Communication-Efficient Collaborative Regret Minimization in Multi-Armed Bandits

Nikolai Karpov, Qin Zhang
Department of Computer Science
Indiana University
Bloomington, IN 47405, USA
nkarpov@iu.edu, qzhangcs@indiana.edu

Abstract

In this paper, we study the collaborative learning model, which concerns the tradeoff between *parallelism* and *communication overhead* in multi-agent multi-armed bandits. For regret minimization in multi-armed bandits, we present the first set of tradeoffs between the number of rounds of communication among the agents and the regret of the collaborative learning process.

1 Introduction

One of the biggest challenges with machine learning is *scalability*. In recent years, a series of papers (Tao, Zhang, and Zhou 2019; Karpov, Zhang, and Zhou 2020; Wang et al. 2020b; Karpov and Zhang 2022b,a) studied bandit problems in the collaborative learning (CL) model, where multiple agents interact with the environment to learn simultaneously and cooperatively. One of the most expensive resources in the CL model is *communication*, which consists of the number of communication steps (round complexity) and the total bits of messages exchanged between agents (bit complexity). Communication directly contributes to the learning time due to network bandwidth constraints and latency, and it can also lead to significant energy consumption, especially for deep-sea or outer-space exploration tasks. Moreover, when messages are sent using mobile devices, communication can result in significant data usage. In this paper, we focus on the round complexity in the CL model and consider a basic problem in the bandit theory named *regret minimization in multi-armed bandits* (MAB for short). We try to investigate the round-regret tradeoffs for MAB in the CL model.

In the rest of this section, we will first introduce the CL model and the MAB problem. We then describe our results and place them within the context of the literature.

Regret Minimization in MAB. In the single-agent learning model, we have one agent and a set of arms $I = \{1, 2, \ldots, N\}$; the arm i is associated with a distribution $\mathcal{D}_i$ with support $[0, 1]$ and (unknown) mean μ_i. At each time step $t = 1, 2, \ldots, T$, the agent pulls arm π_t and receives a reward r_t from distribution $\mathcal{D}_{\pi_t}$. The expected regret of a T-time single-agent algorithm $\mathcal{A}$ on input I is defined to be

$$\mathbb{E}[\mathrm{Reg}(\mathcal{A}(I,T))] = \mathbb{E}\left[\sum_{t\in[T]}(\mu_\star - \mu_{\pi_t})\right], \qquad (1)$$

where $\mu_\star \triangleq \max_{i\in[N]}\{\mu_i\}$.[1] Without loss of generality, we assume that the best arm is unique.

The Collaborative Learning Model. The CL model was formalized in Tao, Zhang, and Zhou (2019). In this model, we have K agents and a set of N arms $I = \{1, 2, \ldots .N\}$, where arm i has mean μ_i. Again let $\mu_\star \triangleq \max_{i\in[N]}\{\mu_i\}$. The learning proceeds in rounds. Within each round, at each time step t, each agent k ($k \in [K]$) pulls arm $\pi_t^{(k)}$ based on its previous pull outcomes and messages received from other agents; the arms $\{\pi_t^{(k)}\}_{k\in[K]}$ can be the same or different for different agents. At the end of each round, the K agents communicate with each other to exchange newly observed information and determine the number of time steps for the next round. The number of time steps for the first round is fixed at the beginning. See Figure 1 for an illustration of the CL model.

It is worth mentioning that the lengths of rounds are *not* required to be determined beforehand in the CL model. Though for most CL algorithms in the literature (including the one proposed in this paper), the round lengths are indeed fixed at the beginning of the algorithms. This relaxation will only make the lower bound proof harder/stronger.

The expected regret of a T-time K-agent collaborative algorithm $\mathcal{A}_K$ for MAB on input I is defined to be

$$\mathbb{E}\left[\mathrm{Reg}(\mathcal{A}_K(I,T))\right] = \mathbb{E}\left[\sum_{t\in[T]}\sum_{k\in[K]}\left(\mu_\star - \mu_{\pi_t^{(k)}}\right)\right]. \qquad (2)$$

The Batched Learning Model. The CL model is closely related to the batched learning model, which has recently received considerable attention in bandit theory (Perchet et al. 2015; Jun et al. 2016; Agarwal et al. 2017; Jin et al. 2019; Gao et al. 2019; Esfandiari et al. 2019; Bai et al. 2019; Karpov and Zhang 2020; Jin et al. 2021).

[1] We use $[n]$ to denote $\{1, 2, \ldots, n\}$.

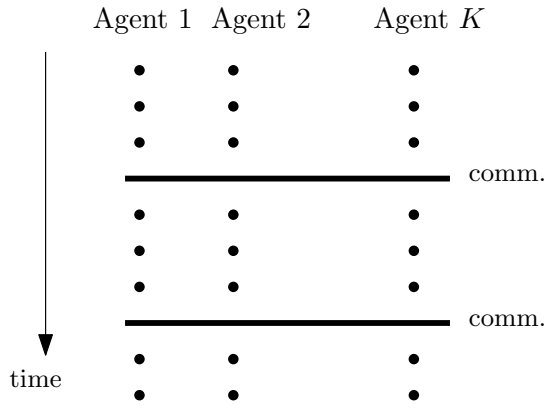

Figure 1: The collaborative learning model. Each dot represents an arm pull. The number of rounds of the learning process shown in the figure is 3.

In the batched model, there is one agent interacting with the arms. The learning proceeds in batches. The sequence of arm pulls in each batch need to be determined at the beginning of the batch. The goal is for the agent to minimize the regret over a sequence of T pulls using a small number of batches.

The batched model is motivated by applications in which there is a significant delay in getting back the observations, such as clinical trials (Thompson 1933; Robbins 1952) and crowdsourcing (Kittur, Chi, and Suh 2008).

The following observation connects the CL model and the batched model.

Observation 1. *If there is a T-time R-batch single-agent algorithm that achieves an expected regret $f(I)$ for any input I, then there is a (T/K)-time R-round K-agent collaborative algorithm that achieves an expected regret $f(I)$ for any input I.*

To see this, just note that each round of z (non-adaptive) pulls in a batched algorithm can be evenly distributed to the K agents in a collaborative algorithm so that each agent makes z/K non-adaptive pulls. Observation 1 allows us to establish a lower bound in the batched model by proving a corresponding lower bound in the CL model, and to design an algorithm for the CL model using an algorithm for the batched model.

It is important to note that Observation 1 is *one-way*; the other direction does *not* hold. This is because the CL model is **strictly stronger** than the batched model in the sense that in the CL model, each agent can make *adaptive* pulls within each round. While in the batched model, the sequence of pulls are *non-adaptive* in each batch. The requirement to accommodate *local agent adaptivity* makes the previous approaches for proving lower bounds in the batched model inapplicable to the CL model. As an example, in the previous work (Tao, Zhang, and Zhou 2019), it has been shown that for the problem of *best arm identification* (BAI) in multi-armed bandits, where the goal is to identify the arm with the highest mean rather than minimizing the regret, $O(\log K)$ rounds is sufficient to achieve almost optimal error probability under a time budget in the adaptive CL model. On the other hand, the same paper shows that $\Omega(\log N/\log\log N)$ rounds is necessary to achieve almost optimal error probability under a time budget in the batched model.[2] This shows that the local adaptivity does make the BAI problem more difficult in the CL model when $N \gg K$.

Our Results. Let $\star = \arg\max_{i\in[N]} \mu_i$ be the index of the best arm. Let $\Delta_i \triangleq \mu_\star - \mu_i$, and $\Delta(I) = \min_{i\neq\star} \Delta_i$. All logarithms have base 2 unless specified explicitly. For readability, we use '$\tilde{}$' to hide some logarithmic factors. All these factors will be spelled out in the corresponding theorems and corollaries.

The results of this paper include the followings.

1. Our main result is a lower bound for MAB in the CL model (Theorem 2). We show that for any T-time K-agent collaborative algorithm $\mathcal{A}_K$, there is an input I such that if $\mathcal{A}_K$ runs on I using at most $R \leq \frac{\log(KT)}{2\log\log\log(KT)}$ rounds, then $\mathcal{A}_K$ incurs an expected regret of $\tilde{\Omega}\left(\min\{K, (KT)^{\frac{1}{R}}\} \cdot \frac{1}{\Delta(I)}\right)$.
2. Using Observation 1, our lower bound for MAB in the CL model also gives a lower bound for MAB in the batched model (Corollary 3), which is comparable to the previous best lower bound (Gao et al. 2019).
3. We also design an algorithm for batched MAB (Theorem 17). Again via Observation 1, we obtain an algorithm for MAB in the CL model. Our upper bound matches the lower bound up to logarithmic factors in regret.

We note that there is a single-agent algorithm $\mathcal{A}^{\texttt{BPR}}$ (Bubeck, Perchet, and Rigollet 2013) such that for an input of two arms, given a time budget T, the algorithm incurs a regret of $\tilde{O}\left(\frac{1}{\Delta(I)}\right)$. Therefore, in the multi-agent setting by asking each of the K agents to run $\mathcal{A}^{\texttt{BPR}}$, we obtain a multi-agent algorithm $\mathcal{A}_K^{\texttt{BPR}}$ that incurs a regret of $\tilde{O}\left(K \cdot \frac{1}{\Delta(I)}\right)$ *without* using any communication. This is why the $\min\{K, (KT)^{\frac{1}{R}}\}$ term in our lower bound is necessary.

On the other hand, we observe that when $R \leq \frac{\log(KT)}{\log K}$, any collaborative algorithm $\mathcal{A}_K$, if runs for T time steps, incurs an expected regret of at least $\tilde{\Omega}\left(K \cdot \frac{1}{\Delta(I)}\right)$, which matches the upper bound given by $\mathcal{A}_K^{\texttt{BPR}}$ up to logarithmic factors. Therefore, Theorem 2 indicates that to achieve any super-logarithmic reduction in regret for MAB in the CL model, the agents need to use at least $\frac{\log(KT)}{\log K}$ rounds of communication.

To the best of our knowledge, our proof strategy for the round lower bound in the CL model is new. The only previous technique for proving round lower bound in the CL model is the *generalized round elimination*, which was proposed in Tao, Zhang, and Zhou (2019) for the problem

[2] In Tao, Zhang, and Zhou (2019), the $\Omega(\log N/\log\log N)$ (recall that N is the number of arms) round lower bound was proved for the *non-adaptive* CL model, which is equivalent to the batched model.

of best arm identification. However, we found it difficult to use this technique for proving a regret-round tradeoff, primarily because of the different nature of hard input instances between best arm identification and regret minimization. Specifically, Tao, Zhang, and Zhou (2019) exploited a pyramid-type construction on arms for best arm identification, while our hard inputs for regret minimization only involve two arms. If we directly apply round elimination to our hard inputs, only the best arm would survive after one elimination step. Consequently, the maximum round lower bound we can prove using round elimination is only two.

2 Related Work

Work in the Collaborative Learning Model. To the best of our knowledge, the study of the CL model began from Hillel et al. (2013), in which the authors considered the problem of best arm identification (BAI) in MAB. However, Hillel et al. (2013) only considered a special case for the lower bound, and the CL model was *not* formally defined in their paper.

The CL model that we use in this paper was introduced by Tao, Zhang, and Zhou (2019), in which almost tight round-time tradeoff was given for BAI in MAB.[3] Karpov, Zhang, and Zhou (2020) extended this line of work to the top-m arm identification in MAB. The work of Karpov and Zhang (2022b) investigated the bit complexity of BAI in the CL model. Wang et al. (2020b) studied regret minimization in MAB in the same model, but their primary focus is the bit complexity. Dai et al. (2023) investigated neural contextual bandits; their focus was only on the upper bounds.

Several recent papers (Shi and Shen 2021; Shi, Shen, and Yang 2021; Karpov and Zhang 2022a) studied problems in MAB in the *non-IID* CL model, where agents interact with possibly different environments. More specifically, Shi and Shen (2021); Shi, Shen, and Yang (2021) studied regret minimization with a focus on the bit complexity, but the bit cost in their model is integrated into the regret formulation. Karpov and Zhang (2022a) gave almost tight round-time tradeoff for BAI in the non-IID CL model. Réda, Vakili, and Kaufmann (2022) gave collaborative algorithms for non-IID BAI and regret minimization in MAB in a similar setting, but their algorithmic results only consider the fixed-confidence setting, and they did not give any lower bound on the round-regret tradeoffs.

Work in the Batched Learning Model. Batched algorithms for bandit problems have attracted significant attention in the past decade. As discussed previously, Gao et al. (2019); Esfandiari et al. (2019) studied regret minimization in MAB mentioned. An earlier work (Perchet et al. 2015) studied the same problem on two arms. Jin et al. (2021) considered asymptotic regret in MAB in the batched model. Several recent papers studied batched regret minimization in MAB using Thompson sampling (Kalkanli and Özgür 2021; Karbasi, Mirrokni, and Shadravan 2021; Karpov and Zhang 2021). Another series of works (Jun et al. 2016; Agarwal et al. 2017; Jin et al. 2019) studied batched BAI in MAB.

Other Work in Multi-Agent Bandit Learning. There are many other papers investigating multi-agent bandit learning, but they do not focus on the round complexity of the learning process. A series of papers (Szörényi et al. 2013; Landgren, Srivastava, and Leonard 2016, 2018) considered MAB problems in the peer-to-peer (P2P) computing models such that at each time step, agents can only communicate with their neighbors in the P2P network. Several papers (Liu and Zhao 2010; Rosenski, Shamir, and Szlak 2016; Bistritz and Leshem 2018; Bubeck and Budzinski 2020) considered the collision model, in which if multiple agents try to pull the same arm at a particular time step, then their rewards will be reduced due to collision.

There is a line of research that studies the bit complexity of the messages transmitted between the agents (Madhushani and Leonard 2021; Chawla et al. 2020; Wang et al. 2023, 2020a,b; Huang et al. 2021; He et al. 2022; Li et al. 2022). Some recent work has extended this line of research to related models such as the Markov Decision Processes (Dubey and Pentland 2021; Min et al. 2023).

3 The Lower Bound

In this section, we give the following theorem, which is the main result of this paper.

Theorem 2. *For any R such that $1 \leq R \leq \frac{\log(KT)}{2\log\log\log(KT)}$, and for any R-round T-time K-agent collaborative algorithm $\mathcal{A}_K$ for MAB, there is an input I such that $\mathcal{A}_K$ incurs an expected regret of $\Omega\left(\frac{1}{\log(KT)\log\log\log(KT)} \cdot \min\left\{K, (KT)^{\frac{1}{R}}\right\} \cdot \frac{1}{\Delta(I)}\right)$ on input I.*

By Observation 1, we have the following corollary (we chose a value K such that $K = (KT)^{\frac{1}{R}}$ in Theorem 2, and note that a time budget T/K in the CL model corresponds to a time budget T in the batched model).

Corollary 3. *For any R such that $1 \leq R \leq \frac{\log T}{2\log\log\log T}$, and for any R-round T-time batch algorithm $\mathcal{A}$ for MAB, there is an input I such that $\mathcal{A}$ incurs an expected regret of $\Omega\left(\frac{1}{\log T\log\log\log T} \cdot T^{\frac{1}{R}} \cdot \frac{1}{\Delta(I)}\right)$ on input I.*

This result is comparable with the lower bound result for (adaptive grid) batched algorithms in Gao et al. (2019). In particular, both results show that $\Omega(\log T/\log\log T)$ rounds is necessary to achieve the optimal regret $O\left(\log T \cdot \frac{1}{\Delta(I)}\right)$.

In the rest of this section we prove Theorem 2.

3.1 The Setup

We start by introducing some concepts and notations.

Notations. We list in Table 1 a set of key notations that we will use throughout this paper. Readers can always come back to this table when encounter an unfamiliar notation.

We will use R to denote the number of rounds used by the K-agent collaborative algorithm $\mathcal{A}_K$.

[3] In Tao, Zhang, and Zhou (2019), the time cost was presented as *speedup*, defined as the ratio between the running time of the collaborative algorithm and that of the best centralized algorithm.

Table 1: Summary of Notations

Notation	Definition
N	number of arms
K	number of agents
R	number of rounds
T	time horizon
ϵ, λ, β	fixed constants: $\epsilon = 0.1$, $\lambda = 10^{-6}$, and $\beta = 4$
α	$\alpha \triangleq \log L/(2\lambda)$
L	$L = \frac{\log(4KT)}{4}$ is the number of pairs of hard inputs
Δ_ℓ	$\Delta_\ell = 2/\beta^\ell$ is the mean gap of two arms in the level ℓ hard inputs (Section 3.2)
γ	pull transcript; a sequence of (arm pull index, reward) pairs
$r(\gamma)$	see Definition 9; intuitively, it is the index of a "big" round under transcript γ
$\ell(\gamma)$	see Definition 9; it is roughly the logarithm of the time step of the beginning of the $r(\gamma)$-th round
$\tau(\gamma, \ell)$	see Definition 11; can be seen as a mapping from $\ell(\gamma)$ back to the round index
ℓ^*	defined in Inequality (9)

For a time horizon T, we will create $L = \frac{\log(4KT)}{4}$ pairs of hard inputs, and focus on R in the range

$$\frac{4L}{\log K} \le R \le \frac{2L}{\log\log L}. \tag{3}$$

Note that in the case when $R < \frac{4L}{\log K}$, the regret will certainly be *lower* bounded by the case when $R = \frac{4L}{\log K}$, in which case $(KT)^{\frac{1}{R}}$ becomes

$$(KT)^{\frac{\log K}{4L}} = 2^{\log(KT)\cdot\frac{\log K}{\log(4KT)}} = \Theta(K).$$

This is why there is a $\min\{K, (KT)^{\frac{1}{R}}\}$ term inside the regret in Theorem 2. As mentioned in "our results" in the introduction, this $\min$ operation is also necessary to be there.

We will use the following constants in the proof: $\epsilon = 10^{-1}$, $\lambda = 10^{-6}$, and $\beta = 4$. We will use the notations instead of the actual constants in most places of this section for the sake of readability.

Pull Transcript. Let $\gamma = ((j_1, o_1), \ldots, (j_n, o_n))$ be a sequence of pulls and reward outcomes on an input I for MAB, where j_t is the index of the arm in I being pulled at the t-th time step and o_t is the corresponding reward. We call γ the *transcript* of a sequence of arm pulls, and use $|\gamma| = n$ to denote the length of γ (i.e., the number of (j_t, o_t) pairs in γ). For convenience, we use $j(\gamma) = (j_1, \ldots, j_n)$ to denote the sequence of arm indices and $o(\gamma)$ to denote the corresponding sequence of rewards.

For a sequence of arm indices $j(\gamma)$, let $\Theta_I(j(\gamma))$ be the sequence of (random) rewards by pulling the arms of I according to $j(\gamma)$. We define

$$g_I(\gamma) \triangleq \Pr[\Theta_I(j(\gamma)) = o(\gamma)], \tag{4}$$

which is the probability of observing the reward sequence $o(\gamma)$ by pulling the arms in input I following the index sequence $j(\gamma)$.

For a single-agent algorithm $\mathcal{A}$ for MAB, an input I and a time horizon n, we use $\Gamma \sim \mathcal{A}(I, n)$ to denote a random transcript generated by running $\mathcal{A}$ on input I for n time steps. For a K-agent collaborative algorithm $\mathcal{A}_K$, we write $\Gamma \sim \mathcal{A}_K(I, n)$ as the *round-robin* concatenation of the K transcripts generated by the K agents on input I for n time steps. That is,

$$\Gamma = \left\{(J_1^{(1)}, O_1^{(1)}), \ldots, (J_1^{(K)}, O_1^{(K)}), \ldots, (J_n^{(K)}, O_n^{(K)})\right\},$$

where $(J_t^{(k)}, O_t^{(k)})$ is the (arm index, reward) pair of the pull of agent k at time t. We use capital letters $J_t^{(k)}$ and $O_t^{(k)}$ since they are random variables depending on the previous pulls and outcomes.

3.2 The Hard Inputs

We begin by introducing the set of hard inputs.

For each $\ell \in \{1, \ldots, L\}$ and each $\sigma \in \{+1, -1\}$, let I_ℓ^σ be an input on two Bernoulli arms (i.e., the reward is either 0 or 1 on each pull), where the first arm has mean $\frac{1}{2} + \frac{\sigma}{\beta^\ell}$ and the second arm has mean $\frac{1}{2} - \frac{\sigma}{\beta^\ell}$.

For the convenience of writing, we will abbreviate I_ℓ^{+1} and I_ℓ^{-1} to I_ℓ^+ and I_ℓ^- respectively.

For $\ell \in [L]$, let $\Delta_\ell = 2/\beta^\ell$ be the mean gap between the two arms in the inputs I_ℓ^+ (or I_ℓ^-).

We define the set of hard inputs to be

$$\mathcal{I} = \{I_1^+, I_1^-, \ldots, I_L^+, I_L^-\}.$$

Let $\mathcal{I}_\ell = \{I_\ell^+, I_\ell^-, \ldots, I_L^+, I_L^-\}$ denote a suffix of $\mathcal{I}$ starting from index ℓ.

The set of hard inputs $\mathcal{I}$ have some nice properties which we will use in the lower bound proof. Due to the space constraints, we leave them to the full version of this paper.

3.3 Indistinguishable Input Pairs

We introduce the following event defined on a transcript γ.

Definition 4. Event $\mathcal{E}(\gamma)$: For any $\ell \in [L]$ such that $\frac{\lambda\beta^{2\ell}}{\log L} \ge |\gamma|$, and for any pair of inputs $A, B \in \mathcal{I}_\ell$, we have

$$\ln\frac{g_A(\gamma)}{g_B(\gamma)} \le 2\epsilon\,.$$

Intuitively, it says that when the length of transcript γ is smaller than $\frac{\lambda\beta^{2\ell}}{\log L}$, the probabilities of producing γ under all inputs in $\mathcal{I}_\ell$ are similar. We will often abbreviate $\mathcal{E}(\gamma)$ to $\mathcal{E}$ when γ is clear from the context.

The following lemma states that for a random transcript Γ generated by running a single-agent algorithm on any input in $\mathcal{I}$, $\mathcal{E}(\Gamma)$ holds with high probability. The technical proof can be found in the full version of this paper.

Lemma 5. *For any single-agent algorithm $\mathcal{A}$ for MAB, any $I \in \mathcal{I}$, and any $n > 0$, let $\Gamma \sim \mathcal{A}(I, n)$ denote a random*

transcript Γ generated by running $\mathcal{A}$ on input I for n time steps. It holds that

$$\Pr_{\Gamma\sim\mathcal{A}(I,n)}[\mathcal{E}(\Gamma)] \geq 1 - 1/L^6.$$

The next lemma shows that *short* transcripts generated by a single-agent algorithm on two inputs in $\mathcal{I}_\ell$ are statistically indistinguishable. Its proof can be found in the full version of this paper.

Lemma 6. *Let $\mathcal{A}$ be any single-agent algorithm for MAB. For a transcript γ, let $\mathcal{G}(\gamma)$ be any event determined by γ. For any $\ell \in [L]$, any pair of inputs $A, B \in \mathcal{I}_\ell$, and any $n \leq \frac{\lambda\beta^{2\ell}}{\log L}$, we have*

$$\Pr_{\Gamma\sim\mathcal{A}(A,n)}[\mathcal{G}(\Gamma)\wedge\mathcal{E}(\Gamma)] \leq e^{2\epsilon}\Pr_{\Gamma\sim\mathcal{A}(B,n)}[\mathcal{G}(\Gamma)\wedge\mathcal{E}(\Gamma)]\,.$$

3.4 The Lower Bound Proof

Now we are ready to give the proof of Theorem 2.

Intuition. The high level intuition is that if the number of rounds of the CL algorithm is small, then for some pair of inputs $(I^+_{\ell^*}, I^-_{\ell^*})$ in the set of hard inputs $\mathcal{I}$, we have (1) the algorithm will make many pulls in the ℓ^*-th round, and (2) the information collected from the pull transcript and previous communication *at each local agent* is not enough to distinguish $I^+_{\ell^*}$ from $I^-_{\ell^*}$. The second item implies that all sequences of pulls on the pair of inputs $(I^+_{\ell^*}, I^-_{\ell^*})$ are approximately equally likely, which, together with the first item, implies that the regret of the algorithm should be large on at least one of these two inputs.

Identifying A Critical Pair of Inputs. We start by identifying the pair $(I^+_{\ell^*}, I^-_{\ell^*})$.

Observe that by our choices of L and β, it holds that

$$T = 1/(K\Delta_L^2) = \beta^{2L}/(4K). \tag{5}$$

Let $\mathcal{A}_K$ be a R-round collaborative algorithm. Let γ be any transcript produced by $\mathcal{A}_K$. Let $t_r \triangleq t_r(\gamma)$ $(r = 1,\ldots,R)$ be the time step at the end of the r-th round. We thus have $t_R = T$. For convenience, we define $t_0 = 1/K$. Note that $t_1,\ldots,t_{R-1}$ are determined by γ, and t_0 and t_R are two fixed values.

We have the following simple fact on the ratio of finishing times of two consecutive rounds.

Fact 7. *For any $T > 0$, $R > 0$, and any transcript γ, there is a $r \in [R]$ such that $\frac{t_r}{t_{r-1}} \geq (KT)^{\frac{1}{R}}$.*

We define the following event $\mathcal{F}_r(\gamma)$ for $r = 1,\ldots,R$.

Definition 8. Event $\mathcal{F}_r(\gamma)$: For any $i < r$, it holds that $t_i/t_{i-1} < (KT)^{\frac{1}{R}}$; and for $i = r$, we have $t_r/t_{r-1} \geq (KT)^{\frac{1}{R}}$.

It is clear that $\mathcal{F}_1(\gamma),\ldots,\mathcal{F}_R(\gamma)$ are disjunctive and they together partition the probability space.

For convenience of writing, let $\alpha \triangleq \frac{\log L}{2\lambda}$. We first introduce two notations $r(\gamma)$ and $\ell(\gamma)$; intuitively, the former is the index of a "big" round under transcript γ, and the latter is roughly the logarithm of the time step of the beginning of the $r(\gamma)$-th round.

Definition 9 ($r(\gamma)$ and $\ell(\gamma)$). For a transcript γ, let $r = r(\gamma)$ be the round index such that $\mathcal{F}_r(\gamma)$ holds. And let $\ell(\gamma)$ be the integer such that

$$\frac{\beta^{2(\ell(\gamma)-1)}}{\alpha K} \leq t_{r(\gamma)-1} < \frac{\beta^{2\ell(\gamma)}}{\alpha K}. \tag{6}$$

The next claim shows that the value of $\ell(\gamma)$ will not be larger than L. Its proof can be found in the full version of this paper.

Claim 10. *For any γ, it holds that $1 \leq \ell(\gamma) \leq L$.*

By the definition of $\mathcal{F}_r(\gamma)$, we have

$$t_{r(\gamma)} \geq (KT)^{\frac{1}{R}}\cdot t_{r(\gamma)-1} = \left(\frac{\beta^{2L}}{4}\right)^{\frac{1}{R}} t_{r(\gamma)-1}\,. \tag{7}$$

Let $m_{r(\gamma)} = t_{r(\gamma)} - t_{r(\gamma)-1}$ be the length of the $r(\gamma)$-th round. By (6) and (7), we have

$$m_{r(\gamma)} \geq \left(\left(\frac{\beta^{2L}}{4}\right)^{\frac{1}{R}} - 1\right)\frac{\beta^{2(\ell(\gamma)-1)}}{\alpha K}. \tag{8}$$

Now, consider a particular ℓ^* such that

$$\Pr_{\Gamma\sim\mathcal{A}_K(I^+_L,T)}[\ell(\Gamma) = \ell^*] \geq \frac{1}{L}\,. \tag{9}$$

Such an ℓ^* must exist, since each transcript $\Gamma = \gamma$ corresponds to a unique $r(\gamma)$ and consequently a unique $\ell(\gamma)$. And by Claim 10, $\ell(\gamma) \leq L$ always holds.

Our goal is show that the expected regret of $\mathcal{A}_K$ is high on either the input $I^+_{\ell^*}$ or the input $I^-_{\ell^*}$. We call $(I^+_{\ell^*}, I^-_{\ell^*})$ the *critical input pair* for $\mathcal{A}_K$.

Projection of A Collaborative Algorithm. To facilitate the regret analysis on the critical pair of inputs, we would like to introduce a concept termed as *projection of a collaborative algorithm on a single agent*.

We first introduce a notation $\tau(\gamma,\ell)$, which can be seen as a mapping from (the logarithm of) time step $\ell(\gamma)$ *back* to the round index.

Definition 11 ($\tau(\gamma,\ell)$). Let γ be an arbitrary transcript generated by running $\mathcal{A}_K$ on I for T time steps. Let $\tau(\gamma,\ell)$ be the round index such that

$$\frac{\beta^{2(\ell-1)}}{\alpha K} \leq t_{\tau(\gamma,\ell)-1} < \frac{\beta^{2\ell}}{\alpha K}. \tag{10}$$

By the Definition 9 and Definition 11, it is not difficult to check that $r(\gamma) = \tau(\gamma,\ell(\gamma))$.

Let $\mathcal{A}_K$ be a collaborative algorithm. For any $k \in [K]$, we use $\mathrm{Proj}_k^{\mathcal{A}_K}(I,\ell)$ to denote a *single-agent* algorithm that simulates $\mathcal{A}_K$.

And let

$$\zeta_\ell = \frac{\beta^{2\ell}}{\alpha}\cdot\frac{\beta^{2(\frac{L}{R}-1)}}{8K}. \tag{11}$$

$\mathrm{Proj}_k^{\mathcal{A}_K}$ simulates $\mathcal{A}_K$ as follows: In the first $(\tau(\gamma,\ell)-1)$ rounds, at each time step t, if agents $1,\ldots,K$ pull arms $a_t^{(1)},\ldots,a_t^{(K)}$ in I respectively under $\mathcal{A}_K$, then $\mathrm{Proj}_k^{\mathcal{A}_K}$ pulls arms $a_t^{(1)},\ldots,a_t^{(K)}$ in I in order. In the $\tau(\gamma,\ell)$-th

round, at each time step t when $t \le \zeta_\ell + t_{\tau(\gamma,\ell)-1}$, if agent k pulls arm $a_t^{(k)}$ in I under $\mathcal{A}_K$, then $\text{Proj}_k^{\mathcal{A}_K}$ also pulls arm $a_t^{(k)}$ in I.

For a transcript γ generated by running $\mathcal{A}_K$ and each $k \in [K]$, we introduce two concepts:

Definition 12 ($\text{Proj}_k(\gamma,\ell)$)**.** Let $\text{Proj}_k(\gamma,\ell)$ be the sequence of (j_t, o_t) pairs in γ generated by the K agents in the round-robin fashion in the first $(\tau(\gamma,\ell)-1)$ rounds, followed by the first ζ_ℓ of (j_t, o_t) pairs in the $\tau(\gamma,\ell)$-th round (or until the end of the $\tau(\gamma,\ell)$-th round) in γ generated by agent k.

$\text{Proj}_k(\gamma,\ell)$ connects a pull transcript produced by a K-agent algorithm *in the eye of the k-th agent* at the time of the ζ_ℓ-th time step in the $\tau(\gamma,\ell)$-th round (or until the end of the $\tau(\gamma,\ell)$-th round) with that produced by a single-agent algorithm.

Definition 13 ($\text{Last}_k(\gamma,\ell)$)**.** Let $\text{Last}_k(\gamma,\ell)$ be the sequence of the first ζ_ℓ of (j_t, o_t) pairs in the $\tau(\gamma,\ell)$-th round (or until the end of the $\tau(\gamma,\ell)$-th round) in γ generated by agent k.

$\text{Last}_k(\gamma,\ell)$ can be seen as a suffix of $\text{Proj}_k(\gamma,\ell)$ that is only observed locally at the k-agent in the $\tau(\gamma,\ell)$-th round.

Large Regret on the Critical Input Pair. Now we are ready to lower bound the regret. Let $\mathcal{A}_K$ be any K-agent collaborative algorithm, and ℓ^* satisfying Inequality (9). We define the following event for a transcript γ.

Definition 14. Event $\mathcal{Q}(\gamma)$: $\ell(\gamma) = \ell^*$.

Inequality (9) implies that

$$\Pr_{\Gamma\sim\mathcal{A}_K(I_L^+,T)}[\mathcal{Q}(\Gamma)] \ge \frac{1}{L}\,. \tag{12}$$

Intuitively, Event $\mathcal{Q}(\gamma)$ says that the "big" round under transcript γ coincides to at least a $1/L$ fraction of transcripts produced by $\mathcal{A}_K$ on a particular input I_L^+.

Let $\Gamma_L \sim \mathcal{A}_K(I_L^+, T)$. Let

$$\Upsilon = \{\gamma \in \texttt{supp}(\Gamma_L) \mid \mathcal{Q}(\gamma)\},$$

and for any $k \in [K]$,

$$\Upsilon_k(\ell^*) = \{\text{Proj}_k(\gamma,\ell^*) \mid \gamma \in \Upsilon\}. \tag{13}$$

We will try to show that $I_{\ell^*}^+, I_{\ell^*}^-$ are indistinguishable w.r.t. transcripts in $\Upsilon_k(\ell^*)$, and we will use the special input I_L^+ as a bridge. Specifically, we show for each transcript $\gamma \in \Upsilon_k(\ell^*)$, the probability of producing γ when the input instance is $I_{\ell^*}^+$ is close to the probability of producing γ when the input instance is $I_{\ell^*}^-$.

Before doing this, we first upper bound the length of transcripts in $\Upsilon_k(\ell^*)$. By (11) and (8), we have

$$\zeta_{\ell^*} \le \left(\left(\frac{\beta^{2L}}{4}\right)^{\frac{1}{R}} - 1\right)\frac{\beta^{2(\ell^*-1)}}{\alpha K} \le m_{\tau(\gamma,\ell^*)}. \tag{14}$$

Consequently, for any $\gamma \in \Upsilon_k(\ell^*)$,

$$\begin{aligned} |\gamma| &= K\cdot t_{\tau(\gamma,\ell^*)-1} + \zeta_{\ell^*} \\ &\le K\cdot\frac{\beta^{2\ell^*}}{\alpha K} + \frac{\beta^{2\ell^*}}{\alpha}\cdot\frac{\beta^{2(\frac{L}{R}-1)}}{8K} \\ &\le \frac{\lambda\beta^{2\ell^*}}{\log L}\,, \end{aligned} \tag{15}$$

where the last inequality holds because $\beta^{2(\frac{L}{R}-1)} \le 8K$ by the first inequality in (3).

The following two claims exhibit properties of transcripts in $\Upsilon_k(\ell^*)$. The first claim states that the probability of a random transcript being in $\Upsilon_k(\ell^*)$ is significant. Its proof makes use of Lemma 5 and Lemma 6.

Claim 15. *For any $I \in \{I_{\ell^*}^+, I_{\ell^*}^-\}$ and any $k \in [K]$, we have*

$$\Pr_{\Gamma\sim\mathcal{A}_K(I,T)}[\textit{Proj}_k(\Gamma,\ell^*) \in \Upsilon_k(\ell^*) \wedge \mathcal{E}(\textit{Proj}_k(\Gamma,\ell^*))] \ge \frac{e^{-2\epsilon}}{2L}\,.$$

The next claim states that it is difficult to use a transcript in $\Upsilon_k(\ell^*)$ to differentiate inputs $I_{\ell^*}^+$ (or $I_{\ell^*}^-$) from I_L^+. Its proof makes use of Lemma 6.

Claim 16. *For any $I \in \{I_{\ell^*}^+, I_{\ell^*}^-\}$ and any $k \in [K]$, for any $\gamma \in \Upsilon_k(\ell^*)$ such that $\mathcal{E}(\gamma)$ holds, we have*

$$\begin{aligned} &\Pr_{\Gamma\sim\mathcal{A}_K(I,T)}[\textit{Proj}_k(\Gamma,\ell^*) = \gamma] \\ =\ & c_\epsilon \Pr_{\Gamma\sim\mathcal{A}_K(I_L^+,T)}[\textit{Proj}_k(\Gamma,\ell^*) = \gamma] \end{aligned}$$

for some $c_\epsilon \in [e^{-2\epsilon}, e^{2\epsilon}]$.

Recall that Lemma 5 and Lemma 6 concern single-agent algorithms. We use the following relation between a K-agent algorithm $\mathcal{A}_K$ and a single-agent algorithm $\mathcal{A}$ to connect Claim 15 and Claim 16 with Lemma 5 and Lemma 6:

$$\Pr_{\Gamma\sim\mathcal{A}_K(I_L^+,T)}[\mathcal{Q}(\Gamma)] = \Pr_{\Gamma\sim\text{Proj}_k^{\mathcal{A}_K}(I_L^+,\ell^*)}[\Gamma \in \Upsilon_k(\ell^*)].$$

We leave the proofs of Claim 15 and Claim 16 to the full version of this paper.

We now try to prove Theorem 2.

Let $\Gamma^+ \sim \mathcal{A}_k(I_{\ell^*}^+, T)$, and $\Gamma^- \sim \mathcal{A}_k(I_{\ell^*}^-, T)$. By Claim 16 and Claim 15, we know that

$$\begin{aligned} &\sum_{\gamma\in\Upsilon_k(\ell^*)} \min\left\{\begin{array}{l}\Pr[\text{Proj}_k(\Gamma^+,\ell^*)=\gamma],\\ \Pr[\text{Proj}_k(\Gamma^-,\ell^*)=\gamma]\end{array}\right\} \\ \ge\ & \frac{e^{-2\epsilon}}{2L}\cdot(c_\epsilon)^2 \ge \frac{e^{-8\epsilon}}{2L}. \end{aligned} \tag{16}$$

For an input I and transcript $\gamma = ((j_1,o_1),\ldots,(j_{|\gamma|},o_{|\gamma|}))$, let $\text{Reg}(I,\gamma)$ denote the regret of pulling the arm sequence $j(\gamma)$ on the input I, that is, $\text{Reg}(I,\gamma) = \sum_{t=1,\ldots,|\gamma|}(\mu_* - \mu_{j_t})$.

For any transcript $\gamma \in \Upsilon_k(\ell^*)$ and any $k \in [K]$, we consider the regret $U_k = \text{Reg}\left(I_{\ell^*}^+, \text{Last}_k(\gamma,\ell^*)\right)$ and $V_k = \text{Reg}\left(I_{\ell^*}^-, \text{Last}_k(\gamma,\ell^*)\right)$. Due to our constructions of $I_{\ell^*}^+$ and $I_{\ell^*}^-$, we have for any $k \in [K]$,

$$U_k + V_k \ge \Delta_{\ell^*}\cdot\zeta_{\ell^*}\,. \tag{17}$$

Since for $k = 1,\ldots,K$, $\text{Last}_k(\gamma,\ell^*)$ are disjoint, we have

Algorithm 1: BATCHEDMAB(I, λ, T)

1: Initialize a set of active arms $I_0 \leftarrow I$
2: Set $T_0 \leftarrow 0$
3: **for** $i = 1, 2, \ldots, \log_\lambda T$ **do**
4: Set $T_i \leftarrow \lambda^i$
5: **end for**
6: Set $r \leftarrow \log_\lambda \log(T^3 N)$
7: Pull each arm for T_{r-1} times
8: **while** $r \le \log_\lambda T$ or $|I_r| > 1$ **do**
9: **for** $a \in I_r$ **do**
10: Make $(T_r - T_{r-1})$ pulls on arm a
11: Compute $\hat{\mu}_a^r$, the estimated mean after T_r pulls
12: **end for**
13: Let $\hat{\mu}_{\max}^r \leftarrow \max_{a \in I_r} \hat{\mu}_a^r$
14: Set $I_{r+1} \leftarrow \left\{ a \mid \hat{\mu}_{\max}^r - \hat{\mu}_a^r < 2\sqrt{\frac{\ln(T^3|I|)}{T_r}} \right\}$
15: Update $r \leftarrow r + 1$
16: **end while**
17: **if** $r < \log_\lambda T$ **then**
18: Assign the rest of pulls to the single arm in I_r
19: **end if**

for any $\gamma \in \Upsilon$,

$$\begin{aligned} & \mathrm{Reg}(I_{\ell^*}^+, \gamma) + \mathrm{Reg}(I_{\ell^*}^-, \gamma) \\ \ge & \sum_{k \in [K]} (U_k + V_k) \\ \overset{(17)}{\ge} & K\Delta_{\ell^*} \cdot \zeta_{\ell^*} \\ = & K\Delta_{\ell^*} \cdot \frac{\beta^{2\ell^*}}{\alpha} \cdot \frac{\beta^{2(\frac{L}{R}-1)}}{8K} \\ = & \frac{\beta^{2(\frac{L}{R}-1)}}{2\alpha} \cdot \frac{1}{\Delta_{\ell^*}}. \end{aligned} \tag{18}$$

By (16) and (18), we have that

$$\begin{aligned} & \max\left\{ \mathbb{E}\left[\mathrm{Reg}(\mathcal{A}_K(I_{\ell^*}^+, T))\right], \mathbb{E}\left[\mathrm{Reg}(\mathcal{A}_K(I_{\ell^*}^-, T))\right] \right\} \\ & \ge \frac{1}{2} \cdot \frac{e^{-8\epsilon}}{2L} \cdot \frac{\beta^{2(\frac{L}{R}-1)}}{2\alpha} \cdot \frac{1}{\Delta_{\ell^*}} \\ & = \Omega\left(\frac{\beta^{\frac{2L}{R}}}{L \log L} \cdot \frac{1}{\Delta_{\ell^*}} \right). \end{aligned}$$

This concludes the proof of Theorem 2.

4 The Algorithm

In this section, we design a batched algorithm for MAB, which implies an algorithm for MAB in the CL model via Observation 1. Our batched algorithm is described in Algorithm 1. It uses the successive elimination method. In each batch, we pull the remaining arms for an equal number of times and then eliminate those whose empirical means are smaller than the best one by a good margin.

Algorithm 1 can be seen as a variant of the algorithm in Gao et al. (2019). The main differences are: (1) Algorithm 1 employs an early stopping rule (triggered when $|I_r| = 1$ at Line 8), which leads to an instance-dependent batch complexity; and (2) it uses a preliminary exploration step (Line 7) to further reduce the number of batches.

The proof of the following theorem can be found in the full version of this paper. We note that Algorithm 1 does *not* need to know $\Delta(I)$, but in the analysis we can upper bound both the number of batches and the regret in terms of $\Delta(I)$.

Theorem 17. *For any $\lambda \ge 2$, BATCHEDMAB(I, λ, T) uses $\eta \le \log_\lambda T$ rounds and incurs an expected regret of $O\left(\sum_{a \ne \star} \frac{\lambda \log T}{\Delta_a}\right)$. We also have $\eta = O\left(\log_\lambda \frac{1}{\Delta(I)}\right)$ with probability $\left(1 - \frac{1}{T^3}\right)$.*

By Observation 1, we have the following corollary.

Corollary 18. *There is a collaborative algorithm $\mathcal{A}_K$ for MAB such that under time horizon T, for any input I, $\mathcal{A}_K$ uses $\eta \le \log_\lambda(KT)$ rounds and incurs an expected regret of $O\left(\sum_{a \ne \star} \frac{\lambda \log(KT)}{\Delta_a}\right)$. We also have $\eta = O\left(\log_\lambda \frac{1}{\Delta(I)}\right)$ with probability $\left(1 - \frac{1}{T^3}\right)$.*

We would like to give a brief comparison between our upper bound and the lower bound.

1. If we set $\lambda = (KT)^{\frac{1}{R}}$. By Corollary 18, for the case of two arms, Algorithm 1 uses at most R rounds and incurs an expected regret $O\left((KT)^{\frac{1}{R}} \cdot \log(KT) \cdot \frac{1}{\Delta(I)}\right)$. Recall by Theorem 2 that the expected regret needs to be $\Omega\left((KT)^{\frac{1}{R}} \cdot \frac{1}{\log(KT)\log\log(KT)} \cdot \frac{1}{\Delta(I)}\right)$. For $R = O(1)$, which is of practical interest, our upper and lower bounds match up to a term that is logarithmic of $(KT)^{\frac{1}{R}}$.
2. If we set $\lambda = \Theta(1)$, by Corollary 18, Algorithm 1 uses $O(\log(KT))$ rounds and achieves asymptotically optimal regret $O\left(\sum_{a \ne \star} \frac{\log(KT)}{\Delta_a}\right)$. While the best centralized algorithm has essentially the same regret $O\left(\sum_{a \ne \star} \frac{\log(T)}{\Delta_a}\right)$ (Garivier, Ménard, and Stoltz 2016); recall that time T in the centralized model corresponds to KT in the CL model.

5 Concluding Remarks

In this paper, we present the first set of round-regret tradeoffs for regret minimization in multi-armed bandits in the collaborative learning model. To the best of our knowledge, our lower bound results are the first to address the local adaptivity of agents for regret minimization in the collaborative learning model.

We observe that a poly-logarithmic factor gap remains between our upper and lower bounds, potentially to be bridged in the future work. It would also be interesting to generalize the results to non-IID environments, and investigate the round-regret tradeoffs for other bandits and reinforcement learning problems in the collaborative learning model.

Acknowledgments

Nikolai Karpov and Qin Zhang are supported in part by NSF CCF-1844234 and CCF-2006591.

References

Agarwal, A.; Agarwal, S.; Assadi, S.; and Khanna, S. 2017. Learning with Limited Rounds of Adaptivity: Coin Tossing, Multi-Armed Bandits, and Ranking from Pairwise Comparisons. In *COLT*, 39–75.

Bai, Y.; Xie, T.; Jiang, N.; and Wang, Y.-X. 2019. Provably Efficient Q-Learning with Low Switching Cost. In *NeurIPS*.

Bistritz, I.; and Leshem, A. 2018. Distributed Multi-Player Bandits - a Game of Thrones Approach. In *NeurIPS*, 7222–7232.

Bubeck, S.; and Budzinski, T. 2020. Coordination without communication: optimal regret in two players multi-armed bandits. In Abernethy, J. D.; and Agarwal, S., eds., *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, 916–939. PMLR.

Bubeck, S.; Perchet, V.; and Rigollet, P. 2013. Bounded regret in stochastic multi-armed bandits. In Shalev-Shwartz, S.; and Steinwart, I., eds., *COLt 2013*, volume 30 of *JMLR Workshop and Conference Proceedings*, 122–134.

Chawla, R.; Sankararaman, A.; Ganesh, A.; and Shakkottai, S. 2020. The Gossiping Insert-Eliminate Algorithm for Multi-Agent Bandits. In Chiappa, S.; and Calandra, R., eds., *AISTATS*, volume 108, 3471–3481.

Dai, Z.; Shu, Y.; Verma, A.; Fan, F. X.; Low, B. K. H.; and Jaillet, P. 2023. Federated Neural Bandits. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Dubey, A.; and Pentland, A. 2021. Provably Efficient Cooperative Multi-Agent Reinforcement Learning with Function Approximation. *CoRR*, abs/2103.04972.

Esfandiari, H.; Karbasi, A.; Mehrabian, A.; and Mirrokni, V. S. 2019. Batched Multi-Armed Bandits with Optimal Regret. *CoRR*, abs/1910.04959.

Gao, Z.; Han, Y.; Ren, Z.; and Zhou, Z. 2019. Batched Multi-armed Bandits Problem. In *NeurIPS*.

Garivier, A.; Ménard, P.; and Stoltz, G. 2016. Explore First, Exploit Next: The True Shape of Regret in Bandit Problems. *CoRR*, abs/1602.07182.

He, J.; Wang, T.; Min, Y.; and Gu, Q. 2022. A Simple and Provably Efficient Algorithm for Asynchronous Federated Contextual Linear Bandits. In *NeurIPS*.

Hillel, E.; Karnin, Z. S.; Koren, T.; Lempel, R.; and Somekh, O. 2013. Distributed Exploration in Multi-Armed Bandits. In *NIPS*, 854–862.

Huang, R.; Wu, W.; Yang, J.; and Shen, C. 2021. Federated Linear Contextual Bandits. In *NeurIPS*, 27057–27068.

Jin, T.; Shi, J.; Xiao, X.; and Chen, E. 2019. Efficient Pure Exploration in Adaptive Round model. In *NeurIPS*, 6605–6614.

Jin, T.; Tang, J.; Xu, P.; Huang, K.; Xiao, X.; and Gu, Q. 2021. Almost Optimal Anytime Algorithm for Batched Multi-Armed Bandits. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 5065–5073. PMLR.

Jun, K.; Jamieson, K. G.; Nowak, R. D.; and Zhu, X. 2016. Top Arm Identification in Multi-Armed Bandits with Batch Arm Pulls. In *AISTATS*, 139–148.

Kalkanli, C.; and Özgür, A. 2021. Batched Thompson Sampling. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *NeurIPS*, 29984–29994.

Karbasi, A.; Mirrokni, V. S.; and Shadravan, M. 2021. Parallelizing Thompson Sampling. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *NeurIPS*, 10535–10548.

Karpov, N.; and Zhang, Q. 2020. Batched Coarse Ranking in Multi-Armed Bandits. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NeurIPS*.

Karpov, N.; and Zhang, Q. 2021. Batched Thompson Sampling for Multi-Armed Bandits. *CoRR*, abs/2108.06812.

Karpov, N.; and Zhang, Q. 2022a. Collaborative Best Arm Identification with Limited Communication on Non-IID Data. *CoRR*, abs/2207.08015.

Karpov, N.; and Zhang, Q. 2022b. Communication-Efficient Collaborative Best Arm Identification.

Karpov, N.; Zhang, Q.; and Zhou, Y. 2020. Collaborative Top Distribution Identifications with Limited Interaction (Extended Abstract). In *FOCS*, 160–171. IEEE.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with Mechanical Turk. In Czerwinski, M.; Lund, A. M.; and Tan, D. S., eds., *CHI*, 453–456. ACM.

Landgren, P.; Srivastava, V.; and Leonard, N. E. 2016. Distributed cooperative decision-making in multiarmed bandits: Frequentist and Bayesian algorithms. In *CDC*, 167–172. IEEE.

Landgren, P.; Srivastava, V.; and Leonard, N. E. 2018. Social Imitation in Cooperative Multiarmed Bandits: Partition-Based Algorithms with Strictly Local Information. In *CDC*, 5239–5244. IEEE.

Li, C.; Wang, H.; Wang, M.; and Wang, H. 2022. Communication Efficient Distributed Learning for Kernelized Contextual Bandits. In *NeurIPS*.

Liu, K.; and Zhao, Q. 2010. Distributed Learning in Multi-Armed Bandit with Multiple Players. *IEEE Transactions on Signal Processing*, 58(11): 5667–5681.

Madhushani, U.; and Leonard, N. E. 2021. When to Call Your Neighbor? Strategic Communication in Cooperative Stochastic Bandits. *CoRR*, abs/2110.04396.

Min, Y.; He, J.; Wang, T.; and Gu, Q. 2023. Cooperative Multi-Agent Reinforcement Learning: Asynchronous Communication and Linear Function Approximation. *CoRR*, abs/2305.06446.

Perchet, V.; Rigollet, P.; Chassang, S.; and Snowberg, E. 2015. Batched Bandit Problems. In *COLT*, 1456.

Réda, C.; Vakili, S.; and Kaufmann, E. 2022. Near-Optimal Collaborative Learning in Bandits. *CoRR*, abs/2206.00121.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5): 527–535.

Rosenski, J.; Shamir, O.; and Szlak, L. 2016. Multi-Player Bandits - a Musical Chairs Approach. In *ICML*, 155–163.

Shi, C.; and Shen, C. 2021. Federated Multi-Armed Bandits. In *AAAI*, 9603–9611. AAAI Press.

Shi, C.; Shen, C.; and Yang, J. 2021. Federated Multi-armed Bandits with Personalization. In Banerjee, A.; and Fukumizu, K., eds., *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, 2917–2925. PMLR.

Szörényi, B.; Busa-Fekete, R.; Hegedűs, I.; Ormándi, R.; Jelasity, M.; and Kégl, B. 2013. Gossip-Based Distributed Stochastic Bandit Algorithms. In *ICML*, 19–27.

Tao, C.; Zhang, Q.; and Zhou, Y. 2019. Collaborative Learning with Limited Interaction: Tight Bounds for Distributed Exploration in Multi-armed Bandits. In Zuckerman, D., ed., *FOCS*, 126–146. IEEE Computer Society.

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

Wang, P.; Proutière, A.; Ariu, K.; Jedra, Y.; and Russo, A. 2020a. Optimal Algorithms for Multiplayer Multi-Armed Bandits. In *AISTATS*, volume 108, 4120–4129.

Wang, X.; Yang, L.; Chen, Y. J.; Liu, X.; Hajiesmaili, M.; Towsley, D.; and Lui, J. C. S. 2023. Achieving Near-Optimal Individual Regret & Low Communications in Multi-Agent Bandits. In *ICLR*.

Wang, Y.; Hu, J.; Chen, X.; and Wang, L. 2020b. Distributed Bandit Learning: Near-Optimal Regret with Efficient Communication. In *ICLR*. OpenReview.net.