Optimal Batched Linear Bandits

Xuanfei Ren [1] Tianyuan Jin [2] Pan Xu [3]

Abstract

We introduce the E^4 algorithm for the batched linear bandit problem, incorporating an Explore-Estimate-Eliminate-Exploit framework. With a proper choice of exploration rate, we prove E^4 achieves the finite-time minimax optimal regret with only $O(\log\log T)$ batches, and the asymptotically optimal regret with only 3 batches as $T \to \infty$, where T is the time horizon. We further prove a lower bound on the batch complexity of linear contextual bandits showing that any asymptotically optimal algorithm must require at least 3 batches in expectation as $T \to \infty$, which indicates E^4 achieves the asymptotic optimality in regret and batch complexity simultaneously. To the best of our knowledge, E^4 is the first algorithm for linear bandits that simultaneously achieves the minimax and asymptotic optimality in regret with the corresponding optimal batch complexities. In addition, we show that with another choice of exploration rate E^4 achieves an instance-dependent regret bound requiring at most $O(\log T)$ batches, and maintains the minimax optimality and asymptotic optimality. We conduct thorough experiments to evaluate our algorithm on randomly generated instances and the challenging *End of Optimism* instances (Lattimore & Szepesvari, 2017) which were shown to be hard to learn for optimism based algorithms. Empirical results show that E^4 consistently outperforms baseline algorithms with respect to regret minimization, batch complexity, and computational efficiency.

1. Introduction

Sequential decision-making problems including bandits and reinforcement learning (Bubeck et al., 2012; Slivkins et al., 2019; Lattimore & Szepesvári, 2020) have become pivotal in modeling decision processes where an agent actively interacts with an uncertain environment to achieve a long-term goal. In particular, we study the linear contextual bandit problems (Langford & Zhang, 2007; Li et al., 2010; Abbasi-Yadkori et al., 2011; Chu et al., 2011; Agrawal & Goyal, 2013; Kirschner et al., 2021; Xu et al., 2022), where the agent can choose from K arms to play at each step, and each arm is associated with a feature vector in the $\mathbb{R}^d$ space. Each arm when played emits a noisy reward with its mean assumed to be a linear function of the feature vector with the unknown weight parameter shared across different arms. These bandit problems are prevalent in numerous real-world applications, from online advertising (Abe et al., 2003) to recommendation systems (Agarwal et al., 2008; Li et al., 2010), where abundant side information is available and thus provides powerful feature representation of each arm. The most critical dilemma in designing algorithms for learning bandit problems is the trade-off between exploitation and exploration, where the agent must decide whether to exploit the currently known best option or to explore unknown possibly suboptimal options for maximizing the long-run gains.

In conventional bandit setting, the agent adjusts its strategy or policy step-by-step, by first choosing an arm to play, observing a reward for this arm, and adjusting its strategy accordingly in the immediate next step. This ideal setting is also referred to as the fully-sequential setting, where immediate outcome can be obtained and switch polices is cost-efficient. In real-world problems, fully sequential learning strategies are usually expensive or infeasible in critical applications where the interaction with the environment takes months to observe the rewards like in clinical trials or medical treatment (Robbins, 1952), or switching between different policies is expensive in e-commerce (Bertsimas & Mersereau, 2007). Therefore, the batched bandit setting is more realistic and feasible, where an agent selects a batch of actions rather than a single action at one decision point to play, which significantly minimizes the number of policy switching and expedites the learning process by enabling parallel experiments (Perchet et al., 2015; Gao et al., 2019; Esfandiari et al., 2021; Han et al., 2020; Jin et al., 2021b;a; Ruan et al., 2021; Jin et al., 2023).

To evaluate the effectiveness of bandit algorithms, a crucial metric is the *regret*, which represents the gap between

[1] University of Science and Technology of China [2] National University of Singapore [3] Duke University. Correspondence to: Pan Xu <pan.xu@duke.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

the expected cumulative rewards of the learning agent and those of an oracle who knows the optimal action at each step in hindsight. We denote the regret as R_T, where T is the learning time horizon. The goal of a bandit algorithm is to find a tight bound for the regret to guarantee good performance theoretically. Generally speaking, there are three types of bounds of regret that are considered in the bandit literature. The first type of regret bound is called the worst-case regret bound where we measure the performance of a bandit algorithm with respect to the worst possible bandit instance. For linear bandits with K arms, it is well-established (Lattimore & Szepesvári, 2020) that there exists a bandit instance such that, for all bandit strategies denoted by π, the regret satisfies $R_T \geq \Omega(\sqrt{dT})$. The second type of regret is called the instance-dependent regret bound, where the regret R_T is bounded by problem dependent parameters including the time horizon, the mean rewards of different arms, the number of arms, etc. Instance-dependent regret bounds provide performance evaluation of a bandit algorithm specific to certain bandit instances, offering more delicate insights into an algorithm's behavior in practice. The above regret bounds are both in the finite-time regime, where we fixed the time horizon T. The third type of regret bound, the asymptotic regret bound, characterizes the performance of an algorithm when T goes infinite. It has been proved that the regret of any consistent algorithm satisfies $\liminf_{T\to\infty} R_T/\log T \geq c^*$, where $c^* := c^*(\boldsymbol{\theta}^*)$ is a statistical complexity measure of the problem under the bandit instance $\boldsymbol{\theta}^*$ (Graves & Lai, 1997; Combes et al., 2017; Lattimore & Szepesvari, 2017), which we provide a formal definition in Section 3.

Achieving the optimal regret across various regret types poses a significant challenge. Notably, instance-dependent analysis proves to be more intricate than worst-case analysis. Many algorithms, while considered minimax optimal, may encounter difficulties in specific instances (Lattimore & Szepesvari, 2017). Conversely, certain existing instance-optimal algorithms lack a guarantee for worst-case regret (Wagenmaker & Foster, 2023). It is even more challenging to achieve optimal regret for batched bandit algorithms. Due to their batch design, learning agents cannot update the exploration policy after each data point, resulting in a rarely-switching nature. In this paper, we show that we can achieve the same order of all three types of regret as full-sequential algorithm but with much fewer batches. In particular, we first provide a lower bound on the batch complexity for asymptotic algorithms, and then propose an algorithm that attains both non-asymptotic and asymptotic optimal regret simultaneously and attains the corresponding optimal batch complexities. At the key of our algorithm is a careful design of the Explore-Estimate-Eliminate-Exploit framework leveraging D-optimal design exploration, optimal allocation learning, and arm elimination.

Our contributions are summarized as follows.

- We propose the algorithm, Explore-Estimate-Eliminate-Exploit (E^4), for solving batched linear contextual bandits. With a proper choice of exploration rate, we prove that E^4 attains $\tilde{O}(\sqrt{dT})$ worst-case regret after only $O(\log\log T)$ batches, which matches the existing lower bounds on the regret and batch complexity for linear contextual bandits with finite arms (Gao et al., 2019). Under the same configuration, when $T \to \infty$, E^4 achieves the asymptotically optimal regret with only 3 batches. We also prove that any asymptotically optimal algorithm must have at least 3 batches in expectation as $T \to \infty$. Therefore, E^4 is the first algorithm in linear bandits that achieves simultaneously optimal in terms of regret bound and batch complexity in both the finite-time worst-case setting and the asymptotic setting. We present the comparison of regret and batch complexities with existing work in Table 1 for the readers' reference.
- With a different exploration rate, we prove that our E^4 algorithm attains an instance-dependent regret bound $O(d\log(KT)/\Delta_{\min})$ with at most $O(\log T)$ batches, where $\Delta_{\min}$ is the minimum gap between the mean rewards of the best arm and the suboptimal arms. Under the same configuration, E^4 also maintains the minimax optimality and asymptotic optimality. Our result nearly matches the batch complexity lower bound (Gao et al., 2019) that states any algorithm with $O(d\log T/\Delta_{\min})$ regret must have at least $\Omega(\log T)$ batches.
- We conduct experiments on challenging linear bandit problems, including the *End of Optimism* instances (Lattimore & Szepesvari, 2017), where it has been demonstrated that algorithms based on optimism are suboptimal, as well as on a suite of randomly generated instances. Our empirical evaluation verifies that E^4 not only outperforms the existing baselines in terms of regret and batch complexity but attains superior computational efficiency by exhibiting a significant speedup over baseline algorithms.

Notation We denote a set $\{1,\ldots,K\}$ as $[K]$, $K \in \mathbb{N}^+$. We use bold letter $\mathbf{x} \in \mathbb{R}^d$ to denote a vector. $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top\mathbf{x}}$ is its Euclidean norm. For a semipositive definite matrix $\mathbf{V} \in \mathbb{R}^{d\times d}$, $\|\mathbf{x}\|_{\mathbf{V}} = \sqrt{\mathbf{x}^\top\mathbf{V}\mathbf{x}}$ is the Mahalanobis norm. For an event E within a probability space, we denote its complement as E^c. For $f(T)$ and $g(T)$ as functions of T, we write $g(T) = O(f(T))$ to imply that there is a constant C independent of T such that $g(T) \leq Cf(T)$, and use $\tilde{O}(f(T))$ to omit logarithmic dependencies. We also define $f(T) = \Omega(g(T))$ if $g(T) = O(f(T))$. When both $f(T) = O(g(T))$ and $g(T) = O(f(T))$ hold, we write $f(T) = \Theta(g(T))$. We assume $0 \times \infty = 0$.

Table 1: Regret and batch complexity comparison in the finite-time and asymptotic settings for linear bandits.

Algorithm	Non-asymptotic setting		Asymptotic setting	
	Worst-case regret	Batch complexity	Asymptotic regret	Batch complexity
Abbasi-Yadkori et al. (2011)	$\tilde{O}(d\sqrt{T})$	$O(\log T)$	-	-
Esfandiari et al. (2021)	$\tilde{O}(\sqrt{dT})$	$O(\log T)$	-	-
Ruan et al. (2021)	$\tilde{O}(\sqrt{dT})$	$O(\log\log T)$	-	-
Hanna et al. (2023)	$\tilde{O}(\sqrt{dT})$	$O(\log\log T)$	-	-
Lower bound (Gao et al., 2019)	$\Omega(\sqrt{dT})$	$\Omega(\log\log T)$	-	-
Lattimore & Szepesvari (2017)	-	-	Optimal	Sequential
OSSB (Combes et al., 2017)	-	-	Optimal	Sequential
OAM (Hao et al., 2020)	-	-	Optimal	Sequential
SOLID (Tirinzoni et al., 2020)	$\tilde{O}((d+\log K)\sqrt{T})$	Sequential	Optimal	Sequential
IDS (Kirschner et al., 2021)	$\tilde{O}(d\sqrt{T})$	$\geq O\big(d^4\log^4 T/\Delta_{\min}^2\big)$	Optimal	$\geq O\big(\log^4 T\big)$
Lower bound (Theorem 5.2)	-	-	Optimal	≥ 3
E^4 (Algorithm 1)	$\tilde{O}(\sqrt{dT})$	$O(\log\log T)$	Optimal	3

2. Related Work

Existing works on batched linear bandits predominantly concentrated on achieving non-asymptotic optimal regret with minimal batch complexity. Gao et al. (2019) argued that, for an algorithm to achieve minimax optimality, it should incorporate at least $\Theta(\log\log T)$ batches. Additionally, to attain an instance-dependent regret of order $O(d\log T/\Delta_{\min})$, an algorithm should include at least $\Theta(\log T)$ batches. In the contextual linear bandits setting, Han et al. (2020); Ruan et al. (2021); Zhang et al. (2021); Hanna et al. (2023) delved into linear bandits with both stochastic and adversarial contexts. However, the techniques proposed by Han et al. (2020) cannot be directly applied to our fixed K arms setting, as it assumes that each arm is i.i.d. sampled from some distribution. Ruan et al. (2021); Zhang et al. (2021); Hanna et al. (2023) researched on stochastic contextual linear bandit problems, and our setting can be seen as a special case of theirs when the number of contexts is one, meaning the arm set is fixed. They both achieved $O(d\sqrt{T})$ regret bound with $O(\log\log T)$ batches. In the context of linear bandits with fixed K arms, a simple algorithm with $O(\log T)$ batch complexity achieves an optimal regret bound of $\tilde{O}(\sqrt{dT})$ (Esfandiari et al., 2021; Lattimore & Szepesvári, 2020). However, this falls short of the optimality compared to the lower bound in Gao et al. (2019).

Another related line of research is the algorithms that achieve asymptotically optimal regret for linear bandits (Lattimore & Szepesvari, 2017; Combes et al., 2017; Hao et al., 2020; Tirinzoni et al., 2020; Kirschner et al., 2021; Wagenmaker & Foster, 2023), which strive for the best possible performance in a sufficiently large horizon. Nevertheless, traditional well-performing optimistic algorithms like Upper Confidence Bound (UCB) and Thompson Sampling (TS) have been proven to fall short of achieving asymptotic optimality (Lattimore & Szepesvari, 2017). Lattimore & Szepesvari (2017) demonstrated an asymptotic lower bound for the regret of any consistent algorithm and proposed a simple algorithm with a matching upper bound, but no finite-time performance was guaranteed. Similarly, Combes et al. (2017) achieved asymptotic optimality for a large class of bandit problems, but no finite-time regret guarantees were provided either. Hao et al. (2020) achieved asymptotic optimality with good empirical performance within finite horizon, considering a contextual bandit setting with a finite number of contexts, sharing similarities with our setting. They also proved that an optimistic algorithm could have bounded regret when the set of optimal arms spans the arm space, aligning with the concept of the *End of Optimism* phenomenon (Lattimore & Szepesvari, 2017).

Further, Tirinzoni et al. (2020) provided the first asymptotically optimal regret bound with non-asymptotic instance-dependent and worst-case regret bounds, achieving the minimax optimality. Kirschner et al. (2021) proposed the frequentist IDS algorithm that leverages an information theoretic method, achieving minimax and asymptotic optimality simultaneously with a non-asymptotic instance-dependent regret bound. Wagenmaker & Foster (2023) proposed a method for achieving instance optimality in both asymptotic and non-asymptotic settings, along with a unifying framework for non-asymptotic instance optimality. Even so, they fell short of achieving minimax optimality. More importantly, all of the mentioned works that focus on asymptotic regret analysis are designed in a fully sequential way and cannot be applied to the batched linear bandit setting.

3. Preliminary

In this paper, we study the linear contextual bandit problem, where the reward of an arm $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ is given by $r(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle + \varepsilon$. Here $\mathbf{x}$ is the arm feature, $\mathcal{X}$ is the arm set with $|\mathcal{X}| = K$ arms in total, $\boldsymbol{\theta}^* \in \mathbb{R}^d$ is an unknown weight parameter, and ε is a zero-mean 1-subgaussian noise. For simplicity, we assume $\mathcal{X}$ spans $\mathbb{R}^d$. Otherwise, we can always find a corresponding arm set with a lower dimension that does by applying an orthogonal transformation to both $\boldsymbol{\theta}^*$ and actions. For normalization purposes, we assume bounded expected rewards: $|\langle \mathbf{x}, \boldsymbol{\theta}^* \rangle| \leq L, \forall \mathbf{x} \in \mathcal{X}$, where $L > 0$ is some constant.

Conventional bandit algorithms operate in a fully sequential way. In each round $t = 1, \ldots, T$, the learner chooses an action $\mathbf{x}_t \in \mathcal{X}$ and then observes a reward $r_t = r(\mathbf{x}_t)$. The objective is to minimize the *accumulated regret* of the algorithm defined as $R_T = \mathbb{E}[\max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \langle \mathbf{x} - \mathbf{x}_t, \boldsymbol{\theta}^* \rangle]$. In the batched setting, the algorithm pulls a batch of b_ℓ arms at round ℓ. For T total arms played, we can equivalently rewrite the regret as $R_T = \mathbb{E}[\max_{\mathbf{x}\in\mathcal{X}} \sum_{\ell=1}^{B} \sum_{j=1}^{b_\ell} \langle \mathbf{x} - \mathbf{x}_{\ell,j}, \boldsymbol{\theta}^* \rangle]$. where b_ℓ is the batch size of batch ℓ, B is the total number of batches, and $\sum_{\ell=1}^{B} b_\ell = T$. Note that the two definitions are exactly the same by re-indexing the arms pulled in each batch.

We assume the best arm $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle$ is unique. We define $\Delta_{\mathbf{x}} = \langle \mathbf{x}^* - \mathbf{x}, \boldsymbol{\theta}^* \rangle$ to be the suboptimality gap for an arm $\mathbf{x} \neq \mathbf{x}^*$, and $\Delta_{\min} = \min_{\mathbf{x}\neq\mathbf{x}^*} \Delta_{\mathbf{x}}$. We also define $\mu_{\mathbf{x}} = \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle$ to be the mean reward for pulling arm $\mathbf{x}$. For a K-dimensional non-negative vector $\boldsymbol{\alpha} \in \mathbb{R}^K_{\geq 0}$, we use $\boldsymbol{\alpha} = \{\alpha_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ to denote an allocation of arm pulls over arm set $\mathcal{X}$.

Asymptotic lower bound Asymptotic lower bounds are usually defined for consistent policies (Lattimore & Szepesvári, 2020). A policy π is called *consistent* (Lattimore & Szepesvari, 2017) if its regret is sub-polynomial for any bandit instance, i.e., $R_T = o(T^p)$ for all $\boldsymbol{\theta}^*$ and $p > 0$.

For a linear bandit instance with arm set $\mathcal{X}$ and weight $\boldsymbol{\theta}^*$, define $\mathbf{H}_\alpha = \sum_{\mathbf{x}\in\mathcal{X}} \alpha_{\mathbf{x}} \mathbf{x}\mathbf{x}^\top$, where $\boldsymbol{\alpha} \in \mathbb{R}^K_{\geq 0}$ is an allocation for the number of pulls of each arm.[1] $c^*(\boldsymbol{\theta}^*)$ is the solution to the following convex program,

$$\begin{aligned} c^*(\boldsymbol{\theta}^*) &\overset{\Delta}{=} \inf_{\alpha\in\mathbb{R}^K_{\geq 0}} \sum_{\mathbf{x}\in\mathcal{X}} \alpha_{\mathbf{x}} \Delta_{\mathbf{x}} \\ &\quad \text{s.t.} \quad \|\mathbf{x} - \mathbf{x}^*\|^2_{\mathbf{H}_\alpha^{-1}} \leq \frac{\Delta_{\mathbf{x}}^2}{2}, \forall \mathbf{x} \in \mathcal{X}^-, \end{aligned} \tag{3.1}$$

where $\mathcal{X}^- = \mathcal{X} - \{\mathbf{x}^*\}$. We write $c^* := c^*(\boldsymbol{\theta}^*)$ when no ambiguity arises.

Lemma 3.1 (Lattimore & Szepesvari (2017))**.** *Any consistent algorithm π for the linear bandits setting with Gaussian noise has regret R_T satisfying $\liminf_{T\to\infty} R_T/\log T \geq c^*$, where c^* is given by* (3.1)*.*

We call an algorithm asymptotically optimal if it achieves $\lim_{T\to\infty} R_T/\log T = c^*$.

Least square estimators For a dataset $\{(\mathbf{x}_s, r_s)\}_{s=1}^t$, we define the following least squares estimators which will be used in our algorithm design. Let $\mathbf{H} = \sum_{s=1}^t \mathbf{x}_s \mathbf{x}_s^\top$, and

$$\hat{\boldsymbol{\theta}} = \mathbf{H}^{-1} \textstyle\sum_{s=1}^t \mathbf{x}_s r_s, \tag{3.2a}$$

$$\hat{\mu}_{\mathbf{x}}(t) = \langle \hat{\boldsymbol{\theta}}, \mathbf{x} \rangle, \tag{3.2b}$$

$$\hat{\mathbf{x}}^* = \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \langle \hat{\boldsymbol{\theta}}, \mathbf{x} \rangle, \tag{3.2c}$$

$$\hat{\Delta}_{\mathbf{x}} = \langle \hat{\boldsymbol{\theta}}, \hat{\mathbf{x}}^* - \mathbf{x} \rangle. \tag{3.2d}$$

4. Optimal Batched Linear Bandits Algorithm

Algorithm 1 Explore, Estimate, Eliminate, and Exploit (E^4)

```
Input: arm set $\mathcal{X}$, horizon $T$, $\alpha > 0$, $\gamma > 0$, D-optimal design allocation function $f(\cdot,\cdot,\cdot)$ defined in Definition 4.1, exploration rate $\{T_1, T_2, \ldots\}$, elimination parameters $\{\varepsilon_1, \varepsilon_2, \ldots\}$.
Initialization: set $\mathcal{A} \leftarrow \mathcal{X}$ and $t \leftarrow 0$.
Batch $\ell = 1$
  Exploration: Pull arm $\mathbf{x} \in \mathcal{A}$ for $f(\mathbf{x}, \mathcal{A}, T_1)$ times.
  $t \leftarrow t + \sum_{\mathbf{x}\in\mathcal{A}} f(\mathbf{x}, \mathcal{A}, T_1)$.
  Estimation: Update least squares estimators $\hat{\boldsymbol{\theta}}, \hat{\mathbf{x}}^*, \hat{\Delta}$ based on the current batch of data by (3.2).
  Calculate $\mathbf{w}(\hat{\Delta}) = \{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{A}}$ by Definition 4.3.
Batch $\ell = 2$
  Exploration: Pull arm $\mathbf{x} \in \mathcal{A}$ for $f(\mathbf{x}, \mathcal{A}, T_2) + n_{\mathbf{x}}$ times, where $n_{\mathbf{x}} = \min\{w_{\mathbf{x}} \cdot \alpha \log T, (\log T)^{1+\gamma}\}$.
  $b_2 \leftarrow \sum_{\mathbf{x}\in\mathcal{A}} f(\mathbf{x}, \mathcal{A}, T_2) + n_{\mathbf{x}}$, $t \leftarrow t + b_2$.
  Estimation: Update least squares estimators $\hat{\boldsymbol{\theta}}, \hat{\mathbf{x}}^*, \hat{\Delta}$ based on the current batch of data by (3.2).
  Elimination:
    if stopping rule (4.2) holds
      $\mathcal{A} \leftarrow \{\hat{\mathbf{x}}^*\}$.
    end if
Batch $\ell \geq 3$
  $\ell \leftarrow 3$.
  while $t < T$ and $|\mathcal{A}| > 1$ do
    Exploration: Pull arm $\mathbf{x} \in \mathcal{A}$ for $f(\mathbf{x}, \mathcal{A}, T_\ell)$ times.
    Estimation: Update least squares estimator $\hat{\boldsymbol{\theta}}$ based on the current batch of data by (3.2).
    Elimination: $\mathcal{A} \leftarrow \{\mathbf{x} \in \mathcal{A} : \max_{\mathbf{y}\in\mathcal{A}} \langle \hat{\boldsymbol{\theta}}, \mathbf{y} - \mathbf{x} \rangle \leq 2\varepsilon_\ell\}$.
    $\ell \leftarrow \ell + 1$, $t \leftarrow t + \sum_{\mathbf{x}\in\mathcal{A}} f(\mathbf{x}, \mathcal{A}, T_\ell)$.
  end while
  Exploitation: Pull arm $\mathbf{x} \in \mathcal{A}$ for $T - t$ times .
```

In this paper, we design a general algorithm framework, called **Explore-Estimate-Eliminate-Exploit** (E^4), which is presented in Algorithm 1. The algorithm proceeds in a batched fashion.

In each batch we use an *Exploration* stage that explores the arm space in some way we would specify below, an *Estima-*

[1] For the ease of presentation, we may overload the notation of covariance matrix $\mathbf{H}$ a little bit in this paper. Specifically, for an integer $t \in \mathbb{N}_+$, we define $\mathbf{H}_t = \sum_{s=1}^t \mathbf{x}_s \mathbf{x}_s^\top$. For a vector $\mathbf{w} = \{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}} \in \mathbb{R}^K_{\geq 0}$, we define $\mathbf{H}_{\mathbf{w}} = \sum_{\mathbf{x}\in\mathcal{X}} w_{\mathbf{x}} \cdot \mathbf{x}\mathbf{x}^\top$.

tion stage that calculate some statistics with least squares estimators, and an *Elimination* stage that eliminates low-rewarding arms. Note that in a specific Estimation stage, we exclusively use data from that particular batch for the computation of estimators and statistics. This ensures the independence, aligning with our theoretical objectives. In Algorithm 1, we maintain an active set, denoted by $\mathcal{A} \subseteq \mathcal{X}$, which only consists of arms that we are unsure about its sub-optimality and need to further explore. If after some batch, there is only one arm in the active arm set $\mathcal{A}$, the algorithm enters the *Exploitation* stage. This means it identifies the best arm and would pull it until the total pulls reach T.

Let ℓ be the index of batches. We denote b_ℓ as the total number of arms pulled in the ℓ-th batch, which is also called as the batch size. And we denote T_ℓ as the D-optimal design exploration rate in the ℓ-th batch. In what follows, we elaborate the details in Algorithm 1 for different batches.

Batch $\ell = 1$: *Explore with D-optimal design and estimate the exploration allocation* $\mathbf{w}$ *for the next batch.*

Exploration: In the first batch, the agent has no prior information on arms and thus explores all arms through the classical D-optimal design (Lattimore & Szepesvári, 2020). Specifically, given an arm set $\mathcal{A}$ and and a user defined exploration rate M, we use D-optimal design rule to decide the number of pulls of each arm $\mathbf{x}$, denoted by $f(\mathbf{x}, \mathcal{A}, M)$, which is defined as follows.

Definition 4.1. For an arm set $\mathcal{A}$ and an exploration rate M, the D-optimal design allocation of arm $\mathbf{x} \in \mathcal{A}$ is given by $f(\mathbf{x}, \mathcal{A}, M) = \lceil 2\pi^*_{\mathbf{x}} g(\boldsymbol{\pi}^*)M/d \rceil$, where $\boldsymbol{\pi}^* := \boldsymbol{\pi}^*(\mathcal{A})$ is a probability measure over the action set defined by

$$\boldsymbol{\pi}^*(\mathcal{A}) = \mathrm{argmin}_{\boldsymbol{\pi}} \max_{\mathbf{x} \in \mathcal{A}} \|\mathbf{x}\|^2_{\mathbf{H}_{\boldsymbol{\pi}}^{-1}}, \tag{4.1}$$

and $g(\boldsymbol{\pi}) = \max_{\mathbf{x} \in \mathcal{A}} \|\mathbf{x}\|^2_{\mathbf{H}_{\boldsymbol{\pi}}^{-1}}$.

Throughout our algorithm, we employ this D-optimal design exploration rule. Specifically, we set the exploration rate $M = T_\ell$ for the ℓ-th batch.

Remark 4.2. We present Definition 4.1 here just for simplicity. In both our theoretical analysis and experiments, we use the Frank–Wolfe algorithm (see Chapter 21.2 in Lattimore & Szepesvári (2020)) to approximately solve the optimization problem in (4.1) and get a near-optimal solution $\tilde{\boldsymbol{\pi}}$. See Appendix C for more details.

Estimation: Then Algorithm 1 calculates least squares estimators by (3.2). Based on these estimators, we solve an allocation $\mathbf{w} = \{w_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{X}}$ using the following definition, which will be used in the *Exploration* stage of batch 2.

Definition 4.3. Let $\epsilon = 1/\log\log T$, and $\mathbf{w} = \{w_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{X}} \in [0, \infty)^K$ be the solution to the problem

$$\min_{\mathbf{w} \in [0,\infty)^K} \sum_{\mathbf{x} \neq \hat{\mathbf{x}}^*} w_{\mathbf{x}} (\hat{\Delta}_{\mathbf{x}} - 4\epsilon)$$

$$\text{s.t.} \quad \|\mathbf{x} - \hat{\mathbf{x}}^*\|^2_{\mathbf{H}_{\mathbf{w}}^{-1}} \leq (\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2/2, \quad \forall \mathbf{x} \neq \hat{\mathbf{x}}^*,$$
$$\text{and} \quad w_{\hat{\mathbf{x}}^*} \leq (\log T)^\gamma/\alpha,$$

where $\hat{\Delta}$ and $\hat{\mathbf{x}}^*$ are defined in (3.2), $\mathbf{H}_{\mathbf{w}}$ is defined in Section 3, $\gamma \in (0, 1)$ and $\alpha = (1 + 1/\log\log T)(1 + d\log\log T/\log T)$.

Remark 4.4. Our Definition 4.3 is novel and different from literature. Compared with the naive allocation solved from an optimization problem like (3.1), we use $\hat{\Delta}_{\mathbf{x}} - 4\epsilon$ instead of $\hat{\Delta}_{\mathbf{x}}$, and add a constraint $w_{\hat{\mathbf{x}}^*} \leq (\log T)^\gamma/\alpha$ in the program. From Lemma C.6 we can see the solution from Definition 4.3 convergences to the solution of naive method as $T \to \infty$, so our modification tricks in Definition 4.3 doesn't influence the asymptotic regret analysis. In fact, with high probability, estimated gaps based on data in the second batch of our algorithm are larger than those underestimated gaps $\hat{\Delta}_{\mathbf{x}} - 4\epsilon$. We elaborate this trick in our proof of Lemma C.7, which shows that with high probability, the stopping rule (4.2) holds after the second batch.

Batch $\ell = 2$: *Explore with D-optimal design and the allocation* $\mathbf{w}$ *from Definition 4.3, and eliminate all suboptimal arms based on Chernoff's stopping rule.*

Exploration: The exploration stage of batch 2 requires playing arms according to two allocations: (1) D-optimal design exploration specified by Definition 4.1, and (2) optimal allocation specified by Definition 4.3. More specifically, for the second part, we pull $\mathbf{x} \in \mathcal{A}$ for $\min\{w_{\mathbf{x}} \cdot \alpha \log T, (\log T)^{1+\gamma}\}$ times, where $\gamma \in (0, 1)$ is an arbitrarily small constant and $\mathbf{w} = \{w_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{X}}$ is solved from Definition 4.3 based on the estimators from batch 1.

Estimation: We calculate the least squares estimators according to (3.2) based on the current batch of data.

Elimination: We then test the Chernoff's stopping rule and only keeps the current best arm based on our *Estimation* stage if it holds. This means we will eliminate all suboptimal arms and directly go to the Line 1, *Exploitation* stage of Algorithm 1. In particular, the stopping rule is defined as

$$\big\{Z(b_2) \geq \beta(b_2, 1/T) \text{ and } \textstyle\sum_{s=1}^{b_2} \mathbf{x}_s \mathbf{x}_s^\top \geq c\mathbf{I}_d\big\}, \tag{4.2}$$

where b_2 is the batch size of the second batch defined in Line 1 of Algorithm 1, $c = \max_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_2^2$ and

$$Z(b_2) = \min_{\mathbf{x} \neq \hat{\mathbf{x}}^*} \hat{\Delta}^2_{\mathbf{x}} / (2\|\mathbf{x} - \hat{\mathbf{x}}^*\|^2_{\mathbf{H}_{b_2}^{-1}}), \tag{4.3}$$

$$\beta(t, \delta) = (1 + 1/\log\log T) \log\big((t \log\log T)^{\frac{d}{2}}/\delta\big). \tag{4.4}$$

If the stop rule (4.2) does not hold, Algorithm 1 will enter the **while** loop and conduct more batches of exploration.

The use of Chernoff's stopping rule is inspired from best arm identification problems (Garivier & Kaufmann, 2016;

Jedra & Proutiere, 2020; Jin et al., 2023). Specifically, if (4.2) holds, we can prove that $\hat{\mathbf{x}}^*$ is the true best arm with probability at least $1 - 1/T$. This further ensures that no more exploration is needed and the regret is small.

Batch $\ell \geq 3$: *Perform phased elimination with D-optimal design exploration.* When Algorithm 1 does enter these batches, it means the previous two batches are not sufficient to identify the best arm and thus more exploration is needed. To this end, we adopt the phased elimination with the D-optimal design exploration (Lattimore & Szepesvári, 2020).

Exploration: We use D-optimal design exploration defined in Definition 4.1 with rate T_ℓ for the ℓ-th batch.

Estimation: We calculate the least squares estimators according to (3.2) based on the current batch of data.

Elimination: We eliminate arms whose estimated mean reward on the estimated parameters are smaller than that of the empirically best arm by a margin of $2\varepsilon_\ell$. The active arm set is updated as $\mathcal{A} \leftarrow \{\mathbf{x} \in \mathcal{A} : \max_{\mathbf{y}\in\mathcal{A}} \langle \hat{\boldsymbol{\theta}}, \mathbf{y} - \mathbf{x}\rangle \leq 2\varepsilon_\ell\}$.

Final batch: *Pull the estimated best arm $\hat{\mathbf{x}}^*$ until the end.* If the Chernoff's stopping rule (4.2) holds at the end of the second batch, or $|\mathcal{A}| = 1$ after some batch $\ell \geq 3$, Algorithm 1 will enter Line 1, the *Exploitation* stage. In this stage, the agent just commits to the estimated best arm $\hat{\mathbf{x}}^* \in \mathcal{A}$ until total pulls reach $t = T$.

5. Theoretical Analysis

In this section, we provide theoretical analysis results on the regret optimality and batch complexity of Algorithm 1.

First we give a formal definition of *batch*.

Definition 5.1. For a linear bandit problem with time horizon T, we say that the batch complexity of a learning algorithm is (at most) M, if the learner decides T_1 and π_1 before the learning process starts, and executes π_1 in the first T_1 time steps (which corresponds the first batch). Based on the data (context sets, played actions and the rewards) obtained from the first T_1 steps, the learner then decides T_2 and π_2, and executes π_2 for T_2 time steps (the second batch). The learner repeats the process for M times/batches. In general, at the beginning of the k-th batch, the learner decides T_k (the size of the batch) and π_k based on the data collected from the first $(k-1)$ batches. The batch sizes should satisfy that $\Sigma_{k=1}^{M} T_k = T$.

We present the following result on the batch complexity lower bound of asymptotically optimal algorithms.

Theorem 5.2. *If an algorithm achieves asymptotic optimality defined in Lemma 3.1, then on some bandit instances it must have at least 3 batches in expectation as $T \to \infty$.*

Remark 5.3. In our proof detailed in Appendix B, we first show that any algorithm that performs at most 2 batches is not asymptotically optimal. Then we further show that even with randomly chosen batch sizes, the expected number of batches of an asymptotically optimal algorithm is at least 3. This is the first batch complexity lower bound in the literature for asymptotically optimal algorithms. Though the lower bound in Theorem 5.2 seems to be conservative, we will show in the next theorem that our proposed algorithm E^4 achieves asymptotic optimality in regret and its this batch matches the lower bound in Theorem 5.2.

Next we present the upper bounds on the regret and the batch complexity of Algorithm 1.

Theorem 5.4. *In Algorithm 1, let the parameters be set to $\alpha = (1 + 1/\log\log T)(1 + d\log\log T/\log T)$, $\delta = 1/(KT^2)$, $\gamma \in (0, 1)$, and $\varepsilon_\ell = \sqrt{d\log(1/\delta)/T_\ell}$, $\forall \ell \geq 1$. Let the exploration rate be defined as*

$$\mathcal{T}_1 = \{T_1 = (\log T)^{1/2}, T_2 = (\log T)^{1/2}, T_3 = (\log T)^{1+\gamma}, T_\ell = T^{1-\frac{1}{2^{\ell-3}}} \textit{ for } \ell \geq 4\}.$$

Then the regret of Algorithm 1 satisfies

$$R_T = O\big(\log\log T \cdot \sqrt{dT\log(KT)}\big) = \tilde{O}\big(\sqrt{dT}\big),$$

and only needs at most $O(\log\log T)$ batches. As $T \to \infty$,

$$\limsup_{T\to\infty} \frac{R_T}{\log T} \leq c^*,$$

only runs with 3 batches in expectation. Furthermore, Algorithm 1 runs with 3 batches with probability at least $1 - 2/(\log T)^2$ as $T \to \infty$.

Remark 5.5. Our $\tilde{O}(\sqrt{dT})$ regret bound nearly matches the lower bound in Lattimore & Szepesvári (2020), and the $O(\log\log T)$ batch complexity matches the lower bound of $\Omega(\log\log T)$ in (Gao et al., 2019). Furthermore, according to Theorem 5.2, our algorithm also achieves the asymptotically optimal regret bound with the optimal batch complexity, i.e., E^4 only need 3 batches to achieve the asymptotic optimality. To the best of our knowledge, E^4 is the only algorithm in the literature that is simultaneously optimal in terms of regret bound and batch complexity in both the finite-time worst-case setting and the asymptotic setting.

In contrast, existing linear bandit algorithms can be categorized as follows: (1) achieving the minimax optimal regret bound but with $O(\log T)$ batch complexity (Esfandiari et al., 2021); (2) achieving the minimax optimal regret bound with optimal batch complexity but no asymptotic guarantees (Ruan et al., 2021; Hanna et al., 2023); (3) achieving the asymptotically optimal regret but no finite time minimax optimality (Lattimore & Szepesvari, 2017; Combes et al., 2017; Hao et al., 2020; Wagenmaker & Foster, 2023); or (4) achieving the asymptotically optimal regret and the minimax optimality but in a fully sequential way (Tirinzoni et al., 2020). Compared to these works, Frequentist Information-Directed Sampling (IDS) (Kirschner et al., 2021) achieves both minimax and asymptotic optimality. Furthermore, it can be proved to have a batch complexity slightly larger than $O(\log^4 T)$[2], which is still much higher than the batch complexity of our algorithm E^4. Moreover, the IDS algorithm can only achieve a worst case regret bound $\tilde{O}(d\sqrt{T})$, which is suboptimal for K-armed linear bandits when K is fixed.

Remark 5.6. Lastly, we would like to highlight that when T is large enough, the stopping rule (4.2) used in Line 1 of Algorithm 1 holds with probability at least $1 - 2/(\log T)^2$. When this happens, our algorithm would skip the **while** loop and directly goes to the final exploitation batch. Therefore, the algorithm will only need 3 batches to achieve the asymptotic optimality. Notably, since the stopping rule in Line 1 only depends on the first two batches of Algorithm 1, different choices of exploration rates in later do not change the asymptotic behavior of Algorithm 1 when $T \to \infty$, as we will see in the next theorem.

In the following theorem, we show that with a slightly different choice of exploration rates, Algorithm 1 can achieve an instance-dependent regret bound with the optimal batch complexity as well.

Theorem 5.7. *Let the exploration rate in Algorithm 1 be chosen as*

$$\mathcal{T}_2 = \{T_1 = (\log T)^{1/2}, T_2 = (\log T)^{1/2}, T_3 = (\log T)^{1+\gamma}, \\ T_\ell = d\log(KT^2)\cdot 2^{\ell-3} \text{ for } \ell \geq 4\},$$

and all other parameters remain the same as in Theorem 5.4. Then Algorithm 1 satisfies $R_T = O(\sqrt{dT}\cdot\log(KT)) = \tilde{O}(\sqrt{dT})$, and

$$R_T = O((\log T)^{1+\gamma} + d\log(KT)/\Delta_{\min})$$

with at most $O(\log T)$ batches and in expectation $O(\log(1/\Delta_{\min}))$ batches. Furthermore, the regret of Algorithm 1 satisfies $\limsup_{T\to\infty} R_T/\log T \leq c^*$ with 3 batches in expectation.

Remark 5.8. Theorem 5.7 shows that E^4 can achieve the minimax optimal regret and the instance-dependent regret bound $O(d\log T/\Delta_{\min})$ with at most $O(\log T)$ batches. Gao et al. (2019) proved that any algorithm with $O(d\log T/\Delta_{\min})$ regret have at least $\Omega(\log T/\log\log T)$ batches. Hence, in this context, E^4 attains optimal batch complexity while achieving this instance-dependent regret bound. Notably, the instance-dependent bound gives a tighter regret bound within the class of bandit instances where $\Delta_{\min} \geq \sqrt{d/T}$ than the worst-case regret bound.

Remark 5.9. Notably, our instance-dependent batch complexity $O\big(\log(1/\Delta_{\min})\big)$ is novel and doesn't depend on time horizon T, which shows that for instances with large reward gaps the batch complexity of our Algorithm 1 is a small constant and does not increase as the horizon T increases.

6. Experiments

In this section, we conduct experiments on challenging linear bandit instances, assessing the performance of our E^4 algorithm in comparison to several baseline algorithms. In particular, we first do simulations on hard linear bandit instances we constructed inspired from the End of Optimism instance (Lattimore & Szepesvari, 2017) on which optimistic algorithms such as UCB and TS are proved to be suboptimal as $T \to \infty$. We then conduct experiments on more general random instances. The implementation could be found at https://github.com/panxulab/optimal-batched-linear-bandits.

Baselines: We compare E^4(Algorithm 1) with Rarely Switching OFUL (denoted by rs-OFUL) (Abbasi-Yadkori et al., 2011), the Optimal Algorithm (denoted by EndOA) in Lattimore & Szepesvari (2017), IDS (Kirschner et al., 2021), and Phased Elimination Algorithm with D-optimal design (denoted by PhaElimD) (Esfandiari et al., 2021; Lattimore & Szepesvári, 2020). Note that IDS was shown to outperform other asymptotically optimal algorithms (Lattimore & Szepesvari, 2017; Combes et al., 2017; Hao et al., 2020; Tirinzoni et al., 2020), and thus we do not include the rest of them in our experiments.

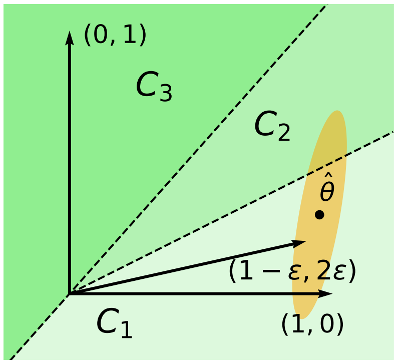
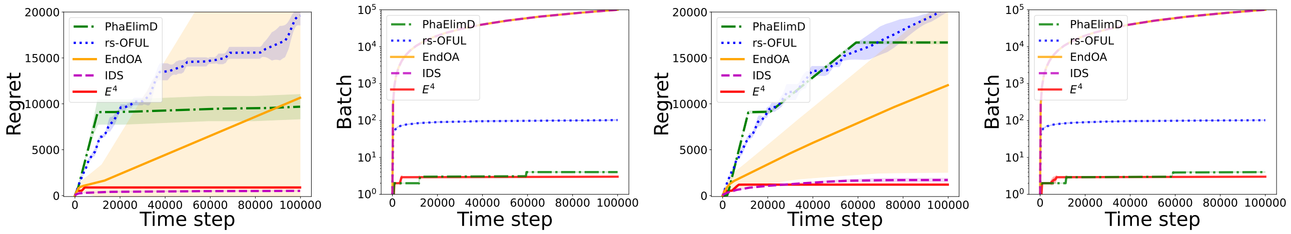


Figure 1: The *End of Optimism* instance in $\mathbb{R}^2$. The true parameter $\boldsymbol{\theta}^*$ is $(1, 0)$. The arms are $\mathbf{x}_1 = (1, 0), \mathbf{x}_2 = (0, 1), \mathbf{x}_3 = (1-\varepsilon, 2\varepsilon)$. Note that $\mathbf{x}_i$ is the best arm if $\boldsymbol{\theta}^*$ lies in the colored region C_i, $i = 1, 2, 3$.

***End of Optimism* Instances:** Lattimore & Szepesvari

[2] IDS is not a batched algorithm, but it has a rarely-switching structure and thus can be easily adapted into an batched algorithm. By some calculations, their batch complexity is larger than $O(d^4\log^4 T/\Delta^2_{\min})$, which we also include in Table 1.



(a) $d = 5, K = 9, \epsilon = 0.01$ (b) $d = 5, K = 9, \epsilon = 0.01$ (c) $d = 5, K = 9, \epsilon = 0.2$ (d) $d = 5, K = 9, \epsilon = 0.2$

Figure 2: Regret and Batch Analysis: *End of Optimism* instances ($d = 5, K = 9$).

(2017) designed a hard instance in $\mathbb{R}^2$ such that optimism based algorithms cannot achieve the asymptotic optimality. For completeness, we present the original *End of Optimism* instance in $\mathbb{R}^2$ in Figure 1. Following a similar idea, we design several hard instances in more complicated settings to evaluate our proposed algorithm. Let $\mathbf{e}_j^d$ denote the natural basis vector in $\mathbb{R}^d$ where the j-th variable is 1 and all others are 0. Then we construct the following hard instances inspired by the End of Optimism (Lattimore & Szepesvari, 2017). Let $\boldsymbol{\theta}^* = \mathbf{e}_1^d$, the arm set $\mathcal{X}$ defined as follows

$$\mathcal{X} = \{\mathbf{e}_i^d\}_{i=1}^d \cup \{(1-\epsilon)\mathbf{e}_1^d + 2\epsilon\mathbf{e}_j^d\}_{j=2}^d, \tag{6.1}$$

where $\epsilon = 0.01, 0.2$ and the dimension $d = 2, 3, 5$. Hence there are 6 different hard instances in total. For each instance, the number of arms is $K = 2d - 1$. We conduct experiments on these instances and present the result for $d = 5$ in Figure 3. Due to space limit, we defer the results for $d = 2$ and $d = 3$ to Appendix A.1.

Implementation: Based on the weight parameter $\boldsymbol{\theta}^*$ and arm set $\mathcal{X}$ defined in (6.1), we generate the noise reward for arm $\mathbf{x} \in \mathcal{X}$ as $r(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, 1)$. For the parameters in E^4 (Algorithm 1), we fix the exploration rate to be $\mathcal{T}_1$. We set the other parameters as described in Theorem 5.4. For rs-OFUL, we implement the rarely switching OFUL algorithm in Abbasi-Yadkori et al. (2011) with switching parameter $C = 0.5$. For EndOA, we follow the Warmup phase and Success phase exactly the same way as Lattimore & Szepesvari (2017) describes, and use OFUL in its Recovery phase. For IDS, we follow their computationally efficient variant (Kirschner et al., 2021, Section 2) step by step. For PhaElimD, we follow the framework in Esfandiari et al. (2021), and improve the batch sizes design by using exploration rate $T_i = T^{1-1/2^i}$ instead of q^i in their paper for better performance. For each instance, we repeat the experiment for each method for 10 simulations and calculate the mean value and standard error of the cumulative regret and the batch complexity.

Notably, we proposed two options for the exploration rates in Theorems 5.4 and 5.7. However, as we discussed in Remark 5.6, our algorithm skips the **while** loop with extremely high probability. Consequently, different choices of exploration rates would not affect the performance of the algorithm in practice. Therefore, we fix the exploration rate to be $\mathcal{T}_1$ defined in Section 5 for the sake of convenience.

Results: We observe that E^4 (Algorithm 1) achieves a similar regret as IDS and outperforms all other baseline algorithms, which demonstrates the effectiveness of our proposed algorithm. Moreover, E^4 only needs 3 batches to achieve this optimal regret, which is in sharp contrast with IDS which needs more than 10^5 batches to achieve the same regret. The only algorithm that enjoys a similar batch complexity as E^4 is PhaElimD, which only runs in around 4 batches. Nevertheless, the regret of PhaElimD is significantly higher than our E^4 algorithm. As ϵ goes from 0.01 to 0.2, the regret of our algorithm and IDS remains optimal and very small in magnitude, while the performance of other algorithms becomes worse.

Table 2: Runtime (seconds) comparison.

ϵ	ALGORITHM	$d = 2$	$d = 3$	$d = 5$
0.01	PHAELIMD	0.18	0.71	1.46
	RS-OFUL	0.45	1.47	3.72
	ENDOA	3.15	3.17	8.94
	IDS	9.48	30.22	178.31
	E^4	0.04	0.12	0.25
0.2	PHAELIMD	0.15	0.76	1.40
	RS-OFUL	0.28	1.60	2.90
	ENDOA	2.23	3.87	10.19
	IDS	6.42	31.24	246.53
	E^4	0.06	0.15	0.33

Runtime comparison: We also present the averaged runtime for all the competing algorithms across 10 independent trials, as displayed in Table 2. In all instances regardless of the dimension or ϵ, our algorithm E^4 is consistently the most computationally efficient one, offering **a speedup ranging from** 5**-fold to** 1000**-fold** when compared to other baselines.

Due to the space limit, we defer more experimental results to Appendix A. In particular, we conducted an ablation study on the sensitivity of different algorithms' performance on the instance parameter ϵ in Appendix A.2. We also conducted extra experiments on some randomly generated instances and present the results in Appendix A.3.

7. Conclusion and Future Work

In this paper, we proposed the Explore, Estimate, Eliminate and Exploit (E^4) algorithm, which only needs 3 batches to achieve the asymptotically optimal regret in linear bandits. To the best of our knowledge, E^4 is the first batched linear bandit algorithm that is simultaneously optimal in terms of regret bound and batch complexity in both the finite-time worst-case setting and the asymptotic setting. We conducted numerical experiments on challenging linear bandit instances, which unequivocally show that our method outperforms the current baseline methods in multiple critical aspects: it achieves the lowest level of regret, it requires the minimal batch complexity, and it enjoys superior running time efficiency.

For future research, an intriguing objective is to investigate whether we can attain the optimal instance-dependent regret while maintaining the optimal batch complexity. While recent work by Wagenmaker & Foster (2023) achieves a specially defined instance-dependent optimality, their algorithms fail to achieve the minimax optimality, and more importantly, are fully sequential and thus cannot be applied to the batch setting. We hope our strategy proposed in this paper will provide a viable solution towards building an optimal instance-dependent algorithm with optimal batch complexity, which we leave for future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research is supported by the Whitehead Scholars Program, by the US National Science Foundation Award 2323112, by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG-PhD/2021-01004[T]), and by the Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2 under grant number A-8000423-00-00. In particular, P. Xu was supported in part by the National Science Foundation (DMS-2323112) and the Whitehead Scholars Program at the Duke University School of Medicine. T. Jin was supported by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG-PhD/2021-01004[T]), and by the Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2 under grant number A-8000423-00-00. The views and conclusions in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Abe, N., Biermann, A. W., and Long, P. M. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37:263–293, 2003.

Agarwal, D., Chen, B.-C., Elango, P., Motgi, N., Park, S.-T., Ramakrishnan, R., Roy, S., and Zachariah, J. Online models for content optimization. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pp. 17–24, 2008.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pp. 127–135. PMLR, 2013.

Bertsimas, D. and Mersereau, A. J. A learning approach for interactive marketing to a customer segment. *Operations Research*, 55(6):1120–1135, 2007.

Bubeck, S., Cesa-Bianchi, N., et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5 (1):1–122, 2012.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214. JMLR Workshop and Conference Proceedings, 2011.

Combes, R., Magureanu, S., and Proutiere, A. Minimal exploration in structured stochastic bandits. *Advances in Neural Information Processing Systems*, 30, 2017.

Esfandiari, H., Karbasi, A., Mehrabian, A., and Mirrokni, V. Regret bounds for batched bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7340–7348, 2021.

Gao, Z., Han, Y., Ren, Z., and Zhou, Z. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.

Garivier, A. and Kaufmann, E. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pp. 998–1027. PMLR, 2016.

Graves, T. L. and Lai, T. L. Asymptotically efficient adaptive choice of control laws incontrolled markov chains. *SIAM Journal on Control and Optimization*, 35(3):715–743, 1997.

Han, Y., Zhou, Z., Zhou, Z., Blanchet, J., Glynn, P. W., and Ye, Y. Sequential batch learning in finite-action linear contextual bandits. *arXiv preprint arXiv:2004.06321*, 2020.

Hanna, O. A., Yang, L., and Fragouli, C. Contexts can be cheap: Solving stochastic contextual bandits with linear bandit algorithms. In Neu, G. and Rosasco, L. (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 1791–1821. PMLR, 12–15 Jul 2023.

Hao, B., Lattimore, T., and Szepesvari, C. Adaptive exploration in linear contextual bandit. In *International Conference on Artificial Intelligence and Statistics*, pp. 3536–3545. PMLR, 2020.

Jedra, Y. and Proutiere, A. Optimal best-arm identification in linear bandits. *Advances in Neural Information Processing Systems*, 33:10007–10017, 2020.

Jin, T., Tang, J., Xu, P., Huang, K., Xiao, X., and Gu, Q. Almost optimal anytime algorithm for batched multi-armed bandits. In *International Conference on Machine Learning*, pp. 5065–5073. PMLR, 2021a.

Jin, T., Xu, P., Xiao, X., and Gu, Q. Double explore-then-commit: Asymptotic optimality and beyond. In *Conference on Learning Theory*, pp. 2584–2633. PMLR, 2021b.

Jin, T., Yang, Y., Tang, J., Xiao, X., and Xu, P. Optimal batched best arm identification. *arXiv preprint arXiv:2310.14129*, 2023.

Kirschner, J., Lattimore, T., Vernade, C., and Szepesvári, C. Asymptotically optimal information-directed sampling. In *Conference on Learning Theory*, pp. 2777–2821. PMLR, 2021.

Langford, J. and Zhang, T. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems*, 20(1):96–1, 2007.

Lattimore, T. and Szepesvari, C. The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *Artificial Intelligence and Statistics*, pp. 728–737. PMLR, 2017.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Perchet, V., Rigollet, P., Chassang, S., and Snowberg, E. Batched bandit problems. In Grünwald, P., Hazan, E., and Kale, S. (eds.), *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 1456–1456, Paris, France, 03–06 Jul 2015. PMLR.

Robbins, H. Some aspects of the sequential design of experiments. 1952.

Ruan, Y., Yang, J., and Zhou, Y. Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 74–87, 2021.

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

Tirinzoni, A., Pirotta, M., Restelli, M., and Lazaric, A. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. *Advances in Neural Information Processing Systems*, 33:1417–1427, 2020.

Wagenmaker, A. J. and Foster, D. J. Instance-optimality in interactive decision making: Toward a non-asymptotic theory. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 1322–1472. PMLR, 2023.

Xu, P., Zheng, H., Mazumdar, E. V., Azizzadenesheli, K., and Anandkumar, A. Langevin monte carlo for contextual bandits. In *International Conference on Machine Learning*, pp. 24830–24850. PMLR, 2022.

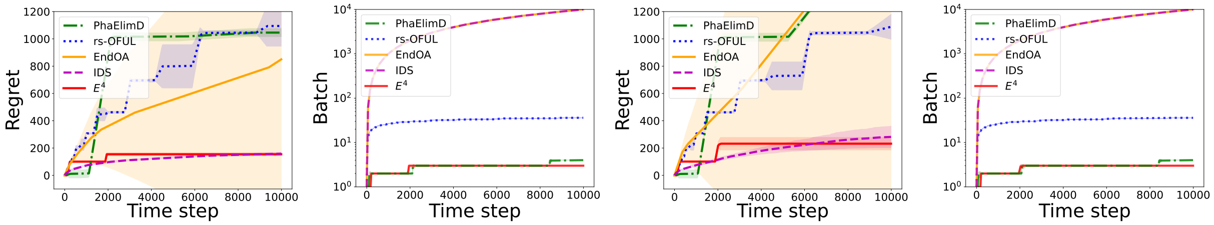
Zhang, Z., Ji, X., and Zhou, Y. Almost optimal batch-regret tradeoff for batch linear contextual bandits. *arXiv preprint arXiv:2110.08057*, 2021.

A. Additional Experiments

In this section, we provide more experimental results.

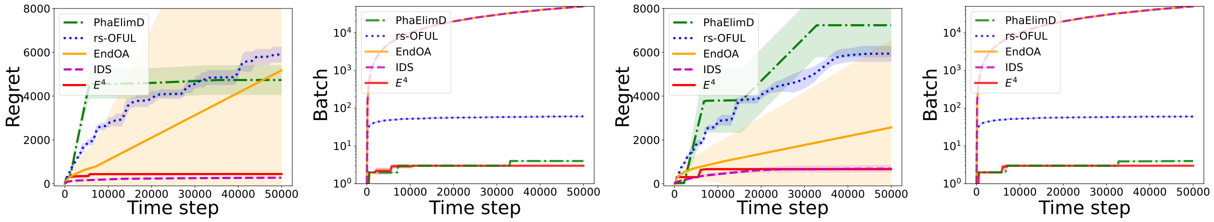
A.1. More Results on End of Optimism Instances

We provide more experimental results on instances defined in (6.1) for $d = 2$ and 3 in Figure 3 and Figure 4 respectively. We also present the detailed batch complexities of algorithms in Table 3. These experiments demonstrate that the proposed algorithm E^4 consistently outperforms baseline algorithms in terms of regret bound and batch complexity across different bandit instances.



(a) $d = 2, K = 3, \epsilon = 0.01$ (b) $d = 2, K = 3, \epsilon = 0.01$ (c) $d = 2, K = 3, \epsilon = 0.2$ (d) $d = 2, K = 3, \epsilon = 0.2$

Figure 3: Regret and Batch Analysis: *End of Optimism* instances ($d = 2, K = 3$).



(a) $d = 3, K = 5, \epsilon = 0.01$ (b) $d = 3, K = 5, \epsilon = 0.01$ (c) $d = 3, K = 5, \epsilon = 0.2$ (d) $d = 3, K = 5, \epsilon = 0.2$

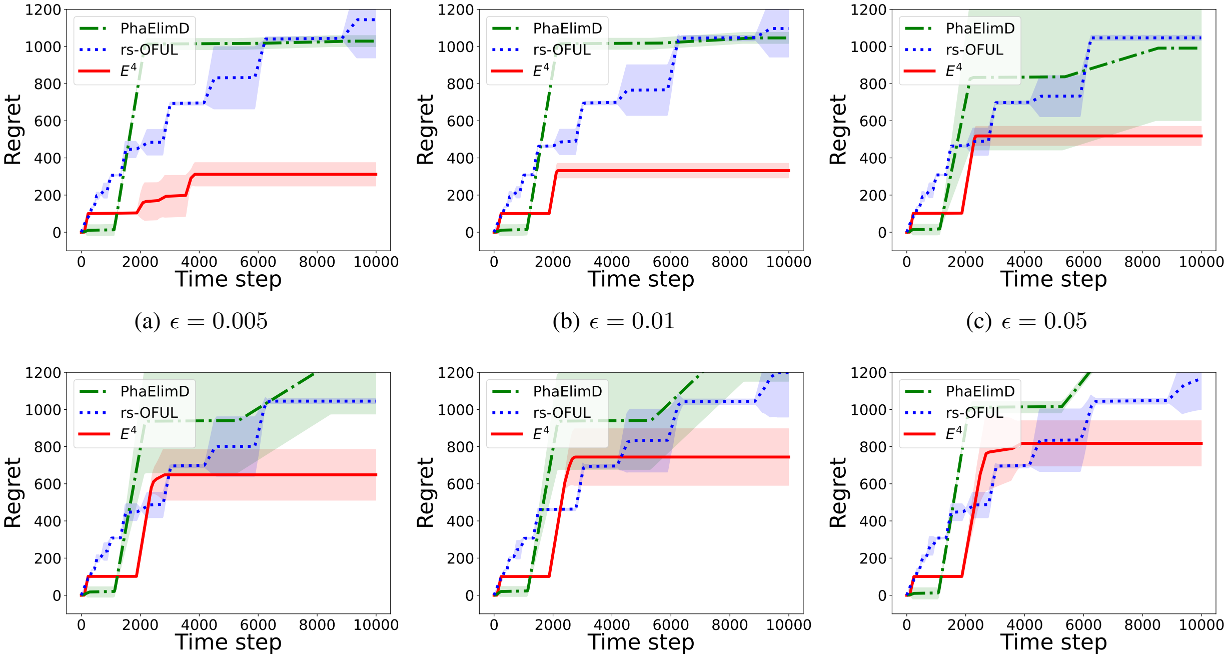
Figure 4: Regret and Batch Analysis: *End of Optimism* instances ($d = 3, K = 5$).

Table 3: Batch Complexity Analysis: *End of Optimism* instances. Note that batch complexity of sequential algorithms like EndOA and IDS equals time horizon T.

INSTANCE		E^4	PHAELIMD	RS-OFUL	ENDOA	IDS
$d = 2, K = 3, T = 10000$	$\epsilon = 0.01$	3.0 ± 0.0	4.0 ± 0.0	36.1 ± 0.3	-	-
	$\epsilon = 0.2$	3.0 ± 0.0	4.0 ± 0.0	37.0 ± 0.0	-	-
$d = 3, K = 5, T = 50000$	$\epsilon = 0.01$	3.0 ± 0.0	4.0 ± 0.0	61.0 ± 0.5	-	-
	$\epsilon = 0.2$	3.0 ± 0.0	4.0 ± 0.0	60.5 ± 0.8	-	-
$d = 5, K = 9, T = 100000$	$\epsilon = 0.01$	3.0 ± 0.0	4.0 ± 0.0	102.3 ± 0.9	-	-
	$\epsilon = 0.2$	3.0 ± 0.0	4.0 ± 0.0	101.8 ± 0.6	-	-

A.2. Ablation Study of E^4 on the End of Optimism Instances

In this section, our focus is solely on batched algorithms, as our primary concern lies in evaluating the performance of batched linear bandits algorithms. According to Lattimore & Szepesvari (2017), optimism based algorithms would fail to achieve asymptotic optimality on these hard *End of Optimism* instances. And the construction of such instances need sufficiently small ϵ. So it is meaningful to verify this statement by changing the value of ϵ. Note that rs-OFUL and PhaElimD are both algorithms based on optimism, with the latter one discards low-rewarding arms without considering their information gain. We use the simplest *End of Optimism* instances (see Figure 1) given by (6.1) where we choose $d = 2$, $\epsilon = 0.005, 0.01, 0.05, 0.1, 0.15, 0.2$.



(a) $\epsilon = 0.005$

(b) $\epsilon = 0.01$

(c) $\epsilon = 0.05$

(d) $\epsilon = 0.1$

(e) $\epsilon = 0.15$

(f) $\epsilon = 0.2$

Figure 5: Ablation study on the parameter ϵ.

As the results in Figure 5 show, when ϵ gets larger, the regret of algorithms gets larger. This is because agents can quickly discard the worst arm $\mathbf{x}_3$ and suffer the regret from $\mathbf{x}_2$ in the following rounds. A larger ϵ leads to a larger regret (worse case).

However, we can also see that as ϵ increases, the performance difference between E^4 and other optimistic algorithms decreases. This is because optimism based algorithms can achieve worst-case (instances with large ϵ) optimal regret, but fail to achieve asymptotically optimality under instances with small enough ϵ. In fact, we can see the regret of optimism based algorithms as the minimax optimal regret, and see the regret of E^4 as the asymptotically optimal regret. Hence these experimental results help to understand the difference between these two types of optimality.

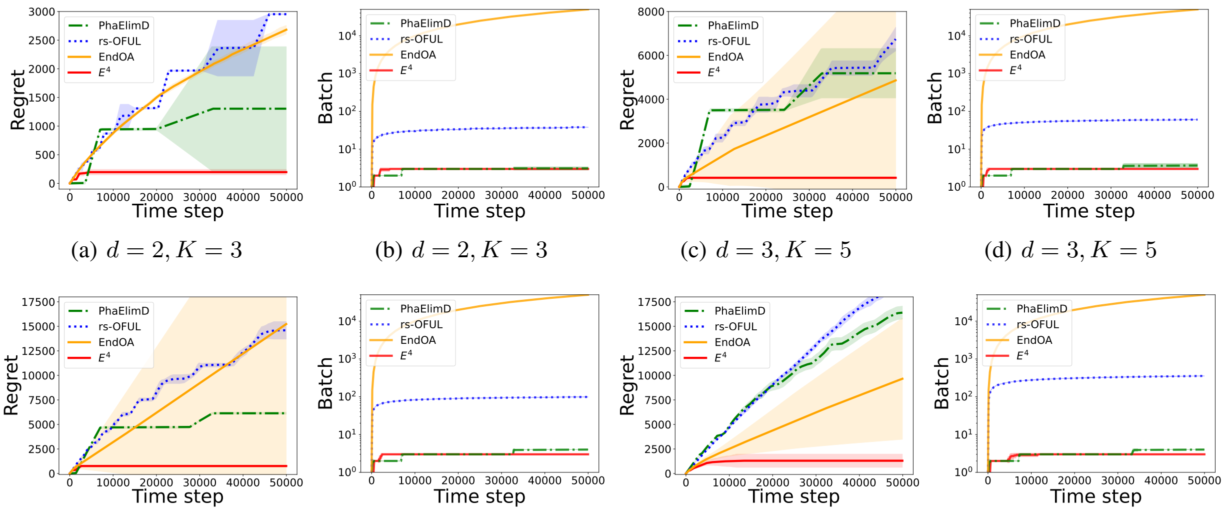
A.3. Evaluations of E^4 on Randomly Generated Instances

In this section, we empirically evaluate algorithms on randomly generated instances. Notably, we have excluded IDS from consideration here due to its consistently poor performance in batch complexity and runtime efficiency as is shown in Figure 2 and Table 2. For each chosen dimension of features and number of arms, we generate an instance with $\|\boldsymbol{\theta}^*\|_2 = 1$, where arm features are sampled from $U([0, 1]^K)$. We set the time horizon to $T = 50000$ and conduct 10 independent experiments for each instance. The regret and batch complexity results are illustrated in Figure 6. The outcomes indicate that our algorithm E^4 exhibits commendable performance even in randomly generated instances, which further demonstrates its advantages.

B. Lower Bounds on the Batch Complexity

The following lemma implies that any algorithm cannot find the best arm correctly with a high enough probability using only one exploration stage of $o(\log T)$ steps.

Lemma B.1. *Consider any linear bandit instances with two arms. Suppose a best-arm-identification algorithm has only $o(\log T)$ steps of exploration, let $\tilde{\mathbf{x}}$ be its output, then there are infinite many T such that $\mathbb{P}(\tilde{\mathbf{x}} \neq \mathbf{x}^*) > 1/\sqrt{T}$.*



(a) $d = 2, K = 3$ (b) $d = 2, K = 3$ (c) $d = 3, K = 5$ (d) $d = 3, K = 5$

(e) $d = 5, K = 9$ (f) $d = 5, K = 9$ (g) $d = 20, K = 50$ (h) $d = 20, K = 50$

Figure 6: Regret and Batch Analysis: Random generated instances ($d = 2, 3, 5, 20$).

Table 4: Batch Complexity Analysis: Random generated instances. Note that again the batch complexity of the sequential algorithm EndOA equals the time horizon 50000.

Instance ($T = 50000$)	E^4	PhaElimD	rs-OFUL	EndOA
$d = 2, K = 3$	3.0 ± 0.0	3.0 ± 0.0	38.0 ± 0.0	-
$d = 3, K = 5$	3.0 ± 0.0	3.7 ± 0.5	60.7 ± 0.5	-
$d = 5, K = 9$	3.0 ± 0.0	4.0 ± 0.0	98.8 ± 0.7	-
$d = 20, K = 50$	3.0 ± 0.0	4.0 ± 0.0	355.5 ± 0.7	-

B.1. Algorithms with Deterministic Number of Batches

We first present the following theorem that states algorithms performing at most 2 batches are not asymptotically optimal.

Theorem B.2. *Algorithms that only perform* 1 *or* 2 *batches are not asymptotically optimal.*

Proof. An algorithm is called asymptotically optimal if its regret satisfies $\limsup_{T\to\infty} R_T/\log T \le c^*(\boldsymbol{\theta}^*)$ for all bandit instances. To prove the desired result, we just need to consider instances given by given by arm set $\mathcal{X} = \{(1, 0), (0, 1)\}$, parameter $\boldsymbol{\theta}^* = (\alpha, 0)$ or $(0, \alpha)$, where $\alpha \in (0, 1)$. Here the gap $\Delta = \alpha$. We aim to prove algorithms that only perform 1 or 2 batches fail to achieve asymptotic optimality on these instances.

First, no consistent algorithm can achieve sub-linear regret with only 1 batch. Algorithms with 1 batch must decide how many times to pull each arm at the beginning, then pull arms for T times in total according to the decision. With no information of unknown instance, it must suffer $O(T)$ regret on some instance.

Then we prove that no algorithm can achieve asymptotic optimality with 2 batches. Assume certain algorithm with policy π can achieve this, meaning its regret satisfies

$$\lim_{T\to\infty} \frac{R_T}{\log T} = c^*(\boldsymbol{\theta}^*) = \sum_{\mathbf{x}\in\mathcal{X}^-} \frac{2}{\Delta_{\mathbf{x}}} = \frac{2}{\alpha}, \tag{B.1}$$

where the second equality is the explicit form of the c^* on multi-armed bandits setting. The detail can be seen from the Example 3 in Lattimore & Szepesvari (2017). (B.1) implies the asymptotically optimal algorithm pulls the optimal arm for $\Theta(T)$ times in expectation, and pulls the suboptimal arm for $\frac{2}{\alpha^2}\log T + o(\log T)$ times in expectation. Since the algorithm has only 2 batches, let t_1 and t_2 be the expected batch sizes for the first and second batch, where $t_1 + t_2 = T$. Without loss of generality, we can assume the agent use a uniform exploration (pull all arms equal times) in the first batch.

Note that we consider the problem in an asymptotic setting, where $T \to \infty$. The statements above show that an optimal algorithm has $\frac{2}{\alpha} \log T + o(\log T)$ regret. Then we claim $t_1 = O(\log T)$. Here we can see that if t_1 has an order higher than $O(\log T)$, the agent must pull one arm for some times with the order larger than $O(\log T)$. On some instances, this leads regret with the order larger than $O(\log T)$, which is an contradiction with asymptotic optimality. We see $t_2 = T - t_1 = T - o(T)$. This means in the second batch, the algorithm must pull one arm for at least $T/2 - o(T)$ times.

Then we prove that t_1 cannot be $o(\log T)$ by contradiction. Suppose an algorithm (denoted as Algorithm A) is a 2-batch asymptotically optimal algorithm with the initial batch size being $o(\log T)$. Another algorithm for best-arm identification, referred to as Algorithm B, can be derived from Algorithm A. This derivation is straightforward: it begins with the exploration phase using the first batch of Algorithm A, collects the data, updates the policy, and then diverges from Algorithm A's approach by outputting the arm, denoted as $\tilde{\mathbf{x}}$, that is to be pulled more frequently in the second batch of Algorithm A as the best-arm. Since the second batch of Algorithm A has size $\Theta(T)$, it must pull $\tilde{\mathbf{x}}$ for $\Theta(T)$ times. Given that Algorithm A is asymptotically optimal, the probability of selecting an incorrect arm is $\mathbb{P}(\tilde{\mathbf{x}} \neq \mathbf{x}^*) \leq 1/\sqrt{T}$ for sufficiently large T. Otherwise, the regret of Algorithm A would be at least $\Theta(T) \times \alpha \times 1/\sqrt{T} = \Theta(\sqrt{T})$, for infinite many T, contradicting its asymptotic optimality, which is characterized by a regret of $O(\log T)$ order. However, viewing Algorithm B as a best-arm identification algorithm introduces a new issue: achieving the bound $\mathbb{P}(\tilde{\mathbf{x}} \neq \mathbf{x}^*) \leq 1/\sqrt{T}$ for sufficiently large T necessitates an exploration process involving more than $\Theta(\log T)$ samples, as indicated by Lemma B.1, which is a contradiction. This implies that for Algorithm A to be asymptotically optimal, it cannot limit its initial batch to only $o(\log T)$ samples. So t_1 cannot be $o(\log T)$. Combining with $t_1 = O(\log T)$ we know $t_1 = \Theta(\log T)$.

But the main problem arises here: how many steps do we need to explore in the first batch? In fact, we have no information about the unknown instance at the beginning. In the first batch, we may choose to pull each arm for $\frac{2}{\beta^2} \log T + o(\log T)$ times, where $\beta > 0$ is a predetermined constant parameter. Then for instances with $\Delta = \alpha > \beta$ and sufficiently large T, the regret of batch 1 is $\alpha \cdot \frac{2}{\beta^2} \log T + o(\log T) > \frac{2}{\alpha} \log T + o(\log T)$, which implies the algorithm is suboptimal. This shows algorithms with only 2 batches cannot achieve asymptotic optimality, which ends the proof. □

B.2. Proof of Theorem 5.2

Firstly, any consistent algorithm must have at least 2 batches. We still consider instances given by given by arm set $\mathcal{X} = \{(1, 0), (0, 1)\}$, parameter $\boldsymbol{\theta}^* = (\alpha, 0)$ or $(0, \alpha)$, where $\alpha \in (0, 1)$. Then we aim to prove any asymptotically optimal must have 3 batches in expectation as $T \to$ on all these instances, which leads to the desired conclusion. Now we prove by contradiction. Suppose that an asymptotically optimal algorithm has $3 - \delta$ batches in expectation on some instance among these, for some constant $\delta \in (0, 1)$. Then we discuss the behavior of this algorithm on this certain bandit instance. Let $\mathcal{M}$ be the event that the algorithm has only 2 batches, and $p = \mathbb{P}(\mathcal{M})$. Therefore the algorithm has at least 3 batches with probability $1 - p$. We have that the expected batch complexity of this algorithm

$$3 - \delta \geq 2 \cdot p + 3 \cdot (1 - p) = 3 - p,$$

which implies $p \geq \delta$.

We use the event $\mathcal{M}$ to decompose the regret:

$$\begin{aligned} R_T &= \mathbb{E}\bigg[\sum_{t=1}^{T} \langle \mathbf{x}^* - \mathbf{x}_t, \boldsymbol{\theta}^* \rangle \bigg| \mathcal{M}\bigg] \cdot p + \mathbb{E}\bigg[\sum_{t=1}^{T} \langle \mathbf{x}^* - \mathbf{x}_t, \boldsymbol{\theta}^* \rangle \bigg| \mathcal{M}^c\bigg] \cdot (1 - p) \\ &:= A_T \cdot p + B_T \cdot (1 - p) \\ &\geq A_T \cdot \delta, \end{aligned}$$

here we use $A_T = \mathbb{E}\big[\sum_{t=1}^{T} \langle \mathbf{x}^* - \mathbf{x}_t, \boldsymbol{\theta}^* \rangle | \mathcal{M}\big]$ to denote the regret given $\mathcal{M}$, and $B_T = \mathbb{E}\big[\sum_{t=1}^{T} \langle \mathbf{x}^* - \mathbf{x}_t, \boldsymbol{\theta}^* \rangle | \mathcal{M}^c\big]$ to denote the regret given $\mathcal{M}^c$. Since we suppose the algorithm is asymptotically optimal, i.e.,

$$\lim_{T \to \infty} \frac{A_T \cdot \delta}{\log T} \leq \lim_{T \to \infty} \frac{R_T}{\log T} = c^*,$$

which implies $A_T \leq c^*/\delta \log T + o(\log T)$. Given $\mathcal{M}$, we denote $T = t_1 + t_2$, where t_1 and t_2 are the expected batch sizes for two batches, respectively. Similar to the proof of Theorem B.2, we have $t_1 = O(\log T)$ and $t_2 = \Theta(T)$, otherwise the order of the regret would be larger than $\Omega(\log T)$. We can also use the same arguments as the proof of Theorem B.2 to prove that t_1 cannot be $o(\log T)$. Otherwise, $A_T = \Omega(\sqrt{T})$. Combining these results we know $t_1 = \Theta(\log T)$.

The behavior of the algorithm in the first batch is not influenced by bandit instances. The algorithm must pull $\Theta(\log T)$ arms in the first batch. Then we just analyze the regret in the first batch. And the algorithm faces the same problem as the proof of Theorem B.2: how many steps does it need to explore in the first batch? Use exactly the same way we can find an instance where the algorithm is suboptimal, which is a contradiction. Actually, no matter how the agent allocates these $\Theta(\log T)$ steps of exploration, there exists an instance that makes the algorithm suboptimal. This ends the proof.

C. Proofs of Main Theorems in Section 5

The two lemmas outlined below offer a theoretical basis for the exploration of D-optimal design and the application of Chernoff's stopping rule mentioned in Section 4.

In practice, we use the Frank–Wolfe algorithm to solve the optimization problem in (4.1) to get a near-optimal solution $\tilde{\boldsymbol{\pi}}$, which only exerts an impact on our theoretical analysis up to a constant order. In this way, the agent pulls each arm $\mathbf{x} \in \mathcal{X}$ for $\lceil 2\tilde{\pi}_{\mathbf{x}} g(\tilde{\boldsymbol{\pi}}) M/d \rceil$ times. Besides, the number of iterations for this algorithm can be $O(d \log\log d)$, which shows the efficiency. See Section 21.2 in Lattimore & Szepesvári (2020) for more details.

Lemma C.1 (D-optimal design concentration (Lattimore & Szepesvári, 2020))**.** *Let $\mathcal{A}$ be the active arm set. If the agent performs a D-optimal design exploration on $\mathcal{A}$ with rate M defined in Definition 4.1, we can get the concentration results for related least squares estimators: for any $\mathbf{x} \in \mathcal{X}$, with probability at least $1 - 1/(KT^2)$,*

$$|\langle \mathbf{x}, \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* \rangle| \le \sqrt{\frac{d \log(KT^2)}{M}}.$$

Lemma C.2 ((Jedra & Proutiere, 2020))**.** *Let $\delta \in (0,1)$, $u > 0$. Regardless of the sampling strategy, the Chernoff's Stopping Rule defined as*

$$Z(t) \ge \beta(t,\delta) \text{ and } \sum_{s=1}^{t} \mathbf{x}_s \mathbf{x}_s^\top \ge c\mathbf{I}_d$$

with $c = \max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_2^2$ and

$$Z(t) = \min_{\mathbf{b} \ne \hat{\mathbf{x}}^*} \frac{\hat{\Delta}_{\mathbf{b}}^2}{2\|\hat{\mathbf{x}}^* - \mathbf{b}\|_{\mathbf{H}_t^{-1}}^2},$$
$$\beta(t,\delta) = (1+u) \log\left(\frac{u^{-\frac{d}{2}} t^{\frac{d}{2}}}{\delta}\right).$$

ensures that $\mathbb{P}\big(\tau < \infty, \langle \boldsymbol{\theta}^*, \mathbf{x}^* - \hat{\mathbf{x}}_\tau^* \rangle > 0\big) \le \delta$, where $\hat{\boldsymbol{x}}_\tau^*$ is the estimated best arm at time step τ. This shows that if the stopping rule holds, we can find the best arm correctly with probability of at least $1-\delta$.

Note that we need different concentrations for each batches, with each just based on one batch of independent data. This makes the proof much easier. The subsequent lemmas are crucial as they assist in demonstrating the asymptotic properties of Algorithm 1. Lemma C.3 gives a concentration result in case that the agents use at least a D-optimal design exploration with rate $(\log T)^{1/2}$.

Lemma C.3. *Suppose in one batch of Algorithm 1, the agent conducts at least a D-optimal design exploration with rate $(\log T)^{1/2}$ given in Definition 4.1. This means (1) $t_1 = \Theta\big((\log T)^{1/2}\big)$ pulls for D-optimal design exploration, (2) possibly some other pulls. Let $t \ge t_1$ be the total number of pulls in this batch, $\hat{\boldsymbol{\theta}}_{t_1}, \hat{\boldsymbol{\theta}}_t$ be the least squares estimators of $\boldsymbol{\theta}^*$ after the D-optimal design exploration and the whole batch of exploration, $\mu_{\mathbf{x}} = \langle \boldsymbol{\theta}^*, \mathbf{x} \rangle$, $\hat{\mu}_{\mathbf{x}}(t) = \langle \hat{\boldsymbol{\theta}}_t, \mathbf{x} \rangle$ for $\mathbf{x} \in \mathcal{X}$, $\epsilon = 1/\log\log T$. Define an event*

$$\mathcal{E}_1 = \{\forall t \ge t_1, \forall \mathbf{x} \in \mathcal{X}, |\hat{\mu}_{\mathbf{x}}(t) - \mu_{\mathbf{x}}| \le \epsilon\}. \tag{C.1}$$

Then for sufficiently large T, we have

$$\mathbb{P}(\mathcal{E}_1^c) \le \frac{1}{(\log T)^2}.$$

Next Lemma C.4 describes the exact concentration result for the first two batches in our Algorithm 1. Note that in the estimation stage of the ℓ-th batch, we only use data collected in the exploration stage of this batch to calculate the least squares estimators.

Lemma C.4. *Let $\epsilon = 1/\log\log T$, b_1 and b_2 be the batch size of the first two batches in Algorithm 1, respectively. For the first two batches, we use $\hat{\mu}_{\mathbf{x}}(b_1) = \langle \hat{\boldsymbol{\theta}}_{b_1}, \mathbf{x}\rangle$ and $\hat{\mu}_{\mathbf{x}}(b_2) = \langle \hat{\boldsymbol{\theta}}_{b_2}, \mathbf{x}\rangle$ to denote the estimated expected rewards for $\mathbf{x} \in \mathcal{X}$ using data collected in the first and the second batches, respectively. Define*

$$\mathcal{E}_2 = \{\forall \mathbf{x} \in \mathcal{X}, |\hat{\mu}_{\mathbf{x}}(b_1) - \mu_{\mathbf{x}}| \le \epsilon, |\hat{\mu}_{\mathbf{x}}(b_2) - \mu_{\mathbf{x}}| \le \epsilon\}. \tag{C.2}$$

Then for sufficiently large T, we have

$$\mathbb{P}(\mathcal{E}_2^c) \le \frac{2}{(\log T)^2}.$$

Now we talk about some properties of the optimal allocation solved from Definition 4.3. Without loss of generality, we can always assume the allocation for $\hat{\mathbf{x}}^*$ solved from Definition 4.3 satisfies

$$w_{\hat{\mathbf{x}}^*} = (\log T)^\gamma/\alpha. \tag{C.3}$$

In fact, if we have one solution $\mathbf{w}(\hat{\Delta}) \in [0,\infty)^K$ to the program in Definition 4.3, then we can increase the value $w_{\hat{\mathbf{x}}^*}$ to $w'_{\hat{\mathbf{x}}^*} \ge w_{\hat{\mathbf{x}}^*}$ to get another solution. This is because the increase of $w_{\hat{\mathbf{x}}^*}$ doesn't affect the objective function $\sum_{\mathbf{x}\neq\hat{\mathbf{x}}^*} w_{\mathbf{x}}(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)$. Moreover, this increase can minimize $\|\mathbf{x} - \hat{\mathbf{x}}^*\|^2_{\mathbf{H}_w^{-1}}$ for $\mathbf{x} \neq \hat{\mathbf{x}}^*$, which makes the constraint still true.

In the following analysis we use $\{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ to denote $\mathbf{w}(\hat{\Delta})$ solved using estimated gaps. Next definition gives the allocation solved from the true parameters, which we will use in our theoretic analysis. It's easy to see Definition 4.3 is an approximate version of Definition C.5. Using the same statements as above, we can assume $w^*_{\mathbf{x}^*} = \infty$.

Definition C.5. Let $\Delta = \{\Delta_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}} \in [0,\infty)^K$ be the vector of gaps, $\mathbf{x}^* = \mathrm{argmax}_{\mathbf{x}\in\mathcal{X}}\langle \boldsymbol{\theta}^*, \mathbf{x}\rangle$ be the best arm. We define $\mathbf{w}(\Delta) = \{w^*_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}} \in [0,\infty]^K$ to be the solution to the optimisation problem

$$\min_{\mathbf{w}\in[0,\infty]^K} \sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}}\Delta_{\mathbf{x}}$$

$$\text{s.t.} \quad \|\mathbf{x} - \mathbf{x}^*\|^2_{\mathbf{H}_{\mathbf{w}}^{-1}} \le \frac{\Delta_{\mathbf{x}}^2}{2}, \quad \forall \mathbf{x} \in \mathcal{X}^-,$$

where $\mathbf{H}_{\mathbf{w}} = \sum_{\mathbf{x}\in\mathcal{X}} w_{\mathbf{x}} \cdot \mathbf{x}\mathbf{x}^\top$.

The following lemma shows the solution from Definition 4.3 converges to the solution from Definition C.5 as $T \to \infty$.

Lemma C.6. *We use $\mathbf{w}(\hat{\Delta}) = \{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ from Definition 4.3 and $\mathbf{w}(\Delta) = \{w^*_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ from Definition C.5. Then if $\mathcal{E}_2$ holds,*

$$\lim_{T\to\infty} w_{\mathbf{x}} = w^*_{\mathbf{x}}, \quad \forall \mathbf{x} \in \mathcal{X}.$$

The following lemmas demonstrate that for sufficiently large T the stopping rule in the second batch of Algorithm 1 holds with probability larger than $1 - 1/(\log T)^2$, and **while** loop breaks after the third batch with probability larger than $1 - 1/T$.

Lemma C.7. *If $\mathcal{E}_2$ holds, then for sufficiently large T, the stopping condition (4.2) holds in the second batch.*

*Namely, Algorithm 1 skip the **while** loop after the second batch and enter the Exploitation stage with probability of at least $1 - 2/(\log T)^2$, since $\mathbb{P}(\mathcal{E}_2^c) \le 2/(\log T)^2$.*

Lemma C.8. *If the stopping rule (4.2) doesn't hold in the second batch, for sufficiently large T, the algorithm would eliminate all suboptimal arms in the third batch with probability of at least $1 - 1/T$.*

Let $Regret_2$ be the regret for batch $\ell = 2$ of Algorithm 1. The following lemma implies that $Regret_2$ matches the leading term in the asymptotic lower bound in Lemma 3.1.

Lemma C.9. *Regret of the second batch can be bounded by:*

$$\lim_{T\to\infty} \frac{Regret_2}{\log T} \le c^*,$$

where $c^* = c^*(\boldsymbol{\theta}^*)$ defined in Lemma 3.1.

C.1. Proof of Theorem 5.4

Now we prove the results in Theorem 5.4. Note that under the choice of exploration rate $\mathcal{T}_1$, we want to show that E^4 achieves the optimal regret and the batch complexity in both the finite-time setting and the asymptotic setting.

C.1.1. The Finite-time Setting

We first prove the batch complexity and the worst case regret bound in the finite-time setting.

Batch complexity: In this part, we analyze the upper bound of batch complexity. It is sufficient to count the number of **while** loops starting from the third batch in Algorithm 1 before *Exploitation* in Line 1. For $\ell \geq 4$, the ℓ-th batch has size $\Theta\big(T^{1-\frac{1}{2^{\ell-3}}}\big)$, so the $(\log\log T + 3)$-th batch has size $\Theta(T)$. Suppose the $(\log\log T + 3)$-th batch size is larger than T/M, where $M > 0$ is a constant. By definition, the batch size increases as t increases, so the sizes of batches after this batch is larger than T/M. Therefore, the algorithm would have at most $\log\log T + 3 + M$ batches. Then we prove the batch complexity upper bound $O(\log\log T)$.

Minimax optimality: First, we assume the agent doesn't eliminate any arms in the first two batches, because it's easier the other way as we will show at the end of this part of the proof. Since the number of time steps for the first two batches is at most $O\big((\log T)^{1+\gamma}\big)$, the regret of the first two batches is at most $O\big((\log T)^{1+\gamma}\big)$, due to the bounded rewards assumption. Besides, the property of Chernoff's Stopping Rule (Lemma C.2) means the agent finds the best arm correctly with probability larger than $1 - 1/T$, then the regret of Line 1 *Exploitation* stage can be bounded by a small constant. Hence we don't need to consider the regret in Line 1 *Exploitation* stage.

In Algorithm 1, we let $\mathcal{A}$ be the current active arm set. Note that $\mathcal{A}$ may shrink after some batches due to the *Elimination* stages. We define an arm $\mathbf{x}$ to be active in batch ℓ, if it is in the active arm set $\mathcal{A}$ of this batch. The same as (Esfandiari et al., 2021) and (Lattimore & Szepesvári, 2020), we define the following *good event* $\mathcal{E}_0$ (C.4) that we will use in the following proof constantly:

$$\begin{aligned}\mathcal{E}_0 = \{&\text{For any arm } \mathbf{x} \text{ that is active in the beginning of batch } \ell, \\ &\text{at the end of this batch we have } |\langle \mathbf{x}, \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\rangle| \leq \varepsilon_\ell.\}\end{aligned} \tag{C.4}$$

By Definition 4.1, for any $\mathbf{x} \in \mathcal{X}$, after the ℓ-th batch, with probability at least $1 - 1/(KT^2)$, we have

$$|\langle \mathbf{x}, \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\rangle| \leq \varepsilon_\ell.$$

Since there are K arms and at most T batches, by union bound we know the good event $\mathcal{E}_0$ (C.4) happens with probability at least $1 - 1/T$. Consequently, we know

$$\begin{aligned}R_T &= \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \langle \mathbf{x} - \mathbf{x}_t, \boldsymbol{\theta}^*\rangle \middle| \mathcal{E}_0^c\right] \cdot \mathbb{P}(\mathcal{E}_0^c) + \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \langle \mathbf{x} - \mathbf{x}_t, \boldsymbol{\theta}^*\rangle \middle| \mathcal{E}_0\right] \cdot \mathbb{P}(\mathcal{E}_0) \\ &\leq 2LT\mathbb{P}(\mathcal{E}_0^c) + \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \langle \mathbf{x} - \mathbf{x}_t, \boldsymbol{\theta}^*\rangle \middle| \mathcal{E}_0\right] \cdot \mathbb{P}(\mathcal{E}_0) \\ &= O(1) + \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \langle \mathbf{x} - \mathbf{x}_t, \boldsymbol{\theta}^*\rangle \middle| \mathcal{E}_0\right] \cdot \mathbb{P}(\mathcal{E}_0).\end{aligned}$$

So we can just bound the regret when $\mathcal{E}_0$ happen. Under $\mathcal{E}_0$, at the end of each batch $\ell \geq 3$, we know for each arm,

$$|\langle \mathbf{x}, \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\rangle| \leq \varepsilon_\ell.$$

Denote the best arm and estimated best arm (ties broken arbitrarily) by:

$$\begin{aligned}\mathbf{x}^* &= \operatorname*{argmax}_{\mathbf{x}\in\mathcal{A}} \langle \mathbf{x}, \boldsymbol{\theta}^*\rangle, \\ \hat{\mathbf{x}}^* &= \operatorname*{argmax}_{\mathbf{x}\in\mathcal{A}} \langle \mathbf{x}, \hat{\boldsymbol{\theta}}\rangle.\end{aligned}$$

Then,

$$\begin{aligned}\max_{y\in\mathcal{A}}\langle\hat{\boldsymbol{\theta}},\mathbf{y}-\mathbf{x}^*\rangle &= \langle\hat{\boldsymbol{\theta}},\hat{\mathbf{x}}^*-\mathbf{x}^*\rangle\\ &= \langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*+\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*-\mathbf{x}^*\rangle\\ &= \langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*\rangle-\langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,\mathbf{x}^*\rangle+\langle\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*-\mathbf{x}^*\rangle\\ &\le |\langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*\rangle|+|\langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,\mathbf{x}^*\rangle|+\langle\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*-\mathbf{x}^*\rangle\\ &\le 2\varepsilon_\ell+\langle\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*-\mathbf{x}^*\rangle\\ &\le 2\varepsilon_\ell.\end{aligned}$$

Compared this with the elimination rule in Algorithm 1 for $\ell\ge 3$, we know the best arm won't be eliminated, namely,

$$\mathbf{x}^*\in\mathcal{A}_\ell,\quad \forall\ell\ge 3,$$

here we use $\mathcal{A}_\ell$ to denote the active action set in the ℓ-th batch. Correspondingly, for each suboptimal arm $\mathbf{x}$, define $\ell_{\mathbf{x}}=\min\{\ell:4\varepsilon_\ell<\Delta_{\mathbf{x}}\}$ to be the first phase where the suboptimality gap of arm $\mathbf{x}$ bigger than $4\varepsilon_\ell$. Then if $\mathbf{x}\neq\mathbf{x}^*$ is not eliminated in the first $\ell-1$ batches, in the ℓ-th batch,

$$\begin{aligned}\max_{y\in\mathcal{A}}\langle\hat{\boldsymbol{\theta}},\mathbf{y}-\mathbf{x}\rangle &\ge \langle\hat{\boldsymbol{\theta}},\mathbf{x}^*-\mathbf{x}\rangle\\ &= \langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*+\boldsymbol{\theta}^*,\mathbf{x}^*-\mathbf{x}\rangle\\ &= \langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,\mathbf{x}^*-\mathbf{x}\rangle+\langle\boldsymbol{\theta}^*,\mathbf{x}^*-\mathbf{x}\rangle\\ &\ge -|\langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,\hat{\mathbf{x}}^*\rangle|-|\langle\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*,x\rangle|+\Delta_{\mathbf{x}}\\ &> 2\varepsilon_\ell.\end{aligned}$$

Compared this with the elimination rule in Algorithm 1 for $\ell\ge 3$, we know $\mathbf{x}\neq\mathbf{x}^*$ would be eliminated in the first ℓ batches, namely,

$$x\notin\mathcal{A}_{\ell_{\mathbf{x}}+1},\quad\forall\mathbf{x}\neq\mathbf{x}^*, \tag{C.5}$$

where $\mathcal{A}_{\ell_{\mathbf{x}}+1}$ is the active action set in the $\ell_{\mathbf{x}}+1$ batch. Thus when the good event $\mathcal{E}_0$ (C.4) happens, arms active in phase ℓ have the property: $\Delta_{\mathbf{x}}\le 4\varepsilon_{\ell-1}$.

When $T_3=(\log T)^{1+\gamma}, T_\ell=T^{1-\frac{1}{2^{\ell-3}}}, \ell\ge 4$ and $\varepsilon_\ell=\sqrt{d\log(KT^2)/T_\ell}$, Algorithm 1 has at most $B=O(\log\log T)$ batches. We use $Regret_\ell$ to denote the regret of the ℓ-th batch, then the regret for batches $\ell\ge 5$ under $\mathcal{E}_0$ can be bounded as

$$\begin{aligned}\sum_{\ell=5}^{B} Regret_\ell &\le \sum_{\ell=5}^{B} 4T_\ell\varepsilon_{\ell-1}\\ &\le \sum_{\ell=5}^{B} 4CT^{1-\frac{1}{2^{\ell-3}}}\sqrt{d\log(KT^2)/T^{1-\frac{1}{2^{\ell-4}}}}\\ &= \sum_{\ell=5}^{B} 4C\sqrt{dT\log(KT^2)}\\ &= O\big(\log\log T\cdot\sqrt{dT\log(KT)}\big).\end{aligned}$$

As a result, the regret for batches $\ell\ge 3$ under $\mathcal{E}_0$ can be bounded as

$$\begin{aligned}\sum_{\ell=3}^{B} Regret_\ell &\le \sum_{\ell=3}^{B} 4T_\ell\varepsilon_{\ell-1}\\ &= O(T_3)+O(T_4)+\sum_{\ell=5}^{B} 4T_\ell\varepsilon_{\ell-1}\end{aligned}$$

$$= O\big((\log T)^{1+\gamma}\big) + O\big(\sqrt{T}\big) + \sum_{\ell=5}^{B} 4C\sqrt{dT\log(KT^2)}$$
$$= O\big(\log\log T \cdot \sqrt{dT\log(KT)}\big).$$

All of the proofs above suppose the algorithm doesn't eliminate any arms in the first two batches, i.e. the stopping rule in the second batch doesn't hold. Now we suppose the algorithm eliminates all suboptimal arms after the second batch. The property of Chernoff's Stopping Rule Lemma C.2 means the agent finds the optimal arm with probability at least $1 - 1/T$, then the regret of Line 1 *Exploitation* after the second batch can be bounded by a constant. We can combine these to end the proof.

C.1.2. The Asymptotic Setting

Now we investigate the batch complexity and the regret bound of E^4 when $T \to \infty$.

Batch complexity: By Lemma C.7, we know for sufficiently large T, with probability $1 - 2/(\log T)^2$, the stopping rule (4.2) in the second batch of Algorithm 1 holds. This means the algorithm identifies the best arm and goes into the Line 1, the *Exploitation* stage. According to the last parts of proof, with time grid selections $\mathcal{T}_1$ (and even $\mathcal{T}_2$), the algorithm has at most $O(\log T)$ batches in total. So we can calculate the expected batch complexity by

$$\lim_{T\to\infty} 3 \cdot \big(1 - 2/(\log T)^2\big) + O\big(\log T/(\log T)^2\big) = 3, \tag{C.6}$$

because the algorithm just has 3 batches when the stopping rule holds, and this event happens with probability at least $1 - 2/(\log T)^2$. Thus we complete the proof of asymptotic batch complexity.

Asymptotic optimality: Lemmas C.7 and C.8 demonstrate that for sufficiently large T the stopping rule in the second batch of Algorithm 1 holds with probability larger than $1 - 2/(\log T)^2$, and the **while** loop starting from batch $\ell = 3$ breaks after the third batch with probability larger than $1 - 1/T$.

Decompose accumulative regret by batches as:

$$R_T = \mathit{Regret}_1 + \mathit{Regret}_2 + \mathit{Regret}_3 + \mathit{Regret}_{else},$$

where Regret_ℓ represents the regret in the ℓ-th batch ($\ell = 1, 2, 3$), and Regret_{else} represents the regret after the third batch.

We research on the asymptotic setting, so all of the statements below assumes that T is sufficiently large. With the assumption of bounded rewards, regret in the first batch is negligible compared with $\Theta(\log T)$ order, namely, $\mathit{Regret}_1 = o(\log T)$:

$$\lim_{T\to\infty} \frac{\mathit{Regret}_1}{\log T} = \frac{(\log T)^{1/2}}{\log T} = 0.$$

Note that the stopping rule (4.2) in the second batch holds with probability at least $1 - 2/(\log T)^2$. But the batch $\ell = 3$ in the **while** loop is encountered only when the stopping rule fails to hold. Given that the batch 3 has a size of $O\big((\log T)^{1+\gamma}\big)$, the regret in this batch is considered negligible:

$$\lim_{T\to\infty} \mathit{Regret}_3 = \lim_{T\to\infty} \frac{2}{(\log T)^2} \cdot O\big((\log T)^{1+\gamma}\big) = 0.$$

Similarly, note that the agent eliminates all suboptimal arms after the third batch with probability $1 - 1/T$, according to Lemma C.8, so the regret after this batch Regret_{else} is negligible, too:

$$\lim_{T\to\infty} \frac{\mathit{Regret}_{else}}{\log T} = \lim_{T\to\infty} \frac{\frac{1}{T}\cdot O(T)}{\log T} = 0.$$

The only term of care about is the regret of the second batch of Algorithm 1. Therefore, it is sufficient to use Lemma C.9 to conclude the proof of asymptotically optimal regret, which shows

$$\lim_{T\to\infty} \frac{\mathit{Regret}_2}{\log T} \le c^*.$$

Now we can conclude our prove of the main Theorem 5.4 by combining all these results about different parts of regret:

$$\begin{aligned}\limsup_{T\to\infty}\frac{R_T}{\log T} &= \limsup_{T\to\infty}\frac{Regret_1+Regret_2+Regret_3+Regret_{else}}{\log T}\\ &= \limsup_{T\to\infty}\frac{Regret_2}{\log T}\\ &\le c^*,\end{aligned}$$

which matches the lower bound in Lemma 3.1.

C.2. Proof of Theorem 5.7

Finite-time instance-dependent batch complexity and regret bound: This part of proof is similar to the analysis in the proof of Theorem 5.4. We use the same good event $\mathcal{E}_0$ in (C.4). Here we choose $T_\ell = \Theta(d\log(KT^2)\cdot 2^{\ell-3})$, $\varepsilon_\ell = \sqrt{1/2^{\ell-3}}, \ell \ge 4$. Then the $\log T$-th batch in Algorithm 1 has size $\Theta(d\log(KT^2)T)$. Therefore, the **while** loop in the algorithm would break with batches at most $O(\log T)$.

In this part we can also suppose the agent doesn't eliminate any arms in the first two batches. The reason for this is the same as the proof of Theorem 5.4. Similarly, the regret of the first three batches is at most $O\big((\log T)^{1+\gamma}\big)$ and the regret of Line 1 *Exploitation* stage is $O(1)$, which is negligible. To analyze the instance-dependent regret, we focus on batches for $\ell \ge 4$. Define $\Delta_{\min} = \min_{\mathbf{x}\neq\mathbf{x}^*}\Delta_{\mathbf{x}}$. Recall that under the good event $\mathcal{E}_0$ (C.4), the arm $\mathbf{x}\neq\mathbf{x}^*$ would be eliminated after the $\ell_{\mathbf{x}}$ batch. We can explicitly calculate when would all suboptimal arms be eliminated. Solving

$$\Delta_{\min} > 4\varepsilon_\ell \tag{C.7}$$

we get

$$\ell > \log\left(\frac{1}{\Delta_{\min}^2}\right)+7. \tag{C.8}$$

Define $L_0 = \left\lceil \log\left(\frac{1}{\Delta_{\min}^2}\right)\right\rceil + 7 = O\big(\log(\frac{1}{\Delta_{\min}})\big)$. We just need to consider regret in the first L_0 batches. Because after that the only active action is the optimal one.

Absolute the same as the proof of Theorem 5.4, we have

$$\begin{aligned}\sum_{\ell=4}^{L_0} Regret_\ell &\le \sum_{\ell=4}^{L_0} 4T_\ell\varepsilon_{\ell-1}\\ &\le \sum_{\ell=4}^{L_0} 4C\cdot d\log(KT^2)\cdot 2^{\ell-3}\sqrt{1/2^{\ell-4}}\\ &= \sum_{\ell=4}^{L_0} 8C\cdot d\log(KT^2)2^{(\ell-4)/2}\\ &\le 8C\cdot d\log(KT^2)2^{(L_0-3)/2}\\ &= O(d\log(KT)/\Delta_{\min}).\end{aligned} \tag{C.9}$$

Combining with the regret of the first three batches, we have:

$$R_T = O\big((\log T)^{1+\gamma} + d\log(KT)/\Delta_{\min}\big). \tag{C.10}$$

Since the good event $\mathcal{E}_0$ (C.4) happens with probability at least $1-1/T$, the expected batch complexity of the algorithm is

$$O\bigg(\log\left(\frac{1}{\Delta_{\min}}\right) + (\log T)/T\bigg) = O\bigg(\log\left(\frac{1}{\Delta_{\min}}\right)\bigg).$$

Minimax optimality: We can also prove $R_T = \tilde{O}(\sqrt{dT})$, which implies the algorithm with $\{T_\ell\}_{\ell=1}^\infty = \mathcal{T}_2$ is nearly minimax optimal.

When $\Delta_{\min} > \sqrt{d/T}$, (C.10) is a tighter bound compared to $\tilde{O}(\sqrt{dT})$. So we just need to consider the case $\Delta_{\min} \le \sqrt{d/T}$. Here the same as the methods in (C.7)-(C.8), we can prove that under $\mathcal{E}_0$, arms with gap larger than $\sqrt{d/T}$ would be eliminated in the first $L' = \lceil \log(T/d) \rceil + 5$ batches. However, the regret caused by arms with gaps less than $\sqrt{d/T}$ is at most $T \cdot \sqrt{d/T} = \sqrt{dT}$. Note that the regret for batches $\ell = 4, \ldots, L'$ in the **while** loop can be bounded by

$$\begin{aligned}\sum_{\ell=4}^{L'} Regret_\ell &\le \sum_{\ell=4}^{L'} 4T_\ell \varepsilon_{\ell-1} \\ &\le 8C \cdot d\log(KT^2)2^{(L'-3)/2} \\ &= O\big(\sqrt{dT} \cdot \log(KT)\big),\end{aligned} \tag{C.11}$$

where we get (C.11) the same as (C.9). As a result, the total regret can be bounded by

$$\begin{aligned}R_T &\le \sum_{\ell=1}^{3} Regret_\ell + \sum_{\ell=4}^{L'} Regret_\ell + \sqrt{dT} \\ &= O\big((\log T)^{1+\gamma}\big) + \sum_{\ell=4}^{L'} Regret_\ell + \sqrt{dT} \\ &= O\big(\sqrt{dT} \cdot \log(KT)\big).\end{aligned} \tag{C.12}$$

Asymptotic regret and batch complexity: When $\{T_\ell\}_{\ell=1}^\infty$ is set to either $\mathcal{T}_1$ or $\mathcal{T}_2$, Algorithm 1 yields identical performances within the initial three batches. Thus the corresponding portion of the proof aligns with that of Theorem 5.4. Specifically, according to Lemmas C.7 and C.8, for a sufficiently large T, the probability that the algorithm avoids batch $\ell = 3$ exploration at least $1 - 2/(\log T)^2$. Similarly, the probability that it avoids batch $\ell = 4$ is at least $1 - 1/T$. By utilizing the same rationale, we can complete this part of the proof.

D. Proofs of Technical Lemmas

D.1. Proof of Lemma B.1

We prove the statement by contradiction. Suppose there is an algorithm π such that for sufficient large T

$$\mathbb{P}(\tilde{\mathbf{x}} \ne \mathbf{x}^*) \le 1/\sqrt{T}.$$

The algorithm terminates after $b = o(\log T)$ steps and output the estimated best arm $\tilde{\mathbf{x}}$. By the assumption above we know $\mathbb{P}(\tilde{\mathbf{x}} \ne \mathbf{x}^*) \le 1/\sqrt{T}$, so this is a δ-PAC algorithm with $\delta = 1/\sqrt{T}$.

However, by the Theorem 1 in Jedra & Proutiere (2020), for a $(1/\sqrt{T})$-PAC strategy, any bandit instance $\boldsymbol{\theta}^*$ and sufficiently large T,

$$b \ge T^*(\boldsymbol{\theta}^*) \log\big(\sqrt{T}\big) = \Theta(\log T),$$

which is contradictory with $b = o(\log T)$. Therefore, the initial assumption is incorrect. We can now conclude the proof of $\mathbb{P}(\tilde{\mathbf{x}} \ne \mathbf{x}^*) > 1/\sqrt{T}$ for infinite many T.

D.2. Proof of Lemma C.4

Here we can use Lemma C.3 directly.

In the first batch of Algorithm 1, the agent just use a D-optimal design exploration given in Definition 4.1 with $b_1 = t_1 = \Theta\big((\log T)^{1/2}\big)$ pulls. In the second batch of Algorithm 1, the agent use a D-optimal design exploration given in

Definition 4.1 with $b_1 = t_1 = \Theta\big((\log T)^{1/2}\big)$ pulls, and pull arms $\mathbf{x} \in \mathcal{X}$ for another $\min\big\{w_{\mathbf{x}} \cdot \alpha \log T, (\log T)^{1+\gamma}\big\}$ times. The behavior of the first two batches satisfies the conditions of Lemma C.3. Therefore, by the conclusion of Lemma C.3,

$$\mathbb{P}(\forall \mathbf{x} \in \mathcal{X}, |\hat{\mu}_{\mathbf{x}}(b_1) - \mu_{\mathbf{x}}| \le \epsilon) \ge 1 - \frac{1}{(\log T)^2},$$
$$\mathbb{P}(\forall \mathbf{x} \in \mathcal{X}, |\hat{\mu}_{\mathbf{x}}(b_2) - \mu_{\mathbf{x}}| \le \epsilon) \ge 1 - \frac{1}{(\log T)^2}.$$

Hence by union bound we have

$$\mathbb{P}(\mathcal{E}_2) = \mathbb{P}(\forall \mathbf{x} \in \mathcal{X}, |\hat{\mu}_{\mathbf{x}}(b_1) - \mu_{\mathbf{x}}| \le \epsilon, |\hat{\mu}_{\mathbf{x}}(b_2) - \mu_{\mathbf{x}}| \le \epsilon) \ge 1 - \frac{2}{(\log T)^2},$$

as needed.

D.3. Proof of Lemma C.6

First, if $\mathcal{E}_2$ holds, for sufficiently large T, $\epsilon = 1/\log\log T < \Delta_{\min}/2$, so the estimated best arm is really the best arm, namely, $\hat{\mathbf{x}}^* = \mathbf{x}^*$. Then

$$\lim_{T\to\infty} w_{\mathbf{x}^*} = \lim_{T\to\infty} (\log T)^{\gamma}/\alpha = \infty = w^*_{\mathbf{x}^*}.$$

Besides, as $T \to \infty$, $(\log T)^{\gamma}/\alpha \to \infty$, so the last restricted condition $w_{\hat{\mathbf{x}}^*} \le (\log T)^{\gamma}/\alpha$ in the optimisation problem in Definition 4.3 has no effect. In addition, by the concentration result of $\mathcal{E}_2$, we know

$$\lim_{T\to\infty} \hat{\Delta}_{\mathbf{x}} - 4\epsilon = \Delta_{\mathbf{x}}, \quad \forall \mathbf{x} \in \mathcal{X}.$$

Then we can compare the two programs in Definition 4.3 and Definition C.5 line by line. By the continuity of the program we know

$$\lim_{T\to\infty} w_{\mathbf{x}} = w^*_{\mathbf{x}}, \quad \forall \mathbf{x} \in \mathcal{X}.$$

D.4. Proof of Lemma C.7

If $\mathcal{E}_2$ holds,

$$|\hat{\mu}_{\mathbf{x}}(b_1) - \mu_{\mathbf{x}}| \le \epsilon, |\hat{\mu}_{\mathbf{x}}(b_2) - \mu_{\mathbf{x}}| \le \epsilon, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{D.1}$$

For sufficiently large T, $\epsilon = 1/\log\log T < \Delta_{\min}/2$. Hence by (D.1), for sufficiently large T and $\mathbf{x} \ne \mathbf{x}^*$,

$$\begin{aligned}\hat{\mu}_{\mathbf{x}^*}(b_1) - \hat{\mu}_{\mathbf{x}}(b_1) &= \hat{\mu}_{\mathbf{x}^*}(b_1) - \mu_{\mathbf{x}^*} - (\hat{\mu}_{\mathbf{x}}(b_1) - \mu_{\mathbf{x}}) + \Delta_{\mathbf{x}} \\ &\ge -|\hat{\mu}_{\mathbf{x}^*}(b_1) - \mu_{\mathbf{x}^*}| - |\hat{\mu}_{\mathbf{x}}(b_1) - \mu_{\mathbf{x}}| + \Delta_{\mathbf{x}} \\ &\ge -2\epsilon + \Delta_{\mathbf{x}} > 0,\end{aligned}$$

which implies the estimated best arm is really the best arm, namely, $\hat{\mathbf{x}}^* = \mathbf{x}^*$. And for suboptimal arm $\mathbf{x} \ne \mathbf{x}^*$, we have

$$|\hat{\Delta}_{\mathbf{x}} - \Delta_{\mathbf{x}}| = |(\hat{\mu}_{\mathbf{x}^*} - \hat{\mu}_{\mathbf{x}}) - (\mu_{\mathbf{x}^*} - \mu_{\mathbf{x}})| \le |\hat{\mu}_{\mathbf{x}^*} - \mu_{\mathbf{x}^*}| + |\hat{\mu}_{\mathbf{x}} - \mu_{\mathbf{x}}| \le 2\epsilon. \tag{D.2}$$

The second stopping rule $\sum_{s=1}^{t} \mathbf{x}_s \mathbf{x}_s^\top \ge c\mathbf{I}_d$ holds easily for sufficiently large T, with the D-optimal design exploration in the second batch, because features of arms used in D-optimal design exploration spans $\mathbb{R}^d$ (Lattimore & Szepesvári, 2020).

Now, all that's left to do is to calculate the stopping statistic $Z(b_2)$ defined in (4.3) and to show it is larger than the threshold value $\beta(b_2, 1/T)$ defined in (4.4) as $T \to \infty$.

After the first batch, the agent calculates the allocation $\mathbf{w}(\hat{\Delta}) = \{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ according to Definition 4.3, where we use $\hat{\Delta}$ to denote estimated gaps based on data in the first batch. Correspondingly, we define $\mathbf{w}(\Delta) = \{w^*_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ to denote the proportion solved from Definition C.5 using the true instance parameter $\boldsymbol{\theta}^*$. Under the concentration event $\mathcal{E}_2$ and Lemma C.6, we know $\lim_{T\to\infty} \hat{\Delta}_{\mathbf{x}} = \Delta_{\mathbf{x}}$, $\lim_{T\to\infty} w_{\mathbf{x}} = w^*_{\mathbf{x}}, \forall \mathbf{x} \in \mathcal{X}$.

Note that when $\ell = 2$, since $w_{\mathbf{x}}^*$ for $\mathbf{x} \in \mathcal{X}^-$ is independent of T,

$$\lim_{T\to\infty} \frac{w_{\mathbf{x}} \cdot \alpha \log T}{(\log T)^{1+\gamma}} = \lim_{T\to\infty} \frac{w_{\mathbf{x}}^* \cdot \alpha \log T}{(\log T)^{1+\gamma}} = 0, \mathbf{x} \in \mathcal{X}^-.$$

Thus for sufficiently large T,

$$w_{\mathbf{x}} \cdot \alpha \log T \le (\log T)^{1+\gamma}, \mathbf{x} \in \mathcal{X}^-.$$

Besides, by (C.3) we know $w_{\mathbf{x}^*} = (\log T)^\gamma / \alpha$. It follows that

$$w_{\mathbf{x}^*} \cdot \alpha \log T = (\log T)^{1+\gamma}.$$

Therefore, the exploration rule in the second batch should be D-optimal design exploration with rate T_2, then pull each arm $\mathbf{x} \in \mathcal{X}$ for $w_{\mathbf{x}} \cdot \alpha \log T$ times. Here we use b_2 to denote the batch size of the second batch, $a_2 = \sum_{\mathbf{x}\in\mathcal{A}} f(\mathbf{x}, \mathcal{A}, T_2) = \Theta\big((\log T)^{1/2}\big)$ to denote the total number of D-optimal design exploration steps in this batch given by Definition 4.1. Consequently, we know

$$b_2 = a_2 + (\log T)^{1+\gamma} + \sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}} \cdot \alpha \log T. \tag{D.3}$$

Likewise, we can split the data $\{\mathbf{x}_s\}_{s=1}^{b_2}$ into $\{\mathbf{x}_s\}_{s=1}^{a_2}$ and $\{\mathbf{x}_s\}_{s=a_2+1}^{b_2}$, with the former one denoting the arms in D-optimal design exploration and the latter one denoting the remaining arms according to the allocation $\mathbf{w}$ in the second batch.

By the constraint condition in the convex program of Definition 4.3, we have

$$\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{H}_w^{-1}}^2 = \|\mathbf{x} - \hat{\mathbf{x}}^*\|_{\mathbf{H}_w^{-1}}^2 \le \frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2}{2}, \forall \mathbf{x} \in \mathcal{X}^-.$$

Then we get

$$\frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2/2}{\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{H}_w^{-1}}^2} = \frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2/2}{\|\mathbf{x} - \hat{\mathbf{x}}^*\|_{\mathbf{H}_w^{-1}}^2} \ge 1, \forall \mathbf{x} \in \mathcal{X}^-. \tag{D.4}$$

In the second batch of exploration, the agent collects data and estimates the parameters. To avoid confusion, here we use $\bar{\boldsymbol{\theta}}$, $\bar{\Delta}$ and $\bar{\mathbf{x}}^*$ to denote related estimators based on data in the second batch, and they have the same concentration properties with estimators before. Namely, $\lim_{T\to\infty} \bar{\Delta} = \Delta$ and $\bar{\mathbf{x}}^* = \mathbf{x}^*$. And similar to (D.2), we have

$$|\bar{\Delta}_{\mathbf{x}} - \Delta_{\mathbf{x}}| \le 2\epsilon, \forall \mathbf{x} \in \mathcal{X}^-.$$

Further, combining this with (D.2), we have for sufficiently large T,

$$\bar{\Delta}_{\mathbf{x}} \ge \hat{\Delta}_{\mathbf{x}} - 4\epsilon \ge 0, \forall \mathbf{x} \in \mathcal{X}^-, \tag{D.5}$$

where the second inequality is because $\lim_{T\to\infty} \hat{\Delta}_{\mathbf{x}} = \Delta_{\mathbf{x}} > 0, \forall \mathbf{x} \in \mathcal{X}^-$ and $\lim_{T\to\infty} \epsilon = \lim_{T\to\infty} 1/\log\log T = 0$. We can calculate the stopping statistic given by (4.3):

$$Z(b_2) = \min_{\mathbf{x}\neq\bar{\mathbf{x}}^*} \frac{\bar{\Delta}_{\mathbf{x}}^2}{2\|\mathbf{x} - \bar{\mathbf{x}}^*\|_{\mathbf{H}_{b_2}^{-1}}^2} = \min_{\mathbf{x}\neq\mathbf{x}^*} \frac{\bar{\Delta}_{\mathbf{x}}^2}{2\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{H}_{b_2}^{-1}}^2}.$$

Here we get

$$\frac{Z(b_2)}{\alpha \log T} = \frac{\min_{\mathbf{x}\neq\mathbf{x}^*} \frac{\bar{\Delta}_{\mathbf{x}}^2}{2\|\mathbf{x}-\mathbf{x}^*\|_{\mathbf{H}_{b_2}^{-1}}^2}}{\alpha \log T}$$

$$\begin{aligned}
&\geq \min_{\mathbf{x}\neq\mathbf{x}^*} \frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2}{2\cdot\alpha\log T\|\mathbf{x}-\mathbf{x}^*\|^2_{\mathbf{H}_{b_2}^{-1}}} && \text{(D.6)}\\
&= \min_{\mathbf{x}\neq\mathbf{x}^*} \frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2}{2\big\|\mathbf{x}-\mathbf{x}^*\big\|^2_{\left(\mathbf{H}_{b_2}/(\alpha\log T)\right)^{-1}}}\\
&\geq \min_{\mathbf{x}\neq\mathbf{x}^*} \frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2}{2\|\mathbf{x}-\mathbf{x}^*\|^2_{\mathbf{H}_w^{-1}}}, && \text{(D.7)}
\end{aligned}$$

where (D.6) is because of (D.5), and (D.7) holds because

$$\begin{aligned}
\frac{\mathbf{H}_{b_2}}{\alpha\log T} &= \frac{\sum_{s=1}^{b_2}\mathbf{x}_s\mathbf{x}_s^\top}{\alpha\log T}\\
&= \frac{1}{\alpha\log T}\left(\sum_{s=1}^{a_2}\mathbf{x}_s\mathbf{x}_s^\top + \sum_{s=a_2+1}^{b_2}\mathbf{x}_s\mathbf{x}_s^\top\right)\\
&\geq \frac{1}{\alpha\log T}\sum_{s=a_2+1}^{b_2}\mathbf{x}_s\mathbf{x}_s^\top\\
&= \frac{1}{\alpha\log T}\left((\log T)^{1+\gamma}\mathbf{x}^*{\mathbf{x}^*}^\top + \sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}}\alpha\log T\cdot\mathbf{x}_s\mathbf{x}_s^\top\right)\\
&= (\log T)^\gamma/\alpha\cdot\mathbf{x}^*{\mathbf{x}^*}^\top + \sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}}\cdot\mathbf{x}\mathbf{x}^\top\\
&= \mathbf{H}_{\mathbf{w}}.
\end{aligned}$$

Then we can combine (D.4) and (D.7) to get

$$\frac{Z(b_2)}{\alpha\log T} \geq \min_{\mathbf{x}\neq\mathbf{x}^*} \frac{(\hat{\Delta}_{\mathbf{x}} - 4\epsilon)^2}{2\|\mathbf{x}-\mathbf{x}^*\|^2_{\mathbf{H}_w^{-1}}} \geq 1. \tag{D.8}$$

By the selection of $\alpha = (1 + 1/\log\log T)(1 + d\log\log T/\log T)$ in Theorem 5.4, we have

$$\alpha\log T = (1 + 1/\log\log T)(1 + d\log\log T/\log T)\log T = (1 + 1/\log\log T)\log\left(\frac{(\log T)^d}{1/T}\right). \tag{D.9}$$

Besides, for sufficiently large T, by Lemma C.6 we have

$$\sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}} \leq 2\cdot\sum_{\mathbf{x}\in\mathcal{X}^-} w^*_{\mathbf{x}}.$$

Combining this with (D.3), we get

$$\begin{aligned}
b_2 &= a_2 + (\log T)^{1+\gamma} + \sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}}\cdot\alpha\log T\\
&\leq a_2 + (\log T)^{1+\gamma} + 2\sum_{\mathbf{x}\in\mathcal{X}^-} w^*_{\mathbf{x}}\cdot\alpha\log T\\
&= \Theta\big((\log T)^{1/2}\big) + (\log T)^{1+\gamma} + 2\sum_{\mathbf{x}\in\mathcal{X}^-} w^*_{\mathbf{x}}\cdot\alpha\log T\\
&= \Theta\big((\log T)^{1+\gamma}\big),
\end{aligned}$$

since $\{w^*_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}^-}$ is determined by the bandit instance which is independent of T. Hence for sufficiently large T, we have $b_2 \leq U(\log T)^{1+\gamma}$ for some constant number $U > 0$. Then, putting this upper bound of b_2 into the the expression of $\beta(b_2, 1/T)$ from (4.4), we get

$$\beta(b_2, 1/T) = (1 + 1/\log\log T)\log\left(\frac{(b_2\log\log T)^{d/2}}{1/T}\right)$$

$$\leq (1 + 1/\log\log T)\log\left(\frac{(U\log\log T)^{d/2}(\log T)^{d(1+\gamma)/2}}{1/T}\right) \tag{D.10}$$

for some constant number $U > 0$. Combining (D.8), (D.9), (D.10), we have for sufficiently large T,

$$\begin{aligned} Z(b_2) &\geq \alpha \log T \\ &= (1 + 1/\log\log T)\log\left(\frac{(\log T)^d}{1/T}\right) \\ &\geq (1 + 1/\log\log T)\log\left(\frac{(U\log\log T)^{d/2}(\log T)^{d(1+\gamma)/2}}{1/T}\right) \\ &\geq \beta(b_2, 1/T), \end{aligned}$$

which implies $Z(b_2) \geq \beta(b_2, 1/T)$ for sufficiently large T.

So we prove that if $\mathcal{E}_2$ holds, for sufficiently large T, the stopping condition (4.2) holds in the second batch. In addition, by Lemma C.4, $\mathcal{E}_2$ happens with probability at least $1 - 2/(\log T)^2$. Now it can be concluded that the Chernoff's stopping rule in (4.2) holds with probability at least $1 - 2/(\log T)^2$

D.5. Proof of Lemma C.8

In this proof we continue using the good event $\mathcal{E}_0$ (C.4) defined before. Since it happens with probability at least $1 - 1/T$, we can just assume it happens, then prove the elimination fact.

Under the good event $\mathcal{E}_0$ (C.4), we have shown in (C.5) that arm $\mathbf{x} \neq \mathbf{x}^*$ with gap

$$\Delta_{\mathbf{x}} > 4\varepsilon_3 = 4\sqrt{d\log(KT^2)/(\log T)^{1+\gamma}} \tag{D.11}$$

would be eliminated after the third batch. As $T \to \infty$, (D.11) holds for all suboptimal arms $\mathbf{x} \neq \mathbf{x}^*$, since $\Delta_{\min} > 0$ and $\varepsilon_3 \to 0$ as $T \to \infty$. So we finish our proof.

D.6. Proof of Lemma C.9

With $\mathcal{E}_2$ defined in (C.2), we can decompose the regret of the second batch by

$$Regret_2 = Regret_{2\mathcal{E}_2}\mathbb{P}(\mathcal{E}_2) + Regret_{2\mathcal{E}_2^c}\mathbb{P}(\mathcal{E}_2^c),$$

where we let $Regret_{2\mathcal{E}_2}$ and $Regret_{2\mathcal{E}_2^c}$ be the regret of the second batch given $\mathcal{E}_2$ and $\mathcal{E}_2^c$, respectively. Lemma C.4 shows $\mathbb{P}(\mathcal{E}_2) \geq 1 - 2/(\log T)^2$. In $\mathcal{E}_2^c$, the regret is negligible for this $O\big((\log T)^{1+\gamma}\big)$-size batch as $T \to \infty$, because

$$\lim_{T\to\infty} Regret_{2\mathcal{E}_2^c}\mathbb{P}(\mathcal{E}_2^c) = \lim_{T\to\infty}\frac{2}{(\log T)^2}\cdot O\big((\log T)^{1+\gamma}\big) = 0.$$

This implies

$$\begin{aligned} \limsup_{T\to\infty}\frac{Regret_2}{\log T} &= \limsup_{T\to\infty}\frac{Regret_{2\mathcal{E}_2}\mathbb{P}(\mathcal{E}_2) + Regret_{2\mathcal{E}_2^c}\mathbb{P}(\mathcal{E}_2^c)}{\log T} \\ &= \limsup_{T\to\infty}\frac{Regret_{2\mathcal{E}_2}\mathbb{P}(\mathcal{E}_2)}{\log T} \\ &\leq \limsup_{T\to\infty}\frac{Regret_{2\mathcal{E}_2}}{\log T}. \end{aligned}$$

Moreover, when $\mathcal{E}_2$ appears, we can use the continuity of the problem to end the proof.

In fact,when the concentration result $\mathcal{E}_2$ holds, for all arms $\mathbf{x}$, $|\hat{\mu}_{\mathbf{x}} - \mu_{\mathbf{x}}| \leq \epsilon = 1/\log\log T$. So $\lim_{T\to\infty}\hat{\mu}_{\mathbf{x}} = \mu_{\mathbf{x}}$. In a similar way, $\lim_{T\to\infty}\hat{\Delta}_{\mathbf{x}} = \lim_{T\to\infty}(\hat{\mu}_{\mathbf{x}^*} - \hat{\mu}_{\mathbf{x}}) = \mu_{\mathbf{x}^*} - \mu_{\mathbf{x}} = \Delta_{\mathbf{x}}$. Then using Lemma C.6, we get $\lim_{T\to\infty} w_{\mathbf{x}} = w^*_{\mathbf{x}}$ for all $\mathbf{x} \in \mathcal{X}$, where we use $\{w^*_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ to denote $\mathbf{w}(\Delta)$ solved using the true gaps defined in Definition C.5, and $\{w_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}}$ to denote $\mathbf{w}(\hat{\Delta})$ solved using estimated gaps defined in Definition 4.3. As we discussed in (D.3), for sufficiently large T, in the second batch the agent first explores with a D-optimal design multi-set with size $o(\log T)$, then pull the best arm for $(\log T)^{1+\gamma}$ times and each suboptimal arm $\mathbf{x} \in \mathcal{X}$ for $w_{\mathbf{x}} \cdot \alpha \log T$ times. Thus the regret is

$$\begin{aligned}
\limsup_{T\to\infty} \frac{Regret_2}{\log T} &\leq \limsup_{T\to\infty} \frac{Regret_{2\mathcal{E}_2}}{\log T} \\
&= \limsup_{T\to\infty} \frac{o(\log T) + \sum_{\mathbf{x}\in\mathcal{X}^-} \alpha w_{\mathbf{x}} \log T \Delta_{\mathbf{x}}}{\log T} \\
&= \limsup_{T\to\infty} \frac{\sum_{\mathbf{x}\in\mathcal{X}^-} \alpha w_{\mathbf{x}} \log T \Delta_{\mathbf{x}}}{\log T} \\
&= \sum_{\mathbf{x}\in\mathcal{X}^-} w_{\mathbf{x}}^* \Delta_{\mathbf{x}} = c^*.
\end{aligned}$$

This completes the proof.