Context-lumpable stochastic bandits

Chung-Wei Lee[*]
University of Southern California
leechung@usc.edu

Qinghua Liu
Princeton University
qinghual@princeton.edu

Yasin Abbasi-Yadkori
Google DeepMind
yadkori@google.com

Chi Jin
Princeton University
chij@princeton.edu

Tor Lattimore
Google DeepMind
lattimore@google.com

Csaba Szepesvári
Google DeepMind and University of Alberta
szepesva@ualberta.ca

Abstract

We consider a contextual bandit problem with S contexts and K actions. In each round $t = 1, 2, \dots$ the learner observes a random context and chooses an action based on its past experience. The learner then observes a random reward whose mean is a function of the context and the action for the round. Under the assumption that the contexts can be lumped into $r \leq \min\{S, K\}$ groups such that the mean reward for the various actions is the same for any two contexts that are in the same group, we give an algorithm that outputs an ε-optimal policy after using at most $\widetilde{O}(r(S+K)/\varepsilon^2)$ samples with high probability and provide a matching $\Omega(r(S+K)/\varepsilon^2)$ lower bound. In the regret minimization setting, we give an algorithm whose cumulative regret up to time T is bounded by $\widetilde{O}(\sqrt{r^3(S+K)T})$. To the best of our knowledge, we are the first to show the near-optimal sample complexity in the PAC setting and $\widetilde{O}(\sqrt{\text{poly}(r)(S+K)T})$ minimax regret in the online setting for this problem. We also show our algorithms can be applied to more general low-rank bandits and get improved regret bounds in some scenarios.

1 Introduction

Consider a recommendation platform that interacts with a finite set of users in an online fashion. Users arrive at the platform and receive a recommendation. If they engage with the recommendation (e.g., they "click") then the platform receives a reward, otherwise no reward is obtained. Assume that the users can be partitioned into a small number of groups such that users in the same group have the same preferences.

We ask whether there exist algorithms that can take advantage of the lumpability of users into a few groups, even when the identity of the group a user belongs to is unknown and only learnable because they share preferences with other users in the group.

A slightly more general version of this problem can be formalized as follows: Viewing users as "contexts" and recommendations as "actions" (or arms), assume that there are S contexts and K actions. In round $t = 1, 2, \dots$ the learner first receives a context i_t, sampled from an unknown

[*]most works were done when interning at DeepMind.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

distribution on the set $[S] := \{1, \dots, S\}$ of possible contexts. The learner then chooses an action $j_t \in [K] := \{1, \dots, K\}$ and observes a reward $y_t = A(i_t, j_t) + \eta_t$, where given the past, η_t has a subgaussian tail (precise definitions are postponed to Section 2) and $A : [S] \times [K] \to \mathbb{R}$ is an unknown function of mean rewards ($\mathbb{R}$ denotes the set of reals). We consider two settings when the goal of the learner is either to identify a near-optimal policy $\pi : [S] \to [K]$, or to keep its regret small. Policy π is called ε-optimal if

$$\mathbb{E}[A(i_1, \pi(i_1))] \geq \max_{\pi'} \mathbb{E}[A(i_1, \pi'(i_1))] - \varepsilon, \tag{1}$$

while the regret of the learner for a horizon of T is

$$\mathrm{Reg}_T = \mathbb{E}\left[\sum_{t=1}^{T} \max_{j \in [K]} A(i_t, j) - \sum_{t=1}^{T} A(i_t, j_t)\right]. \tag{2}$$

The expectations are taken with respect to the randomness of both the learner and environment, including contexts and rewards. It is well known (e.g., Lattimore and Szepesvári [2020]) that there are algorithms such that an ε-optimal policy will be discovered after

$$\widetilde{O}\left(\frac{SK}{\varepsilon^2}\right) \tag{3}$$

interactions, and there are also algorithms for which the regret satisfies

$$\mathrm{Reg}_T = \widetilde{O}(\sqrt{SKT})\,. \tag{4}$$

Here, the notation $\widetilde{O}(\cdot)$ hides polylogarithmic factors of the variables involved. We say that the stochastic, finite, contextual bandit problem specified by A is *r-lumpable* (or, in short, the bandit problem is *context-lumpable*) if there is a partitioning of $[S]$ into $r \leq \min\{S, K\}$ groups such that $A(i, \cdot) = A(i', \cdot)$ holds whenever $i, i' \in [S]$ belong to the same group. It is not hard to see that any algorithm needs at least $\Omega(r(S + K)/\varepsilon^2)$ interactions to discover an ε-optimal policy (Theorem 3). Indeed, if we view A as an $S \times K$ matrix, the lumpability condition states that $A = UV$ where U is an $S \times r$ binary matrix where each row has a single nonzero element, and V is an $r \times K$ matrix, which gives the unique mean rewards given the group indices. Hence, crude parameter counting suggests that there are $r(S + K)$ parameters to be learned.

The question is whether SK in Eq. (3) and Eq. (4) can be replaced with $(S + K)\mathrm{poly}(r)$ without knowing the grouping of the contexts.

More generally, we can ask the same questions for contextual bandits with the *low-rank* structure, where the matrix A has rank r. Equivalently, the low-rank condition means that we have the same decomposition $A = UV$ as above but no more constraints on U. In the example of recommendation platforms, this assumption is more versatile as the users are modeled as *mixtures* of r preference types instead of belonging to one type only.

1.1 Related works

Our problem can be seen as an instance of contextual bandit problems introduced by Auer et al. [2002]. For a good summary of the history of the contextual bandit problem, the reader is advised to consult the article by Tewari and Murphy [2017]. Further review of prior works can also be found in the books of Slivkins [2019], Lattimore and Szepesvári [2020]. Another way to approach context-lumpable stochastic bandits is to model them as stochastic linear bandits with changing action sets Auer [2002], Chu et al. [2011], Abbasi-Yadkori et al. [2011]. However, lumpablility does not give improvements on the regret bound when running the standard algorithms in these settings (EXP4 and SupLinRel, respectively). We provide more details in the appendix.

We are not the first to study the r-context-lumpable stochastic bandit problem. The earliest work is probably that of Maillard and Mannor [2014] who named this problem *latent bandits*: they consider the group identity of the context as latent, or missing information. While the paper introduces this problem, the main theoretical results are derived for the case when the reward distributions given the group information are known to the learner. Further, the results derived are instance dependent.

Altogether, while this paper makes interesting contributions, it does not help in answering our questions.[2]

The earlier work of Gentile et al. [2014] considered another generalization of our problem where in round $t = 1, 2, \ldots$ the learner gets an action set $\{x_{t,1}, \ldots, x_{t,K_t}\} \subset \mathbb{R}^d$ and the mean payoff of choosing action $1 \le j \le K_t$ given the past and the current context i_t is $x_{t,j}^\top \theta_{g(i_t)}$ with $\theta_1, \ldots, \theta_r \in \mathbb{R}^d$ unknown. When specializing this to our setting, their result depends on the minimum separation $\min_{i,j:g(i)\neq g(j)} \|A(i,\cdot) - A(j,\cdot)\|$, which, according to the authors may be an artifact of their analysis. An extra assumption, which the authors suggest could be removed, is that the distribution of the contexts is uniform (as we shall see, removing this assumption is considerable work). Further, as the authors note, the worst-case for their algorithm is when the contexts are equipartitioned, in which case their bound reduces to the naive bound that one gets for non-lumpable problems. Li et al. [2016] considers the variation of the problem when the set of arms (and their feature vectors) are fixed and is known before learning begins and separation is defined with respect to this fixed set of arms (and hence could be larger than before). From our perspective, their theoretical results have the same weaknesses as the previous paper. Gentile et al. [2017] further generalizes these previous works to the case when the set of contexts is not fixed a priori. However, the previously mentioned weaknesses of the results remain.

Our problem is also a special case of learning in *lumpable Markov Decision Processes*. In such Markov Decision Processes (MDPs) the states can be partitioned such that states within a group of a partition behave identically, both in terms of the transitions and the rewards received [e.g., Ren and Krogh, 2002].[3] We put a survey of this line of work in Appendix A.6.

In summary, although the problem is extensively studied in the literature, we are not aware of any previous results that have implications for the existence of the minimax regret bounds in terms of $\widetilde{O}(\sqrt{\mathrm{poly}(r)(S+K)T})$, which we believe is a fundamental question in this area.

2 Notation and problem definition

For an positive integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. Further, for a finite set U, $\Delta(U)$ denotes the set of probability distributions over U and $\mathrm{unif}(U)$ denotes the uniform distribution over U. The set of real numbers is denoted by $\mathbb{R}$.

As introduced earlier, we consider context-lumpable bandits with S contexts and K actions such that the S contexts can be lumped into r groups so that the mean payoff $A(i,j)$ given that action $j \in [K]$ is used while the context is $i \in [S]$ depends only on the identity of the group that i belongs to and the action j. That is, for some $g : [S] \to [r]$ map

$$A(i,j) = A(i',j) \qquad \text{for any } j \in [K], i, i' \in [S] \text{ such that } g(i) = g(i') \,.$$

Neither A, nor g is known to the learner who interacts with the bandit instance in the following way: In rounds $t = 1, 2, \ldots$ the learner first observes a context $i_t \in [S]$ randomly chosen from a fixed, context distribution $\nu \in \Delta([S])$, independently of the past. The distribution ν is also unknown to the learner. Next, the learner chooses an action $j_t \in [K]$ to receive a reward of $y_t = A(i_t, j_t) + \eta_t$, where η_t is 1-subgaussian given the past: For any $\lambda \in \mathbb{R}$,

$$\mathbb{E}[\exp(\lambda \eta_t) \mid i_1, j_1, \ldots, i_t, j_t] \le \exp(\lambda^2/2) \text{ almost surely (a.s.).}$$

As described earlier, we are interested in two problem settings. In the *PAC setting*, the learner is given a target suboptimality level $\varepsilon > 0$ and a target confidence δ The learner is then tasked to discover a policy $\pi : [S] \to [K]$ that is ε-optimal (cf. Eq. (1)), with probability δ. In this setting, in addition to deciding what actions to choose, the learner also needs to decide when to stop collecting data and when it stops it has to return a map $\hat{\pi} : [S] \to [K]$. If T is the (random) time when the learner stops, then the efficiency of the learner is characterized by $\mathbb{E}[T]$. In the *online setting*, the goal of the learner is to keep its regret (cf. Eq. (2)) low. In Section 3, we will consider the PAC setting, while the online setting will be considered in Section 4.

[2] Somewhat confusingly, Hong et al. [2020] *define* latent bandit problems differently from the definition given by Maillard and Mannor [2014]: They assume that in addition to the context, a latent (unobserved) variable, influences the rewards, however, they also assume that the distributions of the reward given an action, context and latent state are all known. Their results therefore can not help us in answering our question.

[3] Lumpability was first introduced first by Kemeny and Snell [1976] in the context of Markov chains.

In the remainder of the paper, we will also need some further notation. The map g induces a partitioning $\mathcal{P}^\star = \{\mathcal{B}(1), \ldots, \mathcal{B}(r)\}$ of $[S]$ in the natural way:

$$\mathcal{B}(b) = \{i : g(i) = b\}\,.$$

We call each subset $\mathcal{B} \in \mathcal{P}^\star$ a *block* and call $\mathcal{B}(b)$ block b for every $b \in [r]$. We also define block reward $\mu(b, j) = A(i, j)$ of block b for every context $i \in \mathcal{B}(b)$ and arm $j \in [K]$. Finally, we define the block distribution $\omega \in \Delta(r)$ so that $\omega(b) = \sum_{i \in \mathcal{B}(b)} \nu(i)$.

3 Near-optimal PAC Learning in Context-lumpable Bandits

In this section, we present an algorithm for PAC learning in context-lumpable bandits, and prove that its sample complexity guarantee matches the minimax rate up to a logarithmic factor.

3.1 Special case: almost-uniform context distribution

To better illustrate the key ideas behind the algorithm design, we first consider a special case of almost-uniform context distribution. Formally, we assume $\nu(i) \in [1/(8S), 8/S]$ for all $i \in [S]$ throughout this subsection. The pseudocode of our algorithm is provided in Algorithm 1, which consists of three main modules. Below we elaborate on each of them separately.

Data collection (Line 3-4 and Algorithm 2). The high-level goal of this step is to collect a sufficient amount of data for every block/action pair so that in later steps we have sufficient information to select a near-optimal arm for each block. One naive approach is to try every action on every context for a sufficient amount of time (e.g., $\widetilde{O}(1/\varepsilon^2)$). However, this will cause an undesired factor of SK in the sample complexity. To address this issue, note that contexts from the same block share the same reward function, which means that for every block/action pair $(b, j) \in [r] \times [K]$, we only need to sufficiently try action j on a single context i from block b (i.e., $i \in \mathcal{B}(b)$). However, the block structure is unknown a priori. We circumvent this problem by assigning a random action $\psi(i)$ to each context i for them to explore (Line 4). And after action $\psi(i)$ has been tried on context i sufficiently many times, we update $\psi(i)$ by reassigning context i with a new random action (Line 8-10).

However, there is still one key problem unresolved yet: how many samples to use for estimating each $A(i, \psi(i))$ before reassigning context i with another random action, given a fixed budget of total sample complexity. On one hand, if we only try each $(i, \psi(i))$ pair a very few numbers of times before switching, then we can potentially explore a lot of different actions for each block but the accuracy of the reward estimate could be too low to identify a near-optimal policy. On the other hand, estimating each $A(i, \psi(i))$ with a huge number of samples resolves the poor accuracy issue but could potentially cause the problem of under-exploration, especially for those blocks consisting of very few contexts. In the extreme case, if a block only contains a single context, then our total budget may only afford to test a single action on that context (block).

To address this issue, we choose different levels of accuracy for different blocks adaptively depending on their block size. Specifically, contexts from larger blocks can afford to try each random action before switching for a larger number of times to achieve higher estimate accuracy because larger blocks consist of more contexts, which means that the average number of random actions in each context inside a larger block needs to try is smaller. And for smaller blocks, the case is the opposite. Finally, we remark that the above scheme of using adaptive estimate accuracy can be implemented without any prior knowledge of the block structure. We only need to iterate over different accuracy levels using an exponential schedule (Line 3), and each block will be sufficiently explored at its appropriate accuracy level. Specifically, let $N = \lceil \log(1/\varepsilon^2) \rceil$, and consider accuracy levels $n \in [N]$. For accuracy level n, we collect a dataset $\mathcal{D}_n$ by adding a context-action pair after the pair is played for 2^n timesteps (Line 9).

Screening optimal action candidates (Line 5-11). Equipped with the dataset collected from Step 1, we want to identify a subset $\mathcal{W}$ of $[K]$ so that (i) for every block $b \in [r]$, $\mathcal{W}$ contains a near-optimal action of block b, and (ii) the size of $\mathcal{W}$ is as small as possible. We construct such $\mathcal{W}$ in an iterative way. For each accuracy level $n \in [N]$, we first compute the context-action pair $(i^\star, j^\star)$ with the highest reward estimate in $\mathcal{D}_n$ (Line 7). Intuitively, as $(i^\star, j^\star)$ has been tried for 2^n timesteps in constructing $\mathcal{D}_n$, it is natural to guess that $j^\star$ could potentially be a $\widetilde{O}(2^{-n/2})$-optimal action for

certain blocks so we add it into optimal action candidate set $\mathcal{W}$ (Line 8). To identify the contexts from those blocks, we sample new data to estimate the reward of $j^\star$ on each context $i \in [S]$. This can be done by calling Algorithm 2 and setting the exploration set $\mathcal{K}$ to be $j^\star$ (Line 10). If a context i can achieve reward close enough to $\widehat{A}_n(i^\star, j^\star)$ at action $j^\star$, then we peel off every $(i, j) \in \mathcal{D}_n$ (Line 11). This is because we have found $j^\star$ as an optimal action candidate for i at this accuracy level and don't need to consider other j. We repeat the above process, and add a new arm to $\mathcal{W}$ in each iteration, until $\mathcal{D}_n$ becomes empty.

Solving the simplified context-lumpable bandit (Line 12). After obtaining the optimal action candidate set $\mathcal{W}$, we can discard all actions not in $\mathcal{W}$ and solve a simplified context-lumpable bandit problem by replacing the original action set $[K]$ with $\mathcal{W}$. Note that $\widetilde{O}(S|\mathcal{W}|/\varepsilon^2)$ samples suffice for learning an ε-optimal policy for this simplified problem. For example, we can directly try each action $j \in \mathcal{W}$ on each context $i \in [S]$ for $\widetilde{\Theta}(1/\varepsilon^2)$ times, and define $\pi^{\text{out}}(i) \in \arg\max_{j \in \mathcal{W}} \bar{A}(i, j)$ where $\bar{A}(i, j)$ is the empirical estimate of $A(i, j)$ based on $\widetilde{\Theta}(1/\varepsilon^2)$ samples.

Algorithm 1 Algorithm for Almost-uniform Context Distribution

1 Let t denote the current time and initialize $N \leftarrow \lceil \log(1/\varepsilon^2) \rceil$, $\mathcal{W} \leftarrow \emptyset$, $\widetilde{\lg} \leftarrow 16 \log(rSK/\delta)$
2 Define $\mathcal{K}(i) \leftarrow [K]$ for $i \in [S]$, $L \leftarrow r(S+K)\widetilde{\lg}/\varepsilon^2$
Step 1. Data collection
3 **for** *accuracy level* $n = 1, \ldots, N$ **do**
4 Execute Algorithm 2 with input L, n, $\mathcal{K}$, and receive $\mathcal{D}_n$ and $\widehat{A}_n$
Step 2. Screening optimal action candidates
5 **for** *accuracy level* $n = 1, \ldots, N$ **do**
6 **while** $\mathcal{D}_n \neq \emptyset$ **do**
7 Compute $(i^\star, j^\star) \leftarrow \arg\max_{(i,j) \in \mathcal{D}_n} \widehat{A}_n(i, j)$
8 Update optimal action candidates $\mathcal{W} \leftarrow \mathcal{W} \bigcup \{j^\star\}$
9 Update $\mathcal{K}(i) \leftarrow \{j^\star\}$ for $i \in [S]$ and $L' \leftarrow 8\widetilde{\lg} 2^n S$
10 Execute Algorithm 2 with input L', n, $\mathcal{K}$, and reassign output to $\widetilde{\mathcal{D}}_n$ and $\widetilde{A}_n$
11 Shrink $\mathcal{D}_n \leftarrow \{(i, j) \in \mathcal{D}_n : |\widetilde{A}_n(i, j^\star) - \widehat{A}_n(i^\star, j^\star)| \geq \sqrt{\frac{\widetilde{\lg}}{2^n}}\}$
Step 3. Solving the simplified context-lumpable bandit
12 Use $\frac{4S|\mathcal{W}|}{\varepsilon^2} \log \frac{SK}{\delta}$ samples to find π^{out} s.t. $A(i, \pi^{\text{out}}(i)) \geq \max_{j \in \mathcal{W}} A(i, j) - \varepsilon$ for all $i \in [S]$
13 **Output** π^{out}

Algorithm 2 Data Collection

Input L, n, $\mathcal{K}$
Let t denote the current time and initialize $\mathcal{D}_n \leftarrow \emptyset$
for *context* $i \in [S]$ **do**
Assign $\psi_t(i)$ to be an arm drawn from $\text{unif}(\mathcal{K}(i))$
for L *timesteps* **do**
Receive context i_t, play arm $j_t = \psi(i_t)$, and receive reward y_t
Preset $\psi_{t+1}(i) = \psi_t(i)$ for every $i \in [S]$
if $(\sum_{\tau=1}^{t} \mathbb{1}(i_\tau = i_t)) \% 2^n = 0$ **then**
$\mathcal{D}_n \leftarrow \mathcal{D}_n \bigcup \{(i_t, \psi(i_t))\}$ and $\widehat{A}_n(i_t, \psi(i_t)) \leftarrow \frac{\sum_{\tau \leq t} y_\tau \mathbb{1}[i_\tau = i_t, j_\tau = \psi(i_t)]}{\sum_{\tau \leq t} \mathbb{1}[i_\tau = i_t, j_\tau = \psi(i_t)]}$
Reassign $\psi_{t+1}(i_t) \sim \text{unif}(\mathcal{K}(i_t))$;
Output $\mathcal{D}_n$ and $\widehat{A}_n$

Now we present the theoretical guarantee for Algorithm 1. The proof and exact constants in the bound can be found in Appendix C.

Theorem 1. *Let $\delta \in (0, 1)$ and assume $\nu(i) \in [1/(8S), 8/S]$ for all $i \in [S]$. Algorithm 1 outputs an $\widetilde{O}(\varepsilon)$-optimal policy within $\widetilde{O}(r(S+K)\log(1/\delta)/\varepsilon^2)$ samples with probability at least $1 - \delta$.*

3.2 Extension to general context distribution

Algorithm 3 Algorithm for General Context Distribution

1 Let $J = \frac{4S}{\varepsilon}\log(S/\delta)$, $L = \lceil \log(S/\varepsilon) \rceil$, $N = 524\left(\log\frac{rSK}{\delta}\right)\cdot\left(1+2\log\frac{1}{\varepsilon}\right)^2\cdot\frac{r(S+K)}{\varepsilon^2}$

2 Estimate the context distribution by sampling J contexts, and denote the estimate by $\hat{\nu}$

3 Split the context set into L disjoint subsets $\{\mathcal{X}_l\}_{l\in[L]}$ where

$$\mathcal{X}_l \leftarrow \begin{cases} \{i \in [S] :\ \hat{\nu}(i) \in (2^{-l-1}, 2^{-l}]\}, & l \in [0 : L-1] \\ \{i \in [S] :\ \hat{\nu}(i) \in [0, 2^{-l}]\}, & l = L \end{cases}$$

4 **for** $l \in [L-1]$ **do**

5 Execute Algorithm 1 to learn policy π_l for subset $\mathcal{X}_l$ from N time steps

6 **Output** π^{out} such that π^{out} equals to π_l over $\mathcal{X}_l$ for $l \in [0 : L-1]$, and arbitrary over $\mathcal{X}_L$

In this subsection, we show how to generalize Algorithm 1 to handle general context distributions. We present the pseudo-code in Algorithm 3. The algorithm consists of two key steps. In the first step, we use $J = \widetilde{O}(S/\varepsilon)$ samples to obtain an empirical estimate of the context distribution, denoted by $\hat{\nu}$. Then we divide the context set into many disjoint subsets $\{\mathcal{X}_l\}_{l\in[0:L]}$ such that inside each subset $\mathcal{X}_l$, the conditional empirical context distribution is almost uniform. As a result, we can invoke Algorithm 1 to find a near-optimal policy π_l for each subset $\mathcal{X}_l$ (Line 5). It requires $\widetilde{O}(r(S+K)/\varepsilon^2)$ time steps for every ℓ but we only use samples where contexts are from $\mathcal{X}_l$ and ignore the rest. Finally, we glue all π_l together to get a policy π^{out} that is near-optimal for the original problem. Formally, we have the following theoretical guarantee for Algorithm 3. The proof and exact constants are in Appendix C.

Theorem 2. *Let $\delta \in (0,1)$. Algorithm 3 outputs an $\widetilde{O}(\varepsilon)$-optimal policy within $\widetilde{O}(r(S+K)\log(1/\delta)/\varepsilon^2)$ samples with probability at least $1-\delta$.*

Note that we can always learn an ε-optimal policy for any context-lumpable bandit within $\widetilde{O}(SK/\varepsilon^2)$ samples by invoking any existing near-optimal algorithm for finite contextual bandits. As a result, by combining Theorem 2 with the $\widetilde{O}(SK/\varepsilon^2)$ upper bound, we obtain that $\widetilde{O}(\min\{r(S+K), SK\}/\varepsilon^2)$ samples suffice for learning any context-lumpable bandit, which according to the following theorem is minimax optimal up to a logarithmic factor.

Theorem 3. *Learning an ε-optimal policy for a context-lumpable bandit with probability no smaller than $1/2$ requires at least $\Omega(\min\{r(S+K), SK\}/\varepsilon^2)$ samples in the worst case.*

4 Regret Minimization in Context-lumpable Bandits

In this section, we extend the idea from the PAC setting to the online setting. To better introduce the key ideas, we first consider a special case when both context and block distributions are uniform (Section 4.1). Then we consider the most general case in Section 4.2.

4.1 Special Case: Uniform Context and Block Distribution

In this section, we assume that distributions ν and ω are uniform, and thus, g evenly splits the contexts into r blocks so that there are S/r contexts in each block and every context appears with the same probability at each timestep. We will relax these assumptions and consider the general case in the next subsection.

For this case, we introduce Algorithm 4, which uses phased elimination in combination with a clustering procedure. The algorithm runs in phases $h = 1, 2, \ldots$ that are specified by error tolerance parameter $\varepsilon_h = 2^{-h/2}$. Like phased elimination algorithms for multi-armed bandits, we need to ensure at phase h actions the algorithms play are all $\widetilde{O}(\varepsilon_h)$-optimal. Thus, at the end of each phase h, we eliminate all actions that are not $\widetilde{O}(\varepsilon_h)$-optimal. However, the set of $\widetilde{O}(\varepsilon_h)$-optimal actions of each block is different. Therefore, we also perform *clustering* on contexts and perform elimination for each subset of the partition. Specifically, we maintain a partition of contexts $\mathcal{P}_h$ at each phase h and initialize $\mathcal{P}_1 = \{[S]\}$. For each cluster $\mathcal{C} \in \mathcal{P}_h$, we maintain a set of good arms $\text{GOOD}_h(\mathcal{C})$, which we will prove is $\widetilde{O}(\varepsilon_h)$-optimal for contexts in the cluster with high probability.

We use Algorithm 2 to collect data similar to Algorithm 1 (Line 6). At phase h, we try to estimate mean reward up to error $\widetilde{O}(\varepsilon_h)$ with probability $1-\delta_h$. The difference is that we assume ω is uniform, so we don't need different accuracy levels n, which will be required for the algorithm that handles the general case. Also, for every context $i \in \mathcal{C}$, we only explore arms *good for now*, that is, in $\text{GOOD}_h(\mathcal{C})$ instead of exploring all the arms $[K]$. This reflects that in the online setting, we need to minimize regret and cannot afford to explore bad arms too many times.

Based on the data we collect during the exploration stage, we then check if there is a large gap across contexts in the same subset for any arms (Line 7). A large gap suggests that (i) the subset contains at least two blocks (ii) the mean reward of the arm is significantly different in these blocks and we can use this arm to partition the contexts by running Algorithm 5 (clustering stage). We repeatedly do clustering (Line 9) and split heterogeneous subsets (Line 10) until we cannot find a large gap within the same subset. If no large gap is detected, then each arm has similar mean rewards (up to error $\widetilde{O}(\varepsilon_h)$) across all blocks in the same subset. Then we eliminate arms that are significantly worse than the empirical best arm (elimination stage) from $\text{GOOD}_h(\mathcal{C})$ for every subset $\mathcal{C}$ and start a new episode (Line 14).

Algorithm 5 plays j for each context $i \in \mathcal{C}$ for $\widetilde{O}(1/\varepsilon^2)$ rounds and calculates empirical means of arm j for each context in $\mathcal{C}$. It then sorts the contexts by the empirical means and performs clustering (Line 4). Specifically, the algorithm enumerates contexts in descending order of empirical means and splits contexts until a large gap between consecutive values is detected (Line 7). As we call Algorithm 5 only if a large gap is detected, we prove that in the appendix it correctly separates the subset into at least two parts without putting any contexts in the same block into different parts by setting $\varepsilon' = \varepsilon_h / r$.

Algorithm 4 Algorithm for Uniform Block Distribution

1 Initialize $\mathcal{P}_1 \leftarrow \{[S]\}$, $\text{GOOD}_1(\mathcal{C}) \leftarrow [K]$ for $\mathcal{C} \in \mathcal{P}_1$
2 Let t denote the current time
3 **for** *phase* $h = 1, 2, \ldots,$ **do**
4 Let $\varepsilon_h \leftarrow 2^{-h/2}$, $\delta_h \leftarrow \varepsilon_h^2/(r^3 SK)$, $\widetilde{\lg}_h \leftarrow 64 \log(rSK/\delta_h)$, $\widetilde{\varepsilon}_h \leftarrow \sqrt{\widetilde{\lg}_h} \cdot \varepsilon_h$
Step 1. Data collection
5 Define $L_h \leftarrow \frac{r(S+K)\widetilde{\lg}_h}{\varepsilon_h^2}$, $n_h \leftarrow \log(\frac{1}{\varepsilon_h^2})$, $\mathcal{K}_h(i) \leftarrow \text{GOOD}_h(\mathcal{C}), \mathcal{C} \in \mathcal{P}_h, \mathcal{C} \ni i, \forall i \in [S]$
6 Execute Algorithm 2 with input L_h, n_h, $\mathcal{K}_h$, and receive $\mathcal{D}_h$ and $\widehat{A}_h$
Step 2. Test homogeneity and perform clustering on heterogeneous subsets
7 **while** $\exists \mathcal{C} \in \mathcal{P}_h, \bar{i}, \underline{i} \in \mathcal{C}, j \in [K]$ *such that* $\widehat{A}_h(\bar{i}, j) - \widehat{A}_h(\underline{i}, j) \geq \widetilde{\varepsilon}_h$ **do**
8 Define $\varepsilon' \leftarrow \frac{\varepsilon_h}{4r}$, $\delta' \leftarrow \frac{\delta_h}{r}$, $\mathcal{K}(i) \leftarrow \{j\}$ if $i \in \mathcal{C}$ else $\text{GOOD}_h(\mathcal{C})$ for $\mathcal{C} \in \mathcal{P}_h, \mathcal{C} \ni i$
9 Execute Algorithm 5 with input ε', δ', $\mathcal{K}$, $\mathcal{C}$, and j, and get $\mathcal{P}$, a partition of $\mathcal{C}$
10 Initialize $\text{GOOD}_h(\mathcal{C}') \leftarrow \text{GOOD}_h(\mathcal{C}), \forall \mathcal{C}' \in \mathcal{P}$ and update $\mathcal{P}_h \leftarrow (\mathcal{P} \cup \mathcal{P}_h) \backslash \{\mathcal{C}\}$
Step 3. Eliminate suboptimal actions in each subset
11 $\mathcal{P}_{h+1} \leftarrow \mathcal{P}_h$
12 **for** $\mathcal{C} \in \mathcal{P}_{h+1}$ **do**
13 Calculate $\overline{\mu}_h(\mathcal{C}, j) \leftarrow \max_{i: i \in \mathcal{C}, (i,j) \in \mathcal{D}_h} \widehat{A}_h(i, j), \forall j \in \text{GOOD}_h(\mathcal{C})$
14 Update $\text{GOOD}_{h+1}(\mathcal{C}) \leftarrow \{j : j \in \text{GOOD}_h(\mathcal{C}),\ \max_{j'} \overline{\mu}_h(\mathcal{C}, j') - \overline{\mu}_h(\mathcal{C}, j) \leq 2\widetilde{\varepsilon}_h\}$

Similar to the analysis of other phased elimination algorithms, we have to show that in a phase specified by error level ε_h, with high probability, (i) the optimal arm is not eliminated and (ii) all $\widetilde{\omega}(\varepsilon_h)$-suboptimal arms are eliminated, that is, all arms in $\text{GOOD}_h(\mathcal{C})$ for all $\mathcal{C}$ are $\widetilde{O}(\varepsilon_h)$-optimal. We show the final regret here and defer the details to Appendix D

Theorem 4. *Under the assumption that context distribution ν and block distribution ω are uniform, regret of Algorithm 6 is bounded as* $\text{Reg}_T = \widetilde{O}(\sqrt{r^3(S+K)T})$.

Algorithm 5 Split Contexts Into Multiple Blocks

Input: error ε', confidence δ', exploration sets $\mathcal{K}$, subset $\mathcal{C}$, separating arm j
Initialize $\widetilde{\lg} \leftarrow 64\log(S/\delta')$, $L' \leftarrow S\widetilde{\lg}/\varepsilon'^2$, $n' \leftarrow \lceil 1/\varepsilon'^2 \rceil$
Execute Algorithm 2 with input L', n', $\mathcal{K}$, and reassign output to $\mathcal{D}$ and $\widehat{A}$
Sort contexts in $\mathcal{C}$ and label them as $i_1, \ldots, i_{|\mathcal{C}|}$ so that $\widehat{A}(i_1, j) \geq \widehat{A}(i_2, j) \geq \cdots \geq \widehat{A}(i_{|\mathcal{C}|}, j)$
Initialize $\mathcal{P}_1 \leftarrow \{i_1\}$, $b \leftarrow 1$
for $k = 2, 3, \ldots, |\mathcal{C}|$ **do**
    if $\widehat{A}(i_{k-1}, j) - \widehat{A}(i_k, j) \geq \sqrt{\widetilde{\lg}} \cdot \varepsilon'$ **then**
        Update $b \leftarrow b + 1$ and initialize $\mathcal{P}_b \leftarrow \{\}$
    $\mathcal{P}_b \leftarrow \mathcal{P}_b \cup \{i_k\}$
Output $\{\mathcal{P}_1, \ldots, \mathcal{P}_b\}$

Compared to our PAC result, we get an extra dependency on r. This is because we pay $\widetilde{O}(S/\varepsilon'^2) = \widetilde{O}(r^2 S/\varepsilon_h^2)$ samples to do clustering instead of $\widetilde{O}(1/\varepsilon_h^2)$ in order to ensure a "perfect" partition, that is, we never separate contexts in the same block with high probability. This is crucial for our phase-elimination algorithm as we may call Algorithm 5 in different phases. We left getting better than $\widetilde{O}(\sqrt{r^3(S+K)T})$ regret upper bounds or better than $\Omega(\sqrt{r(S+K)T})$ regret lower bounds (even for this uniform special case) as an important future direction.

4.2 Non-uniform Context and Block Distribution

Similar to the PAC learning setting, we can use Algorithm 3 to reduce general context distributions to (nearly) uniform ones. We provide more details in Appendix F. As a result, without loss of generality, we may assume ν is almost-uniform, and we focus on how to handle non-uniform block distribution ω. In the extreme, there may only be one context for some blocks, which becomes challenging to estimate their mean rewards.

For this case, we introduce Algorithm 6. Intuitively, based on Algorithm 4, we can further employ different accuracy levels as Algorithm 1. However, as our goal becomes minimizing regret, it is difficult to control regret for smaller n and larger blocks. Specifically, for smaller n, we explore more actions (with fewer samples) for each context in a single phase. Since fewer samples are used, we may unavoidably play suboptimal actions and suffer large regret. This becomes worse for a large block as more contexts in the block suffer large regret. We note that this is not a problem in the PAC setting because we only need to control sample complexity.

We fix the issue by setting different lengths L for different accuracy levels. Recall in Algorithm 1, we use the same length $L = \widetilde{O}(r(S+K)/\varepsilon^2)$ for every n. Intuitively, since we allow less accurate estimations for smaller n, we may use fewer data. It turns out indeed we can set $L = \widetilde{O}(r(S+K)2^{(n+h)/2})$ for level n at phase h, and with more refined analysis, it provides similar guarantees as before. More importantly, since smaller n uses (exponentially) fewer samples, the overall regret is well controlled.

In addition to setting the lengths sophisticatedly, we also need to carefully maintain sets of good actions $\text{GOOD}_{h,n}$ not only at each phase h but also at each accuracy level n. The partition of contexts $\mathcal{P}_h$, however, only evolves with phases and is shared between different levels at the same phase. Similar to Algorithm 4, we also do clustering (Step 2) and elimination (Step 3) but do the procedures in parallel for every accuracy level n. In the clustering stage, we use different thresholds to define a "large gap". This reflects that n represents different accuracy levels and thus has different widths of confidence intervals. For the elimination stage, we enforce the inclusion relation $\text{GOOD}_h(n, \mathcal{C}) \subseteq \text{GOOD}_h(n', \mathcal{C})$ for $n' \leq n$ (Line 17), which will be useful in the regret analysis. We now present the main theorem and put the complete proof in Appendix G.

Theorem 5. *Regret of Algorithm 6 is bounded as* $\text{Reg}_T = \widetilde{O}(\sqrt{r^3(S+K)T})$.

We remark that Algorithm 6 is *anytime*, that is, it doesn't require the time horizon T as input. However, it does require the number of blocks r as prior knowledge. Removing the knowledge of r is an interesting future direction. One promising idea is to apply a doubling trick on r. Specifically, we have an initial guess $r = \widetilde{O}(1)$ and run Algorithm 6; when $|\mathcal{P}_h| > r$, we double r and restart

Algorithm 6 Algorithm for Non-uniform Block Distribution

Initialize $\mathcal{P}_1 \leftarrow \{[S]\}$, $\text{GOOD}_{1,1}(\mathcal{C}) \leftarrow [K], \mathcal{C} \in \mathcal{P}_1$
Let t denote the current time
for *phase* $h = 1, 2, \ldots,$ **do**
  Let $\varepsilon_h \leftarrow 2^{-h/2}$, $N_h \leftarrow h$, $\delta_h \leftarrow \varepsilon_h^2/(r^3 SK)$, $\widetilde{\lg}_h \leftarrow 128 \log(rSKN_h/\delta_h)$
  Step 1. Data collection
  for *accuracy level* $n = 1, \ldots, N_h$ **do**
    Define $L_{h,n} \leftarrow r(S+K)\widetilde{\lg}_h 2^{(n+h)/2}$, $\mathcal{K}_{h,n}(i) \leftarrow \text{GOOD}_{h,n}(\mathcal{C}), \mathcal{C} \in \mathcal{P}_h, \mathcal{C} \ni i, \forall i \in [S]$
    Execute Algorithm 2 with input L_h, n, $\mathcal{K}_{h,n}$, and receive $\mathcal{D}_{h,n}$ and $\widehat{A}_{h,n}$
  Step 2. Test homogeneity and perform clustering on heterogeneous subsets
  while $\exists \mathcal{C} \in \mathcal{P}_h, \bar{i}, \underline{i} \in \mathcal{C}, j \in [K], n \in [N_h]$ *such that* $\widehat{A}_{h,n}(\bar{i}, j) - \widehat{A}_{h,n}(\underline{i}, j) \geq \sqrt{\frac{\widetilde{\lg}_h}{2^n}}$ **do**
    Define $\varepsilon' \leftarrow \frac{\varepsilon_h}{4r}$, $\delta' \leftarrow \frac{\delta_h}{r}$, $\mathcal{K}(i) \leftarrow \{j\}$ if $i \in \mathcal{C}$ else $\text{GOOD}_{h,n}(\mathcal{C})$ for $\mathcal{C} \in \mathcal{P}_h, \mathcal{C} \ni i$
    Execute Algorithm 5 with input ε', δ', $\mathcal{K}$, $\mathcal{C}$, and j, and get $\mathcal{P}$, a partition of $\mathcal{C}$
    Initialize $\text{GOOD}_{h,n}(\mathcal{C}') \leftarrow \text{GOOD}_{h,n}(\mathcal{C}), \forall \mathcal{C}' \in \mathcal{P}$ and update $\mathcal{P}_h \leftarrow (\mathcal{P} \cup \mathcal{P}_h) \backslash \{\mathcal{C}\}$
  Step 3. Eliminate suboptimal actions in each subset
  $\mathcal{P}_{h+1} \leftarrow \mathcal{P}_h$, $\text{GOOD}_{1,1}(\mathcal{C}) \leftarrow [K], \mathcal{C} \in \mathcal{P}_{h+1}$
  for *accuracy level* $n = 2, \ldots, N_h$ **do**
    for $\mathcal{C} \in \mathcal{P}_{h+1}$ **do**
      Calculate $\overline{\mu}_{h,n}(\mathcal{C}, j) \leftarrow \max_{i: i \in \mathcal{C}, (i,j) \in \mathcal{D}_{h,n}} \widehat{A}_{h,n}(i, j), \forall j \in \text{GOOD}_{h,n}(\mathcal{C})$
      Let $\mathcal{G}_{h,n}(\mathcal{C}) \leftarrow \left\{ j : j \in \text{GOOD}_{h,n}(\mathcal{C}),\ \max_{j'} \overline{\mu}_{h,n}(\mathcal{C}, j') - \overline{\mu}_{h,n}(\mathcal{C}, j) \leq 2\sqrt{\frac{\widetilde{\lg}_h}{2^n}} \right\}$
      Update $\text{GOOD}_{h+1,n}(\mathcal{C}) \leftarrow \text{GOOD}_{h+1,n-1}(\mathcal{C}) \cap \mathcal{G}_{h,n}(\mathcal{C})$

the algorithm. The analysis, though, may be much more complicated. Finally, we note that when knowing r, one can calculate in advance if $\sqrt{SKT}$ is less than $\sqrt{r^3(S+K)T}$ and switch to a standard algorithm achieving $\widetilde{O}(\sqrt{SKT})$ in that case. This modification guarantees the regret is never worse than the standard bound.

5 From Context-lumpable Bandits to Contextual Low-rank Bandits

Finally, we consider the more general *contextual low-rank bandit* problem. Specifically, we allow A to have rank r, that is $A = UV$ for some $S \times r$ matrix U and $r \times K$ matrix V. Lumpability is a special case in the sense that U is binary where each row has a single nonzero element.

To solve the problem, We show a reduction from contextual low-rank bandits to context-lumpable bandits. Consider an α-covering of rows of U, and notice that the covering number, $\mathcal{R}_\alpha$, is $1/\alpha^r$. The context-lumpable bandits can be seen as α-approximate context-lumpable bandits with $\mathcal{R}_\alpha$ blocks, where the reward of contexts on the same block differs at most α. Ignoring this misspecification, we may run Algorithm 1 and Algorithm 6 for the PAC and online settings, respectively. Moreover, it turns out that our algorithms are robust to this misspecification when α is sufficiently small ($O(\varepsilon)$ in the PAC setting, for example). Therefore, the sample complexity and the regret bounds of these algorithms will be in terms of $\mathcal{R}_\alpha$ despite having an exponential dependency on r. The resulting bounds can still be smaller than the naive SK/ε^2 and $\sqrt{SKT}$ bounds in some scenarios, for example, when S and K are super large. We put the details in Appendix H.

6 Conclusions and Future Directions

We consider a contextual bandit problem with S contexts and K actions. Under the assumption that the context-action reward matrix has $r \leq \min\{S, K\}$ unique rows, we show an algorithm that outputs an ε-optimal policy and has the optimal sample complexity of $\widetilde{O}(r(S+K)/\varepsilon^2)$ with high

probability. In the regret minimization setting, we show an algorithm whose cumulative regret up to time T is bounded as $\widetilde{O}(\sqrt{r^3(S+K)T})$.

An immediate next question is whether a regret bound of order $\widetilde{O}(\sqrt{r(S+K)T})$ is achievable in the regret minimization setting. A second open question is concerned with obtaining a $\widetilde{O}(\sqrt{\text{poly}(r)(S+K)T})$ regret bound in contextual low-rank bandits. Our regret analysis heavily relies on the assumption that contexts arrive in an I.I.D fashion. Extending our results to the setting with adversarial contexts remains another important future direction.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models–going beyond svd. In *2012 IEEE 53rd annual symposium on foundations of computer science*, pages 1–10. IEEE, 2012.

P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

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

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

Leonardo Cella, Alessandro Lazaric, and Massimiliano Pontil. Meta-learning with stochastic linear bandits. *Arxiv*, 2020.

Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, and Yuling Yan. Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. *SIAM journal on optimization*, 30(4):3098–3121, 2020.

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

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient PAC RL with rich observations. *Advances in neural information processing systems*, 31, 2018.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

Yaqi Duan, Tracy Ke, and Mengdi Wang. State aggregation learning from Markov transition data. *Advances in Neural Information Processing Systems*, 32, 2019.

Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 169–178, 2011.

Fei Feng, Ruosong Wang, Wotao Yin, Simon S Du, and Lin Yang. Provably efficient exploration for reinforcement learning using unsupervised learning. *Advances in Neural Information Processing Systems*, 33:22492–22504, 2020.

Dylan Foster and Alexander Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*, pages 3199–3210. PMLR, 2020.

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765. PMLR, 2014.

Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1253–1262, 2017.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. In *NeurIPS*, 2020.

Prateek Jain and Soumyabrata Pal. Online low rank matrix completion, 2022.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

Kyoungseok Jang, Kwang-Sung Jun, Se-Young Yun, and Wanmo Kang. Improved regret bounds of bilinear bandits using action space analysis. In *International Conference on Machine Learning*, pages 4744–4754. PMLR, 2021.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Kwang-Sung Jun, Rebecca Willett, Stephen Wright, and Robert Nowak. Bilinear bandits with low-rank structure. In *International Conference on Machine Learning*, pages 3163–3172. PMLR, 2019.

Yue Kang, Cho-Jui Hsieh, and Thomas Chun Man Lee. Efficient frameworks for generalized low-rank matrix bandit problems. In *Advances in Neural Information Processing Systems*, 2022.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, Claire Vernade, and Zheng Wen. Stochastic rank-1 bandits. In *Artificial Intelligence and Statistics*, pages 392–401. PMLR, 2017.

John G Kemeny and James Laurie Snell. *Finite Markov chains*. Springer, 1976.

Branislav Kveton, Csaba Szepesvári, Anup Rao, Zheng Wen, Yasin Abbasi-Yadkori, and S Muthukrishnan. Stochastic low-rank bandits. *arXiv preprint arXiv:1712.04644*, 2017.

Branislav Kveton, Martin Mladenov, Chih-Wei Hsu, Manzil Zaheer, Csaba Szepesvári, and Craig Boutilier. Differentiable meta-learning in contextual bandits. *arXiv:2006.05094v1*, 2020.

Branislav Kveton, Mikhail Konobeev, Manzil Zaheer, Chih wei Hsu, Martin Mladenov, Craig Boutilier, and Csaba Szepesvari. Meta-Thompson sampling. *Arxiv*, 2021.

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. RL for latent MDPs: Regret guarantees and a lower bound. *Advances in Neural Information Processing Systems*, 34: 24523–24534, 2021.

Tor Lattimore and Botao Hao. Bandit phase retrieval. *Advances in Neural Information Processing Systems*, 34:18801–18811, 2021.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

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

Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Low-rank generalized linear bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 460–468. PMLR, 2021.

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *International Conference on Machine Learning*, pages 136–144. PMLR, 2014.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pages 6961–6971. PMLR, 2020.

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank MDPs. *arXiv preprint arXiv:2102.07035*, 2021.

Chengzhuo Ni, Anru R Zhang, Yaqi Duan, and Mengdi Wang. Learning good state and action representations via tensor decomposition. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1682–1687. IEEE, 2021.

Matteo Papini, Andrea Tirinzoni, Aldo Pacchiano, Marcello Restelli, Alessandro Lazaric, and Matteo Pirotta. Reinforcement learning in linear MDPs: Constant regret and representation selection. *Advances in Neural Information Processing Systems*, 34:16371–16383, 2021.

Zhiyuan Ren and B.H. Krogh. State aggregation in Markov decision processes. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 4, pages 3819–3824, 2002.

Rajat Sen, Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G. Dimakis, and Sanjay Shakkottai. Contextual bandits with latent confounders: An NMF approach, 2016.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 47(3):1904–1931, 2022.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

A. Tewari and S. A. Murphy. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, pages 495–517. Springer, 2017.

Cindy Trinh, Emilie Kaufmann, Claire Vernade, and Richard Combes. Solving Bernoulli rank-one bandits with unimodal Thompson sampling. In *Algorithmic Learning Theory*, pages 862–889. PMLR, 2020.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline RL in low-rank MDPs. In *International Conference on Learning Representations*, 2022.

Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning robust state abstractions for hidden-parameter block MDPs. In *International Conference on Learning Representations*, 2020.

Weitong Zhang, Jiafan He, Dongruo Zhou, Amy Zhang, and Quanquan Gu. Provably efficient representation learning in low-rank Markov decision processes. *arXiv preprint arXiv:2106.11935*, 2021.

Xuezhou Zhang, Yuda Song, Masatoshi Uehara, Mengdi Wang, Alekh Agarwal, and Wen Sun. Efficient reinforcement learning in block MDPs: A model-free representation learning approach. In *International Conference on Machine Learning*, pages 26517–26547. PMLR, 2022.

A More Related Works

A.1 From bilinear to low-rank bandits

The low-rank structure has been shown useful in other bandit problems. Katariya et al. [2017] considered the case when the learner chooses both "context" and arm and the reward map A, when viewed as a $S \times K$ matrix, has *rank one*. This is a special case of a linear bandit setting with a fixed action set where both the features underlying the actions and the parameter vectors are "shaped" as matrices of matching dimensions, and both matrices are rank-one. Similarly to our problem, the challenge is to make the regret depend on $S + K$ rather than on $S \times K$ (but note that S does not have the interpretation of the number of contexts here). This paper demonstrates that this is possible when considering gap-dependent bounds.

Trinh et al. [2020] improved the result of Katariya et al. [2017] and gave an asymptotically optimal regret bound. Kveton et al. [2017] generalized the setting of Katariya et al. [2017] to the case when the reward A (viewed as matrix) is of rank $r \leq \min(S, K)$ and is a "hott topics matrix", the learner in each round can choose r row and column indices, observes the entries of A at the resulting submatrix of A in noise and incurs the maximum reward in this submatrix. For this problem they show an instance dependent bound that depends on S, K, r only through $(S + K)\text{poly}(r)$.

Jun et al. [2019] dropped the extra conditions on the mean reward matrix besides assuming that it has low rank. Jang et al. [2021] gave the first upper bound of the form $\tilde{O}((S + K)r\sqrt{T})$, though with inefficient algorithms, which was the first bound better than the naive bound $\tilde{O}(SK\sqrt{T})$.[4] Later, Lu et al. [2021] dropped the condition on the action matrices and also extended the results to the generalized linear setting (they assume that both the action matrices and the parameter matrix has a constant Frobenius norm bound). The regret bound of Jun et al. [2019], Lu et al. [2021] takes the form $\tilde{O}((S + K)^{3/2}\sqrt{rT})$, which is still worse than the earlier mentioned naive bound. Lattimore and Hao [2021] prove that (up to logarithmic factors) when $S = K$ and both the action and reward matrices are rank one and symmetric, the minimax regret is of order $S\sqrt{T}$.

Recently, Kang et al. [2022] improved the previous state-of-the-art for the generalized linear setting. The new regret bound, which they prove "under a mild" extra assumption, takes the form $\tilde{O}((S + K)r\sqrt{T})$ and is conjectured to match the order of the minimax regret. Noticing that the minimax lower bounds developed for the finite-armed stochastic bandits (e.g., Exercise 15.2 of Lattimore and Szepesvári 2020) is applicable to the rank-one setting (as the proofs use actions and rewards that are rank one, one-hot matrices), we see that the regret is at least $\Omega(\sqrt{SKT})$ in these problems. As $\sqrt{S + K} \approx \sqrt{\max(S, K)} \ll \sqrt{SK} \leq \sqrt{\max(S, K)^2} = \max(S, K) \approx S + K$, this lower bound rules out upper bounds of the form $O(\sqrt{(S + K)\text{poly}(r)T})$, though it is compatible with the upper bound mentioned above.[5]

Jain and Pal [2022] study a related problem where the learner chooses one action per context in each round and incurs a reward for each of the actions. They also propose an epsilon-greedy type algorithm and show that its regret scales as $T^{2/3}\text{polylog}(S + K)$ provided some technical conditions hold, one of which is that the condition number of the reward A when viewed as a matrix is constant. The main limitation of this work is that the technical conditions are restrictive and the problem is easier as in each round an observation is available for each context, whereas in our setting the contexts arrive at random from a distribution which may be very far from the uniform distribution, which, as we shall see, will require extra care.

A.2 Bandit meta-learning

Bandit meta-learning is concerned with learning a lower dimensional subspace across bandit tasks [Kveton et al., 2020, 2021, Cella et al., 2020], while we consider only a single task.

[4] To get this naive bound, following an argument of Jang et al. [2021], notice that the problem is an instance of d-dimensional linear bandits with $d = SK$ with actions and parameters belonging to the unit sphere, in which case the minimax regret, up to logarithmic factors, is of order $d\sqrt{T}$ (e.g., Theorem 24.2, Lattimore and Hao 2021). If the action set has one-hot matrices only, this bound improves to $\sqrt{dT}$, as discussed before.

[5] Here, $f(S, K) \approx g(S, K)$ means that f, g are within a constant factor of each other, while $f(S, K) \ll g(S, K)$ means that $f(S, K) = o(g(S, K))$ as $S, K \to \infty$.

A.3 Contextual bandit problems

Contextual bandit problems, introduced by Auer et al. [2002], can be seen as a special case of the *prediction with expert advice problem*, which is an online problem. In a prediction with expert advice problem, the learner is given access to the recommendations of N experts in the form of distributions over the K actions. The learner still needs to choose an action in each round with the goal of competing with the total reward collected by the best expert in hindsight. Auer et al. [2002] consider the adversarial case when in each round, each action is assigned a reward in an arbitrary way from a known, finite interval. The authors propose the EXP4 algorithm that is shown to achieve $O(\sqrt{KT \log N})$ regret in T rounds of interactions. Beygelzimer et al. [2011] extended this result to control the regret with high probability, while Dudik et al. [2011] introduced a computationally efficient variant assuming a computationally efficient cost-sensitive classification oracle.

To reduce an r-lumpable contextual bandit problem to prediction with expert advice, we need to choose the experts. The obvious choice here is that an expert is a $[S] \to [K]$ map that is the composition of an $[S] \to [r]$ map followed by an $[r] \to [K]$ map. Denoting by N the number of such experts, we see that $\log(N) = S \log(r) + r \log(K)$, and hence the regret of EXP4 is of order $\Omega(\sqrt{SKT})$, which is not better than not using the lumpability structure.

Another line of work focuses on oracle-based contextual bandits [Foster and Rakhlin, 2020, Simchi-Levi and Xu, 2022]. In this setting, the learner has access to function class $\mathcal{F}$ and the optimal regret is $O(\sqrt{KT \log(|\mathcal{F}|)})$, where $|\mathcal{F}|$ is the size of the function class. However, with the same argument, we can conclude $|\mathcal{F}| = \Omega(S)$ even under the lumpable assumption. Therefore, the results in this direction also give no direct implication to the problem we consider.

A.4 Stochastic linear bandits with changing action sets

In this setting, in each round $t = 1, 2, \dots$ the learner first receives K d-dimensional vectors, $x_{t,1}, \dots, x_{t,K}$, each corresponding to an action. Choosing action j gives a reward whose mean (given the past) is $x_{t,j}^\top \theta$ for some unknown parameter vector $\theta \in \mathbb{R}^d$. For our case, one can choose $d = SK$: θ can be the "flattening" of A and $x_{t,j}$ is a unit vector so that $x_{t,j}^\top \theta = A(i_t, j)$. Applying the SupLinRel algorithm from Section 4 of Auer [2002] to this setting, we get the regret bound $O(\sqrt{dT \log^3(KT)})$, which shows no improvement compared to ignoring lumpability.

A.5 Matrix completion problems

The offline version of our problem is closely related to matrix completion problems, where the goal is to reconstruct a matrix with missing values under the low-rank constraint [Arora et al., 2012, Jain et al., 2013, Chen et al., 2020]. Based on these ideas, Sen et al. [2016] propose an epsilon-greedy algorithm for the contextual low-rank bandit problem and show that its regret is of order $T^{2/3}(S \operatorname{poly}(r, \log K))^{1/3}$. This result holds under an assumption that the reward matrix has a nonnegative decomposition $A = UV$ where entries of U and V are all nonnegative. If A is nonnegative valued itself, r-lumpability of the contexts implies that such a decomposition exist. However, the main result of this work needs additional conditions. In particular, the context distribution needs to be near-uniform, and due to their "restricted isometry property" assumptions, the context groups need to be of approximately the same size. In particular, if all context groups except one have a single member, their bound degrades to the trivial bound mentioned earlier.

A.6 Lumpable Markov decision processes

Lumpable MDPs in contemporary literature on machine learning are referred to as *block MDPs* [Du et al., 2019]. Learning to act in a block MDPs has been the subject a numerous recent papers [Dann et al., 2018, Du et al., 2019, Misra et al., 2020, Feng et al., 2020, Zhang et al., 2020, 2022]. Furthermore, learning to act in a block MDP is a special case of the so-called low-rank setting, which was also heavily studied [Jiang et al., 2017, Uehara et al., 2022, Modi et al., 2021, Papini et al., 2021, Zhang et al., 2021, Misra et al., 2020], both in the online and in the PAC settings. The learner in these problems is given a set of feature maps with the promise that at least one of the feature maps will allow a factored (low-rank) representation of the transition dynamics and the reward.

Despite the significant effort that went into studying this problem, in our setting none of the existing results improve upon the naive bound that one can get for non-lumpable problems. Duan et al. [2019], Ni et al. [2021] study the offline setting, where the considerations are rather different. Finally, Kwon et al. [2021] consider the problem when over multiple episodes, a learner interacts with a finite-horizon MDP, which is chosen at random from one of r such unknown MDPs. The learner is given no additional information about the hidden identity of the MDP that it faces. The challenge here is that even if the MDPs were known, the problem of acting optimally is nontrivial, while in our case this is not a problem. As such, while superficially, the problems may look similar, they are quite different.

B Auxiliary Lemmas

In this section, we show some probabilistic lemmas that will be useful to prove our results. The following lemma is a high-probability version of the classical "coupon collector's problem", which says that the expected number of coupons required to draw with replacement is $\Theta(K \log K)$ in order to get each of K coupon at least once.

Lemma 6. *Given a set $\mathcal{K}$ with $|\mathcal{K}| \leq K$ and consider M i.i.d samples drawn from $\mathrm{unif}(\mathcal{K})$. Then with probability $1-\delta'$, every element in $\mathcal{K}$ appears at least once in these samples as long as $M \geq K \log(K/\delta')$.*

Proof. Fix an element $j \in \mathcal{K}$. The probability that j appears in none of the M samples can be bounded by

$$\left(1 - \frac{1}{|\mathcal{K}|}\right)^M \leq \left(1 - \frac{1}{K}\right)^M \leq \exp\left(-\frac{M}{K}\right) \leq \frac{\delta'}{K}.$$

A union bound on every j in $\mathcal{K}$ finishes the proof. □

Lemma 7 (Concentration of Subgaussian random variables)**.** *Let $X_1 - \mu, \ldots, X_M - \mu$ be a sequence of independent 1-subgaussian random variables and $\widehat{\mu} = \frac{1}{M}\sum_{m=1}^M X_m$. Then with probability $1-\delta'$, $\delta' < \frac{1}{2}$, we have*

$$|\widehat{\mu} - \mu| \leq 2\sqrt{\frac{\log(1/\delta')}{M}}$$

Lemma 8 (Bernstein's inequality)**.** *Let $X_1, \ldots, X_M$ be a sequence of independent random variables with $X_m - \mathbb{E}[X_m] \leq b$ almost surely for every m and $v = \sum_{m=1}^M \mathrm{Var}[X_m]$. With probability $1-\delta'$, we have*

$$\sum_{m=1}^M X_m \leq \sum_{m=1}^M \mathbb{E}[X_m] + \sqrt{2v\log(1/\delta')} + \frac{2b}{3}\log(1/\delta')$$

The following lemma is a direct consequence of Bernstein's inequality, which states, in terms of the coupon collector's problem, that after drawing M coupons, one can expect with a high probability a particular coupon appears at least $Mp/2$ times if its probability of appearing is p.

Lemma 9. *Let $X_1, \ldots, X_M$ be i.i.d Bernoulli random variables so that $\mathbb{E}[X_m] = p$ for all m. With probability $1-\delta'$, we have $\sum_{m=1}^M X_m \geq Mp/2$ as long as $Mp \geq 16\log(1/\delta')$.*

Proof. Let $Y_m = 1 - X_m$ We have $\mathbb{E}[Y_m] = 1-p$. By Lemma 8, with probability $1-\delta'$, we have

$$M - \sum_{m=1}^M X_m \leq M\cdot(1-p) + \sqrt{2Mp(1-p)\log(1/\delta')} + \log(1/\delta').$$

Rearranging terms and using $Mp \geq 16\log(1/\delta')$ gives

$$\begin{aligned}\sum_{m=1}^{M} X_m &\geq Mp - \sqrt{2Mp(1-p)\log(1/\delta')} - \log(1/\delta') \\ &\geq Mp - \frac{Mp}{\sqrt{8}} - \frac{Mp}{16} \\ &\geq \frac{Mp}{2}\end{aligned}$$

□

C Proofs for Section 3

We first provide an outline of the proof in Appendix C.1 and show the complete proofs of the lemma in the corresponding subsections.

C.1 Proof of Theorem 1

To begin with, we assign adaptive target optimality (accuracy) to each block according to their block size:

$$\varepsilon_b = \max\left\{1, \frac{1}{\sqrt{r\omega(b)}}\right\}\varepsilon, \qquad \text{for } b \in [r].$$

The following lemma states that as long as $\mathcal{W}$ contains an $\widetilde{O}(\varepsilon_b)$-optimal action for each block $b \in [r]$, then the output policy is $\widetilde{O}(\varepsilon)$-optimal with high probability.

Lemma 10. *For a positive constant C, suppose for any $b \in [r]$ with $\omega(b) \geq \varepsilon/r$, $\mathcal{W}$ contains a $C\varepsilon_b$-optimal action, then π^{out} is $2(C+1)\varepsilon$-optimal for the original context-lumpable bandit problem.*

As a result, if we can prove that the precondition of Lemma 10 holds with high probability, then the correctness of Algorithm 1 follows immediately. To do so, we first argue that for every block $b \in [r]$, we have sufficiently explored the entire action set at accuracy level $n = \lceil\log(1/\varepsilon_b^2)\rceil$ in the data collection step, as stated in the next lemma.

Lemma 11. *With probability at least $1-2\delta$, for any $b \in [r]$ with $\omega(b) \geq \varepsilon/r$, we have that: for any $j \in [K]$, there exists $(i,j) \in \mathcal{D}_n$ satisfying $g(i) = b$ where $n = \lceil\log(1/\varepsilon_b^2)\rceil$.*

Intuitively, Lemma 11 says that we have tested all actions on each block $b \in [r]$ at the corresponding accuracy level $n = \lceil\log(1/\varepsilon_b^2)\rceil$. Based on Lemma 11, the following lemma states that in the second step we are able to filter out an $\widetilde{O}(\varepsilon_b)$-optimal action for block b by using the information contained in $\mathcal{D}_n$.

Lemma 12. *With probability at least $1-4\delta$, for any $b \in [r]$ with $\omega(b) \geq \varepsilon/r$, a $10\varepsilon_b\sqrt{\log(rSK/\delta)}$-optimal action of block b is added into $\mathcal{W}$ in the process of shrinking $\mathcal{D}_n$ where $n = \lceil\log(1/\varepsilon_b^2)\rceil$.*

Now we have proved the correctness of Algorithm 1. However, to obtain the desired sample complexity, it remains to show that the size of the optimal action candidate set $\mathcal{W}$ is relatively small, which we prove next.

Lemma 13. *With probability at least $1-4\delta$, $|\mathcal{W}| \leq Nr$.*

Now we are ready to prove Theorem 1 by combining all the above lemmas.

Proof of Theorem 1. The step of data collection uses

$$8\left(1+\log\frac{1}{\varepsilon^2}\right)\cdot\left(2+\log\frac{1}{\varepsilon^2}\right)\cdot\left(\log\frac{rSK}{\delta}\right)\cdot\frac{r(S+K)}{\varepsilon^2}$$

samples. By Lemma 13, the step of screening optimal action candidates uses

$$2(16^2)\left(1+\log\frac{1}{\varepsilon^2}\right)\cdot\left(\log\frac{rSK}{\delta}\right)\cdot\frac{rS}{\varepsilon^2}$$

samples and solving the simplified context-lumpable bandit problem uses

$$4\left(1+\log\frac{1}{\varepsilon^2}\right)\cdot\left(\log\frac{SK}{\delta}\right)\cdot\frac{Sr}{\varepsilon^2}$$

samples. Overall, the number of samples is bounded by

$$524\left(\log\frac{rSK}{\delta}\right)\cdot\left(1+2\log\frac{1}{\varepsilon}\right)^2\cdot\frac{r(S+K)}{\varepsilon^2}$$

By Lemma 10 and Lemma 12, π^{out} is $2\left(10\sqrt{\log(rSK/\delta)}+1\right)\varepsilon$-optimal. □

C.2 Proof of Lemma 10

Proof. We control the suboptimality of π^{out} in the following way:

$$\begin{aligned}
&\sum_{i\in[S]}\nu(i)\left(\max_{j\in[K]}A(i,j)-A(i,\pi^{\mathrm{out}}(i))\right)\\
\leq&\sum_{b\in[r]:\,\omega(b)\geq\varepsilon/r}\sum_{i\in\mathcal{B}(b)}\nu(i)\left(\max_{j\in\mathcal{W}}A(i,j)-A(i,\pi^{\mathrm{out}}(i))+C\varepsilon_b\right)+\sum_{b\in[r]:\,\omega(b)<\varepsilon/r}\omega(b)\\
\leq&\varepsilon+C\sum_{b\in[r]}\omega(b)\varepsilon_b+\varepsilon\leq 2(C+1)\varepsilon,
\end{aligned}$$

where the first inequality uses the precondition of the lemma, the second one uses the ε-optimality of π^{out} in the simplified context-lumpable bandit, and the final one follows from Cauchy-Schwarz inequality. □

C.3 Proof of Lemma 11

Proof. Fix a block $b\in[r]$. For each accuracy level, recall in the step of data collection, we sample $L=r(S+K)\widetilde{\lg}/\varepsilon^2$ contexts i.i.d. from ν. By Lemma 9, with probability $1-\frac{\delta}{r}$, at least $L\omega(b)/2$ of them are from block b as

$$L\omega(b)\geq\frac{L\varepsilon}{r}\geq\widetilde{\lg}\geq 16\log(r/\delta).$$

Further, recall that we define accuracy level $n=\lceil\log(1/\varepsilon_b^2)\rceil$. We first show that this n is well defined, that is, $n\geq 1$. Indeed, we have

$$\frac{1}{\varepsilon_b^2}=\frac{1}{\varepsilon^2/(r\omega(b))}\geq\frac{\varepsilon}{\varepsilon^2}=\frac{1}{\varepsilon}\geq 2$$

as long as $\varepsilon\leq\frac{1}{2}$. Since we add a context-action pair into $\mathcal{D}_n$ once we have collected

$$2^n=2^{\lceil\log(1/\varepsilon_b^2)\rceil}<2^{\log(1/\varepsilon_b^2)+1}=2/\varepsilon_b^2$$

samples for estimating its reward. Note that in the end there are at most $|\mathcal{B}(b)|(2^n-1)$ samples from block b unused. Thus, with probability $1-\frac{\delta}{r}$, there are at least $L\omega(b)/2-|\mathcal{B}(b)|(2^n-1)$ samples used. Therefore, the number of context-action pairs, where the contexts are from block b, that are added into $\mathcal{D}_n$ is at least

$$\begin{aligned}
\frac{L\omega(b)/2-|\mathcal{B}(b)|(2^n-1)}{2^n}&\geq\frac{L\omega(b)/2}{2/\varepsilon_b^2}-S && (2^n\leq 2/\varepsilon_b^2\text{ and }|\mathcal{B}(b)|\leq S)\\
&\geq\frac{L\varepsilon^2/(2r\varepsilon_b^2)}{2/\varepsilon_b^2}-S && (\text{by definition of }\varepsilon_b)\\
&=\geq 16(S+K)\log(rSK/\delta)-S && (\text{the value of }L)\\
&\geq K\log(rSK/\delta).
\end{aligned}$$

Conditioned on this event, with probability $1-\frac{\delta}{r}$, for any $j\in[K]$, there exists $(i,j)\in\mathcal{D}_n$ satisfying $g(i)=b$ by Lemma 6. Therefore, the lemma holds for block b with probability at least $1-\frac{2\delta}{r}$. We complete the proof by a union bound on all blocks. □

C.4 Proof of Lemma 12

Proof. Denote by i the first context being removed from $\mathcal{D}_n$, which satisfies $g(i) = b$. By the rule of shrinking $\mathcal{D}_n$, we have

$$|\widetilde{A}_n(i, j^\star) - \widehat{A}_n(i^\star, j^\star)| \le 4\sqrt{\frac{\log(rSK/\delta)}{2^n}}.$$

By Lemma 7 and a union bound on all pairs in $\mathcal{D}_n$, with probability $1 - 2\delta/r$, we have

$$|\widetilde{A}_n(i'', j'') - A(i'', j'')| \le \frac{1}{2}\sqrt{\frac{\log(rSK/\delta)}{2^n}} \quad \text{and} \quad |\widehat{A}(i'', j'') - A_n(i'', j'')| \le \frac{1}{2}\sqrt{\frac{\log(rSK/\delta)}{2^n}} \tag{5}$$

for every $(i'', j'') \in \mathcal{D}_n$. Therefore, we have

$$|A(i, j^\star) - A(i^\star, j^\star)| \le 5\sqrt{\frac{\log(rSK/\delta)}{2^n}}, \tag{6}$$

By Lemma 11, we know that for any $j \in [K]$, there exists $(i', j) \in \mathcal{D}_n$ satisfying $(g(i'), j) = (b, j)$, before we remove context i. Therefore, by the definition of $(i^\star, j^\star)$, for any $j \in [K]$, there exists $(i', j) \in \mathcal{D}_n$ satisfying $g(i') = b$ and

$$\widehat{A}_n(i^\star, j^\star) \ge \widehat{A}_n(i', j),$$

which, again by Eq. (5) with probability $1 - 2\delta/r$

$$A(i^\star, j^\star) \ge \max_j \mu(b, j) - \sqrt{\frac{\log(rSK/\delta)}{2^n}}. \tag{7}$$

By combining Eq. (6) and Eq. (7), we conclude that $j^\star$ is a $10\sqrt{\log(rSK/\delta)}\varepsilon_b$-optimal action for block b because

$$5\sqrt{\frac{\log(rSK/\delta)}{2^n}} \ge 10\sqrt{\frac{\log(rSK/\delta)}{1/\varepsilon_b^2}} = 10\sqrt{\log(rSK/\delta)}\varepsilon_b$$

This completes the proof by a union bound on all blocks. □

C.5 Proof of Lemma 13

Proof. It suffices to show that for each accuracy level n, we will add at most r actions into $\mathcal{W}$ before shrinking $\mathcal{D}_n$ to $\emptyset$. Below we prove a stronger argument: each time we shrink $\mathcal{D}_n$, we will remove all the contexts from at least one block from $\mathcal{D}_n$.

Denote by (i, j) an arbitrary context-action pair from $\mathcal{D}_n$ such that $g(i) = g(i^\star)$, then by Eq. (5), we have

$$\left|\widetilde{A}_n(i, j^\star) - \widehat{A}_n(i^\star, j^\star)\right| \le \left|\widetilde{A}_n(i, j^\star) - A(i, j^\star)\right| + \left|A(i^\star, j^\star) - \widehat{A}_n(i^\star, j^\star)\right| \le 4\sqrt{\frac{\log(rSK/\delta)}{2^n}},$$

which implies all context-action pairs with context from block $g(i^\star)$ will be removed from $\mathcal{D}_n$. □

C.6 Proof of Theorem 2

Proof. By Lemma 9 and a union bound over $[S]$, we have that with probability $1 - \delta$ for all $i \in [S]$:

$$\frac{1}{2}\left(\nu(i) - \frac{\log(S/\delta)}{J}\right) \le \hat{\nu}(i) \le 2\left(\nu(i) + \frac{\log(S/\delta)}{J}\right),$$

which implies that $\nu(i) \ge \varepsilon/(4S)$ for $\hat{\nu}(i) \ge \varepsilon/S$ and $\nu(i) \le 3\varepsilon/S$ for $\hat{\nu}(i) < \varepsilon/S$. By definition, for any $l \in [L-1]$, $\mathcal{X}_l$ consists of contexts such that $\hat{\nu}(i) > 2^{-l-1} \ge \varepsilon/S$. As a result, for any $l \in [L-1]$, by Lemma 9, we have that at least $N\nu(\mathcal{X}_l)/2$ out of N samples corresponds to $\mathcal{X}_l$ and thus can be used in learning π_l by executing Algorithm 1, where $\nu(\mathcal{X}_l) := \sum_{i \in \mathcal{X}_l} \nu(i) \ge \varepsilon/(4S)$.

Moreover, notice that inside each $\mathcal{X}_l$, the conditional context distribution is almost uniform, so we can invoke Theorem 1 to obtain that π_l is

$$\frac{C\varepsilon}{\sqrt{\nu(\mathcal{X}_l)}}\text{-optimal}$$

over context $\mathcal{X}_l$, where $C = 2\sqrt{2}(10\sqrt{\log(rSK/\delta)} + 1)$. This means that,

$$\sum_{i\in\mathcal{X}_l} \frac{\nu(i)}{\nu(\mathcal{X}_l)} \left(\max_j A(i,j) - A(i,\pi_l(i))\right) = \frac{C\varepsilon}{\sqrt{\nu(\mathcal{X}_l)}}.$$

As a result, the total suboptimality of π^{out} can be upper bounded as following

$$\begin{aligned}
&\sum_{i\in[S]} \nu(i) \left(\max_{j\in[K]} A(i,j) - A(i,\pi^{\text{out}}(i))\right) \\
\leq &\sum_{l\in[L-1]} \sum_{i\in\mathcal{X}_l} \nu(i) \left(\max_j A(i,j) - A(i,\pi_l(i))\right) + \sum_{i\in\mathcal{X}_L} \nu(i) \\
\leq &C \sum_{l\in[L-1]} \nu(\mathcal{X}_l) \times \frac{\varepsilon}{\sqrt{\nu(\mathcal{X}_l)}} + \varepsilon = C\sqrt{L}\varepsilon,
\end{aligned}$$

where the final equality uses Cauchy-Schwartz inequality and $L \leq \log(S/\varepsilon) + 1$. □

C.7 Proof of Theorem 3

Proof. Since $r \leq S$, we have

$$\Omega(\min\{r(S+K), SK\}) = \begin{cases} \Omega(SK), & r \geq K, \\ \Omega(rS), & r < K \;\&\; S \geq K, \\ \Omega(rK), & r < K \;\&\; S < K. \end{cases}$$

Case (1) We construct the following hard instance:

- The reward after pulling action $j \in [K]$ in block $b \in [r]$ is sampled from distribution $\text{Bernoulli}(1/2 + \varepsilon\mathbb{1}(b = j))$.
- The context distribution is uniform over $[S]$. For each $i \in [S]$, we sample $g(i)$ uniformly at random from $[K]$.

The above instance is the standard one commonly used in proving lower bounds for contextual bandit with S contexts and K arms [e.g., Lattimore and Szepesvári, 2020]. And learning an ε-optimal policy for the above hard instance requires at least $\Omega(SK/\varepsilon^2)$ samples.

Case (2) We only need to slightly modify the above hard instance:

- The reward after pulling action $j \in [K]$ in block $b \in [r]$ is sampled from distribution $\text{Bernoulli}(1/2 + \varepsilon\mathbb{1}(b = j))$.
- The context distribution is uniform over $[S]$. For each $i \in [S]$, we sample $g(i)$ uniformly at random from $[r]$.

The above modified problem is equivalent to the original one with S contexts but r arms. As a result, a lower bound of form $\Omega(Sr/\varepsilon^2)$ holds.

Case (3) Similarly, we slightly modify the first hard instance:

- For each block $b \in [r]$, sample $j_b^\star$ uniformly at random from $[K]$. The reward after pulling action $j \in [K]$ in block $b \in [r]$ is sampled from distribution $\mathrm{Bernoulli}(1/2 + \varepsilon \mathbb{1}(j = j_b^\star))$.
- The context distribution is uniform over $[r]$ and $g(i) = \min\{i, r\}$.

The above modified problem is equivalent to the original one with K arms but r contexts. As a result, a lower bound of form $\Omega(Kr/\varepsilon^2)$ holds.

□

D Proof of Theorem 4

In this section, we first show key lemmas to prove Theorem 4. The proofs of the lemmas are deferred to Appendix E.

In the following discussion, we consider a single phase (i.e. fix an error ε_h). Similar to the analysis of other phased elimination algorithms, we have to show that in a phase specified by error level ε_h, with high probability, (i) the optimal arm is not eliminated and (ii) all $\omega(\widetilde{\varepsilon}_h)$-suboptimal arms are eliminated, that is, all arms in $\mathrm{GOOD}_h(\mathcal{C})$ for all $\mathcal{C}$ are $O(\widetilde{\varepsilon}_h)$-optimal.

The following lemma is the counterpart of Lemma 11, which ensures that every arm is played in every block.

Lemma 14. *With probability at least $1 - 2\delta_h$, for any cluster $\mathcal{C} \in \mathcal{P}_h$ and any block b so that $\mathcal{B}(b) \subseteq \mathcal{C}$, we have that: for any $j \in \mathit{GOOD}_h(\mathcal{C})$ there exists $(i, j) \in \mathcal{D}_h$ satisfying $g(i) = b$.*

The above lemma ensures that every block has a least one context i assigned to include (i, j) in $\mathcal{D}_h$. Thus, every arm is explored for every block.

Next, we define an event $\mathcal{E}_h$ under which the estimates $\widehat{A}_h$ are good:

$$\mathcal{E}_h = \left\{ |\widehat{A}_h(i,j) - A(i,j)| \le \frac{\widetilde{\varepsilon}_h}{4},\ \forall (i,j) \in \mathcal{D}_h \right\}.$$

The level of precision $\frac{\widetilde{\varepsilon}_h}{4}$ is more accurate than the elimination step and will be helpful in the analysis. The following lemma states that $\mathcal{E}_h$ is a high probability event.

Lemma 15. *Event $\mathcal{E}_h$ holds with probability $1 - \delta_h$.*

Next, let $j_b = \arg\max_{j \in [K]} \mu(b, j)$ be the optimal arm in block b. The next lemma says that the optimal arm j_b is not eliminated during the execution of the algorithm.

Lemma 16. *Assume action $j_b \in \mathit{GOOD}_h(\mathcal{C})$ for every block $b \in [r]$, and its corresponding partition $\mathcal{P}_h \ni \mathcal{C} \supseteq \mathcal{B}(b)$. Then for every block $b \in [r]$, j_b is not eliminated from $\mathit{GOOD}_{h+1}(\mathcal{C})$ with probability at least $1 - 3\delta_h$.*

The high-level idea of the proof is that the error of the estimated mean is smaller than $\widetilde{\varepsilon}_h$, so $\widehat{A}_t(i, j_b)$ will not be much worse than other arms, given that its true mean is largest. The next lemma shows that arms in $\mathrm{GOOD}_{h+1}(\mathcal{C})$ are all $O(\varepsilon_h)$-optimal. Formally, we say an arm j in a block b is ε-optimal if $\max_{j'} \mu(b, j') - \mu(b, j) \le \varepsilon$. Similarly, we say an arm j in a block b is ε-suboptimal if $\max_{j'} \mu(b, j') - \mu(b, j) \ge \varepsilon$.

Lemma 17. *For any block $b \in [r]$ and its corresponding cluster $\mathcal{P}_h \ni \mathcal{C} \supseteq \mathcal{B}(b)$, all $3\widetilde{\varepsilon}_h$-suboptimal arms in block b are eliminated in $\mathit{GOOD}_{h+1}(\mathcal{C})$ with probability at least $1 - 3\delta_h$. Consequently, only $6\widetilde{\varepsilon}_h$-optimal arms in block b are played in phase $h + 1$ for every context in block b.*

Finally, we need to argue that Algorithm 5 is not called too many times. To show this, we first provide a general guarantee of Algorithm 5.

Lemma 18. *Suppose we have context $i_u \in \mathcal{B}(b_u) \subseteq \mathcal{C}$ in block b_u and context $i_l \in \mathcal{B}(b_l) \subseteq \mathcal{C}$ in block b_l so that*

$$A(i_u, j) - A(i_l, j) = \mu(b_u, j) - \mu(b_l, j) > \frac{3r}{2}\sqrt{\widetilde{\lg}} \cdot \varepsilon'.$$

Then Algorithm 5 separates $\mathcal{B}(b_u)$ and $\mathcal{B}(b_l)$ perfectly with probabilty at least $1-2\delta'$. In other words, there exist indices c_u and c_l, $c_u \neq c_l$, such that $\mathcal{B}(b_u) \subseteq \mathcal{P}_{c_u}$ and $\mathcal{B}(b_l) \subseteq \mathcal{P}_{c_l}$.

Consequently, with a high probability, Algorithm 4 makes progress in terms of clustering contexts every time calling Algorithm 5 and the number of calls is bounded by r, as shown in the following lemma.

Lemma 19. *When Algorithm 5 is called by Algorithm 4, it separates at least two blocks and never separates contexts in the same block with probability at least $1-5\delta_h$. Consequently, Algorithm 5 is called at most r times.*

We are now ready to bound regret.

Proof of Theorem 4. Since there are at most $O(\log_T)$ phases, it suffices to bound regret at a single phase h. Conditioned on all the high probability good events in the previous lemmas, we first bound the total number of timesteps spent in phase h. In the data collection stage, we use at most $L_h = \frac{r(S+K)\widetilde{\lg}_h}{\varepsilon_h^2}$ samples; as for the clustering stage, by Lemma 19, we know the total length of executing Algorithm 5 is at most

$$rL' = rS\widetilde{\lg}/\varepsilon'^2 = 16Sr^3\widetilde{\lg}/\varepsilon_h^2 = \widetilde{O}\left(\frac{r^3 S}{\varepsilon_h^2}\right).$$

Thus, the length of phase h is the minimum of T and $\widetilde{O}\left(\frac{r^3(S+K)}{\varepsilon_h^2}\right)$. Therefore, by Lemma 17, the regret is at most (recall that $\widetilde{\varepsilon}_h = \sqrt{\log_h}\cdot\varepsilon_h = \widetilde{O}(\varepsilon_h)$)

$$\begin{aligned}
6\widetilde{\varepsilon}_h \cdot \min\left\{T, \widetilde{O}\left(\frac{r^3(S+K)}{\varepsilon_h^2}\right)\right\} &= \min\left\{6T\widetilde{\varepsilon}_h, \widetilde{O}\left(\frac{r^3(S+K)}{\varepsilon_h}\right)\right\} \\
&= \min\left\{\widetilde{O}\left(T\varepsilon_h\right), \widetilde{O}\left(\frac{r^3(S+K)}{\varepsilon_h}\right)\right\} \\
&= \widetilde{O}\left(\sqrt{r^3(S+K)T}\right).
\end{aligned}$$

On the other hand, the bad events happen with probability $O(\delta_h) = O(\varepsilon_h^2/(r^3SK))$. In this case the regret contributes at most $\widetilde{O}(1)$. □

E Missing Proofs in Appendix D

E.1 Proof of Lemma 14

Proof. Note that under the uniform block assumption, we have $\omega(b) = \frac{1}{r} \geq \frac{\varepsilon_h}{r}$. Thus, the proof follows directly from the proof of Lemma 11 when $\varepsilon_b = \varepsilon_h$, $\delta = \varepsilon_h^2$, and $n = n_h$. The only difference is when applying Lemma 6, Lemma 11 uses $\mathcal{K} = [K]$ but we need $\mathcal{K} = \text{GOOD}_h(\mathcal{C})$ here. □

E.2 Proof of Lemma 15

Proof. Fix an (i,j) pair. Applying Lemma 7, we have with probability $1-\frac{\delta_h}{SK}$,

$$\left|\widehat{A}_h(i,j) - A(i,j)\right| \leq 2\sqrt{\frac{\log(SK/\delta_h)}{2^{(n_h)}}} \leq \frac{\widetilde{\varepsilon}_h}{4}.$$

We complete the proof by applying a union bound on all (i,j) pairs in $\mathcal{D}_h$. □

E.3 Proof of Lemma 16

Proof. We prove by contradiction. Assume that j_b is removed from $\text{GOOD}_h(\mathcal{C})$. Let

$$j' = \operatorname*{arg\,max}_{j \in \text{GOOD}_h(\mathcal{C})} \overline{\mu}_h(\mathcal{C}, j)$$

be the action achieving the highest empirical mean. By Lemma 14 there exists a context $i' \in \mathcal{B}(b)$ so that $(i', j') \in \mathcal{D}_h$. Moreover, the assumption that j_b is eliminated implies that the condition at Line 7 of Algorithm 4 does not hold for the current partition $\mathcal{P}_h$, which further implies that there exists context $\bar{i} \in \mathcal{C}$ so that

$$\widehat{A}_h(\bar{i}, j') - \widehat{A}_h(i', j') = \overline{\mu}_h(\mathcal{C}, j') - \widehat{A}_h(i', j') < \widetilde{\varepsilon}_h \ .$$

On the other hand, by the assumption that j_b is eliminated, again by Lemma 14 we have that there exists a context $i'' \in \mathcal{B}(b)$, $(i'', j_b) \in \mathcal{D}_h$ such that

$$\overline{\mu}_h(\mathcal{C}, j') - \widehat{A}_h(i'', j_b) \geq \overline{\mu}_h(\mathcal{C}, j') - \overline{\mu}_h(\mathcal{C}, j_b) > 2\widetilde{\varepsilon}_h \ .$$

Combining two inequalities, we have

$$\widehat{A}_h(i'', j_b) + \widetilde{\varepsilon}_h < \overline{\mu}_h(\mathcal{C}, j') - \widetilde{\varepsilon}_h < \widehat{A}_h(i', j').$$

This means that $\mu(b, j_b) + \frac{\widetilde{\varepsilon}_h}{2} < \mu(b, j')$ under $\mathcal{E}_h$, which contradicts the optimality of j_b. Therefore, we conclude that j_b is not eliminated. □

E.4 Proof of Lemma 17

Proof. By Lemma 16, j_b is not eliminated for any block $b \in [r]$ with probability at least $1 - 3\delta_h$. Thus, for any block $b \in [r]$ and arm $j \in \text{GOOD}_h(\mathcal{C})$ so that $\mu(b, j_b) - \mu(b, j) > 3\widetilde{\varepsilon}_h$, we have

$$\max_{j'} \overline{\mu}_h(\mathcal{C}, j') - \overline{\mu}_h(\mathcal{C}, j) \geq \overline{\mu}_h(\mathcal{C}, j_b) - \overline{\mu}_h(\mathcal{C}, j) \geq \mu(b, j_b) - \mu(b, j) - 2 \cdot \frac{\widetilde{\varepsilon}_h}{4} > 2\widetilde{\varepsilon}_h \ .$$

Therefore, j is eliminated at Line 14. □

E.5 Proof of Lemma 18

Proof. With probability $1 - \frac{\delta_h}{S}$, context i receives at least $2/\varepsilon'^2 \geq 2^{n'}$ samples by Lemma 9. Thus, $\widehat{A}(i, j)$ is well defined for every $i \in \mathcal{C}$ with probability $1 - \delta_h$. Moreover, by Lemma 7 and a union bound, we have for every context i, with probability at least $1 - \delta_h$,

$$\left| A(i, j) - \widehat{A}(i, j) \right| \leq \frac{\sqrt{\widetilde{\lg}} \cdot \varepsilon'}{4} \tag{8}$$

Clearly, we have $\widehat{A}(i_u) \geq \widehat{A}(i_l)$ under Eq. (8). Consequently, to simplify the notation, we do the following modification on labels of contexts and blocks. First, we restrict the game to $\mathcal{C}$, where there are S' contexts and r' blocks; also, we relabel contexts so that $i_u = 1$, $i_l = S'$, and

$$\widehat{A}(i_u) = \widehat{A}(1) \geq \widehat{A}(2) \geq \cdots \geq \widehat{A}(S' - 1) \geq \widehat{A}(S') = \widehat{A}(i_l).$$

Finally, given a context $i \in [S']$, we define

$$\overline{\mu}(b) = \max_{i' \in [S'], g(i') = b} \widehat{A}(i', j) \quad \text{and} \quad \underline{\mu}(b) = \min_{i' \in [S'], g(i') = b} \widehat{A}(i', j)$$

for its block $b = g(i)$ and we relabel blocks so that $b_u = 1 \leq g(i) \leq r' = b_l$ and

$$\overline{\mu}(b_u) = \overline{\mu}(1) \geq \overline{\mu}(b_u - 1) \geq \cdots \geq \overline{\mu}(b_l + 1) \geq \overline{\mu}(r') = \overline{\mu}(b_l).$$

It is not hard to see that this modification is without loss of generality. We show next that there exists a context i, $i_u < i \leq i_l$, such that Line 7 of Algorithm 5 holds, that is,

$$\widehat{A}(i - 1, j) - \widehat{A}(i, j) \geq \sqrt{\widetilde{\lg}} \cdot \varepsilon'. \tag{9}$$

We prove this by contradiction. Assume Eq. (9) doesn't hold for any k.

Then we have

$$\begin{aligned}
A(i_u, j) - A(i_l, j) &\le |A(i_u, j) - \overline{\mu}(b_u)| + |A(i_l, j) - \overline{\mu}(b_l)| + \overline{\mu}(b_u) - \overline{\mu}(b_l) \\
&= \frac{\sqrt{\widetilde{\lg}} \cdot \varepsilon}{4} + \frac{\sqrt{\widetilde{\lg}} \cdot \varepsilon}{4} + \sum_{b=b_u+1}^{b_l} \overline{\mu}(i_{b-1}) - \overline{\mu}(i_b) \\
&< \frac{\sqrt{\widetilde{\lg}} \cdot \varepsilon}{2} + \sum_{b=b_u+1}^{b_l} \overline{\mu}(i_{b-1}) - \underline{\mu}(i_b) + \sqrt{\widetilde{\lg}} \cdot \varepsilon' \\
&\le \frac{\sqrt{\widetilde{\lg}} \cdot \varepsilon}{2} + \frac{3}{2} \cdot (r-1)\sqrt{\widetilde{\lg}} \cdot \varepsilon' \\
&< \frac{3}{2} \cdot r\sqrt{\widetilde{\lg}} \cdot \varepsilon',
\end{aligned}$$

which contradicts the condition that $A(i_u, j) - A(i_l, j) \ge \frac{3r}{2}\sqrt{\widetilde{\lg}} \cdot \varepsilon$. Therefore, we conclude that Eq. (9) holds for some k. □

E.6 Proof of Lemma 19

Proof. The proof follows directly from Lemma 18 and the fact that Algorithm 4 use $\varepsilon' = \frac{\varepsilon_h}{4r}$ and thus

$$A(\bar{i}, j) - A(\underline{i}, j) \ge \frac{\widetilde{\varepsilon}_h}{2} \ge \frac{\sqrt{\widetilde{\lg}_h} \cdot \varepsilon_h}{2} > \frac{3}{2} \cdot r\sqrt{\widetilde{\lg}} \cdot \varepsilon',$$

which satisfies the condition of Lemma 18. □

F Non-uniform Context Distribution

We show a reduction to problems with approximately uniform context distributions. The cost of this reduction is an extra $\widetilde{O}(\sqrt{ST})$ additive regret and an extra $O(\log(ST))$ multiplicative factor in regret. The idea is to learn the context distribution in the first $\widetilde{O}(\sqrt{ST})$ timesteps. With high probability, we can estimate $\nu(i)$ for any context i with a constant multiplicative error as long as $\nu(i) = \widetilde{\Omega}(1/\sqrt{ST})$.Then we split the contexts into several buckets so that contexts within the same bucket have the same probability up to a constant factor. For contexts i with $\nu(i) = o(1/\sqrt{ST})$, we can not estimate the probability properly but we can simply ignore such contexts and suffer regret at most $O(T \cdot S \cdot 1/\sqrt{ST}) = O(\sqrt{ST})$. We then run the algorithm that handles uniform context distribution for each bucket separately. Since there are $O(\log(ST))$ buckets, the overall regret is $O(\log(ST))$ times maximum regret over all subsets (or $\sqrt{\log(ST)}$ with refined analysis using a Cauchy–Schwarz inequality).

G Proof of Theorem 5

Define

$$\varepsilon_{h,b} = \max\left\{1, \frac{1}{r\omega(b)}\right\}\varepsilon_h, \quad \text{for } b \in [r].$$

Also, for every phase h and every level n, we define

$$\widetilde{\varepsilon}_{h,n} = \sqrt{\frac{\widetilde{\lg}_h}{2^n}}.$$

Next, we show the counterpart of Lemma 14 in the non-uniform case.

Lemma 20. *With probability at least $1 - 2\delta_h$, for any cluster $\mathcal{C} \in \mathcal{P}_h$, any block b with $\mathcal{B}(b) \subseteq \mathcal{C}$, we have that: for any level $n \le \lceil \log(1/\varepsilon_{h,b}^2) \rceil$, action $j \in \mathit{GOOD}_{h,n}(\mathcal{C})$, there exists $(i, j) \in \mathcal{D}_h$ satisfying $g(i) = b$.*

Proof. Fix a block $b \in [r]$. For each accuracy level, recall in the step of data collection, we sample $L = r(S+K)\widetilde{\lg}_h 2^n$ contexts i.i.d. from ν. By Lemma 9, with probability $1 - \frac{\delta_h}{r}$, at least $L\omega(b)/2$ of them are from block b as

$$L\omega(b) \geq \frac{8L}{S} \geq \widetilde{\lg} \geq 16\log(r/\delta_h),$$

where the first inequality comes from the near-uniform context distribution assumption. Since we add a context-action pair into $\mathcal{D}_n$ once we have collected

$$2^n \leq 2^{\lceil \log(1/\varepsilon_{h,b}^2) \rceil} < 2^{\log(1/\varepsilon_{h,b}^2)+1} = 2/\varepsilon_{h,b}^2$$

samples for estimating its reward. Note that in the end, there are at most $|\mathcal{B}(b)|(2^n - 1)$ samples from block b unused. Thus, with probability $1 - \frac{\delta_h}{S}$, the number of the context-action pairs, where the contexts are from block b, that are added into $\mathcal{D}_n$ is at least

$$\begin{aligned}
\frac{L\omega(b)/2 - |\mathcal{B}(b)|(2^n-1)}{2^n} &\geq \frac{L\omega(b)/2}{2/\varepsilon_{h,b}^2} - S && (2^n \leq 2/\varepsilon_{h,b}^2 \text{ and } |\mathcal{B}(b)| \leq S) \\
&\geq \frac{L\varepsilon_h/(2r\varepsilon_{h,b})}{2/\varepsilon_{h,b}^2} - S && (\text{by definition of } \varepsilon_{h,b}) \\
&= 16(S+K)\log(rSK/\delta_h) - S && (\text{the value of } L) \\
&\geq K\log(rSK/\delta_h).
\end{aligned}$$

Conditioned on this event, with probability $1 - \frac{\delta_h}{r}$, for any $j \in \text{GOOD}_{h,n}(\mathcal{C})$, there exists $(i,j) \in \mathcal{D}_n$ satisfying $g(i) = b$ by Lemma 6. Therefore, the lemma holds for block b with probability at least $1 - \frac{2\delta_h}{r}$ We complete the proof by a union bound on all blocks. □

Now define a good event $\mathcal{E}_h$ as

$$\mathcal{E}_h = \left\{ \left| \widehat{A}_{h,n}(i,j) - A(i,j) \right| \leq \frac{1}{4}\widetilde{\varepsilon}_{h,n},\ \forall (i,j) \in \mathcal{D}_h,\, n \in [N_h] \right\}.$$

Lemma 21. *Event $\mathcal{E}_h$ holds with probability $1 - \delta_h$.*

Proof. Fix an (i,j) pair and level n. Applying Lemma 7, we have with probability $1 - \frac{\delta_h}{SKN_h}$,

$$\left| \widehat{A}_{h,n}(i,j) - A(i,j) \right| \leq 2\sqrt{\frac{\log(SKN_h/\delta_h)}{2^n}} \leq \frac{1}{4}\widetilde{\varepsilon}_{h,n}.$$

We complete the proof by applying a union bound on all (i,j) pairs in $\mathcal{D}_h$ and all levels $n \in [N_h]$. □

In the following, we present and then prove the counterpart of Lemma 16 for Algorithm 6.

Lemma 22. *Assume action $j_b \in \text{GOOD}_{h,n}(\mathcal{C})$ for $b \in [r]$ with $\omega(b) \geq \frac{2\varepsilon_h}{r}$, and its corresponding cluster $\mathcal{P}_h \ni \mathcal{C} \supseteq \mathcal{B}(b)$. Then j_b is not eliminated from $\text{GOOD}_{h+1,n}(\mathcal{C})$ with probability at least $1 - 3\delta_h$ for any level $n \leq \lceil \log(1/\varepsilon_{h,b}^2) \rceil$.*

Proof. We prove by induction on n. The base case is $n = 1$, which satisfies the condition of Lemma 20 as

$$\log\left(\frac{1}{\varepsilon_{h,b}^2}\right) \geq \log\left(\frac{\omega(b)^2 r^2}{\varepsilon_h^2}\right) \geq \log(4) \geq 1 = n.$$

For the inductive step we prove by contradiction. Assume that j_b is eliminated in $\text{GOOD}_{h,n+1}(\mathcal{C})$ but not eliminated in $\text{GOOD}_{h,n}(\mathcal{C})$ for $n \geq 2$. This means that the following inequality hold:

$$\max_{j' \in \text{GOOD}_{h,n}(\mathcal{C})} \overline{\mu}_{h,n}(\mathcal{C}, j') - \overline{\mu}_{h,n}(\mathcal{C}, j_b) > 2\widetilde{\varepsilon}_{h,n}$$

Let $j' = \arg\max_{j \in \text{GOOD}(n,\mathcal{C})} \overline{\mu}_{h,n}(\mathcal{C}, j)$. By Lemma 20 there exists a context $i' \in \mathcal{B}(b)$ so that $\psi(n, i') \ni j'$ as $n \leq \log(1/\varepsilon_{h,b}^2)$. Moreover, the assumption that j_b is eliminated implies that the condition at Line 8 does not hold for the current partition $\mathcal{P}_h$, which further implies that

$$\overline{\mu}_{h,n}(\mathcal{C}, j') - \widehat{A}_{h,n}(i', j') < \widetilde{\varepsilon}_{h,n}$$

On the other hand, by the assumption that j_b is eliminated, again by Lemma 20 we have that there exists a context $i'' \in \mathcal{B}(b)$, $\psi(n, i'') \ni j_b$ such that

$$\overline{\mu}_{h,n}(\mathcal{C}, j') - \widehat{A}_{h,n}(i'', j_b) > 2\widetilde{\varepsilon}_{h,n}$$

Combining two inequalities, we have $\widehat{A}_{h,n}(i'', j_b) + \widetilde{\varepsilon}_{h,n} < \widehat{A}_{h,n}(i', j')$. This means that $\mu(b, j_b) + \frac{\widetilde{\varepsilon}_{h,n}}{2} < \mu(b, j')$ by Lemma 21, which contradicts the optimality of j_b. Therefore, we conclude that j_b is not eliminated. □

Now we present a key lemma similar to Lemma 17.

Lemma 23. *For any block $b \in [r]$ with $\omega(b) \geq \frac{2\varepsilon_h}{r}$ and its corresponding cluster $\mathcal{P}_h \ni \mathcal{C} \supseteq \mathcal{B}(b)$, all $3\widetilde{\varepsilon}_{h,n}$-suboptimal arms in block b are eliminated in GOOD$_{h+1,n}(\mathcal{C})$ for any level $n \leq \lceil \log(1/\varepsilon_{h,b}^2) \rceil$ with probability at least $1 - 3\delta_h$. Consequently, only $6\widetilde{\varepsilon}_{h,n}$-optimal arms in block b are played in phase $h+1$ for every context in block b.*

Proof. By Lemma 22, j_b is not eliminated with high probability, so for every pair of block $b \in [r]$ with $\omega(b) \geq \frac{2\varepsilon_h}{r}$ and arm $j \in \text{GOOD}_{h,n}(\mathcal{C})$ so that $\mu(b, j_b) - \mu(b, j) > 3\widetilde{\varepsilon}_{h,n}$, we have

$$\begin{aligned}\max_{j'} \overline{\mu}_{h,n}(\mathcal{C}, j') - \overline{\mu}_{h,n}(\mathcal{C}, j) &\geq \overline{\mu}_{h,n}(\mathcal{C}, j_b) - \overline{\mu}_{h,n}(\mathcal{C}, j) \geq \mu(b, j_b) - \mu(b, j) - \frac{\widetilde{\varepsilon}_{h,n}}{2} \\ &\geq 2.5 \cdot \widetilde{\varepsilon}_{h,n} > 2\widetilde{\varepsilon}_{h,n}\end{aligned}$$

Therefore, j is eliminated in GOOD$_{h,n}(\mathcal{C})$, and thus eliminated in GOOD$_{h+1,n'}(\mathcal{C})$ for $n' > n$. □

G.1 Proof of Theorem 5

Proof. Fix a phase h and a level n. It suffices to bound regret within a single pair (h, n). For $b \in [r]$, let $n_b = \lceil \log(1/\varepsilon_{h,n}^2) \rceil$. By Lemma 23, if $n > n_b$, all $6\widetilde{\varepsilon}_{h,n_b}$-suboptimal arms are eliminated from GOOD$_{h,n}(\mathcal{C})$ for any cluster $\mathcal{C} \in \mathcal{P}_h$. Therefore, regret of playing an action from GOOD$_{h,n}(\mathcal{C})$ is $6\widetilde{\varepsilon}_{h,n_b}$. In this case, regret is bounded by

$$\begin{aligned}\sum_{b \in [r]} r(S+K)2^{(n+h)/2} \cdot \omega(b) \cdot 6\widetilde{\varepsilon}_{h,n_b} &\leq \sum_{b \in [r]} r(S+K)2^{(n+h)/2} \cdot \omega(b) \cdot 6\sqrt{\frac{\widetilde{\lg}_h}{2^{n_b}}} \\ &\leq 6\sqrt{\widetilde{\lg}_h} \sum_{b \in [r]} r(S+K)2^{(n+h)/2} \cdot \omega(b) \cdot \varepsilon_{h,b} \\ &\leq 6\sqrt{\widetilde{\lg}_h} \sum_{b \in [r]} r(S+K)2^{(n+h)/2} \cdot \omega(b) \cdot \frac{\varepsilon_h}{r\omega(b)} \\ &\leq 6\sqrt{\widetilde{\lg}_h} r(S+K)2^{(n+h)/2} = \widetilde{O}\left(r(S+K)2^{h/2}\right)\end{aligned}$$

If $n \leq n_b$, a similar argument shows that regret of playing an action from GOOD$_{h,n}(\mathcal{C})$ is $6\widetilde{\varepsilon}_{h,n}$. Therefore, regret is bounded by

$$\begin{aligned}\sum_{b \in [r]} r(S+K)2^{(n+h)/2 \cdot} \omega(b) \cdot 6\widetilde{\varepsilon}_{h,n} &\leq 6 \sum_{b \in [r]} \omega(b) \cdot r(S+K)2^{(n+h)/2} \sqrt{\frac{\widetilde{\lg}_h}{2^n}} \\ &\leq 6\sqrt{\widetilde{\lg}_h} r(S+K)2^h = \widetilde{O}\left(r(S+K)2^{h/2}\right)\end{aligned}$$

We conclude the overall regret is $\widetilde{O}\left(\sqrt{r(S+K)T}\right)$ for the data collection stage by noting that $T \geq r(S+K)2^{(h-1)}$ as phase $h-1$ is executed completely. The same argument holds for analyzing the clustering stage when replacing r with r^3, so we conclude the regret is bounded by $\widetilde{O}\left(\sqrt{r^3(S+K)T}\right)$. □

H Omitted Details in Section 5

In this section we discuss how to solve the more general low-rank bandit problem. Note that A has rank r if and only if there exist vectors $w_1, \ldots, w_S \in \mathbb{R}^r$ and $v_1, \ldots, v_K \in \mathbb{R}^r$ so that $A(i,j) = w_i^\top v_j$. We first consider $r = 1$ and assume that every w_i is non-negative. In this case, we have

$$\arg\max_{j \in [K]} A(i,j) = \arg\max_{j \in [K]} w_i v_j = \arg\max_{j \in [K]} v_j. \tag{10}$$

In other words, there exists an arm that is optimal for all contexts. The problem becomes simple as we can run the EXP4 algorithm [Auer et al., 2002] using *constant* experts that recommend the same arm for all contexts. Thus, we will have only K experts and have the following proposition:

Proposition 24. *Consider the rank-1 bandit problem with $r = 1$ and $w_i \geq 0$ for every $i \in [S]$. The regret of the EXP4 algorithm run with K constant experts is bounded by $O(\sqrt{KT \log K})$.*

However, the idea seems hard to generalize when $w_i \in \mathbb{R}$, as Eq. (10) does not hold anymore and we need exponentially many experts for EXP4. Next we introduce a new idea based on a reduction to context-lumpable bandits.

In the following we define constant $B = \max_i \|w_i\|_\infty$. To better illustrate the idea we first assume $r = 1$ and consider the PAC setting. We create an α-covering of $[-B, B]$ and "cluster" each i into one of the segment. Specifically, we have $\frac{2B}{\alpha}$ intervals

$$\left[-B, -B + \frac{1}{\alpha}\right], \ \left(-B + \frac{1}{\alpha}, -B + \frac{2}{\alpha}\right], \ldots, \left(B - \frac{1}{\alpha}, B\right]$$

and each w_i is assigned to the interval that contains it. Given α, let $\mathcal{R}_\alpha$ denote the number of intervals that have at least one context. Conceptually we can view contexts in the same interval as if they are in the same block in context-lumpable bandits, and we have $\mathcal{R}_\alpha$ blocks analogously. For contexts i, i' in the same interval, they are indeed similar in the sense that we have $|A(i,j) - A(i,j)| = O(\alpha)$ for every arm j. Intuitively, if α is much smaller than ε, then Algorithm 1 can proceed normally as a rank-$\mathcal{R}_\alpha$ context-lumpable bandit problem. Consequently, we have the following theorem.

Theorem 25. *For $r = 1$, by choosing $\alpha = \Theta(\varepsilon)$, Algorithm 1 uses $\widetilde{O}\left(\mathcal{R}_\alpha(S+K)/\varepsilon^2\right)$ samples and outputs an $\widetilde{O}(\varepsilon)$-optimal policy.*

Clearly we have $\mathcal{R}_\alpha \leq \min\left\{\frac{1}{\alpha}, S\right\}$, so the above theorem leads to a $\widetilde{O}\left((S+K)/\varepsilon^3\right)$ sample complexity in the worst case. The idea can be generalized to regret minimization and $r > 1$. Specifically, we construct an α-grid of $[-B, B]^r$ so that $\mathcal{R}_\alpha = O(\frac{1}{\alpha^r})$ and run Algorithm 6 for regret minimization. Consequently, we have the following theorem:

Theorem 26. *Let $p = \frac{1}{3r+2}$ and choose $\alpha = (S+K)^p T^{-p}$. Then regret of Algorithm 6 is bounded as $\mathrm{Reg}_T = \widetilde{O}\left((S+K)^p T^{1-p}\right)$.*

Proof. Similar to previous analysis, the regret in a single phase h is

$$\widetilde{O}(\varepsilon_h) \cdot \min\left\{T, \widetilde{O}\left(\frac{\mathcal{R}_\alpha^3(S+K)}{\varepsilon_h^2}\right)\right\} = \min\left\{\widetilde{O}(\varepsilon_h T), \widetilde{O}\left(\frac{\mathcal{R}_\alpha^3(S+K)}{\varepsilon_h}\right)\right\} \tag{11}$$

Recall that $\mathcal{R}_\alpha = O(1/\alpha^r)$. Therefore, when $\alpha = \Theta(\varepsilon_h)$, we have the above regret is bounded by

$$\widetilde{O}\left(\frac{\mathcal{R}_\alpha^3(S+K)}{\alpha} + \alpha T\right) = \widetilde{O}\left(\frac{(S+K)}{\alpha^{3r+1}} + \alpha T\right) = \widetilde{O}\left((S+K)^p T^{1-p}\right)$$

when choosing α optimally as $\alpha = (S+K)^p T^{-p}$. Otherwise, $\alpha = o(\varepsilon_h)$ and Eq. (11) can be bounded by

$$\widetilde{O}\left(\frac{\mathcal{R}_\alpha^3(S+K)}{\varepsilon_h}\right) = \widetilde{O}\left(\frac{\mathcal{R}_\alpha^3(S+K)}{\alpha}\right) = \widetilde{O}\left((S+K)^p T^{1-p}\right).$$

We finish the proof by noting that there are at most $\log T$ phases. □

The bound becomes non-trivial when S and K are large. For example, when $S = K = \sqrt{T}$, the bound is $T^{1-p/2} = o(T)$ for any r while the $\widetilde{O}\left(\sqrt{SKT}\right)$ bound given by EXP4 is vacuous. The factor 3 comes from the r^3 term in the regret bound of Theorem 5. It is a promising direction to first improve the factor in context-lumpable bandits and extend it to low-rank bandits using the reduction introduced here.