# Almost Optimal Anytime Algorithm for Batched Multi-Armed Bandits

Tianyuan Jin [1] Jing Tang [2] Pan Xu [3] Keke Huang [1] Xiaokui Xiao [1] Quanquan Gu [3]

## Abstract

In batched multi-armed bandit problems, the learner can adaptively pull arms and adjust strategy in batches. In many real applications, not only the regret but also the batch complexity need to be optimized. Existing batched bandit algorithms usually assume that the time horizon $T$ is known in advance. However, many applications involve an unpredictable stopping time. In this paper, we study the anytime batched multi-armed bandit problem. We propose an anytime algorithm that achieves the asymptotically optimal regret for exponential families of reward distributions with $\mathcal{O}(\log\log T \cdot \mathrm{ilog}^{\alpha}(T))$[1] batches, where $\alpha \in \mathcal{O}_T(1)$. Moreover, we prove that for any constant $c > 0$, no algorithm can achieve the asymptotically optimal regret within $c \log\log T$ batches.

## 1 Introduction

The multi-armed bandit (MAB) problem provides an elementary model for exploration-exploitation tradeoffs and finds many real applications such as medical trials (Thompson, 1933; Perchet et al., 2016), crowdsourcing (Kittur et al., 2008; Zhou et al., 2014), and marketing (Bertsimas & Mersereau, 2007; Vaswani et al., 2017). The problem is typically described as a game between the agent and the environment. The game proceeds in a total of $T$ time steps. At each time step $t$, the agent pulls an arm $A_t$ from the arm set $[K]$ with the goal of maximizing the accumulated reward over $T$ time steps. Ideally, the agent can observe the immediate feedback of each pull, e.g., reward, and exploit it to guide the next action. However, this is impractical for many real applications where the number of interactions between the agent and environment is limited. For example, in clinical trials, typically, it takes some time to test the efficacy of a treatment on a patient. It is thus computationally prohibitive to conduct the experiments in fully sequential. Instead, patients are usually grouped into batches, and each batch of patients are tested in a parallel manner. In such a case, the outcomes are unavailable till the end of each batch. As another example, in online advertising, the agent cannot immediately update her strategy upon receiving the feedback, since there may be a large amount of responses in every second.

Perchet et al. (2016) modeled the above problem as the batched multi-armed bandit problem. In such problems, the time horizon $T$ is split into a small number of batches, and the outcomes are only revealed at the end of each batch. Previous batched bandit algorithms (Perchet et al., 2016; Gao et al., 2019; Esfandiari et al., 2019; Jin et al., 2020) all assume that the time $T$ is known in advance. However, many real-world applications involve an unpredictable stopping time. Again, consider the clinical trials example, $T$ may correspond to the number of participated patients in the test for a certain period; and in the online advertisement example, $T$ may correspond to the number of visitors of a website for a certain period. In both cases, designing an anytime bandit algorithm is imperative.

Motivated by the above observations, in this paper, we study the anytime batched multi-armed bandit problem, where the horizon length $T$ is unknown ahead of time. In particular, we have a set $[K] = \{1, 2, \ldots, K\}$ of $K$ arms, where each arm $i$ is associated with a reward distribution of some canonical one-dimensional exponential family with mean $\mu_i$. We assume the best arm is unique. Without loss of generality, we assume that arm 1 has the maximum expected reward throughout the paper, i.e., $\mu_1 > \mu_i$ for any $i \in [K]/\setminus\{1\}$. The pulls of each arm yield rewards which are independent and identically distributed (i.i.d.) samples from the arm's distribution. Furthermore, the time horizon $T$ is divided into batches represented by a grid $\mathcal{T} = \{t_1, t_2, \ldots\}$, which means after $j$-th batch, the total number of pulls of all arms reaches $t_j$. In this paper, we study the static grid setting (Perchet et al., 2016; Gao et al., 2019), i.e., $t_1, t_2, \ldots$ are predefined numbers. At each time step $t$, there exists a

[1]School of Computing, National University of Singapore, Singapore [2]Data Science and Analytics Thrust, The Hong Kong University of Science and Technology [3]Department of Computer Science, University of California, Los Angeles, CA 90095, USA. Correspondence to: Xiaokui Xiao <xkxiao@nus.edu.sg>, Quanquan Gu <qgu@cs.ucla.edu>.

*Proceedings of the $38^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. 

[1]Notation $\mathrm{ilog}^{\alpha}(T)$ is the result of iteratively applying the logarithm function on $T$ for $\alpha$ times, e.g., $\mathrm{ilog}^{3}(T) = \log\log\log T$.

unique $j$ such that $t_{j-1} < t \leq t_j$, and the agent makes a decision on pulling arm $A_t$ based on all the outcomes up to time $t_{j-1}$. The ultimate goal is to minimize the regret, which is defined as the expected cumulative difference between playing the best arm and playing the arm according to the strategy. The formal definition is given as follows.

$$R_T = T \cdot \mu_1 - \mathbb{E}\bigg[\sum_{t=1}^{T} \mu_{A_t}\bigg].$$

Lai & Robbins (1985) shows that for distributions that are continuously parameterized by their means,

$$\liminf_{T\to\infty} \frac{R_T}{\log T} \geq \sum_{i\in[K]\setminus\{1\}} \frac{\Delta_i}{\mathrm{kl}(\mu_i, \mu_1)}, \tag{1}$$

where $\Delta_i := \mu_1 - \mu_i$ and $\mathrm{kl}(\mu, \mu')$ is the Kullback-Leibler divergence between two distributions with mean $\mu$ and mean $\mu'$. We refer to $\lim_{T\to\infty} \frac{R_T}{\log T}$ as the asymptotic regret rate, and say that the algorithm is asymptotically optimal if its asymptotic regret rate matches the right hand side of (1).

The well-known algorithms such as KL-UCB (Garivier & Cappé, 2011) and Thompson Sampling (Korda et al., 2013) are shown to be asymptotically optimal in anytime setting. Nevertheless, these algorithms are fully sequential, which require $\mathcal{O}(T)$ batches in batched bandits. A very recent work (Jin et al., 2020) proposes asymptotically optimal algorithms for the 2-armed bandit problem with sub-Gaussian rewards, requiring $\mathcal{O}(1)$ expected batches if $T$ is known. However, in the anytime setting, their algorithm needs at least $\Omega(\log T)$ batches even for 2-armed bandits. Therefore, a natural question is:

*How many batches are needed for anytime $K$-armed bandit algorithms to achieve the asymptotically optimal regret?*

On the other hand, Besson & Kaufmann (2018) conjectured that no anytime algorithm can achieve the asymptotically optimal regret with the exponential time grid (i.e., $t_i = a^{b^i}$ for some constants $a$ and $b$) incurring $\mathcal{O}(\log\log T)$ batches. However, confirming this conjecture theoretically still remains an open problem. This gives rise to another question:

*What is the fundamental limit in batch complexity of anytime $K$-armed bandit algorithms for achieving the asymptotically optimal regret?*

In this paper, we answer the above two questions through the lens of both the upper bound and the lower bound of batch complexities for anytime $K$-armed bandit algorithms that achieve the asymptotically optimal regret.

**Contributions.** Our results can be summarized as follows:

- **(Upper Bound)** We propose an anytime algorithm BABA for batched multi-armed bandits where the reward distributions are from exponential families. We prove that BABA is asymptotically optimal and only requires $\mathcal{O}(\log\log T \cdot \mathrm{ilog}^{\alpha}(T))$ batches, where $\alpha \in \mathcal{O}_T(1)$ is a constant and $\mathrm{ilog}^{\alpha}(T)$ iteratively applies the logarithm function on $T$ for $\alpha$ times.
- **(Lower Bound)** We prove that in the anytime setting, for any positive constant $c$, no bandit algorithm can achieve the asymptotic optimality within $c \log\log T$ batches. This the first lower bound for anytime batched bandit algorithms in the literature. Our lower bound is almost tight since it matches the upper bound of our BABA algorithm up to an iterative logarithm factor.
- We empirically evaluate our proposed algorithm and show that it enjoys comparable performance with the fully sequential algorithm KL-UCB in terms of regret while requiring significantly fewer batches.

## 2 Preliminaries

In this section, we first review the previous work related to ours. We then introduce the definition of exponential families and some useful properties. Finally, we give some notations that will be frequently used.

### 2.1 Previous Results

**MAB.** The MAB problem provides an elementary model for an class of sequential optimization problems, which has been extensively studied since the seminal work by Thompson (1933). In the fully adaptive setting, a large body of research has analyzed the regret (Audibert & Bubeck, 2009; Garivier & Cappé, 2011; Korda et al., 2013; Degenne & Perchet, 2016; Agrawal & Goyal, 2017; Kaufmann et al., 2018; Lattimore, 2018). We refer interested readers to the book by Lattimore & Szepesvári (2020) for a comprehensive introduction of bandit algorithms and various applications of MAB.

**Batched MAB.** Cesa-Bianchi et al. (2013) studied the batched bandit problem under the name of switching cost and show that $\mathcal{O}(\log\log T)$ batches are sufficient for achieving the near-optimal *minimax* regret bound. Perchet et al. (2016) further prove that such batch complexity is sufficient and necessary for achieving the optimal minimax regret for the 2-armed bandit problem, which is later generalized to the $K$-armed case (Gao et al., 2019). Perchet et al. (2016); Gao et al. (2019); Esfandiari et al. (2019) show that $\mathcal{O}(\log T)$ batches are sufficient for achieving the near-optimal instance-dependent regret bound. However, there exists a multiplicative constant between their regret bounds and the optimal bound, which makes their algorithms sub-optimal in the asymptotic sense. Jin et al. (2020) propose DETC, consisting of two exploration and two exploitation stages, that can achieve the asymptotic optimality with $\mathcal{O}(1)$ expected batches for the 2-armed case with sub-Gaussian rewards. However, the generalization to $K$-armed bandit and exponential families of reward distributions is non-trivial and unclear. More importantly, all the aforementioned studies assume that $T$ is known in advance, while we consider anytime batched MAB with unpredictable stopping time $T$.

**Anytime Algorithm.** An effective technique to construct an anytime algorithm from a non-anytime algorithm is the *doubling trick* strategy (Auer et al., 1995). At the $i$-th round/epoch, the doubling trick strategy guesses $T = a^i$ (referred to as geometric doubling trick) or $T = a^{b^i}$ (referred to as exponential doubling trick). For geometric doubling trick, it costs at least $\log T$ batches. For example, the anytime version of DETC requires $\mathcal{O}(\log T)$ exploration and exploitation rounds by guessing $T = 2^i$ at the $i$-th round, and each round takes $\Omega(1)$ batches. For exponential doubling trick with $\mathcal{O}(\log\log T)$ batches, Besson & Kaufmann (2018) conjecture that it cannot achieve the asymptotically optimal regret, which is confirmed by our lower bound in this paper. Motivated by the deficiencies of the above two doubling tricks, we present a new trick strategy that is asymptotically optimal and takes $\mathcal{O}(\log\log T \cdot \mathrm{ilog}^{\alpha}(T))$ batches. Compared with the existing tricks, the number of batches incurred by our trick is significantly smaller than geometric doubling trick and is slightly larger than exponential doubling trick.

### 2.2 Exponential Families

An exponential family is a parametric set of probability distributions $\{\nu_\theta \colon \theta \in \Theta\}$ dominated by a measure $\rho$ on $\mathbb{R}$, with density given by

$$\frac{\mathrm{d}\nu_\theta}{\mathrm{d}\rho}(x) = \exp(x\theta - b(\theta)),$$

where $b(\theta) = \log \int_{\mathbb{R}} e^{x\theta}\mathrm{d}\rho(x)$ and $\Theta = \{\theta \in \mathbb{R} \colon b(\theta) < \infty\}$. Exponential families have the following properties.

$$b'(\theta) = \mathbb{E}[\nu_\theta] \quad \textit{and} \quad 0 < b''(\theta) = \mathrm{Var}(\nu_\theta),$$

where $b'(\theta)$ and $b''(\theta)$ are the first derivative and second derivative of $b(\theta)$ with respect to $\theta$, respectively. A direct computation gives the Kullback-Leibler (KL) divergence as

$$\mathrm{KL}(\nu_\theta, \nu_{\theta'}) = b(\theta') - b(\theta) - b'(\theta)(\theta' - \theta).$$

Let $\mu = b'(\theta)$ and $\mathrm{kl}(\mu, \mu') := \mathrm{KL}(\nu_\theta, \nu_{\theta'})$. In this paper, we assume the variance is bounded, i.e.,

$$0 < b''(\theta) \le V < +\infty.$$

We have the following property on the KL divergence.

**Proposition 1.** *For all $\mu$ and $\mu'$, we have*

$$\mathrm{kl}(\mu, \mu') \ge (\mu - \mu')^2/(2V). \tag{2}$$

*In addition, for $\epsilon > 0$ and $\mu \le \mu' - \epsilon$, we can obtain that*

$$\begin{aligned} &\mathrm{kl}(\mu, \mu') \ge \mathrm{kl}(\mu, \mu' - \epsilon), \\ \textit{and} \quad &\mathrm{kl}(\mu, \mu') \le \mathrm{kl}(\mu - \epsilon, \mu'). \end{aligned} \tag{3}$$

---

**Algorithm 1:** Batched Anytime Bandit Alg. (BABA)

---

**Input:** a set of $K$ arms and parameters $\alpha$ and $I_1$
initialize $t \leftarrow 0$, $r \leftarrow 1$, $c_0 \leftarrow 1$;
**while** *experiment proceeds* **do**
  **Step I:** perform UNIFORMEXPLORATION;
  **Step II:** perform INITIALEXPLOITATION;
  **Step III:** perform OPTIMISTICEXPLORATION;
  **Step IV:** perform CONFIDENTEXPLORATION;
  **Step V:** perform CONFIDENTEXPLOITATION;
  $r \leftarrow r + 1$;
  $I_r \leftarrow f(I_{r-1})$;

---

Interested readers are referred to Appendix A for the proof of Proposition 1.

Exponential families include many of the most common distributions, such as Gaussian, Bernoulli, exponential, etc. In particular, for Gaussian distribution with known variance $\sigma^2$ by choosing $V = \sigma^2$, $\mathrm{kl}(\mu, \mu') = (\mu - \mu')/(2\sigma^2)$; for Bernoulli distribution by choosing $V = 1/4$, $\mathrm{kl}(\mu, \mu') = \mu\log(\mu/\mu') + (1-\mu)\log\big((1-\mu)/(1-\mu')\big)$; and for exponential distribution with known parameter $\lambda$ by choosing $V = 1/\lambda^2$, $\mathrm{kl}(\mu, \mu') = \log(\mu) - \log(\mu') + \mu'/\mu - 1$.

### 2.3 Notations

Denote by $\mathrm{ilog}^m(x)$ the result of iteratively applying the logarithm function on $x$ for $m$ times, i.e., $\mathrm{ilog}^m(x) = \max\big\{\log \mathrm{ilog}^{m-1}(x), 0\big\}$ for any $x > 0$ and $m \in \mathbb{N}$. We also define $\mathrm{ilog}^0(x) = x$. We define $\mathrm{kl}_+(p, q) := \mathrm{kl}(p, q)\mathbb{1}(p \le q)$, where $\mathbb{1}(\cdot)$ is the indicator function. We use $\Delta_i$ to denote the gap between arm 1 and arm $i$, i.e., $\Delta_i = \mu_1 - \mu_i$. Let $\widehat{\mu}_i(t)$ be the average reward of arm $i$ at time $t$ and $\widehat{\mu}_{i,s}$ be the average reward of arm $i$ after its $s$'s pull. Let $T_i(t)$ be the number of pulls of arm $i$ at time step $t$, i.e., $T_i(t) = \sum_{\ell=1}^{t} \mathbb{1}(A_\ell = i)$. Throughout the paper, we adopt the standard asymptotic notations. In particular, we use $f(\cdot) \lesssim g(\cdot)$ to denote $f(\cdot) = \mathcal{O}_T(g(\cdot))$ and $f(\cdot) \gtrsim g(\cdot)$ to denote $f(\cdot) = \Omega_T(g(\cdot))$.

## 3 The Proposed Algorithm

### 3.1 Overview

Algorithm 1 presents the framework of our algorithm, referred to as BABA. In particular, our algorithm guesses $T$ in epochs and proceeds in five batches for each epoch.

In the following, we first introduce two core functions that are essential for constructing our time grid $\mathcal{T}$.

$$f(x) = \max\{\lceil x^{1+1/(1+\mathrm{ilog}^{\alpha} x)}\rceil, 2x\}, \tag{4}$$
$$g(x) = \lceil \log x / \log\log x \rceil, \tag{5}$$

where $\alpha \ge 3$ and $\alpha \in \mathcal{O}_T(1)$ is a constant. That is, at the $r$-th epoch, we guess $T = f^{(r)}(I_1)$, where $f^{(r)}(\cdot)$ iteratively

**Algorithm 2:** UNIFORMEXPLORATION

1 **for** $i = 1, 2, \dots, K$ **do**
2     **while** $T_i(t) \leq g(I_r)$ **do**
3         pull arm $i$;
4         $t \leftarrow t + 1$;
5 **while** $t \leq I_{r-1} + K \cdot g(I_r)$ **do**
6     pull arm $c_{r-1}$;
7     $t \leftarrow t + 1$;

**Algorithm 3:** INITIALEXPLOITATION

1 $a_{1,r} \leftarrow \arg\max_{i \in [K]} \widehat{\mu}_{i, g(I_r)}$;
2 $\ell \leftarrow 1$;
3 **while** $\ell \leq \log^2 I_r$ **do**
4     pull arm $a_{1,r}$;
5     $t \leftarrow t + 1, \ell \leftarrow \ell + 1$;

applies function $f$ for $r$ times and $I_1$ is an input parameter satisfying

$$Kg(I_1) + (2K + 1) \log^2 I_1 < I_1. \tag{6}$$

Let $I_r = f^{(r)}(I_1)$. At the $r$-th epoch, the first four batches pull the arms exactly $Kg(I_r)$, $\log^2 I_r$, $K \cdot \log^2 I_r$ and $K \cdot \log^2 I_r$ times (for ease of presentation, we assume $\log I_r \in \mathbb{N}^+$), respectively, while the fifth batch pulls the arms until a total number of $I_r$ times is pulled. Our time grid $\mathcal{T}$ is given as $\mathcal{T} = \{t_{1,1}, \dots, t_{5,1}, t_{1,2}, \dots, t_{5,2}, \dots\}$, where $t_{j,r}$ denote the checkpoint at $j$-th step of the $r$-th epoch for any $j \in [5]$, defined as follows:

$$\begin{aligned} t_{1,r} &= I_{r-1} + Kg(I_r), \\ t_{2,r} &= I_{r-1} + Kg(I_r) + \log^2 I_r, \\ t_{3,r} &= I_{r-1} + Kg(I_r) + (K+1) \log^2 I_r, \\ t_{4,r} &= I_{r-1} + Kg(I_r) + (2K+1) \log^2 I_r, \\ \text{and} \quad t_{5,r} &= I_r. \end{aligned}$$

It is trivial to see that $\mathcal{T}$ is a static time grid. Note that the trick $\{f^{(r)}(I_1)\}_{r \geq 1}$ grows (i) faster than geometric doubling trick which results in far less number of batches than $\log T$, but (ii) a litter bit slower than exponential doubling trick which ensures the asymptotically optimal regret.

### 3.2 Detailed Design

Next, we elaborate the details of the five steps in each epoch.

**Step I.** UNIFORMEXPLORATION (Algorithm 2) shows the first step, which pulls the arms a total of $Kg(I_r)$ times. Specifically, at the $r$-th epoch, for every arm $i$ that is pulled less than $g(I_r)$ times, we pull arm $i$ till reaching a total of $g(I_r)$ times (Lines 1–4). In addition, we pull the "best arm" $c_{r-1}$ found after the $(r-1)$-th epoch until the total number of pulls of all arms reaches $I_{r-1} + Kg(I_r)$ so that this batch pulls the arms $Kg(I_r)$ times (Lines 5–7).

**Algorithm 4:** OPTIMISTICEXPLORATION

1 $s_1 \leftarrow T_{a_{1,r}}(t)$;
2 $\{a_{2,r}, \dots, a_{K,r}\} \leftarrow [K] \setminus \{a_{1,r}\}$;
3 **for** $i = 2, 3, \dots, K$ **do**
4     **while** $T_{a_{i,r}}(t) \leq \min\{\delta_{i,r}, \log^2 I_r\}$ **do**
5         pull arm $a_{i,r}$;
6         $t \leftarrow t + 1$;
7 $\mathcal{F} \leftarrow \perp$;
8 **for** $i = 2, 3, \dots, K$ **do**
9     $s_i \leftarrow \min\{\log^2 I_r, T_{a_{i,r}}(t)\}$;
10     **if** $\mathrm{kl}_+(\widehat{\mu}_{a_{i,r}, s_i}, \widehat{\mu}_{a_{1,r}, s_1}) < \frac{\log(I_r \cdot \log^2 I_r)}{s_i}$ **then**
11         $\mathcal{F} \leftarrow \top$;
12         **break**;
13 **while** $t \leq I_{r-1} + K \cdot g(I_r) + \log^2 I_r + K \cdot \log^2 I_r$ **do**
14     pull arm $c_{r-1}$;
15     $t \leftarrow t + 1$;

**Purpose.** Let $a_{1,r}$ be the arm with the largest average reward when every arm is pulled exactly $g(I_r)$ times, which is likely to be the best arm. In fact, we will show that $\mathbb{P}(a_{1,r} = 1) \geq 1 - 1/\log^2 I_r$, which supports us to pull arm $a_{1,r}$ additional $\log^2 I_r$ times while keeping the optimal asymptotic regret.

**Step II.** INITIALEXPLOITATION (Algorithm 3) shows the second step, which simply pulls $\log^2(I_r)$ times of arm $a_{1,r}$.

**Purpose.** When $a_{1,r}$ is pulled $\log^2 I_r$ times, the sample average of arm $a_{1,r}$ will concentrate on its true mean. This ensures that when we explore whether other arms have the potential to be the best arm, we do not pull $a_{1,r}$ as its estimated mean is sufficiently accurate.

**Step III.** OPTIMISTICEXPLORATION (Algorithm 4) shows the third step, which pulls the arms $K \log^2 I_r$ times. Define $\{a_{i,r}\}_{i \geq 2} := [K] \setminus \{a_{1,r}\}$ as the set of other arms except for $a_{1,r}$. Let $\epsilon_r := 1/\log\log I_r$. For $i \geq 2$, define

$$\delta_{i,r} := \frac{\log(I_r \cdot \log^2(I_r))}{\mathrm{kl}(\widehat{\mu}_{a_{i,r}, g(I_r)} + \epsilon_r, \widehat{\mu}_{a_{1,r}, s_1} - \epsilon_r)}, \tag{7}$$

where $s_1$ is the number of pulls of $a_{1,r}$ after Step II. Then, we pull $a_{i,r}$ till $T_{a_{i,r}(t)} \geq \min\{\delta_{i,r}, \log^2 I_r\}$ for $i \geq 2$ (Lines 3–6). We further check the following condition

$$\mathrm{kl}_+(\widehat{\mu}_{a_{i,r}, s_i}, \widehat{\mu}_{a_{1,r}, s_1}) < \frac{\log\left(I_r \log^2 I_r\right)}{s_i}, \tag{8}$$

where $s_i = \min\{\log^2 I_r, T_{a_{i,r}}(t)\}$. If for some $i$, (8) holds, then we set $\mathcal{F} = \top$; otherwise we set $\mathcal{F} = \perp$ (Lines 7–12). Finally, we pull arm $c_{r-1}$ until a total number of $K \cdot \log^2 I_r$ pulls are pulled in this batch (Lines 13–15).

**Algorithm 5:** CONFIDENTEXPLORATION

1 **if** $\mathcal{F} = \perp$ **then**
2 $\quad \ell \leftarrow 1$;
3 $\quad$ **while** $\ell \leq K \cdot \log^2(I_r)$ **do**
4 $\quad\quad$ pull arm $a_{1,r}$;
5 $\quad\quad t \leftarrow t + 1$;
6 $\quad c_r \leftarrow a_{1,r}$;
7 **else**
8 $\quad$ **for** $i = 1, 2, 3, \ldots, K$ **do**
9 $\quad\quad \ell \leftarrow 1$;
10 $\quad\quad$ **while** $\ell \leq \log^2 I_r$ **do**
11 $\quad\quad\quad$ pull arm $i$;
12 $\quad\quad\quad t \leftarrow t + 1$;
13 $\quad c_r \leftarrow \arg\max_{i \in [K]} \widehat{\mu}_i(t)$;

**Purpose.** In this batch, we try to explore whether other arms rather than $a_{1,r}$ have potential to be the best arm. In particular, if $\mathrm{kl}_+(\widehat{\mu}_{a_{i,r},s_i}, \widehat{\mu}_{a_{1,r},s_1})$ is small enough to satisfy (8) for some $i$, we know that $\widehat{\mu}_{a_{i,r},s_i} > \widehat{\mu}_{a_{1,r},s_1}$ or the difference between $\widehat{\mu}_{a_{i,r},s_i}$ and $\widehat{\mu}_{a_{1,r},s_1}$ is very small, which indicates that $a_{i,r}$ has potential to be the best arm. Then, we will further pull each arm in the future to determine whether $a_{i,r}$ is better than $a_{1,r}$. Otherwise if (8) does not hold for every $i$, we will confident that $a_{1,r}$ is the best arm. Finally, we pull best arm $c_{r-1}$ found after the $(r-1)$-th epoch to exhaust the budget of this batch.

**Step IV.** CONFIDENTEXPLORATION (Algorithm 5) shows the fourth step, which pulls the arms a total of $K \log^2 I_r$ times. In particular, if $\mathcal{F} = \perp$, we directly pull arm $a_{1,r}$ a total of $K \log^2(I_r)$ times (Lines 2–5) and update the new best arm $c_r = a_{1,r}$ (Line 6); otherwise, we pull every arm $\log^2(I_r)$ times (Lines 8–12) and update the new best arm $c_r = \arg\max_{i \in [K]} \widehat{\mu}_i(t)$ (Line 13).

**Purpose.** Intuitively, if $\mathcal{F} = \perp$, i.e., (8) fails for all $i \geq 2$, we can show that $\mathbb{P}(a_{1,r} = 1) \geq 1 - 1/I_r$, which ensures that the regret of pulling $a_{1,r}$ additional $I_r$ times is bounded in the optimal range. Hence, if $\mathcal{F} = \perp$, all the budget of $K \cdot (I_r)^2$ in this batch is used on arm $a_{1,r}$ and we set $c_r = a_{1,r}$ for future pulls. On the other hand, if $\mathcal{F} = \top$, the arm $a_{1,r}$ may not be the optimal arm. We then pull every arm $\log^2 I_r$ times and update $c_r$ to the arm with largest average reward.

**Step V.** CONFIDENTEXPLOITATION (Algorithm 6) shows the fifth step, which pulls the "best arm" observed so far until the total number of pulls reaches $I_r$.

**Purpose.** After the first four steps, we are confident now that $c_r$ is the best arm. Thus, we keep pulling $c_r$ to optimize regret until the budget of this batch is exhausted.

**Algorithm 6:** CONFIDENTEXPLOITATION

**while** $t \leq I_r$ **do**
$\quad$ pull arm $c_r$;
$\quad t \leftarrow t + 1$;

## 4 Main Results

Now, we present our main theoretical results, which consist of an upper bound for the batch complexity of Algorithm 1, and a lower bound for the batch complexity of anytime batched bandit algorithms that attain the asymptotically optimal regret in (1).

### 4.1 Upper Bound

The batch complexity of Algorithm 1 is given as follows.

**Theorem 2** (Batch Complexity)**.** *For any input $\alpha \in \mathcal{O}_T(1)$ and $I_1$ satisfying (6), the number of batches for Algorithm 1 is $O(\log\log T \cdot \mathrm{ilog}^{\alpha}(T))$.*

Furthermore, the following theorem shows that Algorithm 1 is asymptotically optimal for an unknown horizon $T$.

**Theorem 3** (Regret)**.** *For any input $\alpha \in \mathcal{O}_T(1)$ and $I_1$ satisfying (6), Algorithm 1 achieves the asymptotically optimal regret, i.e., $\lim_{T\to\infty} \frac{R_T}{\log T} = \sum_{i=2}^{K} \frac{\Delta_i}{\mathrm{kl}(\mu_i, \mu_1)}$.*

**Comparison with Previous Work.** Compared with the fully sequential algorithms, such as KL-UCB (Garivier & Cappé, 2011) and Thompson Sampling (Korda et al., 2013) that achieve the asymptotically optimal regret with $O(T)$ batches, our algorithm only needs $O(\log\log T \cdot \mathrm{ilog}^{\alpha}(T))$ batches while maintaining the asymptotically optimal regret. Compared with Anytime-DETC (Algorithm 5 in Jin et al. (2020)) that incurs at least $\Omega(\log T)$ batches, our algorithm not only significantly improves the batch complexity but also expands the applicability since Anytime-DETC only applies to 2-armed bandit with sub-Gaussian rewards, whereas our algorithm generalizes to $K$-armed bandit with exponential families of reward distributions.

### 4.2 Lower Bound

Besson & Kaufmann (2018) conjectures that *the geometric doubling trick can never bring the right constant in asymptotic regret bound* in (1). We present a lower bound that theoretically confirms this conjecture.

**Theorem 4.** *For any static grid and constant $c > 0$, no anytime algorithm can achieve the asymptotically optimal regret in (1) within $c \log\log T$ batches.*

It is worth noting that this is the first lower bound for anytime batched bandit problem in the literature. We observe that the lower bound in Theorem 4 matches the upper bound in Theorem 2 within an $\mathrm{ilog}^{\alpha}(T)$ factor, which shows that our batch complexity is almost optimal for achieving the asymptotic optimality in the anytime setting.

We note that Perchet et al. (2016); Gao et al. (2019) also proved certain lower bounds for the batched bandit problem. However, their lower bounds are significantly different from ours, since we focus on anytime algorithm tailored for minimizing the asymptotic regret with an unknown $T$, whereas they consider minimax regret or problem-dependent regret with a finite known $T$.

## 5 Theoretical Analysis

Now we present the proof of the upper bound results. The proof the lower bound can be found in Appendix C.

### 5.1 Analysis of Batch Complexity

*Proof of Theorem 2.* According to the definition of $f$, it is easy to verify that

$$\begin{aligned} I_{r+n} &\geq (I_{r+n-1})^{1+1/(1+\mathrm{ilog}^\alpha I_{r+n-1})} \\ &\geq (I_{r+n-1})^{1+1/(1+\mathrm{ilog}^\alpha I_r)} \geq (I_r)^{(1+1/(1+\mathrm{ilog}^\alpha I_r))^n}. \end{aligned}$$

For $n = \lceil 1 + \mathrm{ilog}^\alpha I_r \rceil$, we have $I_{r+n} \geq (I_r)^2$. Therefore,

$$f^{(\lceil 1+\mathrm{ilog}^\alpha T \rceil)}(I_r) \geq f^{(\lceil 1+\mathrm{ilog}^\alpha I_r \rceil)}(I_r) \geq (I_r)^2. \quad (9)$$

This implies that it needs at most $\lceil 1 + \mathrm{ilog}^\alpha T \rceil$ epochs for increasing $I_r$ from $(I_1)^{2^\ell}$ to $(I_1)^{2^{\ell+1}}$. Moreover, when $I_1 \geq 2$, $\ell^* = \log_2 \log_2 T$ suffices to ensure $(I_1)^{2^{\ell^*}} \geq T$, which indicates that the algorithm runs at most $\ell^* \cdot \lceil 1 + \mathrm{ilog}^\alpha T \rceil$ epochs. Besides, the number of epochs is proportional to the number of batches. Therefore, the total number of batches is $\mathcal{O}(\log T \log T \cdot \mathrm{ilog}^\alpha T)$. $\square$

### 5.2 Analysis of Regret

*Proof of Theorem 3.* Let $T_i$ be the total number of pulls of arm $i$ in Algorithm 1. Then, the regret can be rewritten as

$$R_T = \sum_{i \geq 2} \mathbb{E}[T_i \Delta_i].$$

Therefore, it suffices to prove the elementary result such that for each $i \geq 2$

$$\lim_{T \to \infty} \frac{\mathbb{E}[T_i]}{\log T} = \frac{1}{\mathrm{kl}(\mu_i, \mu_1)}. \quad (10)$$

For ease of presentation, for the $r$-th epoch, we fix a suboptimal arm $i$ and define some notations as follows.

- $Y_1(r)$: the number of pulls of arm $i$, excluding pulling $c_{r-1}$, in Step I (Lines 1–4 in Algorithm 2).
- $Y_2(r)$: the number of pulls of arm $i$ in Step II (Algorithm 3).
- $Y_3(r)$: the number of pulls of arm $i$ in Step III (Line 1–12 in Algorithm 4);
- $Y_4(r)$: the number of pulls of arm $i$ in Step IV (Algorithm 5) and V (Algorithm 6).
- $Z(r)$: the total number of pulls of arm $c_{r-1}$ when $c_{r-1} = i$ in Step I (Line 5–7 in Algorithm 2) and Step III (Line 13–15 in Algorithm 4).

In addition, since for $I_r \leq \sqrt{\log T}$, the algorithm plays suboptimal arms at most $\sqrt{\log T}$ times. We use $\sqrt{\log T}$ to bound the number of pulls of arm $i$ when $I_r \leq \sqrt{\log T}$. Let $r' := \min\{r \colon I_r > \sqrt{\log T}\}$ and $r^o$ be the total number of epochs. Then, we have

$$\begin{aligned} \mathbb{E}[T_i] &= \sum_{r=1}^{r^o} \left( \sum_{j=1}^{4} \mathbb{E}[Y_j(r)] + \mathbb{E}[Z(r)] \right) \\ &\leq \sqrt{\log T} + \sum_{r=r'}^{r^o} \left( \sum_{j=1}^{4} \mathbb{E}[Y_j(r)] + \mathbb{E}[Z(r)] \right) \\ &= \sqrt{\log T} + \sum_{j=1}^{4} \mathbb{E}[Y_j] + \mathbb{E}[Z], \end{aligned}$$

where $Y_j := \sum_{r=r'}^{r^o} Y_j(r)$ and $Z := \sum_{r=r'}^{r^o} Z(r)$. According to Lemmas 2–6, which shall be given later, we have

$$\frac{\mathbb{E}[T_i]}{\log T} = \frac{\sqrt{\log T}}{\log T} + \frac{1 + 1/(1 + \mathrm{ilog}^\alpha T)}{\mathrm{kl}(\mu_i - \epsilon_{r'}, \mu_1 + \epsilon_{r'})} + o_T(1).$$

Note that $\frac{\sqrt{\log T}}{\log T} \to 0$, $1/(1 + \mathrm{ilog}^\alpha T) \to 0$, $\epsilon_{r'} \to 0$ and $o_T(1) \to 0$ when $T \to \infty$. Therefore,

$$\lim_{T \to \infty} \frac{\mathbb{E}[T_i]}{\log T} = \frac{1}{\mathrm{kl}(\mu_i, \mu_1)}.$$

This completes the proof. $\square$

In the following, we bound $\mathbb{E}[Y_i]$ for $i \in [4]$ and $\mathbb{E}[Z]$, which requires the following concentration bounds for exponential families.

**Lemma 1** (Maximal Inequality (Ménard & Garivier, 2017))**.** *Let $N$ and $M$ be two real numbers in $\mathbb{R}^+ \times \overline{\mathbb{R}^+}$, let $\gamma > 0$, and $\widehat{\mu}_n$ be the empirical mean of $n$ random variables i.i.d. according to the distribution $\nu_{b'^{-1}(\mu)}$. Then*

$$\mathbb{P}(\exists N \leq n \leq M, \mathrm{kl}_+(\widehat{\mu}_n, \mu) \geq \gamma) \leq e^{-N\gamma}. \quad (11)$$

*As a consequence, for every $x \leq \mu$,*

$$\mathbb{P}(\exists N \leq n \leq M, \widehat{\mu}_n \leq x) \leq e^{-N(x-\mu)^2/(2V)}. \quad (12)$$

*Meanwhile, for every $x \geq \mu$,*

$$\mathbb{P}(\exists N \leq n \leq M, \widehat{\mu}_n \geq x) \leq e^{-N(x-\mu)^2/(2V)}. \quad (13)$$

**Lemma 2.** $\mathbb{E}[Y_1] / \log T = o_T(1)$.

*Proof.* Be definition, we have

$$I_{r^o-1} < T \le I_{r^o} = f(I_{r^o-1}) \le f(T) \le T^{1+1/(1+\mathrm{ilog}^\alpha T)}.$$

Note that the total number of pulls of arm $i$ contributed by $Y_1$ is at most $g(I_{r^o})$. Then, we have

$$\begin{aligned} Y_1 \le g(I_{r^o}) &\le \frac{2\log(I_{r^o})}{\log\log(I_{r^o})} \\ &\le \frac{2(1+1/(1+\mathrm{ilog}^\alpha T))\cdot\log T}{\log\log T}. \end{aligned}$$

Since $\alpha = \mathcal{O}_T(1)$, we obtain

$$\frac{\mathbb{E}[Y_1]}{\log T} \le \frac{2(1+1/(1+\mathrm{ilog}^\alpha T))}{\log\log T} = o_T(1). \qquad \square$$

**Lemma 3.** $\mathbb{E}[Y_2]/\log T = o_T(1)$.

*Proof.* Intuitively, when $I_r$ is sufficiently large, $a_{1,r}$ is arm 1 with high probability, i.e., $\mathbb{P}(a_{1,r}=1) \ge 1 - \frac{1}{\log^2 I_r}$. Hence, the expected number of pulls of arm $i$ in Step II is small. In the following, we first bound $\mathbb{P}(a_{1,r}=i)$.

Let $\Delta_{\min} := \min_{i\ge 2}\Delta_i$. After pulling $g(I_r)$ times of arm 1, by (12), we have

$$\mathbb{P}(\widehat{\mu}_{1,g(I_r)} \le \mu_1 - \Delta_{\min}/2) \le e^{-g(I_r)\Delta_{\min}^2/(8V)}. \quad (14)$$

Meanwhile,

$$g(I_r) \ge \frac{\log I_r}{\log\log I_r} \gtrsim \frac{8V\cdot(2\log\log I_r + 2\log K)}{\Delta_{\min}^2}.$$

Combining with (14) gives

$$\mathbb{P}(\widehat{\mu}_{1,g(I_r)} \le \mu_1 - \Delta_{\min}/2) \lesssim^{2} \frac{1}{2K\log^2 I_r}.$$

Similarly, for arm $i$, we can get that

$$\mathbb{P}(\widehat{\mu}_{i,g(I_r)} \ge \mu_i + \Delta_{\min}/2) \lesssim \frac{1}{2K\log^2 I_r}.$$

As a result,

$$\mathbb{P}(\widehat{\mu}_{1,g(I_r)} \le \widehat{\mu}_{i,g(I_r)}) \lesssim \frac{1}{K\log^2 I_r}. \quad (15)$$

Furthermore, we can obtain that

$$\begin{aligned} \mathbb{E}[Y_2] &= \sum_{r=r'}^{r^o}\mathbb{E}[Y_2(r)] \le \sum_{r=r'}^{r^o}\left(\log^2 I_r\cdot\mathbb{P}(a_{1,r}=i)\right) \\ &\le \sum_{r=r'}^{r^o}\left(\log^2 I_r\cdot\mathbb{P}(\widehat{\mu}_{1,g(I_r)} \le \widehat{\mu}_{i,g(I_r)})\right) \lesssim r^o \end{aligned}$$

Note that $r^o = \mathcal{O}(\log\log T\cdot\mathrm{ilog}^\alpha T)$ by Theorem 2. This implies that

$$\frac{\mathbb{E}[Y_2]}{\log T} = \mathcal{O}_T\left(\frac{\log\log T\cdot\mathrm{ilog}^\alpha T}{\log T}\right) = o_T(1). \qquad \square$$

[2] By choosing sufficiently large $T$, all $\lesssim$ hold simultaneously.

**Lemma 4.** $\frac{\mathbb{E}[Y_3]}{\log T} = \frac{1+1/(1+\mathrm{ilog}^\alpha T)}{\mathrm{kl}(\mu_i - 2\epsilon_{r'}, \mu_1 + 2\epsilon_{r'})} + o_T(1)$.

*Proof.* Define events

$$\begin{aligned} &\mathcal{E}_0(r) := \left\{|\widehat{\mu}_{a_{1,r},s_1} - \mu_{a_{1,r}}| < \epsilon_r\right\}, \\ &\mathcal{E}_1(r) := \{a_{1,r} = 1\}, \\ &\mathcal{E}_2(r) := \left\{\forall\, k\in[K]\setminus\{a_{1,r}\}\colon |\widehat{\mu}_{k,g(I_r)} - \mu_k| < \epsilon_r\right\}, \\ &\mathcal{E}(r) := \mathcal{E}_0(r)\cap\mathcal{E}_1(r)\cap\mathcal{E}_2(r). \end{aligned}$$

Based on $\mathcal{E}(r)$, we category the epochs into two sets

$$\begin{aligned} &S_1 := \{r\in[r',r^o]\colon \mathbb{1}(\mathcal{E}(r)) = 1\}, \\ \text{and}\quad &S_2 := \{r\in[r',r^o]\colon \mathbb{1}(\mathcal{E}^c(r)) = 1\}. \end{aligned}$$

Let $a_{i',r} = i$. Thus, for all epochs in $S_1$, arm $i$ is pulled at most $\max_{r\in S_1}\delta_{i',r}$ times in $Y_3$, i.e.,

$$\sum_{r\in S_1} Y_3(r) \le \max_{r\in S_1}\delta_{i',r}.$$

Meanwhile, when $T$ is sufficiently large, by definition, for every $r\in[r',r^o]$, we have

$$\epsilon_r = \frac{1}{\log\log I_r} \le \frac{1}{\log\log\sqrt{\log T}} \le \frac{\Delta_{\min}}{4}. \quad (16)$$

Thus, for any $r\in S_1$, we have

$$\widehat{\mu}_{i,g(I_r)} + \epsilon_r < \mu_i + 2\epsilon_r \le \mu_1 - 2\epsilon_r < \widehat{\mu}_{1,s_1} - \epsilon_r.$$

Then, we can get that

$$\delta_{i',r} \le \frac{\log(I_r\cdot\log^2 I_r)}{\mathrm{kl}(\mu_i+2\epsilon_r, \mu_1-2\epsilon_r)} \le \frac{\log(I_{r^o}\cdot\log^2 I_{r^o})}{\mathrm{kl}(\mu_i+2\epsilon_{r'}, \mu_1-2\epsilon_{r'})}.$$

As a result, we have

$$\begin{aligned} \sum_{r\in S_1} Y_3(r) &\le \max_{r\in S_1}\delta_{i',r} \le \frac{\log(I_{r^o}\cdot\log^2 I_{r^o})}{\mathrm{kl}(\mu_i+2\epsilon_{r'}, \mu_1-2\epsilon_{r'})} \\ &\le \frac{(1+1/(1+\mathrm{ilog}^\alpha T))\cdot\log T + 2\log\log f(T)}{\mathrm{kl}(\mu_i+2\epsilon_{r'}, \mu_1-2\epsilon_{r'})}, \quad (17) \end{aligned}$$

where the last inequality is due to the fact that $I_{r^o} \le f(T)$.

On the other hand,

$$\sum_{r\in S_2} Y_3(r) \le \sum_{r\in S_2}\log^2 I_r,$$

since arm $i$ is pulled at most $\log^2 I_r$ time in $Y_3(r)$. Hence,

$$\mathbb{E}\left[\sum_{r\in S_2} Y_3(r)\right] \le \sum_{r=r'}^{r^o}\mathbb{P}(\mathcal{E}^c(r))\log^2 I_r.$$

According to the definition of $\mathcal{E}^c(r)$, we have

$$\mathbb{P}(\mathcal{E}^c(r)) \le \mathbb{P}(\mathcal{E}_0^c(r)) + \mathbb{P}(\mathcal{E}_1^c(r)) + \mathbb{P}(\mathcal{E}_2^c(r)).$$

Note that after first two steps, $a_{1,r}$ has been pulled at least $\log^2 I_r$ times. Then, by Lemma 1, we have

$$\begin{aligned}\mathbb{P}(\mathcal{E}_0^c(r)) &\le \sum_{k=1}^{K} \mathbb{P}\big(\exists s \ge \log^2 I_r, |\widehat{\mu}_{k,s} - \mu_k| \ge \epsilon_r\big) \\ &\le 2K \cdot \exp\left(-\frac{\log^2 I_r}{2V(\log\log I_r)^2}\right) \lesssim \frac{1}{I_r},\end{aligned} \tag{18}$$

where the first inequality follows from the union bound over all possible event $a_{1,r} = k$. Meanwhile, by (15), we have

$$\mathbb{P}(\mathcal{E}_1^c(r)) \le \sum_{i=2}^{K} \mathbb{P}(\widehat{\mu}_{1,g(I_r)} \le \widehat{\mu}_{i,g(I_r)}) \lesssim \frac{1}{\log^2 I_r}.$$

Finally, observing that after first two steps, arm $i$ is pulled at least $g(I_r)$ times, by Lemma 1, we have

$$\begin{aligned}\mathbb{P}(\mathcal{E}_2^c(r)) &\le \sum_{k=1}^{K} \mathbb{P}\big(\exists s \ge g(I_r), |\widehat{\mu}_{k,s} - \mu_k| \ge \epsilon_r\big) \\ &\le 2K \cdot \exp\left(-\frac{g(I_r)}{2V(\log\log I_r)^2}\right) \lesssim \frac{1}{\log^2 I_r}.\end{aligned}$$

As a result, we have

$$\mathbb{P}(\mathcal{E}^c(r)) \le \sum_{s=1}^{3} \mathbb{P}(\mathcal{E}_s^c(r)) \lesssim \frac{1}{\log^2 I_r}. \tag{19}$$

Therefore,

$$\mathbb{E}\Bigg[\sum_{r\in S_2} Y_3(r)\Bigg] \le \sum_{r=r'}^{r^o}\sum_{n=1}^{3} \mathbb{P}(\mathcal{E}_n^c(r))\log^2(I_r) \lesssim r^o. \tag{20}$$

Combining (17) and (20), we obtain

$$\frac{\mathbb{E}[Y_3]}{\log T} = \frac{1 + 1/(1+\operatorname{ilog}^{\alpha} T)}{\mathrm{kl}(\mu_i - 2\epsilon_{r'}, \mu_1 + 2\epsilon_{r'})} + o_T(1). \qquad \square$$

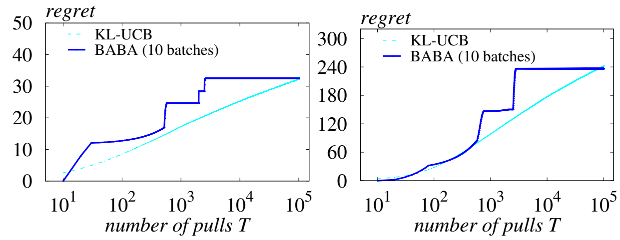
**Lemma 5.** $\mathbb{E}[Y_4]/\log T = o_T(1)$.

**Lemma 6.** $\mathbb{E}[Z]/\log T = o_T(1)$.

Due to the space limit, we refer readers to Appendix B for proofs of Lemma 5 and Lemma 6.

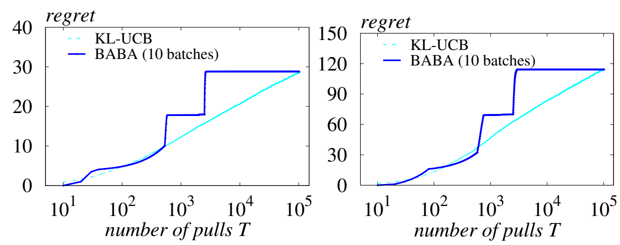
## 6 Experiment

In this section, we compare our algorithm BABA with KL-UCB (Garivier & Cappé, 2011) under two reward distributions, i.e., *Gaussian distribution* and *Bernoulli distribution*. For each distribution, we test BABA and KL-UCB with 2 arms and 5 arms respectively. Specifically, for 2-arm setting, we set $\mu \in \{1, 0\}$ and $\sigma = 1$ for Gaussian distribution; we set $p \in \{0.5, 0.25\}$ for Bernoulli distribution. For 5-arm setting, we set $\mu \in \{1, 0.5, 0.5, 0.5, 0.5\}$ and $\sigma = 1$ for Gaussian distribution; we set $p \in \{0.5, 0.25, 0.25, 0.25, 0.25\}$ for Bernoulli distribution. For our BABA algorithm, we set $\alpha = 3$ and $I_1 = 2000$. All the experiments are averaged over 2000 repetitions.




(a) $K = 2$ (b) $K = 5$

*Figure 1.* Regrets over Gaussian distributions. The experiments are averaged over 2000 repetitions




(a) $K = 2$ (b) $K = 5$

*Figure 2.* Regrets over Bernoulli distributions. The experiments are averaged over 2000 repetitions

For Gaussian rewards, Figure 1(a) and Figure 1(b) report the regret when $K = 2$ and $K = 5$, respectively. As we can see, when $T$ approaches $10^5$, BABA achieves the same regret as KL-UCB while requiring 10 batches opposed to $10^5$. The regret of BABA increases rapidly at some time steps. For example, in Figure 1(a), the regret increases from 17.0 to 24.2 from time steps 520 to 560. The reason is that in BABA, the suboptimal arms are mostly pulled during the exploration stages. In addition, as shown in Figure 1(a) and 1(b), when $T = 2000$, the regret of BABA is larger than KL-UCB. The reason is that for small $T$, BABA may not reach the optimal performance since asymptotic optimality holds only for sufficiently large $T$.

For Bernoulli rewards, Figure 2(a) and Figure 2(b) report the regret when $K = 2$ and $K = 5$, respectively. Again, the BABA achieves the comparable regret with that of KL-UCB while requiring 10 batches opposed to $10^5$.

## 7 Conclusion

We study the anytime bathed multi-armed bandit problem. We propose an algorithm BABA that achieves the asymptotically optimal regret using only $\mathcal{O}(\log\log T \cdot \operatorname{ilog}^{\alpha} T)$

batches. We also show a lower bound on the batch complexity of anytime bandit algorithms, which theoretically confirms the conjecture in Besson & Kaufmann (2018) that no algorithm using static time grid can achieve the asymptotic optimality within $c \log \log T$ batches for any constant $c$. Moreover, we conduct experiments to show that our algorithm achieves the comparable regret with that of KL-UCB while using significantly fewer batches.

## Acknowledgement

We thank the anonymous reviewers for their helpful comments. X. Xiao is supported by the Ministry of Education, Singapore, under Tier-2 Grant R-252-000-A70-112. T. Jin is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-01-004[T]). P. Xu and Q. Gu are partially supported by the National Science Foundation IIS-1904183. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

## References

Agrawal, S. and Goyal, N. Near-optimal regret bounds for thompson sampling. *Journal of the ACM*, 64(5):1–24, 2017.

Audibert, J.-Y. and Bubeck, S. Minimax policies for adversarial and stochastic bandits. In *Proc. COLT*, 2009.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proc. IEEE FOCS*, pp. 322–331, 1995.

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

Besson, L. and Kaufmann, E. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

Cesa-Bianchi, N., Dekel, O., and Shamir, O. Online learning with switching costs and other adaptive adversaries. In *Proc. NeurIPS*, 2013.

Degenne, R. and Perchet, V. Anytime optimal algorithms in stochastic multi-armed bandits. In *Proc. ICML*, pp. 1587–1595, 2016.

Esfandiari, H., Karbasi, A., Mehrabian, A., and Mirrokni, V. Batched multi-armed bandits with optimal regret. *arXiv preprint arXiv:1910.04959*, 2019.

Gao, Z., Han, Y., Ren, Z., and Zhou, Z. Batched multi-armed bandits problem. In *Proc. NeurIPS*, pp. 501–511, 2019.

Garivier, A. and Cappé, O. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proc. COLT*, pp. 359–376, 2011.

Harremoës, P. Bounds on tail probabilities in exponential families. *arXiv preprint arXiv:1601.05179*, 2016.

Jin, T., Xu, P., Xiao, X., and Gu, Q. Double explore-then-commit: Asymptotic optimality and beyond. *arXiv preprint arXiv:2002.09174*, 2020.

Kaufmann, E. et al. On bayesian index policies for sequential resource allocation. *The Annals of Statistics*, 46(2): 842–865, 2018.

Kittur, A., Chi, E. H., and Suh, B. Crowdsourcing user studies with mechanical turk. In *Proc. CHI*, pp. 453–456, 2008.

Korda, N., Kaufmann, E., and Munos, R. Thompson sampling for one-dimensional exponential family bandits. In *Proc. NeurIPS*, 2013.

Lai, T. L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22, 1985.

Lattimore, T. Refining the confidence level for optimistic bandit strategies. *Journal of Machine Learning Research*, 19(1):765–796, 2018.

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

Ménard, P. and Garivier, A. A minimax and asymptotically optimal algorithm for stochastic bandits. In *Proc. ALT*, pp. 223–237, 2017.

Perchet, V., Rigollet, P., Chassang, S., and Snowberg, E. Batched bandit problems. *The Annals of Statistics*, 44(2): 660–681, 2016.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Vaswani, S., Kveton, B., Wen, Z., Ghavamzadeh, M., Lakshmanan, L. V., and Schmidt, M. Model-independent online learning for influence maximization. In *Proc. ICML*, pp. 3530–3539, 2017.

Zhou, Y., Chen, X., and Li, J. Optimal pac multiple arm identification with applications to crowdsourcing. In *Proc. ICML*, pp. 217–225, 2014.

## A Inequalities on Kullback-Leibler Divergence

*Proof of Proposition 1.* We first prove (2). By Lemma 1 of Harremoës (2016),

$$\mathrm{kl}(\mu, \mu') = \int_{\mu}^{\mu'} \frac{x-\mu}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x. \tag{21}$$

By the assumption $\mathrm{Var}(\nu_{b'^{-1}(x)}) \le V$, we have

$$\mathrm{kl}(\mu, \mu') = \int_{\mu}^{\mu'} \frac{x-\mu}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x \ge \int_{\mu}^{\mu'} \frac{x-\mu}{V} \mathrm{d}x = \frac{(\mu'-\mu)^2}{2V}.$$

Next, we prove (3). From (21),

$$\begin{aligned}
\mathrm{kl}(\mu, \mu') &= \int_{\mu}^{\mu'} \frac{x-\mu}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x \\
&= \int_{\mu}^{\mu'-\epsilon} \frac{x-\mu}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x + \int_{\mu'-\epsilon}^{\mu'} \frac{x-\mu}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x \\
&\ge \mathrm{kl}(\mu, \mu'-\epsilon) + \int_{\mu'-\epsilon}^{\mu'} \frac{x-\mu}{V} \mathrm{d}x \\
&\ge \mathrm{kl}(\mu, \mu'-\epsilon) + \frac{\epsilon^2}{2V}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathrm{kl}(\mu-\epsilon, \mu') = \int_{\mu-\epsilon}^{\mu'} \frac{x-\mu+\epsilon}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x &= \int_{\mu-\epsilon}^{\mu} \frac{x-\mu+\epsilon}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x + \int_{\mu}^{\mu'} \frac{x-\mu+\epsilon}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x \\
\ge \int_{\mu-\epsilon}^{\mu} \frac{x-\mu+\epsilon}{\mathrm{Var}(\nu_{b'^{-1}(x)})} \mathrm{d}x + \mathrm{kl}(\mu, \mu') &\ge \mathrm{kl}(\mu, \mu') + \frac{\epsilon^2}{2V}.
\end{aligned}$$

This completes the proof. □

## B Missing Proofs of Theorem 3.

*Proof of Lemma 5.* We first decompose $\mathbb{E}[Y_4]$ according to $\mathcal{F}$.

$$\mathbb{E}[Y_4] = \mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F} = \perp)] + \mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F} = \top)]. \tag{22}$$

In the following, we separately bound $\mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F} = \perp)]$ and $\mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F} = \top)]$.

**Part I: Bounding $\mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F} = \perp)]$.** By definition,

$$\mathbb{E}[Y_4(r) \cdot \mathbb{1}(\mathcal{F} = \perp)] \le I_r \cdot \mathbb{P}(\mathcal{F} = \perp, a_{1,r} = i).$$

We further decompose $\mathbb{P}(\mathcal{F} = \perp, a_{1,r} = i)$ based on $\mathcal{E}_0(r)$.

$$\mathbb{P}(\mathcal{F} = \perp, a_{1,r} = i) \le \mathbb{P}(\mathcal{F} = \perp, a_{1,r} = i, \mathcal{E}_0(r)) + \mathbb{P}(\mathcal{E}_0^c(r)) \lesssim \mathbb{P}(\mathcal{F} = \perp, a_{1,r} = i, \mathcal{E}_0(r)) + \frac{1}{I_r},$$

where the second inequality is by (18). In addition, if $a_{1,r} = i$ and $\mathcal{F} = \perp$, by Line 4 of Algorithm 4, we have $s_1 \le \log^2 I_r$, and by Line 13 of Algorithm 4, we have

$$\mathrm{kl}_+(\widehat{\mu}_{1,s^*}, \widehat{\mu}_{i,s_1}) \ge \frac{\log\left(I_r \log^2 I_r\right)}{s^*},$$

where $s^* = s_{1'}$ such that $a_{1',r} = 1$. Moreover, when $T$ is sufficiently large, if the event $\mathcal{E}_0(r)$ happens, we have

$$\widehat{\mu}_{i,s_1} \le \mu_i + \epsilon_{r'} \le \mu_1.$$

This implies that

$$\frac{\log\left(I_r \log^2 I_r\right)}{s^*} \le \mathrm{kl}_+(\widehat{\mu}_{1,s^*}, \widehat{\mu}_{i,s_1}) \le \mathrm{kl}(\widehat{\mu}_{1,s^*}, \mu_1).$$

Therefore,

$$\mathbb{P}(\mathcal{F}=\perp, a_{1,r}=i, \mathcal{E}_0(r)) \le \mathbb{P}\left(\exists 1 \le n \le \log^2 I_r \colon \mathrm{kl}(\widehat{\mu}_{1,n}, \mu_1) \ge \frac{\log\left(I_r \log^2 I_r\right)}{n}\right) \le \sum_{n=1}^{\log^2 T} \frac{1}{I_r \cdot \log^2 I_r} \le \frac{1}{I_r},$$

where the first inequality is due to $s^* \le \log^2 I_r$ and the second equality is due to (11) of Lemma 1. Putting it together yields

$$\mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F}=\perp)] \lesssim I_r \cdot \left(\frac{1}{I_r} + \frac{1}{I_r}\right) = \mathcal{O}(1). \tag{23}$$

**Part II: Bounding $\mathbb{E}[Y_4 \cdot \mathbb{1}(\mathcal{F}=\top)]$.** We further decompose it based on whether $c_r = i$ is true.

$$\mathbb{E}[Y_4(r) \cdot \mathbb{1}(\mathcal{F}=\top)] \le \mathbb{P}(\mathcal{F}=\top)\log^2 I_r + \mathbb{P}(\mathcal{F}=\top, c_r=i) I_r. \tag{24}$$

In addition, we have

$$\mathbb{P}(\mathcal{F}=\top)\log^2 I_r \le \mathbb{P}(\mathcal{F}=\top, \mathcal{E}(r))\log^2 I_r + \mathbb{P}(\mathcal{E}^c(r))\log^2 I_r = \mathbb{P}(\mathcal{F}=\top, \mathcal{E}(r))\log^2 I_r + \mathcal{O}_T(1),$$

where the last inequality is due to (19). When $T$ is sufficiently large, if $\mathcal{E}(r)$ happens, for $k \ge 2$, we have

$$\delta_{k',r} = \frac{\log(I_r \cdot \log^2 I_r)}{\mathrm{kl}(\widehat{\mu}_{k,s_{k'}} + \epsilon_r, \widehat{\mu}_{1,s_1} - \epsilon_r)} \ge \frac{\log(I_r \cdot \log^2 I_r)}{\mathrm{kl}(\widehat{\mu}_{k,s_{k'}}, \widehat{\mu}_{1,s_1})},$$

and

$$\delta_{k',r} \le \frac{\log(I_r \cdot \log^2 I_r)}{\mathrm{kl}(\mu_k + 2\epsilon_r, \mu_1 - 2\epsilon_r)} \le \frac{2V \cdot \log(I_r \cdot \log^2 I_r)}{(\Delta_{\min} - 4\epsilon_r)^2} \le \log^2 I_r,$$

where the second inequality for deriving the upper bound of $\delta_{k',r}$ is from (2). Hence, for any $k \ge 2$, if $\mathcal{E}(r)$ happens, arm $k$ is pulled at least $\min\{\delta_{k',r}, \log^2 I_r\} = \delta_{k',r}$ times, which indicates that

$$s_{k'} \ge \delta_{k',r} \ge \frac{\log(I_r \cdot \log^2 I_r)}{\mathrm{kl}(\widehat{\mu}_{k,s_{k'}}, \widehat{\mu}_{1,s_1})}. \tag{25}$$

However, if $\mathcal{F}=\top$ and $a_{1,r}=1$, there must exist a $k \ge 2$ such that

$$s_{k'} < \frac{\log(I_r \cdot \log^2 I_r)}{\mathrm{kl}(\widehat{\mu}_{k,s_{k'}}, \widehat{\mu}_{1,s_{1'}})},$$

which contradicts to (25). Therefore, $\mathbb{P}(\mathcal{F}=\top, \mathcal{E}(r)) = 0$. Finally, we have

$$\mathbb{E}[Y_4(r) \cdot \mathbb{1}(\mathcal{F}=\top)] = \mathcal{O}_T(1). \tag{26}$$

Next, we derive $\mathbb{P}(\mathcal{F}=\top, c_r=i)I_r$. Conditioned on $\mathcal{F}=\top$ and $c_r=i$, the algorithm has pulled every arm at least $\log^2 I_r$ times. Similar to (18), we have

$$\mathbb{P}(\mathcal{F}=\top, c_r=i) \le \mathbb{P}(\mathcal{F}=\top, c_r \ne 1) \le \sum_{k=1}^{K} \mathbb{P}\left(\exists s \ge \log^2 I_r, |\widehat{\mu}_{k,s} - \mu_k| \ge \epsilon_r\right) \le 2K\exp\left(-\frac{\log^2 I_r}{2V(\log\log I_r)^2}\right) \lesssim \frac{1}{I_r}. \tag{27}$$

Substituting (26), (27) to (24), we have

$$\mathbb{E}[Y_4(r) \cdot \mathbb{1}(\mathcal{F}=\top)] = \mathcal{O}_T(1). \tag{28}$$

Combining (23) and (28), we obtain

$$\frac{\mathbb{E}[Y_4]}{\log T} = \frac{\mathcal{O}_T(r^o)}{\log T} = o_T(1).$$

□

*Proof of Lemma 6.* From the analysis for bounding $Y_4$, we know that

$$\mathbb{P}(c_r = i) = \mathbb{P}(\mathcal{F} = \perp, a_{1,r} = i) + \mathbb{P}(\mathcal{F} = \top, c_r = i) \lesssim 1/I_r.$$

As a result, we have

$$\begin{aligned}\sum_{r=r'}^{r^o} Z(r) &= Z(r') + \sum_{r=r'+1}^{r^o} Z(r)\\ &\leq I_{r'} + \sum_{r=r'+1}^{r^o} \mathbb{P}(c_{r-1} = i)(K(g(I_r) + \log^2 I_r))\\ &\lesssim I_{r'} + \sum_{r=r'+1}^{r^o} \frac{K(g(I_r) + \log^2 I_r)}{I_{r-1}}.\end{aligned}$$

Note that

$$I_r = (I_{r-1})^{1+1/(1+\mathrm{ilog}^{\alpha} I_{r-1})} \lesssim (I_{r-1})^2.$$

Hence, $I_{r-1} \gtrsim \sqrt{I_r}$. Furthermore, we have

$$\sum_{r=r'+1}^{r^o} \frac{K(g(I_r) + \log^2 I_r)}{I_{r-1}} \lesssim \sum_{r=r'+1}^{r^o} \frac{K(g(I_r) + \log^2 I_r)}{\sqrt{I_r}} \lesssim r^o.$$

Consequently,

$$\frac{\sum_{r=r'}^{r^o} Z(r)}{\log T} \lesssim \frac{I_{r'} + r^o}{\log T} = \frac{f(\sqrt{\log T})}{\log T} + o_T(1) = o_T(1).$$

This completes the proof. $\square$

## C Proof of the Lower Bound

The proof of Theorem 4 requires the following two Lemmas.

**Lemma 7** (Lemma 15.1 of Lattimore & Szepesvári (2020))**.** *Let $\nu = (P_1, \ldots, P_K)$ be the reward distributions associated with one $K$-armed bandit, and let $\nu' = (P'_1, \ldots, P'_K)$ be the reward distributions associated with another $K$-armed bandit. Fix some policy $\pi$ and let $\mathbb{P}_\nu = \mathbb{P}_{\nu\pi}$ and $\mathbb{P}_{\nu'} = \mathbb{P}_{\nu'\pi}$ be the probability measure on the canonical bandit model induced by the $N$-round interconnection of $\pi$ and $\nu$ (respectively, $\pi$ and $\nu'$). Then*

$$\mathrm{KL}(\mathbb{P}_\nu, \mathbb{P}_{\nu'}) = \sum_{i=1}^{K} \mathbb{E}_\nu[T_i(N)]\mathrm{KL}(P_i, P'_i).$$

**Lemma 8** (Lemma 14.2 of Lattimore & Szepesvári (2020))**.** *Let $P$ and $Q$ be the probability measure on the same measurable space $(\Omega, \mathcal{F})$ and let $A \in \mathcal{F}$ be an arbitrary event. Then,*

$$P(A) + Q(A^c) \geq \frac{1}{2}\exp(-\mathrm{KL}(P, Q)),$$

*where $A^c = \Omega \setminus A$ is the complement of $A$.*

*Proof of Theorem 4.* Without loss of generality, we consider that $c > 1$. By contradiction, we assume that there exists a strategy $\pi$ using time gird $\{t_1, t_2, \ldots\}$ (where $t_1 < t_2 < \cdots$) that achieves the asymptotically optimal regret within $c \log\log T$ batches. Hence, given $\epsilon = 1/(16c)$, there exists a (sufficiently large) $k$ such that for any $s$ and any $T$ with $s > k$ and $t_s \leq T < t_{s+1}$, it holds that

$$s \leq c \log\log T \tag{29}$$

and

$$\frac{R_T}{\log T} \le \sum_{i=2}^{K} \frac{(1+\epsilon)\Delta_i}{\mathrm{kl}(\mu_i, \mu_1)}, \tag{30}$$

where (29) is due to the assumption that the algorithm costs at most $c \log\log T$ batches, and (30) is due to the definition of limit for asymptotically optimal regret.

In what follows, we first show that for any $j' \ge k$, there exists a $j > j'$ such that

$$t_j \le N_n < N_{n+1} < t_{j+1} \qquad \text{and} \qquad N_{n+1} = (N_n)^x, \tag{31}$$

where $x = e^{1/(4c)}$, $n \ge 1$ and $N_1$ is a constant. Then, we further show that no algorithm can satisfy (30) at points $N_{n+1}$ and $t_j$, where outcomes up to time $t_j$ are observed. This contradicts to our assumption and hence concludes the theorem.

**Existence of $N_{n+1}$.** Consider the grid $\{t_{j'}, M_1, M_2, \dots\}$ with $M_i = (t_{j'})^{x^i}$, where $x = e^{1/(4c)}$. Let $T_1$ be a constant satisfying $T_1 > (t_{j'})^{e \log t_{j'}}$, and $m_1 \in \mathbb{N}^+$ such that $M_{m_1} \le T_1 < M_{m_1+1}$. Then, $M_{m_1+1} = (t_{j'})^{x^{m_1+1}} > T_1$. By the above definitions, we have

$$\begin{aligned}
\log\log M_{m_1+1} &> \log\log T_1, \\
\log\log M_{m_1+1} &= (m_1+1)\log x + \log\log t_{j'} = \frac{m_1+1}{4c} + \log\log t_{j'}, \\
\log\log T_1 &> 2\log\log t_{j'} + 1.
\end{aligned}$$

Therefore,

$$\frac{m_1+1}{4c} \ge \log\log T_1 - \log\log t_{j'} \ge 1/2 \log\log T_1.$$

Hence,

$$m_1 > 2c \log\log T_1.$$

Note that from (29), for $t_s \le T_1 < t_{s+1}$, it holds that $s \le c \log\log T_1$. As a consequence,

$$m_1 > 2c\log\log T_1 \ge 2s > s+1.$$

By the pigeonhole principle, we have that there exists $n_1 \in [m_1]$ and $t_{j'} < t_j < T_1$, such that

$$t_j \le M_{n_1} < M_{n_1+1} < t_{j+1}.$$

**Conflict with (30).** Consider an instance $t_j \le N_n < N_{n+1} < t_{j+1}$, from (31), $N_n$ can be arbitrarily large. Fix a suboptimal arm $i$, let $\nu = (P_1, P_2, \dots, P_K)$ be a bandit instance, where the $i$-th arm has distribution $P_i$ and mean $\mu_i$. Let $\epsilon' > 0$ be an arbitrary positive number and define $\nu' = (P'_1, \dots, P'_K)$ be another bandit instance, where the $\ell$-th arm has distribution $P'_\ell$ with mean $\mu'_\ell = \mu_\ell$ for $\ell \ne i$ and $\mu'_i$ be such that $\mathrm{kl}(\mu_i, \mu'_i) \le \mathrm{kl}(\mu_i, \mu_1) + \epsilon'$ and $\mu'_i > \mu_1$.

Note that the interconnection before time $N_{n+1}$ is $t_j$, from Lemma 7, we have

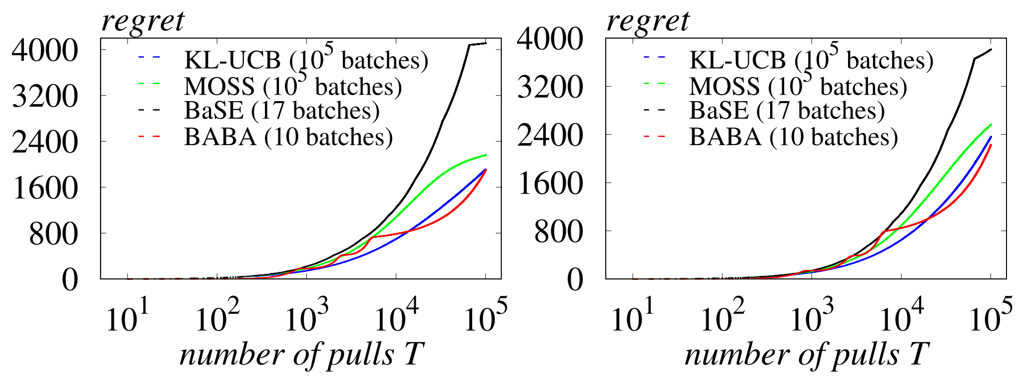
$$\mathrm{KL}(\mathbb{P}_\nu, \mathbb{P}_{\nu'}) = \mathbb{E}_{\nu\pi}[T_i(t_j)]\mathrm{kl}(\mu_i, \mu'_i) \le \mathbb{E}_{\nu\pi}[T_i(t_j)](\mathrm{kl}(\mu_i, \mu_1) + \epsilon').$$

Write $\mathbb{P}'$ and $\mathbb{E}'$ for expectation when rewards are sampled from $\nu'$. Then from Lemma 8, we have

$$\mathbb{P}(T_i(N_{n+1}) \ge N_{n+1}/2) + \mathbb{P}'(T_i(N_{n+1}) < N_{n+1}/2) \ge \frac{1}{2}\exp(-\mathbb{E}_{\nu\pi}[T_i(t_j)](\mathrm{kl}(\mu_i, \mu_1) + \epsilon')).$$

Let $R$ and $R'$ for the regret when rewards are sampled from $\nu$ and $\nu'$. Then we have

$$\begin{aligned}
R_{N_{n+1}} + R'_{N_{n+1}} &\ge \Delta_i N_{n+1}/2 \cdot \mathbb{P}(T_i(N_{n+1}) \ge N_{n+1}/2) + (\mu'_i - \mu_1)N_{n+1}/2 \cdot \mathbb{P}'(T_i(N_{n+1}) < N_{n+1}/2) \\
&\ge \min\{\Delta_i, \mu'_i - \mu_1\} \cdot N_{n+1}/4 \cdot \exp(-\mathbb{E}_{\nu\pi}[T_i(t_j)](\mathrm{kl}(\mu_i, \mu_1) + \epsilon'))
\end{aligned}$$




(a) Uniform Dataset (b) Gaussian Dataset

*Figure 3.* Regrets over Bernoulli distributions. The experiments are averaged over 2000 repetitions

Further, we have

$$\frac{\mathbb{E}_{\nu\pi}[T_i(t_j)]}{\log(N_{n+1})} \geq \frac{1}{\mathrm{kl}(\mu_i,\mu_1)+\epsilon'}\frac{\log\left(\frac{N_{n+1}\min\{\Delta_i,\mu_i'\}}{4(R_{N_{n+1}}+R'_{N_{n+1}})}\right)}{\log(N_{n+1})}.$$

For any $p>0$, when $N_{n+1}$ is sufficiently large, there exists a constant $C_p$ such that $R_{N_{n+1}}+R'_{N_{n+1}} \leq C_p(N_{n+1})^p$. Hence, we can choose a sufficiently large $N_{n+1}$ such that

$$\frac{\log\left(\frac{N_{n+1}\min\{\Delta_i,\mu_i'\}}{4(R_{N_{n+1}}+R'_{N_{n+1}})}\right)}{\log(N_{n+1})} \geq (1-\epsilon').$$

By now, we have

$$\frac{\mathbb{E}_{\nu\pi}[T_i(t_j)]}{\log(N_{n+1})} \geq \frac{1-\epsilon'}{\mathrm{kl}(\mu_i,\mu_1)+\epsilon'}.$$

From the definition of $N_{n+1}$, $\log(N_{n+1}) = \log((N_n)^x) \geq \log((t_j)^x) = x\log t_j$. As a result,

$$\frac{\mathbb{E}_{\nu\pi}[T_i(t_j)]}{\log(t_j)} \geq \frac{x(1-\epsilon')}{\mathrm{kl}(\mu_i,\mu_1)+\epsilon'} \geq \left(1+\frac{c}{4}\right)\frac{(1-\epsilon')}{\mathrm{kl}(\mu_i,\mu_1)+\epsilon'},$$

where the last inequality is because $x = e^{1/(4c)} \geq 1+\frac{1}{4c}$. By choosing a sufficiently small $\epsilon'$, such that

$$\left(1+\frac{c}{4}\right)\frac{(1-\epsilon')}{\mathrm{kl}(\mu_i,\mu_1)+\epsilon'} \geq \frac{1+c/8}{\mathrm{kl}(\mu_i,\mu_1)}, \tag{32}$$

we have that the regret at time $t_j$ satisfies

$$\frac{R_{t_j}}{\log t_j} = \frac{\sum_{i=2}^K \Delta_i\mathbb{E}[T_i(t_j)]}{\log t_j} \geq \sum_{i=2}^K \frac{\Delta_i\cdot(1+c/8)}{\mathrm{kl}(\mu_i,\mu_1)} > \sum_{i=2}^K \frac{\Delta_i\cdot(1+\epsilon)}{\mathrm{kl}(\mu_i,\mu_1)},$$

which contradicts to (30). This completes the proof. □

## D Additional Experiments

We added new experiments to compare BABA with baselines with knowing horizon $T$: BaSE (Gao et al., 2019) which achieves the near-optimal finite regret using $\mathcal{O}(\log T)$ batches, and MOSS (Audibert & Bubeck, 2009), which achieves the minimax optimal regret $\mathcal{O}(\sqrt{KT})$ using $\mathcal{O}(T)$ batches. We set $K=50$, $T=10^5$, and use Bernoulli reward distributions. We test the algorithms on two synthetic datasets as follows: (a) Uniform Dataset: the means of arms are uniformly drawn from $[0,1]$; (b) Gaussian Dataset: the means of arms are generated from a Gaussian distribution $\mathcal{N}(0.5,0.2)$ and are truncated into the interval $[0,1]$. Figure 3 shows the results. We can see that BABA achieves a comparable regret with those of KL-UCB and MOSS, while reducing the batch cost by 4 orders of magnitude. Compared with BaSE, BABA achieves both smaller regret and batch cost, since BaSE has a *hidden constant* in the regret bound and uses $\mathcal{O}(\log T)$ batches while BABA achieves the asymptotically optimal regret bound and uses $\mathcal{O}(\log\log T\cdot \mathrm{ilog}^{\alpha}(T))$ batches.