H}^-(\mathbf{x}_t, \omega_t)$. The analysis uses the volume of $\Pi_{L_t}\mathcal{K}_t$, denoted by $\texttt{vol}(\Pi_{L_t}\mathcal{K}_t)$, as a potential function. After each query either the set of small dimensions S_t increases, thus making $\texttt{vol}(\Pi_{L_t}\mathcal{K}_t)$ increase by a bounded amount (which can happen at most d times), or $\texttt{vol}(\Pi_{L_t}\mathcal{K}_t)$ decreases by a factor of $(1 - 1/e^2)$. This potential function argument leads to a regret of at most $\mathcal{O}(d \log(d/\varepsilon))$.

A.2 Failure of ProjectedVolume against corruptions when $d = 1$

When $d = 1$ and $\bar{c} = 0$, there exists a $\theta^\star \in \mathbb{R}$ and nature replies whether ω_t is *greater* or *smaller* than $\theta^\star$. By appropriate queries, the learner can decrease the size of the knowledge set that is consistent with all past queries so that, after $\log(1/\varepsilon)$ rounds, she identifies an ε-ball containing $\theta^\star$. However, even when $\bar{c} = 1$, the above algorithm can be easily misled. Think about an example as in Figure 2a where $\theta^\star = 3/4$, the learner queries point $1/2$, and nature corrupts the feedback making her retain the interval $[0, 1/2]$, instead of $[1/2, 1]$, as her current knowledge set.



(a) Binary search at round 1. (b) Undesirabilities at round 1. (c) Undesirabilities at round $2\bar{c} + 1$.

Figure 2: Single dimensional binary search. The opaque band is the knowledge set after each query.

That said, if the learner knows $\bar{c}$ then, by repeatedly querying the same point, she can guarantee that if she observes $y_t = +1$ (resp. $y_t = -1$) for *at least* $\bar{c} + 1$ times, then feedback $y_t = +1$ (resp. $y_t = -1$) is definitely consistent with $\theta^\star$. Hence, by repeating each query $2\bar{c} + 1$ times the learner

can incur regret at most $(2\bar{c}+1)\log(1/\varepsilon)$. Unfortunately, in higher dimensions it is impossible for the learner to repeat the exact same query, as nature chooses different contexts.

B Supplementary material for Section 4

B.1 Auxiliary lemmas for landmarks (Analysis of Section 4.1)

Lemma 4.7. Let basis $E_\phi = \{e_1, \ldots, e_{d-|S_\phi|}\}$ orthogonal to S_ϕ. For all $\{(\mathbf{x}_t, \omega_t)\}_{t\in[\tau]}$ such that $w(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \geq \varepsilon$, there exists i such that: $|\langle e_i, \mathbf{x}_t\rangle| \geq \bar{\nu}$, where $\bar{\nu} = \frac{\varepsilon - 2\sqrt{d}\cdot\delta}{4\sqrt{d}}$.

Proof. We first show that $\|\Pi_{L_\phi}\mathbf{x}_t\| \geq \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4}$. Since for the contexts $\{\mathbf{x}_t\}_{t\in[\tau]}$ that we consider in epoch ϕ it holds that: $w(\texttt{Cyl}(\mathcal{K}_\phi, S_\phi), \mathbf{x}_t) \geq \varepsilon$, then there exists a point $\mathbf{p} \in \texttt{Cyl}(\mathcal{K}_\phi, S_\phi)$ such that $|\langle \mathbf{x}_t, \mathbf{p} - \boldsymbol{\kappa}_\phi\rangle| \geq \frac{\varepsilon}{2}$. Applying the triangle inequality:

$$\left|\left\langle \Pi_{L_\phi}\mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\right\rangle\right| + \left|\left\langle \Pi_{S_\phi}\mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\right\rangle\right| \geq |\langle \mathbf{x}_t, \mathbf{p}-\boldsymbol{\kappa}_\phi\rangle| \geq \frac{\varepsilon}{2} \tag{17}$$

Along the directions in S_ϕ the following is true:

$$|\langle \Pi_{S_\phi}\mathbf{x}_t, \Pi_{S_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\rangle| \leq \|\Pi_{S_\phi}\mathbf{x}_t\|_2 \cdot \|\Pi_{S_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\|_2 \leq \|\mathbf{x}_t\|_2 \cdot \sqrt{d} \cdot \|\Pi_{S_\phi}(\mathbf{p}-\kappa_\phi)\|_\infty \leq 1\cdot\delta\sqrt{d}$$

Using the latter, Equation (17) now becomes:

$$\left|\left\langle \Pi_{L_\phi}\mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\right\rangle\right| \geq \frac{\varepsilon}{2} - \sqrt{d}\cdot\delta \tag{18}$$

We next focus on upper bounding term $\left|\left\langle \Pi_{L_\phi}\mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\right\rangle\right|$. By applying the Cauchy-Schwarz inequality, Equation (18) becomes:

$$\left\|\Pi_{L_\phi}\mathbf{x}_t\right\|_2 \left\|\Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\right\|_2 \geq \left|\left\langle \Pi_{L_\phi}\mathbf{x}_t, \Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\right\rangle\right| \geq \frac{\varepsilon}{2} - \sqrt{d}\cdot\delta \tag{19}$$

For $\|\Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\|_2$, observe that $\mathbf{p}$ and $\boldsymbol{\kappa}_\phi$ are inside $\texttt{Cyl}(\mathcal{K}_\phi, S_\phi)$, and $\mathcal{K}_\phi$ has *radius* at most 1. By the fact that $\mathbf{p} \in \texttt{Cyl}(\mathcal{K}_\phi, S_\phi)$ and Definition 3.2, we can write it as $\mathbf{p} = \mathbf{z} + \sum_{i=1}^{|S_\phi|} y_i\mathbf{s}_i$ where $\mathbf{s}_i$ form a basis for S_ϕ (which, recall, is orthogonal to L_ϕ) and $\mathbf{z} \in \Pi_{L_\phi}\mathcal{K}_\phi$. Since $\mathcal{K}_\phi$ is contained in the unit ℓ_2 ball, we also have that $\Pi_{L_\phi}\mathcal{K}_\phi$ is contained in the unit ℓ_2 ball. Hence $\|\Pi_{L_\phi}\mathbf{p}\|_2 = \|\mathbf{z}\|_2 \leq 1$. The same holds for $\boldsymbol{\kappa}_\phi$, and so by the triangle inequality, we have $\|\Pi_{L_\phi}(\mathbf{p}-\boldsymbol{\kappa}_\phi)\|_2 \leq 2$. Hence, from Equation (19) we get that: $\|\Pi_{L_\phi}\mathbf{x}_t\| \geq \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4}$.

Assume now for contradiction that there does not exist i such that $|\langle e_i, \mathbf{x}_t\rangle| \geq \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4\sqrt{d}}$. This means that for all $j \in [d-|S_\phi|]$ and all contexts $\{\mathbf{x}_t\}_{t\in[\tau]}$: $\langle e_i, \mathbf{x}_t\rangle < \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4\sqrt{d}}$. Denoting by $(E\mathbf{x}_t)_j$ the j-th coordinate of $E\mathbf{x}_t$ we have that $(E\mathbf{x}_t)_j = \langle e_j, \mathbf{x}_t\rangle$. Hence, if $|\langle \mathbf{x}_t, e_j\rangle| < \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4\sqrt{d}}$ then:

$$\|E\mathbf{x}_t\|_2 = \|\Pi_{L_\phi}\mathbf{x}_t\|_2 \leq \sqrt{\sum_{i=1}^{d}(\langle \mathbf{x}_t, e_i\rangle)^2} < \sqrt{d\left(\frac{\varepsilon-2\sqrt{d}\cdot\delta}{4\sqrt{d}}\right)^2} < \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4}$$

which contradicts the fact that $\|\Pi_{L_\phi}\mathbf{x}_t\| \geq \frac{\varepsilon-2\sqrt{d}\cdot\delta}{4}$ established above. ∎

B.2 Auxiliary lemmas for CorPV.SeparatingCut (Analysis of Section 4.2)

Lemma B.1 (Cap Volume)**.** With probability at least $\frac{1}{20\sqrt{d-1}}$, a point randomly sampled from a ball of radius ζ around $\mathbf{p}^\star$, $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$, lies on the following halfspace: $\mathbf{H}^+\left(\mathbf{h}^\star_\phi, \left\langle \mathbf{h}^\star_\phi, \mathbf{p}^\star \right\rangle + \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}}\right)$.

Proof. We want to compute the probability that a point randomly sampled from $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$ falls in the following halfspace:

$$\mathbf{H}^+ \equiv \left\{\mathbf{x} : \langle \mathbf{h}^\star, \mathbf{x} - \mathbf{p}^\star \rangle \geq \frac{\zeta \cdot \ln(3/2)}{\sqrt{d-1}}\right\}$$

Hence, we want to bound the following probability: $\mathbb{P}\left[\mathbf{x} \in \mathbf{H}^+ | \mathbf{x} \in \mathcal{B}(\mathbf{p}^\star, \zeta)\right]$. If we normalize $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$ to be the unit ball B, then this probability is equal to:

$$\mathbb{P}\left[\mathbf{x} \in \mathbf{H}^+ | \mathbf{x} \in \mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)\right] = \mathbb{P}\left[\mathbf{x} \in \mathbf{H}^1 | \mathbf{x} \in B\right] = \frac{\texttt{vol}\left(B \bigcap \mathbf{H}^1\right)}{\texttt{vol}(B)} \tag{20}$$

where $\mathbf{H}^1$ is the halfspace such that $\mathbf{H}^1 \equiv \left\{\mathbf{x} : \langle \mathbf{h}^\star, \mathbf{x} \rangle \geq \frac{\ln(3/2)}{\sqrt{d-1}} = r\right\}$, and the last equality is due to the fact that we are sampling uniformly at random.

Similar to the steps in [BHK16, Section 2.4.2], in order to compute $\texttt{vol}\left(B \bigcap \mathbf{H}^1\right)$ we integrate the incremental volume of a disk with width dx_1, with its face being a $(d-1)$-dimensional ball of radius $\sqrt{1-x_1^2}$. Let $V(d-1)$ denote the volume of the $(d-1)$-dimensional unit ball. Then, the surface area of the aforementioned disk is: $(1-x_1^2)^{\frac{d-1}{2}} \cdot V(d-1)$.

$$\begin{aligned}
\texttt{vol}\left(B \bigcap \mathbf{H}^1\right) &= \int_r^1 (1-x_1^2)^{\frac{d-1}{2}} \cdot V(d-1) dx_1 = V(d-1) \cdot \int_r^1 (1-x_1^2)^{\frac{d-1}{2}} dx_1 \\
&\qquad\qquad (V(d-1) \text{ is a constant}) \\
&\geq V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} (1-x_1^2)^{\frac{d-1}{2}} dx_1 \qquad\qquad (\sqrt{\tfrac{\ln 2}{d-1}} < 1, \forall d \geq 2) \\
&\geq V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} \left(e^{-2x_1^2}\right)^{\frac{d-1}{2}} dx_1 \\
&\qquad\qquad (1 - x^2 \geq e^{-2x^2}, x \in [0, 0.8], \tfrac{\ln 2}{d-1} \leq 0.8, \forall d \geq 2) \\
&= V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} e^{-x_1^2(d-1)} dx_1 \\
&\geq V(d-1) \cdot \int_r^{\sqrt{\frac{\ln 2}{d-1}}} \sqrt{\frac{d-1}{\ln 2}} \cdot x_1 \cdot e^{-x_1^2(d-1)} dx_1 \qquad\qquad (x_1 \leq \sqrt{\tfrac{\ln 2}{d-1}}) \\
&\geq -\frac{V(d-1)}{2\sqrt{(d-1)\cdot \ln 2}} \left[e^{-(d-1)x^2}\right]_r^{\sqrt{\frac{\ln 2}{d-1}}} \\
&= \frac{V(d-1)}{2\sqrt{(d-1)\cdot \ln 2}} \left(e^{-\ln(3/2)} - e^{-\ln 2}\right) = \frac{V(d-1)}{2\sqrt{(d-1)\cdot \ln 2}} \left(\frac{2}{3} - \frac{1}{2}\right) \\
&= \frac{V(d-1)}{12\sqrt{(d-1)\cdot \ln 2}}
\end{aligned} \tag{21}$$

Next we show how to upper bound the volume of the unit ball B. First we compute the volume of one of the ball's hemispheres, denoted be $\texttt{vol}(H)$. Then, the volume of the ball is $\texttt{vol}(B) = 2\texttt{vol}(H)$.

The volume of a hemisphere is *at most* the volume of a cylinder of height 1 and radius 1, i.e., $V(d-1)\cdot 1$. Hence, $\texttt{vol}(B) \le 2V(d-1)$. Combining this with Equation (21), Equation (20) gives the following ratio:

$$\frac{\texttt{vol}\left(B \bigcap \mathbf{H}^1\right)}{\texttt{vol}(B)} \ge \frac{1}{24\sqrt{(d-1)\cdot \ln 2}} \ge \frac{1}{20\sqrt{d-1}}.$$

This concludes our proof. ■

This lower bound on the probability that a randomly sampled point has the large enough margin that Perceptron requires for efficient convergence, suffices for us to guarantee that after a polynomial number of rounds, such a $\widetilde{\mathbf{q}}$ has been identified in expectation.

Lemma B.2. In expectation, after $N = 20\sqrt{d-1}$ samples from $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$, at least one of the samples lies in halfspace $\mathbf{H}^+\left(\mathbf{h}_\phi^\star, \left\langle \mathbf{h}_\phi^\star, \mathbf{p}^\star \right\rangle + \frac{\zeta\cdot\ln(3/2)}{\sqrt{d-1}}\right)$.

Proof. From Lemma B.1, the probability that a point randomly sampled from $\mathcal{B}_{L_\phi}(\mathbf{p}^\star, \zeta)$ lies on halfspace $\mathbf{H}^+\left(\mathbf{h}_\phi^\star, \left\langle \mathbf{h}_\phi^\star, \mathbf{p}^\star \right\rangle + \frac{\zeta\cdot\ln(3/2)}{\sqrt{d-1}}\right)$ is at least $\frac{1}{20\sqrt{d-1}}$. Hence, in expectation after $20\sqrt{d-1}$ samples we have identified one such point by union bound. ■

B.3 Auxiliary lemmas for volumetric progress (Analysis of Section 4.2)

The next lemma states that a convex body $\mathcal{K}$ with width at least δ in every direction must fit a ball of diameter δ/d inside it.

Lemma B.3 ([LPLV18, Lemma 6.3])**.** If $\mathcal{K} \subset \mathbb{R}^d$ is a convex body such that $w(\mathcal{K}, \mathbf{u}) \ge \delta$ for every unit vector $\mathbf{u}$, then $\mathcal{K}$ contains a ball of diameter δ/d.

Lemma B.4 (Directional Grünbaum [LPLV18, Theorem 5.3])**.** If $\mathcal{K}$ is a convex body and $\boldsymbol{\kappa}$ is its centroid, then, for *every* unit vector $\mathbf{u} \neq 0$, the set $\mathcal{K}_+ = \mathcal{K} \bigcap \{\mathbf{x} | \langle \mathbf{u}, \mathbf{x} - \boldsymbol{\kappa}\rangle \ge 0\}$ satisfies:

$$\frac{1}{d+1} w(\mathcal{K}, \mathbf{v}) \le w(\mathcal{K}_+, \mathbf{v}) \le w(\mathcal{K}, \mathbf{v}), \quad \text{for all unit vectors } \mathbf{v}.$$

The Approximate Grünbaum lemma, which is stated next, relates the volume of a set $\mathcal{K}_+^\mu = \{\mathbf{x} \in \mathcal{K} : \langle \mathbf{u}, \mathbf{x} - \boldsymbol{\kappa}\rangle \ge \mu\}$ with the volume of set $\mathcal{K}$, when $\mu \le 1/d$ for any unit vector $\mathbf{u}$. Its proof (provided in Appendix B.3 for completeness) is similar to the proof of [LPLV18, Lemma 5.5] with the important difference that μ is no longer $w(\mathcal{K}, \mathbf{u})/(d+1)^2$, but rather, $\mu < 1/d$

Lemma B.5 (Approximate Grünbaum)**.** Let $\mathcal{K}$ be a convex body and $\boldsymbol{\kappa}$ be its centroid. For an arbitrary unit vector $\mathbf{u}$ and a scalar μ such that $0 < \mu < \frac{1}{d}$, let $\mathcal{K}_+^\mu = \{\mathbf{x} \in \mathcal{K} : \langle \mathbf{u}, \mathbf{x} - \boldsymbol{\kappa}\rangle \ge \mu\}$. Then: $\texttt{vol}\left(\mathcal{K}_+^\mu\right) \ge \frac{1}{2e^2}\texttt{vol}(\mathcal{K})$.

In order to prove the Appoximate Grünbaum lemma we make use of Brunn's theorem and the Grünbaum Theorem, both stated below.

Lemma B.6 (Brunn's Theorem)**.** For convex set $\mathcal{K}$ if $g(x)$ is the $(d-1)$-dimensional volume of the section $\mathcal{K} \bigcap \{\mathbf{y} | \langle \mathbf{y}, e_i\rangle = x\}$, then the function $r(x) = g(x)^{\frac{1}{d-1}}$ is concave in x over its support.

Lemma B.7 (Grünbaum Theorem)**.** Let $\mathcal{K}$ denote a convex body and $\boldsymbol{\kappa}$ its centroid. Given an arbitrary non-zero vector $\mathbf{u}$, let $\mathcal{K}_+ = \{\mathbf{x} | \langle \mathbf{u}, \mathbf{x} - \boldsymbol{\kappa} \rangle \geq 0\}$. Then:

$$\frac{1}{e}\texttt{vol}(\mathcal{K}) \leq \texttt{vol}\left(\mathcal{K}_+\right) \leq \left(1 - \frac{1}{e}\right)\texttt{vol}(\mathcal{K})$$

Proof of Lemma B.5. For this proof we assume without loss of generality that $\mathbf{u} = e_1$, and that the projection of $\mathcal{K}$ onto e_1 is interval $[a, 1]$. We are interested in comparing the following two quantities: $\texttt{vol}(\mathcal{K})$ and $\texttt{vol}(\mathcal{K}_+^\mu)$. By definition:

$$\texttt{vol}\left(\mathcal{K}_+\right) = \int_0^1 r(x)^{d-1}dx \qquad \text{and} \qquad \texttt{vol}\left(\mathcal{K}_+^\mu\right) = \int_\mu^1 r(x)^{d-1}dx \tag{22}$$

where $r(x) = g(x)^{\frac{1}{d-1}}$ and $g(x)$ corresponds to the volume of the $(d-1)$-dimensional section $\mathcal{K}_x = \mathcal{K} \bigcap \{\mathbf{x} | \langle \mathbf{x}, e_i \rangle = x\}$. We now prove that $\texttt{vol}(\mathcal{K}_+^\mu) \geq \frac{1}{e}\texttt{vol}(\mathcal{K}_+)$. Combining this with Grünbaum Theorem (Lemma B.7) gives the result. We denote by ρ the following ratio:

$$\rho = \frac{\int_\mu^1 r(x)^{d-1}dx}{\int_0^1 r(x)^{d-1}dx} \geq \frac{\int_{1/d}^1 r(x)^{d-1}dx}{\int_0^1 r(x)^{d-1}dx} \tag{23}$$

We approximate function $r(x)$ with function $\tilde{r}$:

$$\tilde{r}(x) = \begin{cases} r(x) & \text{if } 0 \leq x \leq \delta \\ (1-x) \cdot \frac{r(\delta)}{1-\delta} & \text{if } \delta < x \leq 1 \end{cases}$$

Note that since $0 = \tilde{r}(1) \leq r(1)$ (because $r(x)$ is a non-negative function) and $r(x)$ is concave from Brunn's theorem (Lemma B.6), for functions $r(x)$ and $\tilde{r}(x)$ it holds that $r(x) \geq \tilde{r}(x)$. Using this approximation function $\tilde{r}(x)$ along with the fact that function $f(z) = \frac{z}{y+z}$ is *increasing* for any scalar $y > 0$, we can relax Equation (23) as follows:

$$\rho \geq \frac{\int_{1/d}^1 \tilde{r}(x)^{d-1}dx}{\int_0^{1/d} \tilde{r}(x)^{d-1}dx + \int_{1/d}^1 \tilde{r}(x)^{d-1}dx} \tag{24}$$

Next, we use another approximation function $\hat{r}(x) = (1-x) \cdot \frac{r(\delta)}{1-\delta}, 0 \leq x \leq 1$; this time in order to approximate function $\tilde{r}(x)$. For $x \in [\delta, 1]$: $\tilde{r}(x) = \hat{r}(x)$. For $x \in [0, \delta]$ and since $\tilde{r}(0) = r(0) = 0$ and $\tilde{r}(x)$ is concave in $x \in [0, \delta]$, $\hat{r}(x) \geq \tilde{r}(x) = r(x), x \in [0, \delta]$. Hence, Equation (24) can be relaxed to:

$$\rho \geq \frac{\int_{1/d}^1 \hat{r}(x)^{d-1}dx}{\int_0^{1/d} \hat{r}(x)^{d-1}dx + \int_{1/d}^1 \hat{r}(x)^{d-1}dx} = \frac{\int_{1/d}^1 (1-x)^{d-1} \cdot \left(\frac{r(\delta)}{1-\delta}\right)^{d-1} dx}{\int_0^1 (1-x)^{d-1} \cdot \left(\frac{r(\delta)}{1-\delta}\right)^{d-1} dx}$$

$$= \frac{\int_{1/d}^1 (1-x)^{d-1}dx}{\int_0^1 (1-x)^{d-1}dx} = \frac{-\frac{1}{d}\left(0 - \left(1 - \frac{1}{d}\right)^d\right)}{-\frac{1}{d}\left(0 - 1\right)} = \left(1 - \frac{1}{d}\right)^d \geq \frac{1}{2e}$$

This concludes our proof. ■

We next state the cylindrification lemma, whose proof was provided by [LPLV18], relates the volume of the convex body to the volume of its projection onto a subspace.

Lemma B.8 (Cylindrification [LPLV18, Lemma 6.1])**.** Let $\mathcal{K}$ be a convex body in $\mathbb{R}^d$ such that $w(\mathcal{K}, \mathbf{u}) \geq \delta'$ for every unit vector $\mathbf{u}$. Then, for every $(d-1)$-dimensional subspace L it holds that $\texttt{vol}(\Pi_L \mathcal{K}) \leq \frac{d(d+1)}{\delta'} \texttt{vol}(\mathcal{K})$.

Lemma B.9 (Epoch Based Projected Grünbaum)**.** For $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$ and $\mathcal{K}_{\phi+1} = \mathcal{K}_\phi \bigcap \mathbf{H}^+(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$, where $\mathbf{H}^+(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ was the halfspace returned from CorPV.SeparatingCut it holds that:

$$\texttt{vol}\left(\Pi_{L_\phi} \mathcal{K}_{\phi+1}\right) \leq \left(1 - \frac{1}{2e^2}\right) \texttt{vol}(\Pi_{L_\phi} \mathcal{K}_\phi)$$

Proof. By Lemma 4.11, we know that CorPV.SeparatingCut returned hyperplane $(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)$ orthogonal to all small dimensions, such that $\texttt{dist}(\boldsymbol{\kappa}^\star_\phi, (\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi)) \leq 3\bar{\nu} = 3 \cdot \frac{\varepsilon - 2\sqrt{d}\delta}{4\sqrt{d}}$. Substituting $\delta = \frac{\varepsilon}{4(d+\sqrt{d})}$ we get that:

$$\texttt{dist}\left(\boldsymbol{\kappa}^\star_\phi, \left(\widetilde{\mathbf{h}}_\phi, \widetilde{\omega}_\phi\right)\right) \leq \frac{(2\sqrt{d}+1)\varepsilon}{2\sqrt{d}(\sqrt{d}+1)} \leq \frac{1}{d}$$

where the last inequality uses the fact that $\varepsilon \leq 1/\sqrt{d}$ and that $\frac{2\sqrt{d}+1}{\sqrt{d}+1} \leq 2$. Hence, the clause in the approximate Grünbaum lemma (Lemma B.5) holds and as a result, applying the approximate Grünbaum lemma with $\mathcal{K} = \Pi_{L_\phi} \mathcal{K}_\phi$, the lemma follows. ■

For completeness, we also list here the Perceptron mistake bound lemma, which is well-known ([Nov63]).

Lemma B.10. Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i \in [n]}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, if $\|x_i\| \leq R$ and there exists a linear classifier $\boldsymbol{\theta}$ such that $\|\boldsymbol{\theta}\| = 1$ and $y_i \cdot \langle \boldsymbol{\theta}, \mathbf{x}_i \rangle \geq \gamma$ for a scalar γ. Then, the number of mistakes that the Perceptron algorithm incurs in $\mathcal{D}$ is upper bounded by $(R/\gamma)^2$.

B.4 Auxiliary lemmas for analysis of CorPV.AI (Analysis for Section 4.3)

Lemma B.11 ([LMPL18, Lemma 3.3])**.** For corruption level C, each layer $j \geq \log C$ observes at most $\ln(1/\beta) + 3$ corruptions with probability at least $1 - \beta$.

Lemma B.12. Let $X_1, \ldots, X_n$ denote n random binary variables that take value of 0 with probability *at most* $p_1, \ldots, p_n$ respectively. Then, the following is true:

$$\mathbb{P}\left[\bigcap_{i \in [n]} X_i\right] \geq 1 - \sum_{i \in [n]} p_i$$

Proof. This inequality is proven using the union bound as follows:

$$\mathbb{P}\left[\bigcap_{i \in [n]} X_i\right] = 1 - \mathbb{P}[\exists j : X_j = 0] \geq 1 - \sum_{j \in [n]} p_j$$

■

Lemma B.13. Let X a random variable following the binomial distribution with parameters n and p, such that $p = 1/a$ for some $a > 0$. Then, $\mathbb{P}[X < 1] \leq \delta$ for $n = a \cdot \log(1/\delta)$.

Lemma B.14. Using the definition of the binomial distribution we have that: $\mathbb{P}[X < 1] = \mathbb{P}[X = 0] = (1-p)^n$. For any β in order for the result to hold one needs

$$n \geq \frac{\log(1/\beta)}{\log\left(\frac{1}{1-p}\right)} \tag{25}$$

Since $\log\left(\frac{1}{1-p}\right) \geq \frac{p}{1-p}$ then Equation (25) is satisfied. Substituting $p = 1/a$ we get the result.

B.5 Discussion on the need for improper cuts (Section 4.4)

In order to prove the results of this section, we use a simplified version of undesirability levels; we define a point's $\mathbf{p} \in \mathcal{K}_\phi$ undesirability level as the number of rounds within epoch ϕ, for which

$$u_\phi(\mathbf{p}) = \sum_{t \in [\tau]} \mathbb{1}\left\{\langle \mathbf{p} - \boldsymbol{\kappa}_\phi, \mathbf{x}_t \rangle \cdot y_t < 0\right\}.$$

We next present two propositions regarding the number of contexts needed in order to guarantee that we have found an appropriately undesirable hyperplane, for the cases of $d = 2$ and $d = 3$ respectively.

Proposition B.15. For $d = 2$ and any corruption level $\bar{c}$, after $3\bar{c} + 1$ rounds within an epoch, there exists a hyperplane $(\mathbf{x}', \omega')$ among $\{(\mathbf{x}_t, \omega_t)\}_{t \in [\tau]}$ with undesirability level at least $\bar{c} + 1$ in the entirety of one of its halfspaces.

Proof. Since there exist at most $\bar{c}$ corrupted rounds among the $3\bar{c}+1$ rounds of epoch ϕ, then at least $2\bar{c}+1$ are *uncorrupted*. We say that these rounds are part of the set U_ϕ. For all $t \in U_\phi$, the learner's hyperplanes $\{(\mathbf{x}_t, \omega_t)\}_{t \in U_\phi}$ pass from the same centroid $\boldsymbol{\kappa}_\phi$ and they all *protect* the region where $\boldsymbol{\theta}^\star$ lies. In other words, none among $\{(\mathbf{x}_t, \omega_t)\}_{t \in U_\phi}$ adds an undesirability point to $\boldsymbol{\theta}^\star$ (see e.g., Figure 3 for $\bar{c} = 1$ and each context appears only once). Since all hyperplanes point towards the same direction (i.e., the region containing $\boldsymbol{\theta}^\star$ never gets an undesirability point), starting from the region where $\boldsymbol{\theta}^\star$ lies and moving counter clockwise the undesirability levels of the formed regions first increase (moving from 0 to $2\bar{c} + 1$) and then decrease (moving from $2\bar{c}+1$ to 0). Due to this being a concave function, it is clear to see that there always exists a hyperplane with undesirability level at least $\bar{c} + 1$ in the entirety of one of its halfspaces. ■

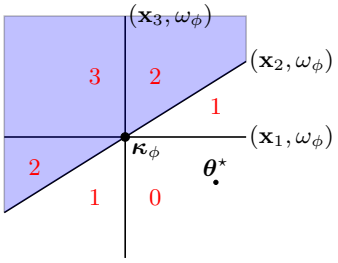


Figure 3: Sketch of the undesirability levels for epoch ϕ, after $2\bar{c}+1$ uncorrupted rounds, assuming that each context appears once. Red numbers denote the undesirability level of each region. The opaque region denotes the knowledge set for epoch $\phi + 1$.

Proposition B.16. For $d = 3$, any corruption $\bar{c}$, any centroid $\boldsymbol{\kappa}$, and any number of rounds N within an epoch, there exists a $\boldsymbol{\theta}^\star$ and a sequence $\{\mathbf{x}_t\}_{t \in [N]}$, such that there does not exist a hyperplane $(\mathbf{x}', \omega')$, where $\mathbf{x}' \in \{\mathbf{x}_t\}_{t \in [N]}$, with one of its halfspaces having undesirability at least $\bar{c} + 1$.

Proof. For any convex body $\mathcal{K}$ with centroid $\boldsymbol{\kappa}$, we show how to construct a problematic instance of a $\boldsymbol{\theta}^\star$ and N contexts. Fix the corruption level to be $\bar{c} = 1$, and $c_t = 0, \forall t \in [N]$. However, the learner does not know that none of the rounds is corrupted. Construct a sequence of contexts $\{\mathbf{x}_t\}_{t\in[N]}$ such that no two are equal and for $\omega_t = \langle \mathbf{x}_t, \boldsymbol{\kappa} \rangle, \forall t \in [N]$ we have that:

$$\{(\mathbf{x}_{t_1}, \omega_{t_1})\} \bigcap \{(\mathbf{x}_{t_2}, \omega_{t_2})\} = \boldsymbol{\kappa}$$

and the smallest region $r^\star$ that contains $\boldsymbol{\theta}^\star$ is defined by all $\{\mathbf{x}_t\}_{t\in[N]}$. Intuitively, these hyperplanes form a conic hull.

Take any hyperplane $h \in \mathbb{R}^3$ neither parallel nor orthogonal with any hyperplane $\{(\mathbf{x}_t, \omega_t)\}_{t\in[N]}$ such that $h \cap r^\star = q \neq \emptyset$. Take q's projection in $\mathbb{R}^2$. Observe that we have constructed an instance where no matter how big N is, there does not exist any hyperplane with undesirability at least $\bar{c}+1$ (i.e., 2 when $\bar{c} = 1$) in either one of its halfspaces. This instance easily generalizes for any $\bar{c} > 1$. ■

References

[AAK+20] Idan Amir, Idan Attias, Tomer Koren, Roi Livni, and Yishay Mansour. Prediction with corrupted expert advice. *Proceedings of 32nd Advances in Neural Processing Systems (NeurIPS)*, 2020. 4

[ARS14] Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 2014. 3

[BB20] Hamsa Bastani and Mohsen Bayati. Online decision making with high-dimensional covariates. *Oper. Res.*, 68(1):276–294, 2020. 1, 3

[BHK16] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. 2016. 9, 28

[BK17] Gah-Yi Ban and N Bora Keskin. Personalized dynamic pricing with machine learning. *Available at SSRN 2972985*, 2017. 3

[BKS20] Ilija Bogunovic, Andreas Krause, and Jonathan Scarlett. Corruption-tolerant gaussian process bandit optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. 4

[BR12] Josef Broder and Paat Rusmevichientong. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980, 2012. 3

[CBCP19] Nicolo Cesa-Bianchi, Tommaso Cesari, and Vianney Perchet. Dynamic pricing with finitely many unknown valuations. In *Algorithmic Learning Theory*, pages 247–273. PMLR, 2019. 3

[CKW19] Xi Chen, Akshay Krishnamurthy, and Yining Wang. Robust dynamic assortment optimization in the presence of outlier customers. *arXiv:1910.04183*, 2019. 4

[CLPL19] Maxime Cohen, Ilan Lobel, and Renato Paes Leme. Feature-based dynamic pricing. *Management Science*, 2019. 1, 3, 4, 5, 24, 25

[dB14] Arnoud V den Boer. Dynamic pricing with multiple products and partially specified demand distribution. *Mathematics of operations research*, 39(3):863–888, 2014. 3

[dB15] Arnoud V den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18, 2015. 3

[dBZ14] Arnoud V den Boer and Bert Zwart. Simultaneously learning and optimizing using controlled variance pricing. *Management science*, 60(3):770–783, 2014. 3

[EZKS16] Ehsan Emamjomeh-Zadeh, David Kempe, and Vikrant Singhal. Deterministic and probabilistic binary search in graphs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 519–532, 2016. 3

[GJL19] Negin Golrezaei, Patrick Jaillet, and Jason Cheuk Nam Liang. Incentive-aware contextual pricing with non-parametric market noise. *arXiv preprint arXiv:1911.03508*, 2019. 3

[GJM19] Negin Golrezaei, Adel Javanmard, and Vahab Mirrokni. Dynamic incentive-aware learning: Robust pricing in contextual auctions. In *Advances in Neural Information Processing Systems*, pages 9759–9769, 2019. 3

[GKT19] Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, 2019. 3

[JN19] Adel Javanmard and Hamid Nazerzadeh. Dynamic pricing in high-dimensions. *The Journal of Machine Learning Research*, 20(1):315–363, 2019. 3

[KK07] Richard M Karp and Robert Kleinberg. Noisy binary search and its applications. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 881–890, 2007. 3

[KL03] Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Symposium on Foundations of Computer Science*. IEEE, 2003. 3

[KZ14] N Bora Keskin and Assaf Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research*, 62(5):1142–1167, 2014. 3

[LG08] Thibault Le Guen. *Data-driven pricing*. PhD thesis, Massachusetts Institute of Technology, 2008. 3

[LLS19] Yingkai Li, Edmund Y Lou, and Liren Shan. Stochastic linear optimization with adversarial corruption. *arXiv:1909.02109*, 2019. 4

[LMPL18] Thodoris Lykouris, Vahab S. Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Symposium on Theory of Computing*, 2018. 1, 2, 3, 5, 9, 31

[LPLS21] Allen Liu, Renato Paes Leme, and Jon Schneider. Optimal contextual pricing and extensions. In *Symposium on Discrete Algorithms*, 2021. 1, 3, 25

[LPLV18] Ilan Lobel, Renato Paes Leme, and Adrian Vladu. Multidimensional binary search for contextual decision-making. *Operations Research*, 2018. 1, 2, 3, 4, 6, 7, 11, 15, 18, 19, 25, 26, 29, 30, 31

[LSSS19] Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. Corruption robust exploration in episodic reinforcement learning. *ArXiv*, abs/1911.08689, 2019. 4

[MPLS18] Jieming Mao, Renato Paes Leme, and Jon Schneider. Contextual pricing for lipschitz buyers. In *Advances in Neural Information Processing Systems*, 2018. 3

[Nov63] Albert B Novikoff. On convergence proofs for perceptrons. Technical report, 1963. 31

[NSLW19] Mila Nambiar, David Simchi-Levi, and He Wang. Dynamic learning and pricing with model misspecification. *Management Science*, 65(11):4980–5000, 2019. 3

[Pel02] Andrzej Pelc. Searching games with errors—fifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002. 3

[PLS18] Renato Paes Leme and Jon Schneider. Contextual search via intrinsic volumes. In *Symposium on Foundations of Computer Science*. IEEE, 2018. 1, 3, 4, 25

[QB16] Sheng Qiang and Mohsen Bayati. Dynamic pricing with demand covariates. *Available at SSRN 2765257*, 2016. 3

[RMK+80] Ronald L. Rivest, Albert R. Meyer, Daniel J. Kleitman, Karl Winklmann, and Joel Spencer. Coping with errors in binary search procedures. *Journal of Computer and System Sciences*, 20(3):396–404, 1980. 3

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958. 8

[SJB19] Virag Shah, Ramesh Johari, and Jose Blanchet. Semi-parametric dynamic contextual pricing. In *Advances in Neural Information Processing Systems*, pages 2360–2370, 2019. 3

[Spe92] Joel Spencer. Ulam's searching game with a fixed number of lies. *Theoretical Computer Science*, 95(2):307–321, 1992. 3

[Ula91] Stanislaw M Ulam. *Adventures of a Mathematician*. Univ of California Press, 1991. 3

[ZS19] Julian Zimmert and Yevgeny Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *arXiv preprint arXiv:1807.07623*, 2019. 3