Linear Bandits with Limited Adaptivity and Learning Distributional Optimal Design*

Yufei Ruan†
University of Illinois at Urbana-Champaign
Department of ISE
Urbana, IL, USA
yufeir3@illinois.edu

Jiaqi Yang†
Tsinghua University
Institute for Interdisciplinary Information Sciences
Beijing, China
yangjq17@gmail.com

Yuan Zhou†
University of Illinois at Urbana-Champaign
Department of ISE
Urbana, IL, USA
yuanz@illinois.edu

ABSTRACT

Motivated by practical needs such as large-scale learning, we study the impact of adaptivity constraints to linear contextual bandits, a central problem in online learning and decision making. We consider two popular limited adaptivity models in literature: batch learning and rare policy switches. We show that, when the context vectors are adversarially chosen in d-dimensional linear contextual bandits, the learner needs $O(d \log d \log T)$ policy switches to achieve the minimax-optimal regret, and this is optimal up to $\mathrm{poly}(\log d, \log\log T)$ factors; for stochastic context vectors, even in the more restricted batch learning model, only $O(\log\log T)$ batches are needed to achieve the optimal regret. Together with the known results in literature, our results present a complete picture about the adaptivity constraints in linear contextual bandits. Along the way, we propose the *distributional optimal design*, a natural extension of the optimal experiment design, and provide a both statistically and computationally efficient learning algorithm for the problem, which may be of independent interest.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**;
• **Computing methodologies → Online learning settings**.

KEYWORDS

Linear bandits, Adaptivity constraints, Batch learning, Optimal design

ACM Reference Format:
Yufei Ruan, Jiaqi Yang, and Yuan Zhou. 2021. Linear Bandits with Limited Adaptivity and Learning Distributional Optimal Design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3406325.3451004

*The preprint is available at https://arxiv.org/abs/2007.01980.
†Author names are listed in alphabetical order.

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451004

1 INTRODUCTION

Online learning and decision making is a fundamental research direction in machine learning where the learner conducts sequential interactions, once per time step, with the environment in order to learn the optimal policies and maximize the total reward. To achieve optimal learning performance, the learner must seek a balance between exploration and exploitation, which is usually done by adaptively selecting actions based on all historical observations. However, full adaptivity at a per-time-step scale significantly sacrifices parallelism and hinders the large-scale deployment of learning algorithms. To facilitate scalable learning, it is worthwhile to study the following question:

> *What is the minimum amount of adaptivity needed to achieve optimal performance in online learning and decision making?*

In this paper, we address the above question through studying the impact of two popular types of adaptivity constraints to the linear contextual bandits, a central problem in online learning literature. We prove tight adaptivity-regret trade-offs for two natural settings of the problem. Along the way, we make a new connection to optimal experiment design: we propose the natural *distributional optimal design* problem, prove the existence of parametric forms for the optimal design, and present sample-efficient algorithms to learn the parameters. Our proposed framework contributes a novel learning component to the classical field of experiment design in statistics, and may be of independent interest.

Linear Contextual Bandits. The linear contextual bandits (or linear bandits for short), also known as "associative reinforcement learning" [3, 7], are a generalization of the ordinary multi-armed bandits. While also encapsulating the fundamental dilemma of "exploration vs. exploitation" in online learning and decision making, linear contextual bandits highlight the guidance of contextual information for decisions, enabling personalized treatments and recommendations in real-world applications such as clinical trial, recommendation systems, and advertisement selection.

In a bandit game, there are T time steps in total. At each time step $t \in [T]$, the learner has to make a decision among K candidate actions (a.k.a. arms in bandit literature). While in ordinary multi-armed bandits, the mean rewards of the actions have to be completely independent from each other, linear bandits allow a linear model for the mean rewards. More specifically, at time step t, each action $i \in [K]$ is associated with a d-dimensional context vector $\boldsymbol{x}_{ti}$ (a.k.a., the feature vector), and the context vectors are

presented to the learner. The expected reward for the i-th action is $\boldsymbol{\theta}^\top \boldsymbol{x}_{ti}$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is hidden from the learner. The goal is to gradually learn $\boldsymbol{\theta}$ and maximize the cumulative expected reward, or equivalently, minimize the expected *regret* (i.e., the difference between the received rewards and the rewards of the best actions in hindsight, as later defined in (1)). For example, in clinical trial, the candidate actions correspond to the K involved treatments. At time step t, an individual patient arrives with the context vectors $\{\boldsymbol{x}_{ti}\}_{i=1}^k$ characterizing his/her response to the candidate treatments, and the recovery probability given treatment i is modeled by the linear function $\boldsymbol{\theta}^\top \boldsymbol{x}_{ti}$, which corresponds to the expected reward in linear bandits.

There are two natural settings of the linear bandits: adversarial and stochastic contexts. The first setting is harder for the learner, as the context vectors are chosen by an adversary and the learner has to minimize the regret in the worst case. In the second setting, in contrast, the sets of context vectors are independently drawn from an unknown distribution $\mathcal{D}$ (while correlation may still exist among the contexts during the same time step), and the learner aims at minimizing the expected regret over $\mathcal{D}$. Note that in the clinical trial example, the individual patients can often be viewed as independent samples from the population which is characterized by $\mathcal{D}$.

Limited Adaptivity Models: Batch Learning and Rare Policy Switches. We consider two popular models of adaptivity constraints. The first model is batch learning, where the time steps are grouped into pre-defined batches. Within a batch, the same (possibly randomized) policy is used to select actions for all data and the rewards are observed only at the end of the batch. The amount of adaptivity is measured by the number of batches, which is expected to be as small as possible. A notable example is designing clinical trials, where each phase (batch) of the trial involves simultaneously applying medical treatments to a batch of patients. The outcomes are observed at the end of the phase, and may be used for designing experiments in future phases. Finding the correct number and sizes of the batches may achieve optimal efficiency for the trial by creating sufficient intra-batch parallelism while still providing sufficient adaptivity at the inter-batch scale.

The other model is learning with rare policy switches, where the amount of adaptivity is measured by the number of times allowed for the learner to change the action-selection policy. For the same amount of adaptivity measure, this model can be viewed as a relaxation of the batch learning model, because the learner in the batch learning model can only change the policy at the pre-defined time steps.

Both of the above models are closely connected to parallel learning, as we will discuss at the end of Section 1.1. We also note that another natural limited adaptivity model is "batch learning with adaptive grid" [17]. This model allows the learner to adaptively decide the size of a batch at the beginning of the batch, which is a more relaxed constraint than batch learning with pre-defined batches (a.k.a., the static grid model) but more restricted than the rare policy switch model, given the same amount of adaptivity measure.[3] Simple arguments will show that the bounds for the adaptive grid model are the same as the static grid model in both linear bandit settings. Therefore, for succinct exposition, we omit further discussions about the adaptive grid model.[4]

Optimal Experiment Design. Optimal experiment design seeks to minimize the estimation variances of parameters via intelligently choosing queries to the given set of data points. Among the multiple optimization criteria, the one most related to linear bandits is the G-optimality criterion which seeks to minimize the maximum estimation variance among the given data points. More precisely, given a set of data points $X \subseteq \mathbb{R}^d$ that spans the full dimension, the goal is to find a distribution $\mathcal{K}$ supported on X, such that $\max_{\boldsymbol{x}\in X} \boldsymbol{x}^\top (\mathbb{E}_{\boldsymbol{y}\sim\mathcal{K}} \boldsymbol{y}\boldsymbol{y}^\top)^{-1}\boldsymbol{x}$ is minimized. Here, $\mathbb{I}(\mathcal{K}) = \mathbb{E}_{\boldsymbol{y}\sim\mathcal{K}} \boldsymbol{y}\boldsymbol{y}^\top$ is the *information matrix* of the design $\mathcal{K}$, and $\boldsymbol{x}^\top \mathbb{I}(\mathcal{K})^{-1}\boldsymbol{x}$ is the variance of the estimate for data point $\boldsymbol{x}$. The General Equivalence Theorem of Kiefer and Wolfowitz [23] implies that there always exists a design $\mathcal{K}$ such that $\max_{\boldsymbol{x}\in X} \boldsymbol{x}^\top \mathbb{I}(\mathcal{K})^{-1}\boldsymbol{x} \le d$ and such designs have been used for linear bandits with fixed candidate action set (see Chapter 22 of [24], and [16]). However, to the best of our knowledge, traditional optimal design does not address the problem when the candidate action set X is stochastic. In this work, motivated by the algorithmic needs from batch linear bandits, we address this problem and develop a framework named *distributional optimal design* that runs at the core of our algorithm. We will introduce this framework in the next subsection.

1.1 Our Contributions

Adaptivity constraints in online learning and decision making have attracted much attention recently. It has been shown that multi-armed bandits only need $O(\log\log T)$ batches to achieve asymptotically minimax-optimal regret [17, 28]. For linear contextual bandits with adversarial contexts, when $\ln K \ge \Omega(d)$, Abbasi-yadkori et al. [1] showed an optimal-regret algorithm with $O(d \log T)$ policy switches. In contrast, for the batch model, Han et al. [19] recently showed that as many as $\Omega(\sqrt{T})$ batches are needed to achieve the optimal regret bound, implying that batch learning is significantly more restrictive than policy switch constraints for adversarial contexts.

In light of these partial results, quite a few questions are intriguing and remain to be explored – What makes the adaptivity requirements of linear contextual bandits fundamentally different from multi-armed bandits? What is the limitation for algorithms with rare policy switches, or in other words, can we extend the algorithm by Abbasi-yadkori et al. [1] to the full parameter range of K, and further improve the number of policy switches to $O(\log\log T)$? Do linear bandits with stochastic contexts require substantially less adaptivity than the adversarial setting? We address these questions and summarize our answers as follows.

[1] Implied by the lower bound for multi-armed bandits.

[2] Implied by the lower bound for multi-armed bandits with rare policy switches. Note that the lower bound by Simchi-Levi and Xu [31] is for deterministic action-selection policies, and becomes $\Omega(K \log\log T)$. A simple adaptation of their argument will prove the $\Omega(\log\log T)$ policy switch lower bound for randomized action-selection policies in multi-armed bandits, and imply the same lower bound for linear bandits.

[3] Indeed, in the adaptive grid model, the time for a policy switch has to be decided when the previous policy switch happens, while in the rare policy switch model, the learner can freely switch the policy, as long as the total number of switches is limited.

[4] A simple argument will prove the $\Omega(\sqrt{T})$ batch lower bound for achieving the asymptotically minimax-optimal regret for the adaptive grid model with adversarial contexts, and the rest bounds can be derived by direct corollaries of this work and the existing results in [17, 19].

Table 1: Amount of adaptivity needed in various models and settings for linear bandits.

	Batch Learning Model	Rare Policy Switch Model
Adversarial Contexts	UB: $O(\sqrt{dT})$ [19] LB: $\Omega(\sqrt{T})$ [19]	UB: $O(d \log T)$ for $\ln K \geq \Omega(d)$ [1] $O(d \log d \log T)$ for $\ln K \leq o(d)$ (by (C1)) LB: $\Omega(\frac{d \log T}{\log(d \log T)})$ (by (C1))
Stochastic Contexts	UB: $O(\log \log T)$ (by (C2)) LB: $\Omega(\log \log T)$ [17][1]	UB: $O(\log \log T)$ (implied by (C2)) LB: $\Omega(\log \log T)$ [31][2]

(C1) (Contribution #1) For linear bandits with adversarial contexts, we show that $d \log T$ (up to $\text{poly}(\log d, \log \log T)$ factors) is the tight amount of policy switches needed to achieve the minimax-optimal regret. To this end, we first extend the algorithm by Abbasi-yadkori et al. [1] to the case where $\ln K \leq o(d)$. Our algorithm achieves the asymptotically minimax-optimal regret with $O(d \log d \log T)$ policy switches. We then prove that our algorithm and the one by Abbasi-yadkori et al. [1] achieve the near-optimal policy switch vs. regret trade-off. In particular, $\Omega(d \log T / \log(d \log T))$ policy switches are needed to achieve any $\sqrt{T}$-type regret.

(C2) (Contribution #2, an informal statement of Theorem 6) For linear bandits with stochastic contexts, even in the more restricted batch learning model, it is possible to achieve the asymptotically minimax-optimal regret using only $O(\log \log T)$ batches. Our algorithm can be easily adapted to use M batches and achieve $\sqrt{d \log K} T^{\frac{1}{2(1-2^{-M})}} \cdot \text{poly} \log T$ regret, for any M.

Together with the known results in literature, we are able to present an almost complete picture about the adaptivity constraints for linear bandits in Table 1. Most interestingly, compared to ordinary multi-armed bandits, linear bandits exhibit a richer set of adaptivity requirements, and strong separations among different models and settings. We also find that adversarially chosen context vectors are the main source of difficulty for reducing adaptivity requirements.

Comparison of (C2) and [19]. Compared to (C1), our result in (C2) requires substantially more technical effort and is also the main motivation for us to develop the framework of distributional optimal design (which will be elaborated soon). We note that Han et al. [19] also studied batch learning for linear bandits with stochastic contexts and showed an algorithm with $O(\log \log T)$ batches. However, their results are for a special case of the problem with the following assumptions: the context vectors are drawn from a Gaussian distribution, the ratio between the maximum and minimum eigenvalues of the Gaussian co-variance matrix should be $O(1)$, and the number of candidate actions K cannot be greater than a polynomial of d. The design and analysis of their algorithm crucially rely on these three assumptions and it seems not obvious that their result can be directly extended to the general context set distribution. Indeed, their algorithm can safely choose the action to maximize the estimated mean reward, thanks to the isotropic Gaussian assumption ensuring sufficient exploration towards other directions. In contrast, without these assumptions, much effort in our algorithm is spent on the careful design of the exploration policy using many candidate actions, which motivates the problem of distributional optimal design.

Distributional Optimal Design. As mentioned above, to facilitate the algorithm for stochastic contexts, we have to extend the traditional experiment design results to the regime where the set X of contexts/data points is stochastic. Suppose that X follows the distribution $\mathcal{D}$, the goal of our proposed *distributional optimal design* problem is to find a sample policy π that maps any set X to a probability distribution supported on X, so as to minimize the *distributional G-variation*, defined as $\mathbb{E}_{X \sim \mathcal{D}} \max_{\boldsymbol{x} \sim X} \boldsymbol{x}^\top \mathbb{I}_{\mathcal{D}}(\pi)^{-1} \boldsymbol{x}$, where $\mathbb{I}_{\mathcal{D}}(\pi) = \mathbb{E}_{X \sim \mathcal{D}} \mathbb{E}_{\boldsymbol{y} \sim \pi(X)} \boldsymbol{y}\boldsymbol{y}^\top$ is the information matrix of sample policy π over $\mathcal{D}$.[5] Note that the traditional G-optimal design is the special case of our problem when $\mathcal{D}$ is deterministic, which was used in the algorithm for linear bandits with fixed candidate action sets (see, e.g., Chapter 22 in [24]). In contrast, the stochasticity of $X \sim \mathcal{D}$ in our problem arises due to the stochastic context in linear bandits.

The first natural question about our proposed problem is on the existence of a good sample policy. Regarding this, we prove the following result.

(C3) (Contribution #3, an informal statement of Theorem 4) For any $\mathcal{D}$, there exists a sample policy π such that the distributional G-variance is bounded by $O(d \log d)$.[6] Moreover, we can construct such a policy from the class of so-called *mixed-softmax policies*, which admits a succinct description using $O(d^3 \log d)$ real-valued parameters.

Since $\mathcal{D}$ is not known beforehand in linear bandits, we have to learn a good sample policy π via finite samples from $\mathcal{D}$. Since even the input of π lie in a continuous space with dK dimensions, proving the existence of the succinct parametric form of π in (C3) is a good news to learning. However, we find that directly constructing a policy based on the uniform distribution over empirical samples does not generalize to the true distribution $\mathcal{D}$. We will come up with a more careful learning procedure to achieve the following goal.

(C4) (Contribution #4, an informal statement of Theorem 5) For any $\mathcal{D}$, we design an algorithm to learn a good mixed-softmax policy π using only $\text{poly}(d)$ independent samples from $\mathcal{D}$.[7]

We remark that the introduction of the distribution $\mathcal{D}$ brings a unique learning challenge to optimal experiment design. It is hopeful that our results and the future study on other criteria in distributional optimal design may lead to broader applications in machine learning and statistics.

[5] For simplicity of presentation, we assume that the vectors in the sets of $\mathcal{D}$ span the full dimension, so that there always exists a sample policy with invertible information matrix. Please refer to Theorem 5.1 for the general definition.

[6] This bound can be improved to $O(d)$ with additional techniques, which will be included in the full version of the paper.

[7] More precisely, the good policy here is defined by the *distributional G-deviation*. Please refer to Theorem 5 for more details.

Implications for Collaborative and Concurrent Learning. The idea of letting multiple learning agents learn in parallel so as to save overall running time has been studied a lot recently in online learning and decision making, which is also the main motivation of this study (as mentioned in the very beginning of the paper). Below we discuss the implications of our algorithmic results for a few parallel learning models.

The first implication is for the *collaborative learning with limited interaction* model, which was recently studied for pure exploration (i.e., top arm(s) identification) in multi-armed bandits [20, 22, 37]. In this model, there are $\mathfrak{K}$ learning agents, and the learning process is partitioned into rounds of pre-defined time intervals. During each round (which is also referred to as the *communication round*), each of the $\mathfrak{K}$ agents learns individually like in the centralized model – image that there is a global buffer of the context vectors, and the agents repeatedly draw a set of context vectors from the buffer and make corresponding decisions. Each play of an arm takes one time step, and the agents may choose to skip a few time steps without playing. The agents can only communicate at the end of each round. The collective regret is defined to be the sum of the regret incurred by each agent. Suppose there are T sets of context vectors in the global buffer, the goal is to finish the game in $O(\lceil T/\mathfrak{K}\rceil)$ time (i.e., achieving the *full speedup*), while minimizing the collective regret and the number of communication rounds R.

Observe that a batch learning algorithm with M batches can be easily transformed to a collaborative algorithm with $R = M$ communication rounds, where in each round i, each agent uses the policy for the i-th batch to play for $\lfloor \mathcal{T}_i/\mathfrak{K}\rfloor$ or $\lceil \mathcal{T}_i/\mathfrak{K}\rceil$ times, where $\mathcal{T}_i$ is the size of the i-th batch. The total running time for collaborative learning is at most $T/\mathfrak{K}+M$, achieving the full speedup when $M\cdot\mathfrak{K} \le O(T)$. Therefore, when $\mathfrak{K} \le O(T/\log\log T)$, our algorithmic result (C2) implies a collaborative algorithm for stochastic-context linear bandits with full speedup and minimax-optimal collective regret, using only $O(\log\log T)$ communication rounds.

The second implication is for the *concurrent learning* model which was recently studied in [8, 18, 41]. In this model, there is no limit on the number of communication rounds and the $\mathfrak{K}$ learning agents may communicate at the end of every time step. By a simple reduction described in [8], any algorithm with at most M policy switches can be transformed to a $\mathfrak{K}$-agent concurrent learning algorithm with full speedup, and the collective regret is at most $M\cdot\mathfrak{K}$ plus the original regret bound. Therefore, our algorithmic result in (C1) implies a concurrent learning algorithm for adversarial-context linear bandits with full speedup and minimax-optimal collective regret, as long as $\mathfrak{K} \le O(\sqrt{(T\log K)/d})$.

1.2 Additional Related Works

The linear contextual bandit problem is a central question in online learning and decision making, and its regret minimization task has been studied during the past decades [1, 2, 7, 10, 12, 25, 30]. The minimax-optimal regret is proved to be $\sqrt{dT\min\{\log K, d\}}$ up to $\mathrm{poly}\log T$ factors, which is also the target regret for our algorithms with limited adaptivity. When the candidate action set is fixed, the task of identifying the best action has also been studied [34, 36, 40], and many of these works borrow the idea of G-optimal design.

Batch regret minimization for multi-armed bandits was introduced by Perchet et al. [28] with 2 arms, and the K-arm general setting was recently studied by Gao et al. [17]. Simchi-Levi and Xu [31] studied the K-arm setting with the rare policy switch constraint and achieved comparable results. For batch linear bandits, Esfandiari et al. [16] and Han et al. [19] recently studied the problem with aforementioned additional assumptions. For batch stochastic contextual bandits, Simchi-Levi and Xu [32] recently proposed an algorithm with $O(\log\log T)$ batches to achieve the minimax-optimal regret. We note that another usage of batch learning (mainly in reinforcement learning) refers to learning from a fixed set of a priori-known samples with no adaptivity allowed, which is very different from the definition in our work.

For the rare policy switch model, Abbasi-yadkori et al. [1] showed a rarely switching algorithm for linear bandits. Rare policy switch constraints have also been studied for a broader class of online learning and decision making problems, such as multinomial logit bandits [14] and Q-learning [8].

Under the broader definition of adaptivity constraints including batch learning and learning with low switching cost (which might not exactly align with the models defined in this work), many other online learning problems are studied, such as adversarial multi-armed bandits [9, 13], the best (multiple-)arm identification problem [4, 21], and convex optimization [15].

The optimal design of experiments is a fundamental problem in statistics, with various optimality criteria proposed and many statistical models studied (see, e.g., [6, 29]). When the sample budget is finite, finding the exact solutions to certain optimality criteria is NP-Hard [11, 35, 39], thus a sequence of recent works have studied approximation algorithms for the problem [5, 26, 27, 33, 38]. However, to the best of our knowledge, all previous works have considered the fixed set of all possible experiments. In contrast, we propose and study the distributional optimal design problem where the set of candidate experiments might be stochastic.

2 TECHNICAL OVERVIEW

Due to space constraints, we will only introduce the technical details related to (C2) in the rest of this extended abstract. In this section, we give an overview of the proof techniques developed for (C2) in Section 4, Section 5 and Section 6. Along the way, the proof techniques for (C3) and (C4) are also explained. In Section 7, we combine all these technical components and prove the main theorem.

The Batch Elimination Framework. All our algorithms are based on batch elimination: at each time step, the confidence intervals are estimated for each candidate action, and the actions whose confidence intervals completely fall below those of other actions are eliminated. All survived actions are likely to be the optimal one, and the learner has to design an intelligent sample policy π to select the action from the survived set. In such a way, the incurred regret can be bounded by the order of the length of the longest confidence interval in the survived set.

We note that this elimination-based approach is not new: it is adopted by the batch algorithms for multi-armed bandit (e.g., [17]) as well as the recent batch algorithm for linear bandits with fixed action set [16]. However, thanks to the simple structures of the two

problems, during each batch, both of their algorithms are able to construct confidence intervals for survived actions with a *uniform* length, so that the regret can be relatively more easily bounded. Indeed, although the algorithm by Han et al. [19] does not explicitly eliminate actions, their analysis relies on the uniform estimation confidence for the actions (which requires the isotropic Gaussian assumption for context vectors). In contrast, we have to deal with confidence intervals with wildly different lengths because of the inherent non-uniformity of the probability mass assigned to each context direction in the general distribution $\mathcal{D}$.

To deal with such non-uniformity, in Section 4, we provide an analysis framework to relate the regret bound to the distributional G-variation of π over $\mathcal{D}$, as introduced in Section 1.1. In particular, we show that if we let $\pi(X) = \pi^{\mathrm{G}}(X)$, which returns the G-optimal design of the input context set X (regardless of $\mathcal{D}$), its distributional G-variation can be bounded by d^2 (for all $\mathcal{D}$), leading to $O(d\sqrt{T \log K}) \times \operatorname{poly} \log T$ regret with $O(\log \log T)$ batches. This regret is $\sqrt{d}$ times greater than the minimax-optimal target. To achieve optimality, we need to improve the distributional G-variation to $O(d)$ (up to logarithmic factors), which requires to optimize π specifically according to $\mathcal{D}$.

Existence of Distributional Optimal Design and its Parametric Form. In Section 5, we show that, given $\mathcal{D}$, there exists a sample policy π whose distributional G-variation is $O(d \log d)$. Our proof is constructive and the algorithm involves an innovative application of the rarely switching linear bandit algorithm [1]. We consider a long enough sequence of independent samples from $\mathcal{D}$: $X_1, X_2, \ldots, X_N$, and sequentially feed the context vector sets to the rarely switching algorithm. Instead of minimizing the regret (as the reward is undefined), the rarely switching algorithm selects the context vector $\boldsymbol{x}$ that maximizes the variance according to the *delayed information matrix*, and updates the total information matrix by adding $\boldsymbol{x}\boldsymbol{x}^\top$ to it.

Borrowing the regret analysis techniques in linear bandits literature, and together with an adapted form of the celebrated Elliptical Potential Lemma, we are able to prove that, with the proper configuration of the initial information matrix, the average maximum confidence interval length throughout the N time steps is $O(d \log d)$. Moreover, the rarely switching trick makes sure that the delayed information matrix switches for at most $O(d \log d)$ times. This allows us to extract $O(d \log d)$ (deterministic) sample policies $\{\pi_j\}$ from the execution trajectory of the algorithm, each of which chooses the variance maximizer according to a delayed information matrix in the trajectory. We also associate each π_j with a probability mass p_j, which is proportional to the number of time steps when the corresponding delayed information matrix is used in the trajectory. We can then construct a so-called *mixed-argmax policy* π as follows: with probability $1/2$, π acts the same as π^{G}; otherwise, π acts the same as π_j with probability p_j.

We are then able to prove that the distributional G-variance of π over $\mathcal{D}$ is $O(d \log d)$. This is done mainly by showing that $\mathbb{I}_{\mathcal{D}}(\pi)$ is comparable to the final information matrix in the trajectory, so that the distributional G-variance of π can be bounded by the empirical average of the maximum confidence interval lengths. To lower bound $\mathbb{I}_{\mathcal{D}}(\pi)$ using the total information matrix in the trajectory, while the portion corresponding to the larger switching window (i.e., greater p_j) in the trajectory can be directly compared, the smaller switching window will be handled by the π^{G} component in π. We note that the π^{G} component is also crucial to configuring the "proper" initial information matrix in the rarely switching algorithm.

We finally observe that π can be characterized by $O(d^3 \log d)$ parameters, because each π_j is parameterized by a $d \times d$ information matrix. Since the arg max operator could be very sensitive to noise when the top input elements are close, to facilitate learning, we will also work on the *mixed-softmax* policy where each π_j uses the softmax operator instead.

CoreLearning *for Distributional Optimal Design.* It is tempting to build the natural learning algorithm that computes the distributional optimal design from the empirical samples, with the hope that the Lipschitz-continuity property of the softmax policies provides a small covering of the policy space, which leads to uniform concentration results, and finally prove that the learned policy generalizes to the true distribution $\mathcal{D}$. However, in Section 6, we construct an example to show that such an approach requires much higher sample complexity than we can afford.

To enable sample-efficient learning, we propose a new algorithm, CoreLearning, that first identifies a *core* set, which is a subset of the empirical samples, and then computes a mixed-softmax policy from the core. To identify the core, we develop a novel procedure to iteratively prune away the sets that contain less explored directions among the empirical samples, so that the set of the remaining samples at the end of the procedure becomes the core. Via a volumetric argument, we show that the directions in the core can be sufficient explored even if *only* using the sets in the core, and the core is still overwhelmingly large. Both properties are crucially used in the CoreLearning algorithm.

The high-level idea behind CoreLearning is that, on one hand, we can prove fast uniform concentration for the information matrix if all directions are sufficiently explored, so that the directions spanned by the core can be handled. On the other hand, the directions not included in the core are infrequent in $\mathcal{D}$ (because the core is large enough), and can be dealt with by the π^{G} component in the mixed-softmax policy.

Much technical effort is devoted to the analysis of CoreLearning because (1) it seems not quite obvious whether a core with the desired properties even exists, and (2) a careful analysis is needed when combining the analysis for sufficiently explored directions and infrequent directions, since the (possible) directions of the context vectors are continuous, and the boundary between the two types of directions may not be always clear. Please refer to Section 6 for more detailed explanation.

3 PRELIMINARIES

Notations. Throughout the paper, we denote $[N] \stackrel{\text{def}}{=} \{1, 2, \ldots, N\}$ for any integer N. We define $\log x \stackrel{\text{def}}{=} \log_2 x$ and $\ln x \stackrel{\text{def}}{=} \log_e x$. We use $\mathbf{1}[\cdot]$ to denote the indicator variable for a given event (i.e., the value of the variable is 1 if the event happens, and 0 otherwise). We use $\|\cdot\|$ to denote the 2-norm of matrices and vectors. Matrix and vector variables are displayed in bold letters. For any discrete

set X, we use $\triangle_X$ to denote the set of all probability distributions supported on X.

Linear Contextual Bandits. There is a hidden vector $\boldsymbol{\theta}$ ($\|\boldsymbol{\theta}\| \le 1$). For a given time horizon T, the context vectors $\{\{\boldsymbol{x}_{ti}\}_{i=1}^K\}_{t=1}^T$ are drawn from the product distribution $\mathcal{D}_1 \otimes \mathcal{D}_2 \otimes \cdots \otimes \mathcal{D}_T$, where $\mathcal{D}_t$ is the distribution for the context vectors at time step t. We assume $\|\boldsymbol{x}_{ti}\| \le 1$ for all i and t almost surely. Before the game starts, the learner only knows T.

At each time step of the game $t = 1, 2, \ldots, T$, the learner has to first decide a policy χ_t that maps any set of context vectors X to a distribution in $\triangle_X$. The learner then observes $X_t = \{\boldsymbol{x}_{ti}\}_{i=1}^K$, samples an action i_t from $\chi_t(X_t)$,[8] plays arm i_t, and finally receives the reward $r_t = \boldsymbol{\theta}^\top \boldsymbol{x}_{t,i_t} + \varepsilon_t$, where ε_t is an independent sub-Gaussian noise with variance proxy at most 1.

The goal of the learner is to minimize the expected regret

$$R^T \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^{T} \max_{i\in[K]} \boldsymbol{x}_{ti}^\top \boldsymbol{\theta} - \boldsymbol{x}_{t,i_t}^\top \boldsymbol{\theta}\right], \tag{1}$$

where the expectation is taken over $\mathcal{D}_1 \otimes \mathcal{D}_2 \otimes \cdots \otimes \mathcal{D}_T$, the noises, and the internal randomness of the learner. In our algorithmic results, we also prove $(1-\delta)$-*high probability expected regret*, which is defined as $\sup_A \mathbb{E}\left[\mathbf{1}[A] \cdot \sum_{t=1}^T \max_{i\in[K]} \boldsymbol{x}_{ti}^\top \boldsymbol{\theta} - \boldsymbol{x}_{t,i_t}^\top \boldsymbol{\theta}\right]$ where the supremum is taken over all events A such that $\Pr[A] \ge 1-\delta$. In this definition, setting $\delta = O(1/T)$ recovers the usual expected regret up to an additive error of $O(1)$.

Settings of Adversarial and Stochastic Contexts. In the setting of adversarial contexts, there are no additional constraints for the distributions $\{\mathcal{D}_t\}$. Note that this corresponds to the *oblivious adversary* in bandit literature, meaning that the adversary has to choose all context vectors beforehand. In contrast, the stronger *non-oblivious adversary* may adaptively choose context vectors for any time step according to all game history before that time. Since we only prove lower bounds for the adversarial context setting in this work, dealing with a weaker adversary actually means a stronger lower bound result.

In the setting of stochastic contexts, we have the additional assumption that $\mathcal{D} = \mathcal{D}_1 = \cdots = \mathcal{D}_T$. However, correlation may still exist among the contexts at the same time step.

4 BATCH ELIMINATION FRAMEWORK AND THE G-OPTIMAL DESIGN

As a warm-up, in this section, we first present BatchLinUCB (Algorithm 1) to illustrate the batch elimination framework for the linear bandit problem with stochastic contexts. Later in Section 4.1, we will introduce the G-optimal experiment design and show how it helps to reduce the regret bound of the algorithm. While the regret bound in Theorem 2 is improved, it still has an extra $\sqrt{d}$ factor compared to the optimal minimax regret bound (without adaptivity constraints). The quest for optimal regret will be addressed in the later sections.

We now introduce our first algorithm. BatchLinUCB (Algorithm 1) uses $M = O(\log\log T)$ batches and a pre-defined static

[8] When clear from the context, we interchangeably use the arm indices and their corresponding context vectors.

Algorithm 1: BatchLinUCB

$M = \lceil \log\log T \rceil$, $\alpha \leftarrow 10\sqrt{\ln \frac{2dKT}{\delta}}$, $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}, \mathcal{T}_0 = 0, \mathcal{T}_M = T, \forall i \in [M-1] : \mathcal{T}_i = T^{1-2^{-i}}$;
for $k \leftarrow 1, 2, \ldots, M$ **do**
  $\lambda \leftarrow 16\ln(2dT/\delta), \boldsymbol{\Lambda}_k \leftarrow \lambda \boldsymbol{I}, \boldsymbol{\xi}_k \leftarrow \boldsymbol{0}$;
  for $t \leftarrow \mathcal{T}_{k-1}+1, \mathcal{T}_{k-1}+2, \ldots, \mathcal{T}_k$ **do**
    $A_t^{(0)} \leftarrow [K], \hat{r}_{ti}^{(0)} \leftarrow 0, \omega_{ti}^{(0)} \leftarrow 1$;
    for $\kappa \leftarrow 1, 2, \ldots, k-1$ **do** ▷ Eliminate
      $\forall i \in A_t^{(\kappa-1)} : \hat{r}_{ti}^{(\kappa)} \leftarrow \boldsymbol{x}_{ti}^\top \hat{\boldsymbol{\theta}}_\kappa, \omega_{ti}^{(\kappa)} \leftarrow \alpha\sqrt{\boldsymbol{x}_{ti}^\top \boldsymbol{\Lambda}_\kappa^{-1} \boldsymbol{x}_{ti}}$;
      $A_t^{(\kappa)} \leftarrow \{i \in A_t^{(\kappa-1)} \mid \hat{r}_{ti}^{(\kappa)} + \omega_{ti}^{(\kappa)} \ge \hat{r}_{tj}^{(\kappa)} - \omega_{tj}^{(\kappa)}, \forall j \in A_t^{(\kappa-1)}\}$;
    $A_t \leftarrow A_t^{(k-1)}$;
    play arm $i_t \sim \mathrm{Unif}(A_t)$, and receive reward r_t;
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t,i_t}, \boldsymbol{\Lambda}_k \leftarrow \boldsymbol{\Lambda}_k + \boldsymbol{x}_t \boldsymbol{x}_t^\top, \boldsymbol{\xi}_k \leftarrow \boldsymbol{\xi}_k + r_t \boldsymbol{x}_t$;
  $\hat{\boldsymbol{\theta}}_k \leftarrow \boldsymbol{\Lambda}_k^{-1} \boldsymbol{\xi}_k$;

grid $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$. For each batch k, BatchLinUCB keeps an estimate $\hat{\boldsymbol{\theta}}_k$ for the hidden vector $\boldsymbol{\theta}$, which is learned using the samples obtained in the batch. To decide an arm during any time t in the k-th batch, the algorithm first performs an elimination procedure that is based on the estimate $\hat{\boldsymbol{\theta}}_\kappa$ and the corresponding confidence region for each previous batch $\kappa \in \{1, 2, \ldots, k-1\}$. Let A_t be the set of survived arms after the elimination. The algorithm then plays a uniformly random arm from A_t. The following theorem upper bounds the regret of BatchLinUCB.

Theorem 1. *With probability at least* $(1-\delta)$*, the expected regret of* BatchLinUCB *is*

$$R^T_{\text{BatchLinUCB}} \le O(\sqrt{dKT\log(dKT/\delta)} \times \log\log T).$$

To prove Theorem 1, we first introduce the following lemma that constructs the confidence intervals of the estimated rewards.

Lemma 1. *Fix any batch k, for each time step t in batch k, with probability at least $(1-\delta/T^2)$, for all $\kappa \in \{1, 2, \ldots, k-1\}$ and all $i \in A_t$, we have that*

$$\left|\boldsymbol{x}_{ti}^\top \hat{\boldsymbol{\theta}}_\kappa - \boldsymbol{x}_{ti}^\top \boldsymbol{\theta}\right| \le \omega_{ti}^{(\kappa)}.$$

The proof of Lemma 1 can be found in many papers in linear bandit literature (e.g., [10, 25]), and will be included in the full version of this paper.

We now start proving Theorem 1. Fix any batch k such that $k \ge 2$, when conditioned on the first $(k-1)$ batches, we let $\mathcal{D}_k$ be the distribution of the survived candidate arms $X = \{\boldsymbol{x}_{ti} : i \in A_t\}$ at any time t during the k-th batch. We also let $\mathcal{D}_0 = \{\boldsymbol{x}_{ti}\}$ be the distribution of all candidate arms at any time t.

Suppose that the desired event in Lemma 1 happens for every time step during the k-th batch (which happens with probability at least $(1-\delta\mathcal{T}_k/T^2)$ by a union bound), it is straightforward to verify that for each time t during the k-th batch, the optimal arm is not

eliminated by the elimination procedure (Line 6 to Line 8) in BatchLinUCB. In other words, we have that $i_t^* \stackrel{\text{def}}{=} \arg\max_{i\in[K]} \boldsymbol{x}_{ti}^\top \boldsymbol{\theta} \in A_t$ for each time step t in the k-th batch. Therefore, we can now upper bound the expected regret incurred during batch k as

$$\begin{aligned} R_k &= \mathbb{E} \sum_{t \text{ in batch } k} (\max_{i\in[K]} \boldsymbol{x}_{ti}^\top \boldsymbol{\theta} - \boldsymbol{x}_{t,i_t}^\top \boldsymbol{\theta}) \\ &\le \mathbb{E} \sum_{t \text{ in batch } k} (\boldsymbol{x}_{t,i_t^*}^\top \hat{\boldsymbol{\theta}}_{k-1} - \boldsymbol{x}_{t,i_t}^\top \hat{\boldsymbol{\theta}}_{k-1} + \omega_{t,i_t^*}^{(k-1)} + \omega_{t,i_t}^{(k-1)}) \quad (2) \\ &\le \mathbb{E} \sum_{t \text{ in batch } k} 2 \cdot (\omega_{t,i_t^*}^{(k-1)} + \omega_{t,i_t}^{(k-1)}) \\ &\le 4\,\mathbb{E} \sum_{t \text{ in batch } k} \max_{i\in A_t} \omega_{ti}^{(k-1)}, \quad (3) \end{aligned}$$

where (2) is due to the successful events of Lemma 1, the both inequalities in (3) are due to the elimination process and that $i_t^* \in A_t$. By the definition of $\omega_{ti}^{(k-1)}$ and the definition of $\mathcal{D}_k$, we further have that

$$\begin{aligned} R_k &\le 4\alpha\,\mathbb{E} \sum_{t \text{ in batch } k} \max_{i\in A_t} \sqrt{\boldsymbol{x}_{ti}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}_{ti}} \\ &\le 4\alpha \times \sum_{t \text{ in batch } k} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_k} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}}. \quad (4) \end{aligned}$$

We finally observe that $X \sim \mathcal{D}_k$ can be sampled by drawing an $X' \sim \mathcal{D}_{k-1}$ and performing an elimination process using $\hat{\boldsymbol{\theta}}_{k-1}$ as well as the corresponding confidence region for X'. We note that $X \subseteq X'$. Therefore, continuing with (4), we have that

$$\begin{aligned} R_k &\le 4\alpha \times \sum_{t \text{ in batch } k} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}} \\ &= 4\alpha\mathcal{T}_k \times \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}}. \quad (5) \end{aligned}$$

Now the goal is to upper bound $\mathbb{E}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}}$. The following lemma can be proved by matrix concentration inequalities, and the proof is deferred to the full version.

Lemma 2. *For each batch k ($k < M$), with probability $(1 - \delta/T^2)$, we have that*

$$\boldsymbol{\Lambda}_k \succcurlyeq \frac{\mathcal{T}_k}{16} \left(\frac{\ln T}{\mathcal{T}_k} \boldsymbol{I} + \mathop{\mathbb{E}}_{X\sim\mathcal{D}_k} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathrm{Unif}(X)} [\boldsymbol{x}\boldsymbol{x}^\top] \right). \quad (6)$$

Assuming that (6) holds for batch $(k-1)$, we have that

$$\begin{aligned} &\mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}} \le \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \sum_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}} \\ &\le \frac{4}{\sqrt{\mathcal{T}_{k-1}}} \sqrt{\mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \sum_{\boldsymbol{x}\in X} \boldsymbol{x}^\top \left(\frac{\ln T}{\mathcal{T}_{k-1}} \boldsymbol{I} + \mathop{\mathbb{E}}_{Y\sim\mathcal{D}_{k-1}} \frac{1}{|Y|} \sum_{\boldsymbol{y}\in Y} \boldsymbol{y}\boldsymbol{y}^\top \right)^{-1} \boldsymbol{x}} \\ &\le \frac{4}{\sqrt{\mathcal{T}_{k-1}}} \sqrt{\mathrm{Tr}\left(\left(\frac{\ln T}{\mathcal{T}_{k-1}} \boldsymbol{I} + \mathop{\mathbb{E}}_{Y\sim\mathcal{D}_{k-1}} \frac{1}{K} \sum_{\boldsymbol{y}\in Y} \boldsymbol{y}\boldsymbol{y}^\top \right)^{-1} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \sum_{\boldsymbol{x}\in X} \boldsymbol{x}\boldsymbol{x}^\top \right)} \\ &\le 4\sqrt{dK/\mathcal{T}_{k-1}}. \end{aligned}$$

Together with (5), and collecting the probabilities, we have that with probability at least $(1 - \delta\mathcal{T}_k/T^2 - \delta/T^2)$, the expected regret incurred during batch k ($k \ge 2$) is

$$R_k \le 16\alpha\mathcal{T}_k \cdot \sqrt{dK/\mathcal{T}_{k-1}} \le 16\alpha\sqrt{dKT}. \quad (7)$$

Note that (7) also holds for $k = 1$ almost surely, because $\mathcal{T}_1 \le \sqrt{dT}$ and the maximum regret incurred per time step is at most 1.

Finally, summing up the expected regret incurred across all batches and collecting the probabilities, we have that, with probability at least $(1 - \delta)$, the expected regret is bounded by

$$R^T \le M \times 16\alpha\sqrt{dKT} = O(\sqrt{dKT \log(dKT/\delta)} \times \log\log T).$$

This concludes the proof of Theorem 1.

4.1 Improved Regret via the G-Optimal Design

In this subsection, we show how a simple application of the G-optimal design can help to replace the K factor in Theorem 1 by (the usually smaller quantity) d. To achieve this, we first introduce the following lemma on G-optimal design, which is a direct corollary of the General Equivalence Theorem of Kiefer and Wolfowitz [23].

Lemma 3. *For any subset $X \subseteq \mathbb{R}^d$, there exists a distribution $\mathcal{K}_X$ supported on X, such that for any $\varepsilon > 0$, it holds that*

$$\max_{\boldsymbol{x}\in X} \boldsymbol{x}^\top \left(\varepsilon \boldsymbol{I} + \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{K}_X} \boldsymbol{y}\boldsymbol{y}^\top \right)^{-1} \boldsymbol{x} \le d. \quad (8)$$

Furthermore, if X is a discrete set with finite cardinality, one can find a distribution such that the right-hand side of (8) is relaxed to $2d$ in time $\mathrm{poly}(|X|)$.

We now describe the new BatchLinUCB-KW algorithm. It is almost the same as BatchLinUCB, while the only difference is that at Line 10 of Algorithm 1, letting $X = \{\boldsymbol{x}_{ti} : i \in A_t\}$, we compute a distribution $\mathcal{K}_X$ satisfying (8) (up to the factor 2 relaxation) and randomly select the action

$$i_t \sim \pi^{\mathrm{G}}(X) \stackrel{\text{def}}{=} \mathcal{K}_X. \quad (9)$$

Theorem 2. *With probability at least $(1 - \delta)$, the expected regret of* BatchLinUCB-KW *is*

$$R^T_{\text{BatchLinUCB-KW}} \le O(d\sqrt{T \log(dKT/\delta)} \times \log\log T).$$

We now prove Theorem 2. Note that the analysis for BatchLinUCB also applies to BatchLinUCB-KW up to (5). Thus, we will focus on bounding $\mathbb{E}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}}$ while keeping in mind that $\boldsymbol{\Lambda}_{k-1}^{-1}$ is a different quantity due to π^{G}.

Similarly to Lemma 2, for each batch k ($k < M$), with probability $(1 - \delta/T^2)$, we have that

$$\boldsymbol{\Lambda}_k \succcurlyeq \frac{\mathcal{T}_k}{16} \left(\frac{\ln T}{\mathcal{T}_k} \boldsymbol{I} + \mathop{\mathbb{E}}_{X\sim\mathcal{D}_k} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\pi^{\mathrm{G}}(X)} [\boldsymbol{x}\boldsymbol{x}^\top] \right). \quad (10)$$

Assuming that (10) holds for batch $(k-1)$, letting

$$\boldsymbol{x}^*(X) = \arg\max_{\boldsymbol{x}\in X} \boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x},$$

we have that

$$\begin{aligned} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}} &= \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \sqrt{(\boldsymbol{x}^*(X))^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}^*(X)} \\ &\le \sqrt{\mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} (\boldsymbol{x}^*(X))^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}^*(X)} \end{aligned}$$

$$= \sqrt{\mathrm{Tr}\left(\Lambda_{k-1}^{-1} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \boldsymbol{x}^*(X)(\boldsymbol{x}^*(X))^\top\right)}, \tag{11}$$

where the inequality is by Jensen's inequality. By (8) (up to the factor 2 relaxation), we have that

$$\boldsymbol{x}^*(X)(\boldsymbol{x}^*(X))^\top \preccurlyeq 2d \times \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi^{\mathrm{G}}(X)} \boldsymbol{y}\boldsymbol{y}^\top. \tag{12}$$

Combining (11) and (12), we have that

$$\mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \Lambda_{k-1}^{-1} \boldsymbol{x}} \le \sqrt{2d \times \mathrm{Tr}\left(\Lambda_{k-1}^{-1} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\pi^{\mathrm{G}}(X)} \boldsymbol{y}\boldsymbol{y}^\top\right)}$$
$$\le 4\sqrt{2}d/\sqrt{\mathcal{T}_{k-1}}, \tag{13}$$

where the last inequality is due to (10). Combining (13) and (5), we have that with probability at least $(1-\delta\mathcal{T}_k/T^2-\delta/T^2)$, the expected regret incurred during batch k ($k \ge 2$) is

$$R_k \le 4\alpha\mathcal{T}_k \cdot 4\sqrt{2}d/\sqrt{\mathcal{T}_{k-1}} \le 16\sqrt{2}\alpha d\sqrt{T}.$$

Using the similar argument as the analysis for Algorithm 1, we have that with probability at least $(1-\delta)$, the expected regret of BatchLinUCB-KW is at most

$$R^T \le O(d\sqrt{T\log(dKT/\delta)} \times \log\log T),$$

proving Theorem 2.

5 DISTRIBUTIONAL G-OPTIMAL DESIGN: EXISTENCE & PARAMETRIC FORMS

We now work towards removing the extra $\sqrt{d}$ factor in the regret of Theorem 2, so as to achieve the optimal $\sqrt{dT}$-type regret. The high level idea is to use a difference sample policy other than uniform sampling over all (survived) candidate arms or the G-optimal-design-based π^{G}.

Given a sample policy π that maps any set of arms ($X \subseteq \mathbb{R}^d$) to a distribution in $\triangle_X$, we will be interested in its performance, defined as follows.

Definition 5.1 (λ-distributional G-variation and information matrix). For any distribution $\mathcal{D}$ of the set of arms $X \subseteq \mathbb{R}^d$ and any sample policy π, we define the *λ-distributional G-variation*, or *λ-variation* for short ($\lambda > 0$), of π over $\mathcal{D}$ as

$$\mathbb{V}_{\mathcal{D}}^{(\lambda)}(\pi) \stackrel{\mathrm{def}}{=} \mathop{\mathbb{E}}_{X\sim\mathcal{D}} \max_{\boldsymbol{x}\in X} \boldsymbol{x}^\top \left(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi)\right)^{-1} \boldsymbol{x},$$

where we define the *information matrix* by

$$\mathbb{I}_{\mathcal{D}}(\pi) \stackrel{\mathrm{def}}{=} \mathop{\mathbb{E}}_{X\sim\mathcal{D}} \mathbb{I}_X(\pi), \qquad \text{where } \mathbb{I}_X(\pi) \stackrel{\mathrm{def}}{=} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\pi(X)} \boldsymbol{x}\boldsymbol{x}^\top.$$

Since $\mathbb{V}_{\mathcal{D}}^{(\lambda)}$ is non-increasing as λ grows, when the limit exists, we also define

$$\mathbb{V}_{\mathcal{D}}^{(0)}(\pi) \stackrel{\mathrm{def}}{=} \lim_{\lambda\to 0^+} \mathbb{V}_{\mathcal{D}}^{(\lambda)}(\pi), \tag{14}$$

and set $\mathbb{V}_{\mathcal{D}}^{(0)}(\pi) = +\infty$ otherwise.

Indeed, the arguments in Section 4 imply the following lemma.

Lemma 4. *For any distribution $\mathcal{D}$ on the context vectors of the K arms, we have that*

$$\mathbb{V}_{\mathcal{D}}^{(0)}(\mathrm{Unif}) \le O(dK), \text{ and } \mathbb{V}_{\mathcal{D}}^{(0)}(\pi^{\mathrm{G}}) \le O(d^2). \tag{15}$$

Algorithm 2: Algorithm for Computing a Distributional G-Optimal Design

Input: A context set sequence $X_1, \ldots, X_\Gamma$
Output: A mixed-argmax policy π

$N \leftarrow 2d^2 \log d, \forall (i,j) \in [N]\times[\Gamma] : X_{(i-1)\Gamma+j} \leftarrow X_j$;
$\boldsymbol{U}_0 \leftarrow \lambda N\Gamma \boldsymbol{I} + \frac{N}{2}\sum_{i=1}^{\Gamma} \mathbb{E}_{\boldsymbol{x}\sim\pi^{\mathrm{G}}(X_i)}[\boldsymbol{x}\boldsymbol{x}^\top] \succcurlyeq \boldsymbol{I}, n \leftarrow 1, \tau_n \leftarrow \emptyset, \boldsymbol{W}_n = \boldsymbol{U}_0$;
for $t \leftarrow 1, 2, \ldots, N\Gamma$ **do**
  $\tau_n \leftarrow \tau_n \cup \{t\}$;
  $\boldsymbol{x}_t \leftarrow \pi^{\mathrm{A}}_{\boldsymbol{W}_n^{-1}}(X_t) = \arg\max_{\boldsymbol{x}\in X_t} \boldsymbol{x}^\top \boldsymbol{W}_n^{-1} \boldsymbol{x}$; ▷ Ties are broken in a deterministic manner.
  $\boldsymbol{U}_t \leftarrow \boldsymbol{U}_{t-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top$;
  if $\det \boldsymbol{U}_t > 2 \det \boldsymbol{W}_n$ **then**
    $n \leftarrow n+1, \tau_n \leftarrow \emptyset, \boldsymbol{W}_n \leftarrow \boldsymbol{U}_t$;
for all $i \in [n]$, **if** $|\tau_i| < \Gamma$ **then** $\tau_i \leftarrow \emptyset$;
for all $i \in [n]$, set $p_i = |\tau_i| / \sum_j |\tau_j|$;
return $\{(p_i, N\Gamma \boldsymbol{W}_i^{-1}) : i \in [n] \text{ and } p_i > 0\}$

In light of Lemma 4, the question whether the regret of our algorithms can be improved to $O(\sqrt{dT\mathrm{poly}\log(KT/\delta)})$ boils down to whether one can find a sample policy π such that the bounds in (15) are improved to $O(d) \times \mathrm{poly}\log d$. In this section, we will show that such policies not only exist, but also admit a succinct parametric form so that we can later study how to efficiently learn the relevant parameters.

To better explain our results, we first define the following class of parameterized sample policies.

Definition 5.2 (Argmax and mixed-argmax policies). Suppose we are given a positive semi-definite matrix $\boldsymbol{V} \succcurlyeq \boldsymbol{0}$. We define the associated *argmax policy* by

$$\pi^{\mathrm{A}}_{\boldsymbol{V}}(X) = \arg\max_{\boldsymbol{x}\in X} \boldsymbol{x}^\top \boldsymbol{V} \boldsymbol{x},$$

where in the arg max operator, ties are broken in a deterministic manner.

In this subsection, we use π^{G} to denote a *fixed* policy with respect to (9) and satisfying (8) (up to the factor 2 relaxation). Suppose we are given a set $\mathcal{V} = \{(p_i, \boldsymbol{V}_i)\}_{i=1}^n$ such that $p_i \ge 0$ and $p_1 + \cdots + p_n = 1$. We define the associated *mixed-argmax policy* by

$$\pi^{\mathrm{MA}}_{\mathcal{V}}(X) = \begin{cases} \pi^{\mathrm{G}}(X), & \text{with probability } 1/2, \\ \pi^{\mathrm{A}}_{\boldsymbol{V}_i}(X), & \text{with probability } p_i/2. \end{cases}$$

The following theorem states that for any $\mathcal{D}$, there exists a good mixed-argmax policy with only $O(d\log d)$ argmax policies in the mixture.[9]

Theorem 3. *Let $S = \{X_1, X_2, \ldots, X_\Gamma\}$ be a (multi-)set and let $\mathcal{D} = \mathrm{Unif}(S)$. For any $\lambda \in (0,1)$, there exists a mixed-argmax policy with parameters $\mathcal{V} = \{(p_i, \boldsymbol{V}_i)\}_{i=1}^n$ such that*

(1) $n \le 4d\log d$;
(2) for all $i \in [n]$, $p_i \ge 1/d^3$ and $d^{-1}\boldsymbol{I} \preccurlyeq \boldsymbol{V}_i \preccurlyeq \lambda^{-1}\boldsymbol{I}$;

[9] Note that although the theorem only works for the uniform distribution over a multi-set, since the properties to be proved in the theorem statement do not truly depend on Γ, the theorem can be generalized to any distribution via a simple discretization argument.

(3) $\mathbb{V}_{\mathcal{D}}^{(\lambda)}(\pi_{\mathcal{V}}^{\mathrm{MA}}) \le O(d \log d)$.

Proof. We will assume $\Gamma > \lambda^{-1}$ without loss of generality, as the properties to be proved do not depend of Γ and S is a multi-set so that we can always duplicate the elements by finitely many times.

We prove the theorem constructively. We consider Algorithm 2, which is very similar to the linear bandits algorithms in literature. For $N = \Theta(d^2 \log d)$, the algorithm creates ΓN times steps, which includes N blocks, each of which contains Γ consecutive time steps. In each block, the Γ sets of arms $X_1, \dots, X_\Gamma$ are sequentially presented. The algorithm then simulates the linear bandit algorithms, where at each time step, the arm with the maximum variance (according to the information matrix $\boldsymbol{W}_n$) is selected. Inspired by the rarely switching algorithm for linear bandits [1], the information matrix $\boldsymbol{W}_n$ is only updated when its determinant doubles. This significantly reduces the number of updates and is crucial to upper bounding the number of individual argmax policies in the returned mixed-argmax policy. We refer to the consecutive time steps between two neighboring updates as a *stage*. Each of the information matrices in a stage corresponds to an individual argmax policy in the returned policy, and the corresponding probability weight is proportional to the length of the stage. The only exception is that we discard the stages that contain less than Γ time steps (i.e., the ones that are shorter than a block).

Proof of Item (a). Note that

$$\boldsymbol{U}_{N\Gamma} = \boldsymbol{U}_0 + \sum_{t=1}^{N\Gamma} \boldsymbol{x}_t \boldsymbol{x}_t^\top = \lambda N\Gamma \boldsymbol{I} + \frac{N\Gamma}{2} \mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) + \sum_{t=1}^{N\Gamma} \boldsymbol{x}_t \boldsymbol{x}_t^\top. \quad (16)$$

By (8) (up to the factor 2 relaxation), for all t, we can show that

$$\boldsymbol{x}_t \boldsymbol{x}_t^\top \preccurlyeq 2d \times \mathop{\mathbb{E}}_{\boldsymbol{y} \in \pi^{\mathrm{G}}(X_t)} \boldsymbol{y}\boldsymbol{y}^\top. \quad (17)$$

Combining (16) and (17), we have that $\boldsymbol{U}_{N\Gamma} \preccurlyeq \lambda N\Gamma \boldsymbol{I} + (1/2 + 2d) N\Gamma \times \mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) \preccurlyeq 4d\boldsymbol{U}_0$. Therefore, we have

$$\det \boldsymbol{U}_{N\Gamma} \le \det(4d\boldsymbol{U}_0) = d^{4d} \det \boldsymbol{U}_0, \quad (18)$$

and $n \le \log\left(d^{4d}\right) = 4d \log d$.

Proof of Item (b). Because we discard the stages whose lengths are less than Γ, for $p_i > 0$, we have that $p_i \ge \frac{\Gamma}{N\Gamma} \ge \frac{1}{d^3}$ for large enough d.

For each $\boldsymbol{W}_i$, we have $\boldsymbol{W}_i \succcurlyeq \boldsymbol{U}_0 \succcurlyeq \lambda N\Gamma \boldsymbol{I}$, and $\boldsymbol{W}_i \preccurlyeq 3N\Gamma \boldsymbol{I}$. Since $\boldsymbol{V}_i = N\Gamma \boldsymbol{W}_i^{-1}$, we have that $d^{-1}\boldsymbol{I} \preccurlyeq \boldsymbol{V}_i \preccurlyeq \lambda^{-1}\boldsymbol{I}$.

Proof of Item (c). We finally upper bound the λ-variation of the returned policy $\pi = \pi_{\mathcal{V}}^{\mathrm{MA}}$. Note that

$$\begin{aligned}
\mathbb{V}_{\mathcal{D}}^{(\lambda)}(\pi) &= \mathop{\mathbb{E}}_{X \sim \mathcal{D}} [\max_{\boldsymbol{x} \in X} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathop{\mathbb{E}}_{X \sim \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \pi(X)} \boldsymbol{x}\boldsymbol{x}^\top)^{-1} \boldsymbol{x}] \\
&= \sum_{t=1}^{N\Gamma} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (N\Gamma(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi)))^{-1} \boldsymbol{x} \\
&= \sum_{i=1}^{n} \sum_{t \in \tau_i} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (N\Gamma(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi)))^{-1} \boldsymbol{x} \\
&\quad + \sum_{t \in \mathcal{B}} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (N\Gamma(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi)))^{-1} \boldsymbol{x}, \quad (19)
\end{aligned}$$

where we let $\mathcal{B}$ be the set of time steps that are discarded in Line 9 of Algorithm 2.

It remains to show that both terms are $O(d \log d)$. For the second term, we have

$$\begin{aligned}
&\sum_{t \in \mathcal{B}} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (N\Gamma(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi)))^{-1} \boldsymbol{x} \\
&= \frac{1}{N\Gamma} \sum_{t \in \mathcal{B}} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi))^{-1} \boldsymbol{x} \\
&\le \frac{2}{N\Gamma} \sum_{t \in \mathcal{B}} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{x}, \quad (20)
\end{aligned}$$

where the inequality is because by definition of a mixed-argmax policy, with probability $1/2$, π^{G} is invoked, and therefore

$$\mathbb{I}_{\mathcal{D}}(\pi) = \mathop{\mathbb{E}}_{X \sim \mathcal{D}, \boldsymbol{x} \sim \pi(X)} \boldsymbol{x}\boldsymbol{x}^\top \succcurlyeq \mathop{\mathbb{E}}_{X \sim \mathcal{D}} \frac{1}{2} \times \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \pi^{\mathrm{G}}(X)} \boldsymbol{x}\boldsymbol{x}^\top.$$

Continuing with (20), since $\mathcal{B}$ contains at most n stages that are shorter than a block, therefore, we have that

$$\begin{aligned}
&\frac{2}{N\Gamma} \sum_{t \in \mathcal{B}} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{x} \\
&\le \frac{2}{N\Gamma} \times n \times \sum_{t=1}^{\Gamma} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{x} \\
&= \frac{2n}{N} \mathop{\mathbb{E}}_{X \sim \mathcal{D}} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{x} \\
&= \frac{2n}{N} \mathbb{V}_{\mathcal{D}}^{(\lambda)}(\pi^{\mathrm{G}}) \le \frac{2n}{N} \times O(d^2) \le O(d \log d), \quad (21)
\end{aligned}$$

where the second inequality is due to (14), (15), and the monotonicity of $\mathbb{V}_{\mathcal{D}}^{(\lambda)}$.

For the first term in (19), we claim that

$$\mathbb{I}_{\mathcal{D}}(\pi) \succcurlyeq \frac{1}{4N\Gamma} \sum_{t=1}^{N\Gamma} \boldsymbol{x}_t \boldsymbol{x}_t^\top, \quad (22)$$

which will be established at the end of this proof. Once we have (22), also noting that $\mathbb{I}_{\mathcal{D}}(\pi) \succcurlyeq (1/2)\mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}})$ because of the $1/2$ portion of π^{G} in the definition of the mixed-argmax policy, we get that

$$\begin{aligned}
\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi) &\succcurlyeq \lambda \boldsymbol{I} + \frac{1}{2}\left(\frac{1}{2}\mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) + \frac{1}{4N\Gamma} \sum_{t=1}^{N\Gamma} \boldsymbol{x}_t \boldsymbol{x}_t^\top\right) \\
&\succcurlyeq \frac{1}{8N\Gamma} \boldsymbol{U}_{N\Gamma} \succcurlyeq \frac{1}{8N\Gamma} \boldsymbol{W}_n. \quad (23)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{t \in \tau_i} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top (N\Gamma(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi)))^{-1} \boldsymbol{x} &\le 8 \sum_{i=1}^{n} \sum_{t \in \tau_i} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top \boldsymbol{W}_n^{-1} \boldsymbol{x} \\
&\le 8 \sum_{i=1}^{n} \sum_{t \in \tau_i} \max_{\boldsymbol{x} \in X_t} \boldsymbol{x}^\top \boldsymbol{W}_i^{-1} \boldsymbol{x} = 8 \sum_{i=1}^{n} \sum_{t \in \tau_i} \boldsymbol{x}_t^\top \boldsymbol{W}_i^{-1} \boldsymbol{x}_t \\
&\le 16 \sum_{i=1}^{n} \sum_{t \in \tau_i} \boldsymbol{x}_t^\top \boldsymbol{U}_t^{-1} \boldsymbol{x}_t \le 16 \sum_{t=1}^{N\Gamma} \boldsymbol{x}_t^\top \boldsymbol{U}_t^{-1} \boldsymbol{x}_t \quad (24) \\
&\le 32 \ln \frac{\det \boldsymbol{U}_{N\Gamma}}{\det \boldsymbol{U}_0} \le O(d \log d). \quad (25)
\end{aligned}$$

where the first inequality in (24) is by Lemma 12 in [1], the first inequality in (25) is by the celebrated elliptical potential lemma (Lemma 5, stated at the end of this subsection)[10], and the second inequality in (25) is due to (18).

It remains to establish (22). Note that

$$\begin{aligned}\mathbb{I}_{\mathcal{D}}(\pi) &= \frac{1}{2}\mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) + \frac{1}{2}\sum_{i=1}^{n}\frac{|\tau_i|}{|\tau_1|+\cdots+|\tau_n|}\mathop{\mathbb{E}}_{X\sim\mathcal{D}}\mathbb{I}_X(\pi^{\mathsf{A}}_{W_i^{-1}})\\ &\geqslant \frac{1}{2}\mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) + \frac{1}{2}\sum_{i=1}^{n}\frac{|\tau_i|}{|\tau_1|+\cdots+|\tau_n|}\frac{1}{2|\tau_i|}\sum_{t\in\tau_i}\boldsymbol{x}_t\boldsymbol{x}_t^\top\\ &= \frac{1}{2}\mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) + \frac{1}{4}\frac{1}{N\Gamma-|\mathcal{B}|}\sum_{i=1}^{n}\sum_{t\in\tau_i}\boldsymbol{x}_t\boldsymbol{x}_t^\top. \end{aligned} \tag{26}$$

By (17), we have

$$\mathbb{I}_{\mathcal{D}}(\pi^{\mathrm{G}}) = \frac{1}{n\Gamma}\sum_{t=1}^{\Gamma} n\times\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\pi^{\mathrm{G}}(X_t)}\boldsymbol{x}\boldsymbol{x}^\top \geqslant \frac{1}{n\Gamma}\sum_{t\in\mathcal{B}}\frac{1}{2d}\boldsymbol{x}_t\boldsymbol{x}_t^\top.$$

Therefore, continuing with (26), we have that

$$\begin{aligned}\mathbb{I}_{\mathcal{D}}(\pi) &\geqslant \frac{1}{2nd\Gamma}\sum_{t\in\mathcal{B}}\boldsymbol{x}_t\boldsymbol{x}_t^\top + \frac{1}{4N\Gamma}\sum_{i=1}^{n}\sum_{t\in\tau_i}\boldsymbol{x}_t\boldsymbol{x}_t^\top\\ &\geqslant \frac{1}{4N\Gamma}\sum_{t\in\mathcal{B}}\boldsymbol{x}_t\boldsymbol{x}_t^\top + \frac{1}{4N\Gamma}\sum_{i=1}^{n}\sum_{t\in\tau_i}\boldsymbol{x}_t\boldsymbol{x}_t^\top = \frac{1}{4N\Gamma}\sum_{t=1}^{N\Gamma}\boldsymbol{x}_t\boldsymbol{x}_t^\top, \end{aligned} \tag{27}$$

which concludes the proof of the theorem. □

Now we state a generalized version of the elliptical potential lemma. Compared to the usual version in literature (e.g., [1]), our versions works for positive semi-definite matrices $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n$ with traces upper bounded by 1 instead of just rank-1 positive semi-definite matrices. However, we also need the extra assumption that $\mathrm{Tr}\left(\boldsymbol{X}_i\boldsymbol{V}_0^{-1}\right)\leq 1$ for all $i\in[n]$. The proof of Lemma 5 is deferred to the full version.

Lemma 5 (Generalized Elliptical Potential Lemma). *Suppose we are given a sequence of positive semi-definite matrices $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n$ such that $\mathrm{Tr}(\boldsymbol{X}_i)\leq 1$ for every $i\in[n]$. Let $\boldsymbol{\Lambda}_0$ be a positive semi-definite matrix and let $\boldsymbol{\Lambda}_i=\boldsymbol{\Lambda}_{i-1}+\boldsymbol{X}_i$ for $i\in[n]$. When $\mathrm{Tr}\left(\boldsymbol{X}_i\boldsymbol{\Lambda}_0^{-1}\right)\leq 1$ for $i\in[n]$, we have $\sum_{i=1}^{n}\mathrm{Tr}\left(\boldsymbol{X}_i\boldsymbol{\Lambda}_{i-1}^{-1}\right)\leq 2\ln\frac{\det\boldsymbol{\Lambda}_n}{\det\boldsymbol{\Lambda}_0}$.*

5.1 The Mixed-Softmax Policies with More Robustness

To make the sample policy learnable, instead of the mixed-argmax policies, we will deal with the more robust mixed-softmax policies. To define this class of policies, we first define the softmax function as a distribution such that

$$\mathrm{softmax}_\alpha(s_1,\ldots,s_k)=i \quad \text{with probability} \quad \frac{s_i^\alpha}{s_1^\alpha+\cdots+s_k^\alpha},$$

where we assume that $s_i\geq 0$ for all $i\in[k]$.

It is easy to check the following fact.

[10] We invoke the lemma by letting $\boldsymbol{X}_t$ in the lemma statement be $\boldsymbol{x}_t\boldsymbol{x}_t^\top$ and letting $\boldsymbol{\Lambda}_t$ in the lemma statement be $\boldsymbol{U}_t$. Note that $\boldsymbol{\Lambda}_0=\boldsymbol{U}_0\succcurlyeq\boldsymbol{I}$ so that $\mathrm{Tr}(\boldsymbol{X}_t\boldsymbol{\Lambda}_0^{-1})\leq 1$ is satisfied.

Fact 1. *Suppose $\alpha\geq\log k$, then*

$$\mathop{\mathbb{E}}_{i\sim\mathrm{softmax}_\alpha(s_1,\ldots,s_k)}[s_i]\geq\frac{1}{4}\times\max\{s_1,\ldots,s_k\}.$$

Proof. Let i^* be an index that maximizes s_i. Note that for all j such that $s_j\leq(1/2)\times s_{i^*}$, the probability mass that softmax put for j is at most $(1/k)$ of that for i^*. Therefore,

$$\Pr_{i\sim\mathrm{softmax}_\alpha(s_1,\ldots,s_k)}[s_i\geq\frac{1}{2}\times s_{i^*}]\geq\frac{1}{2},$$

and the fact follows. □

We now define the class of mixed-softmax policies.

Definition 5.3 (Softmax and mixed-softmax policies). Fix $\alpha=\log K$ (where K is the number of arms per time step). Suppose we are given a positive semi-definite matrix $\boldsymbol{M}\succcurlyeq\boldsymbol{0}$. We define the *softmax policy*

$$\pi^{\mathsf{S}}_{\boldsymbol{M}}(X)=\boldsymbol{x}_i,\quad \text{where}\quad X=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k\},k\leq K,\quad \text{and}\quad i\sim\mathrm{softmax}_\alpha(\boldsymbol{x}_1^\top\boldsymbol{M}\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k^\top\boldsymbol{M}\boldsymbol{x}_k).$$

Suppose we are given a set $\mathcal{M}=\{(p_i,\boldsymbol{M}_i)\}_{i=1}^n$ such that $p_i\geq 0$ and $p_1+\cdots+p_n=1$. We define the *mixed-softmax policy*

$$\pi^{\mathsf{MS}}_{\mathcal{M}}(X)=\begin{cases}\pi^{\mathrm{G}}(X), & \text{with probability } 1/2,\\ \pi^{\mathsf{S}}_{\boldsymbol{M}_i}(X), & \text{with probability } p_i/2.\end{cases}$$

Similarly to Theorem 3, we prove the following theorem on the existence of good mixed-softmax policies.

Theorem 4. *Let $S=\{X_1,X_2,\ldots,X_\Gamma\}$ be a (multi-)set and let $\mathcal{D}=\mathrm{Unif}(S)$. For any $\lambda\in(0,1)$, there exists a mixed-softmax policy $\pi^{\mathsf{MS}}_{\mathcal{M}}$ with parameters $\mathcal{M}=\{(p_i,\boldsymbol{M}_i)\}_{i=1}^n$ such that*

(1) $n\leq 4d\log d$;
(2) for all $i\in[n]$, $p_i\geq 1/d^3$ and $d^{-1}\boldsymbol{I}\preccurlyeq\boldsymbol{M}_i\preccurlyeq\lambda^{-1}\boldsymbol{I}$;
(3) $\mathbb{V}^{(\lambda)}_{\mathcal{D}}(\pi^{\mathsf{MS}}_{\mathcal{M}})\leq O(d\log d)$.

The proof of Theorem 4 is very similar to that of Theorem 3, and is deferred to the full version.

6 LEARNING THE DISTRIBUTIONAL G-OPTIMAL DESIGN

In this section, we present an algorithm to learn a good mixed-softmax policy using only $\mathrm{poly}(d)\log\delta^{-1}$ samples with success probability at least $(1-\delta)$.

The Natural Idea and its Counterexample. The most natural idea is to first draw γ independent samples $X_1,\ldots,X_\gamma\sim\mathcal{D}$ and form an empirical distribution $\mathcal{S}=\mathrm{Unif}\{X_1,\ldots,X_\gamma\}$, learn a good policy π for $\mathcal{S}$ according to Theorem 4, and hope that π also works well for $\mathcal{D}$ (i.e., π generalizes to the true distribution). Unfortunately, such an approach is unlikely to work. Below we illustrate an example where, even when the number of samples γ is very large, a good policy for $\mathcal{S}$ still fails to generalize to $\mathcal{D}$ with significant probability.

Let $\{\boldsymbol{e}_i\}_{i=1}^d$ be the set of canonical basis, and $\varepsilon>0$ be a parameter to be determined later. Let $Y_1=\{\boldsymbol{e}_1\}$ and $Y_i=\{\sqrt{1-\varepsilon^2}\boldsymbol{e}_i+\varepsilon\boldsymbol{e}_1,\boldsymbol{e}_i\}$ for $i\in\{2,3,\ldots,d\}$. Consider $\mathcal{D}$ supported on $\{Y_1,\ldots,Y_d\}$ the probability mass for Y_1 is $1/(d\gamma)$ and the probability for Y_i $(i\geq 2)$ is $q=(1-1/(d\gamma))/(d-1)$. If we make γ independent samples

Algorithm 3: CoreLearning for the Distributional G-Optimal Design

Input: $\lambda \in (\exp(-d), 1)$, and $S = \{X_1, \ldots, X_\gamma\}$
Output: A mixed-softmax policy π

1: Set constant $c = 6$;
2: Find a core $C \subseteq S = \{X_1, \ldots, X_\gamma\}$ (using CoreIdentification (Algorithm 4), see Lemma 6) such that

$$\max_{X \in C} \max_{\boldsymbol{x} \in X} \{\boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathrm{Unif}(C)}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{x}\} \le d^c, \tag{28}$$

$$\text{and} \qquad \frac{|C|}{\gamma} \ge 1 - O(d^{3-c} \log \lambda^{-1}), \tag{29}$$

which is at least $1/2$ for sufficiently large d;
3: Compute the mixed-softmax policy π for the samples in C (according to Theorem 4) and return π;

$X_1, \ldots, X_\gamma \sim \mathcal{D}$, with probability $\Omega(1/d)$, we will see Y_1 once among the samples, and the probability mass of Y_1 in $\mathcal{S}$ becomes $1/\gamma$, which is d times its true probability mass. Due to this discrepancy, we will show that a good sample policy for the empirical distribution $\mathcal{S}$ does not work as well on true distribution $\mathcal{D}$.

We consider the sample policy π such that $\pi(X) = \boldsymbol{e}_i$ when $X = Y_i$. When the event above happens, we have that $\mathbb{I}_{\mathcal{S}}(\pi) = \mathrm{diag}(1/\gamma, p_2, \ldots, p_d)$ where p_i is the probability mass for Y_i in $\mathcal{S}$ (for $i \ge 2$). When $\varepsilon = \sqrt{d/\gamma}$, we can verify that π is a good policy for the empirical distribution $\mathcal{S}$ since

$$\begin{aligned}\mathbb{V}^{(0)}_{\mathcal{S}}(\pi) &= \mathop{\mathbb{E}}_{X \sim \mathcal{S}} \max_{\boldsymbol{x} \in X} \boldsymbol{x}^\top \mathbb{I}_{\mathcal{S}}(\pi)^{-1} \boldsymbol{x} \\ &= \frac{1}{\gamma} \cdot \gamma + \sum_{i=2}^{d} p_i \cdot \max\{\varepsilon^2 \gamma + (1-\varepsilon^2) \cdot \frac{1}{p_i}, \frac{1}{p_i}\} \le O(d).\end{aligned}$$

However, for the true distribution $\mathcal{D}$, we have that $\mathbb{I}_{\mathcal{D}}(\pi) = \mathrm{diag}(1/(d\gamma), q, \ldots, q)$, and for any $\lambda \in [0, 1/(d\gamma))$, it holds that

$$\begin{aligned}\mathbb{V}^{(\lambda)}_{\mathcal{D}}(\pi) &= \mathop{\mathbb{E}}_{X \sim \mathcal{D}} \max_{\boldsymbol{x} \in X} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi))^{-1} \boldsymbol{x} \\ &= \frac{1}{d\gamma} \cdot \frac{1}{\lambda + 1/(d\gamma)} \\ &\quad + (1 - \frac{1}{d\gamma}) \cdot \max\{\varepsilon^2 \cdot \frac{1}{\lambda + 1/(d\gamma)} + (1-\varepsilon^2) \cdot \frac{1}{\lambda + q}, \frac{1}{\lambda + q}\} \\ &\ge \Omega(d^2).\end{aligned}$$

Note that in this example, the only constraint for γ is that $1/(d\gamma) > \lambda \Leftrightarrow \gamma < 1/(d\lambda)$. Therefore, we have illustrated that, even when γ is greater than an arbitrary polynomial of d, with probability $\Omega(1/d)$, a good policy for the empirical distribution $\mathcal{S}$ does not generalize to the true distribution $\mathcal{D}$.[11] By adding more dimensions, we can even strengthen this counterexample so that the failure probability becomes $(1 - o(1))$. Using similar tricks, we can also show that a good mixed-softmax policy does not generalize well.

[11] Although in our later algorithm, we only learn a policy with small λ-*deviation* as defined in (30), however, one can also verify that the λ-deviation of π over $\mathcal{D}$ in this counterexample is also high.

Algorithm 4: CoreIdentification

Input: $\lambda \in (0, 1)$, and $S = \{X_1, \ldots, X_\gamma\}$
Output: A core set $C \subseteq S$

1: $C_1 = S$;
2: **for** $\xi = 1, 2, 3, \ldots$ **do**
3: **if** C_ξ *satisfies (39)* **then return** C_ξ;
4: **else** $C_{\xi+1} = \{X_i \in C_\xi : \max_{\boldsymbol{x} \in X_i} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \frac{1}{\gamma} \sum_{X_i \in C_\xi} \mathbb{I}_{X_i}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{x} \le (1/2) d^c\}$;

Our Algorithm: CoreLearning. The key message from the counterexample above is that if a context direction in $\mathbb{R}^d$ appears with tiny probability in $\mathcal{D}$, a limited amount of samples might greatly change its probability in the empirical distribution $\mathcal{S}$, and fail the generalization argument. To address this issue, the idea of our new algorithm is to prune these infrequent context directions, learn a mixed-softmax policy over the remaining "core" directions, and finally argue that the infrequent directions can be properly handled by the π^{G} component in the mixed-softmax policy.

In light of this idea, we propose CoreLearning (Algorithm 3). In this algorithm, instead of directly learning the policy from the whole set of samples, we first find a large enough core set C at Line 2, and then learn the mixed-softmax policy only using the samples in C. The key property of the core is specified by (28), which is a technical realization of our pruning idea. The property requires that every direction in C should be well explored by the π^{G} policy and *only* the context vectors within C. To see how the core set helps to resolve the issue in our counterexample, we note that the infrequent set Y_1 is the main trouble-maker. However, even if Y_1 happens to appear among the samples $\{X_1, \ldots, X_\gamma\}$, it will not be included in the core since its corresponding variation $\max_{\boldsymbol{y} \in Y_1} \boldsymbol{y}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathrm{Unif}(C)}(\pi^{\mathrm{G}}))^{-1} \boldsymbol{y} \ge (\lambda + 1/\gamma)^{-1} > d^c$ when λ is sufficiently small and $\gamma \gg d^c$. Therefore, CoreLearning will learn a sample policy with Y_1 pruned away, and void our counterexample.

While the core set property (28) is much desirable, even whether such a core set with cardinality constraint (29) exists is not obvious. In Section 6.1, we state Lemma 6 to show its existence, and provide an efficient algorithm CoreIdentification to find one.

We now state the main theorem of this section (the guarantee for Algorithm 3).

Theorem 5. *Suppose that $\lambda \in (\exp(-d), 1)$. Let $X_1, \ldots, X_\gamma \sim \mathcal{D}$ be i.i.d. drawn from the distribution $\mathcal{D}$. Let π be the returned policy of Algorithm 3. We have that*

$$\begin{aligned}&\Pr\Big[\widetilde{\mathbb{V}}^{(\lambda)}_{\mathcal{D}}(\pi) \le O(\sqrt{d \log d})\Big] \\ &\quad \ge 1 - \exp\Big(O(d^4 \log^2 d) - \gamma d^{-2c} \cdot 2^{-16}\Big) \\ &\quad = 1 - \exp\Big(O(d^4 \log^2 d) - \gamma d^{-12} \cdot 2^{-16}\Big),\end{aligned}$$

where we define the λ-deviation of π over $\mathcal{D}$ by

$$\widetilde{\mathbb{V}}^{(\lambda)}_{\mathcal{D}}(\pi) \stackrel{\mathrm{def}}{=} \mathop{\mathbb{E}}_{X \sim \mathcal{D}} \sqrt{\max_{\boldsymbol{x} \in X} \{\boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi))^{-1} \boldsymbol{x}\}}. \tag{30}$$

Note that we are only able to provide the upper bound for $\widetilde{\mathbb{V}}^{(\lambda)}_{\mathcal{D}}(\pi)$ instead of $\mathbb{V}^{(\lambda)}_{\mathcal{D}}(\pi)$. However, this is still enough for our linear bandit application.

We now sketch the proof of Theorem 5, and the details can be found in the full version. For notational convenience, we define $\mathcal{S} \stackrel{\text{def}}{=} \text{Unif}(S)$, $\mathcal{C} \stackrel{\text{def}}{=} \text{Unif}(C)$, and we define the mollifier

$$\varphi_\beta(x) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{when } x \le \beta, \\ \frac{2\beta - x}{\beta}, & \text{when } \beta \le x \le 2\beta, \\ 0, & \text{when } x > 2\beta. \end{cases}$$

which is a continuous surrogate of the indicator function $\mathbf{1}[x \le \beta]$.

The proof of Theorem 5 consists of the following four steps.

Step I: Lower Bounding the Information Matrix. Via uniform concentration inequalities, we are able to prove that with probability $1 - \exp\big(O(d^3 \log d \log(d\lambda^{-1})) - \gamma d^{-2c} \cdot 2^{-16}\big)$, it holds that

$$\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi) \succcurlyeq \frac{1}{8}(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{S}}(\pi)). \tag{31}$$

Step II: Upper Bounding the Variation in the "Core Directions". Let $\boldsymbol{W} = \lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{S}}(\pi) \succcurlyeq \frac{1}{2}(\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{C}}(\pi))$. The goal of this step is to establish (36). Via uniform concentration inequalities, we can show that, with probability $1 - \exp\big(O(d^2 \log(d\lambda^{-1})) - \gamma d^{2-2c}/128\big)$, it holds that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{D}} \varphi_{4d^c}(\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}) \cdot \sqrt{\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}} \le d + \frac{1}{\gamma}\sum_{i=1}^{\gamma} \sqrt{\max_{\boldsymbol{x} \in X_i}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}}. \tag{32}$$

Let $\zeta = 1 - |C|/|S| = 1 - |C|/\gamma \le O(d^{3-c}\log(1/\lambda))$. Note that

$$\frac{1}{\gamma}\sum_{i=1}^{\gamma} \sqrt{\max_{\boldsymbol{x} \in X_i}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}} \le \frac{1}{\gamma}\sum_{X_i \in C} \sqrt{\max_{\boldsymbol{x} \in X_i}\{2\boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_C(\pi))^{-1}\boldsymbol{x}\}} + \frac{1}{\gamma}\sum_{X_i \in S \setminus C} \sqrt{\max_{\boldsymbol{x} \in X_i}\{\boldsymbol{x}^\top (\lambda \boldsymbol{I} + (\zeta/2)\,\mathbb{I}_{\text{Unif}(S\setminus C)}(\pi^{\mathrm{G}}))^{-1}\boldsymbol{x}\}}. \tag{33}$$

For the first term in (33), by the guarantee of Theorem 4, we have

$$\frac{1}{\gamma}\sum_{X_i \in C} \sqrt{\max_{\boldsymbol{x} \in X_i}\{2\boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_C(\pi))^{-1}\boldsymbol{x}\}} \le O(\sqrt{d \log d}). \tag{34}$$

For the second term in (33), by the variation bound for π^{G} (Lemma 4), we can prove that

$$\frac{1}{\gamma}\sum_{X_i \in S \setminus C} \sqrt{\max_{\boldsymbol{x} \in X_i}\{\boldsymbol{x}^\top (\lambda \boldsymbol{I} + (\zeta/2)\,\mathbb{I}_{\text{Unif}(S\setminus C)}(\pi^{\mathrm{G}}))^{-1}\boldsymbol{x}\}} \le O(\sqrt{d}). \tag{35}$$

Combining (32), (33), (34), (35), we have that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{D}} \varphi_{4d^c}(\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}) \cdot \sqrt{\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}} \le O(\sqrt{d \log d}). \tag{36}$$

Step III: Upper Bounding the Variation in the "Infrequent Directions". The goal of this step is to establish (38). Via concentration inequalities, we can prove that, with probability

$$1 - \exp\Big(O(d^2 \log\big(d\lambda^{-1}\big)) - \gamma d^{-2}/2\Big),$$

it holds that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{D}} \varphi_{4d^c}(\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}) \ge 1 - O(d^{-1}).$$

Let $\tau_X = 1 - \varphi_{4d^c}(\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\})$. We have that $\mathbb{E}_{X \sim \mathcal{D}} \tau_X \le O(d^{-1})$. Using Cauchy-Schwarz and by the variation bound for π^{G} (Lemma 4), we can show that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{D}} \tau_X \sqrt{\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}} \le O(\sqrt{d}). \tag{37}$$

Altogether, we have that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{D}} (1 - \varphi_{4d^c}(\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\})) \cdot \max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\} \le O(\sqrt{d}). \tag{38}$$

Step IV: Putting Things Together. Combining (36) and (38), we have

$$\mathop{\mathbb{E}}_{X \sim \mathcal{D}} \sqrt{\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top \boldsymbol{W}^{-1}\boldsymbol{x}\}} \le O(\sqrt{d \log d}).$$

By the definition of $\boldsymbol{W}$, and together with (31), we have that

$$\widetilde{\mathbb{V}}^{(\lambda)}_{\mathcal{D}}(\pi) = \mathop{\mathbb{E}}_{X \sim \mathcal{D}} \sqrt{\max_{\boldsymbol{x} \in X}\{\boldsymbol{x}^\top (\lambda \boldsymbol{I} + \mathbb{I}_{\mathcal{D}}(\pi))^{-1}\boldsymbol{x}\}} \le O(\sqrt{d \log d}),$$

proving Theorem 5.

6.1 Finding the Core

We now present our algorithm (CoreIdentification, Algorithm 4) to find the core, and state the following lemma as its guarantee.

Lemma 6. *Let $S = \{X_1, \ldots, X_\gamma\}$ be a sequence/multi-set of context sets. Algorithm 4 finds a core set $C \subseteq S$ in $O(d \log \lambda^{-1})$ iterations that satisfies (29) and*

$$\max_{X_i \in C} \max_{\boldsymbol{x} \in X_i} \boldsymbol{x}^\top (\lambda \boldsymbol{I} + \frac{1}{\gamma}\sum_{X_i \in C} \mathbb{I}_{X_i}(\pi^{\mathrm{G}}))^{-1}\boldsymbol{x} \le d^c. \tag{39}$$

We remark that (39) implies (28), because $\frac{1}{\gamma}\sum_{X_i \in C} \mathbb{I}_{X_i}(\pi^{\mathrm{G}}) \preccurlyeq \mathbb{I}_{\text{Unif}(C)}(\pi^{\mathrm{G}})$. The proof of Lemma 6 is deferred to the full version due to space constraints.

7 PUTTING EVERYTHING TOGETHER: THE OPTIMAL BATCH ALGORITHM

Our final algorithm with $O(\log \log T)$ static-grid batches and optimal minimax expected regret (up to poly log T factors) is presented in Algorithm 5. Compared with BatchLinUCB and BatchLinUCB-KW, the main difference here is the addition of from Line 11 to Line 16, which not only learns the new estimate $\hat{\boldsymbol{\theta}}_k$, but also the new sample policy π_k. Learning of the two objects are done through disjoint sets of samples ($\mathcal{A}$ and $\mathcal{B}$). This is because that $\mathcal{D}_k$ depends on $\hat{\boldsymbol{\theta}}_k$ (which is learned from $\mathcal{A}$) and we have to make $\mathcal{B}$ disjoint from $\mathcal{A}$ so as to ensure elements in S are independently sampled from $\mathcal{D}_k$. The following theorem bounds the expected regret of Algorithm 5.

Algorithm 5: BatchLinUCB-DG

$M = \lceil \log\log T \rceil + 1$, $\alpha \leftarrow 10\sqrt{\ln \frac{2dKT}{\delta}}$, $\pi^0 = \pi^{\mathrm{G}}$, $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$, where $\mathcal{T}_0 = 0$, $\mathcal{T}_1 = \sqrt{T}$, $\mathcal{T}_2 = 2\sqrt{T}$, and $\mathcal{T}_i = T^{1-2^{-(i-1)}}$ for $i \in \{3, \ldots, M-1\}$, $\mathcal{T}_M = T$;
for $k \leftarrow 1, 2, \ldots, M$ **do**
  for $t \leftarrow \mathcal{T}_{k-1} + 1, \mathcal{T}_{k-1} + 2, \ldots, \mathcal{T}_k$ **do**
    $A_t^{(0)} \leftarrow [K]$, $\hat{r}_{ti}^{(0)} \leftarrow 0, \omega_{ti}^{(0)} \leftarrow 1$;
    for $\kappa \leftarrow 1, 2, \ldots, k-1$ **do** ▷ Eliminate
      $\forall i \in A_t^{(\kappa-1)} : \hat{r}_{ti}^{(\kappa)} \leftarrow \boldsymbol{x}_{ti}^\top \hat{\boldsymbol{\theta}}_\kappa, \omega_{ti}^{(\kappa)} \leftarrow \alpha\sqrt{\boldsymbol{x}_{ti}^\top \boldsymbol{\Lambda}_\kappa^{-1} \boldsymbol{x}_{ti}}$;
      $A_t^{(\kappa)} \leftarrow \{i \in A_t^{(\kappa-1)} \mid \hat{r}_{ti}^{(\kappa)} + \omega_{ti}^{(\kappa)} \geq \hat{r}_{tj}^{(\kappa)} - \omega_{tj}^{(\kappa)}, \forall j \in A_t^{(\kappa-1)}\}$;
    $A_t \leftarrow A_t^{(k-1)}$;
    Select i_t such that $\boldsymbol{x}_{t,i_t} \sim \pi_{k-1}(\{\boldsymbol{x}_{t,i} : i \in A_t\})$, play arm i_t, and receive reward r_t;
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t,i_t}$;
  Evenly divide $\{\mathcal{T}_{k-1} + 1, \ldots, \mathcal{T}_k\}$ into two sets $\mathcal{A}, \mathcal{B}$;
  $\lambda \leftarrow 32\ln(2dT/\delta)$, $\boldsymbol{\Lambda}_k \leftarrow \lambda \boldsymbol{I} + \sum_{\tau \in \mathcal{A}} \boldsymbol{x}_\tau \boldsymbol{x}_\tau^\top$, $\boldsymbol{\xi}_k \leftarrow \sum_{\tau \in \mathcal{A}} r_\tau \boldsymbol{x}_\tau$, $\hat{\boldsymbol{\theta}}_k \leftarrow \boldsymbol{\Lambda}_k^{-1} \boldsymbol{\xi}_k$;
  for $\tau \in \mathcal{B}$ **do**
    $\forall i \in A_\tau^{(k-1)} : \hat{r}_{ti}^{(k)} \leftarrow \boldsymbol{x}_{ti}^\top \hat{\boldsymbol{\theta}}_k, \omega_{ti}^{(k)} \leftarrow \alpha\sqrt{\boldsymbol{x}_{ti}^\top \boldsymbol{\Lambda}_k^{-1} \boldsymbol{x}_{ti}}$;
    $A_t^{(k)} \leftarrow \{i \in A_t^{(k-1)} \mid \hat{r}_{ti}^{(k)} + \omega_{ti}^{(k)} \geq \hat{r}_{tj}^{(k)} - \omega_{tj}^{(k)}, \forall j \in A_t^{(k-1)}\}$;
  Use the context sets $S = \{\{\boldsymbol{x}_{\tau,a} \mid a \in A_\tau^{(k)}\}\}_{\tau \in \mathcal{B}}$ and $\lambda = 1/T$ as the input of Algorithm 3 and learn the sample policy π_k;

Theorem 6. *Assume that $T \leq \Omega(\max\{e^d, d^{32}\log^4 d \log^2 \delta^{-1}\})$. With probability at least $(1-\delta)$, the expected regret of Algorithm 5 is bounded as*

$$R_{\text{BatchLinUCB-DG}}^T \leq O(\sqrt{dT\log d \log(dKT/\delta)} \times \log\log T).$$

Note that the assumption that $T \leq \exp(d)$ is not restrictive since otherwise we have $\log T \geq \Omega(d)$ and BatchLinUCB-KW (Theorem 2) already achieves the minimax optimal regret up to poly $\log T$ factors. We also note that the K in the regret bound can be replaced by $\min\{K, d\log T\}$ by a simple ε-net argument, so that our regret bound becomes minimax-optimal for all K (up to poly $\log T$ factors).

Proof of Theorem 6. We adopt the notations in Section 4. Conditioned on the batches $1, 2, \ldots, k-1$, we can bound the expected regret incurred in batch k similarly as (5), and have that with probability at least $(1 - \delta\mathcal{T}_k/T^2)$,

$$R_k \leq 4\alpha\mathcal{T}_k \times \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1} \boldsymbol{x}}. \tag{40}$$

Furthermore, similar to Lemma 2, we can show that for each batch k ($k < M$), with probability $(1 - \delta/T^2)$, we have that

$$\boldsymbol{\Lambda}_k \succcurlyeq \frac{\mathcal{T}_k}{32}\left(\frac{\ln T}{\mathcal{T}_k}\boldsymbol{I} + \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\pi_{k-1}(X)} [\boldsymbol{x}\boldsymbol{x}^\top]\right)$$

$$\succcurlyeq \frac{\mathcal{T}_k}{32}\left(T^{-1}\cdot\boldsymbol{I} + \mathbb{I}_{\mathcal{D}_{k-1}}(\pi_{k-1})\right). \tag{41}$$

Note that compared with (6), (41) has a worse constant 32 since $\mathcal{A}$ only contains half of the samples.

For each $k < M$, note that at Line 16, $S = \{\{\boldsymbol{x}_{\tau,a} \mid a \in A_\tau^{(k)}\}\}_{\tau\in\mathcal{B}}$ contains *i.i.d.* samples from $\mathcal{D}_k$, and $|S| \geq |\mathcal{T}_k - \mathcal{T}_{k-1}|/2 \geq \sqrt{T}/4$. By Theorem 5, we have that with probability $1 - \exp(O(d^4\log^2 d) - \sqrt{T}d^{-12}\cdot 2^{-18}) \geq 1 - \delta/T^2$ (since $T \geq \Omega(d^{32}\log^4 d \log^2(\delta^{-1}))$, it holds that

$$\widetilde{\mathbb{V}}_{\mathcal{D}_k}^{(1/T)}(\pi_k) \leq O(\sqrt{d\log d}). \tag{42}$$

The expected regret incurred during batch 1 and batch 2 is at most $2\sqrt{T}$. For any $k \geq 3$, assuming (40) holds for batch k, and (41) and (42) hold for batch $(k-1)$, we have that

$$\begin{aligned} R_k &\leq 4\alpha\mathcal{T}_k \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \boldsymbol{\Lambda}_{k-1}^{-1}\boldsymbol{x}} \\ &\leq \frac{4\sqrt{32}\alpha\mathcal{T}_k}{\sqrt{\mathcal{T}_{k-1}}} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-1}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \left(T^{-1}\boldsymbol{I} + \mathbb{I}_{\mathcal{D}_{k-2}}(\pi_{k-2})\right)^{-1}\boldsymbol{x}} \\ &\leq 32\alpha\sqrt{T}\cdot \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{k-2}} \max_{\boldsymbol{x}\in X} \sqrt{\boldsymbol{x}^\top \left(T^{-1}\boldsymbol{I} + \mathbb{I}_{\mathcal{D}_{k-2}}(\pi_{k-2})\right)^{-1}\boldsymbol{x}} \quad (43) \\ &\leq 32\alpha\sqrt{T}\cdot \widetilde{\mathbb{V}}_{\mathcal{D}_{k-1}}^{(1/T)}(\pi_{k-1}) \leq O(\sqrt{dT\log d\log(dKT/\delta)}), \end{aligned}$$

where (43) is because that $X \sim \mathcal{D}_{k-1}$ can be sampled via first drawing $X' \sim \mathcal{D}_{k-2}$, then performing one-step elimination on X', and getting $X \subseteq X'$.

Finally, collecting the failure probabilities for all $O(\log\log T)$ batches, we prove the desired regret bound. □

ACKNOWLEDGMENTS

We thank Yanjun Han, Zhengyuan Zhou, and Zhengqing Zhou for their valuable comments. Yufei Ruan and Yuan Zhou are supported in part by NSF CCF-2006526.

REFERENCES

[1] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.), Vol. 24. Curran Associates, Inc., 2312–2320.
[2] Naoki Abe, Alan W. Biermann, and Philip M. Long. 2003. Reinforcement Learning with Immediate Rewards and Linear Hypotheses. *Algorithmica* 37, 4 (Dec. 2003), 263–293. https://doi.org/10.1007/s00453-003-1038-1
[3] Naoki Abe and Philip M. Long. 1999. Associative Reinforcement Learning Using Linear Probabilistic Concepts. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML '99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3–11.
[4] Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. 2017. Learning with Limited Rounds of Adaptivity: Coin Tossing, Multi-Armed Bandits, and Ranking from Pairwise Comparisons. In *Proceedings of the 2017 Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 65)*, Satyen Kale and Ohad Shamir (Eds.). PMLR, Amsterdam, Netherlands, 39–75.
[5] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. 2020. Near-optimal discrete optimization for experimental design: A regret minimization approach. *Mathematical Programming* (2020), 1–40.
[6] Anthony Atkinson, Alexander Donev, and Randall Tobias. 2007. *Optimum experimental designs, with SAS*. Vol. 34. Oxford University Press.
[7] Peter Auer. 2003. Using Confidence Bounds for Exploitation-Exploration Trade-Offs. *Journal of Machine Learning Research* 3 (March 2003), 397–422.
[8] Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. 2019. Provably Efficient Q-Learning with Low Switching Cost. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 8004–8013.

[9] Nicolò Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. 2013. Online Learning with Switching Costs and Other Adaptive Adversaries. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc., 1160–1168.

[10] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual Bandits with Linear Payoff Functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 15)*, Geoffrey Gordon, David Dunson, and Miroslav Dudík (Eds.). JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 208–214.

[11] Ali Çivril and Malik Magdon-Ismail. 2009. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science* 410, 47-49 (2009), 4801–4811.

[12] Varsha Dani, Thomas P Hayes, and Sham M Kakade. 2008. Stochastic Linear Optimization under Bandit Feedback. In *Conference on Learning Theory*.

[13] Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. 2014. Bandits with Switching Costs: T2/3 Regret. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (New York, New York) *(STOC '14)*. Association for Computing Machinery, New York, NY, USA, 459–467. https://doi.org/10.1145/2591796.2591868

[14] Kefan Dong, Yingkai Li, Qin Zhang, and Yuan Zhou. 2020. Multinomial Logit Bandit with Low Switching Cost. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 2607–2615.

[15] John Duchi, Feng Ruan, and Chulhee Yun. 2018. Minimax Bounds on Stochastic Batched Convex Optimization. In *Proceedings of the 31st Conference On Learning Theory (Proceedings of Machine Learning Research, Vol. 75)*, Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet (Eds.). PMLR, 3065–3162.

[16] Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. 2019. Regret Bounds for Batched Bandits. *arXiv preprint arXiv:1910.04959* (2019).

[17] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. 2019. Batched Multi-armed Bandits Problem. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 503–513.

[18] Zhaohan Guo and Emma Brunskill. 2015. Concurrent PAC RL. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2624–2630.

[19] Yanjun Han, Zhengqing Zhou, Zhengyuan Zhou, Jose Blanchet, Peter W Glynn, and Yinyu Ye. 2020. Sequential Batch Learning in Finite-Action Linear Contextual Bandits. *arXiv preprint arXiv:2004.06321* (2020).

[20] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. 2013. Distributed Exploration in Multi-Armed Bandits. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc., 854–862.

[21] Kwang-Sung Jun, Kevin Jamieson, Robert Nowak, and Xiaojin Zhu. 2016. Top Arm Identification in Multi-Armed Bandits with Batch Arm Pulls. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 51)*, Arthur Gretton and Christian C. Robert (Eds.). PMLR, Cadiz, Spain, 139–148.

[22] Nikolai Karpov, Qin Zhang, and Yuan Zhou. 2020. Collaborative Top Distribution Identifications with Limited Interaction (Extended Abstract). In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. 160–171. https://doi.org/10.1109/FOCS46700.2020.00024

[23] J. Kiefer and J. Wolfowitz. 1960. The Equivalence of Two Extremum Problems. *Canadian Journal of Mathematics* 12 (1960), 363–366. https://doi.org/10.4153/CJM-1960-030-4

[24] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit Algorithms*. Cambridge University Press. https://doi.org/10.1017/9781108571401

[25] Yingkai Li, Yining Wang, and Yuan Zhou. 2019. Nearly Minimax-Optimal Regret for Linearly Parameterized Bandits. In *Proceedings of the Thirty-Second Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 99)*, Alina Beygelzimer and Daniel Hsu (Eds.). PMLR, Phoenix, USA, 2173–2174.

[26] Vivek Madan, Mohit Singh, Uthaipon Tantipongpipat, and Weijun Xie. 2019. Combinatorial Algorithms for Optimal Design. In *Proceedings of the Thirty-Second Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 99)*, Alina Beygelzimer and Daniel Hsu (Eds.). PMLR, Phoenix, USA, 2210–2258.

[27] Aleksandar Nikolov, Mohit Singh, and Uthaipon Tao Tantipongpipat. [n.d.]. *Proportional Volume Sampling and Approximation Algorithms for A-Optimal Design*. 1369–1386. https://doi.org/10.1137/1.9781611975482.84

[28] Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. 2016. Batched bandit problems. *The Annals of Statistics* 44, 2 (2016), 660 – 681. https://doi.org/10.1214/15-AOS1381

[29] Friedrich Pukelsheim. 2006. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719109

[30] Paat Rusmevichientong and John N. Tsitsiklis. 2010. Linearly Parameterized Bandits. *Mathematics of Operations Research* 35, 2 (2010), 395–411. https://doi.org/10.1287/moor.1100.0446

[31] David Simchi-Levi and Yunzong Xu. 2019. Phase Transitions and Cyclic Phenomena in Bandits with Switching Constraints. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 7523–7532.

[32] David Simchi-Levi and Yunzong Xu. 2020. Bypassing the Monster: A Faster and Simpler Optimal Algorithm for Contextual Bandits under Realizability. *Available at SSRN* (2020).

[33] Mohit Singh and Weijun Xie. 2018. Approximate Positive Correlated Distributions and Approximation Algorithms for D-Optimal Design. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (New Orleans, Louisiana) *(SODA '18)*. Society for Industrial and Applied Mathematics, USA, 2240–2255.

[34] Marta Soare, Alessandro Lazaric, and Remi Munos. 2014. Best-Arm Identification in Linear Bandits. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc., 828–836.

[35] Marco Di Summa, Friedrich Eisenbrand, Yuri Faenza, and Carsten Moldenhauer. 2015. On Largest Volume Simplices and Sub-Determinants. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms* (San Diego, California) *(SODA '15)*. Society for Industrial and Applied Mathematics, USA, 315–323.

[36] Chao Tao, Saúl Blanco, and Yuan Zhou. 2018. Best Arm Identification in Linear Bandits with Linear Dimension Dependency. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 4877–4886.

[37] Chao Tao, Qin Zhang, and Yuan Zhou. 2019. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 126–146.

[38] Yining Wang, Adams Wei Yu, and Aarti Singh. 2017. On Computationally Tractable Selection of Experiments in Measurement-Constrained Regression Models. *Journal of Machine Learning Research* 18, 143 (2017), 1–41.

[39] William J. Welch. 1982. Algorithmic complexity: three NP-hard problems in computational statistics. *Journal of Statistical Computation and Simulation* 15, 1 (1982), 17–25. https://doi.org/10.1080/00949658208810560

[40] Liyuan Xu, Junya Honda, and Masashi Sugiyama. 2018. A fully adaptive algorithm for pure exploration in linear bandits. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 84)*, Amos Storkey and Fernando Perez-Cruz (Eds.). PMLR, 843–851.

[41] Zihan Zhang, Yuan Zhou, and Xiangyang Ji. 2020. Almost Optimal Model-Free Reinforcement Learning via Reference-Advantage Decomposition. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 15198–15207.