Best-Arm Identification in Correlated Multi-Armed Bandits

Samarth Gupta, Gauri Joshi, *Member, IEEE*, and Osman Yağan, *Senior Member, IEEE*

***Abstract*—In this paper we consider the problem of best-arm identification in multi-armed bandits in the fixed confidence setting, where the goal is to identify, with probability $1-\delta$ for some $\delta > 0$, the arm with the highest mean reward in minimum possible samples from the set of arms $\mathcal{K}$. Most existing best-arm identification algorithms and analyses operate under the assumption that the rewards corresponding to different arms are independent of each other. We propose a novel correlated bandit framework that captures domain knowledge about correlation between arms in the form of upper bounds on expected conditional reward of an arm, given a reward realization from another arm. Our proposed algorithm C-LUCB, which generalizes the LUCB algorithm utilizes this partial knowledge of correlations to sharply reduce the sample complexity of best-arm identification. More interestingly, we show that the total samples obtained by C-LUCB are of the form $O(\sum_{k\in\mathcal{C}} \log(\frac{1}{\delta}))$ as opposed to the typical $O(\sum_{k\in\mathcal{K}} \log(\frac{1}{\delta}))$ samples required in the independent reward setting. The improvement comes, as the $O(\log(1/\delta))$ term is summed only for the set of *competitive* arms $\mathcal{C}$, which is a subset of the original set of arms $\mathcal{K}$. The size of the set $\mathcal{C}$, depending on the problem setting, can be as small as 2, and hence using C-LUCB in the correlated bandits setting can lead to significant performance improvements. Our theoretical findings are supported by experiments on the Movielens and Goodreads recommendation datasets.**

***Index Terms*—Multi-armed bandits, online learning, sequential decision making, sample complexity analysis.**

I. Introduction

THE MULTI-ARMED *bandit* (MAB) problem falls under the class of sequential decision making problems. In the classical multi-armed bandit setting, the player is asked to sample one of the K arms at every round $t = 1, 2, \ldots$ Upon sampling arm k_t at round t, the player receives a *random* reward R_t drawn from the reward distribution of arm k_t. These reward distributions are assumed to be *unknown* to the player, and the most commonly studied objective is to maximize the *long-term* cumulative reward; e.g., see the early work by Lai and Robbins [1]. Since then, the reward maximization problem has received attention in both classical settings [2], [3] and in variants of the classical multi-armed bandits such as linear [4], contextual [5], structured bandits [6] etc.

Best-arm Identification in Bandits with Independent Arms: Instead of maximizing the cumulative reward, an alternative objective in the Multi-Armed Bandit setting is to identify the *best arm* (i.e., the arm with the largest mean reward) from as few samples as possible. While reward maximization has been studied extensively, the best-arm identification problem is seldom explored in settings outside of the *classical MAB framework*, i.e., the setting where rewards corresponding to different arms are independent of each other. The best-arm identification problem can be formulated in two different ways, namely fixed confidence [7] and fixed budget [8]. In the fixed confidence setting, the player is provided with a *confidence parameter* δ and their goal is to achieve the fastest (i.e., with the least number of samples) possible identification of the best arm with a probability of at least $1-\delta$. In the fixed budget setting, the number of samples that the player can receive is fixed, and the goal is to identify the best arm with the highest possible confidence. In this paper, we focus on the fixed confidence setting.

The best arm identification problem has been explored in the classical MAB framework [9], [10], [11], [12], [13], [14], [15] and three distinct approaches have shown promise, namely, the racing/successive elimination, law of iterated logarithm upper confidence bound (lil'UCB) and lower and upper confidence bound (LUCB) based approaches. These algorithms maintain upper and lower confidence bound indices for each arm and usually stop once the lower confidence index of one arm becomes larger than upper confidence bound of all other arms (discussed in more detail in Section III). These three approaches differ in their approach of sampling arms. The successive elimination approach samples arms in a round robin manner, lil'UCB samples the arm with the largest upper confidence bound index at round t and LUCB samples two distinct arms at each round, first it samples the arm with the largest empirical mean and then amongst the rest it samples an arm with the largest upper confidence bound index.

These best-arm identification algorithms have found their use in a wide variety of application settings, such as clinical trials [16], ad-selection campaigns [17], crowd-sourced ranking [11] and hyperparameter optimization [18] by treating different drugs/treatments, advertisements, items to be ranked

Manuscript received October 1, 2020; revised April 26, 2021 and May 13, 2021; accepted May 16, 2021. Date of publication May 20, 2021; date of current version June 21, 2021. This work was supported in part by NSF under Grant CCF-1840860 and Grant CCF-2007834; in part by Siebel Energy Institute; in part by Carnegie Bosch Institute; in part by Manufacturing Futures Initiative; and in part by CyLab IoT Initiative. The work of Samarth Gupta was supported in part by CyLab Presidential Fellowship and in part by David H. Barakat and LaVerne Owen-Barakat CIT Dean's Fellowship. *(Corresponding author: Samarth Gupta.)*

The authors are with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: samarthg@andrew.cmu.edu; gaurij@andrew.cmu.edu; oyagan@andrew.cmu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSAIT.2021.3082028, provided by the authors.

Digital Object Identifier 10.1109/JSAIT.2021.3082028

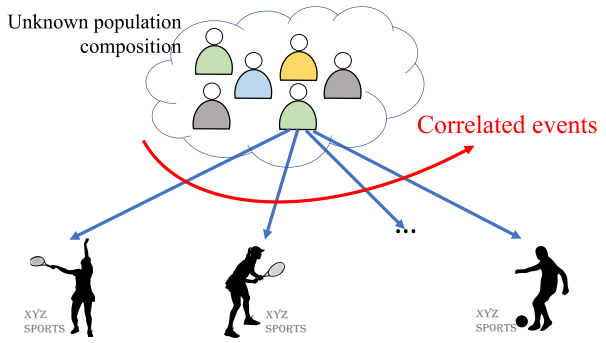

Fig. 1. The ratings of a user corresponding to different versions of the same ad are likely to be correlated. For example, if a person likes first version, there is a good chance that they will also like the 2nd one as it also related to tennis. However, the population composition is unknown, i.e., the fraction of people liking the first/second or the last version is unknown.

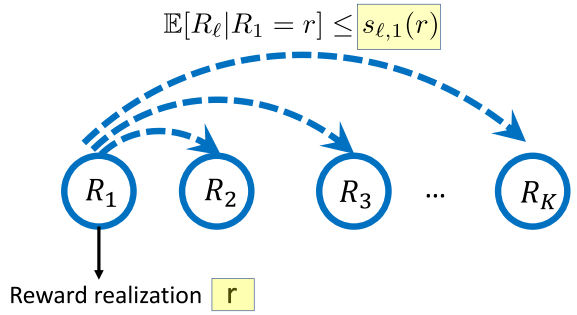

Fig. 2. Upon observing a reward r from an arm k, pseudo-rewards $s_{\ell,k}(r)$, give us an upper bound on the conditional expectation of the reward from arm ℓ given that we observed reward r from arm k. These pseudo-rewards models the correlation in rewards corresponding to different arms.

and hyperparameters as the arms in the multi-armed bandit problem.

Best-arm Identification when Rewards are Correlated across arms: The aforementioned best-arm identification algorithms all operate under the assumption that the rewards from different arms are independent of each other; e.g., at a given round t, the reward obtained from arm k does not provide any information about the reward that one might have received if they sampled another arm ℓ. However, this may not be the case in many applications of MABs. For instance, the response of a user for different advertisements in an ad-campaign is likely to be correlated as the ad designs may be related or starkly different with each other (see Figure 1). One way to learn these correlations would be to pull multiple arms at each round t. Since this is not allowed in the standard MAB setup, we assume that *partial* information about such correlations is available a priori. In practice, the presence of such correlations may be known beforehand either through domain expertise or through controlled studies where each user is presented with multiple arms. For example, before starting ad campaign, partial information may be known about the *expected* reward we would receive from a user by showing that ad version ℓ, given their response to version k. A similar argument can be made in the application domain of clinical trials, namely in identifying the best drug for an unknown disease. There, the effect of different drugs on an individual may be correlated if the drugs share similar or contrasting components among them. In this context, the correlations would be expected to be known by the domain expertise of the physicians involved.

The current best-arm identification algorithms cannot leverage these correlations to reduce the number of samples required in identifying the best arm. This papers aims to fill this gap in the literature through a new MAB model introduced next.

A Novel Correlated MAB model: Motivated by this, we consider a multi-armed bandit framework where rewards corresponding to different arms are correlated. We model the *partial* knowledge of correlations through *pseudo-rewards* that represent upper bounds on the conditional mean rewards. The pseudo-rewards provide us an upper bound on the expected reward from arm ℓ, given that the response from arm k was r (See Figure 2), i.e.,

$$\mathbb{E}[R_\ell | R_k = r] \leq s_{\ell,k}(r). \tag{1}$$

A key advantage of this model is that pseudo-rewards are just *upper bounds* on the conditional expected reward and they can be arbitrarily loose. In the case where all bounds are trivial, our framework reduces to that of the classical Multi-armed bandit setting. This model was first proposed by us in [19], where we studied the problem of reward maximization. Two seemingly related models are the structured [20], [21] and contextual [5] multi-armed bandit models.

Comparison with Contextual and Structured bandits: In contextual bandits, the context features of the user (i.e., the user to whom ad is recommended) are assumed to be known, and the goal is to learn a mapping from the context features to the expected rewards so that each user can be given a *personalized* recommendation. In contrast, our model focuses on a setting where context features of the users are *not* known and the goal is to find a *single* recommendation for the *entire* demographic. Our work falls under the class of structured bandits, which in its full generality, poses restrictions on the joint probability distribution of rewards. To the best of our knowledge, existing work on best-arm identification in structured bandits focus on settings where mean rewards of the arms are related to one another through a hidden parameter θ. In particular, the mean reward of arm k is $\mu_k(\theta)$, where θ is a hidden parameter common to all K arms. It assumes that the mean reward mappings $\mu_k(\theta)$ are known beforehand, but the hidden parameter is unknown. While the mean rewards are related to one another in these works, the rewards are not necessarily correlated. A more detailed comparison is presented in Section III. In this work, we explicitly model the correlation through knowledge of pseudo-rewards.

Proposed C-LUCB Algorithm and its Sample Complexity: After establishing a correlated bandit model, we then focus on designing best-arm identification algorithms, that are able to make use of this correlation information to identify the best-arm in fewer samples than the classical best-arm identification algorithms. In particular, we propose an approach that makes use of the pseudo-reward information and extends the LUCB approach to the correlated bandit setting. Our sample complexity analysis shows that the proposed C-LUCB approach is able to explore certain arms without explicitly sampling them. Due

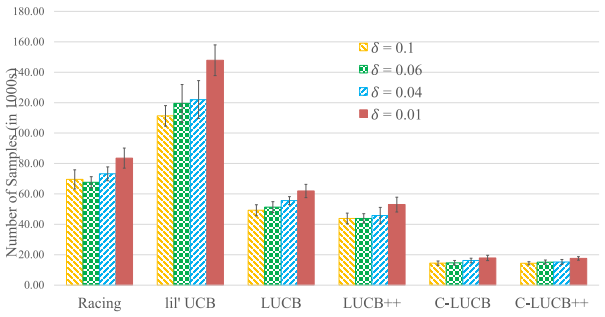

Fig. 3. This plot illustrates the number of samples required by different algorithms to identify the best movie genre out of the 18 possible movie genres in the Movielens dataset with confidence $1-\delta$. As δ decreases, the algorithms need more samples to identify the best arm. As our proposed C-LUCB and C-LUCB++ algorithms utilize correlation information, they identify the best arm in fewer samples relative to Racing, lil'UCB, LUCB and LUCB++.

to this, we see that these arms, termed as non-competitive contribute only an O(1) term in the sample complexity as to the typical O($\log \frac{1}{\delta}$) contribution by each arm. As a result of this, we are able to provide better sample complexity results than LUCB in the correlated bandit setting. In particular, the LUCB algorithm stops with probability $1-\delta$ after obtaining at most $\sum_{k\in\mathcal{K}} \frac{2\zeta}{\Delta_k^2}(\log(\frac{K\log(\frac{1}{\Delta_k^2})}{\delta}))$ samples, where $\Delta_k = \mu_{k^*} - \mu_k$, i.e., the difference in mean reward of optimal arm k^* and mean reward of arm k and $\Delta_{k^*} = \min_{k\neq k^*} \Delta_k$, i.e., the gap between best and second best arm and $\zeta > 0$ is a constant. The C-LUCB stops after at most $\sum_{k\in\mathcal{C}} \frac{2\zeta}{\Delta_k^2}(\log(\frac{2K\log(\frac{1}{\Delta_k^2})}{\delta})) + \mathrm{O}(1)$ samples with probability $1-\delta$. Here, $\mathcal{C} \subseteq \mathcal{K}$ with $2 \leq |\mathcal{C}| \leq K$ depending on the problem instance. As the size of the set $\mathcal{C}$ can be smaller than $\mathcal{K}$, we improve upon the sample complexity results of standard approaches of best-arm identification. This theoretical advantage gets reflected in our experiments on two real-world recommendation datasets, namely, Movielens and Goodreads. For instance, Figure 3 illustrates the performance of our proposed algorithms in a correlated bandit framework, where the goal is to identify the best movie genre from the set of 18 movie genres in the Movielens dataset. As our proposed approach utilizes the correlations in the problem, they draw fewer samples than the Racing, lil'UCB and the LUCB based approaches.

Organization of the rest of the paper: In Section II of this paper, we present a new multi-armed bandit framework, where correlation between arms is captured in the form of pseudo-rewards. We also discuss how pseudo-rewards can be computed in practical settings in Section II. In Section III, we review state-of-the-art best-arm identification algorithms such as successive elimination (or racing), lil'UCB, and LUCB designed for the classical (independent arm) framework. We also discuss how our proposed correlated multi-armed bandit framework compares with the structured and linear bandit frameworks that have been studied previously. In Section IV we propose the C-LUCB algorithm, and compare it with state-of-the-art approaches. We discuss several variants of C-LUCB in Section VI. In Section V we analyze the sample complexity analysis of C-LUCB and discuss its proof technique and implications. This analysis reveals that utilizing correlations can lead to significant reduction in the number of samples required to identify the best-arm. Finally, in Section VII we demonstrate the practical applicability our proposed model and algorithm via extensive experiments on real-world recommendation datasets.

II. The Correlated Multi-Armed Bandit Model

A. Problem Formulation

Consider a Multi-Armed Bandit setting with K arms $\{1, 2, \ldots K\}$. At each round t, we sample an arm $k_t \in \mathcal{K}$ and receive a random reward $R_{k_t} \in [0, b]$. Among the set of K arms, we denote the arm with the largest mean reward as the *best-arm* k^*, i.e., $k^* = \arg\max_{k\in\mathcal{K}} \mu_k$. In the *fixed-confidence* setting [7], the objective is to identify the best-arm in as few samples as possible. In particular, given $\delta > 0$, the goal is to devise a sampling strategy that stops at some round T (a random variable) and declares an arm k^{out} as the optimal arm, where,

$$\Pr\left(k^{\text{out}} = k^*\right) \geq 1 - \delta.$$

Put differently, we aim to find the best arm with probability at least $1-\delta$ while minimizing the total *number of samples* drawn from the arms. We note that the number of samples can be different from the number of rounds T as some algorithms (e.g., LUCB, Racing) sample multiple arms in one round. Using the total number of samples drawn until round T allows us to compare them fairly against algorithms that draw only one sample at each round t (e.g., lil'UCB).

The classical multi-armed bandit setting implicitly assumes that the rewards $R_1, R_2, \ldots, R_K$ are independent. That is, $\Pr(R_\ell = r_\ell | R_k = r) = \Pr(R_\ell = r_\ell) \quad \forall r_\ell, r$ and $\forall \ell, k$, which implies that, $\mathbb{E}[R_\ell | R_k = r] = \mathbb{E}[R_\ell] \quad \forall r, \ell, k$. Motivated by the fact that rewards of a user corresponding to different arms might be correlated, we consider a setup where $f_{R_\ell|R_k}(r_\ell|r_k) \neq f_{R_\ell}(r_\ell)$, with $f_{R_\ell}(r_\ell)$ denoting the probability distribution function of the reward from arm ℓ. Consequently, due to such correlations, we have $\mathbb{E}[R_\ell|R_k] \neq \mathbb{E}[R_\ell]$.

In our problem setting, we consider that the player has partial knowledge about the joint distribution of correlated arms in the form of *pseudo-rewards*, as defined below.

Definition 1 (Pseudo-Reward): Suppose we sample arm k and observe reward r. Then the pseudo-reward of arm ℓ with respect to arm k, denoted by $s_{\ell,k}(r)$, is an upper bound on the conditional expected reward of arm ℓ, i.e.,

$$\mathbb{E}[R_\ell | R_k = r] \leq s_{\ell,k}(r). \tag{2}$$

For convenience, we set $s_{\ell,\ell}(r) = r$.

Remark 1: Note that the pseudo-rewards are upper bounds on the expected conditional reward and not hard bounds on the conditional reward itself. This makes our problem setup practical as upper bounds on expected conditional reward are easier to obtain, as illustrated below.

The pseudo-reward information consists of a set of $K \times K$ functions $s_{\ell,k}(r)$ over $[0, b]$. This information can be obtained in practice through either domain and expert knowledge or

TABLE I
THE TOP ROW SHOWS THE PSEUDO-REWARDS OF ARMS 1 AND 2, I.E., UPPER BOUNDS ON THE CONDITIONAL EXPECTED REWARDS (WHICH ARE KNOWN TO THE PLAYER). THE BOTTOM ROW DEPICTS TWO POSSIBLE JOINT PROBABILITY DISTRIBUTION (UNKNOWN TO THE PLAYER). UNDER DISTRIBUTION (a), ARM 1 IS OPTIMAL WHEREAS ARM 2 IS OPTIMAL UNDER DISTRIBUTION (b)

r	$s_{2,1}(r)$
0	0.7
1	0.4

r	$s_{1,2}(r)$
0	0.8
1	0.5

	$R_1=0$	$R_1=1$
$R_2=0$	0.2	0.4
$R_2=1$	0.2	0.2

(a)

	$R_1=0$	$R_1=1$
$R_2=0$	0.2	0.3
$R_2=1$	0.4	0.1

(b)

TABLE II
IF SOME PSEUDO-REWARD ENTRIES ARE UNKNOWN (DUE TO LACK OF DOMAIN KNOWLEDGE), THOSE ENTRIES CAN BE REPLACED WITH THE MAXIMUM POSSIBLE REWARD AND THEN USED IN THE C-LUCB ALGORITHM. WE DO THAT HERE BY ENTERING 2 FOR THE ENTRIES WHERE PSEUDO-REWARDS ARE UNKNOWN

Observation from Arm 1

r	$s_{2,1}(r)$	$s_{3,1}(r)$
0	0.7	2
1	0.8	1.2
2	2	1

Observation from Arm 2

r	$s_{1,2}(r)$	$s_{3,2}(r)$
0	0.5	1.5
1	1.3	2
2	2	0.8

Observation from Arm 3

r	$s_{1,3}(r)$	$s_{2,3}(r)$
0	1.5	2
1	2	1.3
2	0.7	0.75

from controlled surveys. For instance, in the context of medical testing, where the goal is to identify the best drug to treat an ailment from among a set of K possible options, the effectiveness of two drugs is correlated when the drugs share some common ingredients. Through domain knowledge of doctors, it is possible to answer questions such as "what are the chances that drug B would be effective given drug A was not effective?", through which we can infer the pseudo-rewards.

Computing Pseudo-Rewards from domain knowledge or historical data: The pseudo-rewards can also be obtained from domain knowledge or through *offline* pilot surveys in which users are presented with *all* K arms allowing us to sample $R_1, \ldots, R_K$ jointly. Through such data, we can evaluate an estimate on the conditional expected rewards. For example in Table I, we can look at all users who obtained 0 reward for Arm 1 and calculate their average reward for Arm 2, say $\hat{\mu}_{2,1}(0)$. Since we only need an upper bound on $\mathbb{E}[R_2|R_1=0]$, we can use any one of the following approaches to set the pseudo-reward $s_{2,1}(0)$.

1) The pseudo-reward $s_{2,1}(0)$ can be set to $\hat{\mu}_{2,1}(0) + \hat{\sigma}_{2,1}(0)$, where $\hat{\mu}_{2,1}(0)$ is the empirical average of conditional rewards of R_2 given $R_1=0$ and $\hat{\sigma}_{2,1}(0)$ is the empirical standard deviation. Adding the standard deviation ensures that the pseudo-reward is an upper bound on the conditional expected reward $\mathbb{E}[R_2|R_1=0]$ with high probability.
2) Alternately, pseudo-rewards for any unknown conditional mean reward could be set to b, the maximum possible reward for the arm (recall that $R_k \in [0, b]$). Table II shows an example where unknown pseudo-rewards are set to 2, the maximum possible reward.
3) If through the training data, we obtain a soft upper bound u on $\mathbb{E}[R_2|R_1=0]$ that holds with probability $1-\delta$, then we can translate it to the pseudo-reward $s_{2,1}(0) = u \times (1-\delta) + 2 \times \delta$, (assuming maximum possible reward is 2).

Remark 2 (Reduction to Classical Multi-Armed Bandits): When all pseudo-reward entries are unknown, then all pseudo-reward entries can be filled with maximum possible reward for each arm, that is, $s_{\ell,k}(r) = b \; \forall r, \ell, k$. In that case, the problem framework studied in this paper reduces to the setting of the classical Multi-Armed Bandit problem.

While the pseudo-rewards are known in our setup, the underlying joint probability distribution of rewards is unknown. For instance, Table I(a) and Table I(b) show two joint probability distributions of the rewards that are both possible given the pseudo-rewards at the top of Table I. If the joint distribution is as given in Table I(a), then Arm 1 is optimal, while Arm 2 is optimal if the joint distribution is as given in Table I(b).

B. Application for Correlated Multi-Armed Bandits

Consider a scenario where a company needs to run a display advertising campaign in a community for one of their products, and their design team has proposed several different designs. The traction (i.e., the number of clicks, time spent on the ad) that the company generates is likely to be dependent on the design that is used for publicity. In order to find the best design, the company can run a best-arm identification algorithm by viewing the problem as a multi-armed bandit problem. Here, at each round t, a new user of that community enters the system and they show one of the K designs (i.e., arms) to this user. The reward is received through the response of the user to the ad. A straightforward solution would be to treat this problem as a classical multi-armed bandit problem and use a well known best-arm identification algorithm such as lil'UCB, LUCB or successive elimination to identify the best design for the community. But, in practice, the rewards corresponding to different designs are likely to be correlated to one another. Consider the example shown in Figure 1, over there if a user reacts positively to the first design, the user is also likely to react positively to the second ad as both ads are related to tennis. Such correlations, when accounted for in the form of pseudo-rewards, can help us identify the best-arm in much fewer samples relative to algorithms such as lil'UCB, LUCB and Successive elimination that do not account for correlations in choices.

These correlations could be known from a controlled survey or a previous advertisement campaign performed in a different demographic. For instance, from these surveys one can interpret information such as "users who like ad 1 representing tennis tend to like ad 2 that also represents tennis but not ad K which represents soccer". If a company wants to identify the best ad in a new demographic, it can use this learned correlation information to identify the best-ad in a quick manner. Note that the population composition in the two demographics

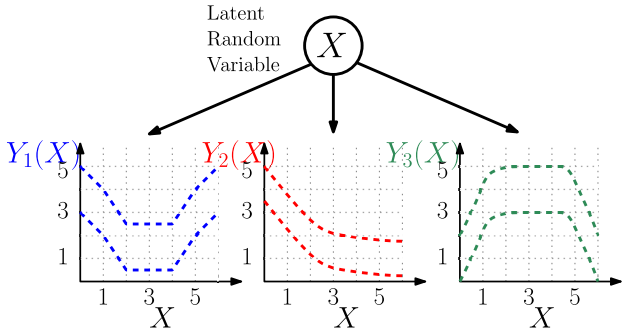

Fig. 4. A special case of our proposed problem framework is a setting in which rewards for different arms are correlated through a hidden random variable X. At each round X takes a realization in $\mathcal{X}$. The reward obtained from an arm k is $Y_k(X)$. The figure illustrates lower bounds and upper bounds on $Y_k(X)$ (through dotted lines). For instance, when X takes the realization 1, reward of arm 1 is a random variable bounded between 2 and 4.

may be very different, i.e., the fraction of users liking tennis may be very different, but it is likely that the correlation in choices remain consistent across the two demographics. One can also consider the example of identifying best policy to publicize for a political campaign, where users preferences towards different policies (i.e., climate change, gun control, abortion laws) are often correlated in all demographics, but the marginal distribution of people advocating for a single policy is very different in different communities. In such scenarios, transferring correlation information from one demographic to another by modeling them through pseudo-reward in our correlated bandit framework can help reduce the number of samples needed to identify the best-arm.

These pseudo-rewards can also be known from domain knowledge. Consider the problem of identifying the best drug for the treatment of an unknown disease. The effectiveness of different drugs is likely to be correlated as they often contain similar components. In such a situation, the domain expertise of doctors can tell us "what are the chances that drug y will be effective given drug x was effective?". One can use a conservative upper bound on the answer to this question to model pseudo-rewards. Alternatively, such correlation information could also be obtained on how different people react to different drugs in a community. As the effectiveness of drugs depends on underlying medical conditions of the patients, their response would be correlated. This correlation knowledge can then be transferred to identify the best treatment in a different community, where the distribution of underlying medical conditions may be very different.

C. Special Case: Correlated Bandits With a Latent Random Source

The studied correlated multi-armed bandit can generalize several other interesting and unexplored multi-armed bandit problems. For example, one special case is the correlated multi-armed bandit model where rewards are correlated through a latent random source [22] (See Figure 4). In this problem setup, the hidden random variable X takes an i.i.d. realization $X_t \in \mathcal{X}$ at round t and upon pulling arm k at round t, reward $Y_k(X_t)$ is observed. For the application setting of ad-recommendation, the random variable X can represent the *features* (i.e., age/occupation/income etc.) of the user. At each round a new user with feature X_t enters the system, and the goal is to identify the single best ad recommendation for the whole population in as few samples as possible. The feature X_t remains hidden to the player due to privacy concerns. Additionally, the reward $Y_k(X_t)$ represents the preference of the k^{th} ad for the user with feature X_t.

In this problem setup, the correlation information is known to the player in the form of upper and lower bounds on $Y_k(X)$, namely $\bar{g}_k(X)$ and $\underline{g}_k(X)$. These upper and lower bounds can be probabilistic, e.g., they may hold with probability 0.8 (80% confidence). For instance, the information on prior information represents the knowledge that *children of age 5-10 rate documentaries only in the range 1-3 out of 5 in* 80% *cases*. While such prior knowledge may be known from domain expertise or previous ad-campaigns performed in a different demographic, the age distribution of the community may be unknown. Due to which, the best-arm remains unknown and it needs to be found in an online manner.

This particular correlated bandit setting can be reduced to our general framework by translating the mappings $Y_k(X)$ to pseudo-rewards $s_{\ell,k}(r)$. Recall the pseudo-rewards represent an upper bound on the conditional expectation of the rewards. In this framework, if $\underline{g}_k(x)$ and $\bar{g}_k(x)$ are soft lower and upper bounds, i.e., $\underline{g}_k(x) \leq Y_k(x) \leq \bar{g}_k(x)$ w.p. $1-\kappa$, we can construct pseudo-reward as follows:

$$s_{\ell,k}(r) = (1-\kappa)^2 \times \left(\max_{\{x:\underline{g}_k(x) \leq r \leq \bar{g}_k(x)\}} \bar{g}_\ell(x) \right) + \left(1 - (1-\kappa)^2\right) \times M, \tag{3}$$

where M is the maximum possible reward an arm can provide. We evaluate this pseudo-reward by first finding the range of values within which x lies based on the reward with probability $1-\kappa$. The maximum possible reward of arm ℓ for values of x is then identified with probability $1-\kappa$. Due to this, with probability $(1-\kappa)^2$, conditional reward of arm ℓ is at-most $\max_{\{x:\underline{g}_k(x) \leq r \leq \bar{g}_k(x)\}} \bar{g}_\ell(x)$. As the maximum possible reward is M otherwise, we get the pseudo-reward as shown in (3). Once these pseudo-rewards are constructed, the problem fits in the general framework described in this paper and we can use the algorithms proposed for this setting directly.

The presented model resembles the structured bandit model studied in [23] in which mean rewards of different arms, $\mu_k(\theta)$, are known as a function of a hidden parameter θ, but the parameter θ is unknown. It is important to see that this presented model differs from [23] in two key ways – i) In [23], instead of a hidden random variable X, there is a hidden feature θ which is fixed and unknown and ii) the mean reward mappings as a function of θ are known, whereas in our model we consider the knowledge of soft upper and lower bounds on $Y_k(X)$. The model studied in [23] is more suitable for settings where the goal is to provide personalized recommendation to a user whose features θ are hidden, whereas the latent random source model (and the general correlated bandit model) is appropriate for application settings where the goal is to identify a single recommendation for the global demographic.

TABLE III
ALL BEST-ARM IDENTIFICATION ALGORITHMS HAVE THREE KEY COMPONENTS, I) SAMPLING STRATEGY AT EACH ROUND t, II) ELIMINATION CRITERIA FOR AN ARM AND III) THE STOPPING CRITERIA OF THE ALGORITHM. WE COMPARE THESE FOR RACING, LIL'UCB, LUCB AND LUCB++ ALGORITHMS AND SEE THE DIFFERENCES IN THEIR OPERATION. THE INDICES USED FOR OUR PROPOSED C-LUCB AND C-LUCB++ ARE DEFINED IN (8) AND (10)

Algorithm	Sampling Strategy	Eliminate Arm k if	Stopping Criteria
Racing	Round Robin in $\mathcal{A}_t$	$U_k\left(\frac{\delta}{K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$
lil'UCB	Sample k_t, $k_t = \arg\max_k U_k(\delta)$	N/A	$n_{k_t} \geq \alpha \sum_{k \neq k_t} n_k$
LUCB	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{A}_t} \hat{\mu}_k(t)$, $m_2 = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} U_k\left(\frac{\delta}{K}\right)$	$U_k\left(\frac{\delta}{K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$* or $L_{m_1}\left(\frac{\delta}{K}\right) > U_{m_2}\left(\frac{\delta}{K}\right)$
LUCB++	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{K}} \hat{\mu}_k(t)$ $m_2 = \arg\max_{k \in \mathcal{K} \setminus \{m_1\}} U_k\left(\frac{\delta}{2}\right)$	N/A	$L_{m_1}\left(\frac{\delta}{2K}\right) > U_{m_2}\left(\frac{\delta}{2}\right)$
C-LUCB (ours)	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{A}_t} I_k(t)$, $m_2 = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \min\left(\tilde{U}_{k,k}\left(\frac{\delta}{2K}\right), I_k(t)\right)$	$\tilde{U}_k\left(\frac{\delta}{2K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{2K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$
C-LUCB++ (ours)	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{A}_t} I_k(t)$, $m_2 = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \min\left(\tilde{U}_{k,k}\left(\frac{\delta}{2}\right), I_k(t)\right)$	$\tilde{U}_k\left(\frac{\delta}{3K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{3K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$ or $L_{m_1}\left(\frac{\delta}{4K}\right) > \tilde{U}_{m_2,m_2}\left(\frac{\delta}{4}\right)$

Note that the model presented in this subsection requires the understanding of hidden random variable X. While in certain problem settings it may be possible to obtain a latent random source representation in the form of X. In general, these hidden features may be more complicated and one may not be able to represent them. It is important to note that our proposed model in the most general setting works without having to construct a hidden feature representation through which arms are correlated. This is a key advantage of our general model over the latent random source model and the model presented in [23], which requires modeling the problem through a hidden parameter θ. Instead, our general model utilizes the available prior information directly and our algorithms adapt to the information to identify the best-arm in fewer samples relative to classical best-arm identification algorithms.

III. RELATED PRIOR WORK

The design of best-arm identification algorithms in the fixed-confidence setting have three key design components: i) their sampling strategy, i.e., which arm to pick at round t; ii) their elimination criteria, i.e., when to declare an arm as sub-optimal and remove it from the rest of the sampling procedure; and iii) their stopping criteria, i.e., when to stop the algorithm and declare an arm as the best arm.

In order to accomplish the task of best-arm identification, algorithms use the empirical mean $\hat{\mu}_k(t)$ for arm k at round t. In addition to this, upper confidence bound and lower confidence bound on the mean of arm k are maintained based on the number of samples of arm k, $n_k(t)$, and the input confidence parameter δ. In particular, the upper confidence index $U_k(n_k, \delta) = \hat{\mu}_k(t) + B(n_k, \delta)$ and lower confidence index $L_k(n_k, \delta) = \hat{\mu}_k(t) - B(n_k, \delta)$ are maintained for each arm $k \in \mathcal{K}$. Here $B(n_k, \delta) \propto \sqrt{\frac{\log\left(\frac{\log(n_k)}{\delta}\right)}{n_k}}$ is an *anytime* confidence bound [9], [24] constructed such that

$$\Pr(\exists\ n_k \geq 1\text{: } \mu_k \notin [L_k(n_k, \delta),\ U_k(n_k, \delta)]) \leq \delta. \tag{4}$$

Note that the anytime confidence interval bound the probability of the mean lying outside the confidence interval uniformly for all $n_k \geq 1$, i.e., the probability that the mean lies outside the confidence interval $[L_k(n_k, \delta),\ U_k(n_k, \delta)]$ at *any* round t is upper bounded by δ. In contrast to the Hoeffding bound, which are only valid for a fixed and deterministic n_k, the anytime confidence bound holds true uniformly for all $t \geq 1$ and for random n_k as well. We refer the reader to [24] for a detailed discussion and developments in anytime confidence bounds $B(n_k, \delta)$.

A. Existing Best-Arm Identification Strategies

There are three well-known approaches to the best-arm identification problem: i) Successive Elimination (also called racing) [14], [15], [25]; ii) lil'UCB (Law of Iterated Logarithms Upper Confidence Bound) [9]; and iii) LUCB [10], [13] (Lower and Upper Confidence Bound). Below, we briefly introduce these algorithms, and present a summary of their arm sampling strategies and elimination and stopping criteria in Table III.[1] For more details, we refer the reader to [7] that

[1]The confidence bound $C(n_k(t), \delta)$, and subsequently lower and upper confidence indices $L_k(n_k(t), \delta)$ and $U(n_k(t), \delta)$, depend on the number of rounds t, the number of samples of arm k till round t $n_k(t)$ and the confidence parameter δ. For brevity purposes, at times we represent the confidence bound as $C(n_k, \delta)$ or $C(\delta)$ and the LCB, UCB indices as $L_k(t, \delta), L_k(n_k, \delta)$ or $L_k(\delta)$ and $U_k(t, \delta), U_k(n_k, \delta)$ or $U_k(\delta)$ respectively.

provides a comprehensive survey of best-arm identification in the fixed confidence setting.

Successive Elimination or Racing: The successive elimination (also called racing) strategy maintains a set of active arms $\mathcal{A}_t$ at each round. It samples arms in a round-robin fashion from the set of active arms and at the end of each round, it eliminates an arm k from the set of active arms if the lower confidence index of some other arm $\ell \neq k$, $L_\ell(n_\ell, \frac{\delta}{K})$, is strictly larger than the upper confidence index of arm k, $U_k(n_k, \frac{\delta}{K})$. It continues this until a single arm is left in the set $\mathcal{A}_t$ and returns that arm as the optimal arm. Two other algorithms, Exponential-gap elimination [26] and PRISM [27], build upon successive elimination to provide stronger theoretical guarantees. However, their empirical performance is not promising as noted in [7].

lil'UCB [9]: The lil'UCB algorithm samples the arm with the largest upper confidence index $U_k(n_k, \delta)$ at round t and stops when an arm has been sampled more than $\frac{\alpha t}{\alpha+1}$ times till round t. In practice, the value of α is taken to be 9. It then declares the most sampled arm as the best-arm.

LUCB [7], [13]: The LUCB approach samples two arms $m_1(t), m_2(t)$ at each round t. Here, $m_1(t)$ is the arm with the largest empirical reward till round t, and $m_2(t)$ is the arm with the largest UCB index $U_k(n_k, \frac{\delta}{K})$ among the rest. The LUCB algorithm stops if the lower confidence bound of the first arm $m_1(t)$ is larger than the upper confidence index of all other arms.[2] Subsequently, another algorithm LUCB++ [11], [12] was designed that operates in a similar manner to LUCB but constructs the upper confidence and lower confidence indices with different confidence parameters for $m_1(t), m_2(t)$. The details of the upper confidence and lower confidence indices for each of these algorithms are presented in Table III. Note that our metric for comparison is the total number of samples collectively drawn from the arms. As LUCB algorithms sample two arms at each round, the total number of samples drawn from the LUCB algorithms is two times the number of rounds t. By comparing the total number of samples and not the number of rounds t, we draw a fair comparison between the performance of LUCB and lil'UCB algorithm.

All the approaches described above work well for the case where rewards are known to be either sub-Gaussian or bounded. Furthermore, if the class of distribution is known (e.g., it is known that rewards are Gaussian with known σ and unknown μ), then there are two more approaches known in the literature, namely Top Two Thompson Sampling (TTTS) [28] and Tracking [29]. In TTTS, the player computes a posterior distribution on the mean reward of each arm and then applies Thompson sampling on the posterior to obtain two samples. It stops when the posterior probability of an arm k being optimal exceeds a certain threshold $\tau_k(n_k, \delta)$. The TTTS algorithm can be computationally intensive as it involves the computation of posterior probability in each round of their algorithm. In [29], authors evaluate a lower bound for the Multi-Armed bandit problem in the form of an optimization problem. They propose a tracking based approach, that solves the optimization problem at each round to obtain an estimated rate at which each arm should be sampled at round t and sample arms in proportion to that rate. More recently, [30] proposed alternative approaches to the track-and-stop algorithm that do not require solving an optimization problem at each round. Instead, they view the optimization problem as an unknown game and have sampling rules based on iterative saddle point strategies. All of the approaches listed above require knowing the *class* of reward distribution. Since we only assume that the rewards are bounded and not the class of distribution, we do not focus on extending TTTS or Tracking based approaches to the correlated bandit setting in this paper.

B. Developments in Confidence Sequence $B(n_k, \delta)$

It is important to note that the performance of the algorithms described above depends critically on the tightness of the confidence bound $B(n_k, \delta)$. For instance, initially the LUCB algorithm was proposed with the confidence interval $B(n_k, \delta) = \sqrt{\frac{\log\left(\frac{405 n_k^{1.1}}{\delta} \log\left(\frac{405 n_k^{1.1}}{\delta}\right)\right)}{2 n_k}}$ (See [13]) for $[0, 1]$ bounded random variables. Subsequently tighter bounds as in [7], [10] were developed, which led to performance improvements in the LUCB algorithm. See Table IV for a comparison different confidence bound developed over time and how they affect the empirical performance of the best-arm identification algorithms.[3] For a more detailed comparison of different confidence bounds $B_k(n_k, \delta)$, we refer the reader to [24, Table 2]. To the best of our knowledge, the tightest $1-\delta$ anytime confidence interval for bounded and sub-Gaussian random variables is proposed in [24], which constructs

$$B(n_k, \delta) = 0.85\sqrt{\frac{\log(\log(0.5 n_k)) + 0.72 \log(5.2/\delta)}{n_k}}. \quad (5)$$

Due to this observation, which is also supported by empirical evidence in Table IV, we use the bound suggested by [24] in all implementations of Successive Elimination, LUCB and our proposed algorithm. However, our algorithm and analysis extend to arbitrary $1-\delta$ anytime confidence interval $B(n_k, \delta)$.

We would also like to highlight the fact that lil'UCB is known to have the best known theoretical sample complexity (in terms of its dependency on the number of arms K). The LUCB algorithm stops with probability $1-\delta$ after obtaining at most $\sum_{k\in\mathcal{K}} \frac{2\zeta}{\Delta_k^2}\left(\log\left(\frac{K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right)$ samples, where $\Delta_k = \mu_{k^*} - \mu_k$, the difference in mean reward of optimal arm k^* and mean reward of arm k. And $\Delta_{k^*} = \min_{k\neq k^*} \Delta_k$, the gap between best and second best arm. It is known that lil'UCB algorithm has a sample complexity $\mathrm{O}\left(\sum_{k\in\mathcal{K}} \frac{1}{\Delta_k^2} \log\left(\frac{\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right)$, i.e., it avoids the $\log(K)$ term in the numerator, and hence has the best

[2]Equivalently, one can eliminate an arm k from $\mathcal{A}_t$ at the end of each round if the upper confidence index of arm k is smaller than the lower confidence index of some other arm, and stop the algorithm when the set of active arms $|\mathcal{A}_t| = 1$. This implementation of the LUCB algorithm has the same guarantees as the one proposed in [7], [13] while obtaining similar empirical performance.

[3]The bound proposed in [10], [11] are KL based bounds that evaluate the indices $U_k(n_k, \delta), L_k(n_k, \delta)$ as $\inf\{j > \hat{\mu}_k : n_k(t) d_{kl}(\hat{\mu}_k, j) < d(B)\}$ and $\sup\{j < \hat{\mu}_k : n_k(t) d_{kl}(\hat{\mu}_k, j) < d(B)$. The distance $d_{kl}(x, y)$ is evaluated as $x \log(x/y) + (1-x)\log((1-x)/(1-y))$.

TABLE IV
DESCRIPTION OF THE WELL-KNOWN BEST-ARM IDENTIFICATION ALGORITHMS AND THE CONFIDENCE BOUND $B(n_k, \delta)$ THAT THEY USE FOR [0,1] BOUNDED REWARDS. ALL THE THREE TYPES OF ALGORITHMS HAVE EVOLVED WITH TIME DUE TO THE DEVELOPMENT OF TIGHTER $1-\delta$ ANYTIME CONFIDENCE INTERVALS $B(n_k, \delta)$. WE SEE THAT THE ALGORITHMS PERFORM BEST WITH THE CONFIDENCE BOUND SUGGESTED IN [24], AND HENCE WE USE THAT FOR ALL OUR IMPLEMENTATIONS OF RACING, LUCB, LUCB++ AND OUR PROPOSED ALGORITHM IN THE REST OF THE PAPER. THE REPORTED SAMPLE COMPLEXITY IS FOR THE TASK OF IDENTIFYING BEST MOVIE GENRE FROM THE SET OF 18 MOVIE GENRES IN THE MOVIELENS DATASET. EXPERIMENTAL SETUP IS DESCRIBED IN DETAIL IN SECTION VII

Algorithm	**Confidence Bound** $B(n_k, \delta)$	**Type**	**Samples Drawn**
Succ Elimination [15]	$\sqrt{\frac{\log\left(\frac{\pi^2 n_k^2}{3\delta}\right)}{2n_k}}$	Racing	577209.4
lil Succ Elimination [7]	$0.85\sqrt{\frac{\log(\log(0.2585n_k))+0.96\log(67.59/\delta)}{n_k}}$	Racing	120498.5
KL-Racing [10]	$d(B) = 2\log\left(\frac{11.1t^{1.1}}{\delta}\right)*$	Racing	147780.4
Racing with [24]	$0.85\sqrt{\frac{\log(\log(0.5n_k))+0.72\log(5.2/\delta)}{n_k}}$	Racing	82504.7
LUCB with [10]	$\sqrt{\frac{\log\left(\frac{405t^{1.1}}{\delta}\log\left(\frac{405t^{1.1}}{\delta}\right)\right)}{2n_k}}$	LUCB	219510.2
lil LUCB [27]	$0.85\sqrt{\frac{\log(\log(0.2585n_k))+0.96\log(67.59/\delta)}{n_k}}$	LUCB	90523.0
KL-LUCB [10]	$d(B) = 2\log\left(\frac{405.5t^{1.1}}{\delta}\right) + \log\log\left(\frac{405.5t^{1.1}}{\delta}\right)$	LUCB	81154.4
LUCB with [24]	$0.85\sqrt{\frac{\log(\log(0.5n_k))+0.72\log(5.2/\delta)}{n_k}}$	LUCB	62533.2
lil'UCB [9]	$0.85\sqrt{\frac{\log(\log(0.2585n_k))+0.96\log(67.59/\delta)}{n_k}}$	lil'UCB	140987.0
lil-KL-LUCB [11]	$d(B) = 1.86\log\left(\kappa\log_2\left(\frac{2n_k}{\delta}\right)\right)$	LUCB++	92000.0
LUCB++ with [24]	$0.85\sqrt{\frac{\log(\log(0.5n_k))+0.72\log(5.2/\delta)}{n_k}}$	LUCB++	55138.8

known theoretical sample complexity. However, it has been observed (both in [7] and our experiments) that its empirical performance is inferior to that of the LUCB algorithm. Due to this reason, we focus on proposing an algorithm C-LUCB that extends the LUCB approach to the correlated bandit setting. We have included the performance of lil'UCB in all our experiments.

C. Algorithms Outside the Classical Setting

Unlike the regret-minimization problem, the best-arm identification problem is relatively unexplored outside of the classical multi-armed bandit setting. A rare exception is the *structured* bandit setting, where mean rewards corresponding to different arms are related to one another through a hidden parameter θ. The underlying value of θ is fixed and unknown, but the mean reward mappings $\theta \rightarrow \mu_k(\theta)$ are known. The *linear* bandit setting is a special case of structured bandits, where mean reward mappings are of the form $x_k^\mathsf{T}\theta$ with x_k known to the player. The best-arm identification problem has been studied in [31], [32] for *linear* bandits and in [21] for the general structured bandit setting. Other special cases of structured bandits include global bandits [33], regional bandits [34] and the generalized linear bandits [35]; to the best of our knowledge the best arm identification problem has not been addressed in these special cases. Note that in the full generality, the structured bandit framework is simply a bandit problem with constraints on the joint probability distribution [36], but that setting has only been studied for the objective of regret minimization and not best-arm identification. To the best of our knowledge, the structured bandits work studying best-arm identification [21], [31], [32] assume the presence of a hidden parameter θ through which mean rewards of different arms are related to one another. Our correlated bandit framework focuses on structured bandit settings by modeling the correlations explicitly through the knowledge of pseudo-rewards.

Recently, best-arm identification was studied under the spectral bandit framework [37], which assumes that the arms are the nodes of known a weighted graph, with $w_{a,b}$ denoting the weight between arms a and arms b. The spectral bandit framework poses a restriction on the relationship between mean rewards of individual arms by assuming that $\sum_{a,b\in\mathcal{K}} w_{a,b}\frac{(\mu_a-\mu_b)^2}{2} \leq R$, where R is known to the player.

The correlated bandit model considered in this paper is fundamentally different from the structured bandit framework as detailed below.

1) The model studied here explicitly models the correlations in the rewards of different arms *at any given round* t. In structured bandits, the mean rewards are related to each other, but the reward realizations at a given round are not necessarily correlated. Similar to structured bandits, the work on spectral bandits [37] considers a setup with constrains between mean rewards of different arms, but does not capture the correlations explicitly in their framework.
2) It is also possible to use the structured bandit framework for the objective of identify best global recommendation in an ad-campaign. However, there are two major challenges i) In deciding upon the hidden parameter θ that we need to use, through which the mean rewards are related to one another. ii) Secondly, in the structured bandits framework, the reward mappings from θ to $\mu_k(\theta)$ need to be *exact*. If they happen to be incorrect, then the algorithms for structured bandit cannot be used as they rely on the correctness of $\mu_k(\theta)$ to construct confidence intervals on the unknown

parameter θ. In contrast, the model studied here only relies on the pseudo-rewards being upper bounds on the conditional expectations $\mathbb{E}[R_\ell|R_k = r]$. Our proposed algorithm works even when these bounds are not tight. The lack of hidden parameter θ and pseudo-rewards being upper bounds on conditional expectations make the model studied in this paper more suitable for practical scenarios where the goal is to identify the best global recommendation.

IV. Proposed Correlated-LUCB Best-Arm Identification Algorithm

In the correlated MAB framework, the rewards observed from one arm can help estimate the rewards from other arms. Our key idea is to use this information to reduce the number of samples taken before stopping. We do so by maintaining the *empirical pseudo-rewards* of all pairs of distinct arms at each round t.

A. Empirical Pseudo-Rewards and New UCB indices

In our correlated MAB framework, pseudo-reward of arm ℓ with respect to arm k provides us an estimate on the reward of arm ℓ through the reward sample obtained from arm k. We now define the notion of empirical pseudo-reward which can be used to obtain an *optimistic estimate* of μ_ℓ through just reward samples of arm k.

Definition 2 (Empirical and Expected Pseudo-Reward): After t rounds, arm k is sampled $n_k(t)$ times. Using these $n_k(t)$ reward realizations, we can construct the empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ for each arm ℓ with respect to arm k as follows.

$$\hat{\phi}_{\ell,k}(t) \triangleq \frac{\sum_{\tau=1}^{t} \mathbb{1}_{k_\tau=k}\ s_{\ell,k}\big(r_{k_\tau}\big)}{n_k(t)}, \qquad \ell \in \{1,\ldots,K\} \setminus \{k\}. \tag{6}$$

The expected pseudo-reward of arm ℓ with respect to arm k is defined as

$$\phi_{\ell,k} \triangleq \mathbb{E}\big[s_{\ell,k}(R_k)\big]. \tag{7}$$

For convenience, we set $\hat{\phi}_{k,k}(t) = \hat{\mu}_k(t)$ and $\phi_{k,k} = \mu_k$. Note that the empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ is defined with respect to arm k and it is only a function of the rewards observed by sampling arm k.

Observe that $\mathbb{E}\big[s_{\ell,k}(R_k)\big] \geq \mathbb{E}[\mathbb{E}[R_\ell|R_k = r]] = \mu_\ell$. Due to this, empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ can serve as an estimated upper bound on μ_ℓ. Using the definitions of empirical pseudo-reward, we now define auxiliary UCB indices, namely crossUCB and pseudoUCB indices, which are used in the selection and elimination strategy of the C-LUCB algorithm.

Definition 3 (CrossUCB Index $\tilde{U}_{\ell,k}(t,\delta)$): At the end of round t, we have $n_k(t)$ samples of arm k. Using these, we define the CrossUCB Index of arm ℓ with respect to arm k as

$$\tilde{U}_{\ell,k}(t,\delta) \triangleq \hat{\phi}_{\ell,k}(t) + B(n_k,\delta). \tag{8}$$

Furthermore, we define

$$\tilde{U}_{\ell}(t,\delta) = \min_k \tilde{U}_{\ell,k}(t,\delta),$$

i.e., the tightest of the K upper bounds, $\tilde{U}_{\ell,k}(t,\delta)$, for arm ℓ.

Note that the CrossUCB index for arm ℓ with respect to arm k, $\tilde{U}_{\ell,k}(t,\delta)$ is constructed only through the samples obtained from arm k. Furthermore, we have $\tilde{U}_{k,k}(t,\delta) = \hat{\mu}_k(t) + B(n_k,\delta)$, which coincides with the standard upper confidence index used in the best-arm identification literature. We use the confidence bound suggested by [24] (see Section III) for the construction of $B(n_k,\delta)$ for $[0,b]$ bounded random variables, i.e.,

$$B(n_k,\delta) = \frac{1.7b}{2}\sqrt{\frac{\log\Big(\log\Big(\frac{b^2 n_k}{2}\Big)\Big) + 0.72\log(5.2/\delta)}{n_k}}. \tag{9}$$

As pseudo-rewards are *upper bounds* on conditional expected reward, they can only be used to construct alternative upper bounds on the mean reward of other arms and not alternative lower bounds. Due to this reason, we keep the definition of lower confidence index $L_k(t,\delta)$ the same as that in the classical multi-armed bandit setting, i.e., $L_k(t,\delta) = \hat{\mu}_k(t) - B(n_k,\delta)$. In addition to the CrossUCB and the LCB index for each arm, we now define the PseudoUCB index of arm ℓ with respect to arm k. The PseudoUCB indices prove useful for the design and analysis of our proposed algorithm.

Definition 4 (PseudoUCB Index $I_{\ell,k}(t)$): We define the PseudoUCB Index of arm ℓ with respect to arm k as follows.

$$I_{\ell,k}(t) \triangleq \hat{\phi}_{\ell,k}(t) + b\sqrt{\frac{2\log t}{n_k(t)}} \tag{10}$$

Furthermore, we define $I_\ell(t) = \min_k I_{\ell,k}(t)$, the tightest of the K upper bounds for arm ℓ.

Note that the PseudoUCB Index uses a confidence bound, $b\sqrt{\frac{2\log t}{n_k(t)}}$, which is typically used in the UCB1 algorithm [2] for the objective of cumulative reward maximization. It has the property that $\Pr(I_\ell(t) < \mu_\ell) \leq Kt^{-3}$ [See Lemma 3 in the Appendix] in the supplementary material, i.e., the probability of mean lying outside the pseudoUCB index $I_\ell(t)$ at round t decays exponentially with the number of rounds t. This property allows us to show desirable sample complexity results for our proposed algorithm in Section V. We now present the C-LUCB algorithm, that makes use of the PseudoUCB, CrossUCB and LCB indices in its strategy for sampling arms, eliminating arms and stopping the algorithm.

B. C-LUCB Algorithm

The C-LUCB algorithm maintains a set of active arms $\mathcal{A}_t$, which is initialized to the set of all arms $\mathcal{K} = \{1,\ldots,K\}$. At each round t, it samples arms, eliminates arms and then decides whether to stop as described below.

1) *Sampling Strategy:* At each round t, the C-LUCB algorithm samples two arms $m_1(t)$ and $m_2(t)$, where

$$m_1(t) = \arg\max_{k\in\mathcal{A}_t}\ I_k(t),$$

$$m_2(t) = \arg\max_{k\in\mathcal{A}_t\setminus\{m_1(t)\}}\ \min\left(\tilde{U}_{k,k}\left(t,\frac{\delta}{2K}\right), I_k(t)\right).$$

2) *Elimination Criteria:* The C-LUCB algorithm removes an arm k from the set $\mathcal{A}_t$, if the CrossUCB index of

arm k is smaller than the LCB index of some other arm in $\mathcal{A}_t$, i.e., if

$$\tilde{U}_k\left(t, \frac{\delta}{2K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(t, \frac{\delta}{2K}\right).$$

Here, $\tilde{U}_\ell(t, \frac{\delta}{2K}) = \min_k \tilde{U}_{\ell,k}(t, \frac{\delta}{2K})$.

3) *Stopping Criteria:* If $|\mathcal{A}_t| = 1$, stop the algorithm and declare the arm in $\mathcal{A}_t$ as the optimal arm with $1-\delta$ confidence.

Both LUCB and C-LUCB sample the top two arms at round t in $m_1(t)$ and $m_2(t)$ so as to resolve the ambiguity among them as fast as possible. However, C-LUCB uses the additional pseudo-reward information to modify its choice of $m_1(t)$ and $m_2(t)$. In particular, the use of $I_k(t)$ in definition of $m_2(t)$ avoids the sampling of an arm that appears sub-optimal from samples of other arms. Similarly, using the CrossUCB index $\tilde{U}_k(t, \delta/2K)$ instead of $\tilde{U}_{k,k}(t, \delta/2K)$, allows the C-LUCB to eliminate some arms earlier than the LUCB algorithm. A comparison of the operation of C-LUCB with LUCB and Racing based algorithms is presented in Table III. We show that the proposed C-LUCB algorithm is $1-\delta$ correct and analyze its sample complexity in the next section. As the key difference between C-LUCB and LUCB is in its sampling strategy, we explore some other variants of C-LUCB in Section VI, where we study the effect of performance on altering the definitions of $m_1(t)$ and $m_2(t)$.

V. Sample Complexity Results

In this section, we analyze sample complexity of the proposed C-LUCB algorithm, that is, the number of samples required to identify the best arm with probability $1-\delta$. We show that some arms, referred to as *non-competitive* arms, are explored implicitly through the samples of the optimal arm k^* and contribute only an O(1) term in the sample complexity, while other arms called *competitive* arms have an $O(\log(1/\delta))$ contribution in the sample complexity of the C-LUCB algorithm. The correlation information enables us to identify the non-competitive arms using samples from other arms and eliminate them early. For the sample complexity analysis, we assume that the rewards are bounded between $[0, 1] \forall k \in \mathcal{K}$. Note that the algorithms do not require this condition and the analysis can also be generalized to any bounded rewards.

A. Competitive and Non-Competitive Arms

We now define the notion of *competitive* and *non-competitive* arms, which are important to interpret our sample complexity results for the C-LUCB algorithm. Let k^* denote the arm with the largest mean and $k^{(2)}$ denote the arm with the second largest mean.

Definition 5 (Non-Competitive and Competitive arms): An arm ℓ is said to be non-competitive if the expected reward of the second best arm $k^{(2)}$ is strictly larger than the expected pseudo-reward of arm ℓ with respect to the optimal arm k^*, i.e., $\tilde{\Delta}_\ell \triangleq (\mu_{k^{(2)}} - \phi_{\ell,k^*}) > 0$. Similarly, an arm ℓ is said to be competitive if $\tilde{\Delta}_\ell = (\mu_{k^{(2)}} - \phi_{\ell,k^*}) \leq 0$. We refer to $\tilde{\Delta}_\ell$ as the pseudo-gap of arm ℓ in the rest of the paper. We denote the set of the competitive arms as $\mathcal{C}$ and the total number of competitive arms as C in this paper.

The best arm k^* and second best arm $k^{(2)}$ have pseudo-gaps $\tilde{\Delta}_{k^*} = (\mu_{k^{(2)}} - \phi_{k^*,k^*}) < 0$ and $\tilde{\Delta}_{k^{(2)}} = (\mu_{k^{(2)}} - \phi_{k^{(2)},k^*}) \leq 0$ respectively, and hence are counted in the set of competitive arms. As $\phi_{\ell,k^*} \geq \mu_\ell$, the pseudo-gap $\tilde{\Delta}_\ell \leq \Delta_\ell$. Due to this, we have $2 \leq C \leq K$.

The central idea behind our C-LUCB approach is that after sampling the optimal arm k^* sufficiently large number of times, the non-competitive (and thus sub-optimal) arms will not be selected as $m_1(t)$ or $m_2(t)$ by the C-LUCB algorithm, and thus will not be explored explicitly. Furthermore, the non-competitive arms can be eliminated from the information obtained through arm k^*. As a result, the non-competitive arms contribute only an O(1) term in the sample complexity, i.e., the contribution is independent of the confidence parameter δ. However, the competitive arms cannot be discerned as sub-optimal by just using the rewards observed from the optimal arm, and have to be explored $O(\log(\frac{1}{\delta}))$ times each. Thus, we are able to reduce a K-armed bandit to a C-armed bandit problem, where C is the number of competitive arms.[4]

B. Analysis of C-LUCB

We start by first proving the $(1-\delta)$-correctness of C-LUCB algorithm and then analyzing its sample complexity in terms of the number of samples obtained until the stopping criterion is satisfied.

Theorem 1 ($(1-\delta)$ correctness of C-LUCB): Upon stopping, the C-LUCB algorithm declares arm k^* as the best arm with probability $1-\delta$.

Proof Sketch: To prove Theorem 1, we define three events $\mathcal{E}_1, \mathcal{E}_2$ and $\mathcal{E}_3$ below. Let $\mathcal{E}_1$ be the event that empirical mean of all arm lie within their confidence intervals uniformly for all $t \geq 1$

$$\mathcal{E}_1 = \left\{ \forall t \geq 1, \forall k \in \mathcal{K}, \hat{\mu}_k(t) - B\left(n_k(t), \frac{\delta}{2K}\right) \leq \mu_k \leq \hat{\mu}_k + B\left(n_k(t), \frac{\delta}{2K}\right) \right\} \tag{11}$$

Define $\mathcal{E}_2$ to be the event that empirical pseudo-reward of optimal arm with respect to all other arms lie within their CrossUCB indices uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_2 = \left\{ \forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{k^*,\ell} \leq \hat{\phi}_{k^*,\ell}(t) + B\left(n_\ell(t), \frac{\delta}{2K}\right) \right\} \tag{12}$$

Similarly define $\mathcal{E}_3$ to be the event that the empirical pseudo-reward of the sub-optimal arms with respect to the optimal arm lies within their CrossUCB indices uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_3 = \left\{ \forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{\ell,k^*} \leq \hat{\phi}_{\ell,k^*}(t) + B\left(n_{k^*}(t), \frac{\delta}{2K}\right) \right\} \tag{13}$$

[4]Observe that k^* and subsequently C are both unknown to the algorithm. Before the start of the algorithm, it is not known which arm is optimal/competitive/non-competitive.

Furthermore, we define $\mathcal{E}$ to be the intersection of the three events, i.e.,

$$\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3. \tag{14}$$

Due to the nature of anytime confidence intervals (See 4) and union bound over the set of arms, we have $\Pr(\mathcal{E}_1^c) \leq \frac{\delta}{2}$, $\Pr(\mathcal{E}_2^c) \leq \frac{\delta}{4}$ and $\Pr(\mathcal{E}_3^c) \leq \frac{\delta}{4}$ giving us $\Pr(\mathcal{E}^c) \leq \delta$. Furthermore, we show that, when event $\mathcal{E}$ occurs, the C-LUCB algorithm always declares k^* as the best arm. This gives us the desired result in Theorem 1. A detailed proof is given in Appendix in the supplementary material.

Theorem 2: Given event $\mathcal{E}$ (defined in 14), the expected number of samples drawn by C-LUCB until stopping, is bounded as

$$\mathbb{E}\Big[N^{\text{C-LUCB}} \mid \mathcal{E}\Big] \leq \sum_{k\in\mathcal{C}} \frac{2\zeta}{\Delta_k^2} \log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) + \frac{3K+2Kt_0}{1-\delta} + \frac{2}{1-\delta}\left(\frac{(K+1)^3}{t_0} + \frac{2}{t_0^2}\right), \tag{15}$$

where $t_0 = \inf\{\tau \geq 2 : \Delta_{k^*} \geq 4\sqrt{\frac{2K\log\tau}{\tau}}\ \forall k \notin \mathcal{C}\}$ and ζ is a universal constant that depends on the type of confidence bound used to construct $B(n_k, \delta)$ (Section III-B) – the tighter the bound, the smaller the ζ. The gap Δ_k is defined as $\Delta_k \triangleq \mu_{k^*} - \mu_k$ for $k \neq k^*$, i.e., the difference in mean reward of optimal arm k^* and mean reward of arm k and $\Delta_{k^*} \triangleq \min_{k\neq k^*} \Delta_k$, i.e., the gap between best and second best arm.

We present a brief proof outline below, while the detailed proof is available in Appendix E in the supplementary material.

Proof Sketch: In order to bound the total number of samples drawn by C-LUCB, we bound the total number of rounds T taken by C-LUCB before stopping. As C-LUCB algorithm pulls two arms $m_1(t)$ and $m_2(t)$ in each round t, the number of samples $N^{\text{C-LUCB}} = 2T$. We obtain an upper bound on the total number of rounds T, considering the following four counts of the number of rounds and obtain an upper bound for each of them under the event $\mathcal{E}$:

1) $T^{(\mathcal{R})}$: Let $T^{(\mathcal{R})}$ denote the number of rounds in which $I_{k^*}(t) < \mu_{k^*}$, i.e., the count of events in which the pseudoUCB index of arm k^* is smaller than the mean of arm k^* at round t.
2) $T^{(C)}$: Define $T^{(C)}$ to be the number of rounds in which $m_1(t), m_2(t) \in \mathcal{C}$ and event $I_{k^*}(t) < \mu_{k^*}$ does not occur.
3) $T^{(NC)}$: Define $T^{(NC)}$ to be the number of rounds in which $m_1(t) \notin \mathcal{C}, m_2(t) \neq k^*$ or $m_2(t) \notin \mathcal{C}, m_1(t) \neq k^*$.
4) $T^{(*)}$: Define $T^{(*)}$ to be the number of rounds in which $m_1(t) = k^*, m_2(t) \notin \mathcal{C}$ or $m_2(t) = k^*, m_1(t) \notin \mathcal{C}$.

We can now see that $T \leq T^{(\mathcal{R})} + T^{(C)} + T^{(NC)} + T^{(*)}$. We show that

$$\Pr(I_{k^*}(t) < \mu_{k^*}|\mathcal{E}) = \frac{\Pr(I_{k^*} < \mu_{k^*}, \mathcal{E})}{\Pr(\mathcal{E})} \leq \frac{\Pr(I_{k^*} < \mu_{k^*}, \mathcal{E})}{1-\delta} \leq \frac{\Pr(I_{k^*} < \mu_{k^*})}{1-\delta} \leq \frac{Kt^{-3}}{1-\delta},$$

giving us $\mathbb{E}\big[T^{(\mathcal{R})}|\mathcal{E}\big] \leq \frac{1}{1-\delta}\sum_{t=1}^{\infty} Kt^{-3} \leq \frac{3K}{2(1-\delta)}$. Next we show that

$$\Pr\left(T^{(C)} + T^{(*)} \geq \sum_{k\in\mathcal{C}} \frac{\zeta}{\Delta_k^2} \log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)|\mathcal{E}\right) = 0.$$

Due to this,

$$T^{(C)} + T^{(*)} \leq \sum_{k\in\mathcal{C}} \frac{\zeta}{\Delta_k^2} \log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) \quad \text{w.p. } 1-\delta.$$

We then evaluate an upper bound on $\mathbb{E}\big[T^{(NC)}|\mathcal{E}\big]$ and show that it is upper bounded by a O(1) constant, i.e.,

$$\mathbb{E}\Big[T^{(NC)}|\mathcal{E}\Big] \leq \frac{Kt_0}{1-\delta} + \frac{1}{1-\delta}\left(\frac{(K+1)^3}{t_0} + \frac{2}{t_0^2}\right).$$

Putting these results together, we obtain the result of Theorem 2.

Furthermore, as $\mathbb{E}\big[T^{(NC)}|\mathcal{E}\big], \mathbb{E}\big[T^{(\mathcal{R})}|\mathcal{E}\big]$ is upper bounded by an O(1) constant as $\delta \to 0$, we have $\sum_{t=1}^{\infty} \Pr(\mathcal{E}_t^{\text{NC}}) < \infty$, where $\mathcal{E}_t^{\text{NC}}$ is the event that $m_1(t) \notin \mathcal{C}, m_2(t) \neq k^*$ or $m_2(t) \notin \mathcal{C}, m_1(t) \neq k^*$. By Borel-Cantelli Lemma 1, this implies that with probability 1, the event $\mathcal{E}_t^{\text{NC}}$ takes place only finitely many time steps t. As a result of this, $\exists d_1 : \Pr(T^{(NC)} > d_1|\mathcal{E}) = 0$ *almost surely*. Similarly $\exists d_2 : \Pr(T^{(\mathcal{R})} > d_2|\mathcal{E}) = 0$ *a.s.* As a consequence of this, we have the following result bounding the total number of samples drawn from the C-LUCB algorithm with probability $1-\delta$.

Corollary 1: The number of samples obtained by C-LUCB is upper bounded as

$$N^{\text{C-LUCB}} \leq \sum_{k\in\mathcal{C}} \frac{2\zeta}{\Delta_k^2} \log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) + d \quad \text{w.p. } 1-\delta, \tag{16}$$

where $d = \max(d_1, d_2)$. Note that the $\text{O}(\log(\frac{1}{\delta}))$ term is only summed for the set of competitive arms $\mathcal{C}$, in contrast to the LUCB algorithm where the sample complexity term involves summation of a $\text{O}(\log(\frac{1}{\delta}))$ for all arms $k \in \mathcal{K}$. In this sense, our proposed algorithm reduces a K-armed bandit problem to a C-armed bandit problem.

The key intuition behind our sample complexity result is that the sampling of $m_1(t) = \arg\max_{k\in\mathcal{A}_t} I_k(t)$ ensures that the optimal arm is sampled at least t/K times till round t with high-probability. This in turn ensures that the non-competitive arms are not selected as $m_1(t)$ or $m_2(t)$, due to which we see that their expected number of samples are bounded above by a O(1) constant.

C. Comparison With the LUCB Algorithm

The LUCB algorithm is known to stop after obtaining at most $\Big(\sum_{k\in\mathcal{K}} \frac{2\zeta}{\Delta_k^2} \log\big(\frac{K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\big)\Big)$ samples with probability at

TABLE V
WE STUDY TWO INTUITIVE VARIANTS OF C-LUCB WHICH DIFFER IN THEIR SAMPLING STRATEGY OF $m_1(t)$ AND $m_2(t)$. BOTH OF THEM HAVE SAME ELIMINATION AND STOPPING CRITERIA AS THE C-LUCB ALGORITHM. WE REPORT THE NUMBER OF SAMPLES NEEDED TO IDENTIFY THE BEST GENRE FROM THE SET OF 18 MOVIE GENRES IN THE MOVIELENS DATASET. WHILE ALL OF THESE ARE SMALLER THAN THE SAMPLES DRAWN BY LUCB (WHICH IS 61175.4 IN THIS CASE), THE DIFFERENCE BETWEEN THE VARIANTS OF C-LUCB IS MINIMAL. EXPERIMENTAL DETAILS ARE DESCRIBED IN DETAIL IN SECTION VII, WE SET THE VALUE OF $p = 0.2$ (I.E., THE FRACTION OF PSEUDO-REWARD ENTRIES THAT ARE REPLACED BY 5) IN THIS EXPERIMENT. SUCH SIMILARITY IN EMPIRICAL PERFORMANCE HAS ALSO BEEN OBSERVED IN OUR OTHER EXPERIMENTS AND WE FOUND NO CLEAR WINNER AMONG THE THREE WHEN COMPARED ON THEIR EMPIRICAL PERFORMANCE

Algorithm	**First arm** $m_1(t)$	**Second arm** $m_2(t)$	Samples drawn
C-LUCB	$\arg\max_{k\in\mathcal{A}_t} I_k(t)$	$\arg\max_{k\in\mathcal{A}_t\setminus\{m_1\}} \min\left(\tilde{U}_{k,k}\left(\frac{\delta}{2K}\right), I_k(t)\right)$	39277.8
maxmin-LUCB	$\arg\max_{k\in\mathcal{A}_t} \min_\ell \hat{\phi}_{k,\ell}(t)$	$\arg\max_{k\in\mathcal{A}_t\setminus\{m_1\}} \tilde{U}_k\left(\frac{\delta}{2K}\right)$	36314.2
2-LUCB	$\arg\max_{k\in\mathcal{A}_t} I_k(t)$	$\arg\max_{k\in\mathcal{A}_t\setminus\{m_1\}} \tilde{U}_k\left(\frac{\delta}{2K}\right)$	39385.8

least $1-\delta$. More formally,

$$N^{\text{LUCB}} \leq \left(\sum_{k\in\mathcal{K}} \frac{2\zeta}{\Delta_k^2} \log\left(\frac{K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right), \qquad \text{w.p. } 1-\delta.$$

We compare this result with the one that we prove for C-LUCB algorithm in Theorem 2.

Reduction to a C-Armed Bandit problem: As highlighted earlier, in the C-LUCB approach, the $O(\log(\frac{1}{\delta}))$ term only comes from the set of competitive arms, as opposed to the LUCB algorithm which has $O((\log(\frac{1}{\delta}))$ contribution from all its arms. In this sense, C-LUCB algorithm reduces a K-armed bandit problem to a C-armed bandit problem. Depending on the problem instance, the value of C can vary between 2 and K.

Slightly larger number of samples from competitive arms: We see that the contribution coming from a competitive arm in C-LUCB algorithm is $\frac{2\zeta}{\Delta_k^2}\log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)$. This is slightly larger than the contribution coming from a sub-optimal arm in LUCB algorithm, where each arm contributes $\frac{2\zeta}{\Delta_k^2}\log\left(\frac{K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)$ in the sample complexity. This is due to the fact that we construct slightly wider confidence intervals, $B(n_k, \frac{\delta}{2K})$ instead of $B(n_k, \frac{\delta}{K})$, in C-LUCB to take advantage of the correlations present in the problem. We see in Section VII that this small increase in the width of confidence intervals does not have a significant impact on the empirical performance of the algorithm.

Theorem 2's result is in conditional expectation: While the sample complexity result of the LUCB algorithm bounds the total number of samples taken with probability $1-\delta$, our sample complexity result bounds the expected samples taken by C-LUCB algorithm under the event $\mathcal{E}$ (Theorem 2). This arises as the analysis of our algorithm requires a transient component, because it tries to avoid sampling non-competitive arm at each round with high probability. We have a result in Corollary 1 that evaluates an upper bound which holds with probability $1-\delta$, but we are unable to quantify the constant d in Corollary 1 and can only characterize d in expectation as done in Theorem 2. An open problem is to evaluate the expected sample complexity of our C-LUCB algorithm for the cases where the event $\mathcal{E}$ does not occur. While such results are hard to obtain theoretically, in all our experiments we observed that the variance in the number of samples drawn by C-LUCB is not much, and is in fact similar to that of the LUCB algorithm in all the experiments performed. This indicates that even when algorithm stops with an incorrect arm, the number of samples obtained are similar to the samples obtained under the good event $\mathcal{E}$.

The $\log(K)$ *term in numerator:* Just like the sample complexity result of the LUCB algorithm [7], our sample complexity result also has a $\log(K)$ in its sample complexity result. This is avoidable in the classical MAB framework if one uses the lil'UCB algorithm, which is known to have the optimal theoretical sample complexity in the classical bandit setting as it avoids the $\log(K)$ term in its sample complexity expression. However the use of lil'UCB algorithm leads to worse empirical performance as seen in our experiments and prior work [7]. Due to this reason, we focus only on the extension of LUCB to the correlated bandit setting. The LUCB++ algorithm has a sample complexity of the form of $\left(\sum_{k\in\mathcal{K}\setminus\{k^*\}} \frac{2\zeta_1}{\Delta_k^2}\log\left(\frac{\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) + \frac{2\zeta_2}{\Delta_{k^*}^2}\log\left(\frac{K\log\left(\frac{1}{\Delta_{k^*}^2}\right)}{\delta}\right)\right)$. The LUCB++ algorithm avoids the $\log(K)$ term in the sample complexity for the sub-optimal arms and has it only for the optimal arm k^*. Due to this, it is seen that LUCB++ slightly outperforms the LUCB algorithm empirically. In our next section, we propose the C-LUCB++ algorithm, which is a heuristic extension of LUCB++ to the correlated bandit setting and show that it finds the optimal arm with probability at least $1-\delta$.

Dependency with K: In our sample complexity results, the dependence with respect to K is loose. For our theoretical results, we focus on studying the dependence of sample complexity on δ in this paper. In Section VII, we show that even when $\delta = 0.1$ (i.e., a moderate confidence regime), our proposed algorithms outperform the classical bandit algorithms (See Figure 3).

VI. VARIANTS OF C-LUCB

In our proposed C-LUCB algorithm, at each round we sample two arms $m_1(t), m_2(t)$, where $m_1(t) = \arg\max_{k\in\mathcal{A}_t} I_k(t)$ and $m_2(t) = \arg\max_{k\in\mathcal{A}_t\setminus\{m_1\}} \min(\tilde{U}_{k,k}(\delta/2K), I_k(t))$. A sampling such as this allowed us to show $1-\delta$ correctness of the algorithm (Theorem 1) and analyse its sample complexity (Theorem 2). In this section, we explore two other algorithms,

that we call maxmin-LUCB and 2-LUCB, that sample different $m_1(t)$ and $m_2(t)$ at round t, but have the same elimination and stopping criteria as that of C-LUCB. In Table V, we contrast their sampling strategy with respect to C-LUCB. While we are able to show that both maxmin-LUCB and 2-LUCB algorithm will stop with the best-arm with probability at least $1-\delta$, we are unable to provide a sample complexity result for them.

We also evaluated the empirical performance of maxmin-LUCB and 2-LUCB on a real-world recommendation dataset, and found their empirical performance to be similar to C-LUCB. We chose to use C-LUCB as our proposed algorithm as it is possible to provide theoretical guarantees as in Theorem 1 and Theorem 2. Moreover, we find its empirical performance to be superior than classical bandit algorithms in correlated bandit settings, as we illustrate through our experiments in the next section.

A. C-LUCB++: Heuristic Extension of LUCB++

The LUCB++ algorithm as illustrated in Section III, is able to improve upon LUCB, by modifying its stopping criteria and in its sampling of $m_1(t)$ and $m_2(t)$. We propose an extension, C-LUCB++, that extends the LUCB++ algorithm to the correlated bandit setting. The comparison of C-LUCB++ and LUCB++ in its sampling, elimination and stopping criteria is presented in Table III. While we are able to show that the C-LUCB++ stops with the best arm with probability at least $1-\delta$ in Appendix G in the supplementary material, analysing its sample complexity remains an open problem. We compare the performance of C-LUCB++, with C-LUCB, LUCB, Racing and lil'UCB algorithms extensively through our experiments on Movielens and Goodreads datasets in the next section.

VII. Experiments

We now evaluate the performance of our proposed C-LUCB and C-LUCB++ algorithms in a real-world setting. By comparing the performance against classical best-arm identification algorithms on the MOVIELENS and GOODREADS datasets, we show that our proposed algorithms are able to exploit correlation to identify the best-arm in fewer samples. All results reported in our paper are presented after conducting 10 independent trials and computing their average. Additionally, in all our plots we show the error bars of width 2σ, where σ is the standard deviation in the number of samples drawn by an algorithm across the 10 independent trials.

A. Experiments on the MovieLens Dataset

The MOVIELENS dataset [38] contains a total of 1M ratings for a total of 3883 Movies rated by 6040 Users. Each movie is rated on a scale of 1-5 by the users. Moreover, each movie is associated with one (and in some cases, multiple) genres. For our experiments, of the possibly several genres associated with each movie, one is picked uniformly at random. To perform our experiments, we split the data into two parts, with the first half containing ratings of the users who provided the most number of ratings. This half is used to learn the pseudo-reward entries, the other half is the test set which is used to evaluate the performance of the proposed algorithms. Doing such a split ensures that the rating distribution is different in the training and test data.

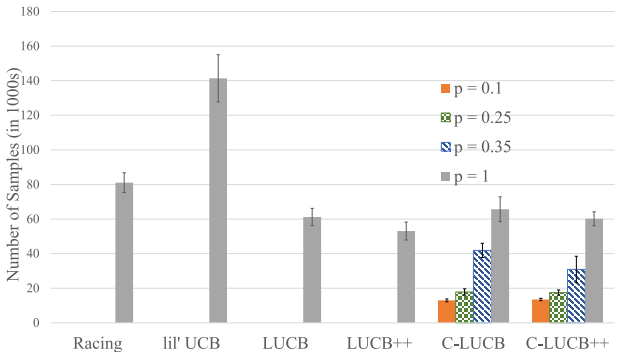

Fig. 5. Number of samples drawn by Racing, lil'UCB, LUCB, LUCB++, C-LUCB and C-LUCB++ to identify the best movie genre out of 18 possible genres in the Movielens dataset. Here, p represents the fraction of pseudo-reward entries that are replaced by the maximum possible reward (i.e., 5). When p is small, there is more correlation information available that our proposed C-LUCB and C-LUCB++ algorithms exploit to reduce the number of samples needed to identify the best movie genre. When $p = 1$, there is no correlation information available, in which case our proposed C-LUCB and C-LUCB++ algorithms have a performance similar to LUCB and LUCB++ respectively.

Best Genre identification: In this experiment, our goal is to identify the most preferred genre among the 18 different genre in the test population in fewest possible samples. The pseudo-reward entry $s_{\ell,k}(r)$ is evaluated by taking the empirical average of the ratings of genre ℓ that are rated by the users who rated genre k as r. As in practice, all such pseudo-reward entries might not be available, we randomly replace p-fraction of the pseudo-reward entries by maximum possible reward, i.e., 5. We then run our best-arm identification algorithms on the test data to identify the best-arm with 99% confidence. Figure 5 shows the average samples taken by C-LUCB and C-LUCB++ algorithm relative to the classical best-arm identification algorithms for different value of p (the fraction of pseudo-reward entries that are removed). We see that C-LUCB and C-LUCB++ algorithms significantly outperform all Racing, lil'UCB, LUCB and LUCB++ algorithms for $p = 0.1, 0.25, 0.35$ as they are able to exploit the correlations present in the problem to identify the best arm in a faster manner.

In the scenario where all pseudo-reward entries are unknown, i.e., $p = 1$, we see that the performance of C-LUCB is only slightly worse than that of LUCB algorithm. This is due to the construction of slightly wide confidence interval $B(n_k, \delta/2K)$ for the C-LUCB algorithm relative to LUCB algorithm that uses $B(n_k, \delta/K)$. We also see that in this scenario, LUCB++ and C-LUCB++ algorithm (which is an extension of LUCB++) outperform C-LUCB, which is due to the known superiority of LUCB++ over LUCB [11], [12].

Variation with δ: We then study the performance of the best-arm identification algorithms for different value of δ. In Figure 3, we plot the number of samples required by C-LUCB and C-LUCB++ to identify the best arm with 90%, 94%, 98% and 99% confidence, with $p = 0.2$ (i.e., 20% of pseudo-reward

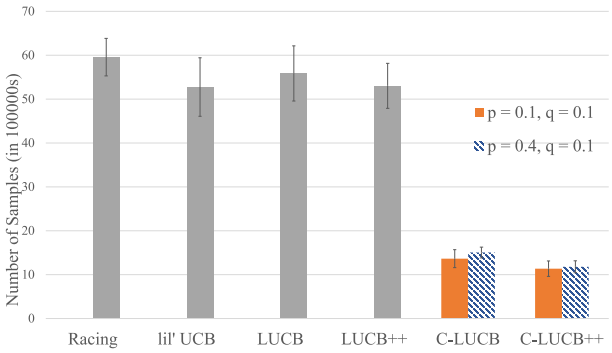

Fig. 6. Number of samples needed by Racing, lil'UCB, LUCB, LUCB++, C-LUCB and C-LUCB++ to identify the best poem out of the set of 25 poem books in the Goodreads dataset. Here p represents the fraction of pseudo-rewards that are replaced by maximum possible reward and $q = 0.1$ is added to each pseudo-reward entry to account for the fact that pseudo-reward entries may be noisy. Our proposed C-LUCB and C-LUCB++ utilize correlation information and require significantly less samples than the classical best-arm identification algorithms.

entries are replaced by 5). As C-LUCB and C-LUCB++ are able to make use of the available correlation information, we see our proposed algorithms require fewer samples than the Racing, lil'UCB, LUCB and LUCB++ algorithms in each of the four settings.

B. Experiments on the GOODREADS Dataset

The GOODREADS dataset [39] contains the ratings for 1,561,465 books by a total of 808,749 users. Each rating is on a scale of 1-5. For our experiments, we only consider the poetry section and focus on the goal of identify the most liked poem for the population. The poetry dataset has 36,182 different poems rated by 267,821 different users. We do the pre-processing of goodreads dataset in the same manner as that of the MovieLens dataset, by splitting the dataset into two halves, train and test. The train dataset contains the ratings of the users with most number of recommendations.

Best book identification: We consider the 25 most rated poetry books in the dataset and aim to identify the best book in fewest possible samples with 99% confidence. After obtaining the pseudo-reward entries from the training data, we replace p fraction of the entries with the highest possible reward (i.e., 5) as some pseudo-rewards may be unknown in practice. To account for the fact that these pseudo-reward entries may be noisy in practice, we add a safety buffer of 0.1 to each of the pseudo-reward entry $s_{\ell,k}(r)$; i.e., we set the pseudo-reward to be empirical conditional mean (obtained from training data) *plus* the safety buffer $q = 0.1$. We perform experiment on the test data and compare the number of samples obtained for different algorithms in Figure 6 for two different values of p. We see that in both the cases, our C-LUCB and C-LUCB++ algorithms outperform other algorithms as they are able to exploit the correlations in the rewards.

VIII. CONCLUDING REMARKS

In this work, we studied a new multi-armed bandit problem, where rewards corresponding to different arms are correlated to each other and this correlation is known and modeled through the knowledge of pseudo-rewards. These pseudo-rewards are *loose* upper bounds on conditional expected rewards and can be evaluated in practical scenarios through controlled surveys or from domain expertise. We then extended an LUCB based approach to perform best-arm identification in the correlated bandit setting. Our approach makes use of the pseudo-rewards to reduce the number of samples taken before stopping. In particular, our approach avoids the sampling of non-competitive arms leading to a stark reduction in sample complexity. The theoretical superiority of our proposed approach is reflected in practical scenarios. Our experimental results on Movielens and Goodreads recommendation dataset show that the presence of correlation, when exploited by our C-LUCB approach, can lead to significant reduction in the number of samples required to identify the best-arm with probability $1 - \delta$.

This work opens up several interesting future directions, including but not limited to the following.

PAC-C-LUCB: In this work, we explored the problem of identifying the best-arm with probability $1 - \delta$. A closely related problem is to find a PAC (probably approximately correct) algorithm, that identifies an arm which is within ϵ from μ_{k^*} with probability at least $1 - \delta$. We believe such an algorithm can be constructed by modifying the elimination and stopping criteria of C-LUCB algorithm. More specifically, if one compares $U_k(n_k, \delta) + \epsilon$ v/s $\max_{k \in \mathcal{A}_t} L_k(n_k, \delta)$ in the C-LUCB's elimination criteria, it may be possible to design and analyse a PAC algorithm in the correlated multi-armed bandit setting.

Using Pseudo-Lower bounds: We assume in our work that only upper bounds on conditional expected rewards, in the form of pseudo-upper-bounds, are known to the player. In practical settings, it may also be possible to obtain pseudo-lower-bounds, that may allow us to know information about lower bound on conditional expected reward. In presence of such knowledge, we believe C-LUCB algorithm will need a modification in its definition of lower confidence bound $L_k(n_k, \delta)$. By defining a crossLCB index $L_{\ell,k}(n_k, \delta)$, equivalent to crossUCB index for upper bound, we can re-define $L_k = \max L_{\ell,k}$. This new definition of the lower confidence bound index can help us to incorporate cases where pseudo-lower bounds are also known.

Top m arms identification: Throughout this work, our focus was to identify just the optimal arm from the set of K arms. Another similar problem is to come up with an approach to find the best m arms from the set of K arms. It is an interesting direction to explore in the correlated-multi armed bandit setting. We believe such a problem would be even more interesting if the pseudo-lower bounds are known. An open problem is to extend a C-LUCB like approach to identify the best m arms from the set of K arms.

Lower bound and optimal solution: While our proposed approach shows promising empirical performance and has some theoretical guarantees, it may not be the optimal solution for the correlated bandit problem studied in this paper. Studying a lower bound and correspondingly an optimal solution to this problem remains an open problem.

REFERENCES

[1] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, 2002.

[3] S. Agrawal and N. Goyal, "Further optimal regret bounds for thompson sampling," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2013, pp. 99–107.

[4] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2011, pp. 2312–2320.

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.

[6] R. Combes, S. Magureanu, and A. Proutière, "Minimal exploration in structured stochastic bandits," in *Advances in Neural Information Processing Systems (NIPS)*. Red Hook, NY, USA: Curran, 2017.

[7] K. Jamieson and R. Nowak, "Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2014 , pp. 1–6.

[8] S. Bubeck, R. Munos, and G. Stoltz, "Pure exploration in multi-armed bandits problems," in *International conference on Algorithmic learning theory*. Heidelberg, Germany: Springer, 2009, pp. 23–37.

[9] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "lil' UCB : An optimal exploration algorithm for multi-armed bandits," in *Proc. 27th Int. Conf. Learn. Theory*, 2014, pp. 423–439.

[10] E. Kaufmann and S. Kalyanakrishnan, "Information complexity in bandit subset selection," in *Proc. Int. Conf. Learn. Theory*, 2013, pp. 228–251.

[11] E. Tánczos, R. Nowak, and B. Mankoff, "A KL-LUCB algorithm for large-scale crowdsourcing," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 5894–5903.

[12] M. Simchowitz, K. Jamieson, and B. Recht, "The simulator: Understanding adaptive sampling in the moderate-confidence regime," 2017, arXiv:1702.05186.

[13] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone, "PAC subset selection in stochastic multi-armed bandits," in *Proc. 29th Int. Conf. Mech. Learn. (ICML)*, vol. 12, 2012, pp. 227–234.

[14] R. E. Bechhofer, "A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs," *Biometrics*, vol. 14, no. 3, pp. 408–429, 1958.

[15] E. Even-Dar, S. Mannor, and Y. Mansour, "PAC bounds for multi-armed bandit and Markov decision processes," in *Proc. 15th Annu. Int. Conf. Comput. Learn. Theory*, 2002, pp. 255–270.

[16] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges," *Stat. Sci. Rev. J. Inst. Math. Stat.*, vol. 30, no. 2, pp. 199–215, 2015.

[17] J. White, *Bandit Algorithms for Website Optimization*. Farnham, U.K.: O'Reilly Media, 2012.

[18] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3122009.3242042

[19] S. Gupta, S. Chaudhari, G. Joshi, and O. Yağan, "Multi-armed bandits with correlated arms," 2019, [Online]. Available: arXiv:1911.03959.

[20] S. Gupta, S. Chaudhari, S. Mukherjee, G. Joshi, and O. Yağan, "A unified approach to translate classical bandit algorithms to the structured bandit setting," 2018, arXiv:1810.08164.

[21] R. Huang, M. M. Ajallooeian, C. Szepesvári, and M. Müller, "Structured best arm identification with fixed confidence," in *Proc. Int. Conf. Algorithmic Learn. Theory (ALT)*, vol. 76, Oct. 2017, pp. 593–616. [Online]. Available: http://proceedings.mlr.press/v76/huang17a.html

[22] S. Gupta, G. Joshi, and O. Yağan, "Correlated multi-armed bandits with a latent random source," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, 2020, pp. 3572–3576.

[23] T. Lattimore and R. Munos, "Bounded regret for finite-armed structured bandits," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2014, pp. 550–558.

[24] S. R. Howard, A. Ramdas, J. McAuliffe, and J. Sekhon, "Time-uniform, nonparametric, nonasymptotic confidence sequences," 2018. [Online]. Available: https://arxiv.org/abs/1810.08240

[25] E. Paulson *et al.*, "A sequential procedure for selecting the population with the largest mean from k normal populations," *Ann. Math. Stat.*, vol. 35, no. 1, pp. 174–180, 1964.

[26] Z. Karnin, T. Koren, and O. Somekh, "Almost optimal exploration in multi-armed bandits," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1238–1246.

[27] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "On finding the largest mean among many," 2013, [Online]. Available: arXiv:1306.3917.

[28] X. Shang, R. Heide, P. Menard, E. Kaufmann, and M. Valko, "Fixed-confidence guarantees for Bayesian best-arm identification," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 1823–1832.

[29] A. Garivier and E. Kaufmann, "Optimal best arm identification with fixed confidence," in *Proc. Annu. Conf. Learn. Theory (COLT)*, vol. 49, Jun. 2016, pp. 998–1027. [Online]. Available: http://proceedings.mlr.press/v49/garivier16a.html

[30] R. Degenne, W. M. Koolen, and P. Ménard, "Non-asymptotic pure exploration by solving games," 2019, [Online]. Available: arXiv:1906.10431.

[31] M. Soare, A. Lazaric, and R. Munos, "Best-arm identification in linear bandits," in *Advances in Neural Information Processing Systems (NIPS)*. Red Hook, NY, USA: Curran, 2014, pp. 828–836. [Online]. Available: http://papers.nips.cc/paper/5460-best-arm-identification-in-linear-bandits.pdf

[32] C. Tao, S. Blanco, and Y. Zhou, "Best arm identification in linear bandits with linear dimension dependency," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, Jul. 2018, pp. 4877–4886. [Online]. Available: http://proceedings.mlr.press/v80/tao18a.html

[33] O. Atan, C. Tekin, and M. van der Schaar, "Global multi-armed bandits with Hölder continuity," in *Proc. 18th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2015, pp. 28–36.

[34] Z. Wang, R. Zhou, and C. Shen, "Regional multi-armed bandits," in *Proc. 21st Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2018, pp. 510–518.

[35] C. Shen, R. Zhou, C. Tekin, and M. van der Schaar, "Generalized global bandit and its application in cellular coverage optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 218–232, Feb. 2018.

[36] B. Van Parys and N. Golrezaei, *Optimal Learning for Structured Bandits*, SSRN, Rochester, NY, USA, 2020.

[37] T. Kocák and A. Garivier, "Best arm identification in spectral bandits," 2020, arXiv:2005.09841.

[38] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, 4, p. 19, 2015.

[39] M. Wan and J. McAuley, "Item recommendation on monotonic behavior chains," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 86–94.