Robust Decentralized Multi-armed Bandits: From Corruption-Resilience to Byzantine-Resilience

Zicheng Hu, Yuchen Wang, Cheng Chen*

MoE Engineering Research Center of Hardware/Software Co-design Technology and Application
East China Normal University
{51275902019, 51285902003}@stu.ecnu.edu.cn, chchen@sei.ecnu.edu.cn

Abstract

Decentralized cooperative multi-agent multi-armed bandits (DeCMA2B) considers how multiple agents collaborate in a decentralized multi-armed bandit setting. Though this problem has been extensively studied in previous work, most existing methods remain susceptible to various adversarial attacks. In this paper, we first study DeCMA2B with adversarial corruption, where an adversary can corrupt reward observations of all agents with a limited corruption budget. We propose a robust algorithm, called DeMABAR, which ensures that each agent's individual regret suffers only an additive term proportional to the corruption budget. Then we consider a more realistic scenario where the adversary can only attack a small number of agents. Our theoretical analysis shows that the DeMABAR algorithm can also almost completely eliminate the influence of adversarial attacks and is inherently robust in the Byzantine setting, where an unknown fraction of the agents can be Byzantine, i.e., may arbitrarily select arms and communicate wrong information. We also conduct numerical experiments to illustrate the robustness and effectiveness of the proposed method.

Extended version — http://arxiv.org/abs/2511.10344

Introduction

The multi-armed bandit (MAB) problem is a classical online learning model. It has been widely applied in many real-world scenarios such as wireless monitoring (Le, Szepesvari, and Zheng 2014), medical trials (Villar, Bowden, and Wason 2015), and online advertising (Schwartz, Bradlow, and Fader 2017). In a typical MAB setup, there are K arms, each with an unknown reward distribution.

With advancements in social networks, data centers, and communication devices, multi-agent MAB problems have gained considerable attention (Boursier and Perchet 2019; Chawla et al. 2020; Huang et al. 2021; Liu, Li, and Li 2021; Wang et al. 2022b,a; Zuo et al. 2023). Among the diverse multi-agent settings, we focus on the decentralized cooperative multi-agent multi-armed bandits (DeCMA2B), where V agents collaborate on a network, sharing information with their neighbors. Many recent studies focus on improving the communication efficiency and achieving near-optimal regret bounds (Martínez-Rubio, Kanade, and Rebeschini 2019; Lalitha and Goldsmith 2021; Zhu et al. 2021, 2023). However, few of them consider the robustness of the algorithms.

In real-world applications, multi-agent systems could be disrupted by many factors, such as click fraud (Lykouris, Mirrokni, and Paes Leme 2018), denial-of-service (DoS) attacks in routing (Zhou et al. 2019), and the presence of malicious agents (Ferdowsi et al. 2019). These are mainly studied under two regimes: (i) *adversarial corruptions* (Liu, Li, and Li 2021; Ghaffari et al. 2024; Hu and Chen 2025), where an adversary can maliciously corrupt the rewards of an unknown proportion $\beta \in [0, 1]$ of the V agents with a total corruption level C, and (ii) *Byzantine agents* (Vial, Shakkottai, and Srikant 2022; Zhu et al. 2023), where agents can behave arbitrarily and send conflicting information to neighbors. Thus, a natural and important question arises:

Is there a robust algorithm that can defend against both adversarial corruptions and Byzantine agents?

In this paper, we provide a positive answer to this question by proposing the DeMABAR (**De**centralized **M**ulti-**A**gent **B**andit **A**lgorithm with **R**obustness). Our method leverages the idea from the BARBAR algorithm (Gupta, Koren, and Talwar 2019), which is robust to adversarial corruptions in single-agent scenarios. Unlike BARBAR where the epoch length depends on the instance, our DeMABAR algorithm uses an instance-independent epoch length, ensuring that all agents have the same epoch length. In this way, DeMABAR allows agents to share information with their neighbors only at the beginning and the end of each epoch, thus improving communication efficiency. Theoretical analysis reveals that DeMABAR achieves a near-optimal regret bound in DeCMA2B under adversarial corruptions, with only a communication cost of $O(wV \ln(T))$. In addition, our DeMABAR includes a novel filtering mechanism to mitigate the influence of up to αV corrupted agents, where the hyperparameter $\alpha \in [0, \frac{1}{2})$ represents the fraction of malicious agents the system can tolerate. This filtering mechanism guarantees the robustness of DeMABAR in the presence of up to αV Byzantine agents.

We summarize the individual regret comparison for the adversarial corruption setting and the Byzantine setting in Table 1 and Table 2, respectively. The main contributions of this paper are summarized as follows:

*The corresponding author.

Methods		Centralized	Decentralized
(Liu, Li, and Li 2021)		$VC + \frac{K\ln^2(T)}{\Delta}$	–
(Ghaffari et al. 2024)		$\frac{C}{V} + \frac{K\ln^2(T)}{V\Delta}$	–
(Hu and Chen 2025)		$\frac{C}{V} + \sum_{\Delta_k>0}\frac{\ln^2(T)}{V\Delta_k} + \frac{K}{V\Delta}$	–
DeMABAR (Ours)	$\beta \le \alpha$	$\frac{1}{1-2\alpha}\left(\sum_{\Delta_k>0}\frac{\ln^2(T)}{V\Delta_k} + \frac{K\ln(T)}{V\Delta}\right)$	$\frac{1}{1-2\alpha}\left(\sum_{\Delta_k>0}\frac{\ln^2(T)}{v_i^w\Delta_k} + \frac{K\ln(T)}{v_{\min}^w\Delta}\right)$
	$\beta > \alpha$	$\frac{1}{1-2\alpha}\left(\frac{C}{V} + \sum_{\Delta_k>0}\frac{\ln^2(T)}{V\Delta_k} + \frac{K\ln(T)}{V\Delta}\right)$	$\frac{1}{1-2\alpha}\left(\frac{C}{v_{\min}^w} + \sum_{\Delta_k>0}\frac{\ln^2(T)}{v_i^w\Delta_k} + \frac{K\ln(T)}{v_{\min}^w\Delta}\right)$

Table 1: Overview of expected individual regret in multi-agent MAB with adversarial corruption. We omit constant terms that are independent of T. Notice that all the above algorithms in the centralized setting need a communication cost of $O(V\ln(T))$.

Methods	Individual regret	Communication cost
(Zhu et al. 2023)	$\sum_{\Delta_k>0}\frac{\ln(T)}{\Delta_k}$	VT
DeMABAR (Ours)	$\frac{1}{1-2\alpha}\left(\sum_{\Delta_k>0}\frac{\ln^2(T)}{v_i\Delta_k} + \frac{K\ln(T)}{v_{\min}\Delta}\right)$	$V\ln(T)$

Table 2: Overview of expected individual regret and communication times in Byzantine DeCMA2B problems.

- We propose the novel DeMABAR algorithm for DeCMA2B and achieve near-optimal regret in both the adversarial corruption and Byzantine settings, with only a logarithmic communication cost.
- For DeCMA2B with adversarial corruptions, our DeMABAR algorithm achieves the following regret upper bounds for each agent i:

  If $\beta \le \alpha$, we have

  $$R_i(T) \le O\Big(\frac{1}{1-2\alpha}\Big(\sum_{\Delta_k>0}\frac{\ln^2(T)}{v_i^w\Delta_k} + \frac{K\ln(T)}{v_{\min}^w\Delta}\Big)\Big),$$

  If $\beta > \alpha$, we have

  $$R_i(T) \le O\Big(\frac{1}{1-2\alpha}\Big(\frac{C}{v_{\min}^w} + \sum_{\Delta_k>0}\frac{\ln^2(T)}{v_i^w\Delta_k} + \frac{K\ln(T)}{v_{\min}^w\Delta}\Big)\Big).$$

  The definitions of v_i^w and $v_{\min}^w$ are introduced in the notation part of the next section.
- For DeCMA2B with Byzantine agents, our DeMABAR algorithm achieves the following regret bound for each agent i:

  $$R_i(T) \le O\left(\frac{1}{1-2\alpha}\left(\sum_{\Delta_k>0}\frac{\ln^2(T)}{v_i\Delta_k} + \frac{K\ln(T)}{v_{\min}\Delta}\right)\right).$$

  The definitions of v_i and $v_{\min}$ are introduced in the notation part of the next section.
- We also perform experiments to verify the robustness and effectiveness of our method.

Related Work

DeCMA2B Most prior works on DeCMA2B (Martínez-Rubio, Kanade, and Rebeschini 2019; Lalitha and Goldsmith 2021; Chawla et al. 2020) typically use gossip-based communication protocols to achieve consensus among agents. However, the algorithms in these works are *not* robust to adversarial corruptions (Jun et al. 2018; Zuo et al. 2023): even a small amount of adversarial corruption can cause such algorithms to suffer linear regret.

DeCMA2B with adversarial corruptions Adversarial corruptions in bandits were first considered by Lykouris, Mirrokni, and Paes Leme (2018), and have attracted significant recent interest. Lykouris, Mirrokni, and Paes Leme introduced an arm-elimination algorithm with regret scaling linearly in the total corruption C, and showed that a linear dependence on C is unavoidable in general. Gupta, Koren, and Talwar (2019) proposed the BARBAR algorithm, which improves the dependence on C by more judiciously sampling suboptimal arms. Building on this idea, several works have designed robust multi-agent bandit algorithms for adversarial corruption in *centralized* settings (Liu, Li, and Li 2021; Ghaffari et al. 2024; Hu and Chen 2025), leveraging inter-agent collaboration to improve individual regret. However, to our knowledge, there is still no algorithm that is robust to adversarial corruptions in DeCMA2B.

DeCMA2B with Byzantine agents Several recent work consider bandit learning in the presence of Byzantine agents. Madhushani et al. (2021) studied an adaptive Byzantine communication model where any communicated reward can be arbitrarily altered. Vial, Shakkottai, and Srikant (2021, 2022) and Mitra et al. (2022) considered settings

where an unknown fraction of agents are Byzantine and can act arbitrarily. The approach of Mitra et al. is specialized to linear contextual bandits and relies on a central coordinator, whereas Vial, Shakkottai, and Srikant mitigate Byzantine influence by partitioning arms among agents (limiting the damage any single Byzantine can do). Most relevant to us, Zhu et al. (2023) were the first to propose a robust algorithm for DeCMA2B with Byzantine agents. They guarantee that the *individual* regret of each normal (non-Byzantine) agent is strictly smaller than in the non-cooperative case; however, the improvement is only by a constant factor, rather than scaling inversely with the number of agents as is typical in benign cooperative settings. By contrast, our approach nearly retains the $\Theta(1/v_i)$ per-agent regret improvement even in the presence of Byzantine agents (see Table 2).

Preliminaries

Problem Setup

Multi-agent multi-armed bandits we consider a setting where $[V] = \{1, 2, \ldots, V\}$ denote the set of V agents and $[K] = \{1, 2, \ldots, K\}$ denote the set of K arms. The multi-agent network of V agents is represented by the nodes of an undirected connected graph $G = ([V], E)$, where E is the set of edges. All agents face the same stochastic K-armed bandit problem over a horizon of T rounds. In each round t, every agent i selects an arm $k_{i,t}$ and receives a reward $r_{i,t}$ that is drawn i.i.d. from a fixed but unknown distribution with mean $\mu_{k_{i,t}} \in [0, 1]$. After obtaining the reward, each agent may broadcast messages to its neighbors and receive messages from its neighbors. The received information can be used in the next round if desired.

Let $k^* \in \arg\max_k \mu_k$ be an optimal arm, and we define $\Delta_k = \mu_{k^*} - \mu_k$ as the suboptimality gap of arm k, and let $\Delta = \min_{\Delta_k > 0} \Delta_k$ be the smallest positive suboptimality gap. Let $n_{i,t}^k$ be the number of times that agent i has pulled arm k up to round t. The individual pseudo-regret of agent i over T rounds is defined as

$$R_i(T) = T\mu_{k^*} - \mathbb{E}\left[\sum_{t=1}^{T} r_{i,t}\right] = \sum_{k=1}^{K} \Delta_k \mathbb{E}[n_{i,T}^k] \,.$$

For simplicity, we quantify communication cost as the total number of messages broadcast by all agents. The total communication cost over T rounds is defined as

$$\mathrm{Cost}(T) = \textstyle\sum_{i=1}^{V} \sum_{t=1}^{T} \mathbb{I}\{\text{agent } i \text{ broadcasts at time } t\}.$$

Adversarially corrupted setting In this setting, at each round $t \in [T]$, the protocol between the agents and the adversary is as follows:

1. The environment generates a reward vector $(r_{i,t}(1), \ldots, r_{i,t}(K))$ for each agent i, according to the reward distributions.
2. The adversary observes all reward vectors and generates a *corrupted* reward vector $(\tilde{r}_{i,t}(1), \ldots, \tilde{r}_{i,t}(K))$ for each agent i, based on the history of the previous $t-1$ rounds.
3. Each agent i chooses an arm $k_{i,t}$ and observes only the corrupted reward $\tilde{r}_{i,t}(k_{i,t})$ for that arm.

The corruption level of the adversary is defined as

$$C \;=\; \sum_{i=1}^{V} \sum_{t=1}^{T} \max_{k \in [K]} \big|\, \tilde{r}_{i,t}(k) - r_{i,t}(k) \,\big|.$$

We assume that for each agent i, the adversary can corrupt at most a fraction $\beta \in [0, 1]$ of its neighbors. Note that both C and β are *unknown* to the agents.

Byzantine setting In the Byzantine agent model, a subset of the agents (called Byzantine agents) may act adversarially. A Byzantine agent can select arbitrary arms in each round and send arbitrary messages to its neighbors, potentially sending different messages to different neighbors. Normal agents do not know which of their neighbors are Byzantine, but for each normal agent i, we assume that at most a fraction $\alpha \in [0, 0.5)$ of its neighbors are Byzantine. As in previous work on the Byzantine model (Vial, Shakkottai, and Srikant 2021, 2022; Mitra et al. 2022; Zhu et al. 2023), we assume that α is known to the algorithm. In the Byzantine setting, we only focus on the regret of the normal agents, since Byzantine agents can behave arbitrarily.

Relationship between adversarially corrupted and Byzantine settings In adversarially corrupted setting, an adversary can manipulate the rewards generated by the environment with a budget C. Conversely, Byzantine agents can send arbitrary information to other agents in every round, acting as if they had an infinite corruption budget. Additionally, in Byzantine setting, we only consider the individual regret of normal agents, while in adversarially corrupted setting, we consider the individual regret of all agents.

Notation

Given a graph $G = ([V], E)$, we let $d(u, v)$ denote the number of edges of a shortest path connecting nodes u and v in G. Note that we have $d(v, v) = 0$ for any node v. For an integer $w \geq 0$, we define $\mathcal{N}_w(i) = \{\, j \in V : d(i, j) \leq w \,\}$ as the set of nodes located within distance w from node i, which is also referred to as the w-neighborhood of node i. Note that we have $\{i\} = \mathcal{N}_0(i) \subseteq \mathcal{N}_1(i) \subseteq \mathcal{N}_2(i) \subseteq \cdots$. Let $D = \max_{u,v \in [V]} d(u, v)$ denote the diameter of the graph G. We define $v_i^w = \min_{j \in \mathcal{N}_w(i)} |\mathcal{N}_w(j)|$ as the minimum number of nodes within distance w of any node in the w-neighborhood of node i. We define $v_{\min}^w = \min_{j \in [V]} |\mathcal{N}_w(j)|$ as the smallest w-neighborhood size among all nodes, so we have $v_{\min}^w = \min_{i \in [V]} v_i^w$. For simplicity, we define $v_i = v_i^1$ and $v_{\min} = v_{\min}^1$.

Algorithm

In this section, we present our robust algorithm DeMABAR, summarized in Algorithm 1. For clarity, we first consider the adversarial corrupted setting, followed by the Byzantine agent model.

DeCMA2B With Adversarial Corruptions

Our DeMABAR operates in synchronized epochs, with agents allowed to broadcast information at the end of each epoch. We denote by $w \in [D]$ the *collaboration distance*,

meaning each agent i will exchange messages with the agents in its w-neighborhood $\mathcal{N}_w(i)$, which incurs a delay of $w-1$ rounds for information to propagate w hops. The hyperparameter $\alpha \in [0, 0.5)$ serves as an estimate of β, the maximum fraction of corrupted agents among any node's neighbors.

At the start of epoch m, each agent i computes an empirical suboptimality-gap estimate $\Delta_{i,k}^{m-1}$ for all arms $k \in [K]$, based on data from the previous epoch. Each arm k is expected to be pulled about $(\Delta_{i,k}^{m-1})^{-2}$ times, but capped by 2^{2m} pulls to avoid over-exploring any arm. For each agent i, all agents j in its w-neighborhood are responsible for roughly a $\frac{1}{(1-2\alpha)|\mathcal{N}_w(i)|}$ fraction of the pulls for each arm k, i.e., $n_{i,k}^m = \frac{\lambda\,(\Delta_{i,k}^{m-1})^{-2}}{(1-2\alpha)\,|\mathcal{N}_w(i)|}$. However, to satisfy the collaboration requirements of all agents in i's w-neighborhood, it may need to pull slightly more than $n_{i,k}^m$ times; thus, we define $\tilde{n}_{i,k}^m$ (line 7) as the expected number of pulls for agent i on arm k in epoch m. On the other hand, if $\sum_{k=1}^K \tilde{n}_{i,k}^m < N_m$, we select the arm k_i^m that exhibited the best performance in the previous epoch and adjust its number of pulls to be $\tilde{n}_{i,k_i^m}^m$. After N_m rounds, each agent i broadcasts the received information $(i, \{S_{i,k}^m\}_{k=1}^K, \{\tilde{n}_{i,k}^m\}_{k=1}^K)$ to its w-neighborhood, and this step requires w rounds. This communication process requires w rounds, during which each agent i selects the arm k_i^m but does not record the received reward. At the end of epoch m, each agent i uses Algorithm 2 to filter out corrupted data before the next epoch's estimates are computed.

Filtering mechanism First, for each arm k, each agent i removes the messages from agents whose number of pulls $\tilde{n}_{j,k}^m$ is lower than $n_{i,k}^m$. If too many neighbors are removed, leaving fewer than $(1-2\alpha)|\mathcal{N}_w(i)|$ neighbors, we define this event as $\mathcal{L}_{i,k}^m$, and agent i resets $n_{i,k}^m$ to the minimum observation count and restores all neighbors. Then, each agent i sets $\mathcal{B}_{i,k}^m = \mathcal{A}_{i,k}^m$ as the set of neighbors whose data will be used for its final estimate on arm k. Each agent i sorts $S_{j,k}^m/\tilde{n}_{j,k}^m$ in descending order for $j \in \mathcal{B}_{i,k}^m$ and removes the indices corresponding to the f largest and f smallest values from $\mathcal{B}_{i,k}^m$. Finally, it uses the filtered data to compute $r_{i,k}^m$ as the trimmed average.

Robustness when $\beta \le \alpha$ When the fraction of adversarially corrupted neighbors satisfies $\beta \le \alpha$, at most $\lfloor \alpha|\mathcal{N}_w(i)| \rfloor$ of agent i's neighbors can be corrupted in any epoch. Consequently, the set $\mathcal{B}_{i,k}^m$ (after filtering) will contain exactly $|\mathcal{N}_w(i)|$ agents, resulting in $f = \lfloor \alpha|\mathcal{N}_w(i)| \rfloor$. Even if up to f corrupted agents remain in $\mathcal{B}_{i,k}^m$, their average rewards $S_{j,k}^m/\tilde{n}_{j,k}^m$ will be bounded above and below by at least an equal number of uncorrupted agents. Consequently, removing the f largest and f smallest values ensures that any artificially inflated or deflated contributions from corrupted agents are eliminated, which guarantees that the estimate $r_{i,k}^m$ is close to the true mean reward.

Robustness when $\beta > \alpha$ If the adversary can corrupt more than an α fraction of its neighbors, DeMABAR still maintains robustness by never permanently eliminating any arm based on possibly corrupted data. Instead, it continues to occasionally explore every arm, allocating a limited number of pulls to each arm in each epoch. Our DeMABAR algorithm guarantees that a corruption amount of C_m in epoch m will only result in $O(C_m 2^{-(s-m)})$ additional pulls for all suboptimal arms in subsequent epochs $s > m$. Thus, the regret of our DeMABAR algorithm will only suffer an additive term that depends on the total corruption budget C.

Algorithm 1: **DeMABAR**

1: **Input:** collaboration distance w, fraction $\alpha \in [0, 0.5)$.
2: **Initialize:** $T_0 \leftarrow 0$, $\Delta_{i,k}^0 \leftarrow 1$, and $\lambda \leftarrow 2^9 \ln(2VT)$.
3: **for** all agent $i \in [V]$ in parallel **do**
4: **for** epoch $m = 1, 2, \ldots$ **do**
5: $N_m \leftarrow \left\lceil \frac{\lambda K\, 2^{m-1}}{(1-2\alpha)\, v_{\min}^w} \right\rceil$, $T_m \leftarrow T_{m-1} + N_m$.
6: $\tilde{n}_{i,k}^m \leftarrow \min\left\{ \frac{16\,\lambda\,(\Delta_{i,k}^{m-1})^{-2}}{(1-2\alpha)\, v_i^w}, \frac{\lambda\, 2^{2(m-1)}}{(1-2\alpha)\, v_i^w} \right\}$.
7: Select arm k_i^m such that $\Delta_{i,k_i^m}^{m-1} = 2^{-(m-1)}$.
8: Set $\tilde{n}_{i,k_i^m}^m \leftarrow N_m - \sum_{k \neq k_i^m} \tilde{n}_{i,k}^m$.
9: **for** $t = T_{m-1} + 1$ **to** T_m **do**
10: Pull arm $k_{i,t} \sim p_i^m$, where $p_i^m(k) = \tilde{n}_{i,k}^m / N_m$.
11: Observe corrupted reward $\tilde{r}_{i,t}(k_{i,t})$.
12: Update $S_{i,k_{i,t}}^m \leftarrow S_{i,k_{i,t}}^m + \tilde{r}_{i,t}(k_{i,t})$.
13: **end for**
14: *Communication step:*
15: **for** $t = T_m + 1$ **to** $T_m + w$ **do**
16: Pull arm k_i^m and observe corrupted reward.
17: Send message $(i, \{S_{i,k}^m\}_{k=1}^K, \{\tilde{n}_{i,k}^m\}_{k=1}^K)$ and all messages received at $t-1$ round to neighbors.
18: Receive messages from the neighboring agents.
19: **end for**
20: *Filter step:* Run Algorithm 2 to obtain $r_{i,k}^m$.
21: Set $T_m \leftarrow T_m + w$.
22: Set $r_{i,*}^m \leftarrow \max_k \{ r_{i,k}^m - \frac{1}{8} \Delta_{i,k}^{m-1} \}$.
23: Set $\Delta_{i,k}^m \leftarrow \max\{2^{-m}, r_{i,*}^m - r_{i,k}^m\}$ for each arm k.
24: **end for**
25: **end for**

Theorem 1. *In DeCMA2B with adversarial corruptions, our DeMABAR algorithm only requires a communication cost of $O(wV \ln(VT))$ to achieve the following individual regret for each agent i:*
If $\beta \le \alpha$, we have

$$R_i(T) = O\left(\frac{\ln(VT)}{1-2\alpha} \left(\sum_{\Delta_k > 0} \frac{\ln(VT)}{v_i^w \Delta_k} + \frac{K \ln(VT) \ln(\frac{1}{\Delta})}{v_{\min}^w \Delta} \right) \right),$$

If $\beta > \alpha$, we have

$$R_i(T) = O\left(\frac{\ln(VT)}{1-2\alpha} \left(\sum_{\Delta_k > 0} \frac{\ln(VT)}{v_i^w \Delta_k} + \frac{K \ln(VT) \ln(\frac{1}{\Delta})}{v_{\min}^w \Delta} \right) \right) + O\left(\frac{C}{(1-2\alpha) v_{\min}^w} \right).$$

Algorithm 2: **Filter** (agent i)

1: **for** arm $k = 1, 2, \ldots, K$ **do**
2: Initialize an available set $\mathcal{A}^m_{i,k} = \mathcal{N}_w(i)$.
3: Remove agent j from $\mathcal{A}^m_{i,k}$ if $\tilde{n}^m_{j,k} > n^m_{i,k}$.
4: **if** $|\mathcal{A}^m_{i,k}| < (1-2\alpha)|\mathcal{N}_w(i)|$ **then**
5: Set $n^m_{i,k} \leftarrow \min_{j \in \mathcal{N}_w(i)} \tilde{n}^m_{j,k}$, and $\mathcal{A}^m_{i,k} \leftarrow \mathcal{N}_w(i)$.
6: **end if**
7: Let $\mathcal{B}^m_{i,k} \leftarrow \mathcal{A}^m_{i,k}$.
8: Let $f = \frac{1}{2}\big\lfloor (|\mathcal{B}^m_{i,k}| - (1-2\alpha)|\mathcal{N}_w(i)|) \big\rfloor$.
9: Sort $S^m_{j,k}/\tilde{n}^m_{j,k}$ in descending order for $j \in \mathcal{B}^m_{i,k}$ and remove the indices corresponding to the f largest and f smallest values from $\mathcal{B}^m_{i,k}$.
10: Set $r^m_{i,k} \leftarrow \min\big\{ \frac{1}{|\mathcal{B}^m_{i,k}|} \sum_{j \in \mathcal{B}^m_{i,k}} \frac{S^m_{j,k}}{\tilde{n}^m_{j,k}}, 1 \big\}$.
11: **end for**
12: **Output:** $r^m_{i,k}$ for $k \in [K]$.

Remark 1. *In the case $\beta \le \alpha$, the regret bound in Theorem 1 is independent of C. Even a strong adversary (Zuo 2024) cannot force the DeMABAR algorithm to suffer linear regret.*

Corollary 1. *For centralized CMA2B with adversarial corruptions and $\beta = 1$, our DeMABAR algorithm with $\alpha = \frac{1}{3}$ has the following individual regret for each agent i:*

$$R_i(T) = O\left(\frac{C}{V} + \sum_{\Delta_k > 0} \frac{\ln^2(VT)}{V\Delta_k} + \frac{K \ln(VT) \ln(\frac{1}{\Delta})}{V\Delta} \right).$$

Remark 2. *As highlighted in Table 1, even in this setting, our algorithm's individual regret bound is strictly smaller (by logarithmic factors or more) than prior results for robust multi-agent bandits (Liu, Li, and Li 2021; Ghaffari et al. 2024). Compared with (Hu and Chen 2025), the main part of our regret bound is consistent with theirs.*

DeCMA2B With Byzantine Agents

Recalling the setup of the Byzantine setting, for each normal agent, at most a fraction $\alpha \in [0, 0.5)$ of its neighbors are Byzantine agents. In the Byzantine setting, we think communication at distances greater than 1 is inherently unsafe because Byzantine agents may maliciously modify the received messages and send wrong information to their neighbors. Thus we set $w = 1$, meaning that each agent only receives messages from its immediate neighbors. This method that leverages only one-hop neighbor information has also been employed in prior studies (Zhu et al. 2023; Wang et al. 2023; Liu et al. 2025) to preserve robustness.

Fortunately, the DeMABAR algorithm described earlier already includes two filtering mechanisms (in Algorithm 2, lines 4-5 and 12-13) specifically designed to handle potentially Byzantine neighbors. These filters ensure that even if some neighbors are Byzantine agents, their influence on an agent's estimates is negligible.

We can thus bound the regret of our algorithm in the Byzantine setting. The proof is deferred to the Appendix.

Theorem 2. *In DeCMA2B with Byzantine agents, with $O(V \ln(VT))$ communication cost, Algorithm 1 achieves the following regret for each normal agent i:*

$$R_i(T) = O\left(\frac{\ln(VT)}{1-2\alpha} \Big(\sum_{\Delta_k > 0} \frac{\ln(VT)}{v_i \, \Delta_k} + \frac{K \, \ln(\frac{1}{\Delta})}{v_{\min} \, \Delta} \Big) \right).$$

Remark 3. *For the case of $\alpha \le 1/3$, Zhu et al.(2023) achieve regret of $\sum_{\Delta_k > 0} \frac{\ln T}{\Delta_k}$. In contrast, our regret bound in Theorem 2 has additional v_i and $v_{\min}$ terms in the denominators, explicitly quantifying the benefit of collaboration.*

Remark 4. *Zhu et al.(2023) show that for sufficiently large T, the regret of DeCMA2B with Byzantine agents satisfies*

$$R_i(T) \ge \Omega\left(\sum_{\Delta_k > 0} \frac{\ln(T)}{(\,1-2\alpha\,)|\mathcal{N}_1(i)| \, \Delta_k} \right).$$

There remains a gap between this lower bound and our upper bound. Closing this gap or determining if it is unavoidable is an interesting open question for future work.

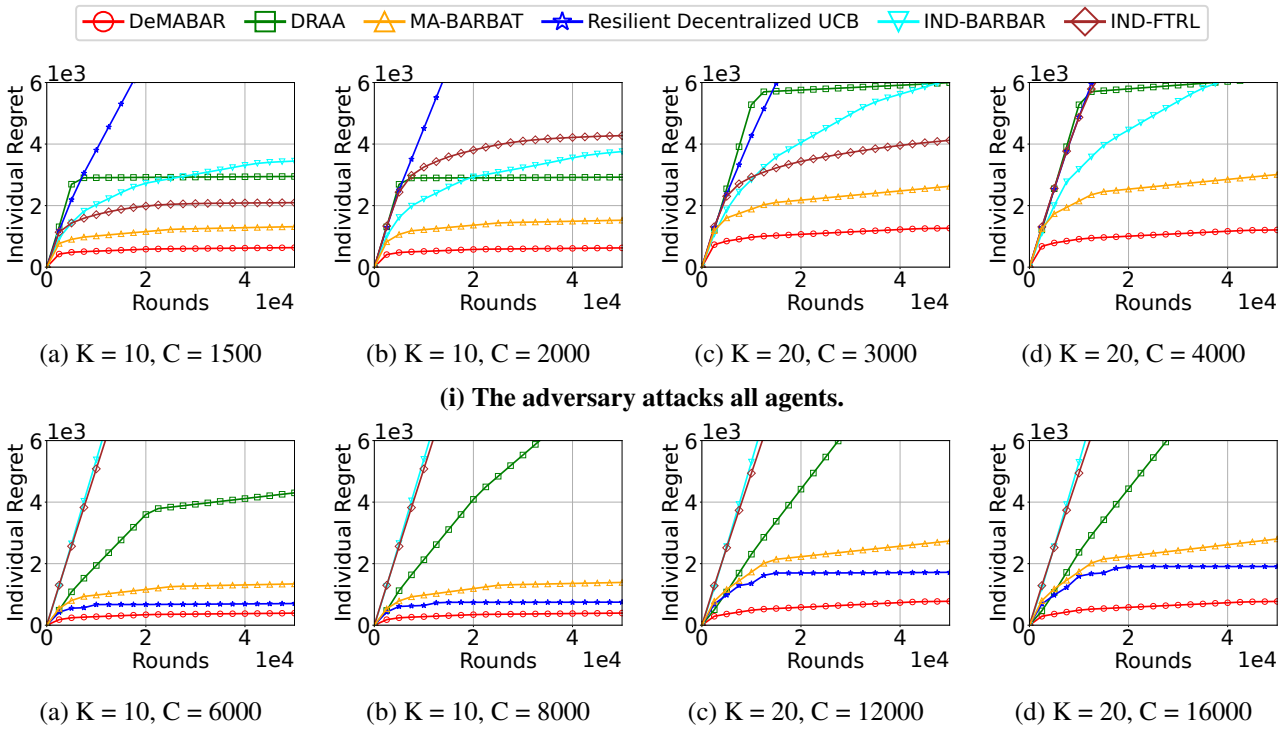
Experiments

In this section, we present numerical results to demonstrate the robustness and effectiveness of our algorithms. We consider the following baseline methods: IND-BARBAR (Gupta, Koren, and Talwar 2019), IND-FTRL (Zimmert and Seldin 2021), Resilient Decentralized UCB (Zhu et al. 2023), and DRAA (Ghaffari et al. 2024). Here, IND-BARBAR and IND-FTRL serve as non-cooperative baselines, wherein each agent runs the respective algorithm locally without inter-agent communication. We do not compare with (Liu, Li, and Li 2021) since the baseline DRAA is an improved version of their method. All experiments are implemented in Python 3.11 and conducted on a Windows laptop equipped with 16 GB of memory and a single core of an Intel i7-13700H processor. For all experiments, we run 50 independent trials and report the average total cumulative regret across all agents.

Centralized CMA2B With Adversarial Corruption

We first consider the distributed (centralized) setting and suppose the adversary can attack all agents. Each arm $k \in \mathcal{K}$ has i.i.d. Gaussian rewards with mean $\mu_k \sim U(0.1, 0.9)$ and standard deviation 0.01. We set $T = 50{,}000$. Following (Lu, Wang, and Zhang 2021), arms with $\mu_k \le 0.5$ are set as target arms. The adversary's goal is to make the agents pull the target arms as much as possible. Whenever an agent i pulls arm k with $\mu_k > 0.5$ and the realized reward is 1, the adversary corrupts this reward to $\widetilde{r}_{i,k}(t) = 0$ until the total corruption budget is exhausted. We evaluate the performance under adversarial corruption in two scenarios:

- The adversary corrupts all agents. The budgets are $C = 1500$ and 2000 for $K = 10$, and $C = 3000$ and 4000 for $K = 20$.
- The adversary targets 3 out of 10 agents (within $\beta = 0.3 < \alpha$), with a stronger attack using a higher budget per agent. The budgets are $C = 6000$ and 8000 for $K = 10$, and $C = 12000$ and 16000 for $K = 20$.



(a) K = 10, C = 1500 (b) K = 10, C = 2000 (c) K = 20, C = 3000 (d) K = 20, C = 4000

(i) The adversary attacks all agents.

(a) K = 10, C = 6000 (b) K = 10, C = 8000 (c) K = 20, C = 12000 (d) K = 20, C = 16000

(ii) The adversary attacks three agents.

Figure 1: DeMABAR vs. DRAA, Resilient Decentralized UCB, MA-BARBAT, IND-BARBAR, and IND-FTRL in centralized CMA2B under adversarial corruption.

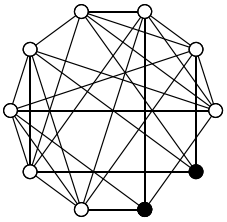


Figure 2: The network structure used in the experiment.

We present the experimental results in Figure 1, which show that our DeMABAR algorithm significantly outperforms all baseline methods, reflecting the advantage of collaboration and verifying our theoretical analysis. Notice that the Resilient Decentralized UCB method suffers nearly linear regret, which is consistent with previous findings (Jun et al. 2018) that even small adversarial attacks can degrade UCB-family algorithms to linear regret.

DeCMA2B With Adversarial Corruptions

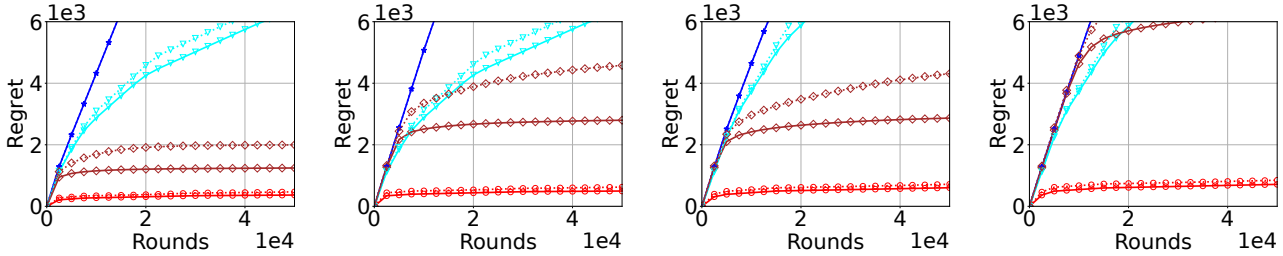
We now switch to the decentralized scenario, using the network depicted in Figure 2. The way to generate rewards and the adversary's policy are the same as in the centralized environment. Given that DRAA and MA-BARBAT are not appropriate for the decentralized setting, we do not include them in this experiment. We present the numerical results in Figure 3, which show that our DeMABAR algorithm also outperforms all baseline methods. Notice that Resilient Decentralized UCB still suffers linear regret in this setting because the adversary can attack all agents. Meanwhile, since non-cooperative algorithms such as IND-FTRL and IND-BARBAR do not exploit inter-agent communication, the performance variations are minimal in centralized and decentralized settings.

DeCMA2B With Byzantine Agents

Finally, we consider the Byzantine decentralized setting. As shown in Figure 2, two agents are Byzantine and each normal agent has at most one Byzantine neighbor. The way to generate rewards remains unchanged. Following (Zhu et al. 2023), we model two types of Byzantine attacks:

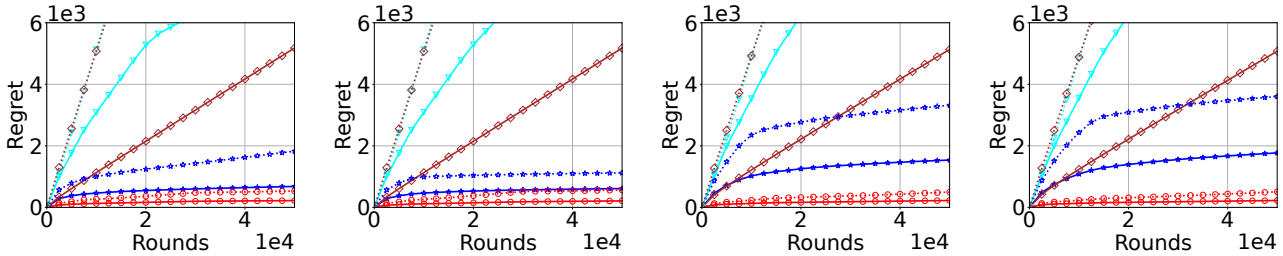
- **Adaptive attack.** A Byzantine agent has full knowledge of the system and broadcasts misleading or opposite information. For example, if $\mu_k = 0.9$, it reports $\mu_k = 0.1$ to its neighbors. It also inflates the reported sample count $n_{i,k}(t)$.
- **Gaussian attack.** Each Byzantine agent i picks a random bias $\beta_{i,k} \in (0, 1)$ for each arm k. For neighbor j at time t, it draws $c_{j,k}(t) \sim \mathcal{N}(\beta_{i,k}, 0.001)$ and adds it to the relevant statistics (e.g., $S^m_{j,k}/\tilde{n}^m_{j,k}$) before transmitting.

Figure 4 illustrates that our DeMABAR algorithm outper-



(a) K = 10, C = 1500 (b) K = 10, C = 2000 (c) K = 20, C = 3000 (d) K = 20, C = 4000

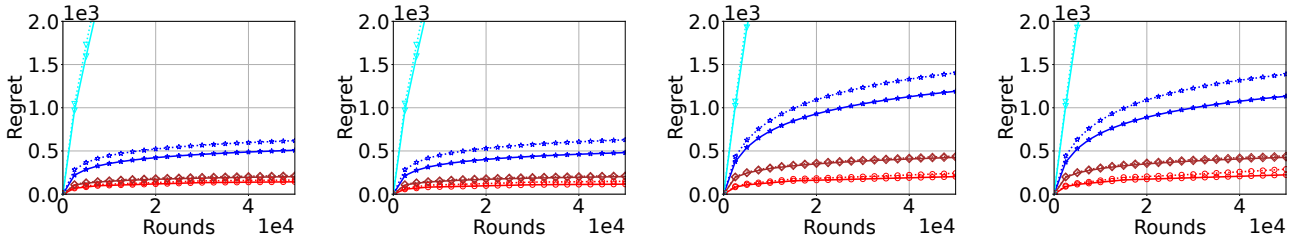
(i) The adversary attacks all agents.



(a) K = 10, C = 6000 (b) K = 10, C = 8000 (c) K = 20, C = 12000 (d) K = 20, C = 16000

(ii) The adversary attacks two agents (black nodes in Figure 2).

Figure 3: DeMABAR vs. Resilient Decentralized UCB, IND-BARBAR, and IND-FTRL in DeCMA2B under adversarial corruption.



(a) K = 10, Adaptive Attack (b) K = 10, Gaussian Attack (c) K = 20, Adaptive Attack (d) K = 20, Gaussian Attack

Figure 4: DeMABAR-F vs. Resilient Decentralized UCB, IND-BARBAR, and IND-FTRL in DeCMA2B with Byzantine agents. The Byzantine agents are the black nodes in Figure 2.

forms all baselines under both attack models. Interestingly, the performance of IND-FTRL is comparable to that of DeMABAR. We believe this is because IND-FTRL is a non-cooperative algorithm, which means that Byzantine agents cannot impact the normal agents. Additionally, the network used in the experiment may not be large enough to demonstrate the advantages of collaboration. We anticipate that in larger networks, DeMABAR will significantly outperform any non-cooperative approaches by more effectively leveraging inter-agent collaboration and filtering out Byzantine agents.

Conclusion

In this work, we present a novel robust algorithm for DeCMA2B, called DeMABAR, which facilitates effective collaboration among agents while remaining robust to both adversarial corruption and Byzantine attacks. The key idea is a novel filtering mechanism to further diminish the influence of corruption. Notably, when the adversary can attack only a small subset of agents, DeMABAR can be almost entirely unaffected by corruption. Our empirical evaluations align with these theoretical insights, showing that DeMABAR consistently outperforms baseline algorithms in both adversarially corrupted and Byzantine environments.

Acknowledgements

Cheng Chen is supported by National Natural Science Foundation of China (No. 62306116).

References

Boursier, E.; and Perchet, V. 2019. SIC-MMAB: Synchronisation involves communication in multiplayer multi-armed bandits. *Advances in Neural Information Processing Systems*, 32.

Chawla, R.; Sankararaman, A.; Ganesh, A.; and Shakkottai, S. 2020. The gossiping insert-eliminate algorithm for multi-agent bandits. In *International conference on artificial intelligence and statistics*, 3471–3481. PMLR.

Ferdowsi, A.; Ali, S.; Saad, W.; and Mandayam, N. B. 2019. Cyber-physical security and safety of autonomous connected vehicles: Optimal control meets multi-armed bandit learning. *IEEE Transactions on Communications*, 67(10): 7228–7244.

Ghaffari, F.; Wang, X.; Zuo, J.; and Hajiesmaili, M. 2024. Multi-Agent Stochastic Bandits Robust to Adversarial Corruptions. *arXiv preprint arXiv:2411.08167*.

Gupta, A.; Koren, T.; and Talwar, K. 2019. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, 1562–1578. PMLR.

Hu, Z.; and Chen, C. 2025. A Near-optimal, Scalable and Corruption-tolerant Framework for Stochastic Bandits: From Single-Agent to Multi-Agent and Beyond. *arXiv preprint arXiv:2502.07514*.

Huang, R.; Wu, W.; Yang, J.; and Shen, C. 2021. Federated linear contextual bandits. *Advances in neural information processing systems*, 34: 27057–27068.

Jun, K.-S.; Li, L.; Ma, Y.; and Zhu, J. 2018. Adversarial attacks on stochastic bandits. *Advances in neural information processing systems*, 31.

Lalitha, A.; and Goldsmith, A. 2021. Bayesian algorithms for decentralized stochastic bandits. *IEEE Journal on Selected Areas in Information Theory*, 2(2): 564–583.

Le, T.; Szepesvari, C.; and Zheng, R. 2014. Sequential learning for multi-channel wireless network monitoring with channel switching costs. *IEEE Transactions on Signal Processing*, 62(22): 5919–5929.

Liu, J.; Li, S.; and Li, D. 2021. Cooperative stochastic multi-agent multi-armed bandits robust to adversarial corruptions. *arXiv preprint arXiv:2106.04207*.

Liu, J.; Zhang, Z.; Wang, X.; Liu, X.; Lui, J.; Hajiesmaili, M.; and Joe-Wong, C. 2025. Offline Clustering of Linear Bandits: Unlocking the Power of Clusters in Data-Limited Environments. *arXiv preprint arXiv:2505.19043*.

Lu, S.; Wang, G.; and Zhang, L. 2021. Stochastic graphical bandits with adversarial corruptions. In *Proceedings of the aaai conference on artificial intelligence*, 8749–8757.

Lykouris, T.; Mirrokni, V.; and Paes Leme, R. 2018. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 114–122.

Madhushani, U.; Dubey, A.; Leonard, N.; and Pentland, A. 2021. One more step towards reality: Cooperative bandits with imperfect communication. *Advances in Neural Information Processing Systems*, 34: 7813–7824.

Mart´inez-Rubio, D.; Kanade, V.; and Rebeschini, P. 2019. Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems*, 32.

Mitra, A.; Adibi, A.; Pappas, G. J.; and Hassani, H. 2022. Collaborative linear bandits with adversarial agents: Near-optimal regret bounds. *Advances in neural information processing systems*, 35: 22602–22616.

Schwartz, E. M.; Bradlow, E. T.; and Fader, P. S. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4): 500–522.

Vial, D.; Shakkottai, S.; and Srikant, R. 2021. 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*, 161–170.

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

Villar, S. S.; Bowden, J.; and Wason, J. 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2): 199.

Wang, A.; Qin, Z.; Zheng, L.; Li, D.; and Gao, L. 2022a. Distributed robust bandits with efficient communication. *IEEE Transactions on Network Science and Engineering*, 10(3): 1586–1598.

Wang, X.; Yang, L.; Chen, Y.-z. J.; Liu, X.; Hajiesmaili, M.; Towsley, D.; and Lui, J. C. 2022b. Achieving near-optimal individual regret & low communications in multi-agent bandits. In *The Eleventh International Conference on Learning Representations*.

Wang, Z.; Xie, J.; Liu, X.; Li, S.; and Lui, J. 2023. Online clustering of bandits with misspecified user models. *Advances in Neural Information Processing Systems*, 36: 3785–3818.

Zhou, P.; Wang, K.; Guo, L.; Gong, S.; and Zheng, B. 2019. 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.

Zhu, J.; Koppel, A.; Velasquez, A.; and Liu, J. 2023. Byzantine-resilient decentralized multi-armed bandits. *arXiv preprint arXiv:2310.07320*.

Zhu, J.; Mulle, E.; Smith, C. S.; and Liu, J. 2021. Decentralized multi-armed bandit can outperform classic upper confidence bound. *arXiv preprint arXiv:2111.10933*.

Zimmert, J.; and Seldin, Y. 2021. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 22(28): 1–49.

Zuo, J.; Zhang, Z.; Wang, X.; Chen, C.; Li, S.; Lui, J.; Hajiesmaili, M.; and Wierman, A. 2023. Adversarial Attacks on Cooperative Multi-agent Bandits. *arXiv preprint arXiv:2311.01698*.

Zuo, S. 2024. Near Optimal Adversarial Attacks on Stochastic Bandits and Defenses with Smoothed Responses. In *International Conference on Artificial Intelligence and Statistics*, 2098–2106. PMLR.