Multi-Agent Stochastic Bandits Robust to Adversarial Corruptions

Fatemeh Ghaffari[†] FGHAFFARI@CS.UMASS.EDU
Xuchuang Wang[†] XUCHUANGWANG@CS.UMASS.EDU
Jinhang Zuo[‡] JINHANGZUO@GMAIL.COM
Mohammad Hajiesmaili[†] HAJIESMAILI@CS.UMASS.EDU
[†]*CICS, UMass Amherst, Amherst, MA* [‡]*CS, CityU, Kowloon, Hong Kong*

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

We study cooperative multi-agent multi-armed bandits with adversarial corruption in a heterogeneous setting, where each agent has access to a subset of the full arm set, and the adversary can corrupt the reward observations for all agents. The objective is to maximize the cumulative total reward of all agents (and not be misled by the adversary). We propose a multi-agent cooperative learning algorithm that is robust to adversarial corruption. For this newly devised algorithm, we demonstrate that an adversary with an unknown corruption budget C only incurs an additive $O((L/L_{\min})C)$ term to the standard regret of the model in non-corruption settings, where L is the total number of agents and $L_{\min}$ is the minimum number of agents with mutual access to an arm. As independent side contributions, our algorithm improves the state-of-the-art regret bounds when reducing to both the single-agent and homogeneous multi-agent scenarios, tightening multiplicative K (the number of arms) and L (the number of agents) factors, respectively. Lastly, we conduct numerical simulations to corroborate the superiority of our proposed algorithm.

Keywords: multi-agent multi-armed bandits, corruption, robust

1. Introduction

Motivated by its broad applicability in large-scale learning systems, the multi-agent multi-armed bandits (`MA2B`) problem has recently been studied extensively in various settings [19; 10; 7; 26; 11; 21; 14; 11; 23; 16]. In this model, each agent can access a number of arms, solving its own multi-armed bandit (`MAB`) problem. The goal for each agent is to maximize the cumulative reward, or in other words, to minimize the aggregate regret, which is the difference between the cumulative rewards of the optimal arm and the agent's choices. Previous works typically assume a homogeneous setting, where all agents have access to the entire set of arms and solve the same `MAB` instance.

In many real-world applications, these agents can access different but possibly overlapping sets of arms and face adversarial corruption. Examples include recommender systems [29; 8; 17], online advertising [3], and shortest path routing [31; 18]. In recommender systems, products (arms) have different ratings (rewards), and platform operators (agents) are associated with distinct geographical jurisdictions where the set of available products, as well as the rules governing their sale, may differ. This results in each agent having access to a different set of arms, possibly overlapping with other agents' arm sets. Each agent aims to maximize user satisfaction by recommending products within its jurisdiction. An agent "pulls" an arm by recommending a product to a user and then observes the reward as the user's rating. However, fake ratings and reviews (corruption) may be introduced by merchants to manipulate their product's ratings or degrade competitors' products. Agents are unaware of the presence or level of this corruption. While sharing information is beneficial, privacy requirements restrict agents from sharing individual customer data. Instead, they

may share aggregated purchase information with limited communication frequency. The objective, therefore, is to develop an algorithm that minimizes the impact of adversarial corruption with a low communication cost.

Overview of technique challenges. We propose a cooperative algorithm for the heterogeneous MA2B robust to adversarial corruptions. There is limited prior research on multi-agent online learning under corruption, and mostly devoted to homogeneous settings [20; 15; 13]. The heterogeneous multi-agent scenario [27; 24] which is of more practical interest is not explored. Especially as robustness requires each agent to incorporate randomness in their decisions to prevent adversaries from exploiting predictable patterns. This randomness results in uncoordinated actions, contrasting with standard cooperative bandit algorithms [23; 21], where agents typically coordinate arm pulls. Additionally, agents are constrained to pull only their local arms, and the heterogeneity of these arms—with varying pull probabilities—further complicates cooperation.

1.1. Contributions

In this paper, we formally model the practical problem of heterogeneous M2AB with adversarial corruptions, where $L \in \mathbb{N}^+$ heterogeneous agents cooperatively play a stochastic MAB game with $K \in \mathbb{N}^+$ arms under corruptions. Each arm k is associated with a reward distribution. Each agent ℓ can access a subset of these K arms, called the *local arm set*. In each decision round, each agent picks one arm from its local arm set to pull and observes a reward sample drawn from the pulled arm, which an adversary possibly corrupts. Similar to the regular MAB setting, each agent aims to maximize the reward over a horizon of $T \in \mathbb{N}^+$ rounds. We assume that all agents can communicate with each other in a fully connected setting. The formal model definition is provided in Section 2.

Algorithm design (Section 3) We introduce the Distributed Robust Arm Activation (DRAA) algorithm for heterogeneous MA2B that is agnostic to the corruption level. This is the first work to address stochastic MA2B with adversarial corruption in a heterogeneous setting. We introduce new algorithmic techniques to address the challenges posed by heterogeneity and corruption. The algorithm incorporates a set-splitting technique to classify arms as active or bad, allocating limited resources to bad arms instead of eliminating them. We also apply a weighted averaging method to estimate the empirical means for arms to interpret information received from other agents properly. By distributing arm exploration among agents and adjusting epoch lengths based on the number of agents accessing arms, we leverage shared arm access to achieve the optimal theoretical upper bound.

Theoretical analyses (Section 4) We show that DRAA achieves a regret of $O((L/L_{\min})C) + O\left(\log T \cdot \log(\log T) \cdot K/\Delta_{\min}\right)$, where $L_{\min}$ is the minimum number of agents with mutual access to the same arm, $\Delta_{\min}$ is the minimum non-zero gap between the expected reward of any arm k and any local best arm, and C is the total corruption level. We also reduce the algorithm to single-agent MAB, achieving a regret bound of $O(C) + O\left(\log T \cdot \log(\log T) \cdot K/\Delta_{\min}\right)$. This improves upon BARBAR [6] by eliminating the K factor from the corruption term. Similarly, reducing our setting to a homogeneous MA2B yields a $O(C) + O\left(\log T \cdot \log(\log T) \cdot K/\Delta_{\min}\right)$ regret, removing a prefactor of L from the corruption term compared to [10]. Besides, DRAA achieves an optimal regret bound up to a $\log(\log T)$ factor without corruption , while maintaining a logarithmic communication cost.

Numerical Experiments (Section 5) We perform experiments to evaluate the performance of DRAA against established baseline algorithms. The results demonstrate the superiority of DRAA in enhancing robustness against different levels of adversarial corruption. In a heterogeneous setting with $K = 100$ and $L = 10$ under full adversarial corruption, where every realized reward is flipped from 0 to 1 and vice versa, DRAA outperforms BARBAR by 15.26% and CO-UCB [27] by 24.92%.

1.2. Related Work

MA2B Among the vast amount of MA2B literature [2; 7; 8; 11; 10; 14; 16; 19; 20; 21; 23], the work of [27]—which first introduces the heterogeneous MA2B model—is the most related to ours. Yang et al. [27] extend the Upper Confidence Bound (UCB) and the Active Arm Elimination (AAE) [4] algorithms to a heterogeneous setting. Their Cooperative UCB (CO-UCB) algorithm achieves a regret of $O(\sum_{k:\tilde{\Delta}_k<0} \log T/\tilde{\Delta}_k)$, where $\tilde{\Delta}_k$ is defined as the minimum gap between the mean reward of arm k and the local optimal arm of any agent with access to k. In the heterogeneous MA2B problem [27; 2], each agent ℓ has access to a subset of arms. Similar to the regular MAB setting, each agent aims to maximize the reward within its own local arm set. We assume that all agents can communicate with each other in a fully connected setting to share information and expedite the learning process. The formal definition of the problem setting is provided in Section 2.

MAB with corruption The notion of corruption was first introduced in [12]. Their proposed algorithm, Multi-Layer Active Arm Elimination Race, exhibits a regret of $\tilde{O}(\sum_{i\neq i^*} C/\Delta_k)$, where C is the corruption level and Δ_k represents the gap between the mean reward of arm k and the optimal arm. This bound is then improved by [6]: Bad Arms Get Resource (BARBAR), which achieves a regret of $O(KC) + \tilde{O}(\sum_{k\neq k^*} 1/\Delta_k)$, introducing an additive term accounting for corruption. BARBAR improves upon the Active Arm Elimination (AAE) algorithm [4] where instead of permanently eliminating suboptimal arms from the active arm set, they are pulled with reduced probability. Zimmert and Seldin [30] have also proposed a best-of-both-worlds algorithm that can be applied to address the bandits with corruption setting. Adversarial corruption has also been explored for linear contextual bandits by [28; 25; 1], and Lipschitz bandits by [9]. All the aforementioned works analyze corruption within a single-agent setting. Extending any of these robust algorithms to a heterogeneous MA2B setting presents unique challenges due to differences in the local arm sets and optimal arms among agents. Section 4 discusses these challenges in more detail.

Corruption in the MA2B setting The first study to introduce the concept of corruption to the homogeneous MA2B setting is by Liu et al. [10], who extend the corruption level definition from [12; 6], utilizing the same total corruption definition formally outlined in Eq.(1). The authors achieve a high probability regret bound of $O(LC)+\tilde{O}(K/\Delta_{\min})$. The key distinction between their work and ours lies in their consideration of a homogeneous MA2B setting. They extend the BARBAR algorithm from [6] to a leader-follower framework, where the leader allocates subsets of arms to followers, who pull arms based on probabilities derived from reward gaps by the leader agent. Extending this algorithm to a heterogeneous setting introduces several technical challenges, particularly in arm assignment based on each agent's local arm set. Lastly, we distinguish our work from another line of research, as exemplified in [19; 20], which aims to develop robust algorithms in the presence of malicious agents rather than corrupted reward observations in our setting.

2. Problem Settings and Corruption Model

Multi-Agent multi-armed bandit (MA2B) model A MA2B model consists of $L \in \mathbb{N}^+$ agents in set $\mathcal{L} \coloneqq \{1, 2, \ldots, L\}$ and $K \in \mathbb{N}^+$ arms in set $\mathcal{K} \coloneqq \{1, 2, \ldots, K\}$. Access of agents to arms could be *heterogeneous* such that each agent $\ell \in \mathcal{L}$ has access to a subset of arms $\mathcal{K}_\ell \subseteq \mathcal{K}$ with size $K_\ell \coloneqq |\mathcal{K}_\ell|$, which remains the same through all rounds. Each arm $k \in \mathcal{K}$ is associated with an i.i.d. random reward bounded in $[0, 1]$ with reward mean μ_k. We refer to the distribution of arm k as $\mathcal{D}(\mu_k)$. The local best arm for agent ℓ, denoted by k^*_ℓ, is the arm with the highest reward mean in $\mathcal{K}_\ell$, i.e., $k^*_\ell \coloneqq \arg\max_{k\in\mathcal{K}_\ell} \mu_k$. For any other arm $k \in \mathcal{K}_\ell \setminus \{k^*_\ell\}$, we define its local reward gap as $\Delta_{k,\ell} \coloneqq \mu_{k^*_\ell} - \mu_k$. We also define $\mathcal{L}_k \subseteq \mathcal{L}$ as the set of agents with access to arm k, i.e.,

Procedure 1 Reward generation and corruption procedure

1 **for** $t = 1, 2, \ldots, T$ **do**
2 Draw stochastic rewards $r^t_{k,\ell} \sim \mathcal{D}(\mu_k)$ for all $\ell \in \mathcal{L}$ and $k \in \mathcal{K}_\ell$.
3 Adversary observes realized rewards $r^t_{k,\ell}$, as well as the rewards and actions of each agent in previous rounds.
4 Adversary returns the corrupted reward $\tilde{r}^t_{k,\ell} \in [0, 1]$ for all arms k and all agents ℓ.
5 Each agent ℓ pulls an arm k^t_ℓ, and observes the corrupted reward $\tilde{r}^t_{k^t_\ell,\ell}$.
end

$\mathcal{L}_k \coloneqq \{\ell \in \mathcal{L} : k \in \mathcal{K}_\ell\}$, and we set L_k as the size of $\mathcal{L}_k$. Denote $T \in \mathbb{N}^+$ as the total number of decision rounds, and set $\mathcal{T} \coloneqq \{1, 2, \ldots, T\}$.

Corruption mechanism The corrupted reward observation process is presented in Procedure 1. In each round t, stochastic rewards $r^t_{k,\ell}$ are drawn for each agent ℓ and arm k from the corresponding reward distribution (Line 2). An adversary then observes these realized rewards as well as the historical actions and rewards of rounds $1, \ldots, t-1$ for each agent (Line 3). Based on this information, the adversary returns corrupted rewards $\tilde{r}^t_{k,\ell} \in [0, 1]$ for each arm k and each agent ℓ (Line 4). Each agent ℓ then selects an arm k^t_ℓ to pull and observes only the corrupted reward $\tilde{r}^t_{k^t_\ell,\ell}$ (Line 5).

The total corruption level C across all agents is commonly defined as the summation of all agents' individual corruption level C_ℓ as $C \coloneqq \sum_{\ell=1}^{L} C_\ell$, where $C_\ell \coloneqq \sum_{t=1}^{T} \left\| \tilde{\boldsymbol{r}}^t_\ell - \boldsymbol{r}^t_\ell \right\|_\infty$ (1), with $\boldsymbol{r}^t_\ell$ and $\tilde{\boldsymbol{r}}^t_\ell$ being vectors consisting of $r^t_{k,\ell}$s and $\tilde{r}^t_{k,\ell}$s respectively. Here, the infinity norm for individual corruption C_ℓ is employed to compute the corruption level, capturing the worst-case scenario where agents consistently pull the arm with the highest corruption in each round. This infinity norm aligns with our objective of designing algorithms that are agnostic to the corruption level, and it is a standard definition in prior literature of bandits with corruptions, e.g., Gupta et al. [6]; Lykouris et al. [12].

Multi-agent communication Agents cooperate via sharing information with each other, which incurs communication costs [22; 27]. The total cost is defined as $\mathrm{Comm}(T) \coloneqq \mathbb{E}\left[\sum_{\ell \in \mathcal{L}} \sum_{t \in \mathcal{T}} \mathbb{1}\left\{\text{Agent } \ell \text{ sends a message in } t \text{ to an agent } \ell'\right\}\right]$. In a distributed scenario, all the agents follow the same communication scheme, and messages typically include the agent's pulled arm indices and observed rewards from the previous rounds.

Regret objective Each agent seeks to maximize its cumulative reward, which is equivalent to minimizing regret, a standard metric in MAB problems. Regret for each agent $\ell \in \mathcal{L}$ is defined as the difference between the accumulative rewards of the pulled arm by the bandit algorithm and the accumulative rewards of pulling the locally optimal arm over T rounds, as $\mathrm{Reg}_\ell(T) \coloneqq \mathbb{E}\left[\sum_{t=1}^{T} (\mu_{k^*_\ell} - \mu_{k^t_\ell})\right]$, where k^*_ℓ is the local optimal arm for agent ℓ, $\mu_{k^t_\ell}$ is the mean reward of the arm played by agent ℓ in round t, and the expectation is taken over the randomness of the learning algorithm. The total regret is then defined as the summation of all agents' regrets: $\mathrm{Reg}(T) \coloneqq \sum_\ell \mathrm{Reg}_\ell(T)$. We aim to devise an algorithm to minimize the total regret.

3. Algorithms

In this section, we introduce the Distributed Robust Arm Activation (DRAA) algorithm in detail.

Algorithmic technical challenges The algorithmic design of DRAA presents several technical challenges. The first challenge is due to the heterogeneous nature of the multi-agent system. Secondly, the presence of adversarial corruption complicates communication and the interpretation of shared

information. As agents need to calculate an accurate estimation of their local arms, and this estimation can easily be skewed due to the heterogeneous setting and adversarial corruption. Considering that a "bad" arm for one agent could be a "good" arm for another, estimation under this multi-agent heterogeneous setting requires careful information balancing, e.g., utilizing other agents' information with the right weights.

Another challenge lies in managing corruption while minimizing regret. Agents aim to minimize local regret, which involves pulling their best arms as frequently as possible based on reward estimates. However, adversarial corruption can make these estimates inaccurate. Therefore, traditional MAB algorithms that eliminate suboptimal arms to focus on exploiting good ones risk discarding the best arm if done under corruption. To maintain an optimal regret bound without corruption while adapting gracefully to increasing corruption, our algorithm assigns careful pull probabilities to "bad" arms, sufficient to account for potential corruption but low enough to avoid a substantial regret increase.

Main algorithmic ideas DRAA operates in epochs, each doubling its prior size. Each epoch m is divided into three phases: active arm set construction, arm-pulling and communication, and estimate updates. In the first phase — *active arm set construction* — each agent ℓ constructs an active arm set $\mathcal{A}_\ell^m$, and its complement, bad arm set $\mathcal{B}_\ell^m = \mathcal{K}_\ell \setminus \mathcal{A}_\ell^m$. Then, agents assign the arm pull probabilities to arms according to whether it is in the active or bad arm set. The second phase — *arm-pulling and communication* — involves agents pulling arms according to the assigned probabilities in all rounds of the epoch m, followed by communicating with other agents. In the final phase of *reward estimation*, agents estimates the total reward for arms using the local observed rewards and those received from others, and then calculate each arm's empirical reward gap, determining the arm pull probabilities for the next epoch. In the following paragraphs, we explain each part of the algorithm.

Active arm set construction and exploration probability calculation (Line 2-5) DRAA operates in epochs. The length of the m-th epoch is set to $\lambda K 2^{2(m-1)}/L_{\min}$ uniformly for all agents (Line 2), ensuring synchronization in starting and finishing epochs. This allows agents to share their observations during the epoch and use collective information to update arm pull probabilities for the next one. Each agent then classifies every arm in their local arm set $k \in \mathcal{K}_\ell$ as either active ($\mathcal{A}_\ell^m$) or bad ($\mathcal{B}_\ell^m$) based on the difference between the empirical mean of arm k and that of the local best arm from epoch $m-1$. Specifically, if $r_{\max,\ell}^{m-1} - r_{k,\ell}^{m-1} \leqslant 2^{-(m+2)}\sqrt{L_{\min}/L} - 3 \cdot 2^{-7}$, arm k is placed in $\mathcal{A}_\ell^m$ (Line 3); otherwise, it is placed in $\mathcal{B}_\ell^m$ (Line 4).

Arm pull probabilities differ for active arms $k \in \mathcal{A}_\ell^m$, and bad arms $k \in \mathcal{B}_\ell^m$. The goal is to assign a small but non-zero probability to bad arms to maintain good regret performance while still collecting observations in case the reward estimates are corrupted (Line 5). For bad arms $k \in \mathcal{B}_\ell^m$, the pull probability is proportional to the inverse of the empirical reward gap squared, $(\Delta_{k,\ell}^{m-1})^{-2}$, normalized by the sum of such value for all arms in the agent's local arm set. This ensures that arms with larger gaps are pulled less frequently. Using this normalization however, agents with smaller arm sets are likely to pull an equally bad arm more frequently, so this probability is scaled by K_ℓ/K to balance this effect. To prevent the number of bad arm pulls from growing too large as epochs length increases exponentially, the probability is further scaled by 2^{-2m}. It is also adjusted by $L_{\min}/L_k$, distributing the probability among all agents with access to the arm k. The remaining probability is evenly distributed among active arms in $\mathcal{A}_\ell^m$ (Line 5).

Arm-pulling and communication (Line 7-10) During epoch m, agent ℓ pulls arm k with probability $p_{k,\ell}^m$ (Line 7), and observes the reward. After the epoch m, the agent broadcasts to all other agents the sum of observed rewards $\tilde{R}_{k,\ell}^m$, the arm pull probabilities $p_{k,\ell}^m$, and the estimated local reward gaps of the previous epoch $\Delta_{k,\ell}^{m-1}$ for each arm $k \in \mathcal{K}_\ell$, along with the set $\mathcal{A}_\ell^m$. Since all these values remain

Algorithm 2 Distributed Robust Arm Activation (DRAA) for agent ℓ

Input: Full arm set $\mathcal{K}$, the local arm $\mathcal{K}_\ell$ of all agents, time horizon T, small probability δ.

Initialization: $\Delta_{k,\ell}^0 \leftarrow 1$, *and* $r_{k,\ell}^0 \leftarrow 1$ *for all* $k \in \mathcal{K}_\ell$, *and* $\lambda \leftarrow 2^{24}\log(8KL\log(T)/\delta)$.

1 **for** *epochs* $m \leftarrow 1, 2, \ldots, M$ **do**

▷Phase 1: Active Arm Constructions and Probability Calculation

2 $T^m \leftarrow \lambda K 2^{2(m-1)}/L_{\min}$

3 $\mathcal{A}_\ell^m \leftarrow \left\{k \in \mathcal{K}_\ell : r_{\max,\ell}^{m-1} - r_{k,\ell}^{m-1} < \frac{1}{2^{m+2}}\sqrt{\frac{L_{\min}}{L}} - 3\times 2^{-7}\right\}$ ▷Active arm set

4 $\mathcal{B}_\ell^m \leftarrow \mathcal{K}_\ell - \mathcal{A}_\ell^m$ ▷Bad arm set

5 $p_{k,\ell}^m \leftarrow \begin{cases} 2^{-2m}\frac{(\Delta_{k,\ell}^{m-1})^{-2}}{\sum_{k'\in\mathcal{K}_\ell}(\Delta_{k'}^{m-1})^{-2}}\frac{L_{\min}}{L_k}\frac{K_\ell}{K}, & \text{for } k \in \mathcal{B}_\ell^m \\ \frac{1}{|\mathcal{A}_\ell^m|}\left(1 - \sum_{k'\in\mathcal{B}_\ell^m} p_{k',\ell}^m\right), & \text{for } k \in \mathcal{A}_\ell^m \end{cases}$

▷Phase 2: Arm Pulling and Communication

6 **for** $t = T^{m-1}+1, T^{m-1}+2, \ldots, T^m$ **do**

7 Pick an arm k from $\mathcal{K}_\ell$ according to probability $p_{k,\ell}^m$ to pull and Observe reward $\tilde{r}_{k,\ell}^t$

end

8 $\tilde{R}_{k,\ell}^m \leftarrow \sum_{t\in E^m} \tilde{r}_{k,\ell}^t$ for all arm $k \in \mathcal{K}_\ell$

9 Broadcast $\tilde{R}_{k,\ell}^m$, $p_{k,\ell}^m$, and $\Delta_{k,\ell}^{m-1}$ for all $k \in \mathcal{K}_\ell$ and $\mathcal{A}_\ell^m$ to all agents $\ell' \in \mathcal{L}$

10 Receive $\tilde{R}_{k,\ell'}^m$, $p_{k,\ell'}^m$, and $\Delta_{k,\ell'}^{m-1}$ for all $k \in \mathcal{K}$ and $\mathcal{A}_{\ell'}^m$ from all agents $\ell' \in \mathcal{L}$

▷Phase 3: Reward Estimation

11 $r_{k,\ell}^m \leftarrow$ Estimator$(\{\forall \ell' \in \mathcal{L} : p_{k,\ell'}^m\}, \{\forall \ell' \in \mathcal{L} : \tilde{R}_{k,\ell'}^m\}, T^m)$

12 $r_{\max,\ell}^m \leftarrow \max_{k\in\mathcal{K}_\ell} r_{k,\ell}^m - \frac{1}{16}\Delta_{k,\ell}^{m-1}$

13 $\Delta_{k,\ell}^m \leftarrow \max\{2^{-3}, r_{\max,\ell}^m - r_{k,\ell}^m + 3\times 2^{-7}\}$ for all $k \in \mathcal{K}$

end

the same throughout one epoch, the communication cost is reduced from T to $O(L\log T)$ w.r.t the number of epochs (Lines 9 and 10).

Reward estimation (Line 11-13) Minimizing total regret requires each agent to pull its local good arms as frequently as possible. To do this, agents must accurately estimate the empirical means of their local arms. However, empirical mean estimation is challenging due to (a) the heterogeneous multi-agent setting and (b) adversarial corruption. The quality of an arm for an agent depends heavily on its local arm set and is assessed relative to the agent's local best arm, leading to varying arm pull probabilities for the same arm and different agents. To address this challenge, we design a weighted estimator as $r_{k,\ell}^m := \frac{\sum_{\ell'\in\mathcal{L}_k}(p_{k,\ell'}^m)^{-1}\tilde{R}_{k,\ell'}^m}{L_k T^m}$ (2). This estimator integrates observations from heterogeneous agents while minimizing bias by normalizing the sum of rewards observed by any agent $\ell' \in \mathcal{L}_k$ using the inverse of that agent's original arm pull probability. We also test a naive estimator that simply averages all rewards for arm k across agents accessing it ($\ell' \in \mathcal{L}_k$), dividing by the expected value of the total number of pulls for arm k. The naive estimator is defined as $r_{k,\ell}^m := \frac{\sum_{\ell'\in\mathcal{L}_k}\tilde{R}_{k,\ell'}^m}{\sum_{\ell'\in\mathcal{L}_k} p_{k,\ell'}^m T^m}$ (3). We then compare both estimators theoretically Section 4 and demonstrate that the weighted estimator achieves an optimal regret upper bound.

Then, the local maximum reward is computed as $\max_{k\in\mathcal{K}_\ell}\left(r^m_{k,\ell}-\frac{1}{16}\Delta^{m-1}_{k,\ell}\right)$ (Line 12). Subtracting a fraction of the previous epoch's reward gap accounts for the arm's performance history and helps mitigate sudden changes in rewards due to corruption. Each agent also calculates the empirical reward gap as $\max\{2^{-3}, r^m_{\max,\ell}-r^m_{k,\ell}\}$ for all arms $k\in\mathcal{K}_\ell$ (Line 13). A minimum value is imposed on the reward gap to prevent any arm from being assigned an excessively high pull probability.

Comparison to the BARBAR algorithm [6] and Liu et al. [10] The BARBAR algorithm operates in doubling epochs, estimating reward gaps at the end of each epoch and setting arm pull probabilities for the next epoch based on inverse-gap weighting. In contrast, our algorithm classifies arms into "active" and "bad" sets based on the difference between each arm's estimated mean and the highest estimated mean among local arms. This classification ensures that probabilities assigned to bad arms are inversely related to the total number of arms. The remaining probability is distributed among good arms, making their pulling probability inversely related to the number of good arms. Our technique effectively improves a multiplicative K factor on the corruption C (even in the single-agent setting, detailed in Section 4), resolving an open problem mentioned in [6].

Liu et al. [10]'s algorithm does not account for the number of agents accessing each arm when assigning pull probabilities. This can lead to higher pull probabilities for bad arms accessible to multiple agents. In contrast, our algorithm divides the epoch length by $L_{\min}$, the minimum number of agents with access to the same arm, and assigns pull probabilities for bad arms inversely proportional to the number of agents accessing those arms. These adjustments remove the factor of L from the corruption term in a homogeneous setting, addressing a key open problem in [10]. For another thing, [10]'s algorithm follows a leader-follower structure, which differs from our fully distributed scheme. In their leader-follower setup, each agent only explores a subset of arms, and all of their observations are uploaded to a center leader, who calculates error levels globally. The leader-follower framework in [10] aims to balance arm exploration among agents but poses challenges for extending the algorithm to heterogeneous settings. In contrast, DRAA is fully distributed, with reward gaps estimated locally. To integrate observations across agents, we use a carefully designed weighted estimator, enabling efficient use of communicated data in a local setting.

4. Theoretical Results

This section presents the theoretical regret upper bounds of DRAA. We first present the most general regret bound in the heterogeneous setting for DRAA using our weighted (Eq. (2)), and the naive estimator (Eq. (3)) to demonstrate the effectiveness of our carefully devised estimator. Then, we reduce the bound to two well-known special cases to illustrate the tightness of our results. Detailed proofs of results in this section can be found in the appendix section of our technical report [5][1].

In Theorem 1, we show that the regret upper bound of DRAA— using the weighted estimator introduced in Eq. (2) — consists of two terms: One corresponding to fully stochastic bandits, tight up to a $\log(\log T)$ factor, and another additive term of $O\left((L/L_{\min})C\right)$, where $L_{\min}$ is the minimum number of agents with mutual access to the same arm. In Remark 2, we demonstrate that deploying this weighted estimator improves the corruption term by a $L_{\min}$ factor than the naive one. Additionally, Corollary 3 demonstrates that in the special cases of single-agent MAB and homogeneous MA2B, our regret upper bound matches their corresponding lower bounds and improves upon previous upper bounds proposed in [6] and [10], respectively.

1. https://arxiv.org/abs/2411.08167

Theorem 1 (Regret upper bound of DRAA) *With probability $1-\delta$ for $\delta \in (0,1)$, the* DRAA *algorithm (Algorithm 2) using our weighted estimator incurs $O\left(L \log\left(T/\log((8K/\delta)\log T)\right)\right)$ communication costs, and its regret is upper bounded by $O\left(\frac{L}{L_{\min}} C + \log\left(\frac{KL}{\delta}\log T\right)\log T \frac{K}{\Delta_{\min}}\right)$ (4). where $L_{\min} \coloneqq \min_{k\in\mathcal{K}} L_k$ is the minimum number of agents with mutual access to any of the arms, and $\Delta_{\min} \coloneqq \min_{k\in\mathcal{K},\ell\in\mathcal{L}_k,\Delta_{k,\ell}>0} \Delta_{k,\ell}$ is the minimum non-zero local arm-gap.*

Theorem 1 demonstrates that the impact of corruption on regret decreases as the agents' access to arms becomes more uniform. Remark 2 states a h.p. regret upper bound for DRAA using the naive estimator presented in Eq. (3).

Remark 2 *With a probability $1-\delta$, the* DRAA *algorithm (Algorithm 2) using a naive estimator incurs a regret upper bounded by $O\left(LC + \log\left(\frac{KL}{\delta}\log T\right)\log TK/\Delta_{\min}\right)$ (5).*

Comparing the first term of Eq. (4) in Theorem 1 to that of Eq. (5) in Remark 2 shows that the weighted estimator improves the prefactor from L to $\frac{L}{L_{\min}}$. Because the weighted estimator can effectively leverage the overlap in agents' arm sets, minimizing the regret increase due to adversarial corruption. This also highlights the importance of devising the sophisticated estimator in Eq. (2).

Technical challenges The new algorithmic techniques in Algorithm 2 introduce novel challenges in analyzing its regret upper bound. *(1) Individually bounding arm-pull probabilities for arms in active/bad sets* The algorithm divides each agent's local arms into active and bad sets, assigning different pull probabilities. As a result, the regret depends on both an arm's local reward gap and set assignment. To analyze this, we need to bound the pull probability for agent ℓ for each arm in its active and bad sets in epoch m. *(2) Adaptive thresholding for tight regret bounds* To tightly bound the algorithm's regret, we set a threshold for the local reward gap of every arm, ensuring that the regret from arms with gaps below this threshold equals the regret from arms above it (assuming low corruption and minimal impact on the observed rewards in the latter case). This technique ensures a tight analysis by balancing the contributions to regret across different cases. Additionally, due to the set-splitting technique in Algorithm 2, pull probabilities vary across agents depending on the size of their arm sets. Therefore, the threshold must adapt to the heterogeneity of the setting.

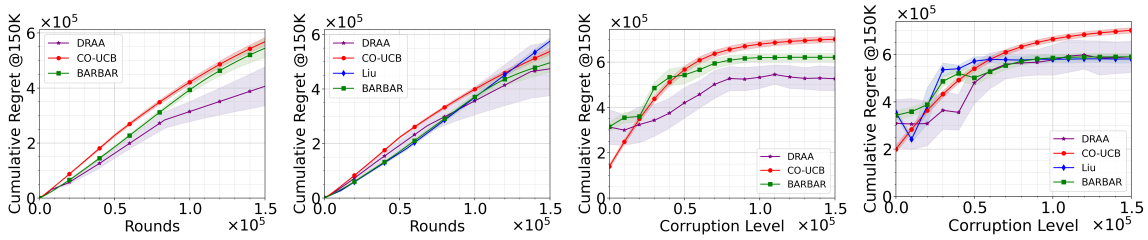
Proof Sketch Here, we provide a proof sketch for Theorem 1.

Key properties of Algorithm 2. In Lemma 1, we separately bound the arm-pull probabilities of each agent for arms in $\mathcal{A}_\ell^m$ (active arms) and $\mathcal{B}_\ell^m$ (bad arms). Then, in Lemma 2, we show that the empirical mean reward of arm k estimated by agent ℓ for epoch m closely approximates the actual mean reward μ_k, with their difference governed by two factors: the estimated reward gap from the previous epoch and the maximum amount of observed corruption for any agent over one round in epoch m. We also show that the actual number of pulls of arm k by agent ℓ during epoch m is less than two times its expected value, $p_{k,\ell}^m T^m$ with high probability. Formally, we prove that the mentioned event

$$\mathcal{E} \coloneqq \left\{\forall \ell \in \mathcal{L}, \forall k \in \mathcal{K}_\ell, m \in \{1,\dots,M\} : \left|r_{k,\ell}^m - \mu_k\right| \leqslant \frac{2C^m}{L_{\min}T^m} + \frac{\Delta_{k,\ell}^{m-1}}{16} \text{ and } \tilde{n}_{k,\ell}^m \leqslant 2p_{k,\ell}^m T^m\right\}$$

holds with probability as least $1-\delta$. Following this, we bound the estimated reward gap for the triad k, ℓ, and m, denoting the upper bound as $2(\Delta_{k,\ell} + \sqrt{L_{\min}/L}2^{-m} + \rho^m + 2^{-4})$ in Lemma 4 and the lower bound as $\frac{1}{2}\Delta_{k,\ell} - 3\rho^m - \frac{3}{4}\sqrt{L_{\min}/L}2^{-m}$ in Lemma 5. Here, ρ_m represents the cumulative corruption per agent up to epoch m. Intuitively, a smaller ρ_m will result in closer estimates of the reward gaps. Furthermore, as agents gather more information about the arms in later epochs, these reward gap estimates become increasingly accurate, hence the inverse relation with m.

Case-by-case regret bounding. Under the good h.p. event $\mathcal{E}$ in Lemma 2, we can decompose the total regret as $\text{Reg}_T \leqslant 2\sum_{m=1}^{M}\sum_{\ell\in\mathcal{L}}\sum_{k\in\mathcal{K}_\ell}\Delta_{k,\ell}p_{k,\ell}^m T^m$, replacing the number of arm pulls



(a) Regret vs. Rounds Heterogeneous (b) Regret vs. Rounds Homogeneous (c) Vary Corruption Rate Heterogeneous (d) Vary Corruption Rate Homogeneous

Figure 1: Regret of DRAA vs. Liu's vs. CO-UCB vs. BARBAR.

by their expected value. We then individually bound the regret for each unique arm k, agent ℓ, and epoch m triad, the $\Delta_{k,\ell} p_{k,\ell}^m T^m$ term in the decomposition, by analyzing the following three cases:

Case 1: The local reward gap for arm k is smaller than a set threshold, $0 \leqslant \Delta_{k,\ell} \leqslant 2^{2-m}\sqrt{L_{\min}/L}$. We apply Lemma 1 to bound the regret for both situations where arm k is in the active or bad arm set of agent ℓ in epoch m. In this case, the regret is bounded by $O(\log\left(\frac{KL}{\delta}\log T\right)\log T\frac{K}{\Delta_{\min}})$.

Case 2: The local reward gap for arm k is larger than the threshold, $\Delta_{k,\ell} \geqslant 2^{2-m}\sqrt{L_{\min}/L}$, and corruption up to epoch m is small, $\rho^{m-1} < \Delta_{k,\ell}/32$. Here, the observed reward for the arm remains close to its original stochastic value, and the chosen threshold ensures that arm k consistently falls into the bad arm set of agent ℓ. As the arm-pull probability of arms in the bad arm set depends on the inverse of the estimated reward gap of the arm, we use the lower bound for this value (Lemma 5) to bound the regret of this case, again resulting in an upper bound of $O(\log\left(\frac{KL}{\delta}\log T\right)\log T\frac{K}{\Delta_{\min}})$.

Case 3: The local reward gap for arm k is larger than the threshold, $\Delta_{k,\ell} \geqslant 2^{2-m}\sqrt{L_{\min}/L}$, and corruption up to epoch m is large, $\rho^{m-1} > \Delta_{k,\ell}/32$. Here regret is mainly controlled by corruption. We bound the effect of corruption in both scenarios, where arm k is in either the active or bad arm set of agent ℓ in epoch m. In this case, the regret is upper bounded by $O((L/L_{\min})C)$.

Summing the bounds of all three cases concludes the final regret bound.

4.1. Special cases for single-Agent MAB and homogeneous MA2B

This subsection specializes Theorem 1 to the single agent MAB and homogeneous MA2B scenarios.

Corollary 3 *In both the single-agent case, where $L = L_{\min} = 1$, and the homogeneous MA2B scenario, where $L_{\min} = L$, with probability at least $1-\delta$, the regret of Algorithm 2 is bounded as* $O\left(C + \log\left(\frac{KL}{\delta}\log T\right)\log T\frac{KL}{\Delta_{\min}}\right)$.

Corollary 3 establishes that in a single-agent MAB scenario, the regret upper bound of Algorithm 2 matches the lower bound proved by [6], improving on BARBAR's upper bound by removing a multiplicative K from the corruption term and thus resolving the primary open question mentioned in their discussions section. Furthermore, it shows that the regret upper bound of Algorithm 2 in a homogeneous MA2B setting matches the lower bound proved by [10], removing a multiplicative L from the corruption term and resolving their main open problem.

5. Numerical Results

In this section, we compare the performance of our proposed algorithm, DRAA, to Liu et al.'s [10] proposed algorithm, the non-robust CO-UCB [27], and non-cooperative BARBAR [6] under different settings. The results of our experiments are presented in Figure 1.

We consider $L = 10$ agents, and $K = 100$ i.i.d. Bernoulli arms with means uniformly sampled from $[0, 1]$. We conduct two experiments, each in both a heterogeneous and a homogeneous setting. Liu's algorithm, designed for homogeneous environments with a leader-follower structure, lacks a direct extension for heterogeneous settings and is therefore excluded from the heterogeneous experiments. In the heterogeneous scenarios, each arm is assigned to an agent with a probability of 0.4, ensuring every arm is accessible to at least one agent, and each agent has access to at least one arm. We use a two-phase corruption approach, where in the first phase, every reward observed by all agents is adversarially altered from 0 to 1 and vice versa, disrupting their exploration process completely. In the second phase, this corruption occurs with a 0.4 probability, slowing convergence after the initial phase. The cumulative regret is reported after $150K$ decision rounds. All experiments are averaged through 50 independent trials. Both CO−UCB and BARBAR are applied to the identical heterogeneous random arm assignment used for DRAA.

Experiments (a) and (b) In these experiments, we compare the cumulative regret evolution of DRAA against Liu et al.'s, CO−UCB, and BARBAR algorithms, while the length of the first phase of corruption is set to $50K$ rounds. In the first experiment shown in Figure (a), in a heterogeneous setting, DRAA considerably outperforms CO−UCB and BARBAR. Figure (b) shows that in a homogeneous setting, DRAA again achieves sublinear regret, outperforming all the other algorithms. The larger margin in the heterogeneous scenario highlights our algorithm's ability to address the challenges of heterogeneity while effectively leveraging shared arms between agents.

Experiments (c) and (d) In these experiments, we vary the first corruption phase length from 0 to $150K$ rounds. Figure (c) shows that in a heterogeneous environment DRAA consistently outperforms BARBAR across all corruption levels. Figure (d) shows that in a homogeneous setting, DRAA outperforms BARBAR and Liu's algorithm when corruption is below $60K$. However, beyond this point, all three perform similarly, as high corruption dominates the regret, surpassing the influence of both K and L. In both cases, CO−UCB performs better when corruption is below $10K$ rounds, as low corruption shifts the dominant regret factor to the stochastic logarithmic term, where CO−UCB is optimal. In contrast, DRAA and the other two robust algorithms are only optimal up to a $\log(\log T)$ factor. However, DRAA surpasses CO−UCB more quickly than BARBAR and Liu's algorithm due to its smaller corruption term $O((L/L_{\min})C)$, compared to BARBAR's $O(KC)$ and Liu's $O(LC)$.

6. Conclusion

In this paper, we introduce DRAA, an algorithm designed for the MA2B setting with heterogeneous agents, ensuring robustness to adversarial corruption. DRAA employs a distributed scenario, where all agents execute the same algorithm, taking logarithmic communication costs to achieve a regret upper bound that includes an additive term linearly related to the corruption level C. This term is in addition to the standard regret in non-corrupted settings. Our theoretical results also show that DRAA outperforms existing state-of-the-art algorithms in single-agent MAB and homogeneous MA2B settings, achieving theoretical lower bounds in these settings as an additional outcome of our approach. This work also opens up multiple future directions. Our focus is on adversarial reward corruption, while exploring communication corruption presents an interesting direction. In such scenarios, the adversary could corrupt the shared information among agents, adding a complex layer to the problem. Furthermore, although our algorithm aligns with the lower bounds of two special cases, proving lower bounds for the heterogeneous MA2B setting remains an open problem.

Acknowledgments

This work is supported by NSF CNS-2325956, CAREER-2045641, CPS-2136199, CNS-2102963, and CNS-2106299.
Xuchuang Wang is the corresponding author of this paper.

References

[1] Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR, 2021.

[2] Ronshee Chawla, Daniel Vial, Sanjay Shakkottai, and R Srikant. Collaborative multi-agent heterogeneous multi-armed bandits. *arXiv preprint arXiv:2305.18784*, 2023.

[3] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

[4] Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

[5] Fatemeh Ghaffari, Xuchuang Wang, Jinhang Zuo, and Mohammad Hajiesmaili. Multi-agent stochastic bandits robust to adversarial corruptions. *arXiv preprint arXiv:2411.08167*, 2024.

[6] Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, pages 1562–1578. PMLR, 2019.

[7] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems*, 26, 2013.

[8] Nicole Immorlica, Jieming Mao, Aleksandrs Slivkins, and Zhiwei Steven Wu. Bayesian exploration with heterogeneous agents. In *The world wide web conference*, pages 751–761, 2019.

[9] Yue Kang, Cho-Jui Hsieh, and Thomas Chun Man Lee. Robust lipschitz bandits to adversarial corruptions. *Advances in Neural Information Processing Systems*, 36, 2024.

[10] Junyan Liu, Shuai Li, and Dapeng Li. Cooperative stochastic multi-agent multi-armed bandits robust to adversarial corruptions. *arXiv preprint arXiv:2106.04207*, 2021.

[11] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE transactions on signal processing*, 58(11):5667–5681, 2010.

[12] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122, 2018.

[13] Shivakumar Mahesh, Anshuka Rangi, Haifeng Xu, and Long Tran-Thanh. Multi-player bandits robust to adversarial collisions. *arXiv preprint arXiv:2211.07817*, 2022.

[14] Abbas Mehrabian, Etienne Boursier, Emilie Kaufmann, and Vianney Perchet. A practical algorithm for multiplayer bandits when arm means vary among players. In *International Conference on Artificial Intelligence and Statistics*, pages 1211–1221. PMLR, 2020.

[15] Aritra Mitra, Hamed Hassani, and George Pappas. Robust federated best-arm identification in multi-armed bandits. *arXiv e-prints*, pages arXiv–2109, 2021.

[16] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. *Advances in neural information processing systems*, 34:22392–22404, 2021.

[17] Nícollas Silva, Heitor Werneck, Thiago Silva, Adriano CM Pereira, and Leonardo Rocha. Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications*, 197:116669, 2022.

[18] Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.

[19] Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent multi-armed bandits. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 161–170, 2021.

[20] Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent bandits over undirected graphs. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3): 1–57, 2022.

[21] Po-An Wang, Alexandre Proutiere, Kaito Ariu, Yassir Jedra, and Alessio Russo. Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 4120–4129. PMLR, 2020.

[22] Xuchuang Wang, Lin Yang, Yu-Zhen Janice Chen, Xutong Liu, Mohammad Hajiesmaili, Don Towsley, and John C.S. Lui. Achieving near-optimal individual regret & low communications in multi-agent bandits. In *The Eleventh International Conference on Learning Representations*, 2023.

[23] Xuchuang Wang, Lin Yang, Yu-Zhen Janice Chen, Xutong Liu, Mohammad Hajiesmaili, Don Towsley, and John C.S. Lui. Achieve near-optimal individual regret & low communications in multi-agent bandits. In *International Conference on Learning Representations*, 2023.

[24] Xuchuang Wang, Lin Yang, Yu-Zhen Janice Chen, Xutong Liu, Mohammad Hajiesmaili, Don Towsley, and John C.S. Lui. Exploration for free: How does reward heterogeneity improve regret in cooperative multi-agent bandits? In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

[25] Chen-Yu Wei, Christoph Dann, and Julian Zimmert. A model selection approach for corruption robust reinforcement learning. In *International Conference on Algorithmic Learning Theory*, pages 1043–1096. PMLR, 2022.

[26] Lin Yang, Yu-Zhen Janice Chen, Stephen Pasteris, Mohammad Hajiesmaili, John C.S. Lui, and Don Towsley. Cooperative stochastic bandits with asynchronous agents and constrained feedback. *Advances in Neural Information Processing Systems*, 34:8885–8897, 2021.

[27] Lin Yang, Yu-Zhen Janice Chen, Mohammad H Hajiemaili, John C.S. Lui, and Don Towsley. Distributed bandits with heterogeneous agents. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 200–209. IEEE, 2022.

[28] Heyang Zhao, Dongruo Zhou, and Quanquan Gu. Linear contextual bandits with adversarial corruptions. *arXiv preprint arXiv:2110.12615*, 2021.

[29] Pan Zhou, Kehao Wang, Linke Guo, Shimin Gong, and Bolong Zheng. A privacy-preserving distributed contextual federated online learning framework with big data support in social recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):824–838, 2019.

[30] Julian Zimmert and Yevgeny Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *The Journal of Machine Learning Research*, 22(1):1310–1358, 2021.

[31] Zhenhua Zou, Alexandre Proutiere, and Mikael Johansson. Online shortest path routing: The value of information. In *2014 American Control Conference*, pages 2142–2147. IEEE, 2014.