Top-k eXtreme Contextual Bandits with Arm Hierarchy

Top-k eXtreme Contextual Bandits with Arm Hierarchy

Rajat Sen[1] Alexander Rakhlin[2,3] Lexing Ying[4,3] Rahul Kidambi[3]
Dean Foster[3] Daniel Hill[3] Inderjit Dhillon[5,3]

February 17, 2021

Abstract

Motivated by modern applications, such as online advertisement and recommender systems, we study the top-k eXtreme contextual bandits problem, where the total number of arms can be enormous, and the learner is allowed to select k arms and observe all or some of the rewards for the chosen arms. We first propose an algorithm for the non-eXtreme realizable setting, utilizing the Inverse Gap Weighting strategy for selecting multiple arms. We show that our algorithm has a regret guarantee of $O(k\sqrt{(A-k+1)T\log(|\mathcal{F}|T)})$, where A is the total number of arms and $\mathcal{F}$ is the class containing the regression function, while only requiring $\tilde{O}(A)$ computation per time step. In the eXtreme setting, where the total number of arms can be in the millions, we propose a practically-motivated arm hierarchy model that induces a certain structure in mean rewards to ensure statistical and computational efficiency. The hierarchical structure allows for an exponential reduction in the number of relevant arms for each context, thus resulting in a regret guarantee of $O(k\sqrt{(\log A-k+1)T\log(|\mathcal{F}|T)})$. Finally, we implement our algorithm using a hierarchical linear function class and show superior performance with respect to well-known benchmarks on simulated bandit feedback experiments using eXtreme multi-label classification datasets. On a dataset with three million arms, our reduction scheme has an average inference time of only 7.9 milliseconds, which is a 100x improvement.

1 Introduction

The *contextual bandit* is a sequential decision-making problem, in which, at every time step, the learner observes a context, chooses one of the A possible actions (arms), and receives a reward for the chosen action. Over the past two decades, this problem has found a wide range of applications, from e-commerce and recommender systems (Yue and Guestrin, 2011; Li et al., 2016) to medical trials (Durand et al., 2018; Villar et al., 2015). The aim of the decision-maker is to minimize the difference in total expected reward collected when compared to an optimal policy, a quantity termed *regret*. As an example, consider an advertisement engine in an online shopping store, where the context can be the user's query, the arms can be the set of millions of sponsored products and the reward can be a click or a purchase. In such a scenario, one must balance between *exploitation* (choosing the best ad (arm) for a query (context) based on current knowledge) and *exploration* (choosing a currently unexplored ad for the context to enable future learning).

The contextual bandits literature can be broadly divided into two categories. The *agnostic* setting (Agarwal et al., 2014; Langford and Zhang, 2007; Beygelzimer et al., 2011; Rakhlin and Sridharan, 2016) is a model-free setting where one competes against the best policy (in terms of expected reward) in a class of policies. On the other hand, in the *realizable* setting it is assumed that a

[1] Google Research, Work done while at Amazon. [2] MIT. [3] Amazon. [4] Stanford University. [5] University of Texas at Austin.

known class $\mathcal{F}$ contains the function mapping contexts to expected rewards. Most of the algorithms in the realizable setting are based on Upper Confidence Bound or Thompson sampling (Filippi et al., 2010; Chu et al., 2011; Krause and Ong, 2011; Agrawal and Goyal, 2013) and require specific parametric assumptions on the function class. Recently there has been exciting progress on contextual bandits in the realizable case with general function classes. Foster and Rakhlin (2020) analyzed a simple algorithm for general function classes that reduced the adversarial contextual bandit problem to online regression, with a minimax optimal regret scaling. The algorithm was then analyzed for i.i.d. contexts using offline regression in (Simchi-Levi and Xu, 2020). The proposed algorithms are general and easily implementable but have two main shortcomings.

First, in many practical settings the task actually involves selecting a small number of arms per time instance rather than a single arm. For instance, in our advertisement example, the website can have multiple slots to display ads and one can observe the clicks received from some or from all the slots. It is not immediately obvious how the techniques in (Simchi-Levi and Xu, 2020; Foster and Rakhlin, 2020) can be extended to selecting k of a total of A arms while avoiding the combinatorial explosion from $\binom{A}{k}$ possibilities. Second, the total number of arms A can be in tens of millions and we need to develop algorithms that only require $o(A)$ computation per time-step and also have a much smaller dependence on the total number of arms in the regret bounds. Therefore, in this paper, we consider the top-k eXtreme contextual bandit problem where the number of arms is potentially enormous and at each time-step one is allowed to select $k \geq 1$ arms.

This extreme setting is both theoretically and practically challenging, due to the sheer size of the arm space. On the theoretical side, most of the existing results on contextual bandit problems address the small arm space case, where the complexity and regret typically scales polynomially (linearly or as square root) in terms of the number of arms (with the notable exception of the case when arms are embedded in a d-dimensional vector space (Foster et al., 2020a)). Such a scaling inevitably results in large complexity and regret in the extreme setting. On the implementation side, most contextual bandit algorithms have not been shown to scale to millions of arms. The goal of this paper is to bridge the gaps both in theory and in practice. We show that the freedom to present more than one arm per time step provides valuable exploration opportunities. Moreover in many applications, for a given context, the rewards from the arms that are correlated to each other but not directly related to the context are often quite similar, while large variations in the reward values are only observed for the arms that are closely related to the context. For instance in the advertisement example, for an electronics query (context) there might be finer variation in rewards among computer accessories related display ads while very little variation in rewards among items in an unrelated category like culinary books. This prior knowledge about the structure of the reward function can be modeled via a judicious choice of the model class $\mathcal{F}$, as we show in this paper.

The **main contributions** of this paper are as follows:

- We define the top-k contextual bandit problem in Section 3.1. We propose a natural modification of the inverse gap weighting (IGW) sampling strategy employed in (Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2020; Abe and Long, 1999) as Algorithm 1. In Section 3.3 we show that our algorithm can achieve a top-k regret bound of $O(k\sqrt{(A-k+1)T\log(|\mathcal{F}|T)})$ where T is the time-horizon. Even though the action space is combinatorial, our algorithm's computational cost for a time-step is $O(A)$ as it can leverage the additive structure in the total reward obtained from a set of arms chosen. We also prove that if the problem setting is only approximately realizable then

our algorithm can achieve a regret scaling of $O(k\sqrt{(A-k+1)T\log(|\mathcal{F}|T)} + \epsilon k\sqrt{A-k+1}T)$, where ϵ is a measure of the approximation.

- Inspired by success of tree-based approached for eXtreme output space problems in supervised learning (Prabhu et al., 2018; Yu et al., 2020; Khandagale et al., 2020), in Section 4 we introduce a hierarchical structure on the set of arms to tackle the eXtreme setting. This allows us to propose an eXtreme reduction framework that reduces an extreme contextual bandit problem with A arms (A can be in millions) to an equivalent problem with only $O(\log A)$ arms. Then we show that our regret guarantees from Section 3.3 carry over to this reduced problem.
- We implement our eXtreme contextual bandit algorithm with a hierarchical linear function class and test the performance of different exploration strategies under our framework on eXtreme multi-label datasets (Bhatia et al., 2016) in Section 5, under simulated bandit feedback (Bietti et al., 2018). On the amazon-3m dataset, with around three million arms, our reduction scheme leads to a 100x improvement in inference time over a naively evaluating the estimated reward for every arm given a context. We show that the eXtreme reduction also leads to a 29% improvement in progressive mean rewards collected on the eurlex-4k dataset. More over we show that our exploration scheme has the highest win percentage among the 6 datasets w.r.t the baselines.

2 Related Work

The relevant prior work can be broadly classified under the following three categories:

General Contextual Bandits: The general contextual bandit problem has been studied for more than two decades. In the agnostic setting where the mean reward of the arms given a context is not fully captured by the function class $\mathcal{F}$, the problem was studied in the adversarial setting leading to the well-known EXP-4 class of algorithms (Auer et al., 2002; McMahan and Streeter, 2009; Beygelzimer et al., 2011). These algorithms can achieve the optimal $\tilde{O}(\sqrt{AT\log(T|\mathcal{F}|)})$ regret bound but the computational cost per time-step can be $O(|\mathcal{F}|)$. This paved the way for oracle-based contextual bandit algorithms in the stochastic setting (Agarwal et al., 2014; Langford and Zhang, 2007). The algorithm in (Agarwal et al., 2014) can achieve optimal regret bounds while making only $\tilde{O}(\sqrt{AT})$ calls to a cost-sensitive classification oracle, however the algorithm and the oracle are not easy to implement in practice. In more recent work, it has been shown that algorithms that use regression oracles work better in practice (Foster et al., 2018). In this paper we will be focused on the realizable (or near-realizable) setting, where there exists a function in the function class, which can model the expected reward of arms given context. This setting has been studied with great practical success under specific instances of the function classes, such as linear. Most of the successful approaches are based on Upper Confidence Bound strategies or Thompson Sampling (Filippi et al., 2010; Chu et al., 2011; Krause and Ong, 2011; Agrawal and Goyal, 2013), both of which lead to algorithms which are heavily tailored to the specific function class. The general realizable case was modeled in (Agarwal et al., 2012) and recently there has been exciting progress in this direction. The authors in (Foster and Rakhlin, 2020) identified that a particular exploration scheme that dates back to (Abe and Long, 1999) can lead to a simple algorithm that reduces the contextual bandit problem to online regression and can achieve optimal regret guarantees. The same idea was extended for the stochastic realizable contextual bandit problem with an offline batch regression oracle (Simchi-Levi and Xu, 2020; Foster et al., 2020b). We build on the techniques introduced in these works. However all the literature discussed so far only address the problem of selecting one arm per time-step, while we are interested in selection

the top-k arms at each time step.

Exploration in Combinatorial Action Spaces: In (Qin et al., 2014) authors study the k-arm selection problem in contextual bandits where the function class is linear and the utility of a set of arms chosen is a set function with some monotonicity and Lipschitz continuity properties. In (Yue and Guestrin, 2011) the authors study the problem of retrieving k-arms in contextual bandits in the context of a linear function class and the assumption that the utility of a set of arms is sub-modular. Both these approaches do not extend to general function classes and are not applicable to the extreme setting. In the context of off-policy learning from logged data there are several works that address the top-k arms selection problem under the context of slate recommendations (Swaminathan et al., 2017; Narita et al., 2019). We will now review the combinatorial action space literature in multi-armed bandit (MAB) problems. Most of the work in this space deals with semi-bandit feedback (Chen et al., 2016; Combes et al., 2015; Kveton et al., 2015; Merlis and Mannor, 2019). This is also our feedback model, but we work in a contextual setting. There is also work in the full-bandit feedback setting, where one gets to observe only one representative reward for the whole set of arms chosen. This body of literature can be divided into the adversarial setting (Merlis and Mannor, 2019; Cesa-Bianchi and Lugosi, 2012) and the stochastic setting (Dani et al., 2008; Agarwal and Aggarwal, 2018; Lin et al., 2014; Rejwan and Mansour, 2020).

Learning in eXtreme Output Spaces: The problem of learning from logged bandit feedback when the number of arms is extreme was studied recently in (Lopez et al., 2020). In (Majzoubi et al., 2020) the authors address the contextual bandit problem for continuous action spaces by using a cost sensitive classification oracle for large number of classes, which is itself implemented as a hierarchical tree of binary classifiers. In the context of supervised learning the problem of learning under large but correlated output spaces has been studied under the banner of eXtreme Multi-Label Classification/Ranking (XMC/ XMR) (see (Bhatia et al., 2016) and references). Tree based methods for XMR have been extremely successful (Jasinska et al., 2016; Prabhu et al., 2018; Khandagale et al., 2020; Wydmuch et al., 2018; You et al., 2019; Yu et al., 2020). In particular our assumptions about arm hierarchy and the implementation of our algorithms have been motivated by (Prabhu et al., 2018; Yu et al., 2020).

3 Top-k Stochastic Contextual Bandit Under Realizability

In the standard contextual bandit problem, at each round, a context is revealed to the learner, the learner picks a single arm, and the reward for only that arm is revealed. In this section, we will study the top-k version of this problem, i.e. at each round the learner selects k distinct arms, and the total reward corresponds to the sum of the rewards for the subset. As feedback, the learner observes some of the rewards for actions in the chosen subset, and we allow this feedback to be as rich as the rewards for all the k selected arms or as scarce as no feedback at all on the given round.

3.1 The Top-k Problem

Suppose that at each time step $t \in \{1, \ldots, T\}$, the environment generates a context $x_t \in \mathcal{X}$ and rewards $\{r_t(a)\}_{a \in [A]}$ for the A arms. The set of arms will be denoted by $\mathcal{A} = [A] := \{1, 2, \cdots, A\}$. As standard in the stochastic model of contextual bandits, we shall assume that $(x_t, r_t(1), \cdots r_t(A))$ are generated i.i.d. from a fixed but unknown distribution $\mathcal{D}$ at each time step. In this work we will assume for simplicity that $r_t(a) \in [0, 1]$ almost surely for all t and $a \in [A]$. We will work under the realizability assumption (Agarwal et al., 2012; Foster et al., 2018; Foster and Rakhlin, 2020).

We also provide some results under approximate realizabilty or the misspecified setting similar to (Foster and Rakhlin, 2020).

Assumption 1 (Realizability). *There exists an $f^* \in \mathcal{F}$ such that,*

$$\mathbb{E}[r_t(a)|X = x] = f^*(x, a) \;\; \forall x \in \mathcal{X}, a \in [A], \tag{1}$$

where $\mathcal{F}$ is a class of functions $\mathcal{X} \times \mathcal{A} \to [0, 1]$ known to the decision-maker.

Assumption 2 (ϵ-Realizability). *There exists an $f^* \in \mathcal{F}$ such that,*

$$|\mathbb{E}[r_t(a)|X = x] - f^*(x, a)| \leq \epsilon \qquad \forall x \in \mathcal{X}, a \in [A]. \tag{2}$$

where $\mathcal{F}$ is a class of functions $\mathcal{X} \times \mathcal{A} \to [0, 1]$ known to the decision-maker.

We assume that the misspecification level ϵ is known to the learner and refer to (Foster et al., 2020a) for techniques on adapting to this parameter.

Feedback Model and Regret. At the beginning of the time step t, the learner observes the context x_t and then chooses a set of k *distinct* arms $\mathcal{A}_t \subseteq \mathcal{A}$, $|\mathcal{A}_t| = k$. The learner receives feedback for a subset $\Phi_t \subseteq \mathcal{A}_t$, that is, $r_t(a)$ is revealed to the learner for every $a \in \Phi_t$.

Assumption 3. *Conditionally on $x_t, \mathcal{A}_t$ and the history $\mathcal{H}_{t-1}$ up to time $t-1$, the set $\Phi_t \subseteq \mathcal{A}_t$ is random and for any $a \in \mathcal{A}_t$,*

$$\mathbb{P}(a \in \Phi_t | x_t, \mathcal{A}_t, \mathcal{H}_{t-1}) \geq c$$

for some $c \in (0, 1]$ which we assume to be known to the learner.

For the advertisement example, Assumption 3 means that the user providing feedback has at least some non-zero probability $c > 0$ of choosing each of the presented ads, marginally. The choice $c = 1$ corresponds to the most informative case – the learner receives feedback for all the k chosen arms. On the other hand, for $c < 1$ it may happen that no feedback is given on a particular round (for instance, if Φ_t includes each $a \in \mathcal{A}_t$ independently with probability c). When $\mathcal{A}_t$ is a ranked list, behavioral models postulate that the user clicks on an advertisement according to a certain distribution with decreasing probabilities; in this case, c would correspond to the smallest of these probabilities. A more refined analysis of regret bounds in terms of the distribution of Φ_t is beyond the scope of this work.

The total reward obtained in time step t is given by the sum $\sum_{a \in \mathcal{A}_t} r_t(a)$ of all the individual arm rewards in the chosen set, regardless of whether only some of these rewards are revealed to the learner. The performance of the learning algorithm will be measured in terms of *regret*, which is the difference in mean rewards obtained as compared to an optimal policy which always selects the top k distinct actions with the highest mean reward. To this end, let $\mathcal{A}_t^*$ be the set of k distinct actions that maximizes $\sum_{a \in \mathcal{A}_t^*} f(x_t, a)$ for the given x_t. Then the expected regret is

$$R(T) := \sum_{t=1}^{T} \mathbb{E}\left[\sum_{a \in \mathcal{A}_t^*} f^*(x_t, a) - \sum_{a \in \mathcal{A}_t} f^*(x_t, a) \right]. \tag{3}$$

Regression Oracle. As in (Foster et al., 2018; Simchi-Levi and Xu, 2020), we will rely on the availability of an optimization oracle regression-oracle for the class $\mathcal{F}$ that can perform least-squares regression,

$$\operatorname*{argmin}_{f\in\mathcal{F}} \sum_{s=1}^{t} (f(x_a, a_s) - r_s)^2 \tag{4}$$

where $(x, a, r) \in \mathcal{X} \times \mathcal{A} \times [0, 1]$ ranges over the collected data.

3.2 IGW for top-k Contextual Bandits

Our proposed algorithm for top-k arm selection in general contextual bandits in a non-extreme setting is provided as Algorithm 1. It is a natural extension of the Inverse Gap Weighting (IGW) sampling scheme (Abe and Long, 1999; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2020). In Section 3.3 we will show that this algorithm with $r = 1$ has good regret guarantees for the top-k problem even though the action space is combinatorial, thanks to the linearity of the regret objective in terms of rewards of individual arms in the subset. Note that a naive extension of IGW by treating each action in $\mathcal{A}^k$ as a separate arm would require a computation of $O\left(\binom{A}{k}\right)$ per time step and a similar regret scaling. In contrast, Algorithm 1 only requires $\tilde{O}(A)$ computation for the sampling per time step.

Algorithm 1 Top-k Contextual Bandits with IGW

Arguments: k and r (number of explore slots, $1 \le r \le k$)
for $l \leftarrow 1$ **to** $e(T)$ **do**
  Fit regression oracle to all past data
  $\widehat{y}_l = \operatorname{argmin}_{f\in\mathcal{F}} \sum_{t=1}^{N_{l-1}} \sum_{a\in\Phi_t} (f(x_t, a) - r_t(a))^2$
  for $s \leftarrow N_{l-1} + 1$ **to** $N_{l-1} + n_l$ **do**
    Receive x_s
    Let $\widehat{a}_s^1, \ldots, \widehat{a}_s^A$ be the arms ordered in decreasing order according to $\widehat{y}_l(x_s, \cdot)$ values.
    $\mathcal{A}_s = \{\widehat{a}_s^1, \cdots, \widehat{a}_s^{k-r}\}$.
    for $c \leftarrow 1$ **to** r **do**
      Compute randomization distribution
      $p = \mathsf{IGW}(\{\mathcal{A} \setminus \mathcal{A}_s\}; \widehat{y}_l(x_s, \cdot))$.
      Sample $a \sim p$. Let $\mathcal{A}_s = \mathcal{A}_s \cup \{a\}$.
    end for
    Obtain rewards $r_s(a)$ for actions $a \in \Phi_s \subseteq \mathcal{A}_s$.
  end for
  Let $N_l = N_{l-1} + n_l$
end for

The Inverse Gap Weighting strategy was introduced in (Abe and Long, 1999) and has since then been used for contextual bandits in the realizable setting with general function classes (Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2020; Foster et al., 2020b). Given a set of arms $\mathcal{A}$, an estimate $\widehat{y} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ of the reward function, and a context x, the distribution $p = \mathsf{IGW}(\mathcal{A}; \widehat{y}(x, \cdot))$ over arms is given by

$$p(a|x) = \begin{cases} \frac{1}{|\mathcal{A}| + \gamma_l(\widehat{y}(x,a_\star) - \widehat{y}(x,a))} & \text{if } a \neq a_\star \\ 1 - \sum_{a'\in\mathcal{A}: a'\neq a_\star} p(a'|x) & \text{otherwise} \end{cases}$$

where $a_\star = \operatorname{argmax}_{a\in\mathcal{A}} \widehat{y}(x, a)$, γ_l is a scaling factor.

Algorithm 1 proceeds in epochs, indexed by $l = 1, \ldots, e(T)$. Note that $N_{e(T)} = \sum_{l=1}^{e(T)} n_l = T$. The regression model is updated at the beginning of the epoch with all the past data and used throughout the epoch (n_l time steps). The arm selection procedure for the top-k problem involves selecting the top $(k - r)$ arms *greedily* according to the current estimate $\widehat{y}_l$ and then selecting the rest of the arms at random according to the Inverse Gap Weighted distribution over the set of remaining arms. For $r > 1$, the distribution is recomputed over the remaining support every time an arm is selected.

3.3 Regret of IGW for top-k Contextual Bandits

In this section we show that our algorithm has favorable regret guarantees. Our regret guarantees are only derived for the case when Algorithm 1 is run with $r = 1$. However, we will see that other values of r also work well in practice in Section 5. For ease of exposition we assume $\mathcal{F}$ is finite; our results can be extended to infinite function classes with standard techniques (see e.g. (Simchi-Levi and Xu, 2020)). We first present the bounds under exact realizability.[1]

Theorem 1. *Algorithm 1 under Assumptions 1 and 3, when run with parameters*

$$r = 1; \quad N_l = 2^l; \quad \gamma_l = \frac{1}{32}\sqrt{\frac{c(A-k+1)N_{l-1}}{162\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)}},$$

has regret bound

$$R(T) = \mathcal{O}\left(k\sqrt{c^{-1}(A-k+1)T\log\left(\frac{|\mathcal{F}|T}{\delta}\right)}\right)$$

with probability at least $1 - \delta$, for a finite function class $\mathcal{F}$.

In the next theorem we bound the regret under ϵ-realizability.

Theorem 2. *Algorithm 1 under Assumptions 2 and 3, when run with parameters*

$$r = 1; \quad N_l = 2^l; \quad \gamma_l = \frac{\sqrt{c(A-k+1)}}{32\sqrt{\frac{420}{N_{l-1}}\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right) + 2\epsilon^2}}$$

has regret bound

$$R(T) = \mathcal{O}\left(k\sqrt{c^{-1}(A-k+1)T\log\left(\frac{|\mathcal{F}|T}{\delta}\right)} + \epsilon kT\sqrt{A-k+1}\right)$$

with probability at least $1 - \delta$, for a finite function class $\mathcal{F}$.

The proofs for both of our main theorems are provided in Appendix A. One of the key ingredients in the proof is an induction hypothesis which helps us relate the top-k regret of a policy with respect to the estimated reward function $\widehat{y}_l \in \mathcal{F}$ at the beginning of epoch l to the actual regret with respect to $f^* \in \mathcal{F}$. The argument can be seen as a generalization of (Simchi-Levi and Xu, 2020) to $k > 1$.

[1] We have not optimized the constants in the definition of γ_l.

4 eXtreme Contextual Bandits and Arm Hierarchy

When the number of arms A is large, the goal is to design algorithms so that the computational cost per round is poly-logarithmic in A (i.e. $\mathcal{O}(\text{polylog}(A))$) and so it the overall regret. However, owing to known lower bounds (Foster and Rakhlin, 2020), this cannot be achieved without imposing further assumptions on the contextual bandit problem.

Main idea. A key observation is that the regression-oracle framework does not impose any restriction on the structure of the arms and in fact the set of arms can even be context-dependent. Let us assume that,

- For each x, there is an x-dependent decomposition

$$\mathcal{A}_x := \{\boldsymbol{a}_{x,1}, \cdots, \boldsymbol{a}_{x,Z}\}, \tag{5}$$

  where $\boldsymbol{a}_{x,1}, \cdots, \boldsymbol{a}_{x,Z}$ form a disjoint union of $\mathcal{A}$ with $Z = \mathcal{O}(\log A)$.
- For any two arms a and a' from any subset $\boldsymbol{a}_{x,i}$, the expected reward function $r(x,a) = \mathbb{E}[r(a)|X = x]$ satisfies the following consistency condition

$$|r(x,a) - r(x,a')| \leq \epsilon. \tag{6}$$

By treating $\boldsymbol{a}_{x,1}, \cdots, \boldsymbol{a}_{x,Z}$ as effective arms, the results of Section 3.3 can be applied by *working with functions that are piecewise constant over each* $\boldsymbol{a}_{x,i}$. Such a context-dependent arm space decomposition is a reasonable assumption, because often the rewards from a large subset of arms exhibit minor variations for a given context x.

Motivating example. To motivate and justify the conditions (5) and (6), consider a simple but representative setting where the contexts in $\mathcal{X}$ and arms in $\mathcal{A}$ are both represented as feature vectors in $\mathbb{R}^d$ for a fixed dimension d and the distance between two vectors is measured by the Euclidean norm $\|\cdot\|$. In many applications, the expected reward $r(x,a)$ satisfies the gradient condition $|\partial_a r(x,a)| \leq \frac{\eta}{\|x-a\|}$, for some $\eta > 0$, i.e., $r(x,a)$ is sensitive in a only when a is close to x and insensitive when a is far away from x.

Let us introduce a hierarchical decomposition $\mathcal{T}$ for $\mathcal{A}$, which in this case is a balanced 2^d-ary tree. At the leaf level, each tree node has a maximum number of m arms from the extreme arm space $\mathcal{A}$. The height of such a tree is $H \approx \lceil \log \lceil A/m \rceil \rceil$ under some mild assumptions on the distributions of the arms in $\mathcal{A}$. For a specific depth h, we use $e_{h,i}$ to denote a node with index i at depth h and $\mathcal{C}_{h,i}$ to denote the 2^d children of $e_{h,i}$ at depth $h+1$. Each node $e_{h,i}$ of the tree is further equipped with a *routing function* $g_{h,i}(x) = \frac{\text{rad}_{h,i}}{\|x - \text{ctr}_{h,i}\|}$, where $\text{ctr}_{h,i}$ is the center of the node $e_{h,i}$ and $\text{rad}_{h,i}$ is the radius of the smallest ball at $\text{ctr}_{h,i}$ that contains $e_{h,i}$. The center $\text{ctr}_{h,i}$ serves as a representative for the set of arms in $e_{h,i}$. Figure 1 (left) illustrates the hierarchical decomposition for the 1D case.

Given a context x, we perform an *adaptive search* through this hierarchical decomposition $\mathcal{T}$, parameterized by a constant $\beta \in (0,1)$. Initially, the sets I_x and S_x are set to be empty and the search starts from the root of the tree. When a node $e_{h,i}$ is visited, it is considered *far from* x if $g_{h,i}(x) = \frac{\text{rad}_{h,i}}{\|x - \text{ctr}_{h,i}\|} \leq \beta$ and *close to* x if $g_{h,i}(x) = \frac{\text{rad}_{h,i}}{\|x - \text{ctr}_{h,i}\|} > \beta$. If $e_{h,i}$ is far from x, we simply place it in I_x. If $e_{h,i}$ close to x, we visit its children in $\mathcal{C}_{h,i}$ recursively if $e_{h,i}$ is an internal node or place it in S_x if it is a leaf. At the end of the search, I_x consists of a list of internal nodes and S_x is a list of *singleton* arms.

We claim that the union of the singleton arms in S_x and the nodes in I_x form an x-dependent decomposition $\mathcal{A}_x$. First, the disjoint union of I_x and S_x covers the whole arm space $\mathcal{A}$. S_x contains only $O(1)$ singleton arms with arm features close to the context feature x while the size of I_x is bounded by $O(\log A)$ as there are at most $O(1)$ nodes $e_{h,i}$ inserted into I_x at each of the $O(\log A)$ levels. Hence, the sum of the cardinalities of S_x and I_x is bounded by $Z = O(\log A)$, i.e., logarithmic in the size A of the extreme arm space $\mathcal{A}$.

Second, for any two original arms a_1, a_2 corresponding to a node $e_{h,i} \in I_x$,

$$|r(x, a_1) - r(x, a_2)| \leq \|\partial_l r(x, a')\| \cdot \|a_1 - a_2\| \leq \frac{\eta}{\|x - a'\|} \cdot (2 \text{ rad}_{h,i}),$$

where a' lies on the segment between a_1 and a_2. Since

$$\|x - a'\| \geq \|x - \text{ctr}_{h,i}\| - \|a' - \text{ctr}_{h,i}\| \geq (1/\beta - 1)\text{rad}_{h,i}$$

holds for $e_{h,i} \in I_x$,

$$|r(x, a_1) - r(x, a_2)| \leq \frac{\eta}{(1/\beta - 1)\text{rad}_{h,i}} \cdot (2 \text{ rad}_{h,i}) = \frac{2\eta\beta}{1 - \beta}.$$

Hence, if one chooses β so that $2\eta\beta/(1 - \beta) \leq \epsilon$, then $|r(x, a_1) - r(x, a_2)| \leq \epsilon$ for any two arms a_1, a_2 in any $e_{h,i} \in I_x$.

Therefore for each x, the union of the singleton arms in S_x and the nodes in I_x form an x-dependent decomposition of $\mathcal{A}$ that satisfies the conditions (5) and (6). Figure 1 (middle) shows the decomposition for a given context x, while Figure 1 (right) shows how the decomposition varies with the context x. In what follows, we shall refer to the members of I_x *node effective arms* and the ones of S_x *singleton effective arms*.

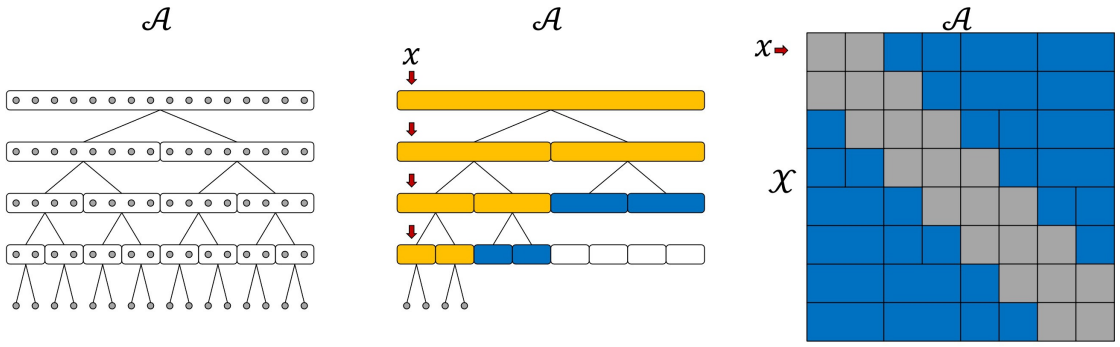


Figure 1: **Left**: an illustration of the hierarchical decomposition for $\mathcal{A}$, where each gray dot indicates an arm. **Middle**: the adaptive search for a given context x. The yellow nodes are further explored as they are close to x while the blue nodes are not as they are far from x. The set of effective arms for x consists of the blue nodes and the singleton arms in the yellow leaf nodes. **Right**: For a fixed x, the corresponding row shows the x-dependent hierarchical arm space decomposition. As x varies, the decomposition also changes. Each blue block stands for a non-singleton effective arm, valid for a contiguous block of contexts. Each gray block contains the singleton effective arms, valid again for a contiguous block of contexts.

General setting. Based on the motivating example, we propose an arm hierarchy for general $\mathcal{X}$ and $\mathcal{A}$. We assume access to a hierarchical partitioning $\mathcal{T}$ of $\mathcal{A}$ that breaks progressively into finer

subgroups of similar arms. The partitioning can be represented by a balanced tree that is p-ary till the leaf level. At the leaf level, each node can have a maximum of $m > p$ children, each of which is a singleton arm in $\mathcal{A}$. The height of such a tree is $H = \left\lceil \log_p \lceil A/m \rceil \right\rceil$. With a slight abuse of notation, we use $e_{h,i}$ to denote a node in the tree as well as the subset of singleton arms in the subtree of the node.

Each internal node $e_{h,i}$ is assumed to be associated with a routing function $g_{h,i}(x)$ mapping $\mathcal{X} \to [0,1]$ and $\mathcal{C}_{h,i}$ is used to denote the immediate children of node $e_{h,i}$. Based on these routing functions and an integer parameter b, we define a beam search in Algorithm 2 for any context $x \in \mathcal{X}$ as an input. During its execution, this beam search keeps at each level h only the top b nodes that return the highest $g_{h,i}(x)$ values. The output of the beam search, denoted also by $\mathcal{A}_x$, is the union of a set of nodes denoted as I_x and a set of singleton arms denoted as S_x. The tree structure ensures that there are at most bm singleton arms in S_x and at most $(p-1)b(H-1)$ nodes in I_x. Therefore, $|\mathcal{A}_x| \leq (p-1)b(H-1) + bm = O(\log A)$, implying that $\mathcal{A}_x$ satisfies (5). Though the cardinality $|\mathcal{A}_x|$ can vary slightly depending on the context x, in what follows we make the simplifying assumption that $|\mathcal{A}_x|$ is equal to a constant $Z = O(\log A)$ independent of x and denote $\mathcal{A}_x = \{\boldsymbol{a}_{x,1}, \ldots, \boldsymbol{a}_{x,Z}\}$.

Algorithm 2 Beam search

Arguments: beam-size b, $\mathcal{T}$, routing functions $\{g\}$, x
Initialize $\texttt{codes} = [(1,1)]$ and $I_x^b = \emptyset$.
for $h = 1, \cdots, H-1$ **do**
  Let $\texttt{labels} = \cup_{(h-1,i) \in \texttt{codes}} \mathcal{C}_{h-1,i}$.
  Let $\texttt{codes}$ be top-b nodes in $\texttt{labels}$ according to the values $g_{h,i}(x)$.
  Add the nodes in $\texttt{labels} \setminus \texttt{codes}$ to I_x.
end for
Let $S_x = \cup_{(H-1,i) \in \texttt{codes}} \mathcal{C}_{H-1,i}$.
Return $\mathcal{A}_x = S_x \cup I_x$.

To ensure the consistency condition (6) in the general case, one requires the expected reward function $r(x,a)$ to be nearly constant over each effective arm $\boldsymbol{a}_{x,i}$ and work with a function class that is constant over each $\boldsymbol{a}_{x,i}$. The following definition formalizes this.

Definition 1. *Given a hierarchy $\mathcal{T}$ with routing function family $\{g_{h,i}(\cdot)\}$ and a beam-width b, a function $f(x,a)$ is $(\mathcal{T}, g, b)$-constant if for every $x \in \mathcal{X}$*

$$f(x,a) = f(x,a') \quad \textit{for all} \quad a, a' \in e_{h,i},$$

for any node $e_{h,i}$ in $I_x \subset \mathcal{A}_x$. A class of functions $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant if each $f \in \mathcal{F}$ is $(\mathcal{T}, g, b)$-constant.

Figure 2 (left) provides an illustration of a $(\mathcal{T}, g, b)$-constant predictor function for the simple case $\mathcal{X} \subset [0,1]$ and $\mathcal{A} \subset [0,1]$. In the eXtreme setting, we always assume that our function class $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant. By further assuming that the expected reward $r(x,a)$ satisfies either Assumption 1 or Assumption 2, Condition (6) is satisfied.

4.1 IGW for top-k eXtreme Contextual Bandits

In this section we provide our algorithm for the eXtreme setting. As Definition 1 reduces the eXtreme problem with A arms to a non-extreme problem with only $Z = \mathcal{O}(\log A)$ effective arms,

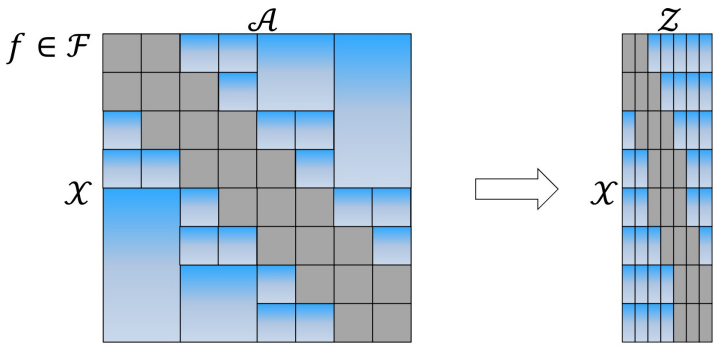


Figure 2: **Left**: A $(\mathcal{T}, g, b)$-constant predictor function $f(x, a)$ in the 1D motivating example with $\mathcal{X} \subset [0, 1]$ and $\mathcal{A} \subset [0, 1]$. Within each blue block, $f(x, a)$ is constant in a but varies with x. **Right**: the function $\tilde{f}$ after the reduction.

Algorithm 3 essentially uses the beam-search method in Algorithm 2 to construct this reduced problem. The IGW randomization is performed over the effective arms and if a non-singleton arm (i.e., an internal node of $\mathcal{T}$) is chosen, we substitute it with a randomly chosen singleton arm that lies in the sub-tree of that node. More specifically, for a $(\mathcal{T}, g, b)$-constant class $\mathcal{F}$, we define for each $f \in \mathcal{F}$ a new function $\tilde{f} : \mathcal{X} \times [Z] \to [0, 1]$ s.t. for any $z = 1, \ldots, Z$ we have $\tilde{f}(x, z) = f(x, a)$ for some fixed $a \in \boldsymbol{a}_{x,z}$. Here, we assume that for any x the beam-search process in Algorithm 2 returns the effective arms in $\mathcal{A}_x$ in a fixed order and $\boldsymbol{a}_{x,z}$ is the z-th arm in this order. The collection of these new functions over the context set $\mathcal{X}$ and the reduced arm space $\mathcal{Z} = [Z]$ is denoted by $\tilde{\mathcal{F}} = \{\tilde{f} : f \in \mathcal{F}\}$. Figure 2 (right) provides an illustration of a function $\tilde{f}(x, z)$ obtained after the reduction.

As a practical example, we can maintain the function class $\mathcal{F}$ such that each member $f \in \mathcal{F}$ is represented as a set of regressors at the internal nodes as well as the singleton arms in the tree. These regressors map contexts to $[0, 1]$. For an $f \in \mathcal{F}$, the regressor at each node is constant over the arms a within this node and is only trained on past samples for which that node was selected as a whole in $\mathcal{Z}_s$ in Algorithm 3; the regressor at a singleton arm can be trained on all samples obtained by choosing that arm. Note that even though we might have to maintain a lot of regression functions, many of them can be sparse if the input contexts are sparse, because they are only trained on a small fraction of past training samples.

4.2 Top-k Analysis in the eXtreme Setting

We can analyze Algorithm 3 under the realizability assumptions (Assumption 1 or Assumption 2) when the class of functions satisfies Definition 1). Our main result is a reduction style argument that provides the following corollary of Theorems 1 and 2.

Corollary 1. *Algorithm 3 when run with parameter $r = 1$ has the following regret guarantees:*

(i) If Assumptions 1 and 3 hold and the function class $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant (Definition 1), then setting parameters as in Theorem 1 ensures that the regret bound stated in Theorem 1 holds with A replaced by $O(\log A)$.

(ii) If Assumptions 2 and 3 hold and the function class $\mathcal{F}$ is $(\mathcal{T}, g, b)$-constant (Definition 1), then setting parameters as in Theorem 2 ensures that the regret bound stated in Theorem 2 holds with A replaced by $O(\log A)$.

Algorithm 3 eXtreme Top-k Contextual Bandits with IGW

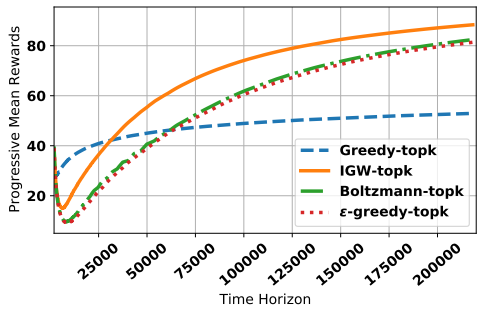
Arguments: k, number of explore slots: $1 \leq r \leq k$
for $l \leftarrow 1$ **to** $e(T)$ **do**
  Fit regression oracle to all past data
  $\widehat{y}_l = \operatorname{argmin}_{\tilde{f} \in \tilde{\mathcal{F}}} \sum_{t=1}^{N_{l-1}} \sum_{z \in \Phi_t} (\tilde{f}(x_t, z) - \tilde{r}_t(z))^2$
  for $s \leftarrow N_{l-1} + 1$ **to** $N_{l-1} + n_l$ **do**
    Receive x_s
    Use Algorithm 2 to get $\mathcal{A}_{x_s} = \{\boldsymbol{a}_{x_s,1}, \ldots, \boldsymbol{a}_{x_s,Z}\}$.
    Let $z_1, \ldots, z_Z$ be the arms in $[Z]$ in the descending order according to $\widehat{y}_l$.
    $\mathcal{Z}_s = \{z_1, \cdots, z_{k-r}\}$.
    for $c \leftarrow 1$ **to** r **do**
      Compute randomization distribution
      $p = \mathsf{IGW}([Z] \setminus \mathcal{Z}_s; \hat{y}_l(x_s, \cdot))$.
      Sample $z \sim p$. Let $\mathcal{Z}_s = \mathcal{Z}_s \cup \{z\}$.
    end for
    $B_s = \{\}$.
    for z in $\mathcal{Z}_s$ **do**
      If $\boldsymbol{a}_{x_s,z}$ is singleton arm, then add it to B_s.
      Otherwise sample a singleton arm a in the subtree rooted at the node $\boldsymbol{a}_{x_s,z}$ and add a to B_s.
    end for
    Choose the arms in B_s.
    Map the rewards back to the corresponding effective arms in $\mathcal{Z}_s$ and record $\{\tilde{r}_s(z), z \in \Phi_s\}$.
  end for
  Let $N_l = N_{l-1} + n_l$
end for

5 Empirical Results

We compare our algorithm with well known baselines on various real world datasets. We first perform a semi-synthetic experiment in a realizable setting. Then we use eXtreme Multi-Label Classification (XMC) (Bhatia et al., 2016) datasets to test our reduction scheme. The different exploration sampling strategies used in our experiments are [2]: **Greedy-topk:** The top-k effective arms for each context are chosen greedily according to the regression score; **Boltzmann-topk:** The top-$(k-r)$ arms are selected greedily. Then the next r arms are selected one by one, each time recomputing the Boltzmann distribution over the remaining arms. Under this sampling scheme the probability of sampling arm $\tilde{a}$ is proportional to $\exp(\log(N_{l-1})\beta\tilde{f}(x, \tilde{a}))$ (Cesa-Bianchi et al., 2017); **ϵ-greedy-topk:** Same as above but the last r arms are selected one by one using a scheme where the probability of sampling arm $\tilde{a}$ is proportional to $(1-\epsilon)+\epsilon/A'$ if $\tilde{a}$ is the arm with the highest score, otherwise the probability is ϵ/A' where A' is the number of arms remaining; **IGW-topk:** This is essentially the sampling strategy in Algorithm 1. We set $\gamma_l = \sqrt{CN_{l-1}A'}$ for the l-th epoch where A' is the number of remaining arms.

Realizable Experiment. In order to create a realizable setting that is realistic, we choose the eurlex-4k XMC dataset (Bhatia et al., 2016) in Table 1 and for each arm/label $a \in A$, we fit linear regressor weights ν_a^* that minimizes $\mathbb{E}_x[([x; 1.0]^T \nu_a^* - \mathbb{E}[r_a(t)|x])^2]$ over the dataset. Then we consider a derived system where $\mathbb{E}[r_a(t)|x] = [x; 1.0]^T \nu_a^*$ for all x, a that is the learnt weights

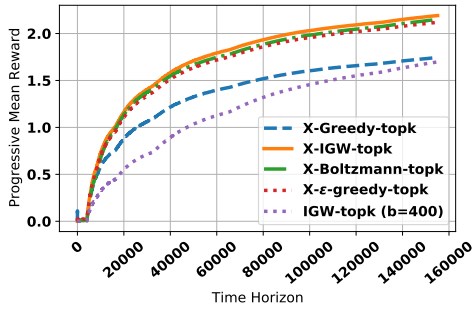
[2] Note that all these exploration strategies have been extended to the top-k setting using the ideas in Algorithm 1 and many popular contextual bandit algorithms like the ones in (Bietti et al., 2018) cannot be easily extended to the top-k setting.



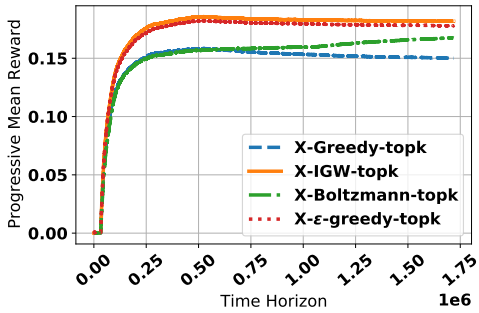
(a) Realizable eurlex-4k

Beam Size (b)	Inference Time (ms)
10	7.85
30	12.84
100	27.83
2.9K (all arms)	799.06

(b) Inference Time per context on amazon-3m



(c) eurlex-4k



(d) amazon-3m

Figure 3: In (a) we compare the different sampling strategies on a realizable setting with $k = 50$ and $r = 25$, derived from the eurlex-4k dataset. In (b) we compare the avg. inference times per context vs different beam sizes on the amazon-3m dataset. Note that for this dataset $b = 290,000$ will include all arms in the beam in our setting and is order wise equivalent to no hierarchy. This comparison is done for inference in a setting with $k = 5, r = 3$. Note that for larger datasets in Table 1 our implementation with $b = 10, 30$ remains efficient for real-time inference as the time-complexity scales only with the beams-size and the height of the tree. We plot the progressive mean rewards collected by each algorithm as a function of time in two of our 6 datasets in (c)-(d) where the algorithms are implemented under our eXtreme reduction framework. In our experiments in (c)-(d) we have $k = 5$ and $r = 3$. The beam size is 10 except for IGW-topk (b=400) in (c), which serves as a proxy for Algorithm 1 without the extreme reduction, as $b = 400$ includes all the arms in this dataset.

from before exactly represent the mean rewards of the arms. This system is then realizable for Algorithm 1 when the function $\mathcal{F}$ is linear. Figure 3(a) shows the progressive mean reward (sum of rewards till time t divided by t) for all the sampling strategies compared. We see that the IGW sampling strategy in Algorithm 1 outperforms all the others by a large margin. For more details please refer to Appendix F. Note that the hyper-parameters of all the algorithms are tuned on this dataset in order to demonstrate that even with tuned hyper-parameter choices IGW is the optimal scheme for this realizable experiment. The experiment is done with $k = 50, r = 25$ and $b = 10$.

eXtreme Experiments. We now present our empirical results on eXtreme multi-label datasets. Our experiments are performed under simulated bandit feedback using real-world eXtreme multi-label classification datasets (Bhatia et al., 2016). This experiment startegy is widely used in the literature (Agarwal et al., 2014; Bietti et al., 2018) with non-eXtreme multi-class datasets (see Appendix F for more details). Our implementation uses a hierarchical linear function class inspired by (Yu et al., 2020). The hyper-parameters in all the algorithms are tuned on the eurlex-4k datasets and then held fixed. This is in line with (Bietti et al., 2018), where the parameters are tuned on a set of datasets and then held fixed.

The tree and routing functions are formed using a small held out section of the datasets, whose sizes are specified in Table 1 (Initialization Size). In the interest of space we refer the readers to Appendix F for more implementation details.

We use 6 XMC datasets for our experiments. Table 1 provides some basic properties of each dataset. We can see that the number of arms in the largest dataset is as large as 2.8MM. The column Initialization Size denotes the size of the held out set used to intialize our algorithms. Note that for the datasets eurlex-4k and wiki10-31k we bootstrap the original training dataset to a larger size by sampling with replacement, as the original number of samples are too small to show noticeable effects.

Dataset	Initialization Size	Time-Horizon	No. of Arms	Max. Leaf Size (m)
eurlex-4k	5000	154490	4271	10
amazoncat-13k	5000	1186239	13330	10
wiki10-31k	5000	141460	30938	10
wiki-500k	20000	1779881	501070	100
amazon-670k	20000	490449	670091	100
amazon-3m	50000	1717899	2812281	100

Table 1: Properties of eXtreme Datasets

	X-Greedy	X-IGW-topk	X-Boltzmann-topk	X-ϵ-greedy-topk
X-Greedy	-	0W/0D/6L	1W/0D/5L	0W/1D/5L
X-IGW-topk	6W/0D/0L	-	4W/1D/1L	6W/0D/0L
X-Boltzmann-topk	5W/0D/1L	1W/1D/4L	-	3W/0D/3L
X-ϵ-greedy-topk	5W/1D/0L	0W/0D/6L	3W/0D/3L	-

Table 2: Win/Draw/Loss statistics among algorithms for the 6 datasets. When the difference in results between two algorithms is not significant according to the statistical significance formula in (Bietti et al., 2018) then it is deemed to be a draw.

We plot the progressive mean rewards (total rewards collected till time t divided by t) for all the algorithms in Figure 3 (c)-(d) for two datasets. The rest of the plots are included in Figure 4

in Appendix E. The algorithm names are prepended with an X to denote that the sampling is performed under the reduction framework of Algorithm 3. In our experiments the number of arms allowed to be chosen each time is $k = 5$. In Algorithm 3 we set the number of explore slots $r = 3$ and beam-size $b = 10$ (unless otherwise specified). We see that all the exploratory algorithms do much better than the greedy version i.e our eXtreme reduction framework works for structured exploration when the number of arms are in thousands or millions. The efficacy of the reduction framework is further demonstrated by X-IGW-topk(b=10) being better than IGW-topk (b=400) by 29% in terms of the mean reward, in Figure 3(c). Note that here IGW-topk(b=400) serves as a proxy for Algorithm 1 directly applied without the hierarchy, as the beam includes all the arms. The IGW scheme is always among the top 2 strategies in all datasets. It is the only strategy among the baselines that has optimal theoretical performance and this shows that the algorithm is practical. Table 2 provides Win(W)/Draw(D)/Loss(L) for each algorithm against the others. We use the same W/D/L scheme as in (Bietti et al., 2018) to create this table. Note that X-IGW-topk has the highest win percentage overall. In Figure 3(b) we compare the inference times for IGW of our hierarchical linear implementation for different beam-sizes on amazon-3m. Note that $b = 2.9K$ will include all arms in this dataset and is similar to a flat hierarchy. This shows that our algorithm will remain practical for real time inference on large datasets when $b \leq 30$ is used.

6 Discussion

We provide regret guarantees for the top-k arm selection problem in realizable contextual bandits with general function classes. The algorithm can be theoretically and practically extended to extreme number of arms under our proposed reduction framework which models a practically motivated arm hierarchy. We benchmark our algorithms on XMC datasets under simulated bandit feedback. There are interesting directions for future work, for instance extending the analysis to a setting where the reward derived from the k arms is a set function with interesting structures such as sub-modularity. It would also be interesting to analyze the eXtreme setting where the routing functions and hierarchy can be updated in a data driven manner after every few epochs.

References

Naoki Abe and Philip M Long. Associative reinforcement learning using linear probabilistic concepts. In *ICML*, pages 3–11. Citeseer, 1999.

Alekh Agarwal, Miroslav Dudík, Satyen Kale, John Langford, and Robert Schapire. Contextual bandit learning with predictable rewards. In *Artificial Intelligence and Statistics*, pages 19–26, 2012.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

Mridul Agarwal and Vaneet Aggarwal. Regret bounds for stochastic combinatorial multi-armed bandits with linear space complexity. *arXiv preprint arXiv:1811.11925*, 2018.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Peter Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory - COLT 2008*, pages 335–342. Omnipress, 2008.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26, 2011.

K. Bhatia, K. Dahiya, H. Jain, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016. URL http://manikvarma.org/downloads/XC/XMLRepository.html.

Alberto Bietti, Alekh Agarwal, and John Langford. A contextual bandit bake-off. *arXiv preprint arXiv:1802.04064*, 2018.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu. Boltzmann exploration done right. In *Advances in neural information processing systems*, pages 6284–6293, 2017.

Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, pages 1659–1667, 2016.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. *Advances in neural information processing systems*, 28:2116–2124, 2015.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, 2001.

Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine Learning for Healthcare Conference*, pages 67–82, 2018.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874, 2008.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

Dylan J Foster and Alexander Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. *arXiv preprint arXiv:2002.04926*, 2020.

Dylan J Foster, Alekh Agarwal, Miroslav Dudík, Haipeng Luo, and Robert E Schapire. Practical contextual bandits with regression oracles. *arXiv preprint arXiv:1803.01088*, 2018.

Dylan J Foster, Claudio Gentile, Mehryar Mohri, and Julian Zimmert. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33, 2020a.

Dylan J Foster, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *arXiv preprint arXiv:2010.03104*, 2020b.

Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hullermeier. Extreme f-measure maximization using sparse probability estimates. In *International Conference on Machine Learning*, pages 1435–1444, 2016.

Sujay Khandagale, Han Xiao, and Rohit Babbar. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning*, pages 1–21, 2020.

Andreas Krause and Cheng Ong. Contextual gaussian process bandit optimization. *Advances in neural information processing systems*, 24:2447–2455, 2011.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 817–824. Citeseer, 2007.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548, 2016.

Tian Lin, Bruno Abrahao, Robert Kleinberg, John Lui, and Wei Chen. Combinatorial partial monitoring game with linear feedback and its applications. In *International Conference on Machine Learning*, pages 901–909, 2014.

Romain Lopez, Inderjit Dhillon, and Michael I Jordan. Learning from extreme bandit feedback. *arXiv preprint arXiv:2009.12947*, 2020.

Maryam Majzoubi, Chicheng Zhang, Rajan Chari, Akshay Krishnamurthy, John Langford, and Aleksandrs Slivkins. Efficient contextual bandits with continuous actions. *arXiv preprint arXiv:2006.06040*, 2020.

H Brendan McMahan and Matthew Streeter. Tighter bounds for multi-armed bandits with expert advice. 2009.

Nadav Merlis and Shie Mannor. Batch-size independent regret bounds for the combinatorial multi-armed bandit problem. *arXiv preprint arXiv:1905.03125*, 2019.

Yusuke Narita, Shota Yasui, and Kohei Yata. Efficient counterfactual learning from bandit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4634–4641, 2019.

Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, pages 993–1002, 2018.

Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 461–469. SIAM, 2014.

Alexander Rakhlin and Karthik Sridharan. Bistro: An efficient relaxation-based method for contextual bandits. In *ICML*, pages 1977–1985, 2016.

Idan Rejwan and Yishay Mansour. Top-k combinatorial bandits with full-bandit feedback. In *Algorithmic Learning Theory*, pages 752–776. PMLR, 2020.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Available at SSRN*, 2020.

Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, pages 3632–3642, 2017.

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczynski. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 6355–6366, 2018.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, pages 5820–5830, 2019.

Hsiang-Fu Yu, Kai Zhong, and Inderjit S Dhillon. Pecos: Prediction for enormous and correlated output spaces. *arXiv preprint arXiv:2010.05878*, 2020.

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pages 2483–2491, 2011.

A Top-k Analysis

Notation: Let l denote epoch index with n_l time steps. Define $N_l = \sum_{i=1}^{l} n_i$. At the beginning of each epoch l, we compute $\widehat{y}_l(x,a)$ as regression with respect to past data,

$$\widehat{y}_l = \operatorname*{argmin}_{f\in\mathcal{F}} \sum_{t=1}^{N_{l-1}} \sum_{a\in\Phi_t} (f(x_t,a) - r_t(a))^2,$$

where Φ_t is the subset for which the learner receives feedback.

Let $\{\phi_l\}_{l\geq 2}$ be a sequence of numbers. The analysis in this section will be carried out under the event

$$\mathcal{E} = \left\{ l \geq 2 : \frac{2}{N_{l-1}} \sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s,\mathcal{A}_s} \left\{ \frac{1}{k} \sum_{a\in\mathcal{A}_s} (\widehat{y}_l(x_s,a) - f^*(x_s,a))^2 | \mathcal{H}_{s-1} \right\} \leq \phi_l^2 \right\} \tag{7}$$

Lemmas 5 and 7 compute ϕ_l for finite class $\mathcal{F}$, such that event $\mathcal{E}$ holds with high probability.

We define $\gamma_l = \sqrt{A-k+1}/(32\phi_l)$, the scaling parameter used by Algorithm 1. In this paper, we analyze Algorithm 1 with $r = 1$, i.e. our procedure deterministically selects top $k-1$ actions of $\widehat{y}_l$ and selects the remaining action according to Inverse Gap Weighting on the remaining coordinates.

A deterministic strategy α is a map $\alpha : \mathcal{X} \to \mathcal{A}$. Throughout the proofs, we employ the following shorthand to simplify the presentation. We shall write $\widehat{y}_i(x,\alpha)$ and $f^*(x,\alpha)$ in place of $\widehat{y}_i(x,\alpha(x))$ and $f^*(x,\alpha(x))$. We reserve the letter α for a strategy and a for an action.

Given x, we let $\widehat{\alpha}_l^j(x)$ be the j-th highest action according to $\widehat{y}_l(x,\cdot)$. Similarly, $\alpha^{*,j}(x)$ is the j-th highest action according to $f^*(x,\cdot)$. We say that the set of strategies $\alpha^1,\ldots,\alpha^k$ is non-overlapping if for any x the set $\{\alpha^1(x),\ldots,\alpha^k(x)\}$ is a set of distinct actions. Let $e(s)$ denote the epoch corresponding to time step s.

Our argument is based on the beautiful observation of (Simchi-Levi and Xu, 2020) that one can analyze IGW inductively, by controlling the differences between estimated gaps (to the best estimated action) and the true gaps (to the best true action in the given context), with a *mismatched factor of* 2. We extend this technique to top-k selection, which introduces a number of additional difficulties in the analysis.

Induction hypothesis (l): For any epoch $i < l$, and all non-overlapping strategies $\alpha^1,\ldots,\alpha^k \in \mathcal{A}^{\mathcal{X}}$,

$$\mathbb{E}_x \left\{ \sum_{j=1}^{k} [\widehat{y}_i(x,\widehat{\alpha}_i^j) - \widehat{y}_i(x,\alpha^j)] - 2\sum_{j=1}^{k} [f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j)] \right\} \leq \frac{k(A-k+1)}{\gamma_i}$$

and

$$\mathbb{E}_x \left\{ \sum_{j=1}^{k} [f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j)] - 2\sum_{j=1}^{k} [\widehat{y}_i(x,\widehat{\alpha}_i^j) - \widehat{y}_i(x,\alpha^j)] \right\} \leq \frac{k(A-k+1)}{\gamma_i}.$$

Lemma 1. *Suppose event (7) holds. For all non-overlapping strategies $\alpha^1,\ldots,\alpha^k$,*

$$\mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} |\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j)| \leq \phi_l \cdot \left((A-k+1) + \sum_{i=1}^{l-1} \frac{n_i}{N_{l-1}} \gamma_i \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \left[\widehat{y}_i(x,\widehat{\alpha}_i^j) - \widehat{y}_i(x,\alpha^j) \right] \right)^{1/2}$$

Hence, by the induction hypothesis (l)*,*

$$\mathbb{E}_x \frac{1}{k}\sum_{j=1}^{k} |\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j)| \le \sqrt{2}\phi_l \cdot \left((A-k+1) + \gamma_l \mathbb{E}_x \frac{1}{k}\sum_{j=1}^{k}\Big[f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j)\Big]\right)^{1/2}$$

assuming γ_i *are non-decreasing.*

Proof. Given x, let $T_x(\widehat{y}_i) \subset [A]$ denote the indices of top $k-1$ actions according to $\widehat{y}_i(x,\cdot)$. Let $p_i(\cdot|x)$ denote the IGW distribution on epoch i, with support on the remaining $A-k+1$ actions. On round s in epoch $e(s)$, given x_s, Algorithm 1 with $r=1$ chooses $\mathcal{A}_s$ by selecting $T_{x_s}(\widehat{y}_{e(s)})$ determistically and selecting the last action according to $p_{e(s)}(\cdot|x_s)$. We write $p_{e(s)}(\alpha|x_s)$ as a shorthand for $p_{e(s)}(\alpha(x_s)|x_s)$.

For non-overlapping strategies $\alpha^1,\dots,\alpha^k$,

$$\mathbb{E}_x \frac{1}{k}\sum_{j=1}^{k} |\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j)| = \frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s}\left\{ \frac{1}{k}\sum_{j=1}^{k}\Big|\widehat{y}_l(x_s,\alpha^j) - f^*(x_s,\alpha^j)\Big|\right\}.$$

This sum can be written as

$$\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s}\left[\frac{1}{k}\sum_{j=1}^{k}\Big|\widehat{y}_l(x_s,\alpha^j) - f^*(x_s,\alpha^j)\Big| \cdot \mathbf{1}\Big\{\alpha^j(x_s) \in T_{x_s}(\widehat{y}_{e(s)})\Big\}\right.$$
$$\left. + \frac{1}{k}\sum_{j=1}^{k} |\widehat{y}_l(x_s,\alpha^j) - f^*(x_s,\alpha^j)| \sqrt{p_{e(s)}(\alpha^j|x_s)} \frac{1}{\sqrt{p_{e(s)}(\alpha^j|x_s)}} \cdot \mathbf{1}\Big\{\alpha^j(x_s) \notin T_{x_s}(\widehat{y}_{e(s)})\Big\}\right].$$

By the Cauchy-Schwartz inequality, the last expression is upper-bounded by

$$\left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s}\frac{1}{k}\sum_{j=1}^{k} |f^*(x_s,\alpha^j) - \widehat{y}_l(x_s,\alpha^j)|^2 \mathbf{1}\Big\{\alpha^j(x_s) \in T_{x_s}(\widehat{y}_{e(s)})\Big\}\right)^{1/2}$$
$$+ \left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s}\frac{1}{k}\sum_{j=1}^{k} |f^*(x_s,\alpha^j) - \widehat{y}_l(x_s,\alpha^j)|^2 p_{e(s)}(\alpha^j|x_s) \mathbf{1}\Big\{\alpha^j(x_s) \notin T_{x_s}(\widehat{y}_{e(s)})\Big\}\right)^{1/2}$$
$$\times \left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s}\frac{1}{k}\sum_{j=1}^{k} \frac{1}{p_{e(s)}(\alpha^j|x_s)} \mathbf{1}\Big\{\alpha^j(x_s) \notin T_{x_s}(\widehat{y}_{e(s)})\Big\}\right)^{1/2}$$
$$\le \left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s}\frac{1}{k}\sum_{a \in T_{x_s}(\widehat{y}_{e(s)})} |f^*(x_s,a) - \widehat{y}_l(x_s,a)|^2\right)^{1/2}$$
$$+ \left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}} \frac{1}{k}\mathbb{E}_{x_s, a\sim p_{e(s)}(\cdot|x_s)} |f^*(x_s,a) - \widehat{y}_l(x_s,a)|^2\right)^{1/2}$$
$$\times \left(\sum_{i=1}^{l-1} \frac{n_i}{N_{l-1}} \mathbb{E}_x \frac{1}{k}\sum_{j=1}^{k} \frac{1}{p_i(\alpha^j|x)} \mathbf{1}\Big\{\alpha^j(x) \notin T_x(\widehat{y}_i)\Big\}\right)^{1/2}.$$

We further upper bound the above by

$$\left\{\left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}}\mathbb{E}_{x_s}\frac{1}{k}\sum_{a\in T_{x_s}(\widehat{y}_{e(s)})}|f^*(x_s,a)-\widehat{y}_l(x_s,a)|^2\right)^{1/2}\right.$$
$$\left.+\left(\frac{1}{N_{l-1}}\sum_{s=1}^{N_{l-1}}\frac{1}{k}\mathbb{E}_{x_s,a\sim p_{e(s)}(\cdot|x_s)}|f^*(x_s,a)-\widehat{y}_k(x_s,a)|^2\right)^{1/2}\right\}$$
$$\times\left(1\vee\sum_{i=1}^{l-1}\frac{n_i}{N_{l-1}}\mathbb{E}_x\frac{1}{k}\sum_{j=1}^{k}\frac{1}{p_i(\alpha^j|x)}\mathbf{1}\Big\{\alpha^j\notin T_x(\widehat{y}_i)\Big\}\right)^{1/2}$$
$$\leq\left(\frac{2}{N_{l-1}}\sum_{s=1}^{N_{l-1}}\frac{1}{k}\mathbb{E}_{x_s}\left[\sum_{a\in T_{x_s}(\widehat{y}_{e(s)})}|f^*(x_s,a)-\widehat{y}_l(x_s,a)|^2+\mathbb{E}_{a\sim p_{e(s)}(\cdot|x_s)}|f^*(x_s,a)-\widehat{y}_l(x_s,a)|^2\right]\right)^{1/2} \tag{8}$$
$$\times\left(1\vee\sum_{i=1}^{l-1}\frac{n_i}{N_{l-1}}\mathbb{E}_x\frac{1}{k}\sum_{j=1}^{k}\frac{1}{p_i(\alpha^j|x)}\mathbf{1}\Big\{a^j(x)\notin T_x(\widehat{y}_i)\Big\}\right)^{1/2} \tag{9}$$

where we use $(\sqrt{a}+\sqrt{b})^2\leq 2(a+b)$ for nonnegative a,b. Now, observe that

$$\mathbb{E}_{x_s}\left[\sum_{a\in T_{x_s}(\widehat{y}_{e(s)})}|f^*(x_s,a)-\widehat{y}_l(x_s,a)|^2+\mathbb{E}_{a\sim p_{e(s)}(\cdot|x_s)}|f^*(x_s,a)-\widehat{y}_l(x_s,a)|^2\right] \tag{10}$$
$$=\mathbb{E}_{x_s,\mathcal{A}_s}\left\{\sum_{a\in\mathcal{A}_s}(\widehat{y}_l(x_s,a)-f^*(x_s,a))^2|\mathcal{H}_{s-1}\right\} \tag{11}$$

by the definition of the selected set $\mathcal{A}_s$ in Algorithm 1 with $r=1$. Under the event (7), the expression in (8) is at most ϕ_l. We now turn to the expression in (9). Note that by definition, for any strategy α^j

$$\frac{1}{p_i(\alpha^j|x)}\mathbf{1}\Big\{a^j(x)\notin T_x(\widehat{y}_i)\Big\}=\Big[(A-k+1)+\gamma_i(\widehat{y}_i(x,\widehat{\alpha}_i^k)-\widehat{y}_i(x,\alpha^j))\Big]\mathbf{1}\Big\{\alpha^j(x)\notin T_x(\widehat{y}_i)\Big\}$$
$$\leq(A-k+1)+\gamma_i\Big[\widehat{y}_i(x,\widehat{\alpha}_i^k)-\widehat{y}_i(x,\alpha^j)\Big]_+,$$

where $[a]_+=\max\{a,0\}$. Therefore, by Lemma 3, for any non-overlapping strategies $\alpha^1,\ldots,\alpha^k$,

$$\frac{1}{k}\sum_{j=1}^{k}\frac{1}{p_i(\alpha^j|x)}\mathbf{1}\Big\{\alpha^j(x)\notin T_x(\widehat{y}_i)\Big\}\leq(A-k+1)+\frac{1}{k}\sum_{j=1}^{k}\gamma_i\Big[\widehat{y}_i(x,\widehat{\alpha}_i^k)-\widehat{y}_i(x,\alpha^j)\Big]_+$$
$$\leq(A-k+1)+\frac{1}{k}\sum_{j=1}^{k}\gamma_i\Big[\widehat{y}_i(x,\widehat{\alpha}_i^j)-\widehat{y}_i(x,\alpha^j)\Big].$$

Since the above expression is at least $(A-k+1)\geq 1$, we may drop the maximum with 1 in (9). Putting

everything together,

$$\mathbb{E}_x \frac{1}{k}\sum_{j=1}^{k} |\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j)| \le \phi_l \cdot \left((A-k+1) + \sum_{i=1}^{l-1} \frac{n_i}{N_{l-1}} \gamma_i \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[\widehat{y}_i(x,\widehat{\alpha}_i^j) - \widehat{y}_i(x,\alpha^j) \Big] \right)^{1/2}$$

To prove the second statement, by induction we upper bound the above expression by

$$\phi_l \cdot \left((A-k+1) + \max_{i<l} \gamma_i \left\{ 2\mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j) \Big] + \frac{A}{\gamma_i} \right\} \right)^{1/2}$$

$$\le \phi_l \cdot \left(2(A-k+1) + 2\gamma_l \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j) \Big] \right)^{1/2}.$$

□

We now prove that inductive hypothesis holds for each epoch l.

Lemma 2. *Suppose we set $\gamma_l = \sqrt{A-k+1}/(32\phi_l)$ for each l, and that event $\mathcal{E}$ in (7) holds. Then the induction hypothesis holds for each $l \ge 2$.*

Proof. The base of the induction ($l = 2$) is satisfied trivially if $\gamma_2 = O(1)$ since functions are bounded. Now suppose the induction hypothesis (l) holds for some $l \ge 2$. We shall prove it for $(l+1)$.

Denote by $\boldsymbol{\alpha} = (\alpha^1, \ldots, \alpha^k)$ any set of non-overlapping strategies. We also use the shorthand $A' = A - k + 1$ for the size of the support of the IGW distribution. Define

$$\mathsf{R}(\boldsymbol{\alpha}) = \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j) \Big], \quad \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) = \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[\widehat{y}_l(x,\widehat{\alpha}_l^j) - \widehat{y}_k(x,\alpha^j) \Big].$$

Since

$$\Big[f^*(x,\alpha^{*,j}) - f^*(x,a) \Big] = \Big[\widehat{y}_l(x,\alpha^{*,j}) - \widehat{y}_l(x,a) \Big] + \Big[f^*(x,\alpha^{*,j}) - \widehat{y}_l(x,\alpha^{*,j}) \Big] + [\widehat{y}_l(x,a) - f^*(x,a)],$$

it holds that

$$\begin{aligned}
&\sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - f^*(x,\alpha^j) \Big] \\
&= \sum_{j=1}^{k} \Big[\widehat{y}_l(x,\alpha^{*,j}) - \widehat{y}_l(x,\alpha^j) \Big] + \sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - \widehat{y}_l(x,\alpha^{*,j}) \Big] + \sum_{j=1}^{k} \Big[\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j) \Big] \\
&\le \sum_{j=1}^{k} \Big[\widehat{y}_l(x,\widehat{\alpha}_l^j) - \widehat{y}_l(x,\alpha^j) \Big] + \sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - \widehat{y}_l(x,\alpha^{*,j}) \Big] + \sum_{j=1}^{k} \Big[\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j) \Big].
\end{aligned}$$

Therefore, for any $\boldsymbol{\alpha}$,

$$\begin{aligned}
\mathsf{R}(\boldsymbol{\alpha}) \le \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} &\Big[\widehat{y}_l(x,\alpha^{*,j}) - \widehat{y}_l(x,\alpha^j) \Big] \\
&+ \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[f^*(x,\alpha^{*,j}) - \widehat{y}_l(x,\alpha^{*,j}) \Big] + \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[\widehat{y}_l(x,\alpha^j) - f^*(x,\alpha^j) \Big]. \qquad (12)
\end{aligned}$$

For the middle term in (12), we apply the last statement of Lemma 1 to $\alpha^{*,1}, \ldots, \alpha^{*,k}$. We have:

$$\mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} \Big[f^*(x, \alpha^{*,j}) - \widehat{y}_l(x, \alpha^{*,j}) \Big] \leq \sqrt{2A'} \phi_l.$$

For the last term in (12),

$$\frac{1}{k} \sum_{j=1}^{k} \mathbb{E}_x \Big[\widehat{y}_l(x, \alpha^j) - f^*(x, \alpha^j) \Big] \leq \sqrt{2} \phi_l \cdot \big(A' + \gamma_l \mathsf{R}(\boldsymbol{\alpha}) \big)^{1/2}.$$

Hence, we have the inequality

$$\begin{aligned}
\mathsf{R}(\boldsymbol{\alpha}) &\leq \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) + \sqrt{2A'} \phi_l + \sqrt{2} \phi_l \cdot \big(A' + \gamma_l \mathsf{R}(\boldsymbol{\alpha}) \big)^{1/2} \\
&\leq \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) + 2\phi_l \sqrt{2A'} + \phi_l \sqrt{2\gamma_l \mathsf{R}(\boldsymbol{\alpha})} \\
&\leq \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) + 2\phi_l \sqrt{2A'} + \gamma_l \phi_l^2 + \frac{1}{2} \mathsf{R}(\boldsymbol{\alpha})
\end{aligned}$$

and thus

$$\mathsf{R}(\boldsymbol{\alpha}) \leq 2\widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) + 4\phi_l \sqrt{2A'} + 2\gamma_l \phi_l^2 \leq 2\widehat{\mathsf{R}}_l(q) + A'/(2\gamma_l)$$

On the other hand,

$$\Big[\widehat{y}_l(x, \widehat{\alpha}_l^j) - \widehat{y}_l(x, \alpha^j) \Big] = \Big[f^*(x, \widehat{\alpha}_l^j) - f^*(x, \alpha^j) \Big] + \Big[\widehat{y}_l(x, \widehat{\alpha}_l^j) - f^*(x, \widehat{\alpha}_l^j) \Big] + \Big[f^*(x, \alpha^j) - \widehat{y}_l(x, \alpha^j) \Big]$$

and so

$$\begin{aligned}
&\sum_{j=1}^{k} \Big[\widehat{y}_l(x, \widehat{\alpha}_l^j) - \widehat{y}_l(x, \alpha^j) \Big] \\
&= \sum_{j=1}^{k} \Big[f^*(x, \widehat{\alpha}_l^j) - f^*(x, \alpha^j) \Big] + \sum_{j=1}^{k} \Big[\widehat{y}_l(x, \widehat{\alpha}_l^j) - f^*(x, \widehat{\alpha}_l^j) \Big] + \sum_{j=1}^{k} \Big[f^*(x, \alpha^j) - \widehat{y}_l(x, \alpha^j) \Big] \\
&\leq \sum_{j=1}^{k} \Big[f^*(x, \alpha^{*,j}) - f^*(x, \alpha^j) \Big] + \sum_{j=1}^{k} \Big[\widehat{y}_l(x, \widehat{\alpha}_l^j) - f^*(x, \widehat{\alpha}_l^j) \Big] + \sum_{j=1}^{k} \Big[f^*(x, \alpha^j) - \widehat{y}_l(x, \alpha^j) \Big].
\end{aligned}$$

Therefore, for any $\boldsymbol{\alpha}$

$$\widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) \leq \mathsf{R}(\boldsymbol{\alpha}) + \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} [\widehat{y}_l(x, \widehat{\alpha}_l^j) - f^*(x, \widehat{\alpha}_l^j)] + \mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} [f^*(x, \alpha^j) - \widehat{y}_l(x, \alpha^j)]. \tag{13}$$

The last term in (13) is bounded by Lemma 1 by

$$\begin{aligned}
\mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} |f^*(x, \alpha^j) - \widehat{y}_l(x, \alpha^j)| &\leq \sqrt{2} \phi_l \cdot \big(A' + \gamma_l \mathsf{R}(\boldsymbol{\alpha}) \big)^{1/2} \\
&\leq \sqrt{2} \phi_l \cdot \Big(A' + 2\gamma_l \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) + A'/2 \Big)^{1/2} \\
&\leq 2\phi_l \sqrt{A'} + 2\phi_l^2 \gamma_l + \frac{1}{2} \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) \\
&\leq \frac{A'}{4\gamma_l} + \frac{1}{2} \widehat{\mathsf{R}}_l(\boldsymbol{\alpha}).
\end{aligned}$$

Now, for the middle term in (13), we use the above inequality with $\widehat{\boldsymbol{\alpha}}_l = (\widehat{\alpha}_l^1, \dots, \widehat{\alpha}_l^k)$:

$$\mathbb{E}_x \frac{1}{k} \sum_{j=1}^{k} [\widehat{y}_l(x, \widehat{\alpha}_l^j) - f^*(x, \widehat{\alpha}_l^j)] \le \frac{A'}{4\gamma_l} + \frac{1}{2} \widehat{\mathsf{R}}_l(\widehat{\boldsymbol{\alpha}}_l) = \frac{A'}{4\gamma_l}.$$

Putting the terms together,

$$\widehat{\mathsf{R}}_l(\boldsymbol{\alpha}) \le 2\mathsf{R}(\boldsymbol{\alpha}) + \frac{A'}{\gamma_l}.$$

Since $\boldsymbol{\alpha}$ is arbitrary, the induction step follows. □

Lemma 3. *For $v \in \mathbb{R}^A$, let $\widehat{a}^1, \dots, \widehat{a}^k$ be indices of largest k coordinates of v in decreasing order. Let $a^1, \dots, a^k$ be any other set of distinct coordinates. Then*

$$\sum_{j=1}^{k} [v(\widehat{a}^k) - v(a^j)]_+ \le \sum_{j=1}^{k} v(\widehat{a}^j) - v(a^j)$$

Proof. We prove this by induction on r. For $r = 1$,

$$[v(\widehat{a}^1) - v(a^1)]_+ = v(\widehat{a}^1) - v(a^1)$$

Induction step: Suppose

$$\sum_{j=1}^{k-1} [v(\widehat{a}^k) - v(b^j)]_+ \le \sum_{j=1}^{k-1} v(\widehat{a}^j) - v(b^j)$$

for any $b^1, \dots, b^{k-1}$. Let $a^m = \operatorname{argmin}_{j=1,\dots,k} v(a^j)$. Since all the values are distinct, it must be that $v(\widehat{a}^k) \ge v(a^m)$. Applying the induction hypothesis to $\{a^1, \dots, a^k\} \setminus \{a^m\}$ and adding

$$[v(\widehat{a}^k) - v(a^m)]_+ = v(\widehat{a}^k) - v(a^m)$$

to both sides concludes the induction step. □

Proof of Theorem 1. Recall that on epoch l, the strategy is $\alpha_l^1 = \widehat{\alpha}_l^1, \dots, \alpha_l^{k-1} = \widehat{\alpha}_l^{k-1}$ for the first $k-1$ arms, and then sampling $\alpha_l^k(x)$ from IGW distribution p_l. Observe that for any x and any draw $\alpha_l^k(x)$, the set of k arms is distinct (i.e. the strategies are non-overlapping), and thus under the event $\mathcal{E}$ in (7), Lemma 1 and inductive statements hold. Hence, expected regret per step in epoch l is bounded as

$$\begin{aligned}
&\mathbb{E}_{x, \alpha_l^k(x)} \sum_{j=1}^{k} [f^*(x, \alpha^{*,j}) - f^*(x, \alpha_l^j)] && (14)\\
&\le \frac{k(A-k+1)}{\gamma_l} + 2\mathbb{E}_{x, \alpha_l^k(x)} \sum_{j=1}^{k} [\widehat{y}_l(x, \widehat{\alpha}_l^j)) - \widehat{y}_l(x, \alpha^j)] \\
&= \frac{k(A-k+1)}{\gamma_l} + 2\mathbb{E}_{x, \alpha_l^k(x)} [\widehat{y}_l(x, \widehat{\alpha}_l^k)) - \widehat{y}_l(x, \alpha_l^k)] \\
&\le \frac{k(A-k+1)}{\gamma_l} + 2\mathbb{E}_x \sum_{a \notin T_x(\widehat{y}_l)} \frac{\widehat{y}_l(x, \widehat{\alpha}_l^k) - \widehat{y}_l(x, a)}{(A-k+1) + \gamma_l [\widehat{y}_l(x, \widehat{\alpha}_l^k) - \widehat{y}_l(x, a)]} \\
&\le \frac{k(A-k+1)}{\gamma_l} + \frac{2(A-k+1)}{\gamma_l} && (15)
\end{aligned}$$

From Lemma 5, the event $\mathcal{E}$ in (7) holds with probability at least $1-\delta$ if we set

$$\phi_l = \sqrt{\frac{162}{cN_{l-1}} \log\left(\frac{|\mathcal{F}|N_{l-1}^3}{\delta}\right)}.$$

Now recall that we set $N_l = 2^l \leq 2T$ and $\gamma_l = \sqrt{A-k+1}/(32\phi_l)$. Combining this with equation (15), we find that the cumulative regret is bounded with probability at least $1-\delta$ by

$$\begin{aligned}
R(T) &\leq \sum_{l=2}^{e(T)} \frac{(k+2)(A-k+1)N_{l-1}}{\gamma_l} \\
&\leq c^{-1/2} 408(k+2)\sqrt{(A-k+1)\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)} \sum_{l=2}^{\log_2(2T)} 2^{(l-1)/2} \\
&\leq c^{-1/2} 2308(k+2)\sqrt{(A-k+1)T\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)}.
\end{aligned}$$

□

Proof of Theorem 2. The proof is essentially the same as the proof of Theorem 1.

From Lemma 7, the event $\mathcal{E}$ in (7) holds with probability at least $1-\delta$ if we set

$$\phi_l = \sqrt{\frac{420}{cN_{l-1}} \log\left(\frac{|\mathcal{F}|N_{l-1}^3}{\delta}\right) + 2\epsilon^2}.$$

Combining this with equation (15) we get that the regret is bounded by,

$$\begin{aligned}
R(T) &\leq \sum_{l=2}^{e(T)} \frac{(k+2)(A-k+1)N_{l-1}}{\gamma_l} \\
&\leq c^{-1/2} 656(k+2)\sqrt{(A-k+1)\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)} \sum_{l=2}^{\log_2(2T)} 2^{(l-1)/2} + 46(k+2)\sqrt{(A-k+1)\epsilon^2} \sum_{l=2}^{e(T)} N_{l-1} \\
&\leq c^{-1/2} 3711(k+2)\sqrt{(A-k+1)T\log\left(\frac{|\mathcal{F}|T^3}{\delta}\right)} + 46(k+2)T\sqrt{(A-k+1)\epsilon^2}
\end{aligned}$$

given $\mathcal{E}$ is true. □

B Regression Martingale Bound

Recall that we have the following dependence structure in our problem. On each round s, context x_s is drawn independently of the past $\mathcal{H}_{s-1}$ and rewards $\boldsymbol{r}_s = \{r_s(a)\}_{a\in\mathcal{A}}$ are drawn from the distribution with mean $f^*(x_s, a)$. The algorithm selects a random set $\mathcal{A}_s$ given x_s, and feedback is provided for a (possibly random) subset $\Phi_s \subseteq \mathcal{A}$. Importantly, $\mathcal{A}_s$ and Φ_s are independent of $\boldsymbol{r}_s$ given x_s.

The next lemma considers a single time step s, conditionally on the past $\mathcal{H}_{s-1}$.

Lemma 4. *Let $x_s, \boldsymbol{r}_s = \{r_s(a)\}_{a\in\mathcal{A}}$ be sampled from the data distribution, and let $\mathcal{A}_s \subseteq \mathcal{A}$ be conditionally independent of $\boldsymbol{r}_s$ given x_s. Let $\Phi_s \subseteq \mathcal{A}_s$ be a random subset given $\mathcal{A}_s$ and x_s, but independent of $\boldsymbol{r}_s$. Fix an arbitrary $f : \mathcal{X} \times \mathcal{A} \to [0, 1]$ and define the following random variable,*

$$Y_s = \frac{1}{k}\sum_{a\in\mathcal{A}}\Big((f(x_s, a) - r_s(a))^2 - (f^*(x_s, a) - r_s(a))^2\Big) \times \mathbf{1}\{a \in \Phi_s\}.$$

Then, under the realizability assumption (Assumption 1), we have the following,

$$\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] = \frac{1}{k}\sum_{a\in\mathcal{A}}\mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\Big\{(f(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \Phi_s\}\Big\}$$

and

$$\mathrm{Var}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] \le 4\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s].$$

Proof. By the conditional independence assumptions,

$$\begin{aligned}\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] &= \frac{1}{k}\sum_{a\in\mathcal{A}}\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}\{(f(x_s, a) - f^*(x_s, a))(f(x_s, a) + f^*(x_s, a) - 2r_s(a)) \times \mathbf{1}\{a \in \Phi_s\}\}\\ &= \frac{1}{k}\sum_{a\in\mathcal{A}}\mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\Big\{(f(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \Phi_s\}\Big\}.\end{aligned}$$

We also have

$$\begin{aligned}Y_s^2 &\le \frac{1}{k}\sum_{a\in\mathcal{A}}(f(x_s, a) - f^*(x_s, a))^2(f(x_s, a) + f^*(x_s, a) - 2r_s(a))^2 \times \mathbf{1}\{a \in \Phi_s\}\\ &\le \frac{4}{k}\sum_{a\in\mathcal{A}}(f(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \Phi_s\}.\end{aligned}$$

□

Lemma 5. *Let $\widehat{y}_l$ be the estimate of the regression function f^* at epoch l. Assume the conditional independence structure in Lemma 4 and suppose Assumption 1 holds. Let $\mathcal{H}_{t-1}$ denote history (filtration) up to time $t-1$. Then for any $\delta < 1/e$,*

$$\mathcal{E} = \left\{l \ge 2 : \sum_{s=1}^{N_{l-1}}\mathbb{E}_{x_s,\mathcal{A}_s}\left\{\frac{1}{k}\sum_{a\in\mathcal{A}_s}(\widehat{y}_l(x_s, a) - f^*(x_s, a))^2|\mathcal{H}_{s-1}\right\} \le c^{-1}81\log\left(\frac{|\mathcal{F}|N_{l-1}^3}{\delta}\right)\right\}$$

holds with probability at least $1 - \delta$.

Proof. Following Lemma 4, let

$$Y_s(f) = \frac{1}{k}\sum_{a\in\mathcal{A}}\Big((f(x_s, a) - r_s(a))^2 - (f^*(x_s, a) - r_s(a))^2\Big) \times \mathbf{1}\{a \in \Phi_s\}.$$

The argument proceeds as in (Agarwal et al., 2012). Let $\mathbb{E}_s$ and Var_s denote the conditional expectation and conditional variance given $\mathcal{H}_{s-1}$. By Freedman's inequality (Bartlett et al., 2008), for any t, with probability at least $1 - \delta' \log t$, we have

$$\sum_{s=1}^{t} \mathbb{E}_s[Y_s(f)] - \sum_{s=1}^{t} Y_s(f) \leq 4\sqrt{\sum_{s=1}^{t} \mathrm{Var}_s[Y_s(f)] \log(1/\delta') + 2\log(1/\delta')}$$

Let $X(f) = \sqrt{\sum_{s=1}^{t} \mathbb{E}_s[Y_s(f)]}$, $Z(f) = \sum_{s=1}^{t} Y_s(f)$ and $C = \sqrt{\log(1/\delta')}$. In view of Lemma 4, with probability at least $1 - \delta' \log t$,

$$X(f)^2 - Z(f) \leq 8CX(f) + 2C^2$$

and hence

$$(X(f) - 4C)^2 \leq Z(f) + 18C^2.$$

Consequently, with the aforementioned probability, for all functions $f \in \mathcal{F}$ (and, in particular, for $\widehat{y}_l$),

$$(X(f) - 4C')^2 \leq Z(f) + 18C'^2$$

where $C' = \sqrt{\log(|\mathcal{F}|/\delta')}$. Now recall that

$$\widehat{y}_l = \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{t=1}^{N_{l-1}} \sum_{a \in \Phi_t} (f(x_t, a) - r_t(a))^2$$

where Φ_t is a random feedback set satisfying Assumption 3. Hence, $Z(\widehat{y}_l) \leq 0$ for $t = N_{l-1}$, implying that with probability at least $1 - \delta'/(N_{l-1}^2)$,

$$\sum_{s=1}^{N_{l-1}} \frac{1}{k} \sum_{a \in \mathcal{A}} \mathbb{E}_{x_s, \mathcal{A}_s, \Phi_s} \left\{ (\widehat{y}_l(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \Phi_s\} | \mathcal{H}_{s-1} \right\} \leq 81 \log\left(\frac{|\mathcal{F}| N_{l-1}^2 \log(N_{l-1})}{\delta'} \right).$$

We now take a union bound over l and recall that $\sum_{i \geq 1} 1/i^2 = \pi^2/6 < 2$.

Finally, observe that by Assumption 3,

$$\begin{aligned}
&\mathbb{E}_{x_s, \mathcal{A}_s, \Phi_s} \left\{ (\widehat{y}_l(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \Phi_s\} | \mathcal{H}_{s-1} \right\} \\
&= \mathbb{E}_{x_s, \mathcal{A}_s} \left\{ (\widehat{y}_l(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \mathcal{A}_s\} \times \mathbb{P}(a \in \Phi_s | x_s, \mathcal{A}_s) | \mathcal{H}_{s-1} \right\} \\
&\geq c \cdot \mathbb{E}_{x_s, \mathcal{A}_s} \left\{ (\widehat{y}_l(x_s, a) - f^*(x_s, a))^2 \mathbf{1}\{a \in \mathcal{A}_s\} | \mathcal{H}_{s-1} \right\}.
\end{aligned}$$

We conclude that with probability at least $1 - 2\delta'$, for all $l \geq 2$,

$$\sum_{s=1}^{N_{l-1}} \frac{1}{k} \mathbb{E}_{x_s, \mathcal{A}_s} \left\{ \sum_{a \in \mathcal{A}_s} (\widehat{y}_l(x_s, a) - f^*(x_s, a))^2 | \mathcal{H}_{s-1} \right\} \leq c^{-1} 81 \log\left(\frac{|\mathcal{F}| N_{l-1}^2 \log(N_{l-1})}{\delta'} \right).$$

□

C Regression Martingale Bound with Misspecification

Lemma 6. *Under the notation and assumptions of Lemma 4, but in the case of misspecified model (Assumption 2 replacing Assumption 1), it holds that*

$$\mathrm{Var}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] \le 8\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] + 16\epsilon^2.$$

Proof. The proof is along the lines of Lemma 4 (see also (Foster and Rakhlin, 2020)). We have for any $f : \mathcal{X} \times \mathcal{A} \to [0,1]$,

$$\begin{aligned}
\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] &= \frac{1}{k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}\{(f(x_s,a) - f^*(x_s,a))(f(x_s,a) + f^*(x_s,a) - 2r_s(a)) \times \mathbf{1}\{a \in \Phi_s\}\} \\
&= \frac{1}{k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\Big\{(f(x_s,a) - f^*(x_s,a))^2 \times \mathbf{1}\{a \in \Phi_s\}\Big\} \\
&+ \frac{2}{k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\{(f(x_s,a) - f^*(x_s,a))(f^*(x_s,a) - \mathbb{E}_{\boldsymbol{r}_s}[r(a)|x_s]) \times \mathbf{1}\{a \in \Phi_s\}\}.
\end{aligned}$$

Rearranging, using AM-GM inequality, and Assumption 2,

$$\begin{aligned}
&\frac{1}{k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\Big\{(f(x_s,a) - f^*(x_s,a))^2 \times \mathbf{1}\{a \in \Phi_s\}\Big\} \\
&= \mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] - \frac{2}{k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\{(f(x_s,a) - f^*(x_s,a))(f^*(x_s,a) - \mathbb{E}_{\boldsymbol{r}_s}[r(a)|x_s]) \times \mathbf{1}\{a \in \Phi_s\}\} \\
&\le \mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] + \frac{1}{2k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\Big\{(f(x_s,a) - f^*(x_s,a))^2 \times \mathbf{1}\{a \in \Phi_s\}\Big\} + 2\epsilon^2.
\end{aligned}$$

Rearranging,

$$\frac{1}{k}\sum_{a\in\mathcal{A}} \mathbb{E}_{x_s,\mathcal{A}_s,\Phi_s}\Big\{(f(x_s,a) - f^*(x_s,a))^2 \times \mathbf{1}\{a \in \Phi_s\}\Big\} \le 2\mathbb{E}_{x_s,\boldsymbol{r}_s,\mathcal{A}_s,\Phi_s}[Y_s] + 4\epsilon^2.$$

On the other hand,

$$\begin{aligned}
Y_s^2 &\le \frac{1}{k}\sum_{a\in\mathcal{A}} (f(x_s,a) - f^*(x_s,a))^2 (f(x_s,a) + f^*(x_s,a) - 2r_s(a))^2 \times \mathbf{1}\{a \in \Phi_s\} \\
&\le \frac{4}{k}\sum_{a\in\mathcal{A}} (f(x_s,a) - f^*(x_s,a))^2 \times \mathbf{1}\{a \in \Phi_s\}.
\end{aligned}$$

Combining the two inequalities concludes the proof. □

Lemma 7. *Let $\widehat{y}_l$ be the estimate of the regression function f^* at epoch l. Assume the conditional independence structure in Lemma 4 and suppose Assumption 2 holds. Let $\mathcal{H}_{t-1}$ denote history (filtration) up to time $t-1$. Then for any $\delta < 1/e$,*

$$\mathcal{E} = \left\{ l \ge 2 : \sum_{s=1}^{N_{l-1}} \mathbb{E}_{x_s,\mathcal{A}_s}\left\{ \frac{1}{k}\sum_{a\in\mathcal{A}_s} (\widehat{y}_l(x_s,a) - f^*(x_s,a))^2 | \mathcal{H}_{s-1} \right\} \le c^{-1} 210 \log\left(\frac{|\mathcal{F}| N_{l-1}^3}{\delta} \right) + \epsilon^2 N_{l-1} \right\}$$

holds with probability at least $1-\delta$.

Proof. We follow the proof of Lemma 5 to see how the misspecification level ϵ^2 enters the bounds.

Let $X(f) = \sum_{s=1}^{t} \mathbb{E}_s[Y_s(f)]$, $Z(f) = \sum_{s=1}^{t} Y_s(f)$, $C = \log(1/\delta')$ and $M = \epsilon^2 t$. Now using Lemma 6 and Freedman's inequality in the proof of Lemma 5, we find that with probability at least $1 - \delta' \log t$,

$$\begin{aligned} & X(f) - Z(f) \leq 8\sqrt{C(2X(f) + 4\epsilon^2 t)} + 2C \\ \implies & (X(f) - Z(f) - 2C)^2 \leq 128 X(f) C + 256 MC \\ \implies & (X(f) - 66C - Z(f))^2 \leq 4352 C^2 + 256 MC + 128 Z(f) C. \end{aligned}$$

The above bound holds for a fixed function f. We now apply an union bound to conclude that for all functions $f \in \mathcal{F}$, with probability at least $1 - \delta' \log t$,

$$\begin{aligned} (X(f) - 66C' - Z(f))^2 &\leq 4352 C'^2 + 256 MC' + 128 Z(f) C' \\ &\leq 20736 C'^2 + M^2 + 128 Z(f) C' \end{aligned}$$

where $C' = \log(|\mathcal{F}|/\delta')$. As in Lemma 5, $Z(\widehat{y}_l) \leq 0$ when $t = N_{l-1}$ and thus with probability at least $1 - \delta' \log(N_{l-1})$,

$$X(\widehat{y}_l) \leq 210 C' + \epsilon^2 N_{l-1}.$$

Hence, with probability at least $1 - \delta'/N_{l-1}^2$,

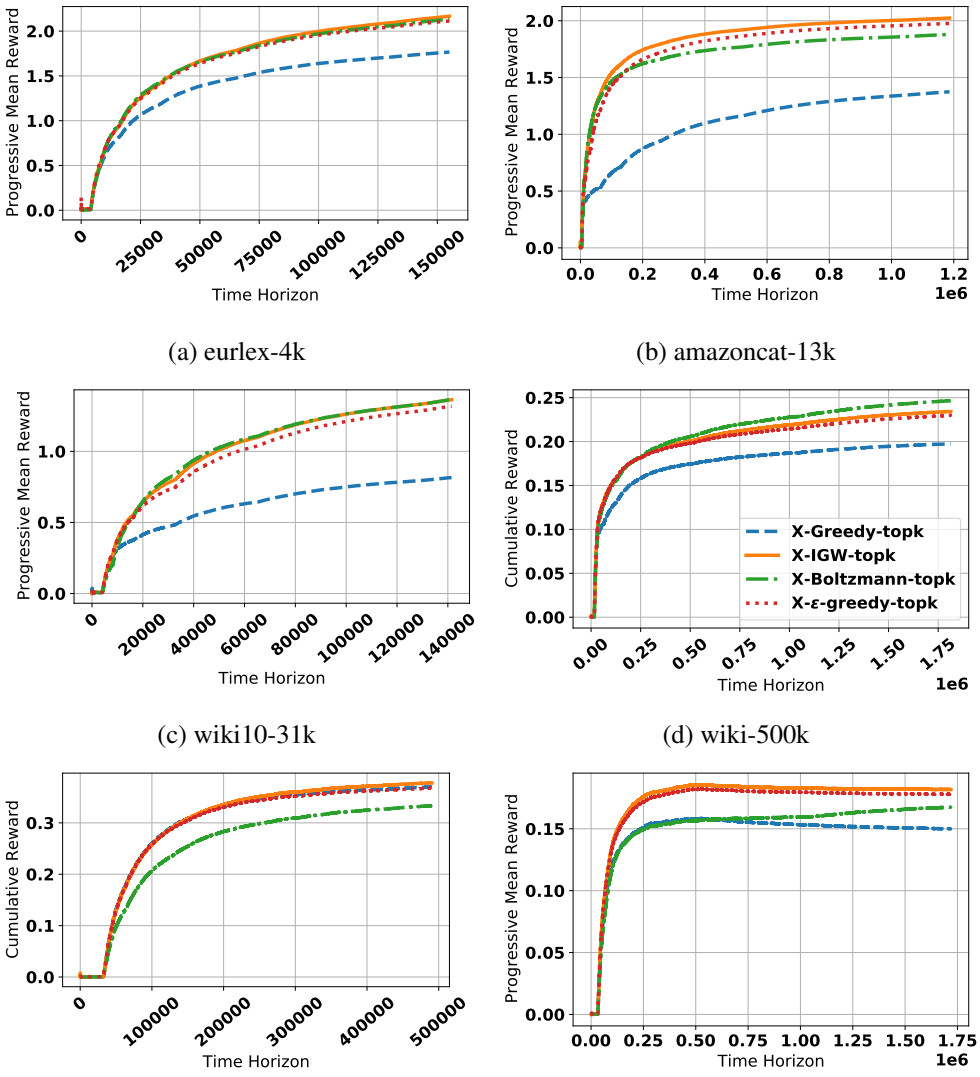
$$\sum_{s=1}^{N_{l-1}} \frac{1}{k} \sum_{a \in \mathcal{A}} \mathbb{E}_{x_s, \mathcal{A}_s, \Phi_s} \left\{ (\widehat{y}_l(x_s, a) - f^*(x_s, a))^2 \times \mathbf{1}\{a \in \Phi_s\} | \mathcal{H}_{s-1} \right\} \leq 210 \log\left(\frac{|\mathcal{F}| N_{l-1}^2 \log(N_{l-1})}{\delta'} \right) + \epsilon^2 N_{l-1}.$$

The rest of the proof proceeds exactly as in Lemma 5. □

D Reduction from eXtreme to $\log(A)$-armed Contextual Bandits

In this section we will prove Corollary 1 which is a reduction style argument. We reduce the A armed top-k contextual bandit problem under Definition 1 to a Z armed top-k contextual bandit problem where $Z = O(\log A)$.

Proof of Corollary 1. Note that the proof of Theorem 1 does not require the physical definition of an arm being consistent across all contexts as long as realizability holds. Let us assume w.l.o.g that Algorithm 2 returns the internal and leaf effective arms for any context x in $\mathcal{A}_x$ in a deterministic ordering. Let us call the j-th effective arm in this ordering for any context as arm j. This defines a system with Z arms where $Z \leq (p-1)b(H-1) + bm$ as Z is the number of effective arms returned by the beam-search in Algorithm 2. Recall the definition of the new function class $\tilde{\mathcal{F}}$ from Section 4.1. We can thus say that when Definition 1 holds this new system is a Z armed top-k contextual bandit system with realizablity (Assumption 1) with function class $\tilde{\mathcal{F}}$. Therefore the first part of coroallary 1 is implied by Theorem 1. Similarly when Definition 1 holds along with Assumption 2, this new system is a Z armed top-k contextual bandit system with ϵ-realizablity (Assumption 2) with function class $\tilde{\mathcal{F}}$. Therefore the second part of corollary 1 is implied by Theorem 2. Note that we have used the fact $|\tilde{\mathcal{F}}| = |\mathcal{F}|$. □



(a) eurlex-4k

(b) amazoncat-13k

(c) wiki10-31k

(d) wiki-500k

(e) amazon-670k

(f) amazon-3m

Figure 4: We plot the progressive mean rewards collected by each algorithm as a function of time. All algorithms are implemented under our eXtreme reduction framework. The initialization held out set for each dataset is used to train the hierarchy and the routing functions. Then the regressors for all nodes are trained on collected data at the beginning of each epoch. In all our experiments we have $k = 5$. In Algorithm 3 we set the number of explore slots $r = 3$. The common legend for all the plots is provided in (d). The beam-size used is $b = 10$.

E More Experiments

In Figure 4 we plot the progressive mean rewards vs time for all the experiments using simulated bandit feedback on eXtreme datasets.

F Implementation Details

For the realizable experiment on Eurlex-4k shown in Figure 3(a), the optimal weights ν^*'s are obtained by training ridge regression on the rewards vs context for each arm in the dataset. During the experiment we also use the same function class, that is one ridge regression is trained per arm on all collected data during the course of the algorithm. The reward for arm a given context x is chosen as $r_t(a) = [x; 1.0]^T \nu_a^* + \epsilon_t$, where ϵ_t is a zero-mean Gaussian noise.

Simulated Bandit Feedback: A sample in a multi-label dataset can be described as $(x, \mathbf{y})$ where $x \in \mathcal{X}$ can be thought of as the context while $\mathbf{y} \in \{0, 1\}^L$ denotes the correct classes. We can shuffle such a dataset into an ordering $\{(x_t, \mathbf{y}^{(t)})\}_{t=1}^T$. Then we feed one sample from the dataset at each time step to the contextual bandit algorithm that we are evaluating, in the following manner,

- at time t, send the input x_t to the contextual bandit algorithm,
- the contextual bandit algorithm then chooses an action corresponding to k arms $\boldsymbol{a}_t$,
- the environment then reveals the reward for **only** the k arms chosen $\boldsymbol{r}_t(\boldsymbol{a}_t)$, i.e. whether the arms chosen are among the correct classes or not.

Note that the algorithm is free to optimize its policy for choosing arms based on everything it has seen so far. In practice however, most contextual bandit algorithms will improve their policy (the $\widehat{y}$ it has learnt) in batches. The total number of positive classes selected by the algorithm in this process is the total reward collected by the algorithm.

eXtreme **Framework:** We follow the framework described in Section 4. We first form the tree and the routing functions from the held out portion of each dataset. The assumption is that there is a small supervised dataset available to each algorithm before proceeding with the simulated bandit feedback experiment. This dataset is used to form a balanced binary tree over the labels till the penultimate level. The nodes in the penultimate level can have a maximum of m children which are the original arms. The value of m is specified in Table 1 for each dataset. The division of the labels in each level of the tree is done through hierarchical 2-means clustering over label embeddings, where at each clustering step we use the algorithm from (Dhillon, 2001). The specific label embedding technique that we use is called Positive Instance Feature Aggregation (PIFA) (see (Prabhu et al., 2018) for more details). The routing functions for each internal node in the tree is essentially a one-vs-all linear classifier trained on the held out set. The classifiers are trained using a SVM ℓ_2-hinge loss. The positive and negative examples for each internal node is selected similar to the strategy in (Prabhu et al., 2018). Finally for the regression function $\tilde{f}(x, \tilde{a})$ where $\tilde{a}$ can be an original arm or an internal node in the tree, we train a linear regressor $\tilde{f}(x, \tilde{a}) = \nu_{\tilde{a}}^T [x; 1]$ as we progress through the experiment as in Algorithm 3. Note that the held out dataset is only used to train the tree and the routing function for each of the algorithms, while the regression functions are trained from scratch only using the samples observed during the bandit feedback experiment. The details are as follows:

- **Tree:** Initially a small part of the dataset is supplied to the algorithms in full-information mode. The size of this portion is captured in Table 1 in the Initialization Size column. This portion is

used to construct an approximately balanced binary tree over the labels. A supervised multilabel dataset can be represented as (X, Y) where $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times L}$. We form an embedding for each label using PIFA (Prabhu et al., 2018; Yu et al., 2020). Essentially the embedding for each label is the average of all instances that the label is connected to, normalized to ℓ_2 norm 1. Then we use approximately balanced 2-means recursively to form the tree until each leaf has less than a predefined maximum number of labels. The exact clustering algorithm used at each step is (Dhillon, 2001).

- **Routing Functions:** The routing functions are essentially one-vs-all linear classifiers at each internal node of the tree. The positive examples for the classifier at an internal node are the input instances in the small supervised dataset that have a positive label in the subtree of that node. The negative instances are the set of all instances that has a positive label in the subtree of the parent of that node but not in that node's subtree. This is the same methodology as in (Prabhu et al., 2018). The routing functions are trained using LinearSVC (Fan et al., 2008).
- **Regression Functions:** After creating the tree and the routing function from the small held out set, they are held fixed. The function class $\tilde{\mathcal{F}}$ as Algorithm 3 progresses is a set of linear regression functions at each internal and leaf node of the tree. They are trained on past data collected during the course of the previous epochs. Note that the examples for training the regressor for an internal node are only from the singleton arms that were shown when the algorithm selected that particular internal node in the IGW sampling. The regression functions are trained using LinearSVR (Fan et al., 2008).
- **Hyper-parameter Tuning:** For all the exploration algorithms in the eXtreme experiments the parameters are tuned over the eurlex-4k dataset and then held fixed. For the IGW scheme C is tuned over a grid of $\{1e-7, 1e-6, \cdots, 1e7\}$. The same is done for the β in the Boltzmann scheme. For ϵ-greedy the ϵ value is tuned between $[1e-7, 1.0]$ in a equally spaced grid in the logarithmic scale. The best parameters that are found are $\beta = 1.0, C = 1.0$ and $\epsilon = 0.167$.
- **Inference:** Inference using a trained model is done exactly according to Algorithm 3. The beam-search over the routing function yields effective arms. Then we evaluate the linear regression functions for each of the effective arms (singleton arms or the internal nodes in the tree). If a non-singleton effective arm is chosen among the k arms we randomly sample a singleton arm in it's subtree. The beam search and IGW sampling is implemented in C++ where the linear operations are implemented using the Eigen package (Guennebaud et al., 2010).