# Communication-Efficient Collaborative Best Arm Identification

**Nikolai Karpov, Qin Zhang**

Indiana University Bloomington
nkarpov@iu.edu, qzhangcs@indiana.edu

## Abstract

We investigate top-$m$ arm identification, a basic problem in bandit theory, in a multi-agent learning model in which agents collaborate to learn an objective function. We are interested in designing collaborative learning algorithms that achieve maximum speedup (compared to single-agent learning algorithms) using minimum communication cost, as communication is frequently the bottleneck in multi-agent learning. We give both algorithmic and impossibility results, and conduct a set of experiments to demonstrate the effectiveness of our algorithms.

## Introduction

As the scale of reinforcement learning continues to grow, multi-agent learning has become a necessity in many scenarios to speed up the learning process. In this paper, we explore *collaborative learning*, a multi-agent learning model introduced in Hillel et al. (2013) and Tao, Zhang, and Zhou (2019) for studying parallel reinforcement learning under communication constraints. Communication is frequently the bottleneck in multi-agent learning owing to network latency, energy consumption, mobile data usage, and privacy considerations. For example, in the deep sea and outer space exploration, data transmission between agents (sensors, robots, etc.) is expensive since it consumes a lot of energy, which cannot be easily recharged. In the setting where communication makes use of a mobile network, data transmission directly adds to our bill. In scenarios where privacy is a concern, we also want to minimize the total amount of communication, since messages will leak information about local data which we do not want to share with other parties. In this paper, we are interested in the following question:

> *How much communication is necessary and sufficient for scalable algorithms in the collaborative learning model?*

We study a basic problem in bandit theory: *top-$m$ arm identification* in multi-armed bandits (MAB); when $m = 1$, the problem is called *best arm identification*. This problem has numerous applications, including medical tests, industrial engineering, evolutionary computation, and crowd-sourcing. We study this problem in both the IID data setting in which agents learn from the same environment and the non-IID data setting in which agents learn from different environments.



**The Collaborative Learning Model.** We begin by introducing the collaborative learning model, in which $K$ agents communicate via a *coordinator* (central server) to learn a common objective function. The learning process proceeds in rounds. During each round, each agent takes a sequence of actions (one at each time step) and receives a sequence of observations. Each action is taken based on the agent's previous actions and observations, messages received from other parties, and the randomness of the algorithm. At the end of each round, each agent sends a message to the coordinator. After collecting messages and performing some computation, the coordinator sends a message back to each agent. At the end of the last round, the coordinator outputs the answer.

The collaborative learning model is similar to the *federated learning* model, which has attracted a lot of attention recently (Kairouz et al. 2019). The main difference is that in federated learning we aim at training machine learning models (e.g., neural networks) on multiple machines with local datasets, while the collaborative learning model is tailored for multi-agent reinforcement learning.

**Top-$m$ Arm Identification in MAB.** In this problem we have $n$ alternative arms; the $i$-th arm is associated with an unknown distribution $\mathcal{D}_i$ with support $[0, 1]$. Given a time budget $T$, we want to identify the top-$m$ arms/distributions with the largest means by a sequence of $T$ arm pulls (each pull takes a unit time). Upon each pull of the $i$-th arm, the agent observes an i.i.d. sample from $\mathcal{D}_i$.

In the collaborative learning model, agents pull arms in parallel in each round. If multiple agents pull the same arm $i$ at a particular time step, then each agent gets an *independent* sample from distribution $\mathcal{D}_i$. At the end of the last round, the coordinator needs to output the top-$m$ arms.

W.l.o.g., we assume that $n \geq 2m$, since otherwise, we can try to find the bottom $(n - m) < n/2$ arms, which is symmetric to identifying the top-$(n - m)$ arms.

**Speedup in Collaborative Learning.** We use the concept of *speedup* introduced in Tao, Zhang, and Zhou (2019) and Karpov, Zhang, and Zhou (2020) to quantify the performance of multi-agent learning in the collaborative learning model. For top-$m$ arm identification, given an input instance $I$ of $n$

arms with means $\mu_1, \ldots, \mu_n$, define

$$H^{\langle m \rangle} \triangleq H^{\langle m \rangle}(I) = \sum_{i \in [n]} \frac{1}{\Delta_i^2}, \qquad \text{where}$$

$$\Delta_i = \max\left\{\mu_i - \mu_{[m+1]}, \mu_{[m]} - \mu_i\right\}, \tag{1}$$

where $\mu_{[m]}$ stands for the $m$-th highest mean of the $n$ arms. We will assume $\forall i \in [n], \Delta_i \neq 0$ (or, equivalently, $\mu_{[m]} \neq \mu_{[m+1]}$ so that the top-$m$ arms are unique), since otherwise $H^{\langle m \rangle} = \infty$.

It is known that there is a centralized algorithm for top-$m$ arm identification in MAB that succeeds with probability at least $0.99$ given a time horizon $\tilde{O}(H^{\langle m \rangle})$ (Bubeck, Wang, and Viswanathan 2013),[1]. On the other hand, any centralized algorithm that achieves a success probability of at least $0.99$ requires $\Omega(H^{\langle m \rangle})$ time (Chen, Li, and Qiao 2017a).

Following Karpov, Zhang, and Zhou (2020), letting $\mathcal{A}$ be a collaborative algorithm that solves top-$m$ arm identification for each input instance $I$ with probability at least $0.99$ using time $T_{\mathcal{A}}(I)$, we define the speedup of $\mathcal{A}$ to be

$$\beta_{\mathcal{A}} = \min_{\text{all input instances } I} \frac{H^{\langle m \rangle}(I)}{T_{\mathcal{A}}(I)}. \tag{2}$$

It has been observed that any collaborative algorithm $\mathcal{A}$ with $K$ agents can achieve a speedup of at most $\beta_{\mathcal{A}} \leq K$ (Karpov, Zhang, and Zhou 2020).

**Non-IID Data.** In the above definition for top-$m$ arm identification, when pulling the same arm, the $K$ agents sample from the same distribution. We call this setting "learning with IID data". However, in many applications, such as channel selection in cognitive radio networks and item selection in recommendation systems, agents may interact with different environments (Kairouz et al. 2019), or, "learn with non-IID data".

For top-$m$ arm identification with non-IID data, by pulling the same arm, agents may sample from different distributions. Let $\mathcal{D}_{i,k}$ be the distribution sampled by the $k$-th agent when pulling the $i$-th arm, and let $\mu_{i,k}$ be the mean of $\mathcal{D}_{i,k}$. We define the *global mean* of the $i$-th arm to be

$$\mu_i \triangleq \frac{1}{K} \sum_{k \in [K]} \mu_{i,k}. \tag{3}$$

At the end of the learning process, the coordinator needs to output the top-$m$ arms with the largest global means.

**Our Results.** In this paper, we are primarily interested in the case that a collaborative algorithm achieves almost full speedup, that is, $\beta = \tilde{\Omega}(K)$ where $K$ is the number of agents. We try to pinpoint the minimum amount of communication needed in the system for achieve such a speedup. For convenience, when talking about communication upper bounds, we assume each numerical value can be stored in one *word*. While for lower bounds, we state the communication cost in terms of *bits*. The difference between these two measurements is just a logarithmic factor.

[1]We use '$\tilde{}$' on top of $O, \Omega, \Theta$ to hide logarithmic factors.

In the IID data setting, we show that there is a collaborative algorithm for top-$m$ arm identification achieving a speedup of $\tilde{\Omega}(K)$ using $\tilde{O}(K + m)$ words of communication (Corollary 3).[2] On the other hand, $\tilde{\Omega}(K+m)$ bits of communication is necessary to achieve $\tilde{\Omega}(K)$ speedup (Corollary 12).

In the non-IID data setting, there is a collaborative algorithm for top-$m$ arm identification achieving a speedup of $\tilde{\Omega}(K)$ using $\tilde{O}(Kn)$ words of communication (Theorem 13 and Corollary 14). We also show that $\Omega(Kn)$ bits of communication is necessary only to output the top-1 arm correctly, *regardless the amount of speedup* (Theorem 15).

The above results give a strong separation between the IID data setting and the non-IID data setting in terms of communication cost.

We have conducted a set of experiments which demonstrate the effectiveness of our algorithms.

## Related Work

Best and top-$m$ arm identification in MAB have been studied extensively and thoroughly in the centralized model where there is a single agent; (almost) tight bounds have been obtained for both problems (Audibert, Bubeck, and Munos 2010; Chen, Li, and Qiao 2017b; Bubeck, Wang, and Viswanathan 2013; Chen, Li, and Qiao 2017a). The two problems have also been investigated in the collaborative learning model (Hillel et al. 2013; Tao, Zhang, and Zhou 2019; Karpov, Zhang, and Zhou 2020). However, these works focus on the *number of rounds of communication* instead of the actual communication cost (measured by the number of words exchanged in the system); they allow agents to communicate any amount of information at the end of each round. The algorithms for top-$m$ arm identification in Karpov, Zhang, and Zhou (2020) need $\tilde{\Omega}(nK)$ communication to achieve a $\tilde{\Omega}(K)$ speedup even in the IID data setting.

On minimizing the communication cost in the collaborative learning model, Wang et al. (2020b) studied *regret minimization* in MAB in the IID data setting. Recently, Shi and Shen (2021) and Shi, Shen, and Yang (2021) studied non-IID regret minimization in MAB in similar models.

Bandit problems have been studied in several other models concerning communication-efficient multi-agent learning. But those models are all different from the collaborative learning model in various aspects. Madhushani and Leonard (2021) studied regret minimization in MAB in the setting where agents communicate over a general graph. Wang et al. (2020a) also studied regret minimization in MAB over a general communication graph, and considered a collision scenario where the reward can be collected only when the arm is pulled by one agent at a particular time. The analyses of algorithms in Madhushani and Leonard (2021) and Wang et al. (2020a) treat various parameters (e.g., the number of arms $n$, the number of players $K$) as constants, and thus the bounds may not be directly comparable. Szörényi et al. (2013) studied gossip-based algorithms for regret minimization in MAB in P2P networks.

[2]All logarithmic factors hidden in '$\tilde{}$' will be spelled out in the concrete theorems in this paper.

Mitra, Hassani, and Pappas (2021) studied best arm identification in MAB in a multi-agent model where the arms are partitioned into groups; each agent can only pull arms from a designated group. One can view this model as a special case of the collaborative learning model, in which agents can pull whichever arm they would like to.

## Learning with IID Data

### The Algorithm

The algorithm for top-$m$ arm identification in the IID data setting is described in Algorithm 1. In the high level, our algorithm follows the successive elimination/halving approach (Audibert, Bubeck, and Munos 2010; Karnin, Koren, and Somekh 2013; Karpov, Zhang, and Zhou 2020). However, in order to be communication-efficient, we need to use a two-phase algorithm design.

Let us describe Algorithm 1 and the subroutines in words. At the beginning, the coordinator picks a $(10 \ln n)$-wise independent hash function $h : [n] \to [K]$ and sends it to the $K$ agents. The hash function is used to partition the $n$ arms into $K$ subsets. One choice of such a hash function is a polynomial of degree $10 \ln n$ with random coefficients (Wegman and Carter 1981); sending this function costs $O(\ln n)$ words.

The body of Algorithm 1 consists of two phases. For convenience, we introduce the following global variables which will be used in all subroutines:

- $R \triangleq \lceil \log n \rceil$: the number of rounds.
- $n_r (0 \le r \le R)$: the number of remaining arms at the beginning of the $r$-th round.
- $T_r (0 \le r \le R)$: the number of time steps allocated for the $r$-th round.
- $I_r^k$: the subset of arms held by the $k$-th agent at the beginning of the $r$-th round.
- $Q_r^k$: the set of accepted arms by the $k$-th agent in the first $r$ rounds
- $m_r \triangleq m - |Q_r|$: the number of top-$m$ arms that are not yet accepted

In the first phase (Line 7-9), each agent is in charge of one subset of the arms. In each round $r$, we call a subroutine LocalElim$(r)$ (Algorithm 2), in which each agent prunes arms in its subset locally *without* communicating the eliminated arms to the coordinator (and consequently other agents). Throughout the phase, we monitor the sizes of subsets held by the $K$ agents. We say the $K$ subsets $\{I_r^k\}_{k\in[K]}$ of arms *balanced* if

$$\forall (i,j), \quad \left|I_r^i\right| \in \left[\tfrac{1}{2}\left|I_r^j\right|, 2\left|I_r^j\right|\right].$$

Whenever the $\{I_r^k\}_{k\in[K]}$ become unbalanced, we end the first phase. The coordinator then collects the set of remaining arms from the agents.

In the second phase (Line 12-14), in each round $r$, we call another subroutine GlobalElim$(r)$ (Algorithm 3) to continue eliminating arms. In each round, $K$ agents and the coordinator work together to eliminate a constant fraction of the arms. The second phase ends when $r = R$.

We now explain LocalElim and GlobalElim in more details.

**Algorithm 1:** Collab-Top-$m$-IID$(I, K, m, T)$

**Input:** a set $I$ of $n$ arms, $K$ agents, parameter $m$, and time horizon $T$.
**Output:** top-$m$ arms with the highest means

1 Coordinator picks a $(10 \ln n)$-wise independent hash function $h : [n] \to [K]$ and sends to each of $K$ agents;
2 for each $k \in [K]$, the $k$-th agent forms the set $I_0^k \leftarrow \{i \in I \mid h(i) = k\}$;
3 let $R \leftarrow \lceil \log n \rceil$;
4 for $r = 0, 1, \ldots, R$, set $n_r \leftarrow \lfloor n/2^r \rfloor$;
5 set $T_0 \leftarrow 0$, and for $r = 1, \ldots, R$, $T_r \leftarrow \lfloor \frac{TK2^r}{4nR} \rfloor$;
6 $r \leftarrow 0$, $m_0 \leftarrow m$;
7 **while** *partition* $\{I_r^k\}_{k\in[K]}$ *is balanced* **do**
8 $\quad \left\{(I_{r+1}^k, Q_{r+1}^k)\right\}_{k\in[K]} \leftarrow$ LocalElim$(r)$;
9 $\quad r \leftarrow r + 1$;
10 for each $k \in [K]$, the $k$-th agent sends $I_r^k$, $\{\hat{\mu}_i^{(r)}\}_{i\in I_r^k}$, and $Q_r^k$ to Coordinator;
11 Coordinator sets $I_r \leftarrow \bigcup_{k=1}^K I_r^k$ and $Q_r \leftarrow \bigcup_{k=1}^K Q_r^k$;
12 **while** $r < R$ **do**
13 $\quad (I_{r+1}, Q_{r+1}) \leftarrow$ GlobalElim$(r)$;
14 $\quad r \leftarrow r + 1$;
15 **return** $Q_r$.

In LocalElim$(r)$, each agent pulls each arm in $I_r^k$ for $(T_{r+1} - T_r)$ times, and calculates the empirical mean of the arm after $T_{r+1}$ pulls have been made on it. After that, The coordinator uses the subroutine CollabSearch (Algorithm 4 in the full version of this paper) to communication-efficiently identify the arms having the $m_r$-th and $(m_r + 1)$-th *largest* empirical means in $\bigcup_{k\in[K]} I_r^k$, and sends the two values to the $K$ agents. Using these two empirical means, each agent $k$ is able to compute the empirical gap $\hat{\Delta}_i^{(r)}$ for each $i \in I_r^k$. Next, the coordinator uses CollabSearch again to find the $n_{r+1}$-th *smallest* empirical gap $\hat{\Delta}_{n_{r+1}}^{(r)}$ and sends it to all agents. The agents then identify all arms whose empirical gaps are larger than $\hat{\Delta}_{n_{r+1}}^{(r)}$, among which they accept (by adding to the set $A_r^k$) those whose empirical means are at least the $m_r$-th largest empirical mean, and discard the others. The key feature of LocalElim is that, in order to save communication, agents accept/eliminate arms locally without sending them to the coordinator.

In GlobalElim$(r)$, the coordinator first uses a subroutine BalancedPullDist (Algorithm 5 in the full version of this paper) to communication-efficiently distribute the workload evenly to the $K$ agents such that each remaining arm in $I_r$ is pulled by $(T_{r+1} - T_r)$ times. After collecting the information from the $K$ agents, the coordinator computes the empirical mean of each arm in $I_r$ after $T_{r+1}$ pulls have been made on it. Next, the coordinator computes the empirical gaps of all arms in $I_r$, identifies those whose empirical gaps are *not* among the top-$n_{r+1}$ smallest ones. For those arms, the coordinator accepts those whose empirical means

**Algorithm 2:** LOCALELIM$(r)$

For each $k \in [K]$, the $k$-th agent pulls each arm $i \in I_r^k$ for $(T_{r+1} - T_r)$ times and calculates its empirical mean $\hat{\mu}_i^{(r)}$ (after being pulled for $T_{r+1}$ times);
let $\sigma_r : [n_r] \to I_r$ be a bijection such that $\hat{\mu}_{\sigma(1)}^{(r)} \geq \ldots \geq \hat{\mu}_{\sigma(n_r)}^{(r)}$;
by calling COLLABSEARCH$\Big(\{\hat{\mu}_i^{(r)}\}_{i \in I_r^1}, \ldots, \{\hat{\mu}_i^{(r)}\}_{i \in I_r^K}, (n_r - m_r + 1)\Big)$ and COLLABSEARCH$\Big(\{\hat{\mu}_i^{(r)}\}_{i \in I_r^1}, \ldots, \{\hat{\mu}_i^{(r)}\}_{i \in I_r^K}, (n_r - m_r)\Big)$, Coordinator finds $\hat{\mu}_{\sigma_r(m_r)}^{(r)}$ and $\hat{\mu}_{\sigma(m_r+1)}^{(r)}$ and sends them to all agents;
for each $k \in [K]$, the $k$-th agent locally computes $\hat{\Delta}_i^{(r)} = \max\left\{\hat{\mu}_i^{(r)} - \hat{\mu}_{\sigma_r(m_r+1)}^{(r)}, \hat{\mu}_{\sigma_r(m_r)}^{(r)} - \hat{\mu}_i^{(r)}\right\}$ for each $i \in I_r^k$;
let $\pi_r : [n_r] \to I_r$ be a bijection such that $\hat{\Delta}_{\pi_r(1)}^{(r)} \leq \ldots \leq \hat{\Delta}_{\pi_r(n_r)}^{(r)}$;
by calling COLLABSEARCH$\Big(\{\hat{\Delta}_i^{(r)}\}_{i \in I_r^1}, \ldots, \{\hat{\Delta}_i^{(r)}\}_{i \in I_r^K}, n_{r+1}\Big)$, Coordinator finds $\hat{\Delta}_{\pi_r(n_{r+1})}^{(r)}$ and sends it to all agents;
for each $k \in [K]$, the $k$-th agent forms the sets $E_r^k \leftarrow \left\{i \in I_r^k \;\middle|\; \hat{\Delta}_i^{(r)} > \hat{\Delta}_{\pi_r(n_{r+1})}^{(r)}\right\}$ and $A_r^k \leftarrow \left\{i \in E_r^k \;\middle|\; \hat{\mu}_i^{(r)} \geq \hat{\mu}_{\sigma_r(m_r)}^{(r)}\right\}$;
for each $k \in [K]$, the $k$-th agent sends $\left|A_r^k\right|$ to Coordinator;
Coordinator sets $m_{r+1} \leftarrow m_r - \sum_{k=1}^K \left|A_r^k\right|$;
for each $k \in [K]$, the $k$-th agent sets $I_{r+1}^k \leftarrow I_r^k \setminus E_r^k$, and updates $Q_{r+1}^k \leftarrow Q_r^k \cup A_r^k$;
**return** $\left\{(I_{r+1}^k, Q_{r+1}^k)\right\}_{k \in [K]}$.

**Algorithm 3:** GLOBALELIM$(r)$

Coordinator distributes the pulls to the $K$ agents such that each arm in $I_r$ is pulled $(T_{r+1} - T_r)$ times. Concretely, let $Q \leftarrow$ BALANCEDPULLDIST$(I_r, T_{r+1} - T_r)$; for each $(i, k, t) \in Q$, Coordinator requests agent $k$ to pull arm $i$ for $t$ times;
agents send the empirical mean of each arm they have pulled to Coordinator;
Coordinator computes for each arm $i \in I_r$ its empirical mean $\hat{\mu}_i^{(r)}$ (after being pulled for $T_{r+1}$ times);
let $\sigma_r : [n_r] \to I_r$ be a bijection such that $\hat{\mu}_{\sigma(1)}^{(r)} \geq \ldots \geq \hat{\mu}_{\sigma(n_r)}^{(r)}$;
Coordinator computes $\hat{\Delta}_i^{(r)} = \max\left\{\hat{\mu}_i^{(r)} - \hat{\mu}_{\sigma_r(m_r+1)}^{(r)}, \hat{\mu}_{\sigma_r(m_r)}^{(r)} - \hat{\mu}_i^{(r)}\right\}$;
let $\pi_r : [n_r] \to I_r$ be a bijection such that $\hat{\Delta}_{\pi_r(1)}^{(r)} \leq \ldots \leq \hat{\Delta}_{\pi_r(n_r)}^{(r)}$ ;
Coordinator computes $E_r \leftarrow \left\{i \in I_r \;\middle|\; \hat{\Delta}_i^{(r)} > \hat{\Delta}_{\pi_r(n_{r+1})}^{(r)}\right\}$, $A_r \leftarrow \left\{i \in E_r \;\middle|\; \hat{\mu}_i^{(r)} \geq \hat{\mu}_{\sigma_r(m_r)}^{(r)}\right\}$, and sets $I_{r+1} \leftarrow I_r \setminus E_r$, $Q_{r+1} \leftarrow Q_r \cup A_r$;
**return** $(I_{r+1}, Q_{r+1})$.

are at least the $m_r$-th largest empirical mean, and discards the rest. The key feature of GLOBALELIM is that all the acceptance/elimination decisions are made by the coordinator.

Due to space constraints, we leave the detailed descriptions of the two auxiliary subroutines COLLABSEARCH and BALANCEDPULLDIST to the full version of this paper.

**Remark 1.** *We remark that the concept of* balanced partition *also appeared in the work of Wang et al. (2020b) when they studied the regret minimization problem in the collaborative learning model, but there are some notable differences between the DEMAB algorithm in Wang et al. (2020b) and our Algorithm 1 for top-$m$ arm identification. First, in DEMAB, in the $r$-th round each agent locally eliminates arms whose estimated means are at least $2^{-r}$ away from that of the best arm. It is not clear how to extend this idea to the fixed-budget setting and achieve a similar success probability as (4) for best/top-$m$ arm identifications. Recall that in Algorithm 1, we eliminate the half of the (global) remaining arms with the lowest estimated means. Second, in DEMAB, every time the partition becomes unbalanced, it performs a re-balancing step. While in Algorithm 1, we do not use any re-balancing step; the algorithm just enters the* GLOBALELIM *phase when the partition becomes unbalanced.*

## The Analysis

We show the following theorem for Algorithm 1. All logarithms, unless otherwise stated, have a base of 2.

**Theorem 2.** *Algorithm 1 returns the set of $m$ arms with highest means with probability at least*

$$1 - 2n\log(2n) \cdot \exp\left(-\frac{TK}{128H^{\langle m \rangle}\log(2n)}\right), \qquad (4)$$

*uses $T$ time steps and $O(K\log^2 n + m)$ words of communication.*

Setting $T = 500H^{\langle m \rangle}\log^2 n/K$, the success probability of (4) is at least 0.99. According to the definition of speedup (Eq. (2)), we have the following corollary.

**Corollary 3.** *Algorithm 1 achieves a speedup of $\Omega(K/\log^2 n)$ using $O(K\log^2 n + m)$ words of communication.*

In the rest of the section, we analyze Algorithm 1.

**Correctness.** Let $\texttt{Top}(I) \subseteq I$ be the subset of $m$ arms with the highest means. We prove the correctness of Algorithm 1 by induction. Let $I_r = \bigcup_{k\in[K]} I_r^k$, $E_r = \bigcup_{k\in[K]} E_r^k$, $A_r = \bigcup_{k\in[K]} A_r^k$, and $Q_r = \bigcup_{k\in[K]} Q_r^k$. We show that for any $r = 0, 1, \ldots, R$,

$$(Q_r \subseteq \texttt{Top}(I)) \wedge (\texttt{Top}(I) \subseteq Q_r \cup I_r) \tag{5}$$

holds with high probability.

In the base case when $r = 0$, $(\emptyset = Q_0 \subseteq \texttt{Top}(I)) \wedge (\texttt{Top}(I) \subseteq I_0 = I)$ holds trivially.

Let $\pi : [n] \to I$ be a bijection such that $\Delta_{\pi(1)} \leq \ldots \leq \Delta_{\pi(n)}$. To conduct the induction step, we introduce the following event.

$$\mathcal{E} : \forall r = 0, \ldots, R-1, \forall i \in I_r : \left|\mu_i - \hat{\mu}_i^{(r)}\right| < \frac{\Delta_{\pi(n_{r+1})}}{8}.$$

The following claim says that $\mathcal{E}$ happens with high probability. Due to space constraints, we delay the proof to the full version of this paper.

**Claim 4.**

$$\Pr[\mathcal{E}] \geq 1 - 2n\log(2n) \cdot \exp\left(-\frac{TK}{128H^{\langle m\rangle}\log(2n)}\right).$$

We assume $\mathcal{E}$ holds in the rest of the analysis. The following lemma implements the induction step. Its proof is technical, and can be found in the full version of this paper.

**Lemma 5.** *In the execution of Algorithm 1, for any $r = 0, 1, \ldots, R-1$, if $Q_r \subseteq \texttt{Top}(I) \subseteq Q_r \cup I_r$, then* LocalElim$(r)$ *(or* GlobalElim$(r)$*) returns $(I_{r+1}, Q_{r+1})$ such that $Q_{r+1} \subseteq \texttt{Top}(I) \subseteq Q_{r+1} \cup I_{r+1}$.*

Lemma 5, together with the trivial base case, gives (5).

Note that for $r = R$, we have $I_R = \emptyset$. By (5), we have $Q_R \subseteq \texttt{Top}(I)$ and $\texttt{Top}(I) \subseteq Q_R$, which implies $Q_R = \texttt{Top}(I)$, and thus the correctness of Algorithm 1.

**Communication Cost.** We analyze the communication cost of LocalElim and GlobalElim separately. We start with LocalElim. The following lemma gives the communication cost of each call of CollabSearch; the proof can be found in the full version of this paper.

**Lemma 6.** *The communication cost of* CollabSearch *(Algorithm 4) is bounded by $O\left(K \log \left|\bigcup_{k\in[K]} A_k\right|\right)$.*

The next lemma bounds the communication cost of LocalElim.

**Lemma 7.** *The communication cost of each call of* LocalElim *is $O(K \log n)$ words.*

*Proof.* By Lemma 6, Line 3 and Line 6 use $O(K \log n)$ communication. It is easy to see that Line 8 uses $O(K)$ communication, and other steps do not need communication. Therefore, the total communication is bounded by $O(K \log n)$. □

Before analyzing the communication cost of GlobalElim, we need the following technical lemma which states that the partition $\{I_r^k\}_{k\in[K]}$ is balanced when the set of remaining arms $I_r$ is large enough. In other words, Algorithm 1 enters the second phase (which calls GlobalElim) only when $|I_r|$ becomes sufficiently small. The proof of the lemma can be found in the full version of this paper.

**Lemma 8.** *At round $r$, if $n_r \geq 100K \log n$, then the partition $\{I_r^k\}_{k\in[K]}$ is balanced with probability at least $1 - \frac{K}{n^3}$.*

Now we are ready to bound the communication cost of GlobalElim.

**Lemma 9.** *The communication cost of each call of* GlobalElim *is $O(K \log n)$ words.*

*Proof.* By Lemma 8, we know that when calling GlobalElim$(r)$, the number of arms in $I_r$ is at most $100K \log n$. The communication cost of the subroutine BalancedPullDist (Algorithm 5 in the full versoin of this paper) can be bounded by $O(K + |I_r|) = O(K \log n)$, since the number of messages the coordinator sends to each agent $k \in [K]$ is at most $(2 + \lfloor |I_r| / K \rfloor)$. □

Combining Lemma 7 and Lemma 9, and noting that there are $R = O(\log n)$ rounds and the communication at Line 1 and Line 10 of Algorithm 1 is bounded by $O(K \log n + m)$, we have

**Lemma 10.** *The communication cost of Algorithm 1 is bounded by $O(K \log^2 n + m)$.*

**Time Complexity.** We start by analyzing LocalElim. Recall that we call LocalElim$(r)$ only when the partition $\{I_r^k\}_{k\in[K]}$ is balanced, which implies that for any $r$, $\max_{k\in[K]} \left\{\left|I_r^k\right|\right\} \leq 2n_r/K$. Therefore, the number of pulls each agent makes in LocalElim$(r)$ is $\left|I_r^k\right|(T_{r+1} - T_r) \leq \frac{2n_r}{K}(T_{r+1} - T_r)$.

In GlobalElim, in the $r$-th round, each agent makes $\lceil n_r(T_{r+1} - T_r)/K \rceil \leq \frac{2n_r}{K}(T_{r+1} - T_r)$ pulls.

Thus, the total running time of Algorithm 1 can be bounded by

$$\sum_{r=0}^{R-1} \frac{2n_r(T_{r+1} - T_r)}{K} \leq \sum_{r=0}^{R-1} \left(\frac{2n}{2^r K} \cdot \frac{TK2^{r+1}}{4nR}\right) \leq T. \tag{6}$$

## Communication Lower Bound

The following theorem gives a communication lower bound for collaborative top-$m$ arm identification on IID data.

**Theorem 11.** *Any collaborative learning algorithm for top-$m$ arm identification that achieves a speedup of $\beta$ needs to use $\Omega(\beta + m)$ bits of communication.*

*Proof.* We first show that $\Omega(\beta)$ bits of communication is needed. We argue that at least $\Omega(\beta)$ agents need to be involved in order to achieve a speedup of $\beta$. Suppose that there is a collaborative algorithm for top-$m$ arm identification that involves $\beta' = o(\beta)$ agents and achieves a speedup of $\beta$, then by the definition of the speedup (Eq. (2)), there is a centralized algorithm that solves the same problem with running time $\beta' \cdot H^{\langle m\rangle}(I)/\beta = o(H^{\langle m\rangle}(I))$ (e.g., by concatenating the pulls of the $\beta'$ agents), contradicting to the $\Omega(H^{\langle m\rangle})$ time lower bound in (Chen, Li, and Qiao 2017a).

We next show that $\Omega(m)$ is also a lower bound of the communication cost. This is simply because we require the coordinator to output the top-$m$ arms at the end, and the coordinator cannot pull arms itself. Note that the cost of

communicating the indices of the top-$m$ arms out of the $n$ arms is $\log \binom{n}{m} = \Omega(m)$ (given $n \geq 2m$; see the definition of the top-$m$ arm identification problem in the introduction).

Summing up, the total bits of communication is at least $\Omega(\max\{\beta, m\}) = \Omega(\beta + m)$. □

Particularly, for $\beta = \tilde{\Omega}(K)$, we have:

**Corollary 12.** *Any collaborative learning algorithm for top-$m$ arm identification that achieves a speedup of $\tilde{\Omega}(K)$ needs to use $\tilde{\Omega}(K + m)$ bits of communication.*

Combined with Corollary 3, our upper and lower bounds are tight up to logarithmic factors.

## Learning with Non-IID Data

### Algorithm and Analysis

The algorithm for top-$m$ arm identification in the non-IID data setting is very similar to GLOBALELIM (Algorithm 3), except that in the non-IID setting, the workload distribution at the beginning of each round is more straightforward: each of the $K$ agents simply pull each of the remaining arms in $I_r$ for $(T_{r+1} - T_r)/K$ times. These information is enough for the coordinator to calculate the empirical global means of the $n$ arms for the elimination process. Due to space constraints, we leave the full description of Algorithm 6 to the full version of this paper.

We have the following theorem, whose proof can be found in the full version of this paper.

**Theorem 13.** *There is an algorithm (Algorithm 6 in the full version of this paper) that solves top-$m$ arm identification in the non-IID data setting with probability at least*

$$1 - 2n\log(2n) \cdot \exp\left(-\frac{TK}{128H^{\langle m \rangle}\log(2n)}\right), \quad (7)$$

*uses $T$ time steps and $O(Kn\log n)$ words of communication.*

Setting $T = 500H^{\langle m \rangle}\log^2 n/K$, the success probability of (7) is at least 0.99.

**Corollary 14.** *Algorithm 6 achieves a speedup of $\Omega(K/\log^2 n)$ using $O(Kn\log n)$ words of communication.*

### Communication Lower Bound

The following theorem states that in the non-IID data setting, even if each agent knows the local means of all the $n$ arms exactly, they still need to spend $\Omega(nK)$ bits of communication to solve top-1 (i.e., $m = 1$) arm identification. This communication lower bound holds regardless the speedup of the collaborative algorithm. The proof of the theorem makes use of a reduction from a problem called *Set-Disjointness* in multi-party communication complexity.

**Theorem 15.** *Any collaborative learning algorithm that solves top-1 arm identification with probability 0.99 needs $\Omega(nK)$ bits of communication.*

*Proof.* We make use of a problem called $K$-Set-Disjointness (*DISJ*) in multi-party communication complexity. In the coordinator model, the *DISJ* problem can be described as follows: each agent $k$ ($k \in [K]$) holds a $n$-bit vector $X^k = (X_1^k, \ldots, X_n^k) \in \{0,1\}^n$, and the coordinator has no input. The $K$ agents and the coordinator want to compute the following function via communication:

$$\textit{DISJ}(X^1, \ldots, X^K) = \begin{cases} 1, & \exists i \in [n] \text{ s.t. } \forall j \in [k], X_i^j = 1, \\ 0, & \text{otherwise.} \end{cases}$$

It is known that any randomized algorithm that solves *DISJ* with probability at least 0.99 needs $\Omega(nk)$ bits of communication (Braverman et al. 2013).

We first perform an input reduction from *DISJ* to top-1 arm identification. Each agent $k$ converts its input vector (of *DISJ*) $X^k = (X_1^k, \ldots, X_n^k) \in \{0,1\}^n$ to $n$ arms (of top-1 arm identification) with the following means: the $i$-th arm has a local mean of

$$(1 + X_i^k)/3 + i\delta,$$

where $\delta = 1/n^2$ is a small noise that we add to means to make sure that the best arm is unique. The global mean of the $i$-th ($i \in [n]$) arm is

$$\mu_i = \frac{1}{K}\sum_{k\in[K]}\left(\frac{1 + X_i^k}{3} + i\delta\right) \in (0,1).$$

We further introduce a special arm (the $(n+1)$-th arm) with all $K$ local means being

$$\mu_{n+1,1} = \ldots = \mu_{n+1,K} = \frac{n-1}{n} + \frac{1}{2n}.$$

Thus $\mu_{n+1} = \frac{n-1}{n} + \frac{1}{2n}$.

Clearly, for any $i \in [n]$ such that $X_i^1 = \ldots = X_i^K = 1$, we have $\mu_i > \mu_{n+1}$. On the other hand, for any $i \in [n]$ such that there exists a $k \in [K]$ such that $X_i^k = 0$, we have $\mu_i < \mu_{n+1}$. Therefore, if the top-1 arm is the $(n+1)$-th arm, then *DISJ*$(X^1, \ldots, X^K) = 0$. Otherwise, if the top-1 arm is *not* the $(n+1)$-th arm, then *DISJ*$(X^1, \ldots, X^K) = 1$. Therefore, any algorithm $\mathcal{A}$ that solves top-1 arm identification on the $(n+1)$ arms with aforementioned local means can also solve the original *DISJ* problem. The theorem follows from the $\Omega(nK)$ lower bound of the *DISJ* problem. □

## Experiments

In this section, we present the experimental study on our algorithms.

**Algorithms.** In the IID data setting, we implement Algorithm 1 (denoted by `CollabTopIID`), and compare it with the (only) existing algorithm (denoted by `KZZ`) for collaborative top-$m$ arm identification (Karpov, Zhang, and Zhou 2020). We note that the original design goal of `KZZ` is to minimize the number of rounds of the collaborative learning process. The communication cost of `KZZ` is $\Omega(Kn)$, which is far from being optimal.

In the non-IID data setting, we have implemented Algorithm 6 (denoted by `CollabTopNIID`) and tested its performance. Since there is no direct competitor in the non-IID data setting, we create a baseline algorithm named `Uniform`. `Uniform` only uses one round. Given a time horizon $T$, each agent pulls each of the $n$ arms for $T/n$ times, and then sends

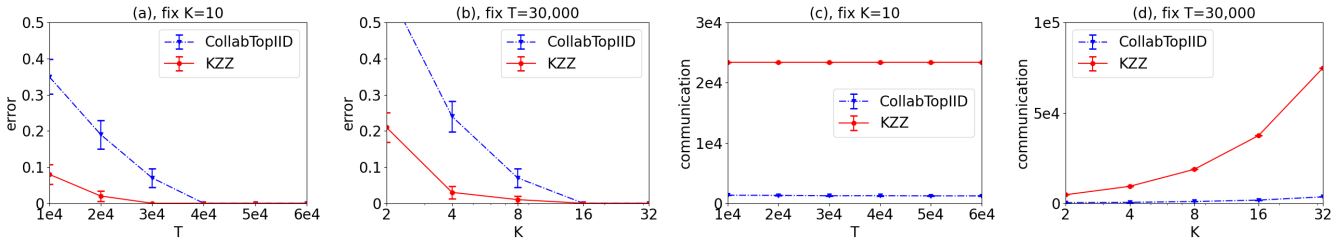


Figure 1: Performance of algorithms for top-1 arm identification in the IID data setting.

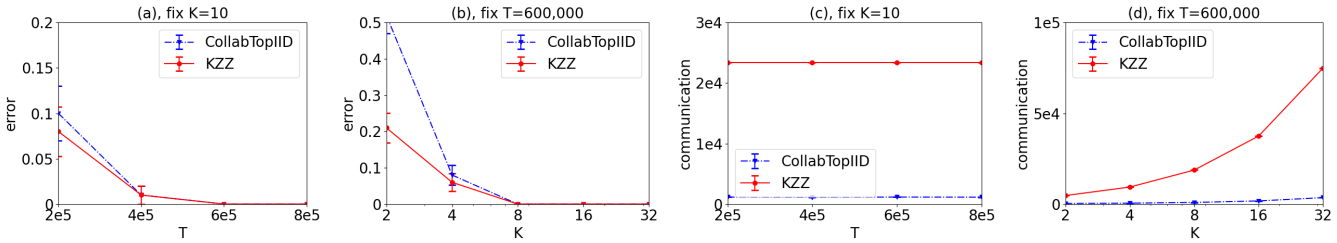


Figure 2: Performance of algorithms for top-8 arm identification in the IID data setting.

the local empirical means to the coordinator. The coordinator computes for each arm its global empirical mean, and then outputs the one with the highest global mean. Let $\Delta_{\min}$ be the mean gap between the best arm and the second best arm. It is easy to show that when $T \geq \frac{c_T n \log(Kn)}{K\Delta_{\min}^2}$ for a sufficiently large constant $c_T$, then `Uniform` correctly outputs the best arm with probability at least $0.99$. The communication cost of `Uniform` is $Kn$, which is the *lower bound* in the non-IID data setting (recall Theorem 15).

**Datasets.** We use a real-world dataset MovieLens (Harper and Konstan 2016). We select $588$ movies scored by at least $20,000$ users. For the $i$-th movie, we set $\mu_i$ to be the average rating of the movie divided by $5$ (to make sure that $\mu_i \in (0,1)$). In the IID data setting, we regard each movie as an arm associated with a Bernoulli distribution of mean $\mu_i$.

In the non-IID data setting, we split users into $K = 10$ groups (the user with ID $x$ is included in Group ($x \bmod 10$)), and select $574$ movies scored by at least $1,000$ users in each of the $K$ groups. For the $i$-th movie, we set $\mu_{i,k}$ to be the average rating of the movie in group $k$ divided by 5.

**Computation Environment.** All algorithms are implemented using programming language Kotlin. All experiments were conducted in PowerEdge R740 server equipped 2×Intel Xeon Gold 6248R 3.0 GHz (24-core/48-thread per CPU) and 256GB RAM.

**Results.** Our experimental results for the IID data setting are depicted in Figure 1 (for $m = 1$) and Figure 2 (for $m = 8$). All results are averages over $100$ runs. The error bars stand for the standard deviation.

Figure 1(a) shows the influence of the time horizon $T$ on the error probability of outputting the best arm. We fix the number of agents to be $10$. We observe that the error probabilities of both `CollabTopIID` and `KZZ` quickly converge to zero. `KZZ` converges slightly faster than `CollabTopIID`, which is because in order to achieve the communication efficiency, in the first phase of `CollabTopIID`, the sizes of local arm sets across the $K$ agents may not be equal, which could result idle agents at a constant fraction of time steps. To achieve error probability $0.01$, `KZZ` outperforms `CollabTopIID` by a factor of 1.3 in running time.

Figure 1(b) shows the influence of the number of agents $K$ on the error probability of outputting the best arm. We fix the time horizon to be $30,000$. The curves are similar to those in Figure 1(a). For the same reason mentioned above, `KZZ` converges faster than `CollabTopIID`.

In Figure 1(c) and (d), we compare the communication cost of two collaborative algorithms `CollabTopIID` and `KZZ`. In Figure 1(c) we fix $K$ and vary $T$, while in Figure 1(d) we fix $T$ and vary $K$. It is clear that `CollabTopIID` significantly outperforms `KZZ`. For example, for $K = 10$ and $T = 30,000$, the communication cost of `KZZ` is 20 times of that of `CollabTopIID`. Recall that the communication cost of `CollabTopIID` is $O(K \ln^2 n + m)$, which is independent of $T$; the same holds for `KZZ`. This is why the two curves in Figure 1(c) are flat.

Figure 2 presents the results when $m$ is set to $8$ (i.e., we try to identify the top-8 arms). Generally speaking, the performance trends for `CollabTopIID` and `KZZ` are similar as that in the case of $m = 1$. The running time gaps between the two algorithms are smaller in Figure 2(a) and (b), compared with that in Figure 1(a) and (b). For example, when $K$ is fixed to be 10, to achieve error probability $0.01$, the running time of `KZZ` and `CollabTopIID` are the same.

Due to space constraints, we leave the experimental results on non-IID data to the full version of this paper.

## Acknowledgments

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

## References

Audibert, J.; Bubeck, S.; and Munos, R. 2010. Best Arm Identification in Multi-Armed Bandits. In *COLT*, 41–53.

Braverman, M.; Ellen, F.; Oshman, R.; Pitassi, T.; and Vaikuntanathan, V. 2013. A Tight Bound for Set Disjointness in the Message-Passing Model. In *FOCS*, 668–677.

Bubeck, S.; Wang, T.; and Viswanathan, N. 2013. Multiple identifications in multi-armed bandits. In *ICML*, 258–265.

Chen, L.; Li, J.; and Qiao, M. 2017a. Nearly Instance Optimal Sample Complexity Bounds for Top-k Arm Selection. In *AISTATS*, 101–110.

Chen, L.; Li, J.; and Qiao, M. 2017b. Towards Instance Optimal Bounds for Best Arm Identification. In *COLT*, 535–592.

Harper, F. M.; and Konstan, J. A. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 5(4): 19:1–19:19.

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

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K. A.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konečný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Raykova, M.; Qi, H.; Ramage, D.; Raskar, R.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2019. Advances and Open Problems in Federated Learning. *CoRR*, abs/1912.04977.

Karnin, Z.; Koren, T.; and Somekh, O. 2013. Almost Optimal Exploration in Multi-Armed Bandits. In *ICML*, 1238–1246.

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

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

Mitra, A.; Hassani, H.; and Pappas, G. J. 2021. Exploiting Heterogeneity in Robust Federated Best-Arm Identification. *CoRR*, abs/2109.05700.

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 *FOCS*, 126–146.

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. PMLR.

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

Wegman, M. N.; and Carter, L. 1981. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci.*, 22(3): 265–279.