

---

# Robust Contextual Pricing

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

We provide an algorithm with regret  $O(Cd \log \log T)$  for contextual pricing with  $C$  corrupted rounds, improving over the previous bound of  $O(d^3 C \log^2(T))$  of Krishnamurthy et al. (2020). The result is based on a reduction that calls the uncorrupted algorithm as a black-box, unlike the previous approach that modifies the inner workings of the uncorrupted algorithm. As a result, it leads to a conceptually simpler algorithm.

Finally, we provide a lower bound ruling out a  $O(C + d \log \log T)$  algorithm. This shows that robustifying contextual pricing is harder than robustifying contextual search with  $\epsilon$ -ball losses, for which it is possible to design algorithms where corruptions add only an extra additive term  $C$  to the regret.

## 1 Introduction

In the *contextual pricing* problem, the seller is a learner who, in each round, gets a different product described by a feature vector, for which it then posts a price. At this point, it observes if the buyer purchases the product or not. The goal of the learner (i.e., seller) is to maximize their revenue. As usual in pricing problems, the learner faces the exploration/exploitation trade-off: it can “explore” by setting aggressive prices that allow it to learn about the buyer’s valuation, but then it risks losing the sale; or it can “exploit” by setting conservative prices that guarantee a sale, but potentially get a suboptimal revenue. In the revenue management literature, this is referred to as the *learn-and-earn* problem.

Pricing problems pose two main challenges:

- (i) we only observe binary feedback (*sale* or *no-sale*), but do not directly observe the loss in each round;
- (ii) the loss function is *asymmetric* and *discontinuous*: while under-pricing an item by an amount  $x$  will result in a sale with loss  $x$  with respect to the best possible revenue, over-pricing will cause the buyer not to purchase the produce and incur a large constant loss.

The seminal paper of Kleinberg and Leighton (2003) identifies the optimal way to handle this asymmetry, obtaining the optimal regret of  $\Theta(\log \log T)$  for the single-dimensional non-contextual version of this problem. Since then, this paper has been extended to the  $d$ -dimensional contextual case by a series of papers that have obtained increasingly better bounds: the work of Cohen et al. (2016) gave  $O(d^2 \log T)$ , Leme and Schneider (2018) showed  $O(d^4 \log \log T)$ , and Liu et al. (2021) gave  $O(d \log \log T)$ , matching the lower bound.

So far, our discussion has described the realizable and noise-free version of the problem: the buyer has a fixed value  $v_t$ , which is a constant in  $[0, 1]$  in the non-contextual case, or a linear function  $v_t(x_t) = \langle v^*, x_t \rangle$  in the contextual case. Moreover, the purchasing decision is based on whether the posted price  $p_t$  is above or below this value  $v_t$ . There has been considerable work extending the

model to handle stochastic noise models: see, e.g., Javanmard and Nazerzadeh (2016); Cohen et al. (2016); Javanmard (2017); Shah et al. (2019); Liu et al. (2021); Xu and Wang (2021, 2022).

Comparatively, adversarial noise modes (also known as *corrupted feedback*) are much less understood. The study of corrupted feedback for contextual pricing was pioneered by Krishnamurthy et al. (2020). In this model, there is still a ground truth value  $v_t$  and the buyer again makes most of the purchasing decisions based on whether  $p_t \leq v_t$ . However, for a certain number of rounds  $C$ , the decisions are arbitrary. This can model that buyers often make irrational choices (as in Krishnamurthy et al. (2020)) or that an adversary can corrupt the feedback given to the learner (as in Lykouris et al. (2018); Paes Leme et al. (2022)).

## 1.1 Our Results

We give improved algorithms and lower bounds for the contextual pricing problem.

**Our Results: Upper Bounds.** Our paper proposes a reduction from corrupted to uncorrupted case in contextual pricing and through that, obtains conceptually simpler algorithms with improved regret. Our results are the following:

1. For the *known corruption setting*, We give an algorithm that achieves a regret of  $O((C + 1) \cdot d \log \log T)$ . This improves over the  $O(Cd^3 \log^2 T)$  bound of Krishnamurthy et al. (2020).
2. For the setting of *unknown corruption  $C$  with a known upper bound  $U$*  (i.e., we are promised that the actual number of corruptions  $C$  is at most  $U$ ), we give an algorithm with regret  $O((C + \log U)d \log \log T)$ . This improves over the regret bound of  $O((C + \log U)d^3 \log^2 T)$  given by Krishnamurthy et al. (2020).

At a high level, we keep parallel copies of an uncorrupted contextual pricing algorithm and query each of them in each round for advice on which price to set, choosing the maximum of such prices to post. In the case of a sale, we choose a random algorithm in the pool to update. We set the update probabilities in a way that one of the parallel copies will never observe a corruption with high probability. Then we bound the regret of the entire algorithm by comparing it with the regret of the instance that never observes a corruption.

The main observation is to exploit the following property of the pricing loss: if we choose the maximum price and it results in a sale, the loss incurred is at most the loss of any given algorithm. This allows us to charge the loss of the various algorithms to the loss of a single uncorrupted algorithm.

**Our Results: Lower Bounds.** The lower bound on the regret for the uncorrupted case is  $\Omega(d \log \log T)$ , so the best regret bound one can hope for is  $O(C + d \log \log T)$ . For the related problem of contextual search with  $\epsilon$ -ball loss (i.e., with a different loss function), Paes Leme et al. (2022) show a regret bound of  $O(C + d \log(1/\epsilon))$  which adds only an additive  $C$  overhead with respect to the tight bound of  $O(d \log(1/\epsilon))$  for the uncorrupted case. The natural question is: can we hope to achieve algorithms with regret  $O(C + d \log \log T)$ ?

In Theorem 5.2 we show this is not possible: we give a lower bound ruling out algorithms with regret  $O(C + d \log \log T)$ , and hence showing that robustifying contextual pricing is harder than robustifying other contextual search problems (e.g., for the  $\epsilon$ -ball loss function).

These lower bounds also inspire new algorithmic results: using the ideas in the lower bound, we are able to provide in Theorem 6.1 a  $O(C + \log T)$ -regret algorithm for the *corrupted non-contextual case* when corruptions are one-sided: i.e., the adversary is allowed to corrupt no-sales to sales but not the other way round.

## 1.2 Related Work

Our paper lies in the intersection of the literature of learning to price, contextual search, and learning with corrupted feedback. Learning to price was initiated by Kleinberg and Leighton (2003) who provide matching upper and lower bounds for the regret achievable by a seller posting prices to a buyer under different assumptions on the buyer's valuation. This setting was later extended to auctions Mohri and Medina (2014); Cesa-Bianchi et al. (2014) and strategic agents Amin et al. (2013); Drutsa (2017).

Contextual pricing is a generalization of learning to price to higher dimensional settings, which is a part of a larger class of problems known as contextual search which explore binary feedback in higher dimensional settings for various loss functions. A long line of paper starting with Amin et al. (2014) provide increasingly better bounds for contextual search with the pricing loss and other types of losses: Cohen et al. (2016); Lobel et al. (2018); Leme and Schneider (2018); Liu et al. (2021).

Finally, learning with corruptions correspond to the study of settings in which the data follows a predictable pattern with the exception of a few examples whose label is arbitrary. A very early example is called Ulam’s game (Ulam (1976)) in which a learner tried to identify a number on a sorted list using comparison queries but  $C$  of those queries can have erroneous answers. Upper and lower bounds to this setting were given by Rivest et al. (1978).

There has been recent interest in providing corruption robust bounds to contextual search problems. This was pioneered by Krishnamurthy et al. (2020) who study contextual pricing in the presence of corruptions – the exact same setting as in our paper. Recently, Paes Leme et al. (2022) studies the contextual search problem under the symmetric and  $\epsilon$ -ball losses with corrupted feedback.

## 2 Model and Definitions

**(Uncorrupted) Contextual Pricing** In the contextual pricing problem, the buyer’s valuation is represented by a vector  $v^* \in \mathbb{B}_d = \{x \in \mathbb{R}^d; \|x\|_2 \leq 1\}$  that is unknown to the learner. In each round  $t = 1 \dots T$ , the learner will be presented an item described by a feature vector  $x_t \in \mathbb{B}$  and will have the opportunity to set a price  $p_t$  for the buyer. In the standard (un-corrupted) model, the buyer buys if the value  $\langle v^*, x_t \rangle \geq p_t$  in which case the learner obtains revenue  $p_t$ ; otherwise, if the value  $\langle v^*, x_t \rangle < p_t$  then the buyer doesn’t buy, in which case the learner obtains revenue zero. As usual, the learner is faced with the trade-off of proposing aggressive prices to learn more information about  $v^*$ , and setting conservative prices that will be guaranteed to sell and generate revenue. To summarize, the feedback  $\sigma_t \in \{0, 1\}$  and the loss  $\ell_t$  are given by:

$$\sigma_t = \mathbf{1}\{v_t \geq p_t\} \quad \ell_t(p_t; v_t) := \sigma_t p_t - v_t$$

for  $v_t = \langle v^*, x_t \rangle$ . Liu et al. (2021) recently showed matching upper and lower bounds of  $O(d \log \log T)$  on the total loss  $\sum_t \ell_t$  (also referred to as the *regret*) which is a contextual version of the algorithm of Kleinberg and Leighton (2003).

**Corrupted Feedback** In the corrupted version of this problem, the learner has access to feedback  $\sigma_t \in \{0, 1\}$ . The corruption level  $C$  is defined to be the number of periods for which the feedback doesn’t match  $\mathbf{1}\{v_t \geq p_t\}$ , i.e.,

$$C = |\{t \in [T]; \sigma_t \neq \mathbf{1}\{v_t \geq p_t\}\}|$$

In the known corruption case, the algorithm is given access to  $C$ . In the unknown corruption case, the algorithm knows an upper bound  $U$  on  $C$  (e.g.,  $U = T$  in the worst case), and the regret bounds can depend on both  $C$  (the actual corruption), and on  $U$  (the upper bound, ideally logarithmically or better).

## 3 Reduction in the Known-Corruption Setting

We will present a reduction from the uncorrupted setting to the corrupted setting for contextual pricing. In this reduction, we will treat the uncorrupted algorithm as a black-box described by two subroutines: **Query**( $x_t$ ) which returns a price  $p_t$  and **Update**( $x_t, p_t, \sigma_t$ ). Given those subroutines, the uncorrupted algorithm can be described as follows: (i) **Query**( $x_t$ ) returns a price  $p_t$  and doesn’t modify the internal state of the algorithm; (ii) the learner posts price  $p_t$  and observes feedback  $\sigma_t$ ; (iii) **Update**( $x_t, p_t, \sigma_t$ ) updates the state of the algorithm.

Given an algorithm defined by **Query**( $x_t$ ) and **Update**( $x_t, p_t, \sigma_t$ ) and  $q \in [0, 1]$  we define its  $q$ -slowed-down version as the algorithm that prices according to **Query**( $x_t$ ) but only calls **Update**( $x_t, p_t, \sigma_t$ ) with probability  $q_t \geq q$  upon receiving feedback. With remaining probability the state of the algorithm is unchanged.

**Theorem 3.1.** If there exists an uncorrupted algorithm with loss at most  $R(T)$  such that:

1. its suggested price results in at most  $R(T)$  no sales
  2. the loss of its  $q$ -slowed-down version is at most  $R(T)/q$
- then there exists an algorithm with total loss at most  $O(C \cdot R(T))$  in the corrupted setting with known corruptions.

**Algorithm Description.** We instantiate  $C + 1$  copies of the uncorrupted algorithm denoting the  $i$ -th copy by  $\text{Query}_i(x_t)$  and  $\text{Update}_i(x_t, p_t, \sigma_t)$ . We also maintain a set of active instances  $A$  that is initially  $\{1, 2, \dots, C + 1\}$ . Now, for each  $t = 1 \dots T$  we:

- obtain the price proposed by each algorithm  $p_{t,i} = \text{Query}_i(x_t)$ .
- choose the largest price  $p_t = \max_{i \in A} p_{t,i}$  to post.
- observe feedback  $\sigma_t$ 
  - if  $\sigma_t = 0$  (no-sale), let  $i \in A$  be the index such that  $p_t = p_{t,i}$ . Then call  $\text{Update}_i(x_t, p_t, \sigma_t)$ . If the  $i$ -th algorithm was updated with more than  $R(T)$  no sales, then remove  $i$  from the active set  $A$ .
  - if  $\sigma_t = 1$  (sale), then let  $i$  be a random index from  $A$  and call  $\text{Update}_i(x_t, p_t, \sigma_t)$ .

*Proof.* First we observe that there is at least one index  $i^*$  that never observes a corruption, hence the guarantees of the uncorrupted algorithm apply to it. By the first assumption in Theorem 3.1, this algorithm is never eliminated from the active set.

To bound the total loss, we first observe that the total loss from periods which resulted in a no-sale is at most  $(C + 1)R(T)$  since each copy of the algorithm can generate at most  $R(T)$  no-sales until it is eliminated.

Now, to bound the total loss from periods in which there was a sale, observe that if the maximum price  $p_t$  resulted in a sale, then the price under  $i^*$  would also have resulted in a sale since  $p_{t,i^*} \leq p_t$  and furthermore, the loss  $\ell(p_t; v_t) \leq \ell(p_{t,i^*}; v_t)$ . Hence, the expected loss from the periods resulting in sales can be upper bounded by the loss of a  $(C + 1)^{-1}$ -slowed-down version of the copy  $i^*$  of the algorithm which is  $(C + 1) \cdot R(T)$ .  $\square$

### 3.1 Analysis of the Slowed Down Algorithm

To conclude the proof, we show that the uncorrupted algorithm of Liu et al. (2021) satisfies the conditions of properties in Theorem 3.1. We start by giving an overview of that procedure:

**Summary of the algorithm of Liu et al. (2021)** The algorithm keeps a set  $S_t \subseteq \mathbb{B}$  of vectors  $v$  consistent with the feedback observed so far. It also keeps  $k = \log \log T$  potential functions  $\Phi_1(S_t), \dots, \Phi_k(S_t)$  associated with the doubly-exponential intervals  $I_j = [1/2^{2^j}, 1/2^{2^{j-1}}]$  for  $j > 1$  and  $I_1 = [1/4, 1]$  for  $j = 1$ . The potential functions depend only on the set  $S_t$ .

In each round  $t$ , the algorithm computes the width

$$\text{width}(S_t; x_t) = \max_{v \in S_t} \langle v, x_t \rangle - \min_{v \in S_t} \langle v, x_t \rangle$$

If the width is below  $1/T$ , the algorithm prices at the lower bound  $\min_{v \in S_t} \langle v, x_t \rangle$  guaranteeing a sale with loss at most  $1/T$ . Otherwise, we choose an index  $j$  such that  $\text{width}(S_t; x_t) \in I_j$ . The algorithm then chooses a price  $p_t$  as a function of  $j$ ,  $S_t$  and  $x_t$  and update the set  $S_t$  to

$$S_{t+1} = \{v \in S_t; (2\sigma_t - 1)(\langle v, x_t \rangle - p_t) \geq 0\}$$

- if there is a sale, then the loss is at most  $\text{width}(S_t, x_t) = O(1/2^{2^j})$  and the  $j$ -th potential decreases by  $\Phi_j(S_{t+1}) \leq (1 - O(1/2^{2^j})) \cdot \Phi_j(S_{t+1})$ . The remaining potentials weakly decrease.
- if there is no sale, then the loss is at most 1 and the  $j$ -th potential decreases by  $\Phi_j(S_{t+1}) \leq O(1/2^{2^j}) \cdot \Phi_j(S_{t+1})$ . The remaining potentials weakly decrease.

156 Finally, the potentials are such that

$$\Phi_j(S_t) \geq 1/2^{\Omega(d \cdot 2^j)},$$

157 which bounds the number of possible sales and no-sales for  $\text{width}(S_t; x_t) \geq 1/T$ . In particular, there  
158 can be at most

$$N_j^s := O(d \log d 2^{2^{j-1}} + 2^{j+2^{j-1}})$$

159 sales and

$$N_j^n := O(d \log d / 2^{j-1} + d)$$

160 no-sales whenever the width is in the interval  $I_j$ . This leads to the overall bound of

$$\sum_j N_j^n + (1/2^{2^j}) \cdot N_j^s = O(d \log \log T + d \log d).$$

161 **Lemma 3.2.** *The uncorrupted algorithm of Liu et al. (2021) satisfies the properties in Theorem 3.1*  
162 *for  $R(T) = O(d \log \log T + d \log d)$ .*

163 *Proof.* The first property is trivially satisfied since no-sales are accounted as a loss of 1 in the analysis  
164 above. Now, to analyze the  $q$ -slowed down algorithm we can keep the same potentials  $\Phi_j(S_t)$  with  
165 the difference that they are updated with probability  $q$ . The time between two updates of each  
166 potential is a geometric random variable with probability  $q$ . So the expected number queries with  
167 width in  $I_j$  with a sale is  $N_j^s/q$ , for  $N_j^s$  defined above in the algorithm summary. The total loss from  
168 sales is therefore at most  $(N_j^s/q) \cdot 1/2^{2^j}$  in expectation. The lemma follows from the fact that

$$\sum_j N_j^s \cdot 1/2^{2^j} \leq O(d \log \log T + d \log d)$$

169 by the analysis of the uncorrupted algorithm. □

## 170 4 Unknown-Corruption Setting

171 Here we assume that we have only an upper bound  $U$  on the number of corruptions, i.e., we know  
172 that  $C \leq U$  but don't know the value of  $C$ . We provide an algorithm that depends linearly on  $C$  (the  
173 actual corruptions) but only logarithmically on  $U$ . The ideas will be to use a version of the multi-layer  
174 approach of Lykouris et al. (2018), in which  $O(\log U)$  copies of the original uncorrupted algorithm  
175 are run in parallel and each copy is updated with exponentially smaller probability. The copies in  
176 which the update probability is smaller than  $1/C$  are unlikely to experience a corruption.

177 **Algorithm Description** For a parameter  $\delta \in [0, 1]$ , we instantiate  $k = \log(U/\delta)$  copies of the  
178 uncorrupted algorithm denoting the  $i$ -th copy by  $\text{Query}_i(x_t)$  and  $\text{Update}_i(x_t, p_t, \sigma_t)$ . We also  
179 maintain a set of active instances  $A$  that is initially  $\{1, 2, \dots, k\}$ . Now, for each  $t = 1 \dots T$  we:

- 180 • obtain the price proposed by each algorithm  $p_{t,i} = \text{Query}_i(x_t)$ .
- 181 • choose the largest price  $p_t = \max_{i \in A} p_{t,i}$  to post.
- 182 • observe feedback  $\sigma_t$ 
  - 183 – if  $\sigma_t = 0$  (no-sale), let  $i \in A$  be the index such that  $p_t = p_{t,i}$ . Then call
  - 184  $\text{Update}_i(x_t, p_t, \sigma_t)$ . If the  $i$ -th algorithm was updated with more than  $R(T)$  no sales,
  - 185 then remove all indices smaller or equal to  $i$  from the active set  $A$ .
  - 186 – if  $\sigma_t = 1$  (sale), then choose an index  $i$  with probability  $1/2^i$  and call
  - 187  $\text{Update}_i(x_t, p_t, \sigma_t)$ .

188 We note that the algorithm only differs from the known-corruption case in the probabilities used  
189 to update each algorithm. Also note that with non-zero probability we update algorithm that is no  
190 longer active. Such update is clearly useless, but we do so in order to preserve the probability that the  
191 still-active algorithms are updated.

192 **Theorem 4.1.** Given an uncorrupted algorithm with regret  $R(T)$  following the assumptions of  
193 Theorem 3.1 and a known upper bound  $U$  on the (unknown) level of corruption  $C$ , there for every  
194  $\delta > 0$ , there exists an algorithm that with probability  $1 - \delta$  has total loss at most  $O((C/\delta) +$   
195  $\log(U/\delta)) \cdot R(T)$ .

196 *Proof.* Let  $i^*$  be the index such that  $C/(2\delta) \leq 2^{i^*} \leq C/\delta$ . For each corrupted round  $t$  we update  
 197  $i^*$  with probability  $O(\delta/C)$ . Across all the rounds, the probability it experiences no corruption is  
 198  $(1 - \delta/C)^C = 1 - O(\delta)$ . With that probability, this algorithm is never eliminated from the active set  
 199  $A$  and the bounds from the uncorrupted algorithm will hold.

200 We again bound the total loss of sales and no-sales separately. Since each algorithm can incur at most  
 201  $R(T)$  no sales before it is eliminated, the total loss from no-sales is at most  $\log(U/\delta) \cdot R(T)$ .

202 To bound the loss from sales, observe that if the maximum price resulted in a sale, then  $p_{t,i^*} \leq p_t$   
 203 must also result in a sale, so the loss is at most the loss of the loss of a  $(\delta/C)$ -slowed down version of  
 204 the uncorrupted algorithm with is at most  $(C/\delta) \cdot R(T)$ .  $\square$

## 205 5 Lower Bound

206 We will study lower bounds for the special case of  $d = 1$  in which the valuation is a scalar and is  
 207 non-contextual. This is the original setting of Kleinberg and Leighton (2003). The lower bound for  
 208 the uncorrupted pricing problem is  $\Omega(\log \log T)$ . The best upper bound we can hope for (even with a  
 209 known number of corruptions  $C$ ) is  $O(C + \log \log T)$ . The following lemma rules out this bound by  
 210 a reduction to the lower bound for the *online product design* problem given by Emamjomeh-Zadeh  
 211 et al. (2021).

212 **Theorem 5.1.** No deterministic algorithm for the one-dimensional contextual pricing problem with  
 213 (known) corrupted feedback can guarantee total loss at most  $O(C + \log \log T)$ . In particular, when  
 214  $C = \Theta(\log \log T)$ , any deterministic algorithm must incur at least  $\Omega(C^2 / \log C)$  total loss.

215 We will prove Theorem 5.1 via reduction to the previously studied problem of *online product design*  
 216 by Emamjomeh-Zadeh et al. (2021). Online product design is a variant of contextual pricing where  
 217 there are  $n$  different possible items, each with an unknown (one-dimensional) true value  $v_i \in [0, 1]$ .  
 218 At each round  $t \in [T]$ , the learner can choose an item  $i_t \in [d]$  and set a price  $p_t \in [0, 1]$  for this item.  
 219 If  $p_t \leq v_{i_t}$ , the item sells at this price and the learner obtains revenue  $p_t$ , otherwise the item goes  
 220 unsold and the learner obtains revenue 0. The learner would like to minimize their regret compared  
 221 to their optimal strategy, which is to always select the item  $i^*$  with the largest value  $v_{i^*}$  and price it at  
 222 its value.

223 Emamjomeh-Zadeh et al. (2021) present an algorithm for this problem (a parallelization of the  
 224 algorithm of Kleinberg and Leighton (2003)) that incurs at most  $O(n \log \log T)$  regret. More  
 225 relevantly for us, they show that this multiplicative dependence is necessary by establishing the  
 226 following lower bound.

227 **Theorem 5.2** (Theorem 8 of Emamjomeh-Zadeh et al. (2021)). If  $n = \Theta(\log \log T)$ , then any  
 228 deterministic algorithm for online product design must incur  $\Omega(n^2 / \log n)$  regret in the worst case.

229 We will show that it is possible to reduce the problem of online product design to the problem of  
 230 contextual pricing with corruptions. In particular, we will show that any algorithm for contextual  
 231 pricing with  $C$  corruptions can be applied to get an equivalent regret bound in the  $C$ -item online  
 232 product design problem.

233 **Lemma 5.3.** *If there exists a deterministic algorithm incurring at most  $R(C, T)$  regret for the*  
 234 *contextual pricing problem with  $C$  corruptions, then there exists a deterministic algorithm incurring*  
 235 *at most  $R(C, T)$  regret for the online product design problem with  $n = C + 1$  items.*

236 *Proof.* Consider the following algorithm for online product design (that uses the corrupted contextual  
 237 pricing algorithm as a black box). At any point it will maintain a state consisting of  $n + 1$  numbers:  
 238  $x \in [0, 1]$ , the lowest price that any item has ever sold for, and  $y_1, y_2, \dots, y_n \in [0, 1]$ , where  $y_i$  is  
 239 the highest price that item  $i$  has *not* successfully sold for. Initially,  $x$  is initialized to 0 and all  $y_i$  are  
 240 initialized to 1.

241 The algorithm repeats the following procedure for  $T$  steps:

- 242 1. Choose an item  $i_t \in \arg \max_{i \in [n]} y_i$  with a maximal unsold price.
- 243 2. Ask the corrupted contextual pricing subroutine for a query price  $p_t$ .

- 244 3. Post price  $p_t$  for item  $i_t$ .
- 245 4. If item  $i_t$  sells at this price, update  $x \leftarrow \max(x, p_t)$ , and report that the item sold to the
- 246 contextual pricing subroutine.
- 247 5. If item  $i_t$  fails to sell at this price, update  $y_{i_t} \leftarrow \min(y_{i_t}, p_t)$ , and report that the item failed
- 248 to sell to the contextual pricing subroutine.

249 We first argue that the feedback provided to the contextual pricing subroutine over the course of  
 250 this process must be consistent with some valid transcript for contextual pricing with at most  $C$   
 251 corruptions. In particular, let  $Y$  be the maximum value of all  $y_i$  at the end of the  $T$  rounds, and  
 252 consider the scenario where the true value of the item in the corrupted contextual pricing subroutine  
 253 is equal to  $Y - \epsilon$  (for an arbitrary small  $\epsilon > 0$  guaranteeing  $Y - \epsilon > x$ ). We claim this is consistent  
 254 with all but at most  $n - 1 = C$  observations reported to this subroutine. To see this, assume without  
 255 loss of generality that  $Y$  is the value of  $y_1$  at the end of  $T$  rounds. We claim that any other item  
 256 can fail to sell at a price below  $Y$  at most once – once it does so, it updates its  $y_i$  to a value below  
 257  $Y$ , and would only be picked over  $y_1$  once  $y_1$  drops below  $Y$  (which it never does). Since there are  
 258  $n - 1 = C$  other items, there are at most  $C$  such events like this, which would constitute corrupted  
 259 observations reported to the contextual pricing subroutine. On the other hand, since  $x$  remains below  
 260  $Y$  throughout this process, we never report an error in the other direction (a sale at a price above  $Y$ ),  
 261 and hence these are the only such corruptions.

262 The regret of the corrupted contextual pricing subroutine under this scenario is given by

$$(Y - \epsilon)T - \sum_{t=1}^T \sigma_t p_t.$$

263 where, as before, we write  $\sigma_t = 1$  if the item sells and  $\sigma_t = 0$  otherwise. But now, note that the  
 264 maximum value of any item in the optimal product design instance is at most  $Y$ , so the regret incurred  
 265 by the online product design algorithm is at most

$$YT - \sum_{t=1}^T \sigma_t p_t.$$

266 It follows (by taking  $\epsilon \rightarrow 0$ ) that any regret bound  $R(T)$  for the corrupted contextual pricing  
 267 subroutine extends to this algorithm for online product design.  $\square$

268 *Proof of Theorem 5.1.* Follows immediately from Theorem 5.2 and Lemma 5.3.  $\square$

## 269 6 One Sided Error: the Cautious Buyer

270 Finally, we use the ideas of the lower bound to derive an algorithm with regret  $O(C + \log T)$  for  
 271 what we call an "cautious buyer", which is a buyer who may not buy even if the price is below their  
 272 value but at most  $C$  times. The buyer never buys above their value. This corresponds to a corrupted  
 273 setting with one-sided corruption:  $\sigma_t \leq \mathbf{1}\{v_t \geq p_t\}$ .

274 As in the previous section, we will state and prove this result in the non-contextual setting where the  
 275 buyer's valuation can be expressed by a scalar  $v^* \in [0, 1]$ .

276 **Theorem 6.1.** There is a learning algorithm for posting a price to an cautious buyer (one-sided  
 277 corrupted feedback with at most  $C$  corruptions) with regret  $O(C + \log T)$ .

278 The theorem will use the algorithm by Kirkpatrick and Gao (1990) to for the following problem:  
 279 consider a list of numbers  $[v_1, \dots, v_n]$ , all in  $[0, 1]$ , unknown to the learner. The learner can at each  
 280 time  $t$  choose an index  $i_t$  and price  $p_t$  and query whether  $p_{i_t} \leq v_{i_t}$ . The goal of the learner is to  
 281 identify an index  $i$  and a price  $p$  such that:  $v_i \geq p \geq \max_i v_i - \epsilon$ . Kirkpatrick and Gao (KG) provide  
 282 an algorithm to accomplish this task with  $O(n + \log(1/\epsilon))$  comparisons.

283 *Proof.* Apply the KG algorithm find the maximum of a list of  $C+1$  numbers with parameter  $\epsilon = 1/T$ .  
 284 For every query  $p_{i_t} \leq v_i$  of the KG algorithm, respond with the result of a query  $p_t \leq v^*$  to the  
 285 corrupted contextual pricing algorithm. Because the buyer is cautious, the buyer may not buy a  
 286 product even though the price is below  $v^*$  hence the feedback for some items in the list will be  
 287 consistent with a value  $v_i \leq v^*$  since the KG algorithm only queries prices above the last sale and  
 288 below the last no-sale for each item. For at least one of the items, however, the algorithm won't  
 289 observe a corruption, so it will be able to identify the  $v^*$  within a margin of  $\epsilon = 1/T$  in  $O(C + \log T)$   
 290 rounds. In such rounds, it will incur that much regret. From then on, the algorithm will incur at most  
 291  $O(1/T)$  regret per round.  $\square$

292 **Open Problems** We note that the lower bound in Theorem 5.2 applies to the cautious buyer as  
 293 well, hence it is not possible to improve this bound to  $O(C + \log \log T)$ . However, we leave as an  
 294 open question whether a bound of  $O(C + \log T)$  is possible for a buyer with  $C$  corruptions in either  
 295 direction. Another interesting open direction is to provide a  $O(C + d \log T)$  algorithm for a cautious  
 296 buyer in the contextual setting.

## 297 References

- 298 Amin, K., Rostamizadeh, A., and Syed, U. (2013). Learning prices for repeated auctions with  
 299 strategic buyers. *Advances in neural information processing systems*, 26.
- 300 Amin, K., Rostamizadeh, A., and Syed, U. (2014). Repeated contextual auctions with strategic  
 301 buyers. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural*  
 302 *Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages  
 303 622–630.
- 304 Cesa-Bianchi, N., Gentile, C., and Mansour, Y. (2014). Regret minimization for reserve prices in  
 305 second-price auctions. *IEEE Transactions on Information Theory*, 61(1):549–564.
- 306 Cohen, M. C., Lobel, I., and Paes Leme, R. (2016). Feature-based dynamic pricing. In *Proceedings*  
 307 *of the 2016 ACM Conference on Economics and Computation*, pages 817–817. ACM.
- 308 Drutsa, A. (2017). Horizon-independent optimal pricing in repeated auctions with truthful and  
 309 strategic buyers. In *Proceedings of the 26th International Conference on World Wide Web*, pages  
 310 33–42.
- 311 Emamjomeh-Zadeh, E., Leme, R. P., Schneider, J., and Sivan, B. (2021). Jointly learning prices and  
 312 product features. In *IJCAI*, pages 2360–2366.
- 313 Javanmard, A. (2017). Perishability of data: dynamic pricing under varying-coefficient models. *The*  
 314 *Journal of Machine Learning Research*, 18(1):1714–1744.
- 315 Javanmard, A. and Nazerzadeh, H. (2016). Dynamic pricing in high-dimensions. *Working paper,*  
 316 *University of Southern California*.
- 317 Kirkpatrick, D. G. and Gao, F. (1990). Finding extrema with unary predicates. In *International*  
 318 *Symposium on Algorithms*, pages 156–164. Springer.
- 319 Kleinberg, R. and Leighton, T. (2003). The value of knowing a demand curve: Bounds on regret  
 320 for online posted-price auctions. In *Foundations of Computer Science, 2003. Proceedings. 44th*  
 321 *Annual IEEE Symposium on*, pages 594–605. IEEE.
- 322 Krishnamurthy, A., Lykouris, T., and Podimata, C. (2020). Corrupted multidimensional binary search:  
 323 Learning in the presence of irrational agents. *arXiv preprint arXiv:2002.11650*.
- 324 Leme, R. P. and Schneider, J. (2018). Contextual search via intrinsic volumes. In *59th IEEE Annual*  
 325 *Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*,  
 326 pages 268–282.
- 327 Liu, A., Leme, R. P., and Schneider, J. (2021). Optimal contextual pricing and extensions. In  
 328 *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1059–  
 329 1078.



- 330 Lobel, I., Leme, R. P., and Vladu, A. (2018). Multidimensional binary search for contextual decision-  
331 making. *Operations Research*, 66(5):1346–1361.
- 332 Lykouris, T., Mirrokni, V., and Paes Leme, R. (2018). Stochastic bandits robust to adversarial  
333 corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*,  
334 pages 114–122.
- 335 Mohri, M. and Medina, A. M. (2014). Learning theory and algorithms for revenue optimization  
336 in second price auctions with reserve. In *International conference on machine learning*, pages  
337 262–270. PMLR.
- 338 Paes Leme, R., Podimata, C., and Schneider, J. (2022). Corruption-robust contextual search through  
339 density updates. In *Conference on Learning Theory*, pages 3504–3505. PMLR.
- 340 Rivest, R. L., Meyer, A. R., and Kleitman, D. J. (1978). Coping with errors in binary search  
341 procedures (preliminary report). In *Proceedings of the tenth annual ACM symposium on Theory of*  
342 *computing*, pages 227–232.
- 343 Shah, V., Johari, R., and Blanchet, J. (2019). Semi-parametric dynamic contextual pricing. *Advances*  
344 *in Neural Information Processing Systems*, 32.
- 345 Ulam, S. M. (1976). *Adventures of a Mathematician*. Univ of California Press.
- 346 Xu, J. and Wang, Y.-X. (2021). Logarithmic regret in feature-based dynamic pricing. *Advances in*  
347 *Neural Information Processing Systems*, 34.
- 348 Xu, J. and Wang, Y.-X. (2022). Towards agnostic feature-based dynamic pricing: Linear policies vs  
349 linear valuation with unknown noise. *arXiv preprint arXiv:2201.11341*.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Yes, this is a theoretical paper in which all the claims are backed by mathematical proofs.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We provide open problems making explicit the questions that are not fully solved by the paper.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide proofs to all the claims and checked them to the best of our abilities.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: No experimental results.

### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not include experiments requiring code.

### 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper does not include experiments.

### 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

397 **9. Code of ethics**

398 Question: Does the research conducted in the paper conform, in every respect, with the  
399 NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

400 Answer: [Yes]

401 Justification: Theoretical research with no ethics red flags.

402 **10. Broader impacts**

403 Question: Does the paper discuss both potential positive societal impacts and negative  
404 societal impacts of the work performed?

405 Answer: [Yes]

406 Justification: More efficient marketplaces and pricing can lead to positive societal effects.  
407 Since it is mostly a theoretical investigation on an abstract setting, we believe the societal  
408 impact is tangential.

409 **11. Safeguards**

410 Question: Does the paper describe safeguards that have been put in place for responsible  
411 release of data or models that have a high risk for misuse (e.g., pretrained language models,  
412 image generators, or scraped datasets)?

413 Answer: [NA]

414 Justification: The paper poses no such risks.

415 **12. Licenses for existing assets**

416 Question: Are the creators or original owners of assets (e.g., code, data, models), used in  
417 the paper, properly credited and are the license and terms of use explicitly mentioned and  
418 properly respected?

419 Answer: [NA]

420 Justification: The paper does not use existing assets.

421 **13. New assets**

422 Question: Are new assets introduced in the paper well documented and is the documentation  
423 provided alongside the assets?

424 Answer: [NA]

425 Justification: The paper does not release new assets.

426 **14. Crowdsourcing and research with human subjects**

427 Question: For crowdsourcing experiments and research with human subjects, does the paper  
428 include the full text of instructions given to participants and screenshots, if applicable, as  
429 well as details about compensation (if any)?

430 Answer: [NA]

431 Justification: The paper does not involve crowdsourcing nor research with human subjects.

432 **15. Institutional review board (IRB) approvals or equivalent for research with human**  
433 **subjects**

434 Question: Does the paper describe potential risks incurred by study participants, whether  
435 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)  
436 approvals (or an equivalent approval/review based on the requirements of your country or  
437 institution) were obtained?

438 Answer: [NA]

439 Justification: The paper does not involve crowdsourcing nor research with human subjects.

440 **16. Declaration of LLM usage**

441 Question: Does the paper describe the usage of LLMs if it is an important, original, or  
442 non-standard component of the core methods in this research? Note that if the LLM is used  
443 only for writing, editing, or formatting purposes and does not impact the core methodology,  
444 scientific rigor, or originality of the research, declaration is not required.

445

Answer: [NA]

446

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

447