Collaborative Learning in General Graphs with Limited Memorization: Learnability, Complexity and Reliability

Feng Li, Member, IEEE, Xuyang Yuan, Lina Wang, Huan Yang, Member, IEEE, Dongxiao Yu, Member, IEEE, Weifeng Lv, Member, IEEE, Xiuzhen Cheng, Fellow, IEEE,

Abstract—We consider $K$-armed bandit problem in general graphs where agents are arbitrarily connected and each of them has limited memorization and communication bandwidth. The goal is to let each of the agents learn the best arm. Although recent studies show the power of collaboration among the agents in improving the efficacy of learning, it is assumed in these studies that the communication graphs should be complete or well-structured, whereas such an assumption is not always valid in practice. Furthermore, limited memorization and communication bandwidth also restrict the collaborations of the agents, since very few knowledge can be drawn by each agent from its experiences or the ones shared by its peers in this case. Additionally, the agents may be corrupted to share falsified experience, while the resource limit may considerably restrict the reliability of the learning process. To address the above issues, we propose a three-staged collaborative learning algorithm. In each step, the agents share their experience with each other through light-weight random walks in the general graphs, and then make decisions on which arms to pull according to the randomly memorized suggestions. The agents finally update their adoptions (i.e., preferences to the arms) based on the reward feedback of the arm pulling. Our theoretical analysis shows that, by exploiting the limited memorization and communication resources, all the agents eventually learn the best arm with high probability. We also reveal in our theoretical analysis the upper-bound on the number of corrupted agents our algorithm can tolerate. The efficacy of our proposed three-staged collaborative learning algorithm is finally verified by extensive experiments on both synthetic and real datasets.

Index Terms—Multi-armed bandits, collaborative learning, limited memorization, general topology.

1 INTRODUCTION

Given a set of unknown options, making a sequence of decisions to choose among them is a commonly encountered issue in a wide spectrum of applications, e.g., economy [1], robotics [2], and biology [3]. The problem is usually formulated as a stochastic Multi-Armed Bandit (MAB) problem [4], [5]. Specifically, a $K$-armed bandit problem is defined as follows. Given $K$ arms $a_1, a_2, \ldots, a_K$, a player (a.k.a. agent) can select one of them to pull according to some policy, and observe the corresponding reward feedback in each step. Let $\phi_{k,r}$ denote the reward obtained by the $r$-th pull of arm $a_k$, and $\phi_{k,1}, \phi_{k,2}, \ldots$ are assumed to be i.i.d. random variables (e.g., Bernoulli variables parameterized by unknown $p_k$). The aim is to design a policy, according to which the player can make sequential selection decisions to learn the best arm (which yields the highest reward on expectation) with the resulting cumulative revenue maximized.

Recent studies, e.g., [6], [7], [8], have investigated a variation of the MAB problem where multiple agents independently make decisions to pull the arms; nevertheless, most of them focus on addressing the collisions/interference among the agents, whereas a very handful of recent proposals exploit the collaboration among the players. Inspired by the fact that individuals in a social group (such as human society, social insect colonies and swarm robotics) can learn experiences from their peers [3], [9], [10], [11], [12], a collaborative learning dynamics consists of the following two stages in each step: in the sampling stage, each agent chooses one of the arms to pull based on the suggested adoptions from its peers, while in the adopting stage, the agent decides whether or not to adopt the chosen arm as preference according to the stochastic reward feedback. In fact, the above two-staged collaborative learning dynamics has been investigated in recent studies [13], [14]. Unfortunately, the existing proposals consider either complete graphs or well-structured ones such that the information exchange among agents can be guaranteed. For example, in [13], the agents can directly communicate with each other and observe the exact popularity of each arm and [14] assumes that the communication graph is regular or doubly-stochastic. Therefore, it is a very challenging issue to let the agents learn collaboratively in a general graph with arbitrary topology.

The collaboration among the agents is also restricted by their limited memorizing capacities. In the well-known Upper Confidence Bound (UCB) algorithm, an agent is required to memorize the cumulative rewards and the total number
of pulls for each arm [4]. However, such a requirement may not always be fulfilled, since the agents usually have limited memory [14], [15]. An agent (e.g., a human being) could remember only the most recent arm pulling which yields reward. Additionally, although receiving multiple suggestions from its peers during the learning process, a “forgetful” agent may remember only part of them. Therefore, the question is, with limited history of the learning process and limited knowledge drawn from their peers, are the agents still able to learn the best arm in the end?

Our another concern is the fault tolerance of the collaborative learning. Some of the agents may be corrupted by an adversary to elaborately share falsified experience with their peers. One popular choice to tolerate such faults is to let the agents make their decisions by collecting redundant experience from their peers. Nevertheless, considering the above memorization limit, whether or not the information redundancy can be utilized to reliably serve the learning goal is still an open problem.

In this paper, we propose a collaborative learning algorithm for the multiplayer MAB problem in a general graph where each agent has bounded memorizing capacity and may be corrupted to disseminate falsified information. Specifically, each round in our collaborative learning algorithm includes the following three stages:

- **Disseminating**: For each agent, if it has a preference over the K arms (and thus has a non-null adoption), it disseminates its adoption over the graph through Metropolis-Hasting Random Walks (MHRWs) in parallel.
- **Sampling**: For each agent with no preference (and thus with a null adoption), with probability μ, it chooses one of the K arms uniformly at random to pull; with probability 1 − μ, it randomly chooses one of the arms suggested by its peers in the last disseminating stage (i.e., the one it can memorize among all the suggestions) or chooses no arm if there is no suggestion received. For the agents with non-null adoptions, each of them makes its choice by the second branch (i.e., by letting μ = 0).
- **Adopting**: If pulling the arm yields reward, the agent updates its adoption to the arm; otherwise, it keeps its adoption unchanged.

We study the dynamics of the above three-staged collaborative learning algorithm from the perspectives of learnability, complexity and reliability, respectively. We demonstrate the reliability of our collaborative learning algorithm. At most \((1-\alpha)(p_1+p_2)N\) corrupted nodes can be tolerated in each round when the number of agents adopting the best arm is at most \(\alpha N\) (0 < \(\alpha\) < 1), where \(p_1\) and \(p_2\) denote probabilities for the best arm and the second best one to yield rewards, respectively.

The remaining of this paper is organized as follows. We first survey related literature in Sec. 2. We then introduce our system model and formulate our problem in Sec. 3. The details of our three-staged collaborative learning algorithm and the corresponding theoretic analysis are then given in Sec. 4 and Sec. 5, respectively. We also perform extensive numerical experiments to verify the efficacy of our proposed algorithm in Sec. 6. We finally conclude this paper in Sec. 7.

## 2 Related Work

MAB is a very powerful framework for designing algorithms which make decisions over time under uncertainty [16]. Although there has been a vast body of work investigating single agent MAB problem (e.g., [4], [5], [17], [18], [19]), proposals on how multiple agents work collaboratively were rather rare until recent years.

The power of the collaboration to improve the efficiency of learning process has been revealed in [20]; nevertheless, most of exiting proposals focus on utilizing rich historical data (e.g., about which arms were pulled and how many rewards were obtained through the pulls in the past) in a distributed manner. Therefore, those methods to address single agent MAB problems (e.g., UCB method, \(\varepsilon\)-greedy method and SoftMax method [4], [21]) are still very useful for resolving the multi-agent version. In [22], each agent either chooses one of the arms to pull or broadcast its local historical data. The agents which choose to pull arms make their decisions based on SoftMax method. In [23] proposes a gossip-based algorithm for the MAB problem in Peer-to-Peer (P2P) networks. Specifically, in every round, each agent shares its empirical data to two randomly chosen neighbors and the agent performs \(\varepsilon\)-greedy method to make a decision to choose an arm to pull. However, this algorithm relies on constructing an overlay network with special topology. General social graphs (with arbitrary topologies) are considered in [24] where a hierarchical learning algorithm is adopted. In [24], dominating set of a graph should be first recognized. Each agent in the dominating set (i.e., the so-called “leader”) applies a UCB-based learning policy to choose among the arms according to historical information collected from its one-hop neighbors, while each of the others follows the action taken by its leader. The UCB policy is also used in [25]; nevertheless, since only a limited number of bits (i.e., the ID of the recommended arm) can be shared by each agent to a random peer, the UCB policy is based on incomplete historical information.

However, rich historical data may not always be available for individual agents, since an agent may have no sufficient memory. Referring to human choice behavior [9], [11], [12] and animal behavior [3], [10], a two-staged algorithmic paradigm for collaborative learning is considered in [13], [14], which includes sampling stage and adopting stage as mentioned in Sec. 1. In [13], arms are sampled according to their popularities, calculating which in a complete graph is easy to complement but may induce considerable communication overhead in a general graph. [14] considers memory-bounded agents. Specifically, each agent has a finite-valued memory such that only the latest adoption can be memorized [15], [26]. Nevertheless, the algorithm proposed in [14] works with regular or doubly-stochastic graphs.
Different from the stochastic MAB problems investigated in the above proposals, another variant of MAB is (non-stochastic) adversarial MAB where reward feedback is controlled by an adversary [5], [27], [28], [29], [30]. In [30], arms yield Bernoulli rewards and a malicious agent recommends an arbitrary arm instead of the one which it believes is the best. Although the collaborative learning algorithm proposed in [30] is of high robustness in face of malicious node, the agents adopt UCB policy to make arm selection decisions with no memorization limit considered.

3 System Model and Problem Description

Multi-agent graph. We consider a communication graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \cdots, N\}$ denotes a group of agents and $\mathcal{E}$ is a set of edge among $\mathcal{N}$. If there is an edge between agents $i$ and $i' \in \mathcal{N}$, they can exchange messages with each other. Each agent $i$ has a set of neighbors $\mathcal{N}_i \subseteq \mathcal{N}$ and let $d_i = |\mathcal{N}_i|$ denote the degree of agent $i$. We suppose that the graph $G$ is connected and non-bipartite, while there is no any other assumption on the structure of the graph. It should be noted that, the assumption is only for our theoretical analysis and our algorithm still works even the assumption does not strictly hold. We also suppose that the agents are synchronized such that time can be divided into a sequence of time slots $t = 1, 2, \cdots$. We employ the CONGEST model to characterize the communications among the agents, which has been highly recognized in distributed computing and communication [31], [32], [33], [34], [35]. By the CONGEST model, an agent $i$ transmits up to $O(\log N)$ messages to each of $\mathcal{N}_i$ in a slot, while each message consists of $O(\log N)$ bits. As will be shown in Sec. 4, our collaborative learning algorithm proceeds iteratively. Each round $r$ is composed by $R$ slots and each agent can choose an arm to pull in each round.

Collaborative Learning. The agents collaboratively solve a $K$-armed stochastic bandit problem. For each arm $a_k$ $(k = 1, \cdots, K)$, the reward process is a Bernoulli process parameterized by $p_k$. In another word, if arm $a_k$ is pulled in round $r$, the obtained reward $\phi_{k,r}$ is Bernoulli($p_k$) such that

$$\phi_{k,r} = \begin{cases} 1, & \text{with probability } p_k; \\ 0, & \text{otherwise}. \end{cases}$$

Without loss of generality, we assume there exists a unique best arm (i.e., $a_1$) and $p_1 > p_2 \geq p_3 \geq \cdots \geq p_K \geq 0$. We also suppose that $p_1, p_2, \cdots, p_K$ are unknown to the agents initially. Our goal is to design a learning algorithm, based on which, the agents in graph $G$ can collaboratively make decisions sequentially (through exchanging experiences) to choose among the arms to pull, so as to learn the best arm $a_1$ according to reward feedback.

We assume that an agent has at most one adoption (or preference) over the $K$ arms in each round. Since an agent may have no preference over the $K$ arms, we hereby introduce a “virtual” arm $a_0$ (which is so-called null arm in the following) such that an agent is said to virtually adopt $a_0$ if it has no preference. Let $X_{i,k}(r) \in \{0, 1\}$ be a binary variable indicating if agent $i$ adopts (or prefers) arm $a_k$ ($k = 0, 1, \cdots, K$) in round $r$. It is apparent that $\sum_{k=0}^{K} X_{i,k}(r) = 1$ for $\forall i, r$. Then, the adoption of agent $i$ in round $r$ can be represented by $X_i(r) = (X_{i,0}(r), X_{i,1}(r), \cdots, X_{i,K}(r))$.

Specially, if $X_{i,0}(r) = 1$ (or $X_{i,k}(r) = 0$ for $\forall k = 1, \cdots, K$), agent $i$ is said to have a null adoption in round $r$. Suppose $Z_k(r) = \sum_{i=1}^{N} X_{i,k}(r)$ denotes the number of the agents adopting arm $k$ in round $r$. Our learning process is said to succeed when all the agents learn the best arm $a_1$ within a sufficiently large time horizon. Therefore, we formally define the success event for the $N$-agent system as

$$\text{Succ}(N) \triangleq \{ Z_1(r) = N \text{ holds when } r \text{ is sufficiently large} \}.$$  

Limited memorization. We assume that each agent has limited capacity of memorizing. On one hand, an agent keeps only the current adoption in its mind [14], [15]. Specifically, each agent $i$ maintains a variable $\omega_i \in \{0, 1, \cdots, K\}$ which indicates which arm it currently prefers. Therefore, a $\lceil \log(K + 1) \rceil$-bit local memory is sufficient to represent the $K+1$ memory states. On the other hand, although there may be multiple suggestions agent $i$ can receive from its peers in each round $r$, the agent $i$ may “memorize” only one of them (i.e., $a_i(r) \in \{1, \cdots, K\}$). For example, a human being may remember only quite few of the suggestions at random which his friends told him. Likewise, the temporary variable also can be represented by $\lceil \log(K + 1) \rceil$ bits.

Adversarial setting. We assume that the agents may be corrupted by an adversary. In each round, the corrupted agents recommend random arms (instead of the ones they currently adopt) to their peers [30]. We suppose that the adversary may have limited power such that there are up to $\tau$ out of the $N$ agents suffering the corruptions. Without loss of generality, we also assume that the adversary uniformly pick up $\tau N$ agents to corrupt at random. Furthermore, we take into account time-varying corruptions and thus assume the adversary may corrupt different agents in each round.

4 Algorithm

In this section, we present our collaborative learning algorithm for general graphs with limited memorization. The algorithm is performed iteratively and each agent performs the following three stages in every round: i) in the disseminating stage, we leverage Metropolis-Hasting Random Walks (MHRWs) to let the agents share their current adoptions (a.k.a. tokens in the following) with their peers in the general graph; ii) in the sampling stage, each agent randomly chooses one arm to pull based on the received suggestions; iii) in the adopting stage, each agent updates its adoption according to the reward feedback of the pulling. The pseudo-code of our algorithm is given in Algorithm 1. Initially, each agent has a null adoption such that $\omega_i \leftarrow 0$ for $\forall i = 1, 2, \cdots, N$.

1. The length of each round, i.e. $R$, will be discussed later in Sec. 5.
the token will be forwarded in the residual disseminating stage and is initialized to \( O(\log N) \). A token is said to be feasible if it has a non-zero length counter. The agent \( i \) then launches \( h \log N \) MHRWs in parallel, each of which carries a copy of its token. The agent \( i \) uses a First-in-First-out (FIFO) queue to buffer the received feasible tokens. As shown in Lines 8-11, in each slot of round \( r \), the agent \( i \) pops the first up to \( h \log N \) feasible tokens out of the queue and then forwards the tokens individually to its neighbors according to the probability distribution \( \Psi(i, i') \) (for \( \forall i' \in N_i \cup \{i\} \))

\[
\Psi(i, i') = \begin{cases} 
\min \left\{ \frac{1}{d_i}, \frac{1}{d_{i'}} \right\}, & \text{for } \forall i' \in N_i \\
1 - \sum_{j \in N_i} \Psi(i, j), & \text{for } i = i'
\end{cases}
\]

Before being forwarded, each of the tokens has its length counter decreased by one. For each agent \( i \), once receiving a token, it pushes the token into the FIFO queue if the token is feasible (with non-zero length counter); otherwise, it merges the token into \( V_i(r) \) (see Line 12-17). Note that the tokens are disseminated in a randomized manner, an agent may receive no token such that \( V_i(r) = \emptyset \).

(2) Sampling. Considering the agents have only very limited memories, each agent may only memorize one of the received suggestions; therefore, to mimic the “forgetfulness” of the agents, we let each agent uniformly choose one of its received non-feasible tokens (if any) at random. In particular, for each agent \( i \), if it does not have any preference (i.e., \( \omega_i = 0 \)), then

- With probability \( \mu \in [0, 1] \), the agent \( i \) pulls one of the \( K \) arms uniformly at random (see Line 21);
- With probability \( 1 - \mu \), if there is no token received (i.e., \( V_i(r) = \emptyset \)), agent \( i \) does not choose any arm in round \( r \) such that \( a_i(r) = 0 \) (see Line 22a); otherwise, it choose one of the suggested arms uniformly at random (see Line 22b).

If agent \( i \) has a non-null adoption, it directly chooses one out of the suggestions \( V_i(r) \) uniformly at random, if \( V_i(r) = \emptyset \); otherwise, it chooses no arm (see Lines 24-25).

(3) Adopting. Each agent \( i \) pulls arm \( a_i(r) \) and let \( \phi_i(r) \in \{0, 1\} \) denote if pulling arm \( a_i(r) \) yields any reward. If \( \phi_i(r) = 0 \), agent \( i \) updates its state \( \omega_i \leftarrow a_i(r) \); otherwise, \( \omega_i \) is unchanged.

According to the above algorithm, we have the following propositions which will be very helpful in our later analysis.

**Proposition 1.** For every agent \( i \), if there exists round \( r \) such that \( X_{i,0}(r) \neq 0 \), we then have \( X_{i,0}(r') \neq 0 \) for \( \forall r' \geq r + 1 \).

**Proposition 2.** If there exists round \( r \) in which all agents adopt the best arms \( a_1 \) such that \( Z_1(r) = N \), then \( Z_1(r') = N \) holds for \( \forall r' \geq r + 1 \).

The first proposition states that an agent already taking a non-null adoption will not adopt the null arm thereafter; while the second one indicates that when all the agents adopt arm \( a_1 \), they will not adopt any other arms thereafter. We hereby omit the proofs as the propositions can be derived straightforwardly from the algorithm.

## 5 Analysis

### 5.1 Learnability

We hereby investigate the learnability of our collaborative learning algorithm. In this section, we focus on a simplified case with no agent corrupted and will discuss the reliability of our algorithm later in Sec. 5.3. In the following, we first show in Lemma 1 that each agent \( i \) chooses arm \( a_k \) proportional to the current popularity of \( a_k \) in round \( r \) especially when there are a sufficient number of agents, based on which, we then illustrate the number of agents adopting the best arm is increased with high probability in in
each round in Lemma 2 and Lemma 3. We finally conclude in Theorem 1 that the learnability of our algorithm can be guaranteed such that all agents eventually adopt the best arm as preference with high probability.

According to Algorithm 1, the (conditional) sampling distribution \( P(a_i(r) = a_k \mid X(r-1)) \) is dependent on the distribution of the tokens in the disseminating stage (i.e., \( X(r-1) \)). Therefore, before diving into the proof of Lemma 1, we first demonstrate how the tokens are disseminated uniformly through the MHRWs. In our algorithm, a random walk is a random process of token forwarding. In each slot, a token is forwarded by the current agent (e.g., \( i \)) to one of its neighbors according to probability \( \Psi(i, i') \) (see Eq. (3)). According to [36], the random walk achieves a nearly uniform distribution when the token is forwarded \( O(\log(\frac{N}{\delta})) \) times in a connected and non-bipartite graph \( 4 \).

In particular, the probability for the token to reach any of the agents is in the range \( \left[ \frac{1}{N} - \kappa, \frac{1}{N} + \kappa \right] \). For example, when \( \kappa = \frac{1}{\sqrt{N}} \) such that the token is forwarded \( O(\log N) \) times, it is “sampled” by any agent \( i \) in the sampling stage with probability \( \left[ \frac{1}{N} - \frac{1}{\sqrt{N}}, \frac{1}{N} + \frac{1}{\sqrt{N}} \right] \). In the following, we prove Lemma 1 by taking into account the above specific nearly uniform distribution. Lemma 1 can be surely guaranteed when the distribution is more “uniform”.

**Lemma 1.** In each round \( r \), there are \( M = h \log N \sum_{k=1}^{K} Z_k \) tokens disseminated over graph \( G \) through MHRWs. Let \( M_k(r) = Z_k(r-1)h \log N \) denote the number of \( k \)-tokens (i.e., the tokens encapsulating \( a_k \)) in round \( r \) and \( Q_k(r-1) = Z_k(r-1)/N = M_k(r)/M \) be the popularity of arm \( k \) in round \( r-1 \) (or at the beginning of round \( r \)). Given the current adoption state \( X(r-1) = \{ X_1(r-1), X_2(r-1), \ldots, X_N(r-1) \} \), we have

\[
\lim_{N \to \infty} P(a_i(r) = a_k \mid X(r-1)) = Q_k(r-1)
\]

(4)

**Proof.** Recall that, for each token, the probability that it reaches agent \( i \) in the disseminating stage is in the range \( \left[ \frac{1}{N} - \frac{1}{\sqrt{N}}, \frac{1}{N} + \frac{1}{\sqrt{N}} \right] \). Hence, the probability that it is chosen by agent \( i \) in the sampling stage of round \( r \) can be upper-bounded by

\[
\frac{1}{M} \sum_{v=1}^{M} \left( \frac{M}{1 + \frac{1}{N^3}} \right)^v \left( \frac{1}{1 - \frac{1}{N^3}} \right)^{M-v} \leq \frac{1}{M} \left( 1 + \frac{2}{N^3} \right)^M
\]

Considering there are \( M_k(r) \) \( a_k \)-tokens, each of which reaches agent \( i \) independently, we have

\[
P(a_i(r) = a_k \mid X(r-1)) \leq \frac{M_k(r)}{M} \left( 1 + \frac{2}{N^3} \right)^M = Q_k(r-1) \left( 1 + \frac{2}{N^3} \right)^{Nh \log N}
\]

Therefore,

\[
\lim_{N \to \infty} P(a_i(r) = a_k \mid X(r-1)) \leq Q_k(r)
\]

(5)

4. We hereby only discuss about how a single MHRW achieves a uniform stationary distribution; nevertheless, in our algorithm, each agent launches \( h \log N \) random walks in parallel. We will illustrate in Sec. 5.2 that how long a round should be such that each of these concurrent random walks achieves a uniform stationary distribution with high probability.

Similarly, for each token in round \( r \), the lower-bound on the probability that it is chosen by agent \( i \) in the sampling stage can be defined as

\[
\frac{1}{M} \left( 1 - \frac{2}{N^3} \right)^M - \left( 1 - \frac{1}{N} - \frac{1}{N^3} \right)^M \geq \frac{1}{M} \left( 1 - \frac{2h \log N}{N^2} - \left( 1 - \frac{1}{N} \right)^{Nh \log N} \right) \geq \frac{1}{M} \left( 1 - \frac{2h \log N}{N^2} - \left( 1 - \frac{1}{N} \right)^{Nh \log N} \right)
\]

where the first inequality holds since ii) \( 1 - \frac{2}{N^3} \) \( \geq 1 - \frac{2h \log N}{N^2} \) due to the Bernoulli inequality, and ii) \( 1 - \frac{1}{N} - \frac{1}{N^3} \leq 1 - \frac{1}{N^2} \), when \( N \geq 2 \). Therefore, we have

\[
\lim_{N \to \infty} P(a_i(r) = a_k \mid X(r-1)) \geq \lim_{N \to \infty} Q_k(r) \left( 1 - \frac{2h \log N}{N^2} - \left( 1 - \frac{1}{N} \right)^{Nh \log N} \right) \geq Q_k(r) \lim_{N \to \infty} \left( 1 - \frac{2h \log N}{N^2} - \left( 1 - \frac{1}{N} \right)^{Nh \log N} \right) = Q_k(r)
\]

(6)

We complete the proof by combining (5) and (6). \( \square \)

**Lemma 2.** Let \( Z_1^+ (r) \) and \( Z_1^- (r) \) be the number of agents whose adoptions are changed from \( a_2 \) to \( a_1 \) in round \( r \) and the number of agents whose adoptions are changed from the best arm \( a_1 \) to another one \( a_2 \in \{ a_2, \ldots, a_K \} \) in round \( r \), respectively. For any round \( r \) such that \( Z_1(r-1) < N \), we have

\[
\mathbb{E}[Z_1^+ (r) - Z_1^- (r) \mid X(r-1)] > 0
\]

(7)

Proof. Let \( q_{i,k} (r) \triangleq P(a_i(r) = a_k \mid X(r-1)) \) be the probability that agent \( i \) chooses \( a_k \) in the sampling stage of round \( r \) conditioned on \( X(r-1) \). At the beginning of round \( r \), we can divide the agents into three subsets \( \mathcal{S}_0(r-1) = \{ i \in \mathcal{N} \mid \omega_i(r-1) = a_0 \} \), \( \mathcal{S}_1(r-1) = \{ i \in \mathcal{N} \mid \omega_i(r-1) = a_1 \} \) and \( \mathcal{S}_{21}(r-1) = \{ i \in \mathcal{N} \mid \omega_i(r-1) = \{a_2, \ldots, a_K\} \}. \) In another word, \( \mathcal{S}_0(r-1), \mathcal{S}_1(r-1) \) and \( \mathcal{S}_{21}(r-1) \) denote the subsets of the agents adopting \( a_0, a_1 \) and \( a_2 \) in round \( r-1 \), respectively. According to Algorithm 1, we have

- For \( \forall i \in \mathcal{S}_1(r-1) \), it adopts the same arm (i.e., \( a_1 \)) in round \( r \) with probability \( \mu_i \) (or \( \mu_i(r) + \sum_{k=2}^{K} q_{i,k}(r)(1-p_k) \)), while adopting one of the others (i.e., \( a_2 \)) with probability \( \sum_{k=2}^{K} q_{i,k}(r)p_k \).
- For \( \forall i \in S_{21}(r-1) \), it adopts the best arm \( a_1 \) in round \( r \) with probability \( q_{i,1}(r)p_1 \).
- For \( \forall i \in S_0(r-1) \), the probability to adopt arm \( a_1 \) in round \( r \) is \( \mu_i + (1-\mu)q_{i,1}(r) \).

Therefore, the conditional expectations of \( Z_1^+ (r) \) and \( Z_1^- (r) \) can then be defined as

\[
\mathbb{E}[Z_1^+ (r) \mid X(r-1)] - \mathbb{E}[Z_1^- (r) \mid X(r-1)] = \sum_{i \in S_0(r-1)} \left( \frac{\mu_i}{K} + (1-\mu)q_{i,1}(r) \right) + \sum_{i \in S_{21}(r-1)} q_{i,1}(r)p_1
\]
and
\[
\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)] = \sum_{i \in S_i(r-1)} \sum_{k=2}^{K} q_{i,k}(r)p_k
\]
respectively. According to Lemma 1, when there are a sufficient number of agents (e.g., \(N \to \infty\)), we can re-write \(Z_1^+(r)\) and \(Z_1^-(r)\) as
\[
\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)]
= Z_0(r-1) \bigg( \frac{\mu}{K} + (1 - \mu)Q_1(r-1) \bigg) p_1
+ Z_{\geq 2}(r-1)Q_1(r-1)p_1
\]
(8)
and
\[
\mathbb{E}[Z_1^-(r) | \mathcal{X}(r-1)]
= Z_1(r-1) - \sum_{k=2}^{K} Q_k(r-1)p_k \tag{9}
\]
where \(Z_{\geq 2}(r-1) = \sum_{k=2}^{K} Z_k(r-1)\) denotes the number of the agents adopting arm \(a_{\geq 2} \in \{a_2, \ldots, a_K\}\) in round \(r-1\). Therefore, when \(Z_i < N\) we have
\[
\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)] + \mathbb{E}[Z_1^-(r) | \mathcal{X}(r-1)] \geq \frac{\xi_0}{\xi_1} \tag{10}
\]
where
\[
\begin{align*}
\xi_0 &= (N - \mu Z_0(r-1) - Z_1(r-1))p_1Q_1(r-1) \\
\xi_1 &= N - \mu Z_0(r-1) - Z_1(r-1) - p_1Q_1(r-1) + Z_1(r-1) - \sum_{k=2}^{K} Q_k(r-1)p_k
\end{align*}
\]
Further, considering \(p_2 \geq p_k\) for \(\forall k = 3, 4, \ldots, K\), \(N \geq 2\) and \(0 < \mu < 1\), we have
\[
Z_1(r-1) - \sum_{k=2}^{K} Q_k(r-1)p_k
\leq Z_1(r-1)p_2 - \sum_{k=2}^{K} Q_k(r-1)
= Q_1(r-1)p_2(N - Z_0(r-1) - Z_1(r-1))
\leq Q_1(r-1)p_2(N - \mu Z_0(r-1) - Z_1(r-1))
\]
By substituting the above inequality into (10), we have
\[
\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)] + \mathbb{E}[Z_1^-(r) | \mathcal{X}(r-1)]
\geq \frac{p_1}{p_1 + p_2} > \frac{1}{2}
\]
(11)
and thus \(\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)] + \mathbb{E}[Z_1^-(r) | \mathcal{X}(r-1)]\) which completes the proof. \(\square\)

**Remark 1.** The above lemma implies that, there exists \(\delta_0 > 0\) (which may be very small), such that \(\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)] \geq \delta_0\) holds when \(N\) is sufficiently large (e.g., \(N \geq N_0\) where \(N_0\) is a very large positive integer). Another implication from (11) is \(\mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)] \geq \frac{p_1}{p_1 + p_2}\). That is, if the best arm \(a_1\) is much better than the second best one, we could have much more agents adopting \(a_1\) in expectation in each round.

**Lemma 3.** Assume \(\Delta Z_1(r) = Z_1(r) - Z_1(r-1)\). In each round \(r\), with high probability (at least \(1 - e^{-2\delta_0 N}\)) conditioned on \(\mathcal{X}(r-1)\), we have
\[
\Delta Z_1(r) \geq 1 \tag{12}
\]
holds when \(N \geq N_0\). Especially, when \(N \to \infty\), the above inequality holds with probability \(\to 1\), i.e.,
\[
\lim_{N \to \infty} \mathbb{P}(\Delta Z_1(r) \geq 1 | \mathcal{X}(r-1)) = 1 \tag{13}
\]

**Proof.** Assume \(U_1(r) = \mathbb{E}[Z_1^+(r) | \mathcal{X}(r-1)]\). According to the Hoeffding’s inequality [37],
\[
\mathbb{P}(|Z_1(r) - U_1(r)| \leq \delta | \mathcal{X}(r-1)) \geq 1 - e^{-2\delta^2 N}
\]
holds for \(\forall \delta > 0\). Hence, given \(\forall \delta > 0\), with probability at least \(1 - e^{-2\delta^2 N}\) conditioned on \(\mathcal{X}(r-1)\), we have
\[
\Delta Z_1(r) = Z_1(r) - Z_1(r-1)
\geq U_1(r) - Z_1(r-1) - \delta
\]
Furthermore, since \(U_1(r) = Z_1(r-1) + \mathbb{E}[Z_1^+(r) - Z_1^-(r) | \mathcal{X}(r-1)]\), the above inequality can be re-written as
\[
\Delta Z_1(r) \geq \mathbb{E}[Z_1^+(r) - Z_1^-(r) | \mathcal{X}(r-1)] - \delta \tag{14}
\]
Hence, by considering Remark 1, \(\Delta Z_1(r) > 0\) with probability at least \(1 - e^{-2\delta^2 N}\) when \(N \geq N_0\). Furthermore, since both \(Z_1^+(r)\) and \(Z_1^-(r)\) are positive integers, \(\Delta Z_1(r) > 0\) implies \(\Delta Z_1(r) \geq 1\).

**Theorem 1.** Let \(\text{Succ}_z(N) = \{\text{Succ}(N) | Z_1(0) = z\}\) represent that event \(\text{Succ}(N)\) happens with condition \(Z_1(0) = z\). Suppose \(Z_1(0) = z \geq 1\) initially. When \(N \geq \max \{\frac{\ln 2}{\delta^2}, N_0\}\),
\[
\mathbb{P}(\text{Succ}_z(N)) \geq 1 - \left(\frac{1}{e^{2\delta^2 N} - 1}\right)^z \tag{15}
\]
Especially, as \(N \to \infty\), we have \(\text{Succ}_z(N) \to 1\).

**Proof.** It is stated in Lemma 3 that, when there are a sufficient number of agents, with high probability, we have the number of agent adopting the best arm \(a_1\) in each round increased by at least 1. We hereby consider the learnability of our algorithm in the “worst” case where \(\Delta Z_1(r)\) is increased by at most 1 (i.e., by 1 or 0) in each round with probability \(q \geq 1 - e^{-2\delta^2 N}\) where \(N \geq N_0\) is sufficiently large. If we had inequality (15) hold in the above “worst” case, it would hold when \(\Delta Z_1(r)\) is increased by more than 1. In the following, we use a recursive approach to prove the theorem. First, with the initialization condition \(Z_1(0) = z\), we have
\[
P_z = \mathbb{P}(\text{Succ}(N) | Z_1(0) = z)
= \mathbb{P}(\text{Succ}(N), \Delta Z_1(1) = 1 | Z_1(0) = z)
+ \mathbb{P}(\text{Succ}(N), \Delta Z_1(1) \leq -1 | Z_1(0) = z)
= \mathbb{P}(\Delta Z_1(1) = 1 | Z_1(0) = z) \cdot \mathbb{P}(\text{Succ}(N) | \Delta Z_1(1) = 1, Z_1(0) = z)
+ \mathbb{P}(\Delta Z_1(1) \leq -1 | Z_1(0) = z) \cdot \mathbb{P}(\text{Succ}(N) | \Delta Z_1(1) \leq -1, Z_1(0) = z)
= q^z \mathbb{P}(\text{Succ}(N) | \Delta Z_1(1) = 1, Z_1(0) = z)
+ (1 - q) \mathbb{P}(\text{Succ}(N) | \Delta Z_1(1) \leq -1, Z_1(0) = z)
= q^z \mathbb{P}(\text{Succ}(N) | Z_1(0) = z + 1)
+ (1 - q) \mathbb{P}(\text{Succ}(N) | Z_1(0) \leq z - 1)
= q^{z+1} + (1 - q)P_{\leq z-1} \tag{16}
\]
Furthermore, since \( qP_\zeta + (1-q)P_\zeta = qP_{\zeta + 1} + (1-q)P_{\zeta - 1} \), we have
\[
P_{\zeta + 1} - P_\zeta = \frac{1-q}{q} (P_\zeta - P_{\zeta - 1})
\geq \frac{1-q}{q} (P_\zeta - P_{\zeta - 1})
\]
By recurrence,
\[
P_{\zeta + 1} - P_\zeta \geq \left( \frac{1-q}{q} \right)^\zeta (P_1 - P_0)
\]
Due to Proposition 1, the agents who have already adopted the non-null arms will not adopt the null arm thereafter. Hence, we have \( P_0 = 0 \) and the inequality (18) can be rewritten as
\[
P_{\zeta + 1} - P_\zeta \geq \left( \frac{1-q}{q} \right)^\zeta P_1
\]
and thus
\[
P_{\zeta + 1} - P_1 \geq \sum_{z=1}^{\zeta} (P_{z+1} - P_z) = \sum_{z=1}^{\zeta} \left( \frac{1-q}{q} \right)^z P_1
\]
Therefore,
\[
P_{\zeta + 1} \geq \sum_{z=0}^{\zeta} \left( \frac{1-q}{q} \right)^z P_1 = 1 - \frac{\left( \frac{1-q}{q} \right)^{\zeta + 1}}{1 - \frac{1-q}{q}} P_1
\]
Note that the above inequality holds for \( \forall \zeta \in (0, 1] \). Especially, when \( \zeta = \frac{N-1}{N} \), \( P_N \geq \frac{1 - \left( \frac{1-q}{q} \right)^N}{1 - \frac{1-q}{q}} P_1 \). Since \( P_N = 1 \),
\[
P_1 \leq 1 - \frac{\left( \frac{1-q}{q} \right)^N}{1 - \frac{1-q}{q}}
\]
in which the equality holds when \( \Delta Z_1(t) \geq -1 \). Hence, \( P_{\zeta+1} \) should satisfy
\[
P_{\zeta + 1} \geq 1 - \frac{\left( \frac{1-q}{q} \right)^{\zeta + 1}}{1 - \frac{1-q}{q}}
\]
When \( N > \max \left\{ \frac{\ln 2}{2\delta_0}, \delta_0 \right\} \) such that \( q \geq \frac{1}{2} \), in a more general form,
\[
P_\zeta \geq \frac{1 - \left( \frac{1-q}{q} \right)^\zeta}{1 - \left( \frac{1-q}{q} \right)^N} \geq 1 - \frac{1}{e^{2\delta N} - 1}
\]
which completes the proof. \( \square \)

**Remark 2.** Since \( \mathbb{P}(\text{Succ}_{\zeta+1}(N)) \geq \mathbb{P}(\text{Succ}_{\zeta}(N)) \), Theorem 1 also implies that \( \text{Succ}_{\zeta}(N) = \{ \text{Succ}(N) \mid Z_1(0) \geq \zeta \} \) happens with probability at least \( 1 - \frac{1}{e^{2\delta N} - 1} \).

As shown in Theorem 1 (as well as Remarks 2), the success event \( \text{Succ}(N) \) happens with an exponentially increasing probability under an initial condition \( Z_1(0) \geq \zeta \), whereas our algorithm begins with a more unified initial condition that all agents having no preference such that \( Z_1(0) = 0 \) as demonstrated in Sec. 4. In the following Lemma 4, we extend Theorem 1 to the initial condition \( X_{i,0}(0) = 1 \) for \( \forall i \).

**Lemma 4.** In any round \( r \) such that \( X_{i,1}(r) = 0 \) for \( \forall i \in \mathcal{N} \), we have
\[
\mathbb{P} \left( Z_1(r) \geq (1 - \delta_1) \frac{\mu_{p1}}{K} N \right) \geq 1 - e^{-\frac{N\delta_2 k^2}{2K}}
\]
where \( \delta_1 \) is a constant such that \( 0 < \delta_1 < 1 \).

**Proof.** According to Algorithm 1, for each agent \( i \) in round \( r \), it adopts the best arm \( a_1 \) through either uniform sampling or learning from its peers. Therefore, the probability for agent \( i \) to adopt \( a_1 \) (where \( k = 1, \cdots, K \)) in each round is at least \( \frac{K}{2K} \). We complete the proof by directly applying Chernoff bound [37]. \( \square \)

### 5.2 Complexity

As illustrate in Algorithm 1, the agents make decisions in each round by very light-weight computations, while the complexity mainly stems from the MHRW-based information dissemination. As shown in Lemma 3 that the number of agents adopting the best arm is increased by 1 with high probability, the rate at which our algorithm achieves convergence is \( \Omega(1) \), while the convergence can be further accelerated by the gap between \( p_1 \) and \( p_2 \) according to Remark 1. Therefore, in the following, we concentrate on revealing the communication complexity in each round. That is, how long each round should be (or how many slots each round should consist of) such that all these random walks achieve a nearly uniform stationary distribution under the CONGEST communication model.

**Theorem 2.** In Algorithm 1 where each agent launches \( h \log N \) MHRWs in parallel in each round, all the random walks achieve nearly uniform distribution \( \left[ \frac{1}{N} - \frac{1}{N} \delta, \frac{1}{N} + \frac{1}{N} \delta \right] \) in \( O(\log^2 N) \) slots with high probability.

**Proof.** According to Algorithm 1, for each agent \( i \), the expected number of tokens it receives from its neighbors in every slot is
\[
\sum_{\ell \in \mathcal{N}_i} \min \left\{ \frac{1}{d_{i,\ell}}, \frac{1}{d_{\ell,i}} \right\} \times h \log N
\]
\[
= \sum_{\ell \in \mathcal{N}_i; \frac{1}{d_{i,\ell}} \geq \frac{1}{\delta}} h \log N \frac{d_{i,\ell}}{d_{\ell,i}} + \sum_{\ell \in \mathcal{N}_i; \frac{1}{d_{i,\ell}} \leq \frac{1}{\delta}} h \log N \frac{d_{\ell,i}}{d_{i,\ell}}
\]
\[
\leq \sum_{\ell \in \mathcal{N}_i; \frac{1}{d_{i,\ell}} \geq \frac{1}{\delta}} h \log N \frac{d_{i,\ell}}{d_{i,\ell}} + \sum_{\ell \in \mathcal{N}_i; \frac{1}{d_{i,\ell}} \leq \frac{1}{\delta}} h \log N \frac{d_{i,\ell}}{d_{i,\ell}}
\]
\[
= d_i \times h \log N \frac{d_i}{d_i} = h \log N
\]
According to Hoeffding’s inequality [37], agent \( i \) receives at most \( 2h \log N \) tokens in each slot with probability at least \( 1 - \frac{1}{N^{0.7}} \). Furthermore, considering the agents employ FIFO policy to forward the tokens, the tokens received by agent \( i \) in slot \( t \) will be delayed for at most \( t \) additional slots. Hence, \( O(\log^2 N) \) slots are sufficient for all the tokens to be forwarded \( O(\log N) \) times, which completes the proof. \( \square \)

### 5.3 Reliability

We now look at the reliability of our proposed collaborative learning algorithm in face of agent corruptions. It is worthy to note that the learnability should be re-explained in this case.
There are always a set of corrupted agents disseminating randomly falsified messages in each round, while these falsified messages may “deceive” some of the agents to adopt non-optimal arms. Since an adversary may corrupt different subsets of agents across rounds as mentioned in Sec. 3, we suppose these “temporarily” corrupted agent still perform normally in both sampling and adopting stages according to Algorithm 1. Therefore, we count in the corrupted agents adopting $a_1$ when analyzing the dynamics of $Z_1(r)$. As shown in Theorem 3, when the proportion of the corrupted agents can be bounded, we still have $E[Z_1(r) - Z_1(r - 1) | \mathcal{X}(r - 1)] > 0$.

**Theorem 3.** For any round $r$ such that $Q_1(r - 1) \leq \alpha$, we have

$$\frac{E[Z_1^+(r) | \mathcal{X}(r - 1)]}{E[Z_1^-(r) | \mathcal{X}(r - 1)]} \geq 1$$

when $\tau \leq \frac{(1 - \alpha)(p_1 - p_2)}{(1 - \alpha)p_1 + \alpha p_2}$.

**Proof.** In each round $r$, each agent $i$ with $X_{i,0}(r - 1) = 1$ has a non-null adoption with probability at least $\frac{K}{r} \sum_{k=1}^K p_k$. To benefit our proof, we consider $\tau$ is sufficiently large such that all of the agents already have non-null adoptions in round $r - 1$, which can be ensured within at most $\frac{K}{r} \sum_{k=1}^K p_k$ rounds on expectation.

The adversary chooses up to $\tau N$ agents to corrupt uniformly at random in each round $r$; therefore, the expected number of corrupted agents which adopt arm $a_k$ is $\tau Z_k(r - 1)$. Let $\tilde{X}_{i,k}$ denote the falsified adoptions to be disseminated in round $r$. We also denote by $\tilde{Z}_k(r)$ and $\tilde{Q}_k(r)$ the number of agents adopting $a_k$ (including both the normal agents and the corrupted ones) and the proportion of these agents in round $r$. Since the adversary can arbitrarily change the adoptions of the corrupted agents, we have

$$\tilde{Q}_k(r) = Q_1(r - 1) = \frac{(1 - \tau)Q_1(r - 1) + \tau \sum_{k=2}^K Q_k(r - 1)}{K}$$

(25)

According to Lemma 1, each agent $i$ chooses arm $a_k$ with probability $\tilde{Q}_k(r)$ in the sampling stage of round $r$. The expectations of $Z_1^+(r)$ and $Z_1^-(r)$ can be re-written as

$$E[Z_1^+(r) | \mathcal{X}(r - 1)] = E[(N - Z_1(r - 1))\tilde{Q}_1(r) | \mathcal{X}(r - 1)] \geq p_1(N - Z_1(r - 1))(1 - \tau)Q_1(r - 1)$$

and

$$E[Z_1^-(r) | \mathcal{X}(r - 1)] = E[Z_1(r - 1) \sum_{k=2}^K \tilde{Q}_k(r)p_k | \mathcal{X}(r - 1)] \leq p_2(1 - E[\tilde{Q}_1(r) | \mathcal{X}(r - 1)])Z_1(r - 1) \leq p_2(1 - (1 - \tau)Q_1(r - 1))Z_1(r - 1) = p_2(N - (1 - \tau)Z_1(r - 1)Q_1(r - 1) = p_2(N - Z_1(r - 1) + \tau Z_1(r - 1))Q_1(r - 1)$$

Therefore,

$$E[Z_1^+(r) | \mathcal{X}(r - 1)] \geq p_1(N - Z_1(r - 1))(1 - \tau)$$

$$E[Z_1^-(r) | \mathcal{X}(r - 1)] \geq \frac{p_1}{p_2} \frac{(1 - \alpha)(1 - \tau)}{1 - \alpha + \alpha \tau}$$

(26)

Since $Q_1(r - 1) \leq \alpha$,

$$E[Z_1^+(r) | \mathcal{X}(r - 1)] \geq p_1 \frac{(1 - \alpha)(1 - \tau)}{1 - \alpha + \alpha \tau}$$

(27)

We finally have $E[Z_1^+(r) | \mathcal{X}(r - 1)] \geq E[Z_1^-(r) | \mathcal{X}(r - 1)]$ when $\tau \leq \frac{(1 - \alpha)(p_1 - p_2)}{(1 - \alpha)p_1 + \alpha p_2}$.

**Remark 3.** Theorem 3 implies our algorithm can tolerate more agent corruptions when there is a larger gap between $p_1$ and $p_2$. Furthermore, the fault tolerance of our algorithm also depends on the popularity of the best arm. Specifically, as our algorithm proceeds, more agents learn the best arm, while corrupting a small fraction of the agents can interrupt the increment of the number of agents adopting the best arm. Our algorithm converges to a “stationary” state $\mathcal{X}(r)$, if

$$Q_1(r) = \frac{E[\tilde{Q}_1(r + 1) | \mathcal{X}(r)]p_1}{\sum_{k=1}^K E[\tilde{Q}_k(r + 1) | \mathcal{X}(r)]p_k}$$

(28)

The above equation give a instantaneous characterization on the popularity of the best arm when our collaborative learning algorithm achieves a stationary distribution in some round. One question is, is there any upper-bound on $E[Q_1(r)]$? We answer this question in the following corollary.

**Corollary 1.** For $\forall r$, when there are $\tau N$ corrupted agents,

$$E[Q_1(r)] < 1 - \frac{\tau}{K} \sum_{k=2}^K p_k$$

(29)

**Proof.** Imagine that all the agents adopted the best arm $a_1$ such that $Z_1(r - 1) = N^\frac{3}{4}$. Since the expected number of $a_k$-tokens (for $\forall k \geq 2$) disseminated by the corrupted agents is $\frac{\tau Nk \log N}{K}$, there are $\frac{\tau N \sum_{k=2}^K p_k}{K}$ agents misled to adopt $a_{\geq 2} \in \{a_2, \ldots, a_K\}$ on expectation and we thus have (29) holds for any round $r$.

### 6 Numerical Results

In this section, we perform extensive simulations on both synthetic and real datasets to verify the efficacy of our algorithm. Throughout this section, we adopt $h = 10$ and $\mu = 0.3$ for the disseminating stage and sampling stage of our collaborative learning algorithm, respectively.

5. Note that it is an ideal case which will never happen according to what we have mentioned at the beginning of Sec. 5.3, but this ideal case is indeed helpful for us to derive the upper-bound on $Q_1$. 

We first investigate the performance of our algorithm on synthetic data. We gradually increase the number of agents (i.e., $N$) from $2 \times 10^3$ to $6 \times 10^3$ with a step size $2 \times 10^3$ and connect the agents randomly. We also evaluate our algorithm by letting the number of arms $K = 100, 200, 300$. As shown in Remark 1, the learning process can be accelerated by higher difference between $p_1$ and $p_2$; therefore, we fixed $p_1 = 0.8$ and vary $p_2 = 0.6, 0.5, 0.4$. The numerical results are reported in Fig. 1, where we use the popularity of the best arm $a_1$ (i.e., $Q_1(r)$) to illustrate the learning dynamics of our algorithm. It is demonstrated that, all of the agents eventually learn the best arm (i.e., $Q_1(r) = 1$), according to Theorem 1. Although increasing the number of agents results in a slight increasement in terms of the number of rounds, $\text{Succ}(N)$ can be achieved in about 40 rounds in all of our settings. Note that the number of rounds actually is not meant to the actual temporal complexity. More time is necessitated to achieve $\text{Succ}(N)$ when there are more agents, recalling each round consists of $O(\log^2 N)$ slots for the agents to disseminate their adoptions as shown in Theorem 2. Nevertheless, the temporal cost is an inevitable investment to ensure the learnability in large-scale multi-agent graphs. Furthermore, the number of the agents adopting $a_1$ approaches $N$ with a higher rate when there is a larger gap between $p_1$ and $p_2$, which is consistent with what has been mentioned in Remark 1. Additionally, as implied by Theorem 1, the number of arms, i.e., $K$, actually has a very slight impact on the number of rounds for our algorithm to approaches the convergence state.

We then evaluate the reliability of our algorithm under different settings. We let $N = 1 \times 10^3$ and the generate the edges among the agents randomly and adopt the same setting on $N$ and $K$ as what we did in the last experiments. We vary $\alpha$ from 0.9 to 0.6 with a step size 0.1 and let $\tau$ take its upper-bound, i.e., $\tau = \frac{(1-\alpha)(p_1-p_2)}{(1-\alpha)p_1+a_2}$. The results are reported in Fig. 2. It is shown that a smaller $\alpha$ implies a lower upper-bound of the popularity of $a_1$, since we have
We hereby evaluate our collaborative learning algorithm with Movielens 25M dataset which involves 162,000 users and 62,000 movies [38]. We select a subset of 3,443 users and a subset of 707 movies, such that each of the selected users rated at least 30 of these movies and each of the movies was rated by at least 30 of these users. We extract out the corresponding submatrix and apply the matrix completion method [39] to fill the missing entries in the extracted submatrix. We then calculate the average of each column and normalize the average to $[0, 1]$ by dividing the average by 5. We consider each movie as an arm whose quality can be represented by the normalized score. Likewise, to illustrate the influence of $p_1/p_2$ on the learning process, we fix $p_1 = 0.9$ (which is the maximum score) and randomly take 500 samples from the remaining scores such that $p_2 \in \{0.7, 0.6, 0.5\}$. We also construct a communication graph where the edges among the users are generated randomly. The results shown in Fig. 3 is very similar to the ones we obtained with synthetic dataset. In particular, $\text{Succ}(N)$ can be achieved within 40 rounds even when the gap between $p_1$ and $p_2$ is small (e.g., $p_1 - p_2 = 0.2$), while the temporal overhead can be further reduced with smaller gaps (e.g., around 25 rounds are sufficient when $p_1 - p_2 = 0.4$).

6.2 Simulations with Real Data

We hereby evaluate our collaborative learning algorithm with Movielens 25M dataset which involves 162,000 users and 62,000 movies [38]. We select a subset of 3,443 users and a subset of 707 movies, such that each of the selected users rated at least 30 of these movies and each of the movies was rated by at least 30 of these users. We extract out the corresponding submatrix and apply the matrix completion method [39] to fill the missing entries in the extracted submatrix. We then calculate the average of each column and normalize the average to $[0, 1]$ by dividing the average by 5. We consider each movie as an arm whose quality can be represented by the normalized score. Likewise, to illustrate the influence of $p_1/p_2$ on the learning process, we fix $p_1 = 0.9$ (which is the maximum score) and randomly take 500 samples from the remaining scores such that $p_2 \in \{0.7, 0.6, 0.5\}$. We also construct a communication graph where the edges among the users are generated randomly. The results shown in Fig. 3 is very similar to the ones we obtained with synthetic dataset. In particular, $\text{Succ}(N)$ can be achieved within 40 rounds even when the gap between $p_1$ and $p_2$ is small (e.g., $p_1 - p_2 = 0.2$), while the temporal overhead can be further reduced with smaller gaps (e.g., around 25 rounds are sufficient when $p_1 - p_2 = 0.4$).
rate when the second best arm is of much lower quality than the first one. By these observations, our theoretical analysis is further confirmed.

7 CONCLUSION

In this paper, we have proposed a three-staged collaborative learning algorithm for general multi-agent graphs with constraints on communication bandwidth and memorization. In each round of our algorithm, each agent first disseminates its current adoption (or preference) over the graph through random walks and then choose one arm to pull according to the suggestions received from its peers. It finally makes an adoption decision according to the observation on the reward yielded by the pulling. According to our theoretical analysis, although each agent only learns limited knowledge from its peers due to the constraints on communications and memorizations, the learnability of our algorithm can be ensured such that all the agents eventually adopt the best arm with high probability. We also have quantified the reliability of our collaborative learning algorithm in face of agent corruptions. We finally have conducted extensive experiments on both synthetic and real datasets to verify the efficacy of our algorithm.

REFERENCES

[1] W. Shen, J. Wang, Y. Jiang, and H. Zha. Portfolio Choices with Orthogonal Bandit Learning. In Proc. of the 24th IJCAI, pages 974–980, 2015.

[2] G. Pini, A. Brutschy, G. Francesca, M. Dorigo, and M. Birattari. Multi-armed Bandit Formulation of the Task Partitioning Problem in Swarm Robotics. In Proc. of International Conference on Swarm Intelligence, pages 109–120, 2012.

[3] T. Seeley and S. Buhrman. Group Decision Making in Swarms of Honey Bees. Behavioral Ecology and Sociobiology, 45(1):19–31, 1999.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-Time Analysis of The Multiarmed Bandit Problem. Machine Learning, 47(2):235–256, 2002.

[5] S. Bubeck and N. Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. Foundations and Trends® in Machine Learning, 5(1):1–122, 2012.

[6] N. Nayıyar, D. Kalathil, and R. Jain. On Regret-Optimal Learning in Decentralized Multiplayer Multiarmed Bandits. IEEE Trans. on Control of Network Systems, 5(1):597–606, 2018.

[7] P. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo. Optimal Algorithms for Multiplayer Multi-armed Bandits. In Proc. of the 23rd AISTATS, pages 4120–4129, 2020.

[8] S. Bubeck, Y. Li, Y. Peres, and M. Sellke. Non-Stochastic Multi-Player Multi-Armed Bandits: Optimal Rate With Collision Information, Sublinear Without. In Proc. of the 33rd Conference on Learning Theory, volume 125, pages 961–987, 2020.

[9] G. Bianconi and A. Barabási. Bose-Einstein Condensation in Complex Networks. Physical Review Letters, 86(24):5632, 2001.

[10] S. Pratt, D. Sumpter, E. Mallon, and N. Franks. An Agent-Based Model of Collective Nest Choice by The Ant Temnothorax Albipennis. Animal Behaviour, 70(5):1023–1036, 2005.

[11] B. Beheim, C. Thigpen, and R. McElreath. Strategic Social Learning and The Population Dynamics of Human Behavior: The Game of Go. Evolution and Human Behavior, 35(5):351–357, 2014.

[12] B. Granovskiy, J. Gold, D. Sumpter, and R. Goldstone. Integration of Social Information by Human Groups. Topics in Cognitive Science, 7(3):469–493, 2015.

[13] L. Celis, P. Krafft, and N. Vishnoi. A Distributed Learning Dynamics in Social Groups. In Proc. of the 36th ACM PODC, page 441–450, 2017.

[14] L. Su, M. Zubeldia, and N. Lynch. Collaboratively Learning the
Best Option on Graphs, Using Bounded Local Memory. *Proc. of the ACM on Measurement and Analysis of Computing Systems*, 3(1), 2019.

[15] T. Cover and M. Hellman. The Two-Armed-Bandit Problem with Time-Invariant Finite Memory. *IEEE Trans. on Information Theory*, 15(5):185–195, 1969.

[16] A. Slivkins. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

[17] T. Lai and H. Robbins. Asymptotically Efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[18] J. Audibert, S. Bubeck, and R. Munos. Best Arm Identification in Multi-Armed Bandits. In *Proc. of the 23rd COLT*, pages 41–53, 2010.

[19] E. Kaufmann, O. Cappé, and A. Garivier. On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.

[20] C. Tao, Q. Zhang, and Y. Zhou. Collaborative Learning with Limited Interaction: Tight Bounds for Distributed Exploration in Multi-armed Bandits. In *Proc. of the 60th IEEE FOCS*, pages 126–146, 2019.

[21] J. Vermorel and M. Mohri. Multi-armed Bandit Algorithms and Empirical Evaluation. In *Proc. of the 16th ECML*, pages 437–448, 2005.

[22] M. Chakraborty, K. Chua, S. Das, and B. Juba. Coordinated Versus Decentralized Exploration In Multi-Agent Multi-Armed Bandits. *Proc. of the 26th IJCAI*, pages 164–170, 2017.

[23] B. Szorenyi, R. Busa-Fekete, I. Hegedus, R. Ormandi, M. Jelasity, and B. Kegl. Gossip-based Distributed Stochastic Bandit Algorithms. In *Proc. of the 30th ICML*, pages 19–27, 2013.

[24] R. Kolla, K. Jagannathan, and A. Gopalan. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Trans. on Networking*, 26(4):1782–1795, 2018.

[25] A. Sankararaman, A. Ganesh, and S. Shakkottai. Social Learning in Multi Agent Multi Armed Bandits. *Proc. of the ACM on Measurement and Analysis of Computing Systems*, 3(3):53:1–53:35, 2019.

[26] T. Cover and M. Hellman. A Note on The Two-Armed Bandit Problem with Finite Memory. *Information and Control*, 12(5):371–377, 1968.

[27] T. Lykouris, Y. Mirrokni, and R. Leme. Stochastic Bandits Robust to Adversarial Corruptions. In *Proc. of the 50th ACM STOC*, pages 114–122, 2018.

[28] S. Kapoor, K. Patel, and P. Kar. Corruption-Tolerant Bandit Learning. *Machine Learning*, 108:687–715, 2019.

[29] A. Gupta, T. Koren, and K. Talwar. Better Algorithms for Stochastic Bandits with Adversarial Corruptions. In *Proc. of the 32rd COLT*, pages 1562–1578, 2019.

[30] D. Vial, S. Shakkottai, and R. Srikant. Robust Multi-Agent Multi-Armed Bandits. In *Proc. of the 22nd ACM MobiHoc*, pages 161–170, 2021.

[31] D. Feleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[32] T. Gonen and R. Oshman. Lower Bounds for Subgraph Detection in The CONGEST Model. In *Proc. of the 21st OPODIS*, pages 41–53, 2017.

[33] K. Censor-Hillel, E. Fischer, G. Schwartzman, and Y. Vassou. Fast Distributed aAlgorithms for Testing Graph Properties. *Distributed Computing*, 32(1):41–57, 2019.

[34] U. Agarwal and V. Ramachandran. Faster Deterministic All Pairs Shortest Paths in Congest Model. In *Proc. of the 32nd ACM SPAA*, page 11–21, 2020.

[35] M. Halldórsson, F. Kuhn, Y. Maus, and T. Tonoyan. Efficient Randomized Distributed Coloring in CONGEST. In *Proc. of the 53rd ACM STOC*, pages 1180–1193, 2021.

[36] D. Levin and Y. Peres. *Markov Chains and Mixing Times*, volume 107. American Mathematical Soc., 2017.

[37] D. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[38] *MovieLens 25M Dataset*. https://grouplens.org/datasets/movielens/25m/. [Online; accessed 02-November-2021].

[39] T. Hastie, R. Mazumder, J. Lee, and R. Zadeh. Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.
Weifeng Lv received his Ph.D. degree in computer science from Beihang University in 1998. His current research interests include massive information system, urban cognitive computing, swarm intelligence, and smart cities. He is a professor of computer science and the dean of the School of Computer Science and Engineering, as well as the vice director of the State Key Laboratory of Software Development Environment, at Beihang University. He also serves as the Vice-President and Secretary-General of the China Software Industry Association and the director of National Engineering Research Center for Science and Technology Resources Sharing Service. He received multiple internationally renowned awards, including the second prize of the 2016 China National Science and Technology Invention Award and the first prize of the 2010 Beijing Science and Technology Award.

Xiuzhen Cheng received her M.S. and Ph.D. degrees in computer science from the University of Minnesota – Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber-physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is Fellow of IEEE and a member of ACM.