On Two Continuum Armed Bandit Problems in High Dimensions

Citation for published version:
Tyagi, H, Stich, SU & Gärtner, B 2016, 'On Two Continuum Armed Bandit Problems in High Dimensions', *Theory of Computing Systems*, vol. 58, no. 1, pp. 191-222. https://doi.org/10.1007/s00224-014-9570-8

Digital Object Identifier (DOI):
10.1007/s00224-014-9570-8

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Theory of Computing Systems

Theory Comput Syst (2016) 58:191–222
DOI 10.1007/s00224-014-9570-8

On Two Continuum Armed Bandit Problems in High Dimensions

Hemant Tyagi · Sebastian U. Stich · Bernd Gärtner

Received: 7 February 2014 / Accepted: 23 August 2014 / Published online: 12 September 2014

Abstract We consider the problem of continuum armed bandits where the arms are indexed by a compact subset of $\mathbb{R}^d$. For large d, it is well known that mere smoothness assumptions on the reward functions lead to regret bounds that suffer from the curse of dimensionality. A typical way to tackle this in the literature has been to make further assumptions on the structure of reward functions. In this work we assume the reward functions to be intrinsically of low dimension $k \ll d$ and consider two models: (i) The reward functions depend on only an unknown subset of k coordinate variables and, (ii) a generalization of (i) where the reward functions depend on an unknown k dimensional subspace of $\mathbb{R}^d$. By placing suitable assumptions on the smoothness of the rewards we derive randomized algorithms for both problems that achieve nearly optimal regret bounds in terms of the number of rounds n.

Keywords Bandit problems · Continuum armed bandits · Functions of few variables · Online optimization · Low-rank matrix recovery

1 Introduction

In the continuum armed bandit problem, a player is given a set of strategies S—typically a compact subset of $\mathbb{R}^d$. At each round $t = 1, \ldots, n$, the player chooses a strategy $\mathbf{x}_t$ from S and then receives a reward $r_t(\mathbf{x}_t)$. Here $r_t : S \rightarrow \mathbb{R}$ is the reward

H. Tyagi (✉) · S. U. Stich · B. Gärtner
Department of Computer Science, Institute of Theoretical Computer Science, ETH Zürich, Zürich, Switzerland
e-mail: htyagi@inf.ethz.ch

S. U. Stich
e-mail: sstich@inf.ethz.ch

B. Gärtner
e-mail: gaertner@inf.ethz.ch

function chosen by the environment at time t according to the underlying model. The player selects strategies across different rounds with the goal of maximizing the total expected reward. Specifically, the performance of the player is measured in terms of *regret*. This is defined as the difference between the total expected reward of the best fixed (i.e. not varying with time) strategy and the expected reward of the sequence of strategies played by the player. If the regret after n rounds is sub-linear in n, this implies as $n \to \infty$ that the per-round expected reward of the player asymptotically approaches that of the best fixed strategy.

The problem faced by the player at each round is the classical "exploration-exploitation dilemma". On one hand if the player chooses to focus his attention on a particular strategy which he considers to be the best ("exploitation") then he might fail to know about other strategies which have a higher expected reward. However if the player spends too much time collecting information ("exploration") then he might fail to play the optimal strategy sufficiently often. Some applications of continuum armed bandit problems are in: (i) online auction mechanism design [11, 28] where the set of feasible prices is representable as an interval and, (ii) online oblivious routing [9] where S is a flow polytope.

For a d-dimensional strategy space, if the only assumption made on the reward functions is on their degree of smoothness then any algorithm will incur worst-case regret which depends exponentially on d [26]. To see this, let $S = [-1, 1]^d$ and consider a time invariant reward function that is zero in all but one orthant $\mathcal{O}$ of S. More precisely, let $R(n)$ denote the cumulative regret incurred by the algorithm after n rounds. Bubeck et al. [13] showed that $R(n) = \Omega(n^{\frac{d+1}{d+2}})$ after $n = \Omega(2^d)$ plays for stochastic continuum armed bandits[1] with d-variate Lipschitz continuous mean reward functions defined over $[0, 1]^d$. Clearly the per-round expected regret $R(n)/n = \Omega(n^{\frac{-1}{d+2}})$ which means that it converges to zero at a rate at least exponentially slow in d. This curse of dimensionality is avoided by reward functions possessing more structure, two popular cases being linear reward functions (see for example [2, 33]) and convex reward functions (see for example [21, 26]) for which the regret is *polynomial* in d and sub-linear in n.

Low Dimensional Models for High Dimensional Reward Functions Recently there has been work in the online optimization literature where the reward functions are assumed to be *intrinsically* low-dimensional or in other words have only a few degrees of freedom compared to the ambient dimension. Carpentier et al. [15] and Yadkori et al. [1] consider the linear stochastic bandit problem with rewards $r_t(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \eta_t$. Here η_t; $t = 1, 2, \ldots, n$ is stochastic noise. They consider the setting where the unknown parameter $\mathbf{w}$ (of dimension d) is k-sparse with $k \ll d$. Chen et al. [16] consider the problem of Bayesian optimization of high dimensional functions by again assuming the functions to depend on only a few relevant variables. This model is generalized by Wang et al. [43] and Djolonga et al. [20] where the authors consider the underlying function to effectively vary along a low-dimensional subspace and adopt a Bayesian optimization framework.

[1] Rewards sampled at each round in an i.i.d manner from an unknown probability distribution.

In this work we consider two problems based on the nature of the low dimensional assumption placed on the reward functions. In both problems, we want to find algorithms that lead to small regret.

- **Problem 1.** We assume the reward functions $r_t : S \rightarrow \mathbb{R}$ to depend on an unknown subset of k coordinate variables implying $r_t(x_1, \ldots, x_d) = g_t(x_{i_1}, \ldots, x_{i_k})$. The underlying function g_t is assumed to be sampled by the environment at each round $t = 1, 2, \ldots$ either in an: (i) i.i.d manner from some fixed unknown probability distribution (stochastic environment) or (ii) arbitrarily (adversarial environment).
- **Problem 2.** We assume the reward functions $r_t : S \rightarrow \mathbb{R}$ to depend effectively on a fixed but unknown k-dimensional subspace of $\mathbb{R}^d$ implying $r_t(\mathbf{x}) = g_t(\mathbf{A}\mathbf{x})$ with $\mathbf{A} \in \mathbb{R}^{k \times d}$ being a full rank matrix. The underlying functions g_t are assumed to be sampled by a stochastic environment. This problem is a generalization of the preceding one, since in the special case when each row of $\mathbf{A}$ has a single 1 and 0's otherwise, we arrive at the setting of Problem 1.

Our Contributions Firstly, assuming $(i_1, \ldots, i_k)$ to be fixed across time but unknown to the player, we derive an algorithm CAB(d, k) that achieves a regret bound of $O(C_1(k, d) n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{1}{2\alpha+k}})$, after n rounds for Problem 1. Here $\alpha \in (0, 1]$ denotes the exponent of Hölder continuity of the reward functions while the factor $C_1(k, d) = O(\text{poly}(k) \cdot o(\log d))$ captures the uncertainty of not knowing the k active coordinates. When $\alpha = 1$, i.e. the reward functions are Lipschitz continuous, our bound is nearly optimal[2] in terms of n (up to the $(\log n)^{\frac{1}{2+k}}$ factor). Note that the number of rounds n after which the per-round regret $R(n)/n < c$, for any constant $0 < c < 1$, is exponential in k. Hence for $k \ll d$, we do not suffer from the curse of dimensionality. The algorithm is *anytime* in the sense that n is not required to be known and is a simple modification of the CAB1 algorithm [26]. The modification is in the manner of discretization of the continuous strategy set S for which we consider a probabilistic construction based on creating partitions of $\{1, \ldots, d\}$ into k disjoint subsets. We also extend our results to the setting where the k-tuple is allowed to change over time. Assuming that a sequence of k-tuples $(\mathbf{i}_t)_{t=1}^n = (i_{1,t}, \ldots, i_{k,t})_{t=1}^n$ is chosen by an adversary before the start of plays we show that CAB(d, k) achieves a regret bound of $O(C_1(k, d) H[(\mathbf{i}_t)_{t=1}^n] n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{1}{2\alpha+k}})$. Here $H[(\mathbf{i}_t)_{t=1}^n]$ denotes the[3] "hardness" of $(\mathbf{i}_t)_{t=1}^n$. In case $H[(\mathbf{i}_t)_{t=1}^n] \leq Q$ for some $Q > 0$ known to the player, the regret bound improves to $O(C_1(k, d) Q^{\frac{\alpha}{2\alpha+k}} n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{1}{2\alpha+k}})$.

Secondly, we provide a solution to Problem 2 by deriving an algorithm namely CAB-LP(d, k) which achieves a bound of $O(C_2(k, d) n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}})$ on the regret after n rounds. The factor $C_2(k, d) = O(\text{poly}(k) \cdot \text{poly}(d))$, captures the uncertainty of not knowing the k-dimensional sub-space spanned by the rows of $\mathbf{A}$. This bound is derived for a slightly restricted class of Lipschitz continuous mean reward functions.

[2] See Remark 1 in Section 3.1 for discussion on how the $\log n$ factor can be removed.

[3] See Definition 2 in Section 2.1.

In terms of n, it nearly matches[2] the $\Omega(n^{\frac{1+k}{2+k}})$ lower bound [13], for k-variate Lipschitz continuous mean reward functions. As explained earlier, the per-round regret $R(n)/n$ approaches zero (as n increases), at a rate exponential in k. Thus for $k \ll d$, we avoid the curse of dimensionality. We assume n to be known to the algorithm (hence it is not anytime) and refer to it as the sampling budget. The main idea of the algorithm is to first use a fraction of the budget for estimating the unknown k-dimensional sub-space spanned by the rows of the linear parameter matrix $\mathbf{A}$. After obtaining this estimate we then employ the CAB1 algorithm [26] which is restricted to play strategies only from the estimated subspace. To derive sub-linear regret bounds we show that a careful allocation of the sampling budget is necessary between the two phases.

Comparison with Similar Work in Literature Chen et al. [16] consider the setting of Problem 1. The authors consider a stochastic environment but assume the underlying reward functions to be samples from a Gaussian process (GP). They propose a two-stage scheme where they first learn the set of active variables and then apply a standard GP algorithm to perform Bayesian optimization over this identified set. They also derive regret bounds for their scheme.

Very recently, Djolonga et al. [20] considered the same bandit problem as Problem 2 above. Although they consider the mean reward function to reside in an RKHS (Reproducible Kernel Hilbert space) and adopt a Bayesian optimization framework, the scheme they employ is similar to ours. Wang et al. [43] also consider the setting of Problem 2, however their framework is significantly different from ours. They consider a Bayesian optimization framework and consider noise-less optimization of the reward functions assumed to be samples from a Gaussian process (GP). Furthermore they derive bounds on *simple regret* which is weaker then cumulative regret.

Other Related Work The continuum armed bandit problem was first introduced by Agrawal [3] for the case $d = 1$ where an algorithm achieving a regret bound of $o(n^{(2\alpha+1)/(3\alpha+1)+\eta})$ for any $\eta > 0$ was proposed for local Hölder continuous mean reward functions. Kleinberg et al. [28] proved a lower bound of $\Omega(n^{1/2})$ for this problem. This was then improved upon by Kleinberg [26] where the author derived upper and lower bounds of $O(n^{\frac{\alpha+1}{2\alpha+1}}(\log n)^{\frac{\alpha}{2\alpha+1}})$ and $\Omega(n^{\frac{\alpha+1}{2\alpha+1}})$ respectively. Cope [18] considered a class of mean reward functions defined over a compact convex subset of $\mathbb{R}^d$ which have (i) a unique maximum $\mathbf{x}^*$, (ii) are three times continuously differentiable and (iii) whose gradients are well behaved near $\mathbf{x}^*$. It was shown that a modified version of the Kiefer-Wolfowitz algorithm achieves a regret bound of $O(n^{1/2})$ which is also optimal. Auer et al. [8] considered the $d = 1$ case, with the mean reward function assumed to only satisfy a local Hölder condition around each maximum $\mathbf{x}^*$ with exponent $\alpha \in (0, \infty)$. Under these assumptions the authors considered a modification of Kleinberg's CAB1 algorithm [26] and achieved a regret bound of $O(n^{\frac{1+\alpha-\alpha\beta}{1+2\alpha-\alpha\beta}}(\log n)^{\frac{\alpha}{1+2\alpha-\alpha\beta}})$ for some known $0 < \beta < 1$. Kleinberg et al. [29] and Bubeck et al. [12] studied a very general setting for the multi-armed bandit problem in which S forms a metric space, with the reward function assumed to satisfy a Lipschitz condition with respect to this metric.

There has also been significant effort in other fields to develop tractable algorithms for *approximating* high dimensional functions from point queries. This is typically done by assuming that the functions intrinsically depend on either few variables (cf. [10, 17, 19, 25] and references within) or few linear parameters [22, 40].

Organization of the Paper The rest of the paper is organized as follows. In Section 2 we state the problems formally along with our main results. In Section 3 we describe an algorithm and derive regret bounds achieved by it for Problem 1. Next in Section 4 we describe an algorithm for Problem 2 and provide a detailed analysis deriving regret bounds achieved by it. Finally we provide concluding remarks in Section 5.

2 Problem Setup and Main Results

We now outline the problem setup including relevant notation and assumptions along with our main results for Problems 1 and 2 in Sections 2.1 and 2.2 respectively.

2.1 CAB Problem of Few Variables

The compact set of strategies $S \subset \mathbb{R}^d$ is taken to be $[0, 1]^d$. At each time step $t = 1, \ldots, n$, a reward function $r_t : S \to \mathbb{R}$ is chosen by the environment. Upon playing a strategy $\mathbf{x}_t \in [0, 1]^d$, the player receives the reward $r_t(\mathbf{x}_t)$ at time step t. For some $k \leq d$, we assume that each r_t depends on a fixed but unknown subset of k variables. This means $r_t(x_1, \ldots, x_d) = g_t(x_{i_1}, \ldots, x_{i_k})$ where $(i_1, \ldots, i_k)$ is a k-tuple with distinct integers $i_j \in \{1, \ldots, d\}$ and $g_t : [0, 1]^k \to \mathbb{R}$. For simplicity of notation, we denote the set of such k-tuples of the set $\{1, \ldots, d\}$ by $\mathcal{T}_k^d$ and the ℓ_2 norm by $\| \cdot \|$. We assume that k is known to the player, however it suffices to know that k is an upper bound for the number of active variables.[4] The second assumption that we make is on the smoothness property of the reward functions.

Definition 1 A function $f : [0, 1]^k \to \mathbb{R}$ is Hölder continuous with constant $0 \leq L < \infty$, exponent $0 < \alpha \leq 1$, if we have for all $\mathbf{u}, \mathbf{u}' \in [0, 1]^k$ that $|f(\mathbf{u}) - f(\mathbf{u}')| \leq L \parallel \mathbf{u} - \mathbf{u}' \parallel^{\alpha}$. We denote the class of such functions f as $\mathcal{C}(\alpha, L, k)$.

The function class defined in Definition 1 was also considered by Agrawal [3] and Kleinberg [26]. It can be seen as a generalization of Lipschitz continuity (obtained for $\alpha = 1$). We fix $0 < \alpha \leq 1$ and $0 \leq L < \infty$ throughout and now proceed to define the two models that we analyze for our problem. These models describe how the reward functions g_t are generated at each time step.

Stochastic Model The reward functions g_t are considered to be i.i.d samples from some fixed but unknown probability distribution over functions $g : [0, 1]^k \to \mathbb{R}$. We define the expectation of the reward function as $\bar{g}(\mathbf{u}) = \mathbb{E}[g(\mathbf{u})]$ where $\mathbf{u} \in [0, 1]^k$. We require $\bar{g}$ to belong to $\mathcal{C}(\alpha, L, k)$ and note that the individual samples g_t

[4] Indeed, any function that depends on $k' \leq k$ coordinates also depends on at most k coordinates.

need not necessarily be Hölder continuous. We make the following assumption of *sub-Gaussianity* on the distribution from which the random samples g are generated.

Assumption 1 *We assume that there exist constants* $\zeta, s_0 > 0$ *so that*

$$\mathbb{E}[e^{s(g(\mathbf{u})-\bar{g}(\mathbf{u}))}] \leq e^{\frac{1}{2}\zeta^2 s^2} \quad \forall s \in [-s_0, s_0], \mathbf{u} \in [0,1]^k.$$

The above assumption was considered by Kleinberg [26] for the case $d = 1$ and allows us to consider reward functions g_t whose range is not bounded. Note that the mean reward $\bar{g}$ is assumed to be Hölder continuous and is also defined on a compact set $[0,1]^k$, implying that it is bounded. The optimal strategy $\mathbf{x}^*$ is then defined as any point belonging to the set

$$\underset{\mathbf{x}\in[0,1]^d}{\operatorname{argmax}}\ \mathbb{E}[r(\mathbf{x})] = \underset{\mathbf{x}\in[0,1]^d}{\operatorname{argmax}}\ \bar{g}(x_{i_1},\ldots,x_{i_k}). \tag{1}$$

Adversarial Model The reward functions $g_t : [0,1]^k \to [0,1]$ are a fixed sequence of functions in $\mathcal{C}(\alpha, L, k)$ chosen arbitrarily by an *oblivious* adversary i.e., an adversary not adapting to the actions of the player. The optimal strategy $\mathbf{x}^*$ is then defined as any point belonging to the set

$$\underset{\mathbf{x}\in[0,1]^d}{\operatorname{argmax}} \sum_{t=1}^{n} r_t(\mathbf{x}) = \underset{\mathbf{x}\in[0,1]^d}{\operatorname{argmax}} \sum_{t=1}^{n} g_t(x_{i_1},\ldots,x_{i_k}). \tag{2}$$

Given the above models we measure the performance of a player over n rounds in terms of the *regret* defined as

$$R(n) := \sum_{t=1}^{n} \mathbb{E}\left[r_t(\mathbf{x}^*) - r_t(\mathbf{x}_t)\right] = \sum_{t=1}^{n} \mathbb{E}\left[g_t(\mathbf{x}^*_{i_1},\ldots,\mathbf{x}^*_{i_k}) - g_t\left(\mathbf{x}^{(t)}_{i_1},\ldots,\mathbf{x}^{(t)}_{i_k}\right)\right]. \tag{3}$$

In (3) the expectation is defined over (i) the random choice of g_t in the stochastic model and (ii) the random choice of the strategy $\mathbf{x}_t$ by the player in the stochastic/adversarial models.

Changing k-Tuple Across Time We also consider a more general *adversarial* setting where the k tuple $(i_1,\ldots,i_k)$ is allowed to change over time. Formally this means that the reward functions $(r_t)_{t=1}^n$ now have the form $r_t(x_1,\ldots,x_d) = g_t(x_{i_{1,t}},\ldots,x_{i_{k,t}})$ where $(i_{1,t},\ldots,i_{k,t})_{t=1}^n$ denotes the sequence of k-tuples chosen by the adversary before the start of plays. Here $g_t : [0,1]^k \to [0,1]$ with $g_t \in \mathcal{C}(\alpha, L, k)$ for each $t = 1, 2, \ldots$. We assume that the sequence of k-tuples is not "hard" meaning that it contains a small number of consecutive pairs (relative to the number of rounds n) with different values. This is formally defined as follows.

Definition 2 For any set $\mathcal{B}$ we define the hardness of the sequence $(b_1,\ldots,b_n) \in \mathcal{B}^n$ by:

$$H[b_1,\ldots,b_n] := 1 + |\{1 \leq l < n : b_l \neq b_{l+1}\}|. \tag{4}$$

The above definition was introduced by Auer et al. [7, Section 8] where the authors considered the non-stochastic multi-armed bandit problem. They employed the definition to characterize the hardness of a sequence of actions against which the regret of the player's actions is measured. The optimal strategy $\mathbf{x}^*$ is then defined as any point belonging to the set

$$\operatorname*{argmax}_{\mathbf{x}\in[0,1]^d} \sum_{t=1}^{n} r_t(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{x}\in[0,1]^d} \sum_{t=1}^{n} g_t(x_{i_{1,t}}, \ldots, x_{i_{k,t}}). \tag{5}$$

Furthermore the regret incurred by the player after n rounds is defined as

$$R(n) := \sum_{t=1}^{n} \mathbb{E}\left[r_t(\mathbf{x}^*) - r_t(\mathbf{x}_t)\right] = \sum_{t=1}^{n} \mathbb{E}\left[g_t(\mathbf{x}^*_{i_{1,t}}, \ldots, \mathbf{x}^*_{i_{k,t}}) - g_t\left(\mathbf{x}^{(t)}_{i_{1,t}}, \ldots, \mathbf{x}^{(t)}_{i_{k,t}}\right)\right]. \tag{6}$$

Main Results Our main results are as follows. Firstly, assuming that the k-tuple $(i_1, \ldots, i_k) \in \mathcal{T}_k^d$ is chosen once at the beginning of play and kept fixed thereafter, we provide in the form of Theorem 1 a bound on the regret which is $O(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} C_1(k,d))$ where $C_1(k,d) = O(\text{poly}(k) \cdot o(\log d))$. This bound holds for both the stochastic and adversarial models and is *nearly optimal*. To see this, we note that Bubeck et al. [13] showed a precise lower bound of $\Omega(n^{\frac{d+1}{d+2}})$ after $n = \Omega(2^d)$ plays for stochastic continuum armed bandits, with d-variate Lipschitz continuous mean reward functions, defined over $[0, 1]^d$. In our setting though, the reward functions depend on an unknown subset of k coordinate variables hence any algorithm after $n = \Omega(2^k)$ plays would incur worst case regret of $\Omega(n^{\frac{k+1}{k+2}})$. We see that our upper bound matches this lower bound for the case of Lipschitz continuous reward functions ($\alpha = 1$) up to a mild factor of $(\log n)^{\frac{1}{2+k}} C_1(k,d)$. We also note that the $(\log d)^{\frac{\alpha}{2\alpha+k}}$ factor in (7) accounts for the uncertainty of not knowing which k coordinates are active from $\{1, \ldots, d\}$.

Theorem 1 *Given that the k-tuple $(i_1, \ldots, i_k) \in \mathcal{T}_k^d$ is kept fixed across time but unknown to the player, the algorithm CAB(d, k) incurs a regret of*

$$O\left(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} k^{\frac{\alpha(k+6)}{2(2\alpha+k)}} (\log d)^{\frac{\alpha}{2\alpha+k}}\right) \tag{7}$$

after n rounds of play with high probability for both the stochastic and adversarial models.

The above result is proven in Section 3.1 along with a description of the CAB(d, k) algorithm which achieves this bound. The main idea here is to discretize $[0, 1]^d$ by first constructing a family of partitions $\mathcal{A}$ of $\{1, \ldots, d\}$ with each partition consisting of k disjoint subsets. The construction is probabilistic and the resulting $\mathcal{A}$ satisfies an important property (with high probability) namely the *Partition Assumption* as described in Section 3.1. In particular we have that $|\mathcal{A}|$ is $O(ke^k \log d)$ resulting in a total of $M^k|\mathcal{A}|$ sampling points for some integer $M > 0$. This discrete strategy set is then used with a finite armed bandit algorithm such as UCB-1 [5] for the stochastic setting and Exp3 [6] for the adversarial setting, to achieve the regret bound of Theorem 1.

Secondly we extend our results to the setting where $(i_1, \ldots, i_k)$ can change over time. Considering that an oblivious adversary chooses arbitrarily before the start of plays a sequence of k-tuples $(\mathbf{i}_t)_{t=1}^n = (i_{1,t}, \ldots, i_{k,t})_{t=1}^n$ of *hardness* $H[(\mathbf{i}_t)_{t=1}^n] \leq Q$ with $Q > 0$ known to the player, we show how Algorithm CAB(d, k) can be adapted to this setting to achieve a regret bound of $O\left(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} Q^{\frac{\alpha}{2\alpha+k}} C_1(k, d)\right)$. In case the player has no knowledge of Q, the regret bound then changes to $O\left(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} H\left[(\mathbf{i}_t)_{t=1}^n\right] C_1(k, d)\right)$. Although our bound becomes trivial when $H[(\mathbf{i}_t)_{t=1}^n]$ is close to n (as one would expect), we can still achieve sub-linear regret when $H[(\mathbf{i}_t)_{t=1}^n]$ is small relative to n. We again consider a discretization of the space $[0, 1]^d$ constructed using the family of partitions $\mathcal{A}$ mentioned earlier. The difference lies in now using the Exp3.S algorithm [7] on the discrete strategy set, which in contrast to the Exp3 algorithm is designed to control regret against arbitrary sequences. This is described in Section 3.2.

2.2 CAB Problem of Few Linear Parameters

We consider the set of strategies S to be the ℓ_2-ball of radius $1+\nu$ for some $\nu > 0$ denoted as $B_d(1+\nu)$. At each time $t = 1, \ldots, n$ the environment chooses a reward function $r_t : B_d(1+\nu) \rightarrow \mathbb{R}$. Upon playing the strategy $\mathbf{x}_t$ the player receives the reward $r_t(\mathbf{x}_t)$. Here the number of rounds n (sampling budget) is assumed to be known to the player. We consider the setting where each r_t depends on $k \ll d$ unknown linear parameters $\mathbf{a}_1, \ldots, \mathbf{a}_k \in \mathbb{R}^d$ with k assumed to be known to the player. In particular, denoting $\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_k]^T \in \mathbb{R}^{k \times d}$ we assume that $r_t(\mathbf{x}) = g_t(\mathbf{A}\mathbf{x})$.

The reward functions g_t are considered to be samples from some fixed but unknown probability distribution over functions $g : B_k(1+\nu) \rightarrow \mathbb{R}$. We then have the expected reward function as $\bar{g}(\mathbf{u}) = \mathbb{E}[g(\mathbf{u})]$ where $\mathbf{u} \in B_k(1+\nu)$. More specifically we consider the model:

$$r_t(\mathbf{x}) = \bar{g}(\mathbf{A}\mathbf{x}) + \eta_t \; ; \quad t = 1, 2, \ldots, n \tag{8}$$

where $(\eta_t)_{t=1}^n$ is i.i.d Gaussian noise with mean $\mathbb{E}[\eta_t] = 0$ and variance $\mathbb{E}[\eta_t^2] = \sigma^2$. We assume $\bar{g}$ to be sufficiently smooth - in particular to be two times continuously differentiable. Specifically, we assume for some constant $C_2 > 0$ that the magnitude of all partial derivatives of $\bar{g}$, up to order two, are bounded by C_2:

$$\sup_{|\beta| \leq 2} \parallel D^\beta \bar{g} \parallel_\infty \leq C_2 \; ; \quad D^\beta \bar{g} = \frac{\partial^{|\beta|} \bar{g}}{\partial y_1^{\beta_1} \ldots \partial y_k^{\beta_k}} \; , \; |\beta| = \beta_1 + \cdots + \beta_k. \tag{9}$$

Note that this is slightly stronger then assuming Lipschitz continuity.[5] Also note that the individual samples g_t need not necessarily be smooth. We now make additional

[5] Indeed for a compact domain, any C^2 function is Lipschitz continuous but the converse is not necessarily true. Therefore, the mean reward functions that we consider, belong to a slightly restricted class of Lipschitz continuous functions.

assumptions on the mean reward function $\bar{g}$. In fact it was shown by Fornasier et al. [22] that such additional assumptions are also necessary in order to formulate a tractable algorithm. For example when $k = 1$, if we only make smoothness assumptions on $\bar{g}$, then one can construct $\bar{g}$ so that $\Omega(2^d)$ many samples are needed to distinguish between $\bar{r}(\mathbf{x}) \equiv 0$ and $\bar{r}(\mathbf{x}) \equiv \bar{g}(\mathbf{a}^T\mathbf{x})$ [22].

To this end, we define the following matrix:

$$H^r := \int_{\mathbb{S}^{d-1}} \nabla\bar{r}(\mathbf{x})\nabla\bar{r}(\mathbf{x})^T d\mathbf{x} = \mathbf{A}^T \cdot \int_{\mathbb{S}^{d-1}} \nabla\bar{g}(\mathbf{A}\mathbf{x})\nabla\bar{g}(\mathbf{A}\mathbf{x})^T d\mathbf{x} \cdot \mathbf{A} \tag{10}$$

where the second equality follows from the identity $\nabla\bar{r}(\mathbf{x}) = \mathbf{A}^T\nabla\bar{g}(\mathbf{A}\mathbf{x})$. Let $\sigma_i(H^r)$ denote the i^{th} singular value of H^r. We make a technical assumption related to the conditioning of H^r. This assumption allows us to derive a tractable algorithm for our problem. We assume for some $\alpha > 0$ that:

$$\sigma_1(H^r) \geq \sigma_2(H^r) \geq \cdots \geq \sigma_k(H^r) \geq \alpha > 0. \tag{11}$$

The parameter α determines the tractability of our algorithm. As explained in Section 4.4, there are interesting function classes that satisfy (11) for usable values of α.

Following Fornasier et al. [22], we also assume without loss of generality, $\mathbf{A}$ to be row orthonormal so that $\mathbf{A}\mathbf{A}^T = \mathbf{I}$. Indeed if this is not the case then through SVD (singular value decomposition) of $\mathbf{A}$ we obtain $\mathbf{A} = \underbrace{\mathbf{U}}_{k\times k}\underbrace{\Sigma}_{k\times k}\underbrace{\mathbf{V}^T}_{k\times d}$ where $\mathbf{U}, \Sigma, \mathbf{V}^T$ are unitary, diagonal and row-orthonormal matrices respectively. Therefore we obtain

$$\bar{r}(\mathbf{x}) = \bar{g}(\mathbf{A}\mathbf{x}) = \bar{g}(\mathbf{U}\Sigma\mathbf{V}^T\mathbf{x}) = \bar{g}'(\mathbf{V}^T\mathbf{x})$$

where $\bar{g}'(\mathbf{y}) = \bar{g}(\mathbf{U}\Sigma\mathbf{y})$ for $\mathbf{y} \in B_k(1+\nu)$. Hence within a scaling of the parameter C_2 by a factor depending polynomially on $k, \sigma_1(\mathbf{A})$ we can assume $\mathbf{A}$ to be row-orthonormal.

Regret after n Rounds After n rounds of play the cumulative expected regret is defined as:

$$R(n) = \sum_{i=1}^{n} \mathbb{E}[r_t(\mathbf{x}^*) - r_t(\mathbf{x}_t)] = \sum_{i=1}^{n} [\bar{g}(\mathbf{A}\mathbf{x}^*) - \bar{g}(\mathbf{A}\mathbf{x}_t)], \tag{12}$$

where $\mathbf{x}^*$ is the optimal strategy belonging to the set

$$\underset{\mathbf{x}\in B_d(1+\nu)}{\operatorname{argmax}}\ \mathbb{E}[r_t(\mathbf{x})] = \underset{\mathbf{x}\in B_d(1+\nu)}{\operatorname{argmax}}\ \bar{g}(\mathbf{A}\mathbf{x}) \tag{13}$$

Here $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is the sequence of strategies played by the algorithm. The goal of the algorithm is to minimize regret i.e. ensure $R(n) = o(n)$ so that $\lim_{n\to\infty} R(n)/n = 0$.

Main Results Our main result is to derive a randomized algorithm namely CAB-LP(d, k) which achieves a regret bound of $O(C(k,d)n^{\frac{1+k}{2+k}}(\log n)^{\frac{1}{2+k}})$ after n rounds.

Here, $C(k,d) = O(\text{poly}(k) \cdot \text{poly}(d))$ accounts for the uncertainty of not knowing the k-dimensional sub-space spanned by the rows of $\mathbf{A}$. We state[6] this formally in the form of the following theorem below.

Theorem 2 *For any $p \in (0,1)$, there exists a constant $c' > 0$ so that algorithm CAB-LP(d, k) achieves a total regret of*

$$O\left(\frac{k^{13}d^2\sigma^2(\log(\frac{k}{p}))^4}{\alpha^4}(\max\left\{d,\alpha^{-1}\right\})^2\left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} + n^{\frac{1+k}{2+k}}(\log n)^{\frac{1}{2+k}}\right)$$

after n rounds with probability at least $1 - p - 6\exp(-c'd)$.

Recall that σ denotes the variance of the external Gaussian noise η in (8) while α was defined in (11). The parameter $p \in (0,1)$ controls the probability of success of the algorithm at the expense of increasing the total regret incurred. The algorithm essentially consists of two stages. The first stage involves using a fraction of the sampling budget n to recover an estimate of the row-space of $\mathbf{A}$. The second stage then involves usage of a finite armed bandit algorithm on the estimated subspace. Note that the dependence[7] of the regret bound in terms of n is $O(n^{\frac{1+k}{2+k}}(\log n)^{\frac{1}{2+k}})$. Say the linear parameter matrix $\mathbf{A}$, or even the sub-space spanned by its rows, was known. We then know a lower bound of $\Omega(n^{\frac{1+k}{2+k}})$ on regret, for k-variate Lipschitz continuous mean rewards [13]. In terms of n, our bound nearly matches this lower bound, albeit for a slightly restricted class of Lipschitz continuous mean reward functions. As discussed in Remark 1 in Section 3.1 it seems to be possible to remove the $\log n$ term completely by using recent results for finite-armed bandits. Lastly we also note the dependence of our regret bound on the parameter α. As explained in Section 4.4, α typically decreases as $d \to \infty$. Hence in order to obtain regret bounds that are at most polynomial in d we would like α to be polynomial in d^{-1}. To this end, Proposition 3 in Section 4.4 which was proven by Tyagi et al. [41], describes a fairly general class of functions for which α is $\Theta(d^{-1})$.

3 Analysis for CAB Problem of Few Variables

In this Section we provide a detailed analysis of our results for the first problem where the reward functions depend only on some unknown subset of k coordinate variables. Section 3.1 contains the analysis for the case when the k active coordinates are fixed across time. In Section 3.2 we then analyze the more general setting where the k active coordinates are allowed to change over time.

[6] This theorem is stated again in Section 4 for completeness.

[7] This is actually true for $k \geq 3$. For $k = 1, 2$ the $(n/\log n)^{\frac{4}{k+2}}$ factor dominates. See Remark 3 in Section 4.3 for details.

3.1 Analysis When k Active Coordinates are Fixed Across Time

We begin with the setting where the set of active k coordinates is fixed across time. The core of our analysis involves two parts. First, we construct a discretization of the search space, that is, a finite strategy set $\mathcal{P}_M \subset S = [0,1]^d$ where $M > 0$ is a parameter that will be defined later. Then, the problem reduces to a finite armed bandit problem on the discrete set $\mathcal{P}_M$. This problem can then be solved for instance with the UCB-1 (stochastic model) [5] or Exp3 (adversarial model) [6] algorithms.

UCB-1 is a deterministic algorithm that maintains a score for each arm and at each round selects the arm with the largest score. The score for any arm is the sum of two terms. The first term is the current average reward for the arm. The second term is related to the one sided confidence interval for the average reward within which the expected reward lies with high probability. Exp3 is a randomized algorithm that maintains a probability distribution over the arms. At each round, it selects an arm according to this distribution and then updates the distribution based on the reward observed.

The main property that the discrete strategy set $\mathcal{P}_M$ is required to satisfy is the following. For any $\mathbf{x} \in S = [0,1]^d$, $M > 0$ and any k-tuple $(i_1, \ldots, i_k)$ with distinct $i_j \in \{1, \ldots, d\}$ there exists $\mathbf{y} \in \mathcal{P}_M$ s.t. $|x_{i_j} - y_{i_j}| \leq (1/M)\ \forall j = 1, \ldots, k$. The idea is that for increasing values of M, we would have for any optimal $\mathbf{x}^*$ and any $(i_1, \ldots, i_k)$ the existence of an arbitrarily close point to $(x^*_{i_1}, \ldots, x^*_{i_k})$ in $\mathcal{P}_M$. Coupled with the Hölder continuity of the reward functions this then ensures that the finite armed bandit algorithm progressively plays strategies closer and closer to $\mathbf{x}^*$ leading to a bound on regret.

For $k = 1$, we could simply take the set $\mathcal{P}_M = \{\frac{i}{M}(1, \ldots, 1)^T : i = 1, \ldots, M\}$, i.e. a discretization of the diagonal.[8] For the general case where $k \geq 1$, our discretization uses a union of point sets of the form

$$\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_k) := \left\{ \frac{1}{M} \sum_{j=1}^{k} \lambda_j \mathbf{v}_j : \lambda_1, \ldots, \lambda_k \in \{1, 2, \ldots, M\} \right\},$$

where the $\mathbf{v}_j$'s are d-vectors. Suppose that for a given tuple $(i_1, \ldots, i_k)$ and for all j, the vector $\mathbf{v}_j$ has a 1-entry at coordinate i_j, and zero entries at the other $k-1$ coordinates of the tuple. Then the set $\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ discretizes S w.r.t. the tuple $(i_1, \ldots, i_k)$. An obvious way to achieve this is to set $\mathbf{v}_j = \mathbf{e}_{i_j}$ for all j, where $\mathbf{e}_i$ is the i-th canonical unit vector. Taking the union of the resulting $\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ over all tuples yields the desired discretization of S w.r.t. all tuples.

In the following we describe a more efficient construction in which individual sets $\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ take care of many tuples simultaneously. In this construction, each tuple $(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ will be induced by a partition of $\{1, \ldots, d\}$ into k disjoint subsets, with $\mathbf{v}_j$ being the characteristic vector of the j-th subset.

[8] The interested reader can find a full analysis for the case $k = 1$ in [42].

Definition 3 A family of partitions $\mathcal{A}$ of $\{1, \ldots, d\}$ into k disjoint subsets is said to satisfy the partition assumption if for any k distinct integers $i_1, i_2, \ldots, i_k \in \{1, \ldots, d\}$, there exists a partition $\mathbf{A} = (A_1, \ldots, A_k)$ in $\mathcal{A}$ such that each set in $\mathbf{A}$ contains exactly one of $i_1, i_2, \ldots, i_k$.

The above definition is known as *perfect hashing* in theoretical computer science and has many applications such as in finding juntas [34], table look-up [24] and communication complexity [37]. There exists a fairly simple probabilistic method using which one can construct $\mathcal{A}$ consisting of $O(ke^k \log d)$ partitions satisfying the partition assumption property with high probability. This is shown for instance by DeVore et al. [19]. They consider the significantly different *function approximation* problem as opposed to our setting of online optimization. For our purposes, we consider the aforementioned probabilistic construction. However, there also exist deterministic constructions resulting in larger family sizes such as the one proposed by Naor et al. [35], where a family of size $O(k^{O(\log k)} e^k \log d)$ is constructed deterministically in time $poly(d, k)$. We also note that the size of any family of partitions $\mathcal{A}$ that satisfies the partition assumption is $\Omega(e^k \log d / \sqrt{k})$ [23, 30, 36].

Constructing Strategy Set $\mathcal{P}_M$ Using $\mathcal{A}$ Suppose we are given a family of partitions $\mathcal{A}$ satisfying the partition assumption. Let $\chi_{A_j} = (\chi_{A_j}(1) \ldots \chi_{A_j}(d)) \in \{0, 1\}^d$ be defined as:

$$\chi_{A_j}(l) := \begin{cases} 1; & l \in A_j \\ 0; & \text{otherwise} \end{cases} ; \quad l = 1, 2, \ldots, d. \tag{14}$$

Algorithm 1 Algorithm CAB(d, k)

```
Input: $\alpha \in (0,1)$, $d, k$.
$T = 1$
Construct family of partitions $\mathcal{A}$ satisfying partition assumption
while $T \leq n$ do
    $M = \left\lceil \left(k^{\frac{\alpha-3}{2}} e^{-\frac{k}{2}} (\log d)^{-\frac{1}{2}} \sqrt{\frac{T}{\log T}}\right)^{2/(2\alpha+k)} \right\rceil$
  – Create $\mathcal{P}_M$ using $\mathcal{A}$
  – Initialize MAB with $\mathcal{P}_M$
    for $t = T, \ldots, \min(2T-1, n)$ do
        – get $\mathbf{x}_t$ from MAB
        – Play $\mathbf{x}_t$ and get $r_t(\mathbf{x}_t)$
        – Feed $r_t(\mathbf{x}_t)$ back to MAB
    end for
    $T = 2T$
end while
```

We then construct the discrete set of strategies $\mathcal{P}_M \subset [0, 1]^d$ for some fixed integer $M > 0$ as follows.

$$\mathcal{P}_M := \left\{ \frac{1}{M} \sum_{j=1}^{k} \lambda_j \chi_{A_j} : \lambda_j \in \{1, \ldots, M\}, (A_1, \ldots, A_k) \in \mathcal{A} \right\} \subset [0, 1]^d. \tag{15}$$

The above set of points was also employed by DeVore et al. [19] for the function approximation problem. Note that a strategy $\mathbf{x} = \frac{1}{M}\sum_{j=1}^{k}\lambda_j \chi_{\mathbf{A}_j}$ has coordinate value $\frac{1}{M}\lambda_j$ at each of the coordinate indices in A_j. Therefore we see that for each partition $\mathbf{A} \in \mathcal{A}$ we have M^k strategies implying a total of $M^k|\mathcal{A}|$ strategies in $\mathcal{P}_M$.

Projection Property An important property of the strategy set $\mathcal{P}_M$ is the following. Given any k-tuple of distinct indices $(i_1, \ldots, i_k)$ with $i_j \in \{1, \ldots, d\}$ and any integers $1 \le n_1, \ldots, n_k \le M$, there is a strategy $\mathbf{x} \in \mathcal{P}_M$ such that

$$(x_{i_1}, \ldots, x_{i_k}) = \left(\frac{n_1}{M}, \ldots, \frac{n_k}{M}\right).$$

To see this, one can simply take a partition $\mathbf{A} = (A_1, \ldots, A_k)$ from $\mathcal{A}$ such that each i_j is in a different set A_j for $j = 1, \ldots, k$. Then setting appropriate $\lambda_j = n_j$ when $i_j \in A_j$ we get that coordinate i_j of $\mathbf{x}$ has the value n_j/M.

Upper Bound on Regret We now describe our Algorithm CAB(d, k) and provide bounds on its regret. Note that the outer loop implements a standard doubling trick which is used as the player has no knowledge of the time horizon n. Observe that before the start of the inner loop of duration T, the player constructs the finite strategy set $\mathcal{P}_M$, where M increases progressively with T. Within the inner loop, the problem reduces to a finite armed bandit problem. The **MAB** routine can be any standard multi-armed bandit algorithm such as UCB-1 (stochastic model) or Exp3 (adversarial model). The algorithm is motivated by the CAB1 algorithm [26], however unlike the equispaced sampling done in CAB1 we consider a probabilistic construction of the discrete set of sampling points based on partitions of $\{1, \ldots, d\}$. We now present in the following lemma the regret bound incurred within an inner loop of duration T.

Lemma 1 *Given that $(i_1, \ldots, i_k)$ is fixed across time, say the strategy set $\mathcal{P}_M$ is used with (i) the UCB-1 algorithm for the stochastic setting or, (ii) the Exp3 algorithm for the adversarial setting. We then have for the choice $M = \left\lceil \left(k^{\frac{\alpha-3}{2}} e^{-\frac{k}{2}} (\log d)^{-\frac{1}{2}} \sqrt{\frac{T}{\log T}}\right)^{\frac{2}{2\alpha+k}} \right\rceil$ that the regret incurred by the player after T rounds is given by:*

$$R(T) = O\left(T^{\frac{\alpha+k}{2\alpha+k}} (\log T)^{\frac{\alpha}{2\alpha+k}} k^{\frac{\alpha(k+6)}{2(2\alpha+k)}} (\log d)^{\frac{\alpha}{2\alpha+k}}\right).$$

Proof Let $\mathbf{x}^* \in S$ denote the optimal strategy as defined in (1) or (2). For the k tuple $(i_1, \ldots, i_k) \in \mathcal{T}_k^d$, there exists $\mathbf{z} \in \mathcal{P}_M$ with $z_{i_1} = \frac{\lambda_1}{M}, \ldots, z_{i_k} = \frac{\lambda_k}{M}$ where $1 \le \lambda_1, \ldots, \lambda_k \le M$ are integers such that $|\lambda_j/M - x^*_{i_j}| < (1/M)$. This follows from the projection property of $\mathcal{A}$. We can then split up the total regret $R(T)$ in a part $R_1(T)$ incurred due to the discretization, and a part $R_2(T)$ incurred by **MAB**:

$$R_1(T) = \sum_{t=1}^{T} \mathbb{E}\left[g_t\left(x^*_{i_1}, \ldots, x^*_{i_k}\right) - g_t(z_{i_1}, \ldots, z_{i_k})\right], \tag{16}$$

$$R_2(T) = \sum_{t=1}^{T} \mathbb{E}\left[g_t(z_{i_1}, \ldots, z_{i_k}) - g_t\left(x_{i_1}^{(t)}, \ldots, x_{i_k}^{(t)}\right)\right]. \tag{17}$$

On account of the Hölder continuity of reward functions we have that

$$\mathbb{E}[g_t\left(x_{i_1}^*, \ldots, x_{i_k}^*\right) - g_t(z_{i_1}, \ldots, z_{i_k})] < L\left(\left(\frac{1}{M}\right)^2 k\right)^{\alpha/2}.$$

In other words, $R_1(T) = O(Tk^{\alpha/2}M^{-\alpha})$. In order to bound $R_2(T)$, we note that the problem has reduced to a $|\mathcal{P}_M|$-armed bandit problem. Specifically we note from (17) that we are comparing against a sub-optimal strategy $\mathbf{z}$ instead of the optimal one in $\mathcal{P}_M$. Hence $R_2(T)$ can be bounded by using existing bounds for finite-armed bandit problems. Now for the stochastic setting we can employ the UCB-1 algorithm [5] and play at each t a strategy $\mathbf{x}_t \in \mathcal{P}_M$. In particular, on account of Assumption 1, it can be shown that $R_2(T) = O\left(\sqrt{|\mathcal{P}_M| T \log T}\right)$ [26, Theorem 3.1]. For the adversarial setting we can employ the Exp3 algorithm [6] so that $R_2(T) = O\left(\sqrt{|\mathcal{P}_M| T \log |\mathcal{P}_M|}\right)$. Combining the bounds for $R_1(T)$ and $R_2(T)$ and recalling that $|\mathcal{P}_M| = O(M^k k e^k \log d)$ we obtain:

$$R(T) = O(TM^{-\alpha}k^{\alpha/2} + \sqrt{M^k k e^k \log d \; T \log T}) \text{ (stochastic) and,} \tag{18}$$

$$R(T) = O(TM^{-\alpha}k^{\alpha/2} + \sqrt{M^k k e^k \log d \; T \log(M^k k e^k \log d)}). \text{ (adversarial)} \tag{19}$$

Plugging $M = \left\lceil \left(k^{\frac{\alpha-3}{2}} e^{-\frac{k}{2}} (\log d)^{-\frac{1}{2}} \sqrt{\frac{T}{\log T}}\right)^{\frac{2}{2\alpha+k}} \right\rceil$ in (18) and (19) we obtain the stated bound on $R(T)$ for the respective models. □

Lastly equipped with the above bound we have that the regret incurred by Algorithm 1 over n plays is given by:

$$\sum_{i=0, T=2^i}^{i=\log n} R(T) = O\left(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} k^{\frac{\alpha(k+6)}{2(2\alpha+k)}} (\log d)^{\frac{\alpha}{2\alpha+k}}\right).$$

Remark 1 For the adversarial setting we can use the INF algorithm of Audibert et al. [4], as the MAB routine in our algorithm, and get rid of the $\log n$ factor from the regret bound. The same holds for the stochastic setting, if the range of the reward functions was restricted to [0, 1]. When the range of the reward functions is $\mathbb{R}$, as is the case in our setting, it seems possible to consider a variant of the MOSS algorithm [4] along with Assumption 2 on the distribution of the reward functions. Using proof techniques similar to those by Kleinberg [27], it might then be possible to remove the $\log n$ factor from the regret bound.

3.2 Analysis When k Active Coordinates Change Across Time

In this subsection, we consider the setting where the set of active k coordinates is allowed to change over time. The player does not know which k-tuple is chosen at each time t. As for the situation where the k-tuple was fixed, our algorithm will consist of two parts. First, we again construct the discrete strategy set $\mathcal{P}_M$ (as defined in (15)). Next, we employ the Exp3.S algorithm [7] on this discrete strategy set. This algorithm is designed to minimize the player's regret when measured against a Q-hard sequence, instead of only the constant – or 1-hard – sequence if the k-tuple was fixed over time.

Recall from (5) that the optimal strategy $\mathbf{x}^* \in \underset{\mathbf{x}\in[0,1]^d}{\operatorname{argmax}} \sum_{t=1}^{n} g_t(x_{i_{1,t}}, \ldots, x_{i_{k,t}})$. Since the sequence of k-tuples is Q-hard, this in turn implies for any $\mathbf{x}^*$ that

$$H\left[\left(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}}\right)_{t=1}^{n}\right] \leq Q.$$

Therefore we can now consider this as a setting where the players regret is measured against a Q-hard sequence $(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}})_{t=1}^{n}$.

By construction, we will have for any $\mathbf{x} \in [0,1]^d$ and any k-tuple $(i_1, \ldots, i_k)$, the existence of a point $\mathbf{z}$ in $\mathcal{P}_M$ such that $(z_{i_1}, \ldots, z_{i_k})$ approximates $(x_{i_1}, \ldots, x_{i_k})$ arbitrarily well for increasing values of M. Hence, for the optimal sequence $(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}})_{t=1}^{n}$, we have the existence of a sequence of points $(\mathbf{z}^{(t)})_{t=1}^{n}$ where $\mathbf{z}^{(t)} \in \mathcal{P}_M$ with the following two properties.

1. *Q-hardness.* $H\left[\left(z^{(t)}_{i_{1,t}}, \ldots, z^{(t)}_{i_{k,t}}\right)_{t=1}^{n}\right] \leq Q$. This follows easily from the Q-hardness of the sequence $\left(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}}\right)_{t=1}^{n}$ and by choosing for each $\left(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}}\right)$ a corresponding $\mathbf{z}^{(t)} \in \mathcal{P}_M$ such that $\| \left(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}}\right) - \left(z^{(t)}_{i_{1,t}}, \ldots, z^{(t)}_{i_{k,t}}\right) \|$ is minimized.
2. *Approximation property.* $\| \left(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}}\right) - \left(z^{(t)}_{i_{1,t}}, \ldots, z^{(t)}_{i_{k,t}}\right) \| = O(k^{\alpha/2} M^{-\alpha})$. This is easily verifiable via the projection property of the set $\mathcal{P}_M$.

Therefore by employing the Exp3.S algorithm [7] on the strategy set $\mathcal{P}_M$ we reduce the problem to a finite armed adversarial bandit problem where the players regret measured against the Q-hard sequence $(z^{(t)}_{i_{1,t}}, \ldots, z^{(t)}_{i_{k,t}})_{t=1}^{n}$ is bounded from above. The approximation property of this sequence (as explained above) coupled with the Hölder continuity of g_t ensures in turn that the players regret against the original sequence $(x^*_{i_{1,t}}, \ldots, x^*_{i_{k,t}})_{t=1}^{n}$ is also bounded. With this in mind we present the following lemma, which formally states a bound on regret after T rounds of play.

Lemma 2 *Given the above setting, say:*

1. *the sequence of k-tuples* $(i_{1,t}, \ldots, i_{k,t})_{t=1}^{n}$ *is at most* Q*-hard and,*
2. *the Exp3.S algorithm is used along with the strategy set* $\mathcal{P}_M$.

We then have for the choice $M = \left\lceil \left(k^{\frac{\alpha-3}{2}} e^{-\frac{k}{2}} (Q \log d)^{-\frac{1}{2}} \sqrt{\frac{T}{\log T}}\right)^{\frac{2}{2\alpha+k}} \right\rceil$ *that the regret incurred by the player after* T *rounds is given by:*

$$R(T) = O\left(T^{\frac{\alpha+k}{2\alpha+k}} (\log T)^{\frac{\alpha}{2\alpha+k}} k^{\frac{\alpha(k+6)}{2(2\alpha+k)}} (Q \log d)^{\frac{\alpha}{2\alpha+k}}\right).$$

Proof Recall from (6) the definition of regret $R(T)$ incurred after T rounds. At each time t, for some $\mathbf{z}^{(t)} \in \mathcal{P}_M$ we can split $R(T)$ into $R_1(T) + R_2(T)$ where $R_1(T) = \mathbb{E}\left[\sum_{t=1}^{T} g_t\left(x^*_{i_{1,t}}, \dots, x^*_{i_{k,t}}\right) - g_t\left(z^{(t)}_{i_{1,t}}, \dots, z^{(t)}_{i_{k,t}}\right)\right]$ and $R_2(T) = \mathbb{E}\left[\sum_{t=1}^{T} g_t(z^{(t)}_{i_{1,t}}, \dots, z^{(t)}_{i_{k,t}}) - g_t(x^{(t)}_{i_{1,t}}, \dots, x^{(t)}_{i_{k,t}})\right]$.

Let us consider $R_1(T)$ first. As before, from the projection property of $\mathcal{A}$ we have for each $\left(x^*_{i_{1,t}}, \dots, x^*_{i_{k,t}}\right)$, that there exists $\mathbf{z}^{(t)} \in \mathcal{P}_M$ with $z^{(t)}_{i_{1,t}} = \frac{\lambda^{(t)}_1}{M}, \dots, z^{(t)}_{i_{k,t}} = \frac{\lambda^{(t)}_k}{M}$ where $1 \le \lambda^{(t)}_1, \dots, \lambda^{(t)}_k \le M$ are integers such that $|\lambda^{(t)}_j/M - x^*_{i_{j,t}}| < (1/M)$ holds for $j = 1, \dots, k$ and each $t = 1, \dots, n$. Therefore from Hölder continuity of g_t we obtain $R_1(T) = O(Tk^{\alpha/2}M^{-\alpha})$. It remains to bound $R_2(T)$. To this end, note that the sequence $\left(z^{(t)}_{i_{1,t}}, \dots, z^{(t)}_{i_{k,t}}\right)_{t=1}^{n}$ with $\mathbf{z}^{(t)} \in \mathcal{P}_M$ is at most Q-hard. Hence the problem has reduced to a $|\mathcal{P}_M|$ armed adversarial bandit problem with a Q-hard optimal sequence of plays against which the regret of the player is to be bounded. This is accomplished by using the Exp3.S algorithm [7] which is designed to control regret against *any* Q-hard sequence of plays. In particular, using a result by Auer et al. [7, Corollary 8.3], we have that $R_2(T) = O\left(\sqrt{Q|\mathcal{P}_M|T \log(|\mathcal{P}_M|T)}\right)$. Combining the bounds for $R_1(T)$ and $R_2(T)$ and recalling that $|\mathcal{P}_M| = O(M^k k e^k \log d)$ we obtain the following expression for $R(T)$:

$$R(T) = O\left(Tk^{\alpha/2}M^{-\alpha} + \sqrt{QTM^k k e^k \log d \log(TM^k k e^k \log d)}\right). \tag{20}$$

Lastly after plugging in the value $M = \left\lceil \left(k^{\frac{\alpha-3}{2}} e^{-\frac{k}{2}} (Q \log d)^{-\frac{1}{2}} \sqrt{\frac{T}{\log T}}\right)^{\frac{2}{2\alpha+k}} \right\rceil$ in (20), we obtain the stated bound on $R(T)$. □

By employing Algorithm 1 with **MAB** sub-routine being the Exp3.S algorithm, we have that its regret over n plays is given by

$$\sum_{i=0, T=2^i}^{i=\log n} R(T) = O\left(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} k^{\frac{\alpha(k+6)}{2(2\alpha+k)}} (Q \log d)^{\frac{\alpha}{2\alpha+k}}\right).$$

Remark 2 In case the player does not know Q, a regret of

$$R(n) = O\left(n^{\frac{\alpha+k}{2\alpha+k}} (\log n)^{\frac{\alpha}{2\alpha+k}} k^{\frac{\alpha(k+6)}{2(2\alpha+k)}} (\log d)^{\frac{\alpha}{2\alpha+k}} H[(\mathbf{i}_t)_{t=1}^{n}]\right)$$

would be incurred by Algorithm 1 with the **MAB** routine being the Exp3.S algorithm and for the choice $M = \left\lceil \left(k^{\frac{\alpha-3}{2}} e^{-\frac{k}{2}} (\log d)^{-\frac{1}{2}} \sqrt{\frac{T}{\log T}}\right)^{\frac{2}{2\alpha+k}} \right\rceil$. Here $\mathbf{i}_t$ is shorthand notation for $(i_{1,t}, \ldots, i_{k,t})$. Indeed we simply use a result by Auer et al. [7, Corollary 8.2], on account of which we obtain $R_2(T) = O\left(H[(\mathbf{i}_t)_{t=1}^{n}]\sqrt{|\mathcal{P}_M|T \log(|\mathcal{P}_M|T)}\right)$. The rest follows along the lines of the proof of Lemma 2.

4 Analysis for CAB Problem of Few Linear Parameters

In this section we provide a detailed analysis of our scheme for the second problem. The reward functions here effectively depend on some unknown k dimensional subspace of $\mathbb{R}^d$ with $k \ll d$. Recall that the reward functions have the structure $r_t(\mathbf{x}) = g_t(\mathbf{A}\mathbf{x})$ where $\mathbf{A} \in \mathbb{R}^{k \times d}$ is full rank. The main idea behind our algorithm is to proceed in two phases. In **PHASE 1**, we use a fraction of the sampling budget n to recover an estimate of the (k dimensional) subspace spanned by the rows of $\mathbf{A}$. In **PHASE 2** we employ a standard continuum armed bandit algorithm that plays strategies from the previously estimated k dimensional subspace.

Intuitively we can imagine that the closer the estimated subspace is to the original one, the closer will the regret bound achieved by the CAB algorithm be to the one it would have achieved by playing strategies from the unknown k-dimensional subspace. However one should be careful here since spending too many samples from the budget n on **PHASE 1** can lead to regret which is $\Theta(n)$. On the other hand if the recovered subspace is a bad estimate then it can again lead to $\Theta(n)$ regret since the optimization carried out in **PHASE 2** would be rendered meaningless.

Hence it is important to carefully divide the sampling budget between the two phases in order to guarantee a regret bound that is sub-linear in n. We now describe these two phases in more detail and outline the above idea formally.

1. **PHASE 1**(Subspace recovery phase.) In this phase we use the first $n_1(< n)$ samples from our budget to generate an estimate $\widehat{\mathbf{A}} \in \mathbb{R}^{k \times d}$ of $\mathbf{A}$ such that the row space of $\widehat{\mathbf{A}}$ is close to that of $\mathbf{A}$. In particular we measure this closeness in terms of the Frobenius norm implying that we would like $\| \mathbf{A}^T\mathbf{A} - \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} \|_F$ to be sufficiently small. Denoting the total regret in this phase by R_1 we then have that:
$$R_1 = \sum_{t=1}^{n_1} \left[\bar{r}(\mathbf{x}^*) - \bar{r}(\mathbf{x}_t)\right] = O(n_1). \tag{21}$$
This follows trivially since $\bar{r}$ is a smooth function defined over a compact domain. We can see that n_1 should necessarily be $o(n)$ otherwise the total regret would be dominated by R_1 leading to linear regret. Furthermore $\mathbf{x}_t \in B_d(1+\nu)$ denotes the strategy played at time t.
2. **PHASE 2**(Optimization phase.) Say that we have in hand an estimate $\widehat{\mathbf{A}}$ from **PHASE 1**. We now employ a standard CAB algorithm that is restricted to play strategies from the row space of $\widehat{\mathbf{A}}$. Let us denote $n_2 = n - n_1$ to be the duration

of this phase and $\mathcal{P} \subset B_d(1+\nu)$ where

$$\mathcal{P} := \left\{ \widehat{\mathbf{A}}^T \mathbf{y} \in \mathbb{R}^d : \mathbf{y} \in B_k(1+\nu) \right\}.$$

The CAB algorithm will play strategies only from $\mathcal{P}$ and therefore will strive to optimize against the optimal strategy $\mathbf{x}^{**} = \widehat{\mathbf{A}}^T \mathbf{y}^{**} \in \mathcal{P}$ where

$$\mathbf{y}^{**} \in \underset{\mathbf{y} \in B_k(1+\nu)}{\operatorname{argmax}} \; \bar{g}(\mathbf{A}\widehat{\mathbf{A}}^T \mathbf{y}).$$

Furthermore we also observe that the total regret incurred in this phase can be written as:

$$\sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^*) - \bar{r}(\mathbf{x}_t)] = \underbrace{\sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^*) - \bar{r}(\mathbf{x}^{**})]}_{=R_3} + \underbrace{\sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^{**}) - \bar{r}(\mathbf{x}_t)]}_{=R_2}. \tag{22}$$

Note that R_2 represents the expected regret incurred by the CAB algorithm against the optimal strategy from $\mathcal{P}$. In particular, we will obtain $R_2 = o(n-n_1)$.

Next, the term R_3 captures the offset between the actual optimal strategy $\mathbf{x}^* \in B_d(1+\nu)$ and $\mathbf{x}^{**} \in \mathcal{P}$. In particular R_3 can be bounded by making use of: (i) the Lipschitz continuity of the mean reward $\bar{g}$ and, (ii) the bound on the subspace estimation error : $\| \mathbf{A}^T\mathbf{A} - \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} \|_F$. This is shown precisely in the form of the following Lemma the proof of which is presented in the Appendix.

Lemma 3 *We have that* $R_3 \leq \frac{n_2 C_2 \sqrt{k}(1+\nu)}{\sqrt{2}} \| \mathbf{A}^T\mathbf{A} - \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} \|_F$ *where* $n_2 = n - n_1$ *and* $C_2 > 0$ *is the constant defined in* (9).

We now provide a thorough analysis of the two phase scheme discussed in the previous section. We start by first describing a low-rank matrix recovery scheme which is used for obtaining an estimate of the unknown subspace represented by the row-space of $\mathbf{A}$.

4.1 Analysis of Sub-Space Recovery Phase

We first observe that the Taylor expansion of $\bar{r}$ around any $\mathbf{x} \in B_d(1+\nu)$ along the direction $\phi \in \mathbb{R}^d$ give us:

$$\bar{r}(\mathbf{x}+\epsilon\phi) - \bar{r}(\mathbf{x}) = \epsilon\langle\phi, \nabla\bar{r}(\mathbf{x})\rangle + \frac{1}{2}\epsilon^2\phi^T \nabla^2 \bar{r}(\xi)\phi \tag{23}$$

for any $\epsilon > 0$ and $\xi = \mathbf{x} + \theta\epsilon\phi$ with $0 < \theta < 1$. In particular by using $\nabla\bar{r}(\mathbf{x}) = \mathbf{A}^T \nabla \bar{g}(\mathbf{A}\mathbf{x})$ in (23) we obtain:

$$\langle\phi, \mathbf{A}^T \nabla \bar{g}(\mathbf{A}\mathbf{x})\rangle = \frac{\bar{r}(\mathbf{x}+\epsilon\phi) - \bar{r}(\mathbf{x})}{\epsilon} - \frac{1}{2}\epsilon\phi^T \nabla^2 \bar{r}(\xi)\phi. \tag{24}$$

We now introduce the sampling scheme[9] by stating the choice of $\mathbf{x}$ and sampling direction ϕ in (24). We first construct

$$\mathcal{X} := \left\{ \mathbf{x}_j \in \mathbb{S}^{d-1} \ : \ j = 1, \ldots, m_{\mathcal{X}} \right\}. \tag{25}$$

This is the set of samples at which we consider the Taylor expansion of $\bar{r}$ as in (23). In particular, we form $\mathcal{X}$ by sampling points uniformly at random from $\mathbb{S}^{d-1}$. Next, we construct the set of sampling directions Φ for $i = 1, \ldots, m_\Phi$, $j = 1, \ldots, m_{\mathcal{X}}$ and $l = 1, \ldots, d$ where:

$$\Phi := \left\{ \phi_{i,j} \in B_d(\sqrt{d/m_\Phi}) : [\phi_{i,j}]_l = \pm \frac{1}{\sqrt{m_\Phi}} \text{ with probability } 1/2 \right\}. \tag{26}$$

Note that we consider m_Φ random sampling directions *for each* point in $\mathcal{X}$. Hence we have that the total number of samples collected so far is

$$|\mathcal{X}| + |\Phi| = m_{\mathcal{X}} + m_{\mathcal{X}} m_\Phi = m_{\mathcal{X}}(m_\Phi + 1).$$

Now note that at each time $1 \leq t \leq m_{\mathcal{X}}(m_\Phi + 1)$ upon choosing the strategy $\mathbf{x}_t$ we obtain the reward $r_t(\mathbf{x}_t) = \bar{r}(\mathbf{x}_t) + \eta_t$ where η_t is i.i.d Gaussian noise. Therefore by first sampling at points $\mathbf{x}_1, \ldots, \mathbf{x}_{m_{\mathcal{X}}} \in \mathcal{X}$ and then sampling at $\mathbf{x}_j + \epsilon\phi_{1,j}, \ldots, \mathbf{x}_j + \epsilon\phi_{m_\Phi,j}$ for each $\mathbf{x}_j$, we have from (24) the following for $i = 1, \ldots, m_\Phi$ and $j = 1, \ldots, m_{\mathcal{X}}$.

$$\langle \phi_{i,j}, \mathbf{A}^T \bigtriangledown \bar{g}(\mathbf{A}\mathbf{x}_j) \rangle = \frac{r_{m_{\mathcal{X}}+ij}(\mathbf{x}_j + \epsilon\phi_{i,j}) - r_j(\mathbf{x}_j)}{\epsilon} + \frac{\eta_j - \eta_{i,j}}{\epsilon} - \frac{1}{2}\epsilon\phi_{i,j}^T \bigtriangledown^2 \bar{r}(\xi_{i,j})\phi_{i,j}. \tag{27}$$

We sum up (27) over all j for each $i = 1, \ldots, m_\Phi$. This yields m_Φ equations that can be summarized in the following succinct form:

$$\Phi(\mathbf{X}) = \mathbf{y} + \mathbf{N} + \mathbf{H}. \tag{28}$$

We now describe each term occuring in (28). Here $\mathbf{X} = \mathbf{A}^T\mathbf{G}$ where $\mathbf{G} := \left[\bigtriangledown\bar{g}(\mathbf{A}\mathbf{x}_1) | \bigtriangledown \bar{g}(\mathbf{A}\mathbf{x}_2) | \cdots | \bigtriangledown \bar{g}(\mathbf{A}\mathbf{x}_{m_{\mathcal{X}}})\right]_{k \times m_{\mathcal{X}}}$. Note that $\mathbf{X} \in \mathbb{R}^{d \times m_{\mathcal{X}}}$ has rank at most k. Next, $\Phi(\mathbf{X}) := [\langle \Phi_1 \mathbf{X} \rangle, \ldots, \langle \Phi_{m_\Phi} \mathbf{X} \rangle] \in \mathbb{R}^{m_\Phi}$ where

$$\Phi_i = [\phi_{i,1}\phi_{i,2} \ldots \phi_{i,m_{\mathcal{X}}}] \in \mathbb{R}^{d \times m_{\mathcal{X}}} \tag{29}$$

represents the i^{th} measurement matrix and $\langle \Phi_i, \mathbf{X} \rangle = \text{Tr}(\Phi_i^T \mathbf{X})$ represents the i^{th} measurement of $\mathbf{X}$. The measurement vector is represented by $\mathbf{y} = [y_1 \ldots y_{m_\Phi}] \in \mathbb{R}^{m_\Phi}$ where

$$y_i = \frac{1}{\epsilon} \sum_{j=1}^{m_{\mathcal{X}}} \left(r_{m_{\mathcal{X}}+ij}(\mathbf{x}_j + \epsilon\phi_{i,j}) - r_j(\mathbf{x}_j) \right). \tag{30}$$

[9] The above sampling scheme was first considered by Fornasier et al. [22], and later by Tyagi et al. [40], for the problem of approximating functions of the form $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ from point queries.

Lastly $\mathbf{N} = [N_1 \ldots N_{m_\Phi}]$ and $\mathbf{H} = [H_1 \ldots H_{m_\Phi}]$ represent the noise terms with

$$N_i = \frac{1}{\epsilon} \sum_{j=1}^{m_{\mathcal{X}}} (\eta_j - \eta_{i,j}) \quad \text{(Stochastic noise)},$$

$$H_i = -\frac{\epsilon}{2} \sum_{j=1}^{m_{\mathcal{X}}} \phi_{i,j}^T \nabla^2 \bar{r}(\xi_{i,j}) \phi_{i,j} \quad \text{(Noise due to non-linearity of } \bar{r}).$$

Importantly, we observe that (28) represents (noisy) linear measurements of the matrix $\mathbf{X}$ which has rank $k \ll d$. Hence by employing a standard solver for recovering low-rank matrices from noisy linear measurements, we can hope to recover an approximation $\widehat{\mathbf{X}}$ to the unknown matrix $\mathbf{X}$. Furthermore we note that information about the linear parameter matrix $\mathbf{A}$ is encoded in $\mathbf{X}$. This intuitively suggests that one can hope to recover an approximation to $\mathbf{A}$ with the help of $\widehat{\mathbf{X}}$. In particular the closer $\widehat{\mathbf{X}}$ is to $\mathbf{X}$ the better will be the approximation to the row space of $\mathbf{A}$. We now proceed to demonstrate this formally.

4.1.1 Low-Rank Matrix Recovery

As discussed, (28) represents noisy measurements of the low rank matrix $\mathbf{X}$ with the linear operator Φ. An important property of Φ is that it satisfies the so called Restricted Isometry Property (RIP) for low-rank matrices. This means that for all matrices $\mathbf{X}_k$ of rank at most k:

$$(1 - \delta_k) \parallel \mathbf{X}_k \parallel_F^2 \leq \parallel \Phi(\mathbf{X}_k) \parallel_2^2 \leq (1 + \delta_k) \parallel \mathbf{X}_k \parallel_F^2 \tag{31}$$

holds true for some isometry constant $\delta_k \in (0, 1)$. In general, any Φ that satisfies (31) is said to have δ_k-RIP. In our case since Φ is a Bernoulli random measurement operator, it can be verified via standard covering arguments and concentration inequalities [31, 38] that Φ satisfies δ-RIP for $0 < \delta_k < \delta < 1$ with probability at least $1 - 2\exp(-m_\Phi q(\delta) + k(d + m_{\mathcal{X}} + 1)u(\delta))$ where

$$q(\delta) = \frac{1}{144}\left(\delta^2 - \frac{\delta^3}{9}\right), \quad u(\delta) = \log\left(\frac{36\sqrt{2}}{\delta}\right).$$

An estimate of the low-rank matrix $\mathbf{X}$ from the measurement vector $\mathbf{y}$ can be obtained through convex programming. For our purposes we consider the following nuclear norm minimization problem also known as the matrix Dantzig selector (DS) [14].

$$\widehat{\mathbf{X}}_{DS} = \operatorname{argmin} \parallel \mathbf{M} \parallel_* \ \text{ s.t. } \parallel \Phi^*(\mathbf{y} - \Phi(\mathbf{M})) \parallel \leq \lambda. \tag{32}$$

Here $\Phi^* : \mathbb{R}^{m_\Phi} \to \mathbb{R}^{d \times m_{\mathcal{X}}}$ denotes the adjoint of the linear operator $\Phi : \mathbb{R}^{d \times m_{\mathcal{X}}} \to \mathbb{R}^{m_\Phi}$. Furthermore for any matrix, $\parallel \cdot \parallel_*$ and $\parallel \cdot \parallel$ denote its nuclear norm (sum of singular values) and operator norm (largest singular value) respectively. By making use of the error bound for matrix DS [14], we obtain the following result on the performance of the matrix DS tuned to our problem setting. The proof is deferred to the Appendix.

Lemma 4 *Let $\widehat{\mathbf{X}}_{DS} \in \mathbb{R}^{d \times m_{\mathcal{X}}}$ denote the solution of (32) and let $\widehat{\mathbf{X}}_{DS}^{(k)}$ be the best rank-k approximation to $\widehat{\mathbf{X}}_{DS}$ in the sense of $\| \cdot \|_F$. Then for some constant $\gamma > 2\sqrt{\log 12}$, $0 < \delta_{4k} < \delta < \sqrt{2} - 1$ we have that*

$$\| \widehat{\mathbf{X}}_{DS}^{(k)} - \mathbf{X} \|_F \leq (C_0 k)^{1/2} \left(\frac{C_2 \epsilon d m_{\mathcal{X}} k^2}{\sqrt{m_\Phi}} + \frac{8\gamma\sigma\sqrt{m_{\mathcal{X}} m_\Phi m}}{\epsilon} \right) (1+\delta)^{1/2}$$

with probability at least $1 - 2\exp(-m_\Phi q(\delta) + 4k(d + m_{\mathcal{X}} + 1)u(\delta)) - 4\exp(-cm)$. *Here $m = \max\{d, m_{\mathcal{X}}\}$. Furthermore the constants $C_0, c > 0$ depend on δ and γ respectively.*

*4.1.2 Approximating Row-Space(**A**)*

Let's say we have[10] in hand $\widehat{\mathbf{X}}_{DS}^{(k)} \in \mathbb{R}^{d \times m_{\mathcal{X}}}$ as the best rank-k approximation of the solution to (32). We can now obtain an estimate $\widehat{\mathbf{A}}$ of row-space(**A**) by setting $\widehat{\mathbf{A}}^T$ to be equal to the $(d \times k)$ left singular vector matrix of $\widehat{\mathbf{X}}_{DS}^{(k)}$. The quality of this estimation as measured by $\| \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} - \mathbf{A}^T\mathbf{A} \|_F$ was quantified by Tyagi et al. [41, Lemma 2] for the noiseless case ($\sigma = 0$). We adapt this result to our setting ($\sigma > 0$) and state it below. The proof is presented in the Appendix.

Lemma 5 *For a fixed $0 < \rho < 1$, $m_{\mathcal{X}} \geq 1$, $m_\Phi < m_{\mathcal{X}} d$ let*

$$a_1 = C_2 d k^2, \quad b_1 = \frac{\sqrt{(1-\rho)\alpha}}{C_0^{1/2}(1+\delta)^{1/2}(\sqrt{k}+\sqrt{2})}.$$

For any $0 < f < 1$ we then have for the choice

$$\epsilon \in \left(\frac{f b_1 - \sqrt{f^2 b_1^2 - 32\gamma\sigma a_1 \sqrt{m_{\mathcal{X}} m}}}{2a_1\sqrt{m_{\mathcal{X}}/m_\Phi}}, \frac{f b_1 + \sqrt{f^2 b_1^2 - 32\gamma\sigma a_1 \sqrt{m_{\mathcal{X}} m}}}{2a_1\sqrt{m_{\mathcal{X}}/m_\Phi}} \right) \tag{33}$$

that $\| \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} - \mathbf{A}^T\mathbf{A} \|_F \leq \frac{2f}{1-f}$ holds true with probability at least

$$1 - 2\exp(-m_\Phi q(\delta) + 4k(d + m_{\mathcal{X}} + 1)u(\delta)) - 4\exp(-cm) - k\exp\left(-\frac{m_{\mathcal{X}}\alpha\rho^2}{2kC_2^2}\right).$$

We see in the above lemma that the step size parameter ϵ cannot be chosen to be arbitrarily small.[11] In particular for ϵ too small the stochastic noise will become prominent while for large ϵ, the noise due to higher order Taylor's terms of the mean reward function will start to dominate.

[10] Of course in practice we will not be able to solve (32) exactly, but will instead obtain a solution that can be made to come arbitrarily close to the actual solution. This difference will hence appear as an additional error term in the error bound of Lemma 4.

[11] In the absence of external stochastic noise (i.e. $\sigma = 0$) we can actually take ϵ to be arbitrarily small as shown by Tyagi et al. [41, Lemma 2]. This is also verified from (33), by plugging $\sigma = 0$.

4.1.3 Handling Stochastic Noise

A point of obvious concern in Lemma 5 is the condition required on the step size parameter ϵ in (33). This condition is well defined if $f^2 b_1^2 - 32\gamma\sigma a_1\sqrt{m_{\mathcal{X}} m} > 0$. This would not have been a problem in the noiseless case where $\sigma = 0$. A natural way to guarantee the well-posedness of (33) is by *re-sampling* and averaging the rewards at each of the sampling points. Indeed if we consider each sampling point to be re-sampled N times and then average the corresponding reward values, the variance of the stochastic noise will be reduced by a factor of N. By choosing a sufficiently large value of N, we can clearly ensure that $f^2 b_1^2 - 32\gamma\sigma a_1\sqrt{m_{\mathcal{X}} m} > 0$ holds true. This is made precise in the following proposition which also states a bound on the total regret R_1 suffered in this phase.

Proposition 1 *Say that we resample N times at each sampling point $\mathbf{x}_j \in \mathcal{X}$ and $\mathbf{x}_j + \epsilon\phi_{i,j}$; $i = 1, \ldots, m_\Phi$ and $j = 1, \ldots, m_{\mathcal{X}}$. Let the reward value at each sampling point be estimated as the average of the N values. If $N > \frac{C' k^6 d^2 \sigma^2 m_{\mathcal{X}} m}{f^4 \alpha^2}$ for some constant $C' > 0$ (depending on $\rho, C_0, \delta, C_2, \gamma$) and with $m = \max\left\{d, m_{\max\{d, m_{\mathcal{X}}\}}\right\}$, then (33) in Lemma 5 is well defined. Consequently the total regret in* ***PHASE 1*** *is*

$$R_1 = O(n_1) = O(N m_{\mathcal{X}}(m_\Phi + 1)) = O\left(\frac{k^6 d^2 \sigma^2}{\alpha^2} \frac{m_{\mathcal{X}}^2 m_\Phi m}{f^4}\right).$$

Proof First note that (33) in Lemma 5 is well defined when

$$f^2 b_1^2 - 32\gamma\sigma a_1\sqrt{m_{\mathcal{X}} m} > 0 \Leftrightarrow \sigma < \frac{f^2 b_1^2}{32\gamma\sqrt{m_{\mathcal{X}} m}\,\underbrace{C_2 d k^2}_{a_1}}.$$

After plugging in the value of b_1 from Lemma 5 we then obtain

$$\sigma < \frac{f^2 b_1^2}{32\gamma\sqrt{m_{\mathcal{X}} m} C_2 d k^2} = \frac{C\alpha f^2}{(\sqrt{k} + \sqrt{2})^2 \sqrt{m_{\mathcal{X}} m} d k^2} \tag{34}$$

where $C = \frac{(1-\rho)}{32\gamma C_0(1+\delta)C_2}$ is a constant. Upon re-sampling N times and subsequent averaging of reward values we have that the variance σ changes to $\sigma/\sqrt{N}$. Replacing σ with $\sigma/\sqrt{N}$ in (34) we obtain the stated condition on N. Lastly, we note that as a consequence of re-sampling, the duration of **PHASE 1** i.e. n_1, is $N m_{\mathcal{X}}(m_\Phi + 1)$. This implies the stated bound on R_1. □

4.2 Analysis of Optimization Phase

We now analyze **PHASE 2** i.e. the optimization phase of our scheme. This phase runs during time steps $t = n_1 + 1, n_1 + 2, \ldots, n$ where $n_1 = N m_{\mathcal{X}}(m_\Phi + 1)$. Given an estimate $\widehat{\mathbf{A}}$ of the row space of $\mathbf{A}$ we now consider optimizing *only* over points lying in the row space of $\widehat{\mathbf{A}}$. In particular consider $\mathcal{P} \subset B_d(1+\nu)$ where

$$\mathcal{P} := \left\{ \widehat{\mathbf{A}}^T \mathbf{y} \in \mathbb{R}^d : \mathbf{y} \in B_k(1+\nu) \right\}.$$

We employ a standard CAB algorithm that plays points only from $\mathcal{P}$ and therefore strives to optimize against the optimal strategy $\mathbf{x}^{**} = \widehat{\mathbf{A}}^T \mathbf{y}^{**} \in \mathcal{P}$ where

$$\mathbf{y}^{**} \in \underset{\mathbf{y} \in B_k(1+\nu)}{\operatorname{argmax}} \ \bar{g}(\mathbf{A}\widehat{\mathbf{A}}^T \mathbf{y}).$$

Recall from (22) that the total regret incurred in this phase can be written as:

$$\sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^*) - \bar{r}(\mathbf{x}_t)] = \underbrace{\sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^*) - \bar{r}(\mathbf{x}^{**})]}_{=R_3} + \underbrace{\sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^{**}) - \bar{r}(\mathbf{x}_t)]}_{=R_2}$$

where R_2 is the regret incurred by the CAB algorithm and R_3 is the regret incurred on account of not playing strategies from the row space of $\mathbf{A}$.

Bounding R_2 In order to bound R_2 we employ the CAB1 algorithm [26], with the UCB-1 algorithm [5] as the finite armed bandit algorithm. Recall that this phase runs for a duration of $n_2 = n - n_1$ time steps. A straightforward generalization of the result by Kleinberg [26, Theorem 3.1] to k dimensions then yields

$$R_2 = O(n_2^{\frac{1+k}{2+k}} (\log n_2)^{\frac{1}{2+k}}) = O(n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}). \tag{35}$$

Indeed for any integer $M > 0$, we simply discretize $[-1-\nu, 1+\nu]^k$ into $(2M+1)^k$ points, with step size $1/M$ in each direction. We retain only those points that lie in $B_k(1+\nu)$ and multiply each of these with $\widehat{\mathbf{A}}^T$. This gives us a finite subset of $\mathcal{P}$ on which we employ the UCB-1 algorithm. Since the time duration n_2 is known, therefore in a manner similar to the proof of Lemma 1, one can find an optimal value of M, for which the regret bound of (35) is attained.

Bounding R_3 The term R_3 can be bounded from above by a straightforward combination of Lemma 3 with Lemma 5. Hence we state this in the form of the following proposition without proof.

Proposition 2 *For fixed* $0 < \rho < 1$, $m_{\mathcal{X}} \geq 1$, $m_{\Phi} < m_{\mathcal{X}} d$ *and* $0 < f < 1$, *let* ϵ *be chosen to satisfy* (33). *This then implies that* $R_3 \leq \frac{n_2 C_2 \sqrt{k}(1+\nu)\sqrt{2f}}{1-f}$ *holds true with probability at least*

$$1 - 2\exp(-m_{\Phi} q(\delta) + 4k(d + m_{\mathcal{X}} + 1)u(\delta)) - 4\exp(-cm) - k\exp\left(-\frac{m_{\mathcal{X}} \alpha \rho^2}{2kC_2^2}\right).$$

4.3 Bounding the Total Regret

Finally, we have all the results sufficient to bound the total regret. Indeed by using bounds on R_1, R_2, R_3 from Proposition 1, (35) and Proposition 2 respectively we have that:

$$R_1 + R_2 + R_3 = O\left(\frac{k^6 d^2 \sigma^2}{\alpha^2} \frac{m_{\mathcal{X}}^2 m_\Phi m}{f^4} + n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}} + n_2\sqrt{k} f\right). \tag{36}$$

holds with probability at least

$$1 - 2\exp(-m_\Phi q(\delta) + 4k(d + m_{\mathcal{X}} + 1)u(\delta)) - 4\exp(-cm) - k\exp\left(-\frac{m_{\mathcal{X}}\alpha\rho^2}{2kC_2^2}\right). \tag{37}$$

In order to bound the overall regret we need to choose the values of: $m_{\mathcal{X}}, m_\Phi$ and f carefully. We state these choices precisely in the following theorem which is also our main theorem that provides a bound on the overall regret achieved by our scheme.

Theorem 3 *Under the assumptions and notations used thus far let:*

$$f = \frac{1}{\sqrt{k}}\left(\frac{\log n}{n}\right)^{\frac{1}{k+2}}, m_{\mathcal{X}} = \frac{2kC_2^2}{\alpha\rho^2}\log(k/p) \quad \textit{and} \quad m_\Phi = \frac{4k(d + m_{\mathcal{X}} + 1)u(\delta)c_1}{q(\delta)}$$

for some $p \in (0, 1)$ and $c_1 > 1$. Then there exists a constant $c' > 0$ so that the total regret achieved by our scheme is bounded as:

$$R_1{+}R_2{+}R_3 = O\left(\frac{k^{13}d^2\sigma^2(\log(\frac{k}{p}))^4}{\alpha^4}(\max\left\{d, \alpha^{-1}\right\})^2\left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} + n^{\frac{1+k}{2+k}}(\log n)^{\frac{1}{2+k}}\right) \tag{38}$$

with probability at least $1 - p - 6\exp(-c'd)$.

Proof We first observe that when $f = \frac{1}{\sqrt{k}}\left(\frac{\log n}{n}\right)^{\frac{1}{k+2}}$ then this results in:

$$n_2\sqrt{k} f = O(n^{\frac{1+k}{2+k}}(\log n)^{\frac{1}{2+k}}).$$

Upon using this in (36) we obtain:

$$R_1 + R_2 + R_3 = O\left(\frac{k^8 d^2 \sigma^2}{\alpha^2}\left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} m_{\mathcal{X}}^2 m_\Phi m + n^{\frac{1+k}{2+k}}(\log n)^{\frac{1}{2+k}}\right) \tag{39}$$

In order to choose $m_{\mathcal{X}}$ and m_Φ we simply note from (37) that the choices

$$m_{\mathcal{X}} = \frac{2kC_2^2}{\alpha\rho^2}\log(k/p), \quad m_\Phi = \frac{4k(d + m_{\mathcal{X}} + 1)u(\delta)c_1}{q(\delta)} \tag{40}$$

for suitable constants $c_1 > 1, p \in (0, 1)$ ensure that the regret bound holds with probability at least $1 - p - 2\exp(-4u(\delta)k(d + m_{\mathcal{X}} + 1)(c_1 - 1)) - 4\exp(-cm)$.

Since $m = \max\{d, m_{\mathcal{X}}\} \geq d$ we obtain via a simple calculation, the stated lower bound on the probability of success. Then plugging the above choice of $m_{\mathcal{X}}$ and m_Φ in (39) and noting that $m = \max\{d, m_{\mathcal{X}}\} \leq (d + m_{\mathcal{X}})$ we obtain:

$$\begin{aligned} R_1 + R_2 + R_3 &= O\left(\frac{k^9 d^2 \sigma^2}{\alpha^2}\left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} m_{\mathcal{X}}^2 (d + m_{\mathcal{X}})^2 + n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}\right) \\ &= O\left(\frac{k^9 d^2 \sigma^2}{\alpha^2}\left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} (k^2 (\log(k/p))^2 \alpha^{-2})(d + k\alpha^{-1}\log(k/p))^2 + n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}\right) \\ &= O\left(\frac{k^{11} d^2 (\log(k/p))^2 \sigma^2}{\alpha^4}(d + k\alpha^{-1}\log(k/p))^2 \left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} + n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}\right) \\ &= O\left(\frac{k^{13} d^2 (\log(k/p))^4 \sigma^2}{\alpha^4}(\max\left\{d, \alpha^{-1}\right\})^2 \left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} + n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}\right). \end{aligned}$$

□

Note that the regret bound in Theorem 3 can be made to hold with high probability. Indeed, the choice $p = d^{-\beta}$, for any constant $\beta > 0$ guarantees that the bound holds with probability approaching one, as $d \to \infty$. Furthermore, this choice of p, leads to an additional $O((\log d)^4)$ factor in the regret bound.

Remark 3 Upon examining the regret bound in Theorem 3, we notice that the $n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}$ term dominates for $k \geq 3$, while for $k = 1, 2$ we have that $\left(\frac{n}{\log n}\right)^{\frac{4}{k+2}}$ dominates. Therefore, the bound is fairly sub-optimal for the cases $k = 1, 2$. This appears to be an artifact of the analysis. While it is unclear how this can be fixed, there are ways to improve the regret bound for $k = 1, 2$. For instance in Theorem 3, let us only change the choice of f to $f = \frac{1}{\sqrt{k}}\left(\frac{\log n}{n}\right)^{\frac{0.5}{k+2}}$. By following the steps in the proof, one can then verify that the regret is bounded by:

$$O\left(\frac{k^{13} d^2 \sigma^2 (\log(\frac{k}{p}))^4}{\alpha^4}(\max\left\{d, \alpha^{-1}\right\})^2 \left(\frac{n}{\log n}\right)^{\frac{2}{k+2}} + n^{\frac{1.5+k}{2+k}} (\log n)^{\frac{0.5}{2+k}}\right).$$

We see that the $n^{\frac{1.5+k}{2+k}} (\log n)^{\frac{0.5}{2+k}}$ term, now dominates for $k \geq 1$. This is also only slightly worse than the $n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}$ term, appearing in (38). Hence for $k = 1, 2$, we can substantially improve the bound in (38), with a different choice of parameter f in Theorem 3.

Our complete scheme which we name CAB-LP(d, k) (Continuum armed bandit of k linear parameters in d dimensions) is presented as Algorithm 2.

Algorithm 2 Algorithm CAB-LP(d, k)

```
Input: $k, d, n, C_2, \sigma$.
Choose $0 < \delta < \sqrt{2}-1$; $\rho, p \in (0,1)$ and $c_1 > 1$. Choose $\alpha$ according to model assumption on mean reward function.
Set $f = \frac{1}{\sqrt{k}}\left(\frac{\log n}{n}\right)^{\frac{1}{k+2}}$, $m_{\mathcal{X}} = \frac{2kC_2^2}{\alpha\rho^2}\log(k/p)$ and $m_{\Phi} = \frac{4k(d+m_{\mathcal{X}}+1)u(\delta)c_1}{q(\delta)}$.
Choose re-sampling factor $N$ according to Proposition 1.
Choose step size $\epsilon$ as in (33) with $\sigma \leftarrow \sigma/\sqrt{N}$.
PHASE 1 (Subspace recovery phase) $t = 1, \ldots, Nm_{\mathcal{X}}(m_{\Phi}+1)$
    – Create random sampling sets $\mathcal{X}$ and $\Phi$ as explained in Section 4.1 so that $|\mathcal{X}| = m_{\mathcal{X}}$ and $|\Phi| = m_{\mathcal{X}}m_{\Phi}$.
    – For $t = 1, \ldots, m_{\mathcal{X}}(m_{\Phi}+1)$ collect rewards $(r_j(\mathbf{x}_j))_{j=1}^{m_{\mathcal{X}}}$ and $(r_{m_{\mathcal{X}}+ij}(\mathbf{x}_j + \epsilon\phi_{i,j}))_{j=1,i=1}^{m_{\mathcal{X}},m_{\Phi}}$.
    – Re-sample and average the reward values $N$ times at each $\mathbf{x}$ and $\mathbf{x}+\epsilon\phi$ respectively ($\mathbf{x} \in \mathcal{X}, \phi \in \Phi$). Form measurement vector $\mathbf{y}$ as in (30) with the averaged reward values.
    – Obtain $\widehat{\mathbf{X}}_{DS}^{(k)}$ as best rank-$k$ approximation to solution of matrix DS (32) and set $\widehat{\mathbf{A}}^T$ to left singular vector matrix of $\widehat{\mathbf{X}}_{DS}^{(k)}$.
PHASE 2 (Optimization phase) $t = Nm_{\mathcal{X}}(m_{\Phi}+1)+1, \ldots, n$
    – Employ CAB1 algorithm [26] on $\mathcal{P} := \left\{\widehat{\mathbf{A}}^T\mathbf{y} \in \mathbb{R}^d : \mathbf{y} \in B_k(1+\nu)\right\}$.
```

4.4 Remarks on the Tractability Parameter α

We now proceed to comment on the parameter α of our scheme which also appears in our regret bounds. Recall from Section 2 that α measures the conditioning of the following matrix:

$$H^r := \int_{\mathbb{S}^{d-1}} \nabla\bar{r}(\mathbf{x})\nabla\bar{r}(\mathbf{x})^T d\mathbf{x} = \mathbf{A}^T \cdot \int_{\mathbb{S}^{d-1}} \nabla\bar{g}(\mathbf{A}\mathbf{x})\nabla\bar{g}(\mathbf{A}\mathbf{x})^T d\mathbf{x} \cdot \mathbf{A}. \tag{41}$$

More specifically, we assume that the mean reward function $\bar{r}$ is such that:

$$\sigma_1(H^r) \geq \sigma_2(H^r) \geq \cdots \geq \sigma_k(H^r) \geq \alpha > 0 \tag{42}$$

where $\sigma_i(H^r)$ denotes the i^{th} singular value of H^r. In other words α measures how far away from 0 the lowest singular value of H^r is, implying that a larger α indicates a well conditioned H^r. A natural question that arises now is on the behaviour of α - in particular on its dependence on dimension d and number of linear parameters k. To this end we first note that the parameter typically decays with increase in d. In fact for $k > 1$ this would always be the case since as $d \to \infty$ the matrix H^r would converge to a rank-1 matrix [41].

We also note from our derived regret bounds that in case $\alpha \to 0$ exponentially fast as $d \to \infty$ then our regret bounds will have a factor exponential in d which is clearly undesirable. Hence it is important to define classes of functions for which α provably decays polynomially as $d \to \infty$ so that our regret bounds depend *at most polynomially* on dimension d. We now state the following result by Tyagi et al. [41] which defines such a class of functions for which $\alpha = \Theta(d^{-1})$.

Proposition 3 ([41]) *Assume that $g : B_k(1) \to \mathbb{R}$, with g being a $\mathcal{C}^2$ function, has Lipschitz continuous second order partial derivatives in an open neighborhood of the* *origin,* $\mathcal{U}_\theta = B_k(\theta)$ *for some fixed* $0 < \theta < 1$*,*

$$\frac{|\frac{\partial^2 g}{\partial y_i \partial y_j}(\mathbf{y}_1) - \frac{\partial^2 g}{\partial y_i \partial y_j}(\mathbf{y}_2)|}{\| \mathbf{y}_1 - \mathbf{y}_2 \|} < L_{i,j} \quad \forall \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{U}_\theta, \mathbf{y}_1 \neq \mathbf{y}_2,\ i, j = 1, \ldots, k.$$

Denote $L = \max_{1 \leq i,j \leq k} L_{i,j}$*. Also under the notation defined earlier, assume that* $\nabla^2 g(\mathbf{0})$ *is full rank. Then provided that* $\frac{\partial g}{\partial y_i}(\mathbf{0}) = 0$*;* $\forall i = 1, \ldots, k$ *we have* $\alpha = \Theta(1/d)$ *as* $d \to \infty$*.*

The class of functions defined in the above Proposition covers a number of function models such as sparse additive models of the form $\sum_{i=1}^{k} g_i(\mathbf{y})$ where g_i's are kernel functions [32]. Further details in this regard are provided by Tyagi et al. [41, Section 5]. Finally, in light of the above discussion on α we arrive at the following Corollary of Theorem 3 with the help of Proposition 3.

Corollary 1 *Assume that the mean reward function* $\bar{r} : B_d(1 + \nu) \to \mathbb{R}$ *where* $\bar{r}(\mathbf{x}) = \bar{g}(\mathbf{Ax})$ *is such that* $\bar{g}$ *satisfies the conditions of Proposition 3. Then there exists a constant* $c' > 0$ *so that the total regret achieved by Algorithm CAB-LP(d,k) is bounded as:*

$$R_1 + R_2 + R_3 = O\left(k^{13} d^8 \sigma^2 (\log(k/p))^4 \left(\frac{n}{\log n}\right)^{\frac{4}{k+2}} + n^{\frac{1+k}{2+k}} (\log n)^{\frac{1}{2+k}}\right), \quad (43)$$

with probability at least $1 - p - 6\exp(-c'd)$*.*

5 Concluding Remarks

In this work we considered the problem of online optimization in the bandit setting with the reward functions residing in a high dimensional space. We handled the notorious curse of dimensionality typically associated with this setting by considering two different intrinsic low-dimensional models for the reward functions.

In the first model we assumed that the reward function $r : [0, 1]^d \to \mathbb{R}$ intrinsically depends at each time t on an unknown subset of k out of the d coordinate variables. We proposed an algorithm and proved upper bounds on the regret, both for the setting when the active k coordinates remain fixed across time and also for the more general scenario when they can change over time. There are several interesting lines of future work for this model. Firstly for the case when $(i_1, \ldots, i_k)$ is fixed across time it would be interesting to investigate whether the dependence of regret on k and dimension d achieved by our algorithm, is optimal or not. Secondly, for the case when $(i_1, \ldots, i_k)$ can also change with time, it would be interesting to derive lower bounds on regret to know what the optimal dependence on the hardness of the sequence of k tuples is.

The second model we considered was a generalization of the first in the sense that we assumed the reward function at each time t to intrinsically depend on k-linear combinations of the d coordinate variables. Assuming the time horizon n to be known we derived a randomized algorithm, and proved an upper bound on the regret incurred. Our algorithm combines results from low rank matrix recovery literature, with existing results on continuum armed bandits. For future work it would be interesting to consider the setting where the time horizon n is unknown to the algorithm and to prove regret bounds for the same. In particular, it would be interesting to derive algorithms which do not involve recovering an approximation of the unknown k dimensional subspace spanned by the k linear parameters. Lastly we mention other directions such as an adversarial version of our problem where the reward functions are chosen arbitrarily by an adversary and also a setting where the unknown matrix $\mathbf{A}$ is allowed to change across time.

Acknowledgments The project CG Learning acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 255827.

(A preliminary version of this paper appeared in the proceedings of the 11^{th} Workshop on Approximation Online Algorithms (WAOA). This is a significantly expanded version including analysis for a generalization of the problem considered in the WAOA paper.)

Appendix: Proofs of Results in Section 4

A.1 Proof of Lemma 3

Proof We can bound R_3 from above as follows.

$$\begin{aligned}
R_3 &= \sum_{t=n_1+1}^{n} [\bar{r}(\mathbf{x}^*) - \bar{r}(\mathbf{x}^{**})] && (44)\\
&= n_2[\bar{g}(\mathbf{A}\mathbf{x}^*) - \bar{g}(\mathbf{A}\mathbf{x}^{**})] && (45)\\
&= n_2[\bar{g}(\mathbf{A}\mathbf{x}^*) - \bar{g}(\mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^{**})] && (46)\\
&\leq n_2[\bar{g}(\mathbf{A}\mathbf{x}^*) - \bar{g}(\mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^{*})] && (47)\\
&\leq n_2 C_2\sqrt{k} \parallel \mathbf{A}\mathbf{x}^* - \mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^* \parallel && (48)\\
&\leq n_2 C_2\sqrt{k}(1+\nu) \parallel \mathbf{A} - \mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}} \parallel_F && (49)\\
&= \frac{n_2 C_2\sqrt{k}(1+\nu)}{\sqrt{2}} \parallel \mathbf{A}^T\mathbf{A} - \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} \parallel_F . && (50)
\end{aligned}$$

In (46) we used the fact that $\mathbf{x}^{**} = \widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^{**}$ since $\mathbf{x}^{**} \in \mathcal{P}$. In (47) we used the fact that $\bar{g}(\mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^{**}) \geq \bar{g}(\mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^{*})$ since $\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}^* \in \mathcal{P}$ and $\mathbf{x}^{**} \in \mathcal{P}$ is an optimal strategy. Equation (48) follows from the mean value theorem along with the smoothness assumption made in (9). In (49) we used the simple inequality : $\parallel \mathbf{B}\mathbf{x} \parallel \leq \parallel \mathbf{B} \parallel_F \parallel \mathbf{x} \parallel$. Obtaining (50) from (49) is a straightforward exercise. □

A.2 Proof of Lemma 4

Proof We first recall the following result by Candes et al. [14, Theorem 1], which we will use in our setting, for bounding the error of the matrix Dantzig selector.

Theorem 4 *For any* $\mathbf{X} \in \mathbb{R}^{d \times m_{\mathcal{X}}}$ *such that rank*$(\mathbf{X}) \leq k$ *let* $\widehat{\mathbf{X}}_{DS}$ *be the solution of* (32). *If* $\delta_{4k} < \delta < \sqrt{2} - 1$ *and* $\| \Phi^*(\mathbf{H}+\mathbf{N}) \| \leq \lambda$ *then we have with probability at least* $1 - 2e^{-m_\Phi q(\delta)+4k(d+m_{\mathcal{X}}+1)u(\delta)}$ *that*

$$\| \mathbf{X} - \widehat{\mathbf{X}}_{DS} \|_F^2 \leq C_0 k \lambda^2$$

where C_0 *depends only on the isometry constant* δ_{4k}. □

What remains to be found for our purposes is λ which is a bound on $\| \Phi^*(\mathbf{H}+\mathbf{N}) \|$. Firstly note that $\| \Phi^*(\mathbf{H}+\mathbf{N}) \| \leq \| \Phi^*(\mathbf{H}) \| + \| \Phi^*(\mathbf{N}) \|$. From Tyagi et al. [41, Lemma 1,Corollary 1], we have that:

$$\| \Phi^*(\mathbf{H}) \| \leq \frac{C_2 \epsilon d m_{\mathcal{X}} k^2}{2\sqrt{m_\Phi}} (1+\delta)^{1/2}$$

holds with probability at least $1 - 2e^{-m_\Phi q(\delta)+4k(d+m_{\mathcal{X}}+1)u(\delta)}$ where δ is such that $\delta_{4k} < \delta < \sqrt{2} - 1$. Next we note that $\mathbf{N} = [N_1 N_2 \ldots N_{m_\Phi}]$ where

$$N_i = \underbrace{\frac{1}{\epsilon}\sum_{j=1}^{m_{\mathcal{X}}} \eta_j}_{L_{1,i}} - \underbrace{\frac{1}{\epsilon}\sum_{j=1}^{m_{\mathcal{X}}} \eta_{i,j}}_{L_{2,i}}$$

with $\mathbf{L_1} = [L_{1,1} \ldots L_{1,m_\Phi}]$ and $\mathbf{L_2} = [L_{2,1} \ldots L_{2,m_\Phi}]$ so that $\mathbf{N} = \mathbf{L_1} - \mathbf{L_2}$. We then have that $\| \Phi^*(\mathbf{N}) \| \leq \| \Phi^*(\mathbf{L_1}) \| + \| \Phi^*(\mathbf{L_2}) \|$. By using Lemma 1.1 of Candes et al. [14] and denoting $m = \max\{d, m_{\mathcal{X}}\}$ we first have that:

$$\| \Phi^*(\mathbf{L_1}) \| \leq \frac{2\gamma\sigma}{\epsilon}\sqrt{(1+\delta) m_\Phi m_{\mathcal{X}} m} \tag{51}$$

holds with probability at least $1 - 2e^{-cm}$ where $c = \frac{\gamma^2}{2} - 2\log 12$ and $\gamma > 2\sqrt{\log 12}$. This can be verified using the proof technique of Candes et al. [14, Lemma 1.1]. Care has to be taken of the fact that the entries of $\mathbf{L}_1$ are correlated as they are identical copies of the same Gaussian random variable $\frac{1}{\epsilon}\sum_{j=1}^{m_{\mathcal{X}}} \eta_j$. Furthermore we also have that:

$$\| \Phi^*(\mathbf{L_2}) \| \leq \frac{2\gamma\sigma}{\epsilon}\sqrt{(1+\delta) m_{\mathcal{X}} m} \tag{52}$$

holds with probability at least $1 - 2e^{-cm}$ with constants c, γ as defined earlier. This is again easily verifiable using the proof technique of Candes et al. [14, Lemma 1.1], as the entries of $\mathbf{L}_2$ are i.i.d Gaussian random variables. Combining (51) and (52) we then have that the following holds true with probability at least $1 - 4e^{-cm}$.

$$\| \Phi^*(\mathbf{L_1}) \| + \| \Phi^*(\mathbf{L_2}) \| \leq \frac{4\gamma\sigma}{\epsilon}\sqrt{(1+\delta) m_{\mathcal{X}} m_\Phi m}. \tag{53}$$

Lastly, it is fairly easy to see that $\| \widehat{\mathbf{X}}_{DS}^{(k)} - \mathbf{X} \|_F \leq 2 \| \widehat{\mathbf{X}}_{DS} - \mathbf{X} \|_F$ where $\widehat{\mathbf{X}}_{DS}^{(k)}$ is the best rank k approximation to $\widehat{\mathbf{X}}_{DS}$ (see for example, the proof by Tyagi et al. [41, Corollary 1]). Combining the above observations we arrive at the stated error bound with probability at least $1 - 2e^{-m_\Phi q(\delta)+4k(d+m_{\mathcal{X}}+1)u(\delta)} - 4e^{-cm}$.

A.3 Proof of Lemma 5

Proof Let τ denote the bound on $\parallel \widehat{\mathbf{X}}_{DS}^{(k)} - \mathbf{X} \parallel_F$ as stated in Lemma 4. We now make use of a result by Tyagi et al. [41, Lemma 2]. This states that if $\tau < \frac{\sqrt{(1-\rho)m_{\mathcal{X}}\alpha k}}{\sqrt{k}+\sqrt{2}}$ holds, then it implies that

$$\parallel \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} - \mathbf{A}^T\mathbf{A} \parallel_F \leq \frac{2\tau}{\sqrt{(1-\rho)m_{\mathcal{X}}\alpha} - \tau} \tag{54}$$

holds true for any $0 < \rho < 1$ with probability at least $1 - k\exp\left(-\frac{m_{\mathcal{X}}\alpha\rho^2}{2kC_2^2}\right)$. The proof makes use of Weyl's inequality [45], Wedin's perturbation bound [44] and a deviation bound for the extremal eigenvalues of the sum of random positive semidefinite matrices [39].

Therefore, upon using the value of τ we have that $\tau < f\frac{\sqrt{(1-\rho)m_{\mathcal{X}}\alpha k}}{\sqrt{k}+\sqrt{2}}$ holds for any $0 < f < 1$ if:

$$C_0^{1/2}k^{1/2}(1+\delta)^{1/2}\left(\frac{C_2\epsilon d m_{\mathcal{X}}k^2}{\sqrt{m_\Phi}} + \frac{8\gamma\sigma\sqrt{m_{\mathcal{X}}m_\Phi m}}{\epsilon}\right) < f\frac{\sqrt{(1-\rho)m_{\mathcal{X}}\alpha k}}{\sqrt{k}+\sqrt{2}} \tag{55}$$

$$\Leftrightarrow \overbrace{C_2dk^2}^{a_1}\epsilon\sqrt{\frac{m_{\mathcal{X}}}{m_\Phi}} + \frac{8\gamma\sigma\sqrt{m_\Phi m}}{\epsilon} < f\left(\overbrace{\frac{1}{C_0^{1/2}(1+\delta)^{1/2}}\frac{\sqrt{(1-\rho)\alpha}}{\sqrt{k}+\sqrt{2}}}^{b_1}\right) \tag{56}$$

$$\Leftrightarrow a_1\sqrt{\frac{m_{\mathcal{X}}}{m_\Phi}}\epsilon^2 - fb_1\epsilon + 8\gamma\sigma\sqrt{m_\Phi m} < 0. \tag{57}$$

From (57) we get the stated condition on ϵ. Lastly upon using $\tau < \frac{f\sqrt{(1-\rho)m_{\mathcal{X}}\alpha k}}{\sqrt{k}+\sqrt{2}}$ in (54) we obtain the stated bound on $\parallel \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} - \mathbf{A}^T\mathbf{A} \parallel_F$. □

References

1. Abbasi-yadkori, Y., Pal, D., Szepesvari, C.: Online-to-confidence-set conversions and application to sparse stochastic bandits. In: Proceedings of AIStats (2012)
2. Abernethy, J., Hazan, E., Rakhlin, A.: Competing in the dark: An efficient algorithm for bandit linear optimization. In: Proceedings of the 21st Annual Conference on Learning Theory (COLT) (2008)
3. Agrawal, R.: The continuum-armed bandit problem. SIAM J. Control Optim. **33**, 1926–1951 (1995)
4. Audibert, J.Y., Bubeck, S.: Regret bounds and minimax policies under partial monitoring. J. Mach. Learn. Res. **11**, 2635–2686 (2010)
5. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Mach. Learn. **47**(2-3), 235–256 (2002)
6. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.: Gambling in a rigged casino: The adversarial multi-armed bandit problem. In: Proceedings of 36th Annual Symposium on Foundations of Computer Science, 1995, pp. 322–331 (1995)
7. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.: The nonstochastic multiarmed bandit problem. SIAM J. Comput. **32**(1), 48–77 (2003)
8. Auer, P., Ortner, R., Szepesvari, C.: Improved rates for the stochastic continuum-armed bandit problem. In: Proceedings of 20th Conference on Learning Theory (COLT), pp. 454–468 (2007)

9. Bansal, N., Blum, A., Chawla, S., Meyerson, A.: Online oblivious routing. In: Proceedings of ACM Symposium in Parallelism in Algorithms and Architectures, pp. 44–49 (2003)
10. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**, 1373–1396 (2003)
11. Blum, A., Kumar, V., Rudra, A., Wu, F.: Online learning in online auctions. In: Proceedings of 14th Symp. on Discrete Alg., pp. 202–204 (2003)
12. Bubeck, S., Munos, R., Stoltz, G., Szepesvari, C.: X-armed bandits. J. Mach. Learn. Res. (JMLR) **12**, 1587–1627 (2011)
13. Bubeck, S., Stoltz, G., Yu, J.: Lipschitz bandits without the Lipschitz constant. In: Proceedings of the 22nd International Conference on Algorithmic Learning Theory (ALT), pp. 144–158 (2011)
14. Candès, E., Plan, Y.: Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements. CoRR abs/1001.0339 (2010)
15. Carpentier, A., Munos, R.: Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In: Proceedings of AIStats, pp. 190–198 (2012)
16. Chen, B., Castro, R., Krause, A.: Joint optimization and variable selection of high-dimensional gaussian processes. In: Proceedings International Conference on Machine Learning (ICML) (2012)
17. Coifman, R., Maggioni, M.: Diffusion wavelets. Appl. Comput. Harmon. Anal. **21**, 53–94 (2006)
18. Cope, E.: Regret and convergence bounds for a class of continuum-armed bandit problems. IEEE Trans. Autom. Control **54**, 1243–1253 (2009)
19. DeVore, R., Petrova, G., Wojtaszczyk, P.: Approximation of functions of few variables in high dimensions. Constr. Approx **33**, 125–143 (2011)
20. Djolonga, J., Krause, A., Cevher, V.: High dimensional gaussian process bandits. In: To Appear in Neural Information Processing Systems (NIPS) (2013)
21. Flaxman, A., Kalai, A., McMahan, H.: Online convex optimization in the bandit setting: gradient descent without a gradient. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 385–394 (2005)
22. Fornasier, M., Schnass, K., Vybiral, J.: Learning functions of few arbitrary linear parameters in high dimensions. Found. Comput. Math. **12**(2), 229–262 (2012)
23. Fredman, M., Komlós, J.: On the size of separating systems and families of perfect hash functions. SIAM. J. Algebr. Discret. Methods **5**, 61–68 (1984)
24. Fredman, M., Komlós, J., Szemerédi, E.: Storing a sparse table with 0(1) worst case access time. J. ACM **31**(3), 538–544 (1984)
25. Greenshtein, E.: Best subset selection, persistence in high dimensional statistical learning and optimization under ℓ_1 constraint. Ann. Stat. **34**, 2367–2386 (2006)
26. Kleinberg, R.: Nearly tight bounds for the continuum-armed bandit problem. In: 18th Advances in Neural Information Processing Systems (2004)
27. Kleinberg, R.: Online decision problems with large strategy sets. Ph.D. thesis. MIT, Boston (2005)
28. Kleinberg, R., Leighton, T.: The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In: Proceedings of Foundations of Computer Science, 2003., pp. 594–605 (2003)
29. Kleinberg, R., Slivkins, A., Upfal, E.: Multi-armed bandits in metric spaces. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC '08, pp. 681–690 (2008)
30. Körner, J.: Fredmankomlós bounds and information theory. SIAM J. Algebraic Discret. Methods **7**(4), 560–570 (1986)
31. Laurent, B., Massart, P.: Adaptive estimation of a quadratic functional by model selection. Ann. Stat. **28**(5), 1302–1338 (2000)
32. Li, Q., Racine, J.: Nonparametric econometrics: Theory and practice (2007)
33. McMahan, B., Blum, A.: Online geometric optimization in the bandit setting against an adaptive adversary. In: Proceedings of the 17th Annual Conference on Learning Theory (COLT), pp. 109–123 (2004)
34. Mossel, E., O'Donnell, R., Servedio, R.: Learning juntas. In: Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC, pp. 206–212. ACM (2003)
35. Naor, M., Schulman, L., Srinivasan, A.: Splitters and near-optimal derandomization. In: Proceedings of the 36th Annual Symposium on Foundations of Computer Science, pp. 182–191 (1995)
36. Nilli, A.: Perfect hashing and probability. Comb. Probab. Comput. **3**, 407–409 (1994)
37. Orlitsky, A.: Worst-case interactive communication i: Two messages are almost optimal. IEEE Trans. Inf. Theory **36**, 1111–1126 (1990)

38. Recht, B., Fazel, M., Parrilo, P.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**, 471–501 (2010)
39. Tropp, J.: User-friendly tail bounds for sums of random matrices. Found. Comput. Math. **12**(4), 389–434 (2012)
40. Tyagi, H., Cevher, V.: Active learning of multi-index function models. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1475–1483 (2012)
41. Tyagi, H., Cevher, V.: Learning non-parametric basis independent models from point queries via low-rank methods. Appl. Comput. Harmonic Anal. (2014)
42. Tyagi, H., Gärtner, B.: Continuum armed bandit problem of few variables in high dimensions. CoRR abs/1304.5793 (2013)
43. Wang, Z., Zoghi, M., Hutter, F., Matheson, D., de Freitas, N.: Bayesian optimization in high dimensions via random embeddings. In: Proc. IJCAI (2013)
44. Wedin, P.: Perturbation bounds in connection with singular value decomposition. BIT **12**, 99–111 (1972)
45. Weyl, H.: Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). Mathematische Annalen **71**, 441–479 (1912)