Best Arm Identification with Possibly Biased Offline Data

Le Yang[1] **Vincent Y. F. Tan**[3,1] **Wang Chi Cheung**[2]

[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2]Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore
[3]Department of Mathematics, National University of Singapore, Singapore

Abstract

We study the best arm identification (BAI) problem with potentially biased offline data in the fixed confidence setting, which commonly arises in real-world scenarios such as clinical trials. We prove an impossibility result for adaptive algorithms without prior knowledge of the bias bound between online and offline distributions. To address this, we propose the LUCB-H algorithm, which introduces adaptive confidence bounds by incorporating an auxiliary bias correction to balance offline and online data within the LUCB framework. Theoretical analysis shows that LUCB-H matches the sample complexity of standard LUCB when offline data is misleading and significantly outperforms it when offline data is helpful. We also derive an instance-dependent lower bound that matches the upper bound of LUCB-H in certain scenarios. Numerical experiments further demonstrate the robustness and adaptability of LUCB-H in effectively incorporating offline data.

1 INTRODUCTION

Best Arm Identification (BAI) is a fundamental problem in multi-armed bandit (MAB) research, where the goal is to find the arm with the highest expected reward through sequential sampling. Traditionally, BAI algorithms rely only on data collected during online. By contrast, in many practical applications, the decision maker (DM) has access to substantial amounts of historical data. Historical data, if relevant, can significantly reduce the number of online samples needed for decision-making if used correctly. Unfortunately, historical data often comes from different environments, making them biased and potentially misleading.

Challenges like this arise frequently in real-world applications. For example, in personalized recommendation systems, companies often possess years of historical user interaction data, which could provide a strong starting point for new recommendation models. However, user preferences and behavior evolve over time, creating a gap between past and present data distributions. Similarly, in clinical trials, previous studies can offer useful insights into treatment effectiveness, but differences in patient demographics or clinical settings may introduce biases that could mislead decision-making.

These examples highlight the importance of developing robust algorithms that can adaptively decide when and how to incorporate historical data. However, no existing algorithm is designed for BAI with possibly biased offline data. In this paper, we bridge this gap by proposing a novel approach.

Inspired by the above discussion, we consider BAI with possibly biased offline data in a fixed confidence setting. The learning process consists of two phases: a warm-start phase and an online phase. In the warm-start phase, the DM receives an offline dataset generated by a latent distribution P^{off}, which can be used to accelerate the identification process in the subsequent online phase. In the online phase, rewards are drawn from another latent distribution P^{on}. The goal is to identify the arm with the highest mean reward at a fixed confidence level while minimizing the sample complexity, specifically the number of samples collected during the online phase.

Intuitively, when P^{off} and P^{on} are "far apart", the DM should ignore the offline dataset and conduct online learning from scratch. For example, the Track-and-Stop policy (Garivier and Kaufmann, 2016) and the LUCB algorithm (Kalyanakrishnan et al., 2012) require

$$\Theta\left(\sum_{i:\Delta_i>0} \Delta_i^{-2} \log\left(\frac{1}{\delta}\right)\right) \tag{1.1}$$

samples to ensure correct identification with probability at least $1-\delta$, where $\Delta_i > 0$ is the suboptimality gap between the expected reward of the optimal arm and arm i and $\delta \in (0,1)$ is the given confidence level. This lower bound holds

for any P^{off} and P^{on}.

In contrast, when P^{off} and P^{on} are "sufficiently close" the DM can incorporate offline data to reduce unnecessary exploration. For instance, when $P^{\text{off}} = P^{\text{on}}$, the batch Track-and-Stop policy (Agrawal et al., 2023) requires

$$\Theta\left(\sum_{i:\Delta_i>0} \max\left(\Delta_i^{-2}\log\left(\frac{1}{\delta}\right) - T_S(i), 0\right)\right) \quad (1.2)$$

samples to achieve the same level of confidence, where $T_S(i)$ represents the number of offline samples for arm i in the warm-start phase. This bound is tighter than (1.1), but it only applies when $P^{\text{off}} = P^{\text{on}}$.

The goal of this paper is to design a BAI algorithm that adaptively decides whether to utilize historical data in each iteration, achieving a tighter bound than (1.1) when $P^{\text{off}} = P^{\text{on}}$, while retaining the bound (1.1) for general (not necessarily equal) P^{off} and P^{on}.

In this paper, we first reveal the fundamental limitations of adaptive algorithms from a theoretical perspective. Specifically, we construct two instances to demonstrate an impossibility result: without prior knowledge of an upper bound on the bias between offline and online data, any adaptive algorithm leveraging offline data may suffer higher sample complexity in certain cases, even exceeding the optimal sample complexity of purely online strategies. This result shows that it is infeasible to fully adaptively decide when and how to use offline data with unknown bias, highlighting the need for additional auxiliary information to design efficient algorithms.

To address this problem, we propose the Lower and Upper Confidence Bounds - History (LUCB-H) algorithm. LUCB-H introduces an effective bias upper bound to achieve adaptive correction of offline data within the LUCB framework. In each iteration, it computes upper and lower confidence bounds based on both online data and a combination of online and offline data. The algorithm then selects the smaller upper confidence bound and the larger lower confidence bound to obtain more accurate estimates, dynamically deciding whether to incorporate offline data. The algorithm significantly reduces sample complexity when offline data is helpful while discarding the data when the information is misleading, ensuring that the sample complexity remains consistent with that of the standard LUCB algorithm.

We present an instance-dependent lower bound on the sample complexity of δ-PAC algorithm, which can significantly reduce the sample complexity when the offline data is helpful for identifying the best arm. Moreover, we prove that LUCB-H can achieve this lower bound in certain specific instances, indicating that the algorithm is theoretically near-optimal in terms of sample complexity. Numerical experiment results further validate the effectiveness and robustness of LUCB-H, especially in complex environments with potentially biased offline data.

1.1 RELATED WORK

For the MAB problem, much attention has been devoted to the integration of offline data into online learning to improve performance guarantees. The majority of existing studies focus on the specific setting where offline and online share the same distribution ($P^{\text{off}} = P^{\text{on}}$) ensuring that historical data can be safely leveraged. Shivaswamy and Joachims (2012) integrated historical data in MAB, showing that logarithmic historical data reduces regret from logarithmic to constant. Agrawal et al. (2023) extended Track-and-Stop (Garivier and Kaufmann, 2016), achieving asymptotical optimal sample complexity under aligned offline and online distributions for BAI. Liu et al. (2025) focused on the integration of historical data in combinatorial bandits, and demonstrated its effectiveness in practical applications. Other research has focused on leveraging offline data in domains such as dynamic pricing (Bu et al., 2020), clustered bandits (Bouneffouf et al., 2019; Ye et al., 2020), reinforcement learning (Hao et al., 2023; Wagenmaker and Pacchiano, 2023), and sequential data settings (Gur and Momeni, 2022).

Thompson Sampling (TS) is a popular Bayesian method that effectively incorporates offline data into online learning by constructing prior distributions, yielding significant performance gains when distributions align (Russo and Van Roy, 2016). However, prior mis-specification can severely affect performance, making TS worse than state-of-the-art online policies without any offline data (Liu and Li, 2016; Simchowitz et al., 2021). Such results illustrate the necessity of adaptive strategies for deciding whether and how historical data should be used, particularly in scenarios with potentially biased offline data.

Zhang et al. (2019) explored the integration of biased offline data into contextual bandits. They proposed the ARROW-CB algorithm, which uses a weighted strategy to balance between ignoring and fully incorporating offline data without requiring prior knowledge of the offline-online discrepancy. While ARROW-CB does not offer improved regret bounds compared to baseline algorithms that ignore offline data. Cheung and Lyu (2024) formally demonstrated that, in the absence of a non-trivial upper bound on the discrepancy, no non-anticipatory policy can achieve better regret bounds than such baselines. To address this issue, Cheung and Lyu (2024) introduced an online policy MIN-UCB, which incorporates a non-trivial bias bound that adaptively determines whether to incorporate or discard historical data, improving regret bounds when historical data is helpful and retaining regret bounds when it is misleading. In contrast to our work, MIN-UCB focuses on regret minimization in MAB, while we address the BAI problem.

The discrepancy between offline and online data distributions plays a crucial role in determining whether historical

data can be effectively leveraged. Several studies have explored strategies to address this challenge. Si et al. (2023) proposed a distributionally robust strategy for learning under environmental changes, while Chen et al. (2022) explored distribution shift in reinforcement learning. Similar studies exist in supervised learning, such as Crammer et al. (2008); Mansour et al. (2009); Ben-David et al. (2010).

2 FORMULATION

We consider the BAI problem with possibly biased offline data. Let $A = \{1, 2, \ldots, k\}$ be the set of arms. The learning process consists of two phases: the warm-start phase and the online phase. During the warm-start phase, DM receives historical reward data for each arm. Specifically, for each arm $i \in A$, the DM is given $T_S(i)$ independent and identically distributed (i.i.d.) samples denoted as $\{X_s(i)\}_{s=1}^{T_S(i)}$, where each sample is drawn from an offline reward distribution P_i^{off}.

In the subsequent online phase, the DM selects an arm $A_t \in A$ to pull at each round t, and observes a stochastic reward $Y_t(A_t)$ drawn from the online reward distribution $P_{A_t}^{\text{on}}$. The offline and online reward distributions may differ, i.e., $P^{\text{off}} \neq P^{\text{on}}$.

The DM follows a policy $\pi = (\pi_t)_{t=1}^{\infty}$, which is a (possibly randomized) rule for selecting an arm A_t at each round t. The decision is based on the available observations up to time $t-1$, given by $\mathcal{F}_{t-1} = (S = \{T_S(i)\}_{i\in A}, \{A_l, Y_l(A_l)\}_{l=1}^{t-1})$. The underlying instance I is defined as a four-tuple $(A, \{T_S(i)\}_{i\in A}, P, \delta)$, where A is the set of arms, $\{T_S(i)\}_{i\in A}$ represents the number of historical reward data collected from the warm-start phase for each arm i, P denotes the reward distribution consisting of the online and offline reward distributions P^{on} and P^{off}, and δ is a given constant. In this setup, the DM has access only to A and $\{T_S(i)\}_{i\in A}$ before the online phase, whereas P and δ are unknown. In this paper, we assume that both the online and offline reward distributions, P^{on} and P^{off}, are normal distributions with known variance, a fact that is also known to the DM. For any arm $i \in A$, we denote its online and offline means as $\mu^{\text{on}}(i) = \mathbb{E}_{Y(i)\sim P_i^{\text{on}}}[Y(i)]$, and $\mu^{\text{off}}(i) = \mathbb{E}_{X(i)\sim P_i^{\text{off}}}[X(i)]$, respectively.

The DM aims to identify the arm with the highest mean reward in the online phase, defined as $I^* = \operatorname{argmax}_{i\in A} \mu^{\text{on}}(i)$. Without loss of generality, we assume that the means of the arms are ordered such that $\mu^{\text{on}}(1) > \mu^{\text{on}}(2) \geq \cdots \geq \mu^{\text{on}}(k)$, which implies that arm 1 is the unique best arm, i.e., $I^* = 1$. Let $\Delta_i = \mu^{\text{on}}(1) - \mu^{\text{on}}(i)$ denote the suboptimality gap between arm i and the best arm under the online distribution. Given $\delta > 0$, the DM seeks to design a non-anticipatory policy π that terminates at round $\tau_\delta > 0$ and outputs an estimate $\hat{I}^*_{\tau_\delta} = \operatorname{argmax}_{i\in A} \hat{\mu}^{\text{on}}_{\tau_\delta}(i)$, where $\hat{\mu}^{\text{on}}_{\tau_\delta}(i)$ represents the estimate of $\mu^{\text{on}}(i)$ at time τ_δ. The policy is required to satisfy $\mathbb{P}(\hat{I}^*_{\tau_\delta} = 1) \geq 1 - \delta$, i.e., it identifies the arm with the highest online mean with probability at least $1-\delta$. Any policy that satisfies this condition is called a δ-PAC algorithm.

While existing BAI algorithms depend solely on online data and not on offline data, an ideal algorithm should be adaptive, making real-time decisions about whether to incorporate historical data in each round. If historical data is useful, integrating it can lower the sample complexity required for a correct identification probability of at least $1-\delta$. Conversely, if it is misleading, the algorithm should discard the offline data and rely solely on the online data, ensuring that its sample complexity is of the same order as the best purely online strategy. The subsequent discussion demonstrates that such an adaptive strategy is impossible without additional auxiliary information.

3 AN IMPOSSIBILITY RESULT

Any δ-PAC algorithm that utilizes historical data benefits when P^{off} and P^{on} are close, but suffers when P^{off} and P^{on} differ significantly. However, an ideal algorithm should distinguish between these cases. In this section, we demonstrate that no ideal algorithm can exist by means of constructing lower bound on an "alternative instance" for any δ-PAC policy.

We consider two instances I_P and I_Q, which share the same arm set $A = \{1, 2\}$, offline sample sizes $\{T_S(i)\}_{i\in A}$ and the failure probability $\delta \in (0, 1)$. However, I_P and I_Q have different reward distributions, P and Q, and their optimal arms are also different. Specifically, for instance I_P,

$$P^{\text{on}} = P^{\text{off}}, \quad P_1^{\text{on}} = \mathcal{N}(0, 1) \quad \text{and} \quad P_2^{\text{on}} = \mathcal{N}(-\delta^\beta, 1).$$

Observe that the squared mean gap under P^{on} satisfies $(\mu^{\text{on}}(1) - \mu^{\text{on}}(2))^2 = \delta^{2\beta}$. If the offline sample numbers satisfy $T_S(1) = T_S(2) \geq 4(\delta^{-2\beta} - \delta^{-2\beta+\epsilon}) \log(1/\delta)$, where $\epsilon \in (0, \delta)$, then, existing online policies such as Track-and-Stop (Agrawal et al., 2023) achieves an expected stopping time at least

$$\begin{aligned} &\Omega(\mathbb{E}[\tau'_\delta] - T_S(1) - T_S(2)) \\ &= \Omega(\delta^{-2\beta} \log(1/\delta) - (\delta^{-2\beta} - \delta^{-2\beta+\epsilon}) \log(1/\delta)) \\ &= \Omega(\delta^{-2\beta+\epsilon} \log(1/\delta)), \end{aligned} \tag{3.1}$$

where $\mathbb{E}[\tau'_\delta]$ is the expected stopping time without using historical data when given the error bound $\delta \in (0, 1)$. Clearly, the bound in (3.1) strictly outperforms that in (1.1) in terms of sample complexity. Despite this apparent improvement, we show that any non-anticipatory policy that outperforms (1.1) on I_P must incur a higher sample complexity than (1.1) on a suitably chosen I_Q. Let $\mathbb{E}[\tau_\delta(P)]$ and $\mathbb{E}[\tau_\delta(Q)]$ be the expected stopping times of the δ-PAC algorithm on Instances P and Q, respectively. The following proposition is proved in Appendix B.

Proposition 3.1 *Consider instance I_P as described above. Suppose the offline sample sizes satisfy $T_S(1) \in \mathbb{N}$ and there exists $\epsilon > 0$ such that $T_S(2) \leq \frac{C}{2}(\delta^{-2\beta} - \delta^{-2\beta+\epsilon})$, where $C > 0$ is an absolute constant and $\delta \in (0,1)$. Then, for any δ-PAC policy π that satisfies*

$$\mathbb{E}[\tau_\delta(P)] \leq C\delta^{-2\beta+\epsilon}\log(1/\delta)$$

on instance I_P, there exists an instance I_Q defined as

$$Q_1 = P_1, \quad Q_2^{\text{on}} = \mathcal{N}(\delta^\beta, 1),$$
$$Q_2^{\text{off}} = \mathcal{N}\bigg(-\sqrt{\frac{8\delta^{-2\beta-\epsilon}}{C(1-\delta^\epsilon)}} - \delta^\beta, 1\bigg),$$

such that as $\delta \to 0^+$, the policy π suffers

$$\mathbb{E}[\tau_\delta(Q)] \geq \Omega\bigg(\frac{1}{\delta^{2\beta+\epsilon}}\log\Big(\frac{1}{\delta}\Big)\bigg) = \omega(\mathbb{E}[\tau'_\delta]).$$

We first analyze instances I_P and I_Q. In instance I_P, arm 1 is the better arm, while in instance I_Q, arm 2 is the better arm. Instances I_P and I_Q share the same arm set, same offline sample sizes which are sufficiently large to reduce the sample complexity on instance I_P with parameter δ. Although the online reward distributions are different, with $P_2^{\text{on}} = \mathcal{N}(-\delta^\beta, 1)$ in I_P and $Q_2^{\text{on}} = \mathcal{N}(\delta^\beta, 1)$ in I_Q, a sign flip and arm relabeling make the two instances equivalent to any online-only policy. The offline datasets are generated from different offline reward distributions. Consequently, offline data plays a critical role in these instances: it can be helpful in I_P, but misleading in I_Q. Without additional information, we cannot determine whether the offline data will be helpful or misleading.

Proposition 3.1 shows that an ideal algorithm does not exist. Even when we consider an extreme case when $T_S(1) = T_S(2) = \infty$, which implies that the DM knows the exact mean value of the offline reward distributions P^{off}, it is not possible to find an algorithm simultaneously (1) if the offline data is helpful (instance I_P), it could reduce sample complexity compared to pure online strategies like LUCB; (2) if the offline data is misleading (instance I_Q), it should be "smart" enough to discard the historical data and match the sample complexity of the standard LUCB algorithm.

In the following section, we introduce an LUCB-based algorithm, called LUCB-H, which incorporates an additional input: the valid biased bound $V = \{V(i)\}_{i\in A} \in (\mathbb{R} \cup \{\infty\})^k$ on an instance I such that

$$V(i) \geq |\mu^{\text{off}}(i) - \mu^{\text{on}}(i)|, \quad \text{for each } i \in A.$$

The quantity $V(i)$ serves as an upper bound on the mean shift between the offline and online reward distributions $P^{\text{off}}(i)$ and $P^{\text{on}}(i)$ for arm i. Before the online phase, the DM can construct the prior knowledge of $V(i)$ by several machine learning estimators, such as the LASSO (Blanchet et al., 2019), by cross-validation (Chen et al., 2022), or obtain some insights on how to construct them empirically (Si et al., 2023).

If $V(i) = \infty$, then the DM has no prior knowledge about the difference between the online and offline distributions before the online phase. In this case, Proposition 3.1 has already demonstrated that no ideal algorithm exists. Therefore, we assume that $V(i) < \infty$. Similar upper bound can be found in the contexts of supervised learning (Crammer et al., 2008), offline policy learning on contextual bandits (Si et al., 2023), stochastic optimization (Besbes et al., 2022) and multi-task bandit learning (Wang et al., 2021).

Algorithm 1 LUCB-H Algorithm

Input: Valid bias bound $\{V(i)\}_{i=1}^k$ on the instance, confidence parameter δ, offline samples S

For each $i \in A$, compute $\hat{X}(i)$ and set $\hat{Y}_0(i) = 0$
At $t = 1, \ldots, k$, pull each arm once, then set $N_{k+1}(i) = 1$ for all $i \in A$
for $t = k+1, k+2, \ldots$ **do**
  // Construct lower and upper bounds.
  Compute $\text{LCB}_t(i)$ and $\text{UCB}_t(i)$ by (4.1)–(4.2)
  Compute $\text{LCB}_t^S(i)$ and $\text{UCB}_t^S(i)$ by (4.3)–(4.4)
  Compute $\text{LCB}^{\text{mix}}(i) = \max\{\text{LCB}_t(i), \text{LCB}_t^S(i)\}$ and $\text{UCB}^{\text{mix}}(i) = \min\{\text{UCB}_t(i), \text{UCB}_t^S(i)\}$
  Set $h_t = \text{argmax}_{i\in A}\, \text{UCB}^{\text{mix}}(i)$
  Set $l_t = \text{argmax}_{i\neq h_t}\, \text{UCB}^{\text{mix}}(i)$
end for
if $\text{LCB}^{\text{mix}}(h_t) \geq \text{UCB}^{\text{mix}}(l_t)$ **then**
  Break the loop.
end if
Pull arms h_t and l_t, and observe independent rewards: $Y_t(h_t) \sim P_{h_t}^{\text{on}}$ and $Y_t(l_t) \sim P_{l_t}^{\text{on}}$.
Update sample mean $\hat{Y}_{t+1}(i)$ and sample counters $N_{t+1}(i)$ by

$$\hat{Y}_{t+1}(i) = \begin{cases} \frac{N_t(i)\cdot\hat{Y}_t(i) + Y_t(i)}{N_t(i)+1}, & \text{if } i = h_t \text{ or } l_t \\ \hat{Y}_t(i), & \text{otherwise,} \end{cases}$$

and $N_{t+1}(i) = N_t(i) + \mathbb{1}\{i \in \{h_t, l_t\}\}$.

Output: $I^* = \text{argmax}_{i\in A}\, \text{UCB}^{\text{mix}}(i)$

4 DESIGN AND ANALYSIS OF THE LUCB-H ALGORITHM

The LUCB algorithm was first proposed by Kalyanakrishnan et al. (2012). Unlike the UCB algorithm, which is more suitable for cumulative reward maximization, LUCB is specifically designed for the BAI problem (more precisely, the best m-arm identification problem where $m \in A$). In each round, LUCB selects two arms: the one with the highest sample mean and the one with the largest upper confidence bound among the remaining arms. LUCB provides

a theoretical upper bound on the error probability under the PAC framework and achieves near-optimal sample complexity in many cases. As an attractive approach for finite-action stochastic bandits, it has been extended to other variants, such as the top feasible arm identification problem (Katz-Samuels and Scott, 2019). With these merits in mind, it seems quite natural to generalize the idea of LUCB to address the BAI problem in a historical data setting.

To extend the LUCB algorithm to a setting with historical data, the main challenge is deciding how much to rely on offline data while ensuring accurate online decision-making. To address this problem, LUCB-H computes both the upper and lower confidence bounds using two types of estimators.

- The first uses **only online data** and follows the standard LUCB method:

$$\mathrm{LCB}_t(i) = \hat{Y}_t(i) - \sqrt{\frac{2\log(kt/\delta)}{N_t(i)}} \quad \text{and} \tag{4.1}$$

$$\mathrm{UCB}_t(i) = \hat{Y}_t(i) + \sqrt{\frac{2\log(kt/\delta)}{N_t(i)}}, \tag{4.2}$$

  where $\hat{Y}_t(i)$ is the empirical mean of arm i.

- The second **incorporates historical data**. Let $\hat{X}(i)$ be the historical mean, i.e., $\hat{X}(i) = \frac{\sum_{s=1}^{T_S(i)} X_s(i)}{T_S(i)}$. Then at round t, the estimated mean is

$$\hat{Y}_t^S(i) = \frac{N_t(i)\cdot\hat{Y}_t(i) + T_S(i)\cdot\hat{X}(i)}{N_t(i)+T_S(i)},$$

  with adjusted confidence bounds

$$\mathrm{LCB}_t^S(i) = \hat{Y}_t^S(i) - \mathrm{rad}_t^S(i) \quad \text{and} \tag{4.3}$$

$$\mathrm{UCB}_t^S(i) = \hat{Y}_t^S(i) + \mathrm{rad}_t^S(i), \tag{4.4}$$

  respectively, where

$$\mathrm{rad}_t^S(i) = \sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}} + \frac{T_S(i)}{N_t(i)+T_S(i)}\cdot V(i).$$

The above confidence interval consists of two parts: the first term is a standard confidence bound after incorporating historical data, and the second term compensates for potential bias, which is quantified by the parameter $V(i)$.

The core idea of the LUCB algorithm is to iteratively narrow down the set of candidate arms using these confidence bounds, which accelerates convergence. Inspired by this insight, the algorithm then conservatively selects the tighter bounds, i.e.,

$$\mathrm{LCB}_t^{\mathrm{mix}}(i) = \max\{\mathrm{LCB}_t(i), \mathrm{LCB}_t^S(i)\}, \quad \text{and}$$
$$\mathrm{UCB}_t^{\mathrm{mix}}(i) = \min\{\mathrm{UCB}_t(i), \mathrm{UCB}_t^S(i)\}.$$

Since the arm with the largest sample mean may differ with or without historical data, LUCB-H selects h_t as the arm with the largest upper bound, then proceeds as in standard LUCB.

Next, we analyze $\mathrm{UCB}_t^S(i)$ and $\mathrm{UCB}_t(i)$. When the bias bound $V(i)$ is small and the offline sample size $T_S(i)$ is large, typically $\mathrm{UCB}_t^S(i) < \mathrm{UCB}_t(i)$, indicating historical data can reduce confidence intervals and speed up convergence. A small $V(i)$ suggests similarity between P^{on} and P^{off}, making historical data beneficial; otherwise, historical data should be discarded.

In addition, let us compare the expectations of $\mathrm{UCB}_t^S(i)$ and $\mathrm{UCB}_t(i)$, to intuitively analyze the impact of the parameter $V(i)$ and the amount of historical data for arm $i \in A$ on the algorithm's performance. Here, we focus on round t and, for the sake of exposition, we treat $N_t(i)$ as a deterministic quantity, which we denote as $n_t(i)$. Then,

$$\begin{aligned}
&\mathbb{E}[\mathrm{UCB}_t^S(i) | N_t(i) = n_t(i)] \\
&= \frac{n_t(i)\cdot\mu^{\mathrm{on}}(i) + T_S(i)\cdot\mu^{\mathrm{off}}(i)}{n_t(i)+T_S(i)} + \frac{T_S(i)}{n_t(i)+T_S(i)}\cdot V(i) \\
&\quad + \sqrt{\frac{2\log(kt/\delta)}{n_t(i)+T_S(i)}} \\
&= \mu^{\mathrm{on}}(i) + \frac{T_S(i)}{n_t(i)+T_S(i)}(V(i) + \mu^{\mathrm{off}}(i) - \mu^{\mathrm{on}}(i)) \\
&\quad + \sqrt{\frac{2\log(kt/\delta)}{n_t(i)+T_S(i)}}
\end{aligned}$$

and

$$\mathbb{E}[\mathrm{UCB}_t(i) | N_t(i) = n_t(i)] = \mu^{\mathrm{on}}(i) + \sqrt{\frac{2\log(kt/\delta)}{n_t(i)}}.$$

As analyzed above, the biased bound $V(i)$ and the actual mean shift between offline and online data $\mu^{\mathrm{off}}(i) - \mu^{\mathrm{on}}(i)$ determine whether the historical data is helpful for identifying the best arm, and the number of offline samples $T_S(i)$ plays a critical role in the reducing sample complexity. Hence, we define a discrepancy measure

$$\eta(i) = V(i) + \mu^{\mathrm{off}}(i) - \mu^{\mathrm{on}}(i)$$

on arm i and we can know that $\eta(i) \in [0, 2V(i)]$. When the length of the confidence interval on arm i is smaller than $\Delta_i/4$, sampling for arm i can stop (Jamieson and Nowak, 2014). Therefore, we can infer that when $\eta_i < \Delta_i/4$, the historical data for arm i is helpful for the identification process. The following theorem further supports this inference.

Theorem 4.1 *The LUCB-H algorithm, which inputs a valid bias bound V on instance I, the expected stopping time $\mathbb{E}[\tau_\delta]$ satisfies*

$$\mathbb{E}[\tau_\delta] = O\Big(\sum_{\Delta_i>0}\Big(\frac{1}{\Delta_i^2}\log(\frac{1}{\delta}) - \mathrm{Sav_u}(i)\Big)\Big), \tag{4.5}$$

where the "Saving" term in the upper bound is

$$\mathrm{Sav_u}(i) = T_S(i) \cdot \max\left\{1 - \frac{4\eta(i)}{\Delta_i}, 0\right\}.$$

Theorem 4.1 is provided in Appendix C. The upper bound in (4.5) is not greater than the expected stopping time in the standard case using LUCB (without utilizing historical data). Note that $\mathrm{Sav_u}(i) \geq 0$. When $\eta(i) \leq \Delta_i/4$, $P^{\mathrm{on}}(i)$ and $P^{\mathrm{off}}(i)$ are sufficiently close, indicating that the historical data is helpful, and the sample complexity of LUCB-H can be *strictly less* than that of LUCB. Otherwise, the historical data is misleading, and the DM should discard it. In this case, LUCB-H adaptively *retains* the sample complexity of the standard LUCB algorithm. If $\eta(i) \leq \Delta_i/4$, then $\mathrm{Sav_u}(i)$ is monotonically increasing with respect to $T_S(i)$. This means that the more helpful historical data we have, the better it is for identifying the best arm, which is consistent with common intuition.

5 LOWER BOUND OF ANY δ-PAC ALGORITHM

This section characterizes the complexity of the BAI problem when using possibly biased offline data. It assumes that the set of probability measures $\mathcal{P}$ satisfies a common assumption used in the BAI and MAB literature (Lai and Robbins, 1985; Kaufmann et al., 2016), specifically related to the continuity of the KL divergence. This assumption enables us to develop change-of-measure arguments for deriving lower bounds on the problem's complexity.

Assumption 5.1 *For all $v, v' \in \mathcal{P}^2$ such that $v \neq v'$, for all $a > 0$, there exists $v'' \in \mathcal{P}$ such that $\mathrm{KL}(v, v') < \mathrm{KL}(v, v'') < \mathrm{KL}(v, v') + a$ and $\mathbb{E}_{X\sim v''}[X] > \mathbb{E}_{X\sim v'}[X]$. Furthermore, there exists $v''' \in \mathcal{P}$ such that $\mathrm{KL}(v, v') < \mathrm{KL}(v, v''') < \mathrm{KL}(v, v') + a$ and $\mathbb{E}_{X\sim v'''}[X] < \mathbb{E}_{X\sim v'}[X]$.*

Theorem 5.1 *Suppose that $\mathcal{P}$ satisfies Assumption 5.1; any algorithm that is δ-PAC on $\mathcal{M}$ satisfies, for $\delta \leq 0.15$, the expected stopping time*

$$\mathbb{E}[\tau_\delta] = \Omega\left(\sum_{\Delta_i > 0}\left(\frac{1}{\mathrm{KL}(\mu^{\mathrm{on}}(i), \mu^{\mathrm{on}}(1))}\log\left(\frac{1}{\delta}\right) - \mathrm{Sav_l}(i)\right)\right),$$

where $\mathrm{Sav_l}(i)$, the "Saving" term in the lower bound, is

$$\mathrm{Sav_l}(i) = T_S(i)\max\left\{\frac{\mu^{\mathrm{off}}(1) - \mu^{\mathrm{off}}(i)}{\Delta_i}, 0\right\}^2.$$

Comparing the lower bound to standard LUCB, LUCB-H improves the bound on the stopping time bound by including the saving term $\mathrm{Sav_l}(i)$ for arms $i \in A$. The proof of Theorem 5.1 can be found in Appendix D.

Next, we will analyze the upper bound of the stopping time of the LUCB-H algorithm and the lower bound of BAI with biased historical data. Clearly, the upper bound of LUCB-H is almost equal to the lower bound of sample complexity for the BAI problem with possibly biased offline data. Specifically, when the offline data are misleading and identical, the two bounds are identical. In the following, we quantify the gap between two bounds in two cases when the saving terms $\mathrm{Sav_u}(i) > 0$ and $\mathrm{Sav_l}(i) > 0$. We define the *gap* between them as $\mathrm{gap}(i) := \mathrm{Sav_l}(i) - \mathrm{Sav_u}(i)$.

Remark 5.1 *We analyze the above-defined gap $\mathrm{gap}(i)$ in two special cases.*

1. *$V(i)$'s are equal: In this case, we can show that gap is non-negative because*

$$\mathrm{gap}(i) = \left(\frac{(\eta(i) - \eta(1))^2}{\Delta_i^2} + 2\frac{\eta(i) + \eta(1)}{\Delta_i}\right)T_S(i) \geq 0.$$

2. *$V(1) = \eta(1)$, i.e., $P^{\mathrm{on}}(1) = P^{\mathrm{off}}(1)$. In this case, we can also show that the gap is non-negative because*

$$\mathrm{gap}(i) = \left(\frac{(\eta(i) - V(i))^2}{\Delta_i^2} + 2\frac{\eta(i) + V(i)}{\Delta_i}\right)T_S(i) \geq 0.$$

While we believe that $\mathrm{gap}(i) \geq 0$ holds generally, we leave the proof of this to future work.

6 NUMERICAL RESULTS

In this section, we conducted numerical experiments to evaluate the performance of the LUCB-H algorithm, which leverages biased offline data, and compared its performance with the Pure LUCB algorithm (the LUCB algorithm, which does not utilize any historical data). All experiments were based on a 5-armed bandit problem with rewards corrupted by Gaussian noise with unit variance, and the results reported are averaged over 1000 trials. The overall experimental design is divided into two groups based on the online reward distributions (staircase and linear), with both groups incorporating the same offline data construction and experimental variable settings.

We will introduce the design of online and offline reward distributions first. In Group 1, the online reward distributions exhibit a staircase pattern with means of $(0.8, 0.4, 0.4, 0.4, 0.4)$. In Group 2, the online reward distributions follow a linear pattern with means of $(0.8, 0.7, 0.6, 0.5, 0.4)$. Note that in both groups, arm 1 is the best arm, and the Pure LUCB algorithm makes its decision solely based on online data.

For each group, three different offline reward distributions are constructed (with the offline data generated from P^{off} and with different number of historical samples per arm):

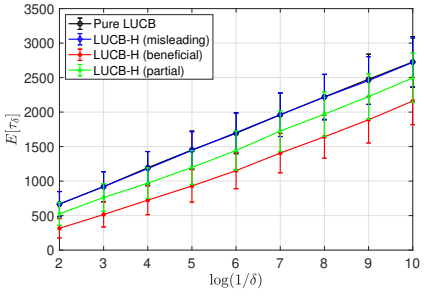

Figure 1: Evolution of $\mathbb{E}[\tau_\delta]$ with $\log(1/\delta)$ in group 1

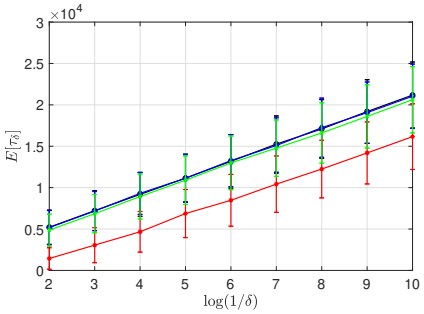

Figure 2: Evolution of $\mathbb{E}[\tau_\delta]$ with $\log(1/\delta)$ in group 2

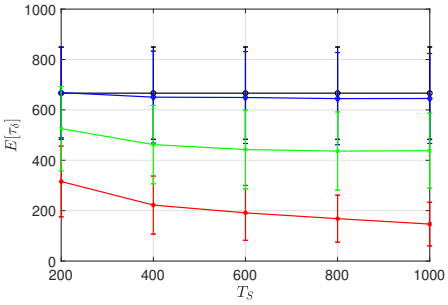

Figure 3: Evolution of $\mathbb{E}[\tau_\delta]$ with T_S in group 1

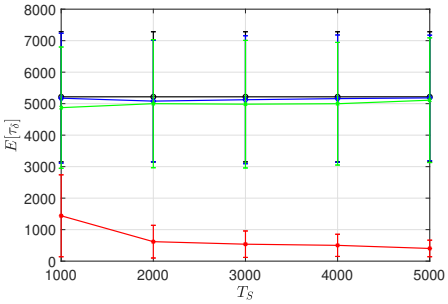

Figure 4: Evolution of $\mathbb{E}[\tau_\delta]$ with T_S in group 2

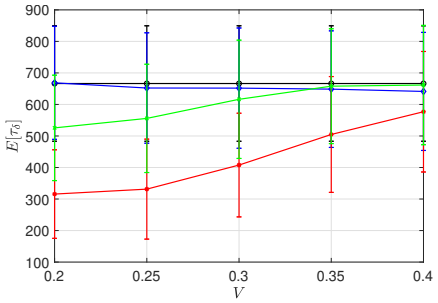

Figure 5: Evolution of $\mathbb{E}[\tau_\delta]$ with V in group 1

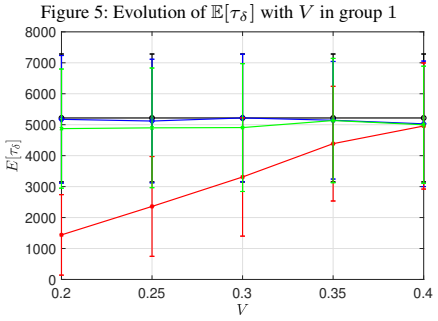

Figure 6: Evolution of $\mathbb{E}[\tau_\delta]$ with V in group 2

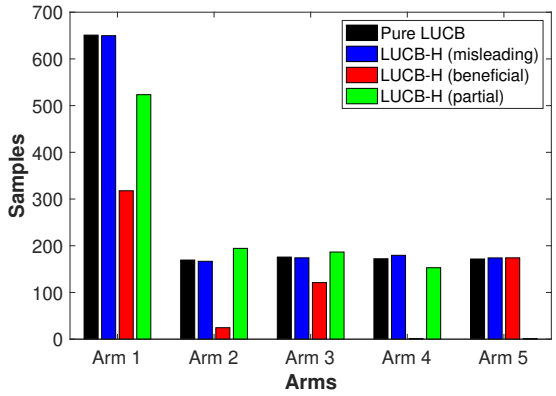

Figure 7: Samples for different arms when $\delta = 0.01$ in group 1

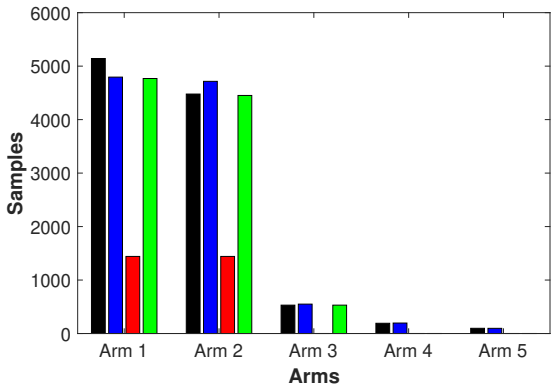

Figure 8: Samples for different arms when $\delta = 0.01$ in group 2

(a) **Case** 1 **(Misleading Bias):** In this case, all arms are adversely affected. Thus, the offline mean for the best arm decreases while the means for the suboptimal arms increase. The offline means in Group 1 are $(0.4, 0.6, 0.6, 0.6, 0.6)$, resulting in the offline phase identifying a best arm that is not arm 1. The offline means in Group 2 are $(0.4, 0.8, 0.7, 0.6, 0.5)$, which likewise leads to an offline best arm being different from arm 1.

(b) **Case** 2 **(Beneficial Bias):** In this case, the offline data assists in the identification process, with the offline mean for the best arm being increased while those for the non-best arms are decreased. The offline means in Group 1 are $(0.9, 0.2, 0.2, 0.2, 0.2)$ so that the offline phase still correctly identifies arm 1 as the best. The offline means in Group 2 are $(0.9, 0.5, 0.4, 0.3, 0.2)$, where the offline best arm remains arm 1.

(c) **Case 3 (Partial Adjustment):** In this case, selective adjustments are made across the arms. The offline means in Group 1 are $(0.4, 0.6, 0.6, 0.2, 0.2)$, under which the offline phase will not select arm 1 as the best. The offline means in Group 2 are $(0.4, 0.8, 0.7, 0.3, 0.2)$, leading to an offline best arm different from arm 1.

We also explored the impact of $T_S(i)$ and $V(i)$ on the performance of the LUCB-H algorithm. The two factors represent the amount of offline data and the deviation between offline and online data, respectively, which can significantly affect the convergence speed and the accuracy of identifying the best arm. We assess the expected stopping times of Pure LUCB and LUCB-H under three instances:

1. **Variation of** δ**:** The offline sample size is fixed at $T_S(i) = 200$ in Group 1 and $T_S(i) = 1000$ in Group

2 for all $i \in A$, and the bias parameter is configured as $V = (0.4, 0.2, 0.2, 0.2, 0.2)$ in Group 1 and Group 2. We set the confidence levels at $\delta = 0.1, 0.1^2, \ldots, 0.1^{10}$.

2. **Variation of Offline Sample Size $T_S(i)$ and Parameter $V(i)$:** Under a fixed $\delta = 0.01$:
   - The offline sample size for each arm is varied among 200, 400, 600, 800 and 1000 in Group 1 and among 1000, 2000, 3000, 4000 and 5000 in Group 2. The bias parameter in these experiments are also $V = (0.4, 0.2, 0.2, 0.2, 0.2)$.
   - When the offline sample size is fixed at 200 for each arm, the effect of varying parameter V (fixed $V(1) = 0.4$ and $V(i)$ ranging from 0.2 to 0.4 for all suboptimal arms in two groups) on the expected stopping time is further examined.

Figures 1 to 6 analyze how the average stopping time $\mathbb{E}[\tau_\delta]$ of the LUCB algorithm, and the LUCB-H algorithm under misleading, beneficial, and partial instances varies from three perspectives: $\log(1/\delta)$, offline sample size T_S, and parameter V. Across all experiments, LUCB-H behaves similarly to Pure LUCB in the misleading instance, both in terms of mean stopping time and error bars, regardless of changes in confidence level δ, offline sample size $T_S(i)$, or biased bound $V(i)$. This suggests that LUCB-H wisely discards misleading offline data. In the partial instances, LUCB-H's performance consistently lies between that of the beneficial and misleading instances. The following analysis focuses on the beneficial instance.

Figures 1 and 2 show how $\mathbb{E}[\tau_\delta]$ evolves with $\log(1/\delta)$. All curves exhibit linear or near-linear growth. The slope of each curve approximates the sample complexity of the corresponding algorithm. LUCB-H achieves sample complexity that matches that of LUCB in general. LUCB-H requires significantly fewer samples in the beneficial instance compared to Pure LUCB and LUCB-H in other instances. As $\delta \to 0$, the error bars in the beneficial instance gradually expand and converge with those of Pure LUCB, indicating a diminishing influence of historical data.

Figures 3 and 4 illustrate how the stopping time varies with offline sample size T_S. In the beneficial instance, the stopping time decreases sharply as T_S increases, demonstrating the advantage of helpful offline data. The error bars also shrink with increasing T_S, indicating that even biased beneficial data can effectively narrow the confidence bounds and accelerate convergence.

Figures 5 and 6 examine how the parameter V influences the expected stopping time. In the beneficial instance, the stopping time increases notably as V grows. This indicates that, even when historical data is advantageous, a large bound on the shift between online and offline reward distributions can adversely impact LUCB-H's performance, revealing its sensitivity to substantial shifts. Although the error bars in the beneficial case widen with increasing V, potentially affecting reliability, LUCB-H still performs comparably to Pure LUCB in this scenario.

Figures 7 and 8 summarize the sample allocation per arm for both Pure LUCB and LUCB-H in Groups 1 and 2 when $\delta = 0.01$. In the misleading case, LUCB-H assigns samples similarly to Pure LUCB, demonstrating its ability to avoid relying on misleading historical data and behave accordingly. In the beneficial instance, LUCB-H allocates significantly fewer samples per arm compared to Pure LUCB, illustrating its effectiveness in leveraging helpful historical information. In the partial scenario, LUCB-H dynamically adapts its sampling; it assigns substantially fewer samples to arms with beneficial offline rewards, while allocating a similar number as Pure LUCB to arms with misleading offline data. This showcases LUCB-H's flexibility in selectively utilizing historical data on an arm-by-arm basis. Additionally, some arms receive only one sample, as LUCB concentrates most samples on the top two arms. When both offline and initial online data strongly suggest an arm is suboptimal, LUCB-H may cease sampling further, reflecting its adaptivity. In Appendix E, we examine the robustness of LUCB-H to misspecifications in the bias bound $V(i)$.

7 CONCLUSIONS

In this paper, we studied the best arm identification problem with potentially biased offline data. We first established an impossibility result, proving that no δ-PAC algorithm can out-perform the state-of-the-art on all instances, without prior knowledge of the bias bound between offline and online distributions. This result highlights the necessity of incorporating auxiliary knowledge, such as bias bounds, to effectively utilize offline data while ensuring robust performance. To address this challenge, we proposed the LUCB-H algorithm. The algorithm adaptively balances offline and online data by calculating upper and lower confidence bounds separately with and without offline data for each arm. It selects the smaller upper confidence bound and larger lower confidence bound to obtain more accurate mean estimates. This design allows LUCB-H to decide how much to rely on offline data at each iteration. Our theoretical analysis shows that LUCB-H matches the sample complexity of standard LUCB in misleading cases. It significantly reduces the sample complexity when offline data is reliable. Furthermore, we derived an instance-dependent lower bound that matches LUCB-Hs upper bound in certain cases. Experimental results demonstrated LUCB-Hs adaptability and robustness. It effectively reduces sample complexity in favorable cases while remaining robust when offline data is unreliable.

Acknowledgements This research work is funded by the Singapore Ministry of Education AcRF Tier 2 grants (A-8000423-00-00 and A-8003063-00-00).

References

Shubhada Agrawal, Sandeep Juneja, Karthikeyan Shanmugam, and Arun Sai Suggala. Optimal best-arm identification in bandits with access to offline data. *arXiv preprint arXiv:2306.09048*, 2023.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Omar Besbes, Will Ma, and Omar Mouchtaki. Beyond iid: data-driven decision-making in heterogeneous environments. *Advances in Neural Information Processing Systems*, 35:23979–23991, 2022.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

Djallel Bouneffouf, Srinivasan Parthasarathy, Horst Samulowitz, and Martin Wistub. Optimal exploitation of clustering and history information in multi-armed bandit. *arXiv preprint arXiv:1906.03979*, 2019.

Jinzhi Bu, David Simchi-Levi, and Yunzong Xu. Online pricing with offline data: Phase transition and inverse square law. In *International Conference on Machine Learning*, pages 1202–1210. PMLR, 2020.

Xinyun Chen, Pengyi Shi, and Shanwen Pu. Data-pooling reinforcement learning for personalized healthcare intervention. *arXiv preprint arXiv:2211.08998*, 2022.

Wang Chi Cheung and Lixing Lyu. Leveraging (biased) information: Multi-armed bandits with offline data. *arXiv preprint arXiv:2405.02594*, 2024.

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of machine learning research*, 9(8), 2008.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027. PMLR, 2016.

Yonatan Gur and Ahmadreza Momeni. Adaptive sequential experiments with unknown information arrival processes. *Manufacturing & Service Operations Management*, 24 (5):2666–2684, 2022.

Botao Hao, Rahul Jain, Tor Lattimore, Benjamin Van Roy, and Zheng Wen. Leveraging demonstrations to improve online learning: Quality matters. In *International Conference on Machine Learning*, pages 12527–12545. PMLR, 2023.

Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th annual conference on information sciences and systems (CISS)*, pages 1–6. IEEE, 2014.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

Julian Katz-Samuels and Clayton Scott. Top feasible arm identification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1593–1601. PMLR, 2019.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Che-Yu Liu and Lihong Li. On the prior sensitivity of thompson sampling. In *International Conference on Algorithmic Learning Theory*, pages 321–336. Springer, 2016.

Xutong Liu, Xiangxiang Dai, Jinhang Zuo, Siwei Wang, Carlee-Joe Wong, John Lui, and Wei Chen. Offline learning for combinatorial multi-armed bandits. *arXiv preprint arXiv:2501.19300*, 2025.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

Pannagadatta Shivaswamy and Thorsten Joachims. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics*, pages 1046–1054. PMLR, 2012.

Nian Si, Fan Zhang, Zhengyuan Zhou, and Jose Blanchet. Distributionally robust batch contextual bandits. *Management Science*, 69(10):5772–5793, 2023.

Max Simchowitz, Christopher Tosh, Akshay Krishnamurthy, Daniel J Hsu, Thodoris Lykouris, Miro Dudik, and Robert E Schapire. Bayesian decision-making under misspecified priors with applications to meta-learning. *Advances in Neural Information Processing Systems*, 34: 26382–26394, 2021.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, pages 35300–35338. PMLR, 2023.

Abraham Wald. *Sequential analysis*. Courier Corporation, 2004.

Zhi Wang, Chicheng Zhang, Manish Kumar Singh, Laurel Riek, and Kamalika Chaudhuri. Multitask bandit learning through heterogeneous feedback aggregation. In *International Conference on Artificial Intelligence and Statistics*, pages 1531–1539. PMLR, 2021.

Li Ye, Yishi Lin, Hong Xie, and John Lui. Combining offline causal inference and online bandit learning for data driven decision. *arXiv preprint arXiv:2001.05699*, 2020.

Chicheng Zhang, Alekh Agarwal, Hal Daumé III, John Langford, and Sahand N Negahban. Warm-starting contextual bandits: Robustly combining supervised and bandit feedback. *arXiv preprint arXiv:1901.00301*, 2019.

Best Arm Identification with Biased Offline Data (Supplementary Material)

Le Yang[1] **Vincent Y. F. Tan**[3,1] **Wang Chi Cheung**[2]

[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2]Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore
[3]Department of Mathematics, National University of Singapore, Singapore

A PRELIMINARIES

Proposition A.1 *(Chernoff Inequality). Let $G_1, \ldots, G_m$ be independent (though not necessarily identically distributed) 1-subGaussian random variables. For any $\delta \in (0,1)$, it holds that*

$$\Pr\left[\left|\frac{1}{m}\mathbb{E}\left[\sum_{i=1}^{m} G_i\right] - \frac{1}{m}\sum_{i=1}^{m} G_i\right| \leq \sqrt{\frac{2\log(k/\delta)}{m}}\right] \geq 1 - \frac{\delta}{k},$$

We consider an identifiable class of bandit models of the form:

$$\mathcal{M} = \{v = (v_1, v_2, \ldots, v_k) : v_i \in \mathcal{P}, \mu^{\text{on}}(1) > \mu^{\text{on}}(2) \geq \ldots \geq \mu^{\text{on}}(k)\} \tag{A.1}$$

For all instances in set $\mathcal{M}$, $I^* = 1$. Let $\mathcal{A} = ((\pi_t), \tau, \hat{I}^*)$ be a δ-PAC algorithm.

Let $I_P = (P^{\text{off}}, P^{\text{on}})$ and $I_Q = (Q^{\text{off}}, Q^{\text{on}})$ be two problem instances defined over a common finite arm set A. Both instances share the same offline sample sizes $\{T_S(i)\}_{i\in A}$ and confidence level δ, but may differ in their offline and online reward distributions. Let $\tau_\delta(Q, i)$ be the (random) number of online samples drawn from arm i under instance I_Q before the stopping time τ_δ; we write $\mathbb{E}[\tau_\delta(Q, i)]$ for its expectation. The probability that an event $\mathcal{E}$ occurs under instance I_Q and instance I_P at stopping time σ are written as $\mathbb{P}_\sigma(Q, \mathcal{E})$ and $\mathbb{P}_\sigma(P, \mathcal{E})$ respectively. The Kullback-Leibler (KL) divergence between distributions P and Q is written as $\mathrm{KL}(P, Q)$, and $d(p, q)$ denotes the KL divergence measure between Bernoulli distributions with means p and q. We assume that the stopping time σ is adapted to the filtration $\{\mathcal{F}_t\}_{t\geq 0}$ and is almost surely finite. To establish the theoretical results, we first introduce the following lemma.

Lemma A.1 *Let $\mathcal{E} \in \mathcal{F}_\sigma$ be any event measurable with respect to the stopping time σ. For any non-anticipatory policy π, the following inequality holds:*

$$\sum_{i=1}^{k} \mathbb{E}[N_\sigma(i)]\,\mathrm{KL}\big(Q^{\text{on}}(i), P^{\text{on}}(i)\big) + \sum_{i=1}^{k} T_S(i)\,\mathrm{KL}\big(Q^{\text{off}}(i), P^{\text{off}}(i)\big) \geq d\big(\mathbb{P}_\sigma(Q, \mathcal{E}), \mathbb{P}_\sigma(P, \mathcal{E})\big). \tag{A.2}$$

Proof: The log-likelihood ratio for the sequence of observations observed by time t under an algorithm $\mathcal{A}$ is

$$L_t = L(\{X_s(i)\}_{s=1}^{T_S(i)}, A_1, \ldots, A_t, Y_{A_1}, \ldots, Y_{A_t}) \coloneqq \sum_{i=1}^{k}\sum_{l=1}^{t} \mathbb{1}\{A_l = i\} \log\left(\frac{Q_i^{\text{on}}(Y_{A_l})}{P_i^{\text{on}}(Y_{A_l})}\right) + \sum_{i=1}^{k}\sum_{s=1}^{T_S(i)} \log\left(\frac{Q_i^{\text{off}}(X_s)}{P_i^{\text{off}}(X_s)}\right).$$

Recall that $\{X_s(i)\}_{s=1}^{T_S(i)}$ are i.i.d. samples from the offline distribution P_i^{off}, while $\{Y_l(i)\}_{l=1}^{N_\sigma(i)}$ are i.i.d. samples from the online distribution P_i^{on}. We can rewritten L_σ as

$$L_\sigma = \sum_{i=1}^{k}\sum_{l=1}^{N_\sigma(i)} \log\left(\frac{Q_i^{\text{on}}(Y_{A_l})}{P_i^{\text{on}}(Y_{A_l})}\right) + \sum_{i=1}^{k}\sum_{s=1}^{T_S(i)} \log\left(\frac{Q_i^{\text{off}}(X_s)}{P_i^{\text{off}}(X_s)}\right).$$

Since $\mathbb{E}_Q\left[\log\left(\frac{Q_i^{\text{on}}(Y_{A_l})}{P_i^{\text{on}}(Y_{A_l})}\right)\right] = \text{KL}(Q^{\text{on}}(i), P^{\text{on}}(i))$ and $\mathbb{E}_Q\left[\log\left(\frac{Q_i^{\text{off}}(X_s)}{P_i^{\text{off}}(X_s)}\right)\right] = \text{KL}(Q^{\text{off}}(i), P^{\text{off}}(i))$, by Wald's Lemma (Wald, 2004),

$$\mathbb{E}_Q[L_\sigma] = \sum_{i=1}^{k} \mathbb{E}[N_\sigma(i)]\,\text{KL}\big(Q^{\text{on}}(i), P^{\text{on}}(i)\big) + \sum_{i=1}^{k} T_S(i)\,\text{KL}\big(Q^{\text{off}}(i), P^{\text{off}}(i)\big). \tag{A.3}$$

By Lemma 19 in Kaufmann et al. (2016), we have

$$\mathbb{E}_Q[L_\sigma] \geq d\big(\mathbb{P}_\sigma(Q,\mathcal{E}), \mathbb{P}_\sigma(P,\mathcal{E})\big). \tag{A.4}$$

Combining (A.3) and (A.4) completes the proof.

B PROOF OF PROPOSITION 3.1

Proof: In the instance I_P, arm 1 is the best arm, and in the instance I_Q, arm 2 is the best arm. In both instances I_P, I_Q, the DM requires $\mathbb{E}[\tau'_\delta] = \Omega(\log(1/\delta)/\delta^{2\beta})$ pulls (without offline data) to be δ-PAC. Let σ be any almost surely finite stopping time with respect to $\mathcal{F}_t$ on instance I_Q. Introducing the event $\mathcal{E} = (\hat{I}^* = 2) \in \mathcal{F}_\sigma$, any δ-PAC algorithm satisfies $\mathbb{P}_\sigma(Q,\mathcal{E}) \geq 1-\delta$ and $\mathbb{P}_\sigma(P,\mathcal{E}) < \delta/2$, where $\mathbb{P}_\sigma(P,\mathcal{E})$ and $\mathbb{P}_\sigma(Q,\mathcal{E})$ are the probabilities of the event $\mathcal{E}$ occurring on instances I_P and I_Q separately in round σ. Denote $\mathbb{E}_\sigma(Q,i)$ as the expectation number of samples allocated to arm i until round σ on instance I_Q, where $i \in \{1,2\}$.

By Lemma A.1, we have

$$d(\mathbb{P}_\sigma(Q,\mathcal{E}), \mathbb{P}_\sigma(P,\mathcal{E})) \leq \sum_{i=1}^{2} \mathbb{E}_\sigma(Q,i)\text{KL}(Q^{\text{on}}(i), P^{\text{on}}(i)) + \sum_{i=1}^{2} T_S(i)\text{KL}(Q^{\text{off}}(i), P^{\text{off}}(i)),$$

where the function $d(x,y) := x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the binary relative entropy, with the convention that $d(0,0) = d(1,1) = 0$. Then the following inequality holds,

$$\mathbb{E}_\sigma(Q,2) \geq \frac{d(1-\delta, \frac{\delta}{2}) - T_S(2)\text{KL}(Q^{\text{off}}(2), P^{\text{off}}(2))}{\text{KL}(Q^{\text{on}}(2), P^{\text{on}}(2))}$$

due to the fact that $Q_1^{\text{off}} = P_1^{\text{off}}$ and $Q_1^{\text{on}} = P_1^{\text{on}}$. The monotonicity properties of $d(x,y)$ is increasing when $x > y$ and decreasing when $x < y$ yield $d(1-\delta, \delta/2) > d(1-\delta,\delta)$. Hence, we can find an $\epsilon' > 0$ such that $d(1-\delta,\delta/2) > \delta^{-\epsilon'} d(1-\delta,\delta)$. Then

$$\begin{aligned}\mathbb{E}_\sigma(Q,2) &\geq 2\delta^{-2\beta-\epsilon'}\log\left(\frac{1}{2.4\delta}\right) - \frac{T_S(2)\text{KL}(Q^{\text{off}}(2), P^{\text{off}}(2))}{\text{KL}(Q^{\text{on}}(2), P^{\text{on}}(2))} \\ &\geq 2\delta^{-2\beta-\epsilon'}\log\left(\frac{1}{2.4\delta}\right) - \delta^{-2\beta-\epsilon'}\log\left(\frac{1}{\delta}\right) = \Omega\left(\delta^{-2\beta-\epsilon'}\log\left(\frac{1}{\delta}\right)\right).\end{aligned}$$

The second inequality holds by the claim assumption that we can find $\epsilon'' > 0$ such that $T_S(2) \leq \frac{C}{2}(\delta^{-2\beta} - \delta^{-2\beta+\epsilon''})$. Let $\epsilon = \{\epsilon', \epsilon'', \beta\}$. Hence we have

$$\mathbb{E}[\tau_\delta(Q)] \geq \Omega\left(\delta^{-2\beta-\epsilon}\log\left(\frac{1}{\delta}\right)\right)$$

as desired.

C PROOF OF THEOREM 4.1

To prove Theorem 4.1, we first state a lemma.

Lemma C.1 *The following events hold over all rounds $t = 1, 2, \ldots$ with probability at least $1-\delta/k$:*

$$\left|\min(\text{UCB}_t^S(i), \text{UCB}_t(i)) - \mu^{\text{on}}(i)\right| \leq \min\left(2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}} + \frac{T_S(i)}{N_t(i)+T_S(i)}\cdot\eta(i), 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)}}\right)$$

and

$$\left|\mu^{\text{on}}(i) - \max(\text{LCB}_t^S(i), \text{LCB}_t(i))\right| \leq \min\left(2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}} + \frac{T_S(i)}{N_t(i)+T_S(i)} \cdot \eta(i), 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)}}\right).$$

Proof: To prove this Lemma, it suffices to prove that

$$\mathbb{P}\left\{\mu^{(\text{on})}(i) \leq \text{UCB}_t^S(i) \leq \mu^{(\text{on})}(i) + 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}} + \frac{T_S(i)}{N_t(i)+T_S(i)} \cdot \eta(i)\right\} \geq 1 - \frac{\delta}{kt}.$$

Indeed, the inequality implies

$$\mathbb{P}\left\{\mu^{(\text{on})}(i) \leq \text{UCB}_t^S(i) \leq \mu^{(\text{on})}(i) + 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)}}\right\} \geq 1 - \frac{\delta}{kt}$$

by setting $T_S(i) = 0$.

$$\begin{aligned}
&\mathbb{P}[\mu^{(\text{on})}(i) \leq \text{UCB}_t^S(i)] \\
&= \mathbb{P}\left[\mu^{(\text{on})}(i) \leq \frac{N_t(i)\cdot\hat{Y}_t(i) + T_S(i)\cdot\hat{X}(i)}{N_t(i)+T_S(i)} + \sqrt{\frac{2\log(kt/\delta)}{N_t^+(i)+T_S(i)}} + \frac{T_S(i)}{N_t^+(i)+T_S(i)} \cdot V(i)\right] \\
&= \mathbb{P}\left[\frac{N_t(i)\mu^{(\text{on})}(i) + T_S(i)\mu^{(\text{off})}(i)}{N_t(i)+T_S(i)} \frac{T_S(i)(\mu^{(\text{on})}(i) - \mu^{(\text{off})}(i))}{N_t(i)+T_S(i)} \leq \frac{N_t(i)\cdot\hat{Y}_t(i) + T_S(i)\cdot\hat{X}(i)}{N_t(i)+T_S(i)}\right. \\
&\quad \left. + \sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}} + \frac{T_S(i)}{N_t(i)+T_S(i)} \cdot V(i)\right] \\
&\geq \mathbb{P}\left[\frac{N_t(i)\mu^{(\text{on})}(i) + T_S(i)\mu^{(\text{off})}(i)}{N_t(i)+T_S(i)} \leq \frac{N_t(i)\cdot\hat{Y}_t(i) + T_S(i)\cdot\hat{X}(i)}{N_t(i)+T_S(i)} + \sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}}\right] \\
&\geq \mathbb{P}_{Y_i \sim P^{(\text{on})}(i)}\left[\frac{n\mu^{(\text{on})}(i) + T_S(i)\mu^{(\text{off})}(i)}{n+T_S(i)} \leq \frac{\sum_{i=1}^n Y_i + T_S(i)\cdot\hat{X}(i)}{n+T_S(i)} + \sqrt{\frac{2\log(kt/\delta)}{n+T_S(i)}} \quad \text{for} \quad n = 1, 2, \ldots, t\right] \\
&\geq 1 - \frac{\delta}{k}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}\left(\text{UCB}_t^S(i) \leq \mu^{(\text{on})}(i) + 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)+T_S(i)}} + \frac{T_S(i)}{N_t(i)+T_S(i)} \cdot \eta(i)\right) \\
&= \mathbb{P}\left(\frac{N_t(i)\cdot\hat{Y}_t(i) + T_S(i)\cdot\hat{X}(i)}{N_t(i)+T_S(i)} \leq \mu^{(\text{on})}(i) + \sqrt{\frac{2\log(kt/\delta)}{N_t^+(i)+T_S(i)}} + \frac{T_S(i)\cdot(\mu^{(\text{off})}(i) - \mu^{(\text{on})}(i))}{N_t^+(i)+T_S(i)}\right) \\
&= \mathbb{P}\left(\frac{N_t(i)\cdot\hat{Y}_t(i) + T_S(i)\cdot\hat{X}(i)}{N_t(i)+T_S(i)} \leq \frac{N_t(i)\mu^{(\text{on})}(i) + T_S(i)\mu^{(\text{off})}(i)}{N_t(i)+T_S(i)} + \sqrt{\frac{2\log(kt/\delta)}{N_t^+(i)+T_S(i)}}\right) \\
&\geq \mathbb{P}_{Y_i \sim P^{(\text{on})}(i)}\left[\frac{\sum_{i=1}^n Y_i + T_S(i)\cdot\hat{X}(i)}{n+T_S(i)} \leq \frac{n\mu^{(\text{on})}(i) + T_S(i)\mu^{(\text{off})}(i)}{n+T_S(i)} + \sqrt{\frac{2\log(kt/\delta)}{n+T_S(i)}} \quad \text{for} \quad n = 1, 2, \ldots, t\right] \\
&\geq 1 - \frac{\delta}{k}.
\end{aligned}$$

Then a union bound on the failure probabilities for $i \in A$ establishes the Lemma. We can obtain a similar bound for $\left|\mu^{\text{on}}(i) - \max(\text{LCB}_t^S(i), \text{LCB}_t(i))\right|$. Clearly, the stopping condition guarantees that the probability of identifying the best arm is no less than $1 - \delta$.

In the following, we bound the total number of measurements. Define $c = \frac{1}{2}[\mu^{\text{on}}(1) + \mu^{\text{on}}(2)]$. We say that arm i^* is BAD in round t, if $\text{LCB}_t(i^*) < c$ and $\text{LCB}_t^S(i^*) < c$, and an arm $i \neq i^*$ is BAD in round t, if $\text{UCB}_t(i) > c$ and $\text{UCB}_t^S(i) > c$. We claim that for all time $t \geq 1$,

$$\big\{\text{Events in Lemma } C.1 \text{ hold }\big\} \cap \big\{ \max\{\text{LCB}_t(i^*), \text{LCB}_t^S(i^*)\} < \min\{\text{UCB}_t(i), \text{UCB}_t^S(i)\}\big\} \\ \Rightarrow \{h_t \text{ is BAD }\} \cap \{l_t \text{ is BAD }\}.$$

Assume that the events in Lemma C.1 hold, then for any $i \neq i^*$ and $s \geq \tau_i$, where τ_i is the first integer such that arm i is not BAD. Define $\tau_{i^*} = \tau_2$.

$$\begin{aligned}\text{UCB}_t(i) &\leq \mu^{(\text{on})}(i) + 2\,\text{rad}_t(i) \\ &= c + 2\,\text{rad}_t(i) + \frac{(\mu^{\text{on}}(i) - \mu^{\text{on}}(i^*)) + (\mu^{\text{on}}(i) - \mu^{\text{on}}(2))}{2} \\ &\leq c + 2\,\text{rad}_t(i) - \frac{\Delta_i}{2} \leq c,\end{aligned}$$

where $\Delta_i = \mu^{\text{on}}(i^*) - \mu^{\text{on}}(i)$.

$$\begin{aligned}\text{UCB}_t^S(i) &\leq \mu^{(\text{on})}(i) + \text{rad}_t^S(i) + \left[\sqrt{\frac{2\log(kt/\delta)}{N_t(i) + T_S(i)}} + \frac{T_S(i) \cdot \big(\mu^{(\text{off})}(i) - \mu^{(\text{on})}(i)\big)}{N_t(i) + T_S(i)}\right] \\ &= c - \frac{\Delta_i}{2} + 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i) + T_S(i)}} + \frac{T_S(i)}{N_t(i) + T_S(i)} \cdot \eta(i) \leq c.\end{aligned}$$

Since $\min\{\text{UCB}_s(i), \text{UCB}_t^S(i)\} \leq c$, we have

$$\min\left\{2\sqrt{\frac{2\log(kt/\delta)}{N_t(i) + T_S(i)}} + \frac{T_S(i)}{N_t(i) + T_S(i)} \cdot \eta(i), 2\sqrt{\frac{2\log(kt/\delta)}{N_t(i)}}\right\} \leq \frac{\Delta_i}{2}.$$

Solving for $N_t(i)$, we find that

$$N_t(i) > \frac{32}{\Delta_i^2}\log\left(\frac{k}{\delta}\right) - T_S(i) \cdot \max\left\{1 - \frac{4\eta(i)}{\Delta_i}, 0\right\}.$$

Since Lemma C.1 holds, the total number of rounds is observed to be no greater than

$$\begin{aligned}\sum_{t=1}^{\infty} \mathbb{1}\{h_t \text{ is BAD or } l_t \text{ is BAD}\} &= \sum_{t=1}^{\infty}\sum_{i=1}^{n} \mathbb{1}\left\{\{h_t = i \text{ or } l_t = i\} \cap \{i \text{ is } \textit{BAD}\}\right\} \\ &\leq \sum_{t=1}^{\infty}\sum_{i=1}^{n} \mathbb{1}\left\{\{h_t = i \text{ or } l_t = i\} \cap \{N_t(i) \leq \tau_i\}\right\} \leq \sum_{i=1}^{n} \tau_i\end{aligned}$$

Then LUCB-H algorithm obtains a sample complexity of order in Theorem 4.1.

D PROOF OF THEOREM 5.1

Fix $a > 0$. For all $i \in \{1, 2, \ldots, k\}$, from Assumption 5.1 there exists an alternative model

$$v' = (v_1, \ldots, v_{i-1}, v_i', v_{i+1}, \ldots, v_k)$$

in which the only arm modified is arm i, $i \neq 1$ and v_i' is such that:

$$\text{KL}(v_i, v_1) < \text{KL}(v_i, v_i') < \text{KL}(v_i, v_1) + a, \quad \mu^{\text{on}'}(i) > \mu^{\text{on}'}(1)$$

and

$$\mu^{\text{off}'}(i) = \begin{cases}\mu^{\text{off}}(i), & \text{if } \mu^{\text{off}}(i) > \mu^{\text{off}}(1) \\ \mu^{\text{off}}(1), & \text{if } \mu^{\text{off}}(i) < \mu^{\text{off}}(1).\end{cases}$$

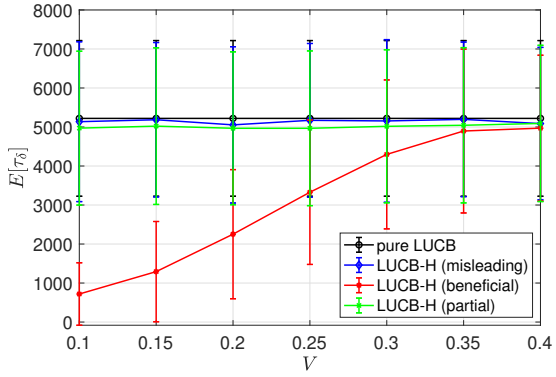

Figure 9: Evolution of $\mathbb{E}[\tau_\delta]$ with V in group 2.

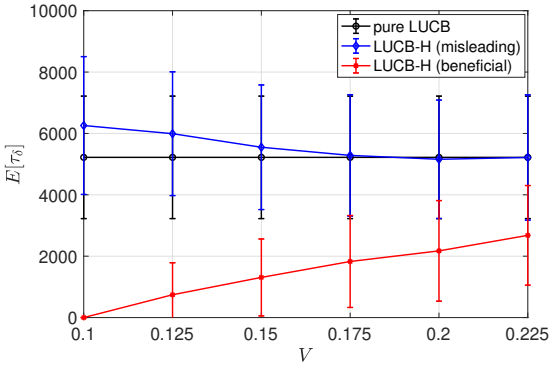

Figure 10: Evolution of $\mathbb{E}[\tau_\delta]$ with V in group 2.

In particular, on the bandit model v' the best arm is no longer arm 1. Introducing the event $\mathcal{E} = \{\hat{I}^* = 1\} \in \mathcal{F}_{\tau_\delta}$, any δ-PAC algorithm satisfies $\mathbb{P}_{\tau_\delta}(v, \mathcal{E}) \geq 1 - \delta$ and $\mathbb{P}_{\tau_\delta}(v', \mathcal{E}) \leq \delta$. By Lemma A.1, we have

$$\sum_{i=1}^{k} \mathbb{E}_v[N_{\tau_\delta}(i)]\mathrm{KL}(v^{\mathrm{on}}(i), v^{\mathrm{on}'}(i)) + \sum_{i=1}^{k} T_S(i)\mathrm{KL}(v^{\mathrm{off}}(i), v^{\mathrm{off}'}(i)) \geq d\big(\mathbb{P}_{\tau_\delta}(v, \mathcal{E}), \mathbb{P}_{\tau_\delta}(v', \mathcal{E})\big) \geq d(1-\delta, \delta).$$

Then the following holds

$$\sum_{i=1}^{k} \mathbb{E}_v[N_{\tau_\delta}(i)]\mathrm{KL}(v^{\mathrm{on}}(i), v^{\mathrm{on}'}(i)) + \sum_{i=1}^{k} T_S(i)\mathrm{KL}(v^{\mathrm{off}}(i), v^{\mathrm{off}'}(i)) \geq \log\left(\frac{1}{2.4\delta}\right),$$

due to the fact that $d(1-\delta, \delta) \geq \log(1/(2.4\delta))$. Thus

$$\begin{aligned}
\mathbb{E}_v[N_{\tau_\delta}(i)] &\geq \frac{\log(1/(2.4\delta)) - T_S(i)\mathrm{KL}(v^{\mathrm{off}}(i), v^{\mathrm{off}'}(i))}{\mathrm{KL}(v^{\mathrm{on}}(i), v^{\mathrm{on}'}(i)) + a} \\
&= \frac{\log(1/(2.4\delta))}{\mathrm{KL}(v^{\mathrm{on}}(i), v^{\mathrm{on}'}(i)) + a} - T_S(i)\frac{\mathrm{KL}(v^{\mathrm{off}}(i), v^{\mathrm{off}'}(i))}{\mathrm{KL}(v^{\mathrm{on}}(i), v^{\mathrm{on}'}(i)) + a} \\
&= \frac{\log(1/(2.4\delta))}{\mathrm{KL}(v^{\mathrm{on}}(i), v^{\mathrm{on}'}(i)) + a} - T_S(i) \cdot \max\left\{\frac{\mu^{\mathrm{off}}(1) - \mu^{\mathrm{off}}(i)}{\Delta_i}, 0\right\}^2
\end{aligned}$$

The last equality holds because $\mathrm{KL}(v^{\mathrm{off}}(i), v^{\mathrm{off}'}(i)) = \max\{0, \mu^{\mathrm{off}}(1) - \mu^{\mathrm{off}}(i)\}^2/2$. Letting a tend to zero and summing over the arms yields the bound on $\mathbb{E}_v[\tau_\delta] = \sum_{i=1}^{k} \mathbb{E}_v[N_{\tau_\delta}(i)]$.

E ROBUSTNESS TO MISSPECIFICATIONS OF THE BIAS BOUND $V(i)$

The LUCB-H algorithm relies on prior knowledge of an auxiliary bias bound $V(i)$, which serves as an upper bound on the mean shift between the offline and online reward distributions. That is, $V(i) \geq |\mu^{\mathrm{on}}(i) - \mu^{\mathrm{off}}(i)|$ for all arms i. In many real-world applications, it is possible to approximate or conservatively upper bound $V(i)$ before the online phase begins. For instance, in recommendation systems or clinical trials, offline and online data often originate from similar but not identical distributions. Practitioners may exploit prior logs, domain expertise, or distribution shift estimation techniques to estimate such discrepancies. We have provided a brief discussion of potential empirical strategies in the last paragraph of Page 2.

Section 6 has already demonstrated the strong empirical performance of LUCB-H under correctly specified bias bounds. In this section, we investigate how LUCB-H performs when $V(i)$ is misestimated, particularly when it is underestimated. This corresponds to the case where $V(i) < |\mu^{\mathrm{on}}(i) - \mu^{\mathrm{off}}(i)|$, which could potentially compromise the algorithms ability to properly discount misleading offline data.

To assess the sensitivity of LUCB-H to misestimation of $V(i)$, we conducted an additional experiment in Group 2 (see Figure 9), where we set $V(i) \in \{0.1, 0.15\}$ for suboptimal arms while the true discrepancy was 0.2. The results show that

even with $V(i)$ set below the true bias, LUCB-H remains stable and competitive. This suggests that LUCB-H is empirically robust to mild underestimation of the bias.

To better understand when such underestimation can lead to performance degradation, we provide the following analysis. In the extreme case where $V(i) = 0$, LUCB-H fully trusts offline data and assumes perfect alignment between offline and online distributions. Consequently, the selection rule $\min\{\mathrm{UCB}_t(i), \mathrm{UCB}_t^S(i)\}$ may favor the offline-based bound $\mathrm{UCB}_t^S(i)$, causing the algorithm to overly rely on historical samples. This behavior is particularly problematic if arm i is the true best arm and the offline mean is significantly biased, leading to suboptimal performance. On the other hand, if arm i is suboptimal, relying on biased offline estimates may reduce its chances of being pulled, which could actually help reduce sample complexity.

In Group 2's previous experiments, we fixed $V(1) = 0.4$ for the best arm and only varied $V(i)$ for suboptimal arms. Since $V(1)$ was not underestimated, LUCB-H maintained its effectiveness. To further investigate the impact of bias misestimation on the best arm, we conducted a follow-up experiment in which all arms share the same bias bound, varying $V(i) \in \{0.1, 0.125, 0.15, 0.175, 0.2, 0.225\}$ for $i = 1, 2, \ldots, 5$ (see Figure 10). This setup allows us to explore both underestimation and overestimation scenarios.

The results confirm that LUCB-H is robust when $V(i)$ is overestimated: its performance remains comparable to that of Pure LUCB. However, when $V(i)$ is underestimated, especially for the best arm, LUCB-H may be misled by biased offline data and underperform relative to the pure online baseline LUCB. These findings provide empirical support for the theoretical necessity of a valid bias bound, as emphasized in our impossibility result in Proposition 3.1. Without reliable information to distinguish helpful from misleading historical data, no algorithm can consistently outperform the conservative online LUCB baseline across all problem instances. A valid $V(i)$ enables LUCB-H to safely incorporate historical data when appropriate and revert to cautious online behavior when necessary.