Multi-Agent Best Arm Identification with Private Communications

Alexandre Rio [1] **Merwan Barlier** [1] **Igor Colin** [1] **Marta Soare** [2]

Abstract

We address multi-agent best arm identification with privacy guarantees. In this setting, agents collaborate by communicating to find the optimal arm. To avoid leaking sensitive data through messages, we consider two notions of privacy withholding different kinds of information: differential privacy and (ϵ, η)-privacy. For each privacy definition, we propose an algorithm based on a two-level successive elimination scheme. We provide theoretical guarantees for the privacy level, accuracy and sample complexity of our algorithms. Experiments on various settings support our theoretical findings.

1. Introduction

The process of gathering data to identify the best option in a stochastic multi-armed bandit is known as *pure exploration* or *best arm identification* (BAI). The goal for the learner is to suggest the most valuable option given the data collected through sequential interaction with the system. In the specific *fixed-confidence* setting, we seek to achieve this goal using as few interactions as possible, while guaranteeing that the actual best option is returned with fixed, high probability. Motivating examples include automatic parameter tuning of telecommunication antennas and pharmaceutical drug selection. Antenna performance depends on a large number of parameters, making it difficult and time-consuming for engineers to manually explore the space of configurations and find the one that will optimize the user experience. Selecting the best drug among multiple candidates is also a complex and costly process. Re-framing these problems as best arm identification in multi-armed bandits and relying on automatic strategies can be very effective.

Such problems usually involve multiple agents that share and interact with the same environment, generating as many data streams. Clinical trials involve many participants to select the most appropriate drug for the majority of the population. Antennas are generally distributed over networks spanning a specific region. Moreover, in many situations, data is distributed by nature, and cannot be consolidated and processed on a single server. It is therefore interesting for the companies to model such problems as multi-agent systems and try to exploit the data collected by each agent (i.e., *locally*) to learn the optimal decision at the system level (i.e., *globally*). We therefore refer to this problem as *multi-agent best arm identification*. Collaboration among agents is made effective as each agent broadcasts messages, reflecting the feedback they receive from the system, to a central coordinator involved in the decision process.

Large amounts of data can be handled during such learning processes, and some of them are potentially sensitive (e.g., personal data or highly valued industrial information). Hence, it is critical to design multi-agent BAI methods that ensure the privacy of the data identified as sensitive. In many applications, one may want to protect the rewards used as inputs of the learning algorithm, especially since they reflect the preferences of the agents. For instance, confidential medical data may be at risk during a drug selection study. In this situation, the well-known *differential privacy* (DP) (Dwork, 2006) framework ensures good guarantees against privacy leakage. Some other cases may require protecting the algorithm's output, that is the global optimal action. In our motivating examples, both the optimal antenna configuration and the best candidate drug are indeed of great value, as the product of the expertise of the engineers. Protecting it from competing firms is therefore of high importance. These concerns call for another notion of privacy, like the lesser known (ϵ, η)-privacy from Féraud et al. (2019).

However, if it may be possible to protect the data held by the different participants (the central coordinator and the agents), the communications may still be easily intercepted by an adversary. Moreover, they cannot always be encrypted, especially in the multi-agent setting where the computational overhead would increase significantly and where the exchange of keys is not obvious. Consequently, whereas communications from agents are a crucial building block of multi-agent algorithms, they are often a weak link in terms of privacy. Indeed, if these messages carry enough information, they can leak sensitive data. Designing

[1]Huawei Noah's Ark Lab, Paris, France [2]Université d'Orléans, Université Grenoble Alpes, CNRS LIG, France. Correspondence to: Alexandre Rio <alexandre.rio2@huawei.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

multi-agent solutions that protect sensitive information from adversaries observing the communications is thus crucial.

We refer to this problem as *multi-agent best arm identification with private communications*. In this paper, we investigate solutions to this problem, considering two scenarios based on the type of information that we want to protect: 1) the rewards, i.e. the input data, and 2) the estimated best arm, i.e., the output data. This leads us to question the most appropriate privacy notion to use in each scenario. In the first scenario, we propose DP-MASE, a multi-agent version of *DP Successive Elimination* (Sajed & Sheffet (2019)). DP-MASE is the first attempt to protect the input data in the multi-agent setting with sensitive communications, using differential privacy. To address the second scenario, we rely on the notion of (ϵ, η)-privacy, introduced in Féraud et al. (2019), and propose CORRUPTED ELIMINATION that improves over their method by decoupling privacy and local BAI accuracy, leading to better sample complexity and providing flexibility to adjust these two essential aspects of the problem. We provide a theoretical analysis for the privacy and performance of both algorithms. In particular, we show that multi-agent collaboration leads to greater accuracy than independent agents, and derive associated sample complexity bounds. Experiments on toy environments support this theoretical findings.

Structure of the paper After presenting related works (Sect. 2), we state the setting for multi-agent BAI with private communications (Sect. 3). Then, we design algorithms based on successive elimination at both local and global levels. In the first scenario where we want to protect the reward, we use differential privacy and propose DP-MASE (Sect. 4). In the second scenario, to protect the output, we rely on (ϵ, η)-privacy and propose CORRUPTED ELIMINATION (Sect. 5). Experiments are provided in (Sect. 6) to support our theoretical claims.

2. Related Work

In multi-armed bandits (MAB), the reference theoretical framework for sequential decision-making (Thompson (1933), Robbins (1952)), the problem of identifying the option (called *arm*) with the highest mean is known as *Pure Exploration* or *Best Arm Identification* (BAI). Unlike the problem of regret minimization, BAI assumes that we incur no cost in exploring suboptimal options and are just interested in returning the best arm for further exploitation: we thus do not face the well-known *exploration-exploitation* trade-off. BAI has been thoroughly studied in two different settings: *fixed-budget* (Audibert & Bubeck (2010)) and *fixed-confidence* (Even-Dar et al. (2006)). In the former, we aim at returning the best arm in a given amount of time while minimizing the probability of failure, whereas in the latter we want to output the best arm as quickly as possible with a fixed probability of failure. In this paper, we focus on the fixed confidence setting, for which Even-Dar et al. (2006) formulate and analyze two standard BAI algorithms: *Successive Elimination* and *Median Elimination*. The theoretical analysis is conducted in the *Probably Approximately Correct* (PAC) framework, where we accept to only find an ϵ-optimal arm with high probability. More advanced analysis and algorithms have been discussed, for instance in Jamieson & Nowak (2014).

Recently, much attention has been paid to multi-agent multi-armed bandits, where multiple agents collaborate to solve the same MAB instance. First, this approach can allow to take advantage of multiple computing nodes to speed up the learning process (Hillel et al. (2013), Szörényi et al. (2013)). More importantly, it accurately models real-world problems where multiple agents combine their efforts to solve a given learning task. A traditional example is the device cloud, where a possibly wide range of remote smart devices, each producing and owning their data, are solicited for the learning of a common model. However, Madhushani & Leonard (2020) point out the limits of natural extensions of traditional bandit algorithms like UCB to the multi-agent setting. A significant part of the research has been directed toward networked agents. Sankararaman et al. (2019) address the regret minimization problem by initially sharing arms among the agents and propagating the best arm using gossip protocols, much like Chawla et al. (2020) and Agarwal et al. (2021). Kumar Kolla et al. (2018) design UCB-like algorithms where statistics are aggregated over graph neighborhoods; a work recently extended to BAI by Jha et al. (2022). Other works consider a slightly different setting, named multi-bandit, where the agents have access to different bandit instances, with different arms (Gabillon et al. (2011), Scarlett et al. (2019)) or different preferences (Shi et al. (2021), Réda et al. (2022)).

In the multi-agent setting, privacy is more than ever a concern, as sensitive user data may be leaked through the messages sent by the participants. Building private algorithms is therefore a major issue. While the very definition of privacy may vary according to the context, there are two main families of techniques to design such algorithms in the MAB framework: *cryptography* ((Ciucanu et al., 2022; Garcelon et al., 2022)), and *differential privacy* (DP) ((Dubey & Pentland, 2020; Li & Song, 2022)). Encrypting the data usually has the advantage of being less harmful to the accuracy of the algorithm at the cost of additional computations, while differential privacy, consisting of adding noise in the data, requires little computation but may reduce accuracy. Differential privacy is more widely used in multi-agent settings. Tossou & Dimitrakakis (2016) and Ren et al. (2020) discuss the practical design of differentially-private MAB algorithms for the regret minimization problem, using additional

Laplace noise. They also provide theoretical regret bounds that exhibit the cost of this mechanism compared to equivalent non-private algorithms. Dubey & Pentland (2020) and Li & Song (2022) use DP mechanisms to address privacy preservation in the specific multi-agent setting, where communications between agents add an additional layer of risk. Privacy guarantees for pure exploration have been less investigated. Sajed & Sheffet (2019), for instance, propose a differentially private version of *Successive Elimination*, first described in Dwork et al. (2009), to eliminate suboptimal arms while guaranteeing privacy. (Féraud et al., 2019), on their part, address privacy in multi-agent best arm identification with a different approach. In contrast to *differential privacy*, the goal is no longer to protect the reward information that serves as input to the learning algorithm, but rather to protect its output, i.e., the identified optimal arm. Arms are eliminated both locally and globally with a voting system while having weak (not accurate) local agents ensures that an adversary cannot infer individual agent preferences by simply observing communications.

3. Multi-Agent BAI with Private Communications

3.1. Multi-Agent Bandits

We model our problem using the stochastic multi-armed bandit framework, stated in Definition 3.1.

Definition 3.1. *(Stochastic Multi-Armed Bandit)* A stochastic multi-armed bandit (MAB) with K arms, or K-armed bandit, is a set of K stochastic real distributions $(R_1, ..., R_K)$, bounded in $[0, 1]$, with means $\mathbb{E}[R_k] = \mu_k$. When a user interacts with the MAB, the user *pulls* arm $k \in \mathcal{K} = \{1, ..., K\}$ and receives a *reward* $r \sim R_k$.

We consider a multi-agent setting where a set $\mathcal{N} = \{1, ..., N\}$ of N agents collaborates to identify the best arm in a common K-armed bandit instance. When an agent $n \in \mathcal{N}$ interacts with the system at time $t \in \mathbb{N}^*$ and pulls an arm $k \in \mathcal{K}$, it receives a stochastic reward $r_k^n(t)$ drawn from some unknown distribution R_k with mean μ_k. Without loss of generality, we assume $\mu_1 \geq ... \geq \mu_K$. The objective for the system is to return an ϵ-optimal arm, *i.e.*, an arm $k^\star$ such that $\mu_{k^\star} \geq \mu_1 - \epsilon$ with high probability.

Assumption 3.2. (*Active agent*) At any round t, only one agent $n_t \in \mathcal{N}$, the *active agent*, can interact with the system.

Assumption 3.3. (*Fixed agent distribution*) The active agent at time t is sampled from a fixed distribution $P_{\mathcal{N}}$ on the agents $\mathcal{N}$, given by the environment.

Once active, an agent pulls every arm in his local active set (that is, the set of arms that the agent still considers to be potentially optimal). Over the course of the algorithm, arm k's mean is estimated through the following quantity:

$$\hat{\mu}_k^n(t) = \frac{1}{t^n} \sum_{\tau=1}^{t} r_k^n(\tau) \mathbb{1}\left[n_\tau = n\right] \quad ,$$

where t^n is the number of rounds agent n has been active up to time t, such that $t^n := t^n(t) = \sum_{\tau=1}^{t} \mathbb{1}\left[n_\tau = n\right]$. Note that Assumption 3.2 simply models asynchronous interactions with the system, which corresponds to several real-world applications (e.g., telecommunication networks).

We consider a centralized setting so that agents collaborate by sending messages to a central coordinator. This is equivalent to considering a decentralized setting where agents are connected with a complete graph. The messages contain the indices of the arms they have eliminated locally. The messages sent by agent n up to time t can be summarized as a vector $(\lambda_k^n(t))_{k\in[K]}$ where $\lambda_k^n(t) = 1$ if agent n has already eliminated arm k at time t, and $\lambda_k^n(t) = 0$ otherwise.

Assumption 3.4. (*Communication between agents*) Each time the active agent n_t eliminates an arm locally, he sends its index k to the central coordinator.

Notation Let $\mathcal{M}_n(t) = (\lambda_k^n(t))_{k\in[K]}$ denote the messages sent by agent $n \in \mathcal{N}$ up to time t, and $\mathcal{M}(t) = (\mathcal{M}_1(t), ..., \mathcal{M}_N(t))$. Let also $\mathcal{M}_n$ denote the set of all messages sent by agent $n \in \mathcal{N}$ and $\mathcal{M} = (\mathcal{M}_1, ..., \mathcal{M}_N)$.

3.2. Multi-Agent BAI with Private Communications

In the context of *multi-agent BAI*, we are concerned with withholding two types of data one may want to protect : 1) the input data, namely the rewards, and 2) the output of the BAI algorithm, namely the identity of the optimal option. Moreover, we assume that the privacy of these data may be compromised through the messages sent by the agents. We thus propose privacy notions that aim at protecting the privacy of either the inputs or the output of multi-agent BAI algorithms from adversaries observing the communications.

3.2.1. Protecting the Input with Differential Privacy

The first situation is typically handled using *differential privacy*. Given two input datasets that only differ from one data point — referred to as *neighboring datasets*, differential privacy ensures that the distribution of the output of the learning algorithm (often called a *mechanism*, i.e., a randomized function of the data) does not change significantly. Definition 3.5 formalizes this notion in the general case.

Definition 3.5. (ϵ-*differential privacy*) For any $\epsilon > 0$, the mechanism $\mathcal{Q}$ is ϵ-differentially private if for any pair of neighboring datasets (D, D') and any event S in $\mathcal{Q}$'s range:

$$\mathbb{P}\left(\mathcal{Q}(D) \in S\right) \leq e^{\epsilon} \mathbb{P}\left(\mathcal{Q}(D') \in S\right) \quad .$$

To obtain a ϵ-DP mechanism from a query (i.e., a function of the data), we usually add an *ad hoc* noise with a well-tuned

scale. The scale usually depends on the global sensitivity of the query, which is the maximum amount by which the query value can change between evaluations on two neighboring datasets. For instance, given a real-valued query f of sensitivity $\mathcal{S}(f)$, we obtain a ϵ-DP mechanism $\mathcal{Q}$ by adding a centered Laplace r.v. with scale $\frac{\mathcal{S}(f)}{\epsilon}$ to f. This procedure is called the Laplace mechanism.

A BAI algorithm learns the best arm based on the sequence of rewards collected through interactions with the system: these rewards may reveal sensitive information as they reflect the preferences of the agents. Since we consider that an adversary may observe the communicated messages (and not the best arm itself), we seek differential privacy with respect to the mechanism that outputs the set of messages sent by one agent. We then guarantee that the presence of one particular reward in the input dataset does not affect the message distribution significantly.

We give in Definition 3.6 a statement of DP adapted to our particular setting. Let $D_{1:t}$ be the set of all the rewards received by the N agents up to time t. $D_{1:t}$ and $D'_{1:t}$ are considered neighbors if they only differ by one reward. We consider the mechanism $\mathcal{Q}^n$ that takes as input a dataset $D_{1:t}$ and outputs $\mathcal{M}_n$.

Definition 3.6. (ϵ-*differential privacy for communications*) For any $\epsilon > 0$ and any agent $n \in \mathcal{N}$, the mechanism $\mathcal{Q}_n$ is ϵ-differentially private if for any time t, any set of messages $\mathcal{M}_n(t) \in \{0,1\}^{[K]}$, and any pair of neighboring datasets $D_{1:t}$, $D'_{1:t}$, the following holds:

$$\mathbb{P}\left(\mathcal{Q}_n(D_{1:t}) = \mathcal{M}_n(t)\right) \leq e^{\epsilon}\mathbb{P}\left(\mathcal{Q}_n(D'_{1:t}) = \mathcal{M}_n(t)\right) \quad .$$

3.2.2. Protecting the Output with (ϵ, η)-Privacy

Some applications may require protecting the algorithm's output, i.e., the global optimal action. This information can indeed be of high value, for instance when it is the product of the engineers' expertise in industrial applications. Protecting it from competing firms is therefore of critical importance. However, in multi-agent systems, it is difficult to prevent adversaries from observing the messages sent by the different participants. If those messages carry enough information about the optimal action, their identity can leak—even if it is well protected at the system level.

We are interested in designing methods intended to protect against this eventuality, which is formalized through the notion of (ϵ, η)-privacy, first introduced in Féraud et al. (2019). (ϵ, η)-privacy, stated in Definition 3.7, ensures that an adversary observing the messages sent by one agent cannot infer an ϵ-best arm with high confidence. Limiting the awareness of the adversary to the messages from a single agent is realistic in many applications, as we can imagine installing a bug on one particular device. However, this could be easily extended to the setting where the adversary may watch up to c agents, given a loss in the privacy guarantees.

Definition 3.7. ((ϵ, η)-*privacy*) An algorithm $\mathcal{A}$ is considered (ϵ, η)-private for finding an ϵ-optimal arm if, for any agent $n \in \mathcal{N}$, an adversary that knows $\mathcal{A}$ and has access to the set of messages $\mathcal{M}_n$ cannot infer what arm is ϵ-optimal for agent n with probability higher than $1 - \eta$.

In particular, if $1 - \eta \leq 1/K$, the adversary cannot make better than a random guess.

3.3. Successive Elimination in Multi-Agent MAB

In multi-agent algorithms, messages are signals that reflect the feedback an agent receives through his interactions with the environment. Sharing the signals with the other agents enables collaboration and allows better decision-making at the global level, compared to the situation where each agent learns independently. In multi-agent bandits, one could imagine a variety of such signals. However, in many applications, the amount of information contained in the messages is constrained by the capacity of the communication channels. In this work, we consider simple messages with integer values, fitting situations where communication is heavily constrained.

BAI algorithms based on successive arm elimination are particularly suited to this setting. These methods work by maintaining an active set of arms that is progressively reduced as the agent eliminates suboptimal arms. An arm is eliminated when it is estimated suboptimal with enough confidence, based on the rewards collected during interactions with the system. The algorithm stops when the active set is reduced to a single arm, which is the estimated best arm. As an important example, Successive Elimination from Even-Dar et al. (2006) guarantees to output an ϵ-optimal arm with confidence $1 - \delta$ in approximately $\frac{K}{\epsilon^2}\log\frac{1}{\delta}$ steps.

It is quite straightforward to use successive elimination schemes within multi-agent systems and enable collaboration through integer-valued messages and a voting process. At a high level, each agent n independently runs a local BAI algorithm, managing his own local set of arms $\mathcal{K}^n$, while a central coordinator separately maintains a global set of arms $\mathcal{K}$. Every time an agent eliminates an arm locally, he sends its index to the central coordinator, hence "voting against" this arm. If the number of votes λ_k against a specific arm exceeds some predefined threshold M, it is removed from the global active set as well as from all local active sets. This multi-agent successive elimination (MASE) scheme at both local and global levels is described in Algorithm 1. In the following, we propose methods based on this multi-agent successive elimination scheme that ensures differential privacy and (ϵ, η)-privacy.

Algorithm 1 Multi-Agent Successive Elimination(MASE)

Input: $\epsilon > 0$, $\delta \in [0,1]$, voting threshold $M \in [\![1, N]\!]$
Output: Estimated best arm in $\mathcal{K}$
Active sets $\mathcal{K} = \mathcal{K}^n = [K]$; global votes $(\lambda_k = 0)_{k\in[K]}$; local epochs $e^n = 0$; local round $r^n = 0$
while $|\mathcal{K}| > 1$ **do**
  Active agent n is sampled: $n \sim P_{\mathcal{N}(t)}$
  n interacts with system and updates mean estimates
  $t^n := t^n + 1$
  if condition for local elimination is met by k **then**
    Remove k from local active set: $\mathcal{K}^n := \mathcal{K}^n \setminus \{k\}$
    $\lambda_k := \lambda_k + 1$
    $e^n := e^n + 1$
    if $\lambda_k > M$ **then**
      Remove k from global active set: $\mathcal{K} := \mathcal{K} \setminus \{k\}$
      Remove k from all local active sets: $\mathcal{K}^n := \mathcal{K}^n \setminus \{k\}$ for all $n \in \mathcal{N}$

Sample Complexity of MASE. We expect the sample complexity, computed as the total number of overall interactions with the system, to scale up with number of agents. Indeed, in the asynchronous case (Assumption 3.2), more agents means more timesteps before a given agent has interacted enough with the system to output the best arm with high confidence. [1]

For a lower bound, we consider the extreme, non-private case, where agents share all their raw reward samples with the central coordinator (the set of messages that can be sent is then larger than what is described in Section 3.1). Up to negligible communication delays, the problem becomes equivalent to the single-agent setting. It is then relevant to compare our algorithms with single-agent SUCCESSIVE ELIMINATION, for which lower bounds exist (see, for instance, Even-Dar et al. (2002)), although such approach does not offer any privacy.

4. Multi-Agent Successive Elimination with Differentially Private Communications

In this section, we introduce DP-MASE, for *Differentially Private Multi-Agent Successive Elimination*. While learning the best option in the multi-armed bandit, DP-MASE guarantees that the communications are differentially private with respect to the rewards. In essence, DP-MASE follows the multi-agent successive elimination scheme described in Algorithm 1, and uses *DP Successive Elimination* from Sajed & Sheffet (2019) as a local BAI routine.

[1] Formally, the per-agent counts $(t^1, ..., t^N)$ follow a multinomial with parameters $(t, (p_1, ..., p_N))$, where $p_i = P_{\mathcal{N}}(i)$. Moreover, the algorithm stops at step t if $\max_M \left(t^1, ..., t^N\right) > T(\eta)$, where $\max_M$ denotes the M-th largest value. For a given T, the probability $\mathbb{P}\left(\max_M (T_1, ..., T_N) > T(\eta)\right)$ decreases with N, which necessarily links the sample complexity to N.

4.1. Differentially Private Successive Elimination

In Successive Elimination (Even-Dar et al. (2006)), we consider the uncertainty over the value of each arm using a confidence interval of decreasing radius around the empirical mean. More specifically, at step t, we want the true mean of arm k to lie within the following confidence interval, for a well-chosen $\alpha(t) \in \mathbb{R}$, with high probability:

$$\mu_k \in [\hat{\mu}_k(t) - \alpha(t); \hat{\mu}_k(t) + \alpha(t)] \quad .$$

If we set $\alpha(t) = \sqrt{\frac{\log(cKt^2/\beta)}{t}}$ for some $c > 0$, we know this holds with high probability $1 - \frac{2\beta}{cKt^2}$ due to Hoeffding's inequality. An arm k is eliminated as soon as its upper bound is less than the lower bound of the current best arm estimate, i.e.:

$$\max_{l\in\mathcal{K}} \hat{\mu}_l(t) - \hat{\mu}_k(t) > 2\alpha(t) \quad . \tag{1}$$

From then on, we refer to (1) as *suboptimality evaluation*. This procedure returns an ϵ-optimal best arm with probability at least $1 - \beta$, in at most $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\beta}\right)$ steps.

Making Successive Elimination differentially private is not straightforward. Naively, one could simply release noisy values for the means instead of raw values (for instance using the Laplace mechanism). However, this approach raises practical challenges. Indeed, a reward sample from arm k is involved in every subsequent empirical mean $\hat{\mu}_k$, making the privacy cost scale linearly with T, the total number of steps. To circumvent this, Sajed & Sheffet (2019) only perform suboptimality evaluation after a given number of steps (*i.e.* an epoch). Empirical means are then reset and samples are discarded. Therefore, each reward sample is only consumed in a single mean estimation. This approach also allows controlling global sensitivity of the queries and the scale of the Laplace noise appropriately.

Thanks to parallel composition, Sajed & Sheffet (2019) prove that DP Successive Elimination is ϵ-differentially private. It also effectively returns the best arm as the epoch length $R(e)$ grows properly over time, each time allowing for more accurate mean estimates. Indeed, $R(e)$ is such that every arm with a suboptimality gap greater than 2^{-e} is eliminated at epoch e with high probability.

4.2. A DP Algorithm for Multi-Agent BAI

Our objective is now to propose a multi-agent version of DP Successive Elimination, based on the MASE scheme. We obtain DP-MASE, a new algorithm for multi-agent BAI with private communications, depicted in Algorithm 2.

Algorithm 2 DP-MASE

1: **Input:** ϵ, δ, β, $M = \lceil \frac{\log \delta}{\log \beta} \rceil$
2: **Output:** Estimated best arm in $\mathcal{K}$
3: Active agents $\mathcal{N}(t) = \mathcal{N}$; global and local active sets $\mathcal{K} = \mathcal{K}^n = [K]$; global votes $(\lambda_k = 0)_{k\in[K]}$; local epochs $e^n = 0$; local round $r^n = 0$; global time $t = 0$
4: **while** $|\mathcal{K}| > 1$ **do**
5: Active agent n is sampled: $n \sim P_{\mathcal{N}(t)}$
6: Sample all arms in $\mathcal{K}^n$ and update mean estimates
7: $r^n := r^n + 1$
8: **if** $r^n = R(e^n)$ **then**
9: $\bar{k}(e^n) = \text{LocalElimination}\left(\epsilon, e^n, (\hat{\mu}_k^n(e^n))_k, \beta\right)$
10: $\mathcal{K}^n = \mathcal{K}^n \setminus \bar{k}(e^n)$
11: $\lambda_k := \lambda_k + 1$ for any $k \in \bar{k}(e^n)$
12: **if** $|\mathcal{K}^n| = 1$ **then**
13: $\mathcal{N}(t) := \mathcal{N}(t) \setminus \{n\}$ {Remove agent if done}
14: **else**
15: $e^n := e^n + 1$
16: Reset $(\hat{\mu}_k^n(e^n) = 0)_k$, $r^n = 0$
17: Compute next epoch length $R(e^n)$
18: $\mathcal{K}^n := \mathcal{K}^n \bigcap \mathcal{K}$ {Remove the latest globally eliminated arms from local active set}
19: **for** $k \in \bar{k}(e^n)$ **do**
20: $\mathcal{K} := \mathcal{K} \setminus \{k\}$ **if** $\lambda_k > M$
21: t := t + 1

In DP-MASE, each agent independently runs a DP Successive Elimination instance on the shared multi-armed bandits. To deal with the asynchronous interactions, every agent keeps track of his current epoch e^n and current round $r^n \in [\![0, R(e^n)]\!]$ within this epoch. At the end of his epoch, an agent performs local elimination as in Sajed & Sheffet (2019), returning a set $\bar{k}(e^n)$ of eliminated arms. If it is not empty, arms in $\bar{k}(e^n)$ are immediately removed from the local active set $\mathcal{K}^n$, and their global voting counts λ_k's are updated (lines 9-11). Empirical means are then reset for the next epoch whose length is also computed using the same formula as in Sajed & Sheffet (2019):

$$\max\left(\frac{32\log\left(8|\mathcal{K}^n|e^{n2}/\beta\right)}{2^{-2e^n}}; \frac{8\log\left(4|\mathcal{K}^n|e^{n2}/\beta\right)}{\epsilon 2^{-e^n}}\right) .$$

Moreover, $\mathcal{K}^n$ is synchronized with global active set $\mathcal{K}$, meaning that arms that have been globally eliminated during epoch e^n are now also removed from $\mathcal{K}^n$ (lines 15-18). Eventually, if votes against arms in $\bar{k}(e^n)$ have reached the voting threshold, they are removed from global active set $\mathcal{K}$ (lines 19-20).

We show that DP-MASE is ϵ-differentially private for communications, in the sense of Definition 3.6, and also derive guarantees over its accuracy for finding an optimal arm. Detailed proofs are provided in the appendix.

Proposition 4.1. *Algorithm 2,* DP-MASE*, is ϵ-differentially private for communications.*

Proposition 4.2. *(*Failure Probability of DP-MASE*) With voting threshold $M = \lceil \frac{\log \delta}{\log \beta} \rceil$,* DP-MASE *fails to return the best arm with probability at most δ.*

Eventually, we derive a problem-dependent sample complexity bound for DP-MASE that holds with high probability.

Proposition 4.3. *(*Sample Complexity of DP-MASE*) With probability at least $(1-\delta)\left(1 - I_{1-p^\star}\left(T - T(\beta), 1 + T(\beta)\right)\right)^M$,* DP-MASE *has sample complexity:*

$$\mathcal{O}\left(\frac{1}{p^\star}T(\beta) + \frac{1}{4p^{\star 2}}\log\frac{1}{\delta}\right) ,$$

with:

$$T(\beta) = \log(K/\beta) + \log\log(1/\Delta_2)\left(\frac{1}{\Delta_2^2} + \frac{1}{\epsilon\Delta_2}\right) .$$

Above, $I_p(a, b)$ is the regularized incomplete beta function evaluated at p with parameters (a, b), and $p^\star$ is the probability of the M-th most likely agent.

In Proposition 4.3, $T(\beta)$ is the local sample complexity, that is the number of steps a single-agent running DP Successive Elimination must execute to output the best arm with confidence $1 - \beta$. $T(\beta)$ naturally increases with the confidence and the strength of the privacy guarantees. It also depends negatively on the smallest suboptimality gap Δ_2, which quantifies the hardness of the BAI problem. The global sample complexity of DP-MASE obviously scales with the local sample complexity. However, it also depends on the agent distribution $P_{\mathcal{N}}$, and particularly on the inverse probability of the M-th most likely player: through the global elimination process, the algorithm roughly "waits" for the M-th most frequent player to output his best arm estimate.

DP-MASE is of practical interest when rewards may carry sensitive information and when it is impossible to completely secure the communication channels. However, we could be concerned about withholding other information, such as the optimal option. In this situation, we consider that DP is not the most adapted notion. In the next section, we therefore propose another method based on multi-agent successive elimination that better fits this objective.

5. Best Arm Privacy in Multi-Agent BAI

Here, we are no longer concerned with preserving the input data, but rather want to protect the learned optimal option. We believe that the most suitable privacy notion for this problem is (ϵ, η)-privacy, first introduced in Féraud et al. (2019). In this section, we first discuss existing solutions and later introduce a new, more efficient multi-agent BAI algorithm that guarantees (ϵ, η)-privacy.

5.1. Multi-Agent Successive Elimination with (ϵ, η)-Private Communications

To address this setting, Féraud et al. (2019) propose a (ϵ, η)-private algorithm based on multi-agent successive elimination. To reach (ϵ, η)-privacy, each local arm selection subroutine is run with precision ϵ and confidence $1-\eta$. Such low local confidence implies that observing a single agent does not bring any useful information, making the algorithm (ϵ, η)-private, while the consensus required to eliminate an arm globally ensures a low failure probability. More precisely, setting the global elimination threshold as $M = \lceil \frac{\log \delta}{\log \eta} \rceil$ ensures that an ϵ-best arm is returned with a probability higher than $1-\delta$.

Moreover, upper bounds on the sample complexity are derived that directly depend on the local subroutine complexity and the agent probability distribution $P_{\mathcal{N}}$. In particular, for any $(1-\eta)$-confident BAI subroutine with sample complexity $T(\eta)$, the global sample complexity scales with $\frac{1}{p^\star} T(\eta)$, where $p^\star$ is the probability of the M-th most likely agent, similarly to DP-MASE.

5.2. Improved Privacy with Message Corruption

The privacy mechanism in Féraud et al. (2019) inherently links the local confidence level with the privacy level. Indeed, the algorithm is private in the sense of Definition 3.7 only because each independent participant is inaccurate in his best arm identification, returning an ϵ-optimal arm with confidence as low as $1-\eta$.

This is likely to harm sample complexity: to guarantee high global accuracy, the voting threshold M must be high to compensate for the low local BAI confidence. In this section, we provide a method that decouples global privacy and local accuracy, providing more flexibility to adjust these two essential aspects of the problem and leading to better sample complexity.

Our method to guarantee a high privacy level without sacrificing local BAI confidence comes from the observation that the adversary is only able to derive critical information from sent messages and has no access to the internal state of the observed agent. Thus, instead of guaranteeing privacy by having poor local best arm identification, we limit the information that could be extracted from the sent messages by corrupting them with a random binary noise. More formally, instead of sending $\mathcal{M}_n$, agent n sends the corrupted messages $\mathcal{M}_n^\xi = (\lambda_1^n \times \nu_1^n, ..., \lambda_k^n \times \nu_K^n)$, where $\xi > 0$ is the corruption probability and $(\nu_k^n)_{k\in[K]}$ is the corruption mask such that $\nu_k^n \sim \mathcal{B}(1-\xi)$ for any k. We can then apply *Successive Elimination*. Algorithm 3 describes the whole procedure. We now establish the privacy guarantees and the failure probability of Algorithm 3. Complete proofs are provided in appendix.

Algorithm 3 CORRUPTED ELIMINATION

Input: η, δ, ξ, $M = \lceil \frac{\log \delta}{\log \eta} \rceil$
Output: Estimated best arm in $\mathcal{K}$
Active agents $\mathcal{N}(t) = \mathcal{N}$; global and local active sets $\mathcal{K} = \mathcal{K}^n = [K]$; local failure probability $\eta_\xi = 1 - \frac{1-\eta'}{(1-\xi)^{K-1}}$, $\mathcal{K}^n := \mathcal{K}$ for all $n \in \mathcal{N}$; global votes $(\lambda_k = 0)_{k\in[K]}$, global time $t = 0$
while $|\mathcal{K}| > 1$ **do**
  Active agent n is sampled: $n \sim P_{\mathcal{N}(t)}$
  Sample all arms in $\mathcal{K}^n$ and update mean estimates
  $\bar{k}, , \bar{k}_G =$ LocalElimination (n, ξ) {Identify arms to eliminate locally}
  $\mathcal{K}^n = \mathcal{K}^n \setminus \bar{k}$ {Remove eliminated arms from local active set}
  $\lambda_k := \lambda_k + 1$ for any $k \in \bar{k}_G(e^n)$ {Vote for global elimination of eliminated arms if corresponding message not corrupted}
  if $|\mathcal{K}^n| = 1$ **then**
    $\mathcal{N}(t) := \mathcal{N}(t) \setminus \{n\}$
  $\mathcal{K}^n := \mathcal{K}^n \bigcap \mathcal{K}$ {Remove the latest globally eliminated arms from local active set}
  for $k \in \bar{k}$ **do**
    $\mathcal{K} := \mathcal{K} \setminus \{k\}$ **if** $\lambda_k > M$
  $t := t + 1$

Proposition 5.1. *(*Privacy of CORRUPTED ELIMINATION*) Running local BAI subroutines with confidence $1-\eta'$, an adversary knowing $\mathcal{A}$ and observing $\mathcal{M}_n^\xi$ cannot infer the best arm of agent n with probability higher than $(1-\eta') \times (1-\xi)^{K-1}$. Therefore, to maintain an apparent privacy level η, we need to set the confidence level of the BAI subroutine as:*

$$\eta_\xi = \max\left(0, 1 - \frac{1-\eta}{(1-\xi)^{K-1}}\right) \leq \eta \ .$$

Proposition 5.2. *(*Failure probability of CORRUPTED ELIMINATION*) The failure probability of* CORRUPTED ELIMINATION *is at most:*

$$\min_{\alpha\in\mathbb{N}} \left(1 - I_\xi\left(\alpha+1, M\right)\right) \times \delta_\alpha \ , \text{ where } \delta_\alpha = \delta \times {\eta_\xi}^\alpha \ .$$

Proposition 5.1 tells us that we can run the local subroutines with better accuracy than Féraud et al. (2019) while keeping the same privacy guarantees. With Proposition 5.2, failure probability is less than without communication noise. Indeed, since some messages are corrupted, more independent agents need to vote against the best arm to remove it globally. In Proposition 5.3, we propose an approximation of the sample complexity of CORRUPTED ELIMINATION.

Proposition 5.3. *(*Sample complexity of CORRUPTED ELIMINATION*) If the agent distribution $P_{\mathcal{N}}$ is uniform, then with probability α, the number of*

rounds needed by CORRUPTED ELIMINATION *to output an ϵ-optimal arm is approximately:*

$$T = N \times Q^{-1}\left(T(\eta_\xi), 1 - \alpha^{1/M}\right) \quad ,$$

where Q^{-1} is the inverse of the regularized gamma function, and $T(\eta_\xi)$ is the number of local samples needed for the BAI subroutine to find the optimal arm with confidence $1 - \eta_\xi$.

As we observe in our experiments (see, e.g., appendix and figures therein), there is a positive correlation between the global sample complexity and the local failure probability η_ξ (corresponding exactly to the privacy level when $\xi = 0$), even with uniform sampling. Proposition 5.3 provides a first theoretical justification for these empirical findings. Indeed, this observation was not reflected in the initial bound from Féraud et al. (2019), as increasing BAI confidence $1 - \eta_\xi$ means more local samples $T(\eta_\xi)$ are needed (see for instance the bound for *Successive Elimination*). In contrast, if the approximation provided in Proposition 5.3 is not enough to conclude that the sample complexity increases with η_ξ (because of the first parameter in Q^{-1}), it shows at least an ambiguity. Indeed, a smaller η_ξ means a smaller voting threshold $M = \lceil \frac{\log \delta}{\log \eta_\xi} \rceil$, which is likely to decrease T *via* the second argument in Q^{-1}. An extended discussion is provided in the appendix.

6. Experiments

We assess the performance of our methods on different stochastic environments suggested in Féraud et al. (2019). In these problems, there are $K = 10$ arms with means $\mu_1 \geq \ldots \geq \mu_{10}$. We display our results on **Problem 1**, where rewards are drawn from Bernoulli distributions, with $\mu_1 = 0.7$, $\mu_2 = 0.5$, $\mu_3 = 0.3$, and $\mu_k = 0.1$ for $k = 4, \ldots, 10$. Agent distribution $P_{\mathcal{N}}$ is uniform. Further experiments, including other problems and agent distributions, are discussed in the appendix. We run experiments with N ranging from 64 to 1024 agents to evaluate how performance is affected by the scale of the problem. For both methods, we fix the global failure probability δ to 5%. Each data point is an average value over 10 runs, and 95% confidence intervals are shown for every plot.

Evaluation and Baselines. We evaluate the performance of our algorithms for different values of the privacy parameters (ϵ for DP-MASE, η and ξ for CORRUPTED ELIMINATION). In addition, we compare them with two baselines, corresponding to the extreme cases with full and no communication.

- CENTRAL: the agents send all their raw reward samples with the central coordinator at each round, which is equivalent (up to negligible communication delays) to a single-agent SUCCESSIVE ELIMINATION. This approach does not offer any privacy since data is fully shared;
- INDEPENDENT: every agent runs an independent SUCCESSIVE ELIMINATION algorithm, without sharing any information with the central coordinator. This approach is fully private as there is no communication.

They allow to illustrate both the gain in performance achieved through multi-agent collaboration and the challenges of distributing data across multiple nodes. To measure performance, we report the sample complexity, which is the total number of rounds (corresponding to a single iteration of the while loop in Algorithms 1, 2 and 3) needed to output the best arm with confidence $1 - \delta$.

6.1. Impact of Multi-Agent Collaboration

Figure 1 shows the performance of our methods against the CENTRAL and INDEPENDENT baselines. We observe that the BAI problem is solved much faster if agents can freely communicate their raw reward samples, which amounts to the single-agent case. This is expected, as the problem is inevitably harder when the data is distributed to multiple locations. In particular, the sample complexity must scale with the number of agents, as already discussed in Section 3.3. We indeed observe a growth in sample complexity with the number of agents for both DP-MASE and CORRUPTED ELIMINATION. However, they scale better than the naive INDEPENDENT baseline, which shows that the communication between agents and the collaborative elimination procedure help to be more efficient and attenuate the growth in sample complexity. Figure 4 in appendix shows the same results without the baselines for a better distinction between the different curves.

6.2. Multi-Agent Successive Elimination with Differentially Private Communications

Our experiments exhibit the influence of the privacy level ϵ on the performance of DP-MASE (Figure 1, left). Unsurprisingly, for a fixed global confidence $1 - \delta$, the sample complexity gets worse with stronger privacy guarantees, i.e., smaller ϵ's. Longer epochs and larger gap thresholds for local elimination are indeed needed to compensate for the noisier mean estimates. We notice a significant increase in sample complexity for small values of ϵ (less than 0.1), while it is relatively stable for larger values.

6.3. Multi-Agent Successive Elimination with (ϵ, η)-Private Communications

We compare the sample complexity of CORRUPTED ELIMINATION against the method proposed by Féraud et al. (2019) (corresponding to $\xi = 0$).

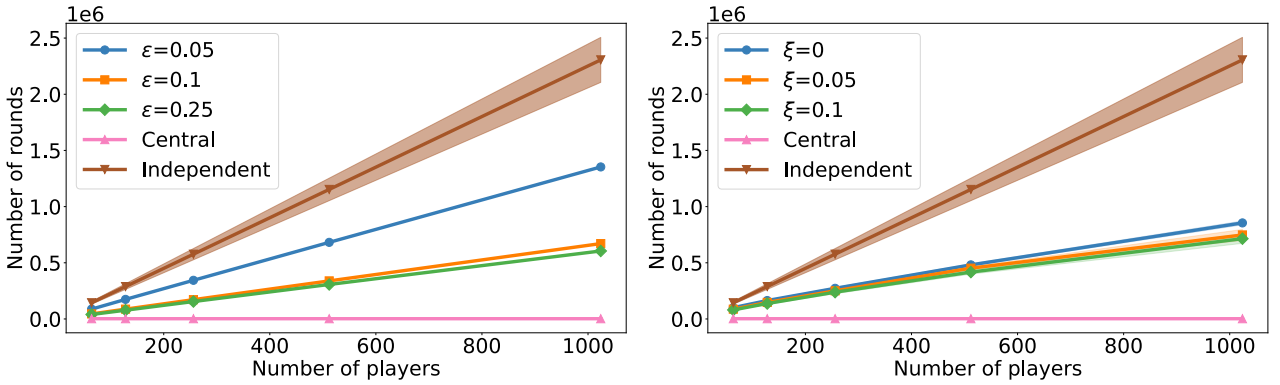

Figure 1. Sample complexity of DP-MASE (left) for $\epsilon = 0.05, 0.1, 0.25$ and CORRUPTED ELIMINATION (right) for $\xi = 0, 0.05, 0.1$ on **Problem 1**.

The apparent privacy level η is set to 0.9. In accordance with Proposition 5.1, for each value of ξ, we compute $\eta_\xi = 1 - \frac{1-\eta}{(1-\xi)^{K-1}}$, and then run local BAI subroutines with confidence $1 - \eta_\xi$, which guarantees the same apparent privacy level and hence a fair comparison.

We observe that corrupting messages actually decreases the sample complexity of CORRUPTED ELIMINATION, and that it tends to further decrease with higher corruption probability ξ (Figure 1, right). In particular, we improve over the method proposed by Féraud et al. (2019). Our intuition is that a smaller η_ξ is likely to speed up the global elimination process, and hence the algorithm, by decreasing the voting threshold M. This was already suggested by our theoretical analysis in Proposition 5.3. We refer to this as the *threshold effect*, and empirical evidence therefore points toward its prevalence over the increase in the number of local samples. Similarly to DP-MASE, we also find that the sample complexity increases as the apparent privacy level η grows, as shown in the appendix. We interpret it as a direct consequence of the threshold effect.

7. Discussion

In this work, we have proposed two methods to address multi-agent best arm identification, where multiple participants work together to identify with fixed confidence the best option in a common multi-armed bandit. Based on the same successive elimination scheme, each method guarantees the privacy of a type of data against adversaries observing the communications. DP-MASE ensures that the communications are differentially private with respect to the rewards, while CORRUPTED ELIMINATION prevents the adversaries from inferring the best arm. We provide theoretical analysis on privacy, sample complexity, and failure probability, and experiments on toy problems give insights into the behavior of our methods. In particular, they show that CORRUPTED ELIMINATION is faster than a similar method without message corruption, for comparable privacy levels.

Direct comparison between our two methods is impossible, as they rely on distinct privacy definitions. However, they are suitable to address complementary privacy concerns in multi-agent best arm identification. Also, since our primary goal was to investigate and address different privacy needs within the relatively new setting of collaborative BAI, our methods could be refined to achieve better stopping time. Designing more efficient approaches thus is a natural direction for future work. Relaxing the assumption regarding the nature of communications, and allowing agents to send more informative messages (like continuous signals), is also worth investigating and certainly more challenging in terms of privacy. From a broader perspective, we believe our approach can be applied to the BAI with a fixed budget setting, to linear bandits, or to more advanced multi-agent scenarios, e.g., considering agents with heterogeneous arm sets.

Acknowledgements

M.S. has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003) and INODE project funded by EU Horizon 2020 research and innovation programme under GA No 863410.

References

Agarwal, M., Aggarwal, V., and Azizzadenesheli, K. Multi-Agent Multi-Armed Bandits with Limited Communication. *Journal of Machine Learning Research*, 23:1–24, 2021. URL https://www.jmlr.org/papers/volume23/21-138/21-138.pdf.

Audibert, J.-Y. and Bubeck, S. Best Arm Identification in Multi-Armed Bandits. In *Proceedings of COLT*, 2010. URL https://hal-enpc.archives-ouvertes.fr/hal-00654404.

Chawla, R., Sankararaman, A., Ganesh, A., and Shakkottai, S. The Gossiping Insert-Eliminate Algorithm for Multi-Agent Bandits. In *Proceedings of AISTATS*, 2020. URL http://proceedings.mlr.press/v108/chawla20a/chawla20a.pdf.

Ciucanu, R., Lafourcade, P., Marcadet, G., and Soare, M. SAMBA: A Generic Framework for Secure Federated Multi-Armed Bandits. *Journal of Artificial Intelligence Research*, 73:737–765, 2022. URL https://www.jair.org/index.php/jair/article/download/13163/26774/.

DasGupta, A. *Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics*. Springer Texts in Statistics. Springer, 2011. URL https://link.springer.com/book/10.1007/978-1-4419-9634-3.

Dubey, A. and Pentland, A. Private and Byzantine-Proof Cooperative Decision-Making. In *Proceedings of AAMAS*, 2020. URL https://www.ifaamas.org/Proceedings/aamas2020/pdfs/p357.pdf.

Dwork, C. Differential Privacy. In *Proceedings of ICALP*, 2006. URL https://www.microsoft.com/en-us/research/publication/differential-privacy/.

Dwork, C. and Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014. URL https://doi.org/10.1561/0400000042.

Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and Vadhan, S. On the Complexity of Differentially Private Data Release: Efficient Algorithms and Hardness Results. In *Proceedings of STOC*, 2009. URL https://doi.org/10.1145/1536414.1536467.

Even-Dar, E., Mannor, S., and Mansour, Y. PAC bounds for multi-armed bandit and markov decision processes. In *Proceedings of COLT*, 2002. URL https://doi.org/10.1007/3-540-45435-7_18.

Even-Dar, E., Mannor, S., and Mansour, Y. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 2006. URL https://jmlr.org/papers/volume7/evendar06a/evendar06a.pdf.

Féraud, R., Alami, R., and Laroche, R. Decentralized Exploration in Multi-Armed Bandits. In *Proceedings of ICML*, 2019. URL http://proceedings.mlr.press/v97/feraud19a/feraud19a.pdf.

Gabillon, V., Ghavamzadeh, M., Lazaric, A., and Bubeck, S. Multi-Bandit Best Arm Identification. In *Proceedings of NeurIPS*, 2011. URL https://proceedings.neurips.cc/paper/2011/file/c4851e8e264415c4094e4e85b0baa7cc-Paper.pdf.

Garcelon, E., Pirotta, M., and Perchet, V. Encrypted Linear Contextual Bandit. In *Proceedings of AISTATS*, 2022. URL https://proceedings.mlr.press/v151/garcelon22a.html.

Hillel, E., Karnin, Z., Koren, T., Lempel, R., and Somekh, O. Distributed Exploration in Multi-Armed Bandits. In *Proceedings of NeurIPS*, 2013. URL https://proceedings.neurips.cc/paper/2013/file/598b3e71ec378bd83e0a727608b5db01-Paper.pdf.

Jamieson, K. and Nowak, R. Best-arm Identification Algorithms for Multi-Armed Bandits in the Fixed Confidence Setting. In *Proceedings of CISS*, 2014. URL https://ieeexplore.ieee.org/document/6814096.

Jha, A., Mohamed, N., and Jagannathan, K. Collaborative Best Arm Identification in Multi-armed Bandits. In *Proceedings of COMSNETS*, 2022. URL https://ieeexplore.ieee.org/document/9668527.

Kumar Kolla, R., Jagannathan, K., and Gopalan, A. Collaborative Learning of Stochastic Bandits Over a Social Network. *Transactions on Networking*, 26(4):1782–1795, 2018. URL https://www.ee.iitm.ac.in/~krishnaj/Publications_files/papers/08418308.pdf.

Li, T. and Song, L. Privacy-Preserving Communication-Efficient Federated Multi-Armed Bandits. *IEEE Journal on Selected Areas in Communications*, 40(3):773–787, 2022. URL https://doi.org/10.1109/JSAC.2022.3142374.

Madhushani, U. and Leonard, N. It Doesn't Get Better and Here's Why: A Fundamental Drawback in Natural Extensions of UCB to Multi-agent Bandits. In *Proceedings of "I Can't Believe It's Not Better!" NeurIPS 2020 workshop*, 2020. URL https://openreview.net/forum?id=eK034ngO05Y.

McSherry, F. D. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, 2009. URL https://doi.org/10.1145/1559845.1559850.

Ren, W., Zhou, X., Liu, J., and Shroff, N. B. Multi-Armed Bandits with Local Differential Privacy. *CoRR*, 2020. URL https://arxiv.org/abs/2007.03121.

Robbins, H. E. Some Aspects of the Sequential Design of Experiments. *Bulletin of American Mathematical Society*, 58(5):527–535, 1952. URL https://www.ams.org/journals/bull/1952-58-05/S0002-9904-1952-09620-8/S0002-9904-1952-09620-8.pdf.

Réda, C., Vakili, S., and Kaufmann, E. Near-Optimal Collaborative Learning in Bandits. In *Proceedings of NeurIPS*, 2022. URL https://arxiv.org/abs/2206.00121.

Sajed, T. and Sheffet, O. An Optimal Private Stochastic-MAB Algorithm based on Optimal Private Stopping Rule. In *Proceedings of ICML*, 2019. URL https://proceedings.mlr.press/v97/sajed19a.html.

Sankararaman, A., Ganesh, A., and Shakkottai, S. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35, 2019. URL https://doi.org/10.1145/3366701.

Scarlett, J., Bogunovic, I., and Cevher, V. Overlapping Multi-Bandit Best Arm Identification. In *Proceedings of ISIT*, 2019. URL http://ilijabogunovic.com/pdf/scarlett2019multibandit_ISIT2019.pdf.

Shi, C., Xu, H., Xiong, W., and Shen, C. (Almost) Free Incentivized Exploration from Decentralized Learning Agents. In *Proceedings of NeurIPS*, 2021. URL https://proceedings.neurips.cc/paper/2021/file/054ab897023645cd7ad69525c46992a0-Paper.pdf.

Szörényi, B., Busa-Fekete, R., Hegedüs, I., Ormándi, R., Jelasity, M., and Kégl, B. Gossip-based Distributed Stochastic Bandit Algorithms. In *Proceedings of ICML*, 2013. URL http://proceedings.mlr.press/v28/szorenyi13.pdf.

Thompson, W. R. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285–294, 1933. URL http://www.jstor.org/stable/2332286.

Tossou, A. and Dimitrakakis, C. Algorithms for Differentially Private Multi-Armed Bandits. In *Proceedings of AAAI*, 2016. URL https://aaai.org/papers/212-algorithms-for-differentially-private-multi-armed-bandits/.

A. Proofs

A.1. Multi-Agent Successive Elimination with Differentially Private Communications

Proposition 4.1. *Algorithm 2,* DP-MASE*, is ϵ-differentially private for communications.*

Proof. Let $D(n, e^n)$ be the sequence of all rewards collected by agent n at epoch e^n. Fix global step t, and denote $\mathcal{T}^n$ the set of agent n's local epochs occurring up to time t. Therefore $D_{1:t} = \bigcup_{n\in\mathcal{N}, e^n\in\mathcal{T}^n} D(n, e^n)$.

Let also $\tilde{\mathcal{Q}}_{e^n}$ be the mechanism that outputs the indices of the arms eliminated by agent n at epoch e^n. Note that an agent n has knowledge of the latest globally eliminated arms only at the end of his current epoch, so that the reward samples in $D(n, e^n)$ has no effect on the local elimination process for any other agent m. Thus, $\tilde{\mathcal{Q}}_{e^n}$ operates only on $D(n, e^n)$. In other words, each reward sample only plays a role for a single agent, at a single epoch.

At epoch e^n, for every arm k in the active set $\mathcal{K}^n$, the empirical mean $\hat{\mu_k}(e^n)$ is computed using $R(e^n)$ samples. The sensitivity of $\hat{\mu_k}(e^n)$ is therefore $1/R(e^n)$. Moreover, each noisy mean is computed only once before suboptimality evaluations. Therefore, the local elimination mechanism which uses Laplace noise with scale $\frac{1}{\epsilon R(e^n)}$ is ϵ-DP. $\tilde{\mathcal{Q}}_{e^n}$ is thus ϵ-DP.

Consider the composed mechanism $\tilde{Q}$ defined as:

$$\tilde{Q}(D_{1:t}) = \left(\tilde{\mathcal{Q}}_{e^n}(D(n, e^n))\right)_{n\in\mathcal{N}, e^n\in\mathcal{T}^n} \quad .$$

Thanks to the parallel composition theorem (see Theorem 4 in McSherry (2009)), $\tilde{Q}$ is ϵ-DP. However, $\mathcal{Q}_n$ can be seen as a deterministic function of $\left(\tilde{\mathcal{Q}}_{e^n}\right)_{n,e^n}$, i.e. $\mathcal{Q}_n = g\left(\left(\tilde{\mathcal{Q}}_{e^n}\right)_{n,e^n}\right)$. So $\mathcal{Q}_n$ is also ϵ-DP by the post-processing property (see Proposition 2.1 in Dwork & Roth (2014)). □

Proposition 4.2. *(*Failure Probability of DP-MASE*) With voting threshold $M = \lceil \frac{\log\delta}{\log\beta} \rceil$,* DP-MASE *fails to return the best arm with probability at most δ.*

Proof. Sajed & Sheffet (2019) show that using DP Successive Elimination, the optimal arm $k^\star$ is never eliminated with probability at least $1-\beta$. The probability of eliminating $k^\star$ is therefore at most β.

To eliminate $k^\star$ globally, $M = \lceil \frac{\log\delta}{\log\beta} \rceil$ independent agents need to eliminate $k^\star$. Since the agents interact independently with the system, $k^\star$ is then globally eliminated with probability at most β^M. But, $\beta^M = \beta^{\lceil \frac{\log\delta}{\log\beta} \rceil} \leq \beta^{\frac{\log\delta}{\log\beta}} = \delta$. □

Proposition 4.3. *(*Sample Complexity of DP-MASE*) With probability at least $(1-\delta)\left(1 - I_{1-p^\star}\left(T - T(\beta), 1 + T(\beta)\right)\right)^M$,* DP-MASE *has sample complexity:*

$$\mathcal{O}\left(\tfrac{1}{p^\star}T(\beta) + \tfrac{1}{4p^{\star 2}}\log\tfrac{1}{\delta}\right) \quad ,$$

with:

$$T(\beta) = \log(K/\beta) + \log\log(1/\Delta_2)\left(\tfrac{1}{\Delta_2^2} + \tfrac{1}{\epsilon\Delta_2}\right) \quad .$$

Above, $I_p\left(a, b\right)$ is the regularized incomplete beta function evaluated at p with parameters (a, b), and $p^\star$ is the probability of the M-th most likely agent.

To prove Proposition 4.3, we first prove the following lemma, which is an updated version of the sample complexity bound from Féraud et al. (2019).

Lemma A.1. *(*Sample complexity of Féraud et al. (2019)*) Using any BAI subroutine, with probability at least $(1-\delta)\left(1 - I_{1-p^\star}\left(T - T(\eta), 1 + T(\eta)\right)\right)^M$, the multi-agent BAI algorithm proposed in Féraud et al. (2019) has sample complexity*

$$\mathcal{O}\left(\tfrac{1}{p^\star}T(\eta) + \tfrac{1}{4p^{\star 2}}\log\tfrac{1}{\delta}\right) \quad ,$$

where $I_p(a,b)$ is the regularized incomplete beta function evaluated at p with parameters (a,b), $p^\star$ is the probability of the M-th most likely agent, and $T(\eta)$ is the number of local samples needed for the BAI subroutine to find the optimal arm with confidence $1-\eta$.

Proof. We follow a similar line of reasoning to Féraud et al. (2019), for the derivation of the sample complexity bound.

Let T denote the number of rounds before the algorithm stops (*i.e.* its sample complexity) and T_n denote the number of rounds agent n has been active. $P_{\mathcal{N}}$ is the probability distribution over agents, so that $P_{\mathcal{N}}(x=n)$ is the probability that the active player x is n at any time. We have $T_n \sim \mathcal{B}(T, P_{\mathcal{N}}(x=n))$, and thus $\mathbb{E}_{P_{\mathcal{N}}}[T_n] = P_{\mathcal{N}}(x=n)T$.

Let $B_{\delta,\eta}$ be the set of agents with the $M := \lceil \frac{\log\delta}{\log\eta} \rceil$ highest T_n. The algorithm does not stop if the following event occurs: $E_1 = \{\exists n \in B_{\delta,\eta}, T_n < T(\eta)\}$.

Applying Hoeffding's inequality, we have:

$$\mathbb{P}\left(T_n - P_{\mathcal{N}}(x=n)T \leq -\epsilon\right) \leq \exp\left(-2\epsilon^2/T\right) \quad . \tag{2}$$

When $\neg E_1$ occurs, all agents in $B_{\delta,\eta}$ have enough samples to output the optimal arm with confidence $1-\eta$, i.e. $\forall n \in B_{\delta,\eta}$, $T_n \geq T(\eta)$. Thus, with probability at most δ:

$$T(\eta) - P_{\mathcal{N}}(x=n)T \leq T_n - P_{\mathcal{N}}(x=n)T \leq -\sqrt{\frac{T}{2}\log\frac{1}{\delta}} \quad . \tag{3}$$

This holds for every agent $n \in B_{\delta,\eta}$, and in particular for n such that $P_{\mathcal{N}}(x=n') = \min_{n\in B_{\delta,\eta}} P_{\mathcal{N}}(x=n) := p_{\delta,\eta}$.

Therefore, the total number of samples T is such that:

$$p_{\delta,\eta}T - \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\sqrt{T} - T(\eta) \geq 0 \quad , \tag{4}$$

which is a second order polynomial equation in $\sqrt{T}$. Solving this equation in T, we have with probability at least $1-\delta$:

$$T \geq \frac{1}{4p_{\delta,\eta}^2}\left(\sqrt{\frac{1}{2}\log\frac{1}{\delta}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta} + 4p_{\delta,\eta}T(\eta)}\right)^2 \quad . \tag{5}$$

Developing and re-ordering the terms, we have:

$$T \geq \frac{1}{p_{\delta,\eta}}T(\eta) + \frac{1}{4p_{\delta,\eta}^2}\left(\log\frac{1}{\delta} + \sqrt{\log^2\frac{1}{\delta} + 8\log\frac{1}{\delta}p_{\delta,\eta}T(\eta)}\right) \quad . \tag{6}$$

In particular:

$$T \geq \frac{1}{p_{\delta,\eta}}\left(T(\eta) + \frac{1}{4p_{\delta,\eta}}\log\frac{1}{\delta}\right) \quad . \tag{7}$$

Therefore, when E_1 does not occur, that is when all M most frequent agents have enough samples to output best arm with confidence η, with probability at least $1-\delta$:

$$T \leq \frac{1}{p_{\delta,\eta}}\left(T(\eta) + \frac{1}{4p_{\delta,\eta}}\log\frac{1}{\delta}\right) \quad . \tag{8}$$

Let now $\mathcal{N}_M$ be the set of the M most likely agents and define $n^\star = \arg\min_{n\in\mathcal{N}_M} P_{\mathcal{N}}(x=n)$ and $p^\star = \operatorname{argmin}_{n\in\mathcal{N}_M} P_{\mathcal{N}}(x=n)$ as the index and the probability of the M-th most likely agent, respectively.

We now consider the following event: $E_2 = \{n^\star \notin \mathcal{B}_{\delta,\eta}\}$. Under $\neg E_1$, by definition of $\mathcal{B}_{\delta,\eta}$, E_2 is equivalent to the event $\{T_{n^\star} < T(\eta)\}$.

But:

$$\mathbb{P}\left(E_2\right) = \mathbf{I}_{1-p^\star}\left(T - T(\eta), 1 + T(\eta)\right) \quad . \tag{9}$$

In order to have exactly $p_{\delta,\eta} = p^\star$, an equivalent formulation of $\neg E_2$ should hold for all agents whose sampling probability $P_{\mathcal{N}}(x = n)$ is greater than $p^\star$. This is why $p_{\delta,\eta} = p^\star$ with probability $\left(1 - \mathbf{I}_{1-p^\star}\left(T - T(\eta), 1 + T(\eta)\right)\right)^M$.

□

Now, we prove Proposition 4.3 using Lemma A.1.

Proof. From Sajed & Sheffet (2019) (Lemma 4.2), the sample complexity of *DP Successive Elimination* is:

$$T(\beta) = \log(K/\beta) + \log\log(1/\Delta_2)\left(\frac{1}{\Delta_2^2} + \frac{1}{\epsilon\Delta_2}\right) \quad .$$

Using the same global elimination mechanism, we can see DP-MASE as the multi-agent algorithm from Féraud et al. (2019) with *DP Successive Elimination* as a BAI subroutine. We can therefore apply Lemma A.1 with local sample complexity $T(\beta)$ and conclude that the sample complexity of DP-MASE is:

$$\mathcal{O}\left(\frac{1}{p^\star}T(\beta) + \frac{1}{4p^{\star 2}}\log\frac{1}{\delta}\right)$$

with probability at least $(1-\delta)\left(1 - I_{1-p^\star}\left(T - T(\beta), 1 + T(\beta)\right)\right)^M$.

□

A.2. Multi-Agent Successive Elimination with (ϵ, η)-Private Communications

Proposition 5.1. *(*Privacy of CORRUPTED ELIMINATION*) Running local BAI subroutines with confidence $1 - \eta'$, an adversary knowing $\mathcal{A}$ and observing $\mathcal{M}_n^\xi$ cannot infer the best arm of agent n with probability higher than $(1 - \eta') \times (1 - \xi)^{K-1}$. Therefore, to maintain an apparent privacy level η, we need to set the confidence level of the BAI subroutine as:*

$$\eta_\xi = \max\left(0, 1 - \frac{1-\eta}{(1-\xi)^{K-1}}\right) \leq \eta \quad .$$

Proof. Let us recall the definition of $\mathcal{M}_n^\xi$:

$$\mathcal{M}_n = (\lambda_1^n, ..., \lambda_K^n) \quad , \tag{10}$$

$$\mathcal{M}_n^\xi = (\lambda_1^n \times \nu_1^n, ..., \lambda_k^n \times \nu_K^n) \quad , \tag{11}$$

where, for any arm k, ν_k^n is the corruption mask and is a random variable with distribution $\mathcal{B}(1-\xi)$.

An adversary can guess the global optimal arm from messages $\mathcal{M}_n$ if these two events hold:

- Event A: the arm selection subroutine of agent n actually finds the optimal arm;
- Conditionally to A, event B: no message associated to sub-optimal arms is corrupted.

By assumption, $\mathbb{P}(A) \geq 1 - \eta_\xi$. Moreover, B holds when all ν_k^n are ones for $k = 2, ..., K$. Since $\nu_k^n \sim \mathcal{B}(1-\xi)$, we have $\mathbb{P}(B) = (1-\xi)^{K-1}$. Therefore, observing messages $\mathcal{M}_n$, an external observer can only infer the best arm with probability $(1 - \eta_\xi) \times (1-\xi)^{K-1}$.

To maintain an apparent privacy level η, we then need to set $\eta_\xi > 0$ such as:

$$1 - \eta = (1 - \eta_\xi) \times (1-\xi)^{K-1} \quad . \tag{12}$$

That is:

$$\eta_\xi = \max\left(0, 1 - \frac{1-\eta}{(1-\xi)^{K-1}}\right) \leq \eta \ . \tag{13}$$

□

Proposition 5.2. *(*Failure probability of CORRUPTED ELIMINATION*) The failure probability of* CORRUPTED ELIMINATION *is at most:*

$$\min_{\alpha\in\mathbb{N}} \left(1 - I_\xi\left(\alpha+1, M\right)\right) \times \delta_\alpha \ \text{, where } \delta_\alpha = \delta \times {\eta_\xi}^\alpha \ .$$

Proof. Assuming it terminates, CORRUPTED ELIMINATION fails, i.e. eliminates the best arm, if there are at least $M = \lceil \frac{\log\delta}{\log\eta} \rceil$ votes against it. Let N_e and N_v denote respectively the number of local independent eliminations of the best arm and the number of votes actually sent against the best arm. N_v follows a binomial distribution with N_e trials:

$$N_v \sim \mathcal{B}\left(N_e, 1-\xi\right) \ . \tag{14}$$

Given N_e independent eliminations, the probability that N_v exceeds the voting threshold M is:

$$\mathbb{P}\left(N_v \geq M\right) = \sum_{m=M}^{N_e} \binom{N_e}{m} (1-\xi)^m \xi^{N_e - m} \ . \tag{15}$$

In order to overcome the randomness of N_e, we want $\alpha \in \mathbb{N}$ such that if $N_e \geq M+\alpha$, $N_v \geq M$ with high probability. As $N_e \geq M+\alpha$, obviously, the probability of a random variable X following $\mathcal{B}\left(M+\alpha, 1-\xi\right)$ being greater than M is lower than if it followed $\mathcal{B}\left(N_e, 1-\xi\right)$. Thus:

$$\begin{aligned}
\mathbb{P}\left(N_v \geq M\right) &\geq \sum_{m=M}^{M+\alpha} \binom{M+\alpha}{m} (1-\xi)^m \xi^{M+\alpha-m} && (16)\\
&= 1 - \sum_{m=0}^{M-1} \binom{M+\alpha}{m} (1-\xi)^m \xi^{M+\alpha-m} && (17)\\
&= 1 - I_\xi\left(\alpha+1, M\right) \ , && (18)
\end{aligned}$$

where $I_p\left(a,b\right)$ is the regularized incomplete beta function evaluated at p with parameters (a,b).

Now since the local BAI routines are run with confidence η_ξ, the probability that at least $M+\alpha$ independent agents eliminate the best arm is at most $\eta_\xi^{M+\alpha}$.

Eventually, for $\alpha \in \mathbb{N}$, CORRUPTED ELIMINATION eliminates the best arm if the two following events happen:

- $M+\alpha$ independent agents eliminate the best arm locally, which happens with probability at most $\eta_\xi^{M+\alpha}$;
- These $M+\alpha$ local eliminations imply at least M votes, which happens with probability $1 - I_\xi\left(\alpha+1, M\right)$.

By independence of the corruption mechanism, the failure probability of the algorithm is at most:

$$\min_{\alpha\in\mathbb{N}} \left(1 - I_\xi\left(\alpha+1, M\right)\right) \times \eta_\xi^{M+\alpha} \ . \tag{19}$$

Noticing that:

$$\eta_\xi^{M+\alpha} = \eta_\xi^{\lceil \frac{\log\delta}{\log\eta} \rceil + \alpha} \leq \delta \times \eta_\xi^\alpha = \eta_\xi^{\frac{\log\delta}{\log\eta} + \alpha} \ , \tag{20}$$

and setting $\delta_\alpha = \delta \times {\eta_\xi}^\alpha$ concludes the proof.

□

Proposition 5.3. *(*Sample complexity of CORRUPTED ELIMINATION*) If the agent distribution $P_{\mathcal{N}}$ is uniform, then with probability α, the number of rounds needed by* CORRUPTED ELIMINATION *to output an ϵ-optimal arm is approximately:*

$$T = N \times Q^{-1}\left(T(\eta_\xi), 1-\alpha^{1/M}\right) \ ,$$

where Q^{-1} is the inverse of the regularized gamma function, and $T(\eta_\xi)$ is the number of local samples needed for the BAI subroutine to find the optimal arm with confidence $1-\eta_\xi$.

Proof. For all $n \in \mathcal{N}$, let T_n denote the number of rounds agent n has been active, and T denote the total number of rounds before the algorithm stops. Then $(T_1, ..., T_N)$ follows a multinomial distribution:

$$(T_1, ..., T_N) \sim \text{Mult}(T, \{p_1, ..., p_N\}) \ , \tag{21}$$

where $p_n = P_{\mathcal{N}}(x = n)$. For simplicity, we consider a uniform distribution, i.e. $p_n = 1/N$ for any $n \in \mathcal{N}$, in the sequel of this proof.

Assuming the local BAI subroutines accurately return the best arm, the algorithm terminates when the M largest $T_1, ..., T_N$ exceed threshold $T(\eta_\xi)$. We thus look for a bound on the probability of this event.

We can make the components of the vector $(T_1, ..., T_N)$ independent by "poissonizing" the multinomial, using the following theorem (DasGupta, 2011):

Theorem A.2. *Let $N \sim \mathcal{P}(\lambda)$ and suppose $(X_1, ..., X_k) \sim$ Multi$(p_1, ..., p_k)$. Then, marginally, $X_1, ..., X_K$ are independent Poisson random variables, with $X_i \sim X_i \sim \mathcal{P}(p_i\lambda)$.*

Instead of considering T, the number of trials of the multinomial, as fixed, we assume that it follows a Poisson distribution, i.e., $T \sim \mathcal{P}(\overline{T})$ for some $\overline{T} \in \mathbb{N}$. Thus, we can now think of $T_1, ..., T_N$ as mutually independent random variables such that $T_n \sim \mathcal{P}\left(\overline{T}/N\right)$ for all $n \in \mathcal{N}$.

The probability that one count T_n is greater than $T(\eta_\xi)$ can then be expressed in terms of the cumulative distribution function of the Poisson distribution:

$$\mathbb{P}(T_n \geq T(\eta_\xi)) = 1 - Q\left(T(\eta_\xi), \overline{T}/N\right) \ , \tag{22}$$

where $Q(s, x)$ is the regularized gamma function evaluated in (s, x).

Consequently, by independence of the $T_1, ..., T_N$, the probability that M different agents have more samples than $T(\eta_\xi)$ at time t is, for some $\alpha \in [0, 1]$:

$$\left(1 - Q\left(T(\eta_\xi), \overline{T}/N\right)^M = \alpha \ .$$

Now, to estimate the sample complexity T, we may just compute $\overline{T}$ given α, M and $T(\eta_\xi)$. To do this, we notice that:

$$Q(T(\eta_\xi), \overline{T}/N) = 1 - \alpha^{1/M} \ .$$

There is no close form for the inverse of Q that would lead to a close form for T. However, we can numerically compute the inverse of Q in its second argument. Thus, the total number of samples would be given by:

$$\overline{T} \approx N \times Q^{-1}\left(T(\eta_\xi), 1-\alpha^{1/M}\right) \ .$$

Confusing T with its expectation $\overline{T}$, we get an approximation of the sample complexity of CORRUPTED ELIMINATION.

We can analyze this approximation in the following way. When η_ξ increases, $M = \lceil \frac{\log \delta}{\log \eta_\xi} \rceil$ increases, and therefore $\alpha^{1/M}$ increases (as $0 < \alpha < 1$). Empirically, we also observe that Q^{-1} is a decreasing function in its second argument. Therefore, although the final effect is unclear due to the presence of $T(\eta_\xi)$ as a first argument, the sample complexity possibly increases with η_ξ *via* its second argument.

□

B. Algorithms

In this section, for self-completeness, we recall the main algorithms of the paper and introduce the elimination subroutines that have been omitted because of space constraints.

B.1. Multi-Agent Successive Elimination with Differentially Private Communications

We recall the algorithm for DP-MASE and introduce in Algorithm 4 the elimination subroutine used at line 7 of Algorithm 2.

Algorithm 2 DP-MASE

Input: ϵ, δ, β, $M = \lceil \frac{\log \delta}{\log \beta} \rceil$
Output: Estimated best arm in $\mathcal{K}$
Active agents $\mathcal{N}(t) = \mathcal{N}$; global and local active sets $\mathcal{K} = \mathcal{K}^n = [K]$; global votes $(\lambda_k = 0)_{k\in[K]}$; local epochs $e^n = 0$; local round $r^n = 0$; global time $t = 0$
while $|\mathcal{K}| > 1$ **do**
  Active agent n is sampled: $n \sim P_{\mathcal{N}(t)}$
  Sample all arms in $\mathcal{K}^n$ and update mean estimates
  $r^n := r^n + 1$
  if $r^n = R(e^n)$ **then**
    $\bar{k}(e^n) =$ LocalElimination $(\epsilon, e^n, (\hat{\mu}_k^n(e^n))_k, \beta)$ {Identify arms to eliminate locally}
    $\mathcal{K}^n = \mathcal{K}^n \setminus \bar{k}(e^n)$ {Remove eliminated arms from local active set}
    $\lambda_k := \lambda_k + 1$ for any $k \in \bar{k}(e^n)$ {Vote against eliminated arms for global elimination}
    if $|\mathcal{K}^n| = 1$ **then**
      $\mathcal{N}(t) := \mathcal{N}(t) \setminus \{n\}$ {Remove agent if done}
    else
      $e^n := e^n + 1$
      Reset $(\hat{\mu}_k^n(e^n) = 0)_k$, $r^n = 0$
      Compute next epoch length $R(e^n)$
      Synchronize $\mathcal{K}^n := \mathcal{K}^n \bigcap \mathcal{K}$ {Remove the latest globally eliminated arms from local active set}
    for $k \in \bar{k}(e^n)$ **do**
      $\mathcal{K} := \mathcal{K} \setminus \{k\}$ **if** $\lambda_k > M$ {Remove arms from global active set if the voting threshold is met}
  t := t + 1

Algorithm 4 DP-MASE Local Elimination

Input: Privacy level ϵ, current local epoch e^n, estimated means $(\hat{\mu}_k^n(e^n))_{\mathcal{K}^n}$, local accuracy β
Output: A set of arms $\bar{k}(e^n)$ to eliminate locally at agent n.
$\bar{k}(e^n) = \{\}$
Set $h(e^n) = \sqrt{\frac{\log(8|\mathcal{K}^n(e^n)|e^{n\,2}/\beta)}{2R(e^n)}}$
Set $c(e^n) = \frac{\log\left(4|\mathcal{K}^n(e^n)|e^{n\,2}/\beta\right)}{\epsilon R(e^n)}$
Perturb means: $\tilde{\mu}_k^n = \hat{\mu}_k^n(e^n) + \text{Lap}\left(1/\epsilon R(e^n)\right)$
for all $k \in \mathcal{K}^n$ **do**
  if $\max_{l\in\mathcal{K}^n} \tilde{\mu}_l^n - \tilde{\mu}_k^n > 2\left(h(e^n) + c(e^n)\right)$ **then**
    $\bar{k}(e^n) := \bar{k}(e^n) \cup \{k\}$ {Add k to the set of arms to eliminate if elimination condition is met}

B.2. Multi-Agent Successive Elimination with (ϵ, η)-Private Communications

We recall the algorithm for CORRUPTED ELIMINATION.

Algorithm 3 CORRUPTED ELIMINATION

Input: η, δ, ξ, $M = \lceil \frac{\log \delta}{\log \eta} \rceil$
Output: Estimated best arm in $\mathcal{K}$
Active agents $\mathcal{N}(t) = \mathcal{N}$; global and local active sets $\mathcal{K} = \mathcal{K}^n = [K]$; local failure probability $\eta_\xi = 1 - \frac{1-\eta'}{(1-\xi)^{K-1}}$, $\mathcal{K}^n := \mathcal{K}$ for all $n \in \mathcal{N}$; global votes $(\lambda_k = 0)_{k\in[K]}$, global time $t = 0$
while $|\mathcal{K}| > 1$ **do**
  Active agent n is sampled: $n \sim P_{\mathcal{N}(t)}$
  Sample all arms in $\mathcal{K}^n$ and update mean estimates
  $\bar{k}, , \bar{k}_G =$ LocalElimination (n, ξ) {Identify arms to eliminate locally}
  $\mathcal{K}^n = \mathcal{K}^n \setminus \bar{k}$ {Remove eliminated arms from local active set}
  $\lambda_k := \lambda_k + 1$ for any $k \in \bar{k}_G(e^n)$ {Vote for global elim. of eliminated arms if corresponding message not corrupted}
  if $|\mathcal{K}^n| = 1$ **then**
    $\mathcal{N}(t) := \mathcal{N}(t) \setminus \{n\}$
  Synchronize $\mathcal{K}^n := \mathcal{K}^n \bigcap \mathcal{K}$ {Remove the latest globally eliminated arms from local active set}
  for $k \in \bar{k}$ **do**
    $\mathcal{K} := \mathcal{K} \setminus \{k\}$ **if** $\lambda_k > M$ {Remove arms from global active set if the voting threshold is met}
  $t := t + 1$

We also introduce in Algorithm 4 the elimination subroutine, *Local Elimination*, used at line 5 of Algorithm 3. *Local Elimination* returns two arm sets: $\bar{k}$, containing the arms to eliminate locally, and $\bar{k}_G$, containing the arms to vote against for global elimination, i.e., those whose corresponding messages have not been corrupted.

Algorithm 4 CORRUPTED ELIMINATION Local Elimination

Input: Active agent n, corruption probability ξ
Output: A set of arms $\bar{k}$ to eliminate locally, and a set of arms $\bar{k}_G$ to vote against for global elimination
$\bar{k} = \{\}$, $\bar{k}_G = \{\}$
for all $k \in \mathcal{K}^n$ **do**
  if $\max_{l\in\mathcal{K}^n} \hat{\mu}_l^n(t) - \hat{\mu}_k^n(t) \geq 2\sqrt{\frac{\log(Kt^{n\,2}/\eta_\xi)}{t^n}}$ **then**
    $\bar{k} := \bar{k} \bigcup \{k\}$ {Add k to the set of arms to eliminate if elimination condition is met}
    Draw $\nu_k^n \sim \mathcal{B}(1-\xi)$
    if $\nu_k^n = 1$ **then**
      $\bar{k}_G := \bar{k}_G \cup \{k\}$ {If message is not corrupted, add k to the set of arms to vote against for global elim.}

C. Additional Experiments

In this section, we provide additional experimental results.

C.1. Experimental Setting

We consider the following stochastic environments suggested in related work (Féraud et al., 2019):

- **Problem 1:** $K = 10$ arms with Bernoulli distributions and means $\mu_1 = 0.7$, $\mu_2 = 0.5$, $\mu_3 = 0.3$ and $\mu_k = 0.1$ for $k = 4, ..., 10$:
- **Problem 2:** $K = 10$ arms with Gaussian distributions and means $\mu_1 = 11$, $\mu_2 = 10.8$, $\mu_k = 10.4$ for $k = 3, ..., 10$.

C.2. Sample Complexity and Privacy Level

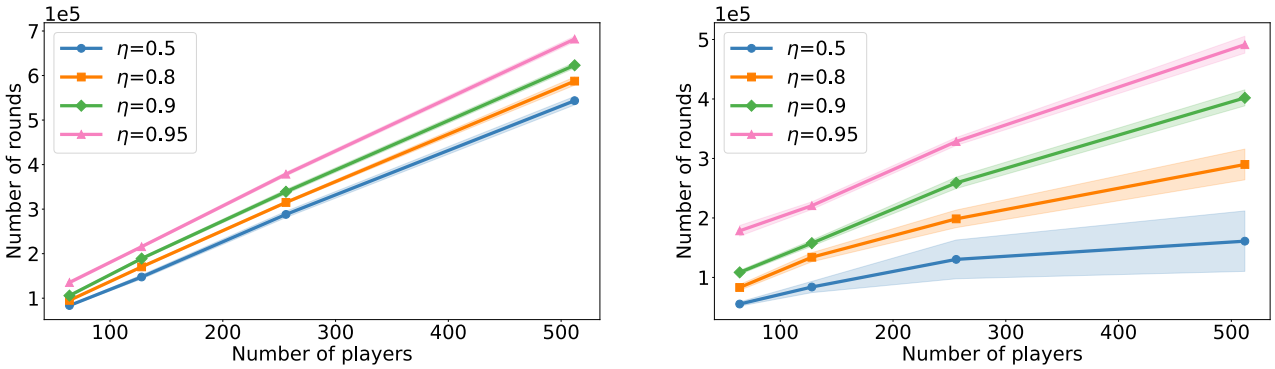

Figure 2. Sample complexity of Féraud et al. (2019) ($\xi = 0$) on **Problem 1** and **Problem 2** as a function of η

For CORRUPTED ELIMINATION, Figure 2 shows a positive correlation between the sample complexity and η, which is both the local BAI failure probability and the privacy level when $\xi = 0$. While this behavior is not reflected in the initial sample complexity bound stated in Féraud et al. (2019), Proposition 5.3 provides a first theoretical justification.

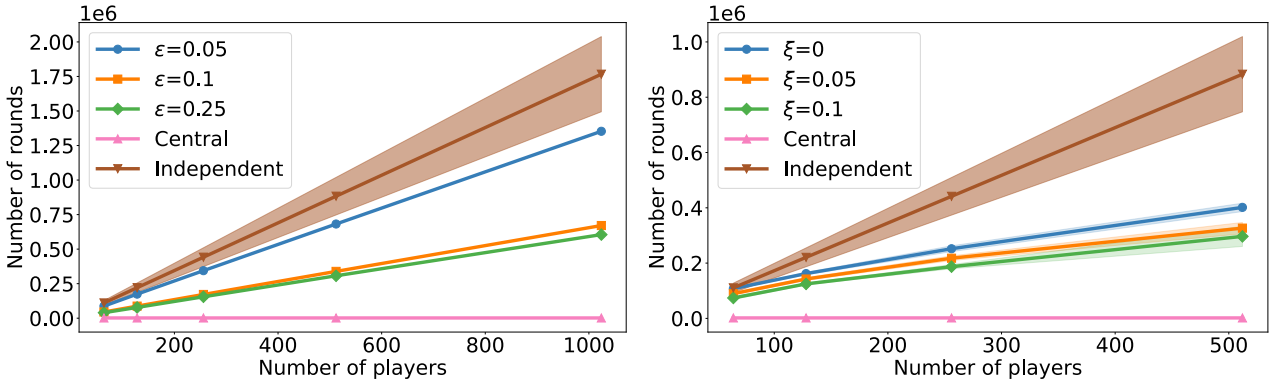

Figure 3. Sample complexity of DP-MASE (left) for $\epsilon = 0.05, 0.1, 0.25$ and CORRUPTED ELIMINATION (right) for $\xi = 0, 0.05, 0.1$ on **Problem 2**.

Experimental results on **Problem 2**, shown in Figure 3, lead to the same conclusions as results on **Problem 1**. Figure 4 shows the same results as in Figures 1 and 3 but without the baselines to distinguish the different curves more clearly.

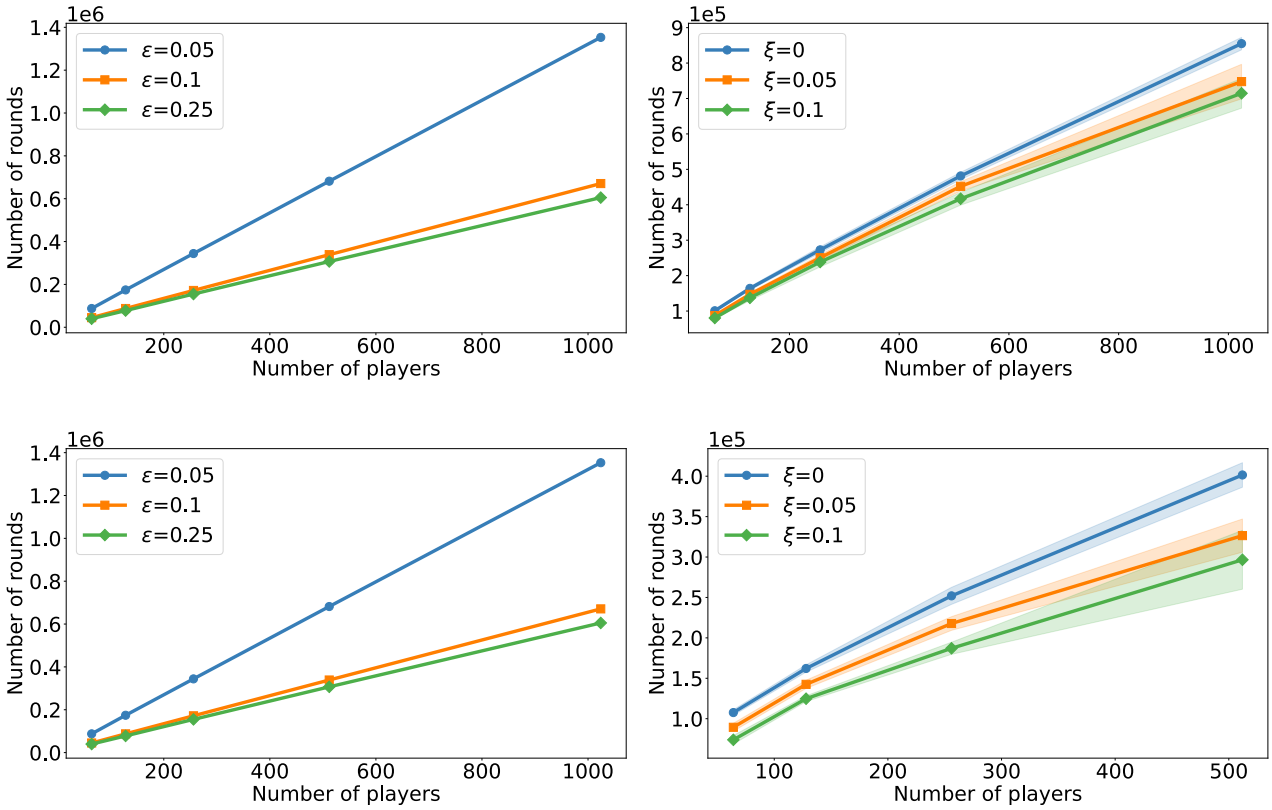

Figure 4. Sample complexity of DP-MASE (left) for $\epsilon = 0.05, 0.1, 0.25$ and CORRUPTED ELIMINATION (right) for $\xi = 0, 0.05, 0.1$ on **Problem 1** (top) and **Problem 2** (without baselines).

C.3. Sample Complexity and Local Accuracy

In the figures above, we displayed experimental sample complexity for DP-MASE for a fixed local accuracy ($\beta = 0.5$). We found it also interesting to compare sample complexity over different local accuracy levels.

In Figure 5, we plot the sample complexity of DP-MASE for $\beta = 0.3, 0.5, 0.8$, for a fixed privacy level $\epsilon = 0.5$. We observe that the sample complexity increases with local accuracy. This is in contradiction with what we observe for CORRUPTED ELIMINATION. Indeed, in the case $\xi = 0$ (corresponding to the algorithm from Féraud et al. (2019)), we already noticed that the sample complexity increases with η, which is both the local probability of failure and the global privacy level. In the case $\xi > 0$, ξ is positively correlated with the local accuracy $1 - \eta_\xi$ for a given apparent privacy level η. Higher corruption probability means we can run more confident local BAI without compromising the global privacy of the best arm. But sample complexity decreases with ξ, as shown empirically in Figure 1. Therefore, the sample complexity of CORRUPTED ELIMINATION decreases with local accuracy $1 - \eta_\xi$.

To explain these two opposite behaviors, we once again stress the implications of a higher local accuracy in terms of sample complexity. More confident local agents means that 1) more local samples are needed to estimate the best arm and 2) less independent eliminations are needed to reach a given global accuracy, hence a smaller voting threshold M. The former effect will tend to increase the sample complexity, while the latter (which we have called the *threshold effect*) will tend to decrease the sample complexity. In Section 5, based on experiments in Figure 1, we already argued that the *threshold effect* is predominant in the case of CORRUPTED ELIMINATION. Why would the local sample complexity matter more for DP-MASE? DP-MASE works in epochs, and with uniform agent sampling, it is likely that all agents will reach the end of a given epoch and proceed to an elimination at about the same time. In CORRUPTED ELIMINATION, each agent may vote for the global elimination of an arm at any time — not only at the end of an epoch, so that it is more likely to gather M votes against an arm early in the process. In terms of speeding up the elimination of suboptimal arms, the global voting process will thus be more important in CORRUPTED ELIMINATION than in DP-MASE.

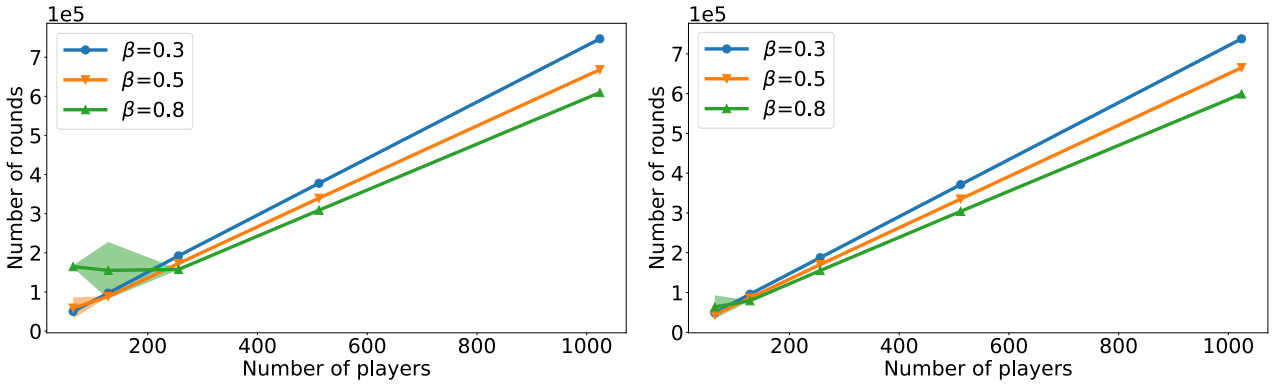

Figure 5. Sample complexity of DP-MASE for $\beta = 0.3, 0.5, 0.8$ on **Problem 1** (left) and **Problem 2** (right).

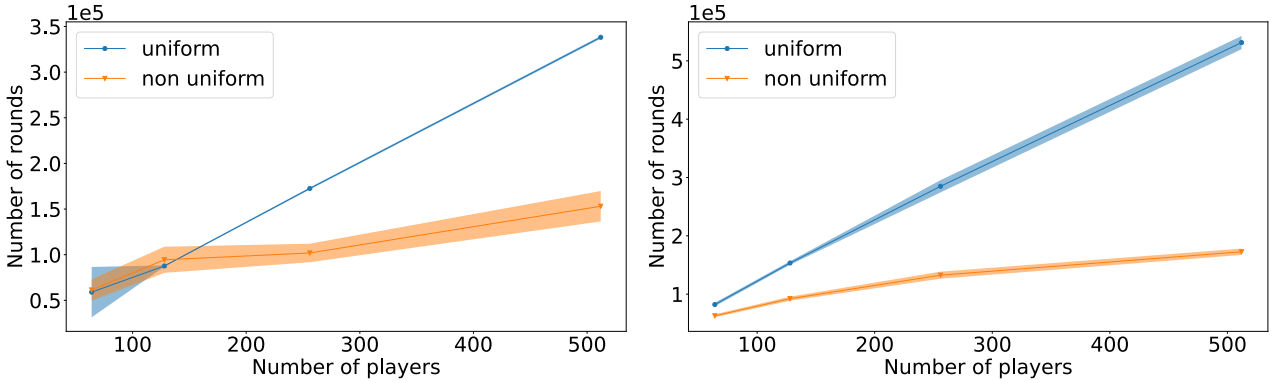

Figure 6. Sample complexity of DP-MASE (left) and CORRUPTED ELIMINATION (right) on **Problem 1** for uniform and non-uniform agent distributions.

C.4. Sample Complexity and Agent Distribution

In previous experiments, we assumed $P_{\mathcal{N}}$ was uniform. However, Proposition 4.3 suggests that $P_{\mathcal{N}}$ affects the sample complexity in the Multi-Agent Successive Elimination scheme. Indeed, through the global elimination process, the algorithm roughly waits for the M-th most frequent player to output his best arm estimate. Therefore, we expect DP-MASE and CORRUPTED ELIMINATION to be faster for unbalanced distributions, without losing accuracy, and indeed observe this behavior empirically.

To simulate an unbalanced agent distribution, we choose the following:

$$P_{\mathcal{N}}^{\gamma}(n) \propto (n+\alpha)^{-\gamma} \ ,$$

for some $\alpha \geq 0$ and $\gamma \in (0,1)$. Thus, the probability of sampling agent n decreases with n.

We illustrate the impact of $P_{\mathcal{N}}$ on the sample complexity for both DP-MASE and CORRUPTED ELIMINATION, comparing the uniform distribution to $P_{\mathcal{N}}^{\gamma}$ with $\gamma = 0.8$. We choose $\epsilon = 0.1$ and $\beta = 0.5$ for DP-MASE, and set $\xi = 0.05$ for CORRUPTED ELIMINATION. As expected, both methods are much faster when the agent distribution is unbalanced.