Asynchronous Multi-Agent Bandits: Fully Distributed *vs.* Leader-Coordinated Algorithms

XUCHUANG WANG*, University of Massachusetts Amherst, USA
YU-ZHEN JANICE CHEN*, University of Massachusetts Amherst, USA
XUTONG LIU, Carnegie Mellon University, USA
LIN YANG, School of Intelligence Science and Technology & National Key Laboratory for Novel Software Technology, Nanjing University, China
MOHAMMAD HAJIESMAILI, University of Massachusetts Amherst, USA
DON TOWSLEY, University of Massachusetts Amherst, USA
JOHN C.S. LUI, The Chinese University of Hong Kong, China

We study the cooperative asynchronous multi-agent multi-armed bandits problem, where each agent's active (arm pulling) decision rounds are asynchronous. That is, in each round, only a subset of agents is active to pull arms, and this subset is unknown and time-varying. We consider two models of multi-agent cooperation, fully distributed and leader-coordinated, and propose algorithms for both models that attain near-optimal regret and communications bounds, both of which are almost as good as their synchronous counterparts. The fully distributed algorithm relies on a novel communication policy consisting of accuracy adaptive and on-demand components, and successive arm elimination for decision-making. For leader-coordinated algorithms, a single leader explores arms and recommends them to other agents (followers) to exploit. As agents' active rounds are unknown, a competent leader must be chosen dynamically. We propose a variant of the `Tsallis-INF` algorithm with low switches to choose such a leader sequence. Lastly, we report numerical simulations of our new asynchronous algorithms with other known baselines.

CCS Concepts: • **Theory of computation → Online learning algorithms**; **Distributed algorithms**; **Multi-agent learning**; **Online learning theory**; **Sequential decision making**; *Multi-agent reinforcement learning*.

Additional Key Words and Phrases: Multi-Agent Bandits, Asynchronous Communication, Leader-Coordinated Algorithms, Fully Distributed Algorithms

ACM Reference Format:
Xuchuang Wang, Yu-Zhen Janice Chen, Xutong Liu, Lin Yang, Mohammad Hajiesmaili, Don Towsley, and John C.S. Lui. 2025. Asynchronous Multi-Agent Bandits: Fully Distributed *vs.* Leader-Coordinated Algorithms . *Proc. ACM Meas. Anal. Comput. Syst.* 9, 1, Article 3 (March 2025), 39 pages. https://doi.org/10.1145/3711696

*Both authors contributed equally to this research.

Authors' addresses: Xuchuang Wang, University of Massachusetts Amherst, Amherst, MA, USA, xuchuangw@gmail.com; Yu-Zhen Janice Chen, University of Massachusetts Amherst, Amherst, MA, USA, yuzhenchen@cs.umass.edu; Xutong Liu, Carnegie Mellon University, Pittsburgh, PA, USA, xutongl@andrew.cmu.edu; Lin Yang, School of Intelligence Science and Technology & National Key Laboratory for Novel Software Technology, Nanjing University, Suzhou, China, linyang@nju.edu.cn; Mohammad Hajiesmaili, University of Massachusetts Amherst, Amherst, MA, USA, hajiesmaili@cs.umass.edu; Don Towsley, University of Massachusetts Amherst, Amherst, MA, USA, towsley@cs.umass.edu; John C.S. Lui, The Chinese University of Hong Kong, Hong Kong, China, cslui@cse.cuhk.edu.hk.

ACM 2476-1249/2025/3-ART3
https://doi.org/10.1145/3711696

1 INTRODUCTION

Multi-agent multi-armed bandit (MA2B) is an important extension of the canonical multi-armed bandit model [38] in sequential decision making. A MA2B model consists of $K \in \mathbb{N}^+$ arms and $M \in \mathbb{N}^+$ agents. Each of these K arms is associated with a reward distribution, and whenever an agent pulls an arm, they receive a reward sample drawn from their distribution. Most prior works on MA2B study the stylized setting where agents are fully synchronous [11, 13, 14, 21, 58, 61, 64], meaning all M agents have synchronously aligned decision rounds. In this setting, at each time slot, every agent chooses one arm to pull and obtains a reward sample. The objective is to maximize the cumulative rewards of all agents over a given time horizon $T \in \mathbb{N}^+$ that is often large in practice.

In real-world distributed systems, however, agents are inherently asynchronous. For example, in clinical trials involving multiple hospitals (i.e., agents) with different patient groups, the timing of treatments may depend on patient availability, leading to non-deterministic and asynchronous decision-making rounds [6, 46]. Similarly, in the operation of multiple drones (i.e., agents), drones are tasked with various missions, and their ability to cooperate for specific tasks, such as path planning, often occurs asynchronously due to differing schedules [56]. Other examples of asynchronous agents include multiple secondary users sensing channel availability in cognitive radio network [43], and multiple edge devices searching for efficient servers in edge computing environments [26].

To address the natural asynchronicity in these online distributed systems, this paper studies the Asynchronous Multi-Agent Multi-Armed Bandits (AMA2B) model, where agents operate in fully asynchronous decision-making scenarios. In AMA2B, not all agents are *active* (available) in each time slot; instead, only a subset of agents is active for pulling arms. The active time slots for each agent can be irregularly spaced and unknown. In this paper, we study two types of cooperation: fully distributed and leader-coordinated. In fully distributed cooperation, all agents participate equally in cooperation (communication) and learning (arm exploration). This type of cooperation is suited for distributed systems with similar agents, such as multiple drones with identical specifications. On the other hand, the leader-coordinated cooperation imposes a hierarchical structure consisting of one leader agent and multiple followers. The leader performs exploration and recommends arms for followers to exploit. This model is applicable to systems with heterogeneous agents, such as drone swarms where some drones have specialized computation and storage units, or clinical trials where certain hospitals possess advanced testing labs. In these cases, the high-capacity agent is designated as the leader. These two cooperation models are fundamentally different, and designing cooperative algorithms for both requires addressing distinct challenges.

The primary objective of AMA2B is to minimize regret, which is the aggregate cumulative differences between rewards from all active agents pulling the optimal arm and those generated by the cooperative algorithm's arm-pulling policy. Minimizing regret is equivalent to maximizing the total reward of all agents. However, in a multi-agent system, the focus extends beyond just minimizing regret to also developing efficient cooperative algorithms that require low communication overhead. To achieve these goals, we pose the following research question:

Can we design asynchronous algorithms that achieve (near-)optimal regret with constant communication costs, on par with their synchronous counterparts, in both fully distributed and leader-coordinated paradigms?

1.1 Overview of Technical Challenges

The fundamental challenge of the AMA2B model arises from the fact that the agents' asynchronous activations are unknown and arbitrarily spaced. Without the alignment of decision rounds, many existing cooperation learning approaches become invalid. In fully distributed cooperation, each agent must decide when and with whom to communicate. In synchronous scenarios, this can be

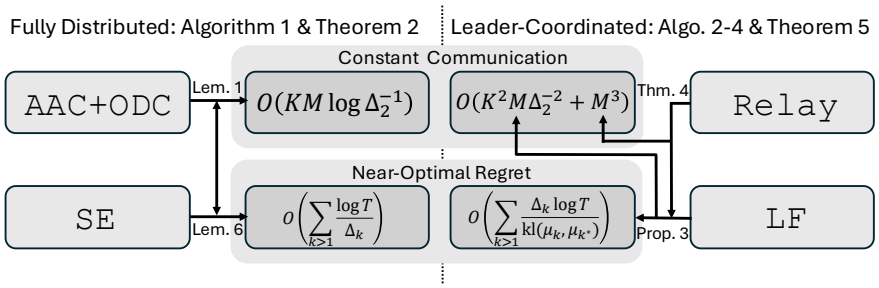


Fig. 1. Overview of algorithms and results: the fully distributed SE-AAC-ODC attains smaller communications, while the leader-coordinated LF-Relay enjoys a better regret bound.

easily managed by uniformly communicating with all other agents, assuming full connectivity among them. However, in asynchronous environments, some agents may be highly active, frequently collecting and sharing observations, while others may operate at a much lower frequency or even exit the system before the time horizon concludes. This makes communicating with less-active agents less beneficial. To overcome this challenge, an adaptive communication policy is required—one that can dynamically adjust both the timing of communication and the choice of communication partners based on agents' activity levels.

In leader-coordinated cooperation, the key challenge is how to select a *competent* leader—an agent remains active at a relatively high frequency throughout the time horizon. In synchronous scenarios, one can simply pick any agent as the leader. In asynchronous scenarios, however, agents' decision rounds are unknown and potentially adversarial, making it impossible to pre-select a competent leader. Tackling this challenging problem requires a dynamic leader selection approach to adaptively switch the leadership among agents as the time horizon progresses. This process introduces additional communication costs, so it is crucial to design a policy that minimizes unnecessary leadership switches, ensuring changes only occur when absolutely necessary.

1.2 Contributions

We formally introduce the asynchronous multi-agent multi-armed bandit problem (AMA2B) and its motivating applications in §2. Following this, we propose fully distributed algorithms (in §3) and leader-coordinated algorithms (in §4). Below, we summarize the key technical contributions of this paper. An overview of the algorithmic techniques and results is provided in Figure 1. Additionally, we compare the results with closely related prior works in §1.3.

First, we propose a fully distributed algorithm for AMA2B. The key challenges for devising an efficient communication policy for each agent are (i) determining when to trigger communication and, if triggered, (ii) selecting which agents to communicate with. To tackle these challenges, we introduce the accuracy adaptive communication (AAC) policy, which uses the relative amount of "new information" from recent local observations to decide when to communicate. We also propose the on-demand communication (ODC) policy, which utilizes token exchanges to identify eligible agents for communication. In addition to these communication protocols, we apply the successive elimination (SE) mechanism for arm pulling, which maintains a candidate arm set for pulling and gradually eliminates suboptimal arms from this set until only one arm remains. Our fully distributed algorithm, SE-AAC-ODC, combines the arm pulling policy SE with the AAC and ODC communication policies (§3.1). In §3.2, we prove that SE-AAC-ODC achieves near-optimal regret while requiring only a constant (time-independent) number of communications.

Table 1. Communication cost comparison of MA2B Algorithms in different Scenarios

	Fully Distributed		Leader-Coordinated	
	Algorithm	Communication	Algorithm	Communication
Synchronous	DoE-bandit	$O(KM\log\Delta_2^{-1})$ [61]	DPE2	$O(K^2M\Delta_2^{-2})$ [57]
Asynchronous	SE-ODC	$O(KM^2\Delta_2^{-2}\log T)$ [23]	–	–
	SE-AAC-ODC	$O(KM\log\Delta_2^{-1})$ (Thm. 2)	LF-Relay	$O(K^2M\Delta_2^{-2}+M^3)$ (Thm. 7)

Δ_2 is the smallest reward gap. All algorithms achieve near-optimal regrets.

Second, we propose a leader-coordinated cooperation algorithm for AMA2B. We begin by introducing a leader-follower scheme for agents with stochastic active decision rounds (§4.1). One key challenge in generalizing this scheme to arbitrary AMA2B is selecting a competent leader with a high activation frequency. Notably, as active decision rounds in AMA2B can be adversarial, it is impossible to pre-assign a competent leader. Addressing this, we propose a leader *relay* algorithm that dynamically transfers leadership among agents over the whole time horizon, ensuring that the leaders in the relay sequence collectively form a competent leader (§4.2). Inspired by the Tsallis-INF algorithm [65], originally proposed for best-of-both-world bandits, we develop a leader relay algorithm designed to minimize the frequency of leadership switches. By combining the leader relay algorithm with the leader-follower scheme, we propose the leader-coordinated LF-Relay algorithm for general AMA2B and prove that LF-Relay achieves near-optimal regret with constant communication (§4.3).

Finally, in §5, we conduct numerical simulations to evaluate the performance of SE-AAC-ODC and LF-Relay compared to known baseline algorithms. The numerical results confirm our theoretical results on achieving near-optimal regret and constant communication costs. In addition to introducing these two novel algorithms, we also explore how to theoretically ensure their privacy under a local differential privacy (LDP) model. The privacy guarantees are detailed in Appendix C.

1.3 Related Works

We compare the theoretical results of this paper and the most relevant prior results in Table 1, and in what follows we review them in detail.

For fully distributed algorithms, Yang et al. [61] propose a constant communication algorithm (DoE-bandit) for fully distributed synchronous MA2B. Their algorithm is based on a distributed online estimator that only needs constant communications to guarantee that *all* agents' estimation is as good as a centralized estimation. However, the aligned decision rounds of all agents, required by their synchronous estimators, becomes invalid in the misaligned asynchronous decision rounds in AMA2B. To circumvent the misaligned decision round issue, instead of aiming for all agents' estimators with good performance as in Yang et al. [61], we devise a weaker estimator, guaranteeing that *at least one* agent's estimator can have comparable performance to a centralized estimator, and we use this estimator with weak property to devise a fully distributed algorithm for AMA2B.

For the leader-coordinated cooperation case, Wang et al. [57] proposed a leader-follower algorithm (DPE2) that achieves the near-optimal regret upper bound with constant communications for synchronous MA2B. In the algorithm, one initially randomly picks one agent as the leader. However, when it comes to the asynchronous AMA2B, simply assigning an agent as the leader does not work, because the asynchronous active decision rounds of agents can be adversarial, and any fixed leader may not have further active rounds after being chosen. To address the challenge, we propose a leader relay algorithm, which dynamically transfers leadership among agents over time to ensure that the leaders in the relay sequence together compose "a competent leader".

In the context of asynchronous multi-agent multi-armed bandits (AMA2B), the only prior work we know is Chen et al. [23], where a fully distributed algorithm with an on-demand communication (ODC) policy is proposed for AMA2B. Chen et al. [23]'s algorithm achieves near-optimal regret at the expense of spending logarithmic $O(\log T)$ communications. While our algorithm also utilizes the ODC idea, it combines the ODC idea with our new accuracy adaptive communication (AAC) policy that helps to reduce the communication to $O(KM \log \Delta_2^{-1})$ (time T independent), where Δ_2 is the smallest reward gap (formally defined in §2). The new communication policy in this work is the first to achieve constant communications in AMA2B. Also, Chen et al. [23] do not provide any results for the leader-coordinated scenario.

More broadly, synchronous MA2B has been extensively studied in either a fully distributed setting [11, 16, 20–22, 27, 30, 32, 39, 42, 44, 49, 53, 59, 60] or a leader-coordinated setting [7, 21, 27, 37, 51, 57, 58], and various communication schemes such as peer-to-peer [28], consensus-based [44], gossip-style [22, 49], and immediate broadcasting [16, 59, 60] have been considered. Several prior works study asynchronous multi-agent cooperation in other online learning models, e.g., linear bandits [31, 40] and online convex optimization [19, 33, 34]. Stochastic asynchronous multi-agent with collision was also considered by [48], which is different from our arbitrary asynchronous with non-collision model. Beyond collaborative MA2B, a branch of prior MA2B works study a competitive setting where simultaneously pulling the same arm degrades the reward [9, 12, 14, 15, 52, 57]. These works are at a clear distance from the asynchronous model studied in this paper.

2 MODEL

We consider an asynchronous multi-agent multi-armed bandits (AMA2B) model including K arms and M agents. Each arm $k \in \mathcal{K} \coloneqq \{1, 2, \ldots, K\}$ is associated with a *Bernoulli* reward distribution with *unknown* mean μ_k, and we assume $1 > \mu_1 > \mu_2 \geqslant \ldots \geqslant \mu_K > 0$ such that arm 1 is the unique optimal arm. We define $\Delta_k \coloneqq \mu_1 - \mu_k$ for $k \geqslant 2$ as the reward gap between optimal arm 1 and suboptimal arm k. Each agent $m \in \mathcal{M} \coloneqq \{1, 2, \ldots, M\}$ is associated with an arbitrary sequence of activation times generated by an adversary at the beginning[1] and is unknown to agents. When an agent is activated, we refer to it as an *active* agent and refer to the time slot that an agent becomes active as an *(active) decision round* because the agent needs to pull an arm in the time slot. Denote by $\mathcal{T} \coloneqq \{1, 2, \ldots, T\}$ the set of time slots in which at least one agent is *active*. In each time slot $t \in \mathcal{T}$, each active agent $m \in \mathcal{M}$ selects one arm $k \in \mathcal{K}$ to pull and obtains a reward $X_k^{(m)}(t)$ drawn from its reward distribution. We denote by $\mathcal{T}^{(m)}$ the set of time slots that agent m is active, and $T^{(m)} \coloneqq |\mathcal{T}^{(m)}|$. The asynchronous setting of AMA2B is very general and can capture various adaptive and dynamic scenarios. For example, the setting allows any agents to freely drop out and rejoin the system, which is common in many real-world applications. We assume there are *no collisions or reward degradation*, i.e., when more than one agent pulls the same arm, each of them gets a reward independently drawn from the arm reward distribution.

Objective. We first define the expected (pseudo) regret of all agents as follows,

$$R(T) \coloneqq \sum_{m \in \mathcal{M}} \mathbb{E}\left[T^{(m)}\mu_1 - \sum_{t \in \mathcal{T}^{(m)}} X_{k^{(m)}(t)}^{(m)}\right] = \sum_{m \in \mathcal{M}} T^{(m)}\mu_1 - \mathbb{E}\left[\sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}^{(m)}} \mu_{k^{(m)}(t)}\right],$$

where the expectation is taken over the randomness of the algorithm and rewards realization. This cooperative AMA2B model allows agents to communicate with each other, and we define the

[1]This kind of adversary is called *oblivious* in online learning because it cannot adaptively alter the active decision rounds based on history and algorithm's action.

Table 2. Applications of the asynchronous multi-agent multi-armed bandits model

Application	Asynchronous Agents	Arms	Rewards
Clinical Trials† [6, 46]	Hospitals with patient arrivals	Medical treatments (e.g., medicines, vaccines)	Treatment effectiveness
Drone Swarm Planning† [56]	Drones hovering near the computing leader	Latency-sensitive task queues (e.g., path planning)	Task completion
Cognitive Radio Networks [45]	Secondary users with signal transmissions	Channels (e.g., radio frequencies)	Channel availability
Mobile Edge Computing [43]	Local stations with computing task arrivals	Computing nodes (e.g., edge, cloud)	Task completion

†: leader-coordinated applications.

expected rounds of communications as

$$C(T) \coloneqq \mathbb{E}\left[\sum_{m\in\mathcal{M}}\sum_{t\in\mathcal{T}} \mathbb{1}\{\text{agent } m \text{ sends a message at time } t\}\right],$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. Our objectives are to minimize both the regret $R(T)$ and the communication costs $C(T)$.

2.1 Representative Applications

In this section, we present several applications of the AMA2B model, as summarized in Table 2.

Clinical trials. Clinical trials aim to evaluate the effectiveness of medical treatments on patients across multiple hospitals [6, 46]. In this setting, K medical treatments (*arms*, e.g., different drugs or therapies) are administered across M hospitals (*agents*), with each hospital asynchronously receiving new patients. The active decision rounds $\mathcal{T}^{(m)}$ for a hospital m depend on when new patients arrive, which can be highly irregular and unpredictable. The effectiveness of treatment k is modeled as a Bernoulli random variable X_k, indicating whether the treatment was successful. Asynchronous patient arrivals lead to hospitals making treatment decisions at different times, complicating coordination across institutions. In leader-coordinated scenarios, a leader hospital, typically one with more resources or patients, must handle the additional challenge of coordinating treatments with other hospitals that may be receiving patients on completely different schedules. The goal is to maximize the overall treatment effectiveness across all hospitals by selecting the most effective treatments. Especially, the communication in clinical trial is privacy-sensitive, and the hospitals (agents) can protect the privacy of their patients by only sharing the aggregrated information of the treatment effectiveness, which is a side-effect of the communication-efficient algorithms.

Drone swarm planning. In drone swarm operations, multiple drones are tasked with performing latency-sensitive tasks, such as path planning or object recognition, in dynamic environments [56]. The system includes K task queues (*arms*) with different task priorities, and M drones (*agents*), each responsible for completing tasks asynchronously. The completion of a task in queue k is modeled as a Bernoulli random variable X_k, indicating whether the task was successfully completed within the required time. The active decision rounds $\mathcal{T}^{(m)}$ for a drone m depend on its operational status, which varies due to factors such as battery life, environmental conditions, and mission schedules. This leads to significant asynchronicity, as some drones may be more active than others, while others may need to recharge or switch tasks. Typically, drone swarm planning involves a leader drone that coordinates path planning for the swarm, which aligns with

the leader-coordinated scheme in §4. The active decision rounds $\mathcal{T}^{(m)}$ for a drone m correspond to the time slots when it is floating near the leader drone, which manages task coordination. We note that as all follower drones are helping the leader drone to complete the tasks from the K queues, they are all in the same context and have the same objective. The objective is to maximize the timely completion of high-priority tasks while efficiently coordinating drones with varying activity levels, especially when using a leader drone to manage swarm coordination.

Cognitive radio networks. A cognitive radio network (CRN) is a wireless communication system where secondary users opportunistically access spectrum allocated to primary users [1, 45]. In such a network, M secondary users (*agents*, such as smartphones) must asynchronously decide when to transmit data, depending on their individual data transmission needs, which occur irregularly. Each user chooses from K available spectrum channels (*arms*), and the availability of a channel k depends on whether a primary user is occupying it, modeled as a Bernoulli random variable X_k. The active decision rounds $\mathcal{T}^{(m)}$ for each secondary user m occur asynchronously, as they depend on the user's communication needs, which vary in both frequency and timing. The asynchronous nature of CRNs makes it difficult for secondary users to coordinate spectrum use efficiently, and the goal is to maximize the successful data transmission rate by selecting available channels.

Mobile edge computing. Mobile edge computing (MEC) is a distributed computing paradigm that brings computation and data storage closer to end-users [43]. In MEC systems, K computing nodes (*arms*, ranging from edge to cloud) provide computational resources to M local stations (*agents*, serving nearby smartphones and vehicles). These local stations asynchronously receive computational tasks from the devices they serve. The active decision rounds $\mathcal{T}^{(m)}$ for a station m occur whenever there is a nearby device offloading a task, which can happen unpredictably as task arrivals depend on end-user behavior. Each computing node k has a different latency based on its processing power and round-trip time (RTT), and whether a task is completed successfully at node k is modeled as a Bernoulli random variable X_k. The asynchronicity in MEC arises from the fact that stations handle tasks at irregular intervals, and computational resources may vary in availability, making coordinated task offloading and communication between stations challenging. The objective is to maximize the overall task completion rate across all stations by minimizing the latency, despite the asynchronous task arrivals and varying node performance.

Other Applications and Extensions. While this paper and the proposed model focus on applications without collisions or reward degradation, certain scenarios, such as cognitive radio networks and mobile edge computing, may involve these challenges. With the current communication algorithm design, the later-proposed AMA2B algorithms can be extended to handle such cases by modifying the optimal action: instead of all M agents converging on a single optimal arm, agents can distribute across a set of top M arms.

3 FULLY DISTRIBUTED ALGORITHM FOR AMA2B

In this section, we introduce a fully distributed asynchronous algorithm, called SE-AAC-ODC, for the AMA2B model. SE-AAC-ODC involves several key technical components (§3.1) — Accuracy Adaptive Communication (AAC), On-Demand Communication (ODC), and Successive Elimination (SE)—each contributing essential functionality to the overall approach. We also present the theoretical analysis of SE-AAC-ODC in §3.2. In the following, we first explain the technical challenges along with our algorithmic ideas for tackling the AMA2B model.

Design challenges and key ideas. While recent works in the synchronous multi-agent bandits setting [57, 61] have significantly improved the communication cost, in the asynchronous setting

with agents active arbitrarily, designing fully distributed cooperative algorithms is more challenging. For example, in communications between a "fast" agent (often active) and a "slow" agent (seldom active), letting all agents communicate in the same frequency as synchronous policies would cause many redundant communications. Because, firstly, the slow agent may not have new observations to communicate with the fast agent, and, secondly, the slow agent, if there are no future active rounds, may not need the extra information from the fast agent. An efficient fully distributed algorithm for asynchronous AMA2B should appropriately answer two critical questions: (1) when to communicate, and (2) who to communicate with, both of which are challenging due to the agent asynchronicity.

Our approach to addressing the question (1) is to have each agent wait until it has accumulated sufficient "new information" since its last communication, called accuracy adaptive communication (AAC). Addressing question (2) regarding whom to communicate with, requires the determination of which agents can benefit from the new information, named as on-demand communication (ODC). In the following, we present how both high-level communication ideas can be implemented in a cooperative multi-agent bandit algorithm.

3.1 SE-AAC-ODC: A Fully Distributed Bandit Algorithm

3.1.1 Accuracy adaptive communication (AAC) policy (Lines 11-15). We use the confidence radius (half of a confidence interval's width) to represent the accuracy of the current estimate of reward mean and determine when an agent decides to share information. We define the confidence radius as follows,

$$\mathtt{CR}(n) \coloneqq \min\left\{1, \sqrt{2\log T/n}\right\}, \tag{1}$$

where n is the number of samples (drawn from a Bernoulli distribution) used in this calculation. This confidence interval guarantees that, with a probability of at least $1-T^{-4}$, the true reward mean μ lies inside the confidence interval $(\hat{\mu}-\mathtt{CR}(n), \hat{\mu}+\mathtt{CR}(n))$, where $\hat{\mu}$ is the empirical average of n samples. To construct a confidence interval, the agents need to determine the number of observations n. However, since agents are distributed and asynchronous, without timely communication, an agent does not know the exact pull times for other agents since their last communication. To address this issue, we use the number of agent m's recent local observations (since the last communication) as a surrogate for the number of other agents' recent local observations.

To facilitate presentation of the AAC policy, we first fix an arm k and consider the task that all asynchronous agents cooperate to estimate arm k's mean μ_k. Denote by $n_k^{(m)}(t)$ the number of local observations of arm k (excluding those received from others) by agent m up to time slot t, and by $n_k(t)$ the total number of times among all agents that arm k has been pulled on and before time slot t, i.e., $n_k(t)=\sum_{m\in\mathcal{M}} n_k^{(m)}(t)$. Denote the last communication rounds for sharing arm k's observations at time slot t as $\tau_k(t)$. We denote $\mathtt{ECR}_k^{(m)}(t)$ an *estimated* confidence radius of agent m for arm k at time t as a representation for accuracy, which can be expressed as

$$\mathtt{ECR}_k^{(m)}(t) \coloneqq \min\{1, \mathtt{CR}(n_k(\tau_k(t)) + M(n_k^{(m)}(t) - n_k^{(m)}(\tau_k(t))))\}. \tag{2}$$

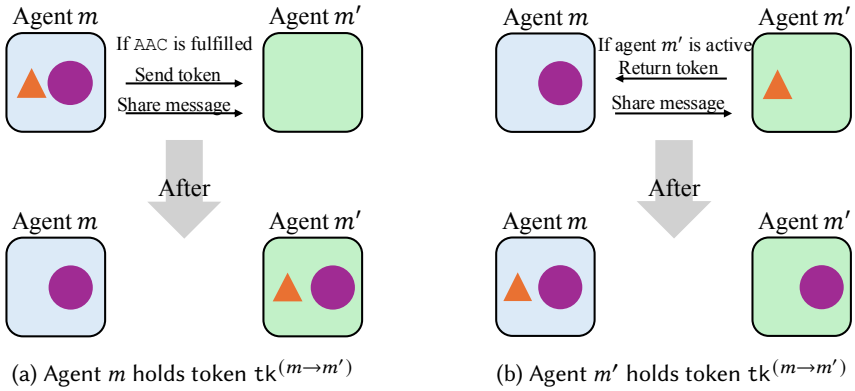
Taking minimum with 1 is because arm reward means lie in $(0,1)$ and the value 1 is the radius upper bound. The term $M(n_k^{(m)}(t) - n_k^{(m)}(\tau_k(t)))$ acts as a surrogate for the number of recent observations of other agents for arm k since the last communication time $\tau_k(t)$ by using the agent m's recent local observations. We note that if the current time slot is a communication round, i.e., $t=\tau_k(t)$, then there is no surrogate observation, therefore, the confidence radius is equal to the estimated one, i.e., $\mathtt{CR}(n_k(\tau_k(t))) = \mathtt{ECR}_k^{(m)}(\tau_k(t)), \forall m$. Hence, we refer to the estimated confidence radius in a communication round, i.e., $\mathtt{ECR}_k^{(m)}(\tau_k(t))$ as the *aligned* confidence radius.

Algorithm 1 SE-AAC-ODC: Fully distributed algorithm for agent m

1: **Inputs:** threshold parameter $\alpha > 1$, time horizon T, the number of arms K and agents M
2: **Initialization:** $\tau_k(t), \hat{\mu}_k^{(m)}(t), n_k^{(m)}(t), S_k^{(m)}(t) \leftarrow 0$ for all agents $m' \in \mathcal{M}$ and arms $k \in \mathcal{K}$
3: **for all** $t \in \mathcal{T}$ **do**
4: **if** $t \in \mathcal{T}^{(m)}$ **then**
5: Update $\hat{\mu}_{k'}^{(m)}(t)$ for all arm $k' \in \mathcal{K}$ according to (3)
6: Update the candidate arm set $C(t)$ according to (4) ▷ Elimination
7: **if** any arm elimination happens **then** notify other agents for this elimination
8: Pick an arm k from candidate arm set $C(t)$ to pull in a Round-Robin manner
9: Obtain arm k's reward observation $X_k^{(m)}(t)$
10: $S_k^{(m)}(t) \leftarrow S_k^{(m)}(t) + X_k^{(m)}(t)$ and $n_k^{(m)}(t) \leftarrow n_k^{(m)}(t) + 1$
11: **if** $\alpha \mathsf{ECR}_k^{(m)}(t) \leqslant \mathsf{ECR}_k^{(m)}(\tau_k(t))$ **then** ▷ AAC communication condition
12: $\tau_k(t) \leftarrow t$ ▷ Update the latest communication time slot
13: Collect $(n_k^{(m')}(t), S_k^{(m')}(t))$ from all agents m' whose token $\mathtt{tk}^{(m \to m')}$ is held by agent m
14: Update $n_k(t)$ and $\hat{\mu}_k(t)$
15: Send $(k, n_k(t), \hat{\mu}_k(t), \tau_k(t))$ to other agents m' whose token $\mathtt{tk}^{(m \to m')}$ is held by agent m
16: Send tokens $\mathtt{tk}^{(m \to m')}$ to corresponding agent m'
17: **end if**
18: Return all tokens $\mathtt{tk}^{(m' \to m)}$ on hold to corresponding agents m'
19: **end if**
20: **if** receive $\mathtt{tk}^{(m \to m')}$ from agent m' **then**
21: Send messages $(k', n_{k'}(t), \hat{\mu}_{k'}(t), \tau_{k'}(t))$ for arms whose updates were blocked due to that agent m did not hold token for agent m'
22: Keep token $\mathtt{tk}^{(m \to m')}$
23: **end if**
24: **if** receive $\mathtt{tk}^{(m' \to m)}$ from agent m' **then**
25: Keep token $\mathtt{tk}^{(m' \to m)}$
26: **end if**
27: **if** receive broadcast $(k', n_{k'}(t), \hat{\mu}_{k'}(t), \tau_{k'}(t))$ from other agents m' **then**
28: Update local $(n_{k'}(t), \hat{\mu}_{k'}(t), \tau_{k'}(t))$ for arm k'
29: **end if**
30: **end for**

We use the ratio between the latest aligned confidence radius and the current estimated confidence radius $\mathsf{ECR}_k^{(m)}(\tau_k(t))/\mathsf{ECR}_k^{(m)}(t)$ to measure the relative amount of "new information" that agent m collects since its last communication. Notice that the ratio increases with the number of observations. Hence, when the ratio exceeds some predetermined threshold α, there is sufficient "new information" to initiate a new communication (Line 11).

3.1.2 On-demand communication (ODC) policy (Lines 15-25). We employ a token-based mechanism to implement On-Demand Communication (ODC). The system has in total $M(M-1)$ tokens, where each token $\mathtt{tk}^{(m \to m')}$ is associated with a pair of agents m and m' ($m \neq m'$) and is either held by agent m or m' at any time slot t. At initialization, each agent $m \in \mathcal{M}$ is assigned $M-1$ tokens $\mathtt{tk}^{(m \to m')}$, each corresponding to one other agent $m' \in \mathcal{M} \setminus \{m\}$. The token $\mathtt{tk}^{(m \to m')}$ (resp., not) held by agent m indicates to agent m that the agent m' is (resp., not) *on-demand* and is (resp., not) interested in receiving information from agent m. More specifically, the mechanism has two $\mathtt{tk}^{(m \to m')}$-related operations:



(a) Agent m holds token $\mathtt{tk}^{(m\to m')}$ (b) Agent m' holds token $\mathtt{tk}^{(m\to m')}$

Fig. 2. Illustration of the token-based ODC mechanism: Token $\mathtt{tk}^{m\to m'}$ Message to share

- **Agent m holds token $\mathtt{tk}^{(m\to m')}$ (Fig. 2a):** Only when agent m holds token $\mathtt{tk}^{(m\to m')}$ can it communicate to agent m' (for actual triggering communication, agent m also needs to fulfill the AAC condition in Line 11). At this type of communications from agents m to agent m' (Line 15, agent m addressing the *demand* of agent m'), agent m also sends the token $\mathtt{tk}^{(m\to m')}$ to agent m' (Line 16).
- **Agent m' holds token $\mathtt{tk}^{(m\to m')}$ (Fig. 2b):** Once agent m' is active, it immediately returns the token $\mathtt{tk}^{(m\to m')}$ to agent m as a signal to agent m notifying that agent m' is *on-demand* (Line 18). Agent m then sends one updating message to agent m' containing information that was not previously communicated due to agent m not having a token for agent m' (Lines 20-22), while agent m still keeps the token $\mathtt{tk}^{(m\to m')}$.

3.1.3 Successive elimination (SE) arm pull policy (Lines 6-10). Denote by $C(t)$ the candidate arm set, which is initialized as the full arm set, i.e., $C(0) = \mathcal{K}$. The main idea of successive elimination is to uniformly explore all remaining arms in the candidate arm set in a round-robin manner (Line 8) and remove an arm from the candidate arm set (Line 6) whenever it is identified as suboptimal. Note that whenever an agent eliminates one arm, this agent notifies all other agents to eliminate the arm from their candidate arm sets as well; hence, all agents have the identical candidate arm set (Line 7).

Next, we introduce notation to illustrate the technical details of eliminating a suboptimal arm from the candidate arm set. Denote by $S_k^{(m)}(t)$ the sum of $n_k^{(m)}(t)$ observations of arm k for agent m at time slot t, which can be expressed as $S_k^{(m)}(t) \coloneqq \sum_{s=1}^{n_k^{(m)}(t)} X_k^{(m)}(s)$, where $X_k^{(m)}(s)$ is the s^{th} reward observation for arm k of agent m. Next, we introduce estimator $\hat{\mu}_k^{(m)}(t)$ for the reward mean of agent m for arm k at time t as follows,

$$\hat{\mu}_k^{(m)}(t) \coloneqq \frac{n_k(\tau_k(t))\hat{\mu}_k(\tau_k(t)) + (S_k^{(m)}(t) - S_k^{(m)}(\tau_k(t)))}{n_k(\tau_k(t)) + n_k^{(m)}(t) - n_k^{(m)}(\tau_k(t))}, \tag{3}$$

where $\tau_k(t)$ is the latest time slot (on or before t) that agent m communicates information about arm k to other agents (i.e., synchronizes globally), and $\hat{\mu}_k(\tau_k(t))$ is the average of all $n_k(\tau_k(t))$ observations. This estimator in (3) together with the confidence radius (1) yields a confidence interval for μ_k with bounds $\hat{\mu}_k^{(m)} \pm \mathtt{CR}(n_k(\tau_k(t)) + (n_k^{(m)}(t) - n_k^{(m)}(\tau_k(t))))$. With the above confidence

interval, an arm k is eliminated by agent m from the candidate set $C(t)$ at time t if there exists an arm $k' \in C(t)$ such that the upper confidence bound of arm k is less than the lower confidence bound of arm k', i.e.,

$$\begin{aligned} \hat{\mu}_k^{(m)}(t) + \mathsf{CR}\left(n_k(\tau_k(t)) + (n_k^{(m)}(t) - n_k^{(m)}(\tau_k(t)))\right) \\ < \quad \hat{\mu}_{k'}^{(m)}(t) - \mathsf{CR}\left(n_{k'}(\tau_{k'}(t)) + (n_{k'}^{(m)}(t) - n_{k'}^{(m)}(\tau_{k'}(t)))\right). \end{aligned} \tag{4}$$

Inequality (4) is the elimination condition used for identifying suboptimal arms in the candidate set.

3.2 Theoretical Analysis of SE-AAC-ODC

This section presents the theoretical analysis of the SE-AAC-ODC algorithm. We start by studying the estimation performance of the estimator in (3) in Lemma 1.

Lemma 1. *Assume M agents independently and asynchronously sample arm k associated with i.i.d. Bernoulli distributions with unknown mean μ_k, as Algorithm 1 (with threshold parameter $\alpha > 1$), and $n_k(t)$ is the total number of available samples across all agents. For any t, there* exists *an agent ℓ such that, with probability $1 - MT^{-3}$, we have*

$$|\hat{\mu}_k^{(\ell)}(t) - \mu_k| \leqslant \alpha \mathsf{CR}(n_k(t)).$$

Proof of Lemma 1 is presented in Appendix A.1. Lemma 1 shows that among the estimates of all agents for reward mean μ_k of arm k, at least one agent ℓ's estimate $\hat{\mu}_k^{(\ell)}(t)$ enjoys the estimate accuracy comparable (up to an α factor) to that of an estimate that uses all of the observations of arm k, i.e., all $n_k(t)$ samples. In the proof of Lemma 1 deferred to Appendix A, this agent ℓ is set to be the one with the largest number of active decision rounds since the last communication time $\tau_k(t)$ for arm k. As the active decision rounds of agents in AMA2B are asynchronous and can be arbitrarily, the agent with the best estimation performance may change over time. This proof relies on dynamically determining which agent has the most active decision rounds since last communication. We highlight that Lemma 1 describes the key property of the estimator in (3) in an asynchronous setting, which is a novel result compared to the synchronous setting.

Our main theorem in Theorem 2 shows that although agents with good estimates can vary, SE-AAC-ODC is able to adapt to the changes and achieve near-optimal regret and constant communication cost.

Theorem 2. *Given parameter $\alpha > 1$, Algorithm 1's regret and communication are upper bounded as follows,*

$$R(T) \leqslant \sum_{k>1} \frac{8(1+\alpha)^2 \log T}{\Delta_k} + \sum_{k>1} M\Delta_k + KM^2, \tag{5}$$

$$C(T) \leqslant \sum_{k>1} 2M \log_\alpha \left(\frac{2(1+\alpha)}{\Delta_k}\right) + 2M \log_\alpha \left(\frac{2(1+\alpha)}{\Delta_2}\right) + 2KM^3. \tag{6}$$

Communication costs discussion. The only prior algorithm for AMA2B [23] needs $O(KM^2\Delta_2^{-1}\log T)$ communications to achieve a near-optimal regret upper bound, while our SE-AAC-ODC only needs $O(KM\log\Delta_2^{-1})$ communications, much smaller than that of [23] especially when T is large.

Regret optimality discussion. We recall MA2B's regret lower bound [57, §1.2] (also proved for AMA2B by Chen et al. [23, Appendix C]), $\liminf_{T\to\infty} \frac{R(T)}{\log T} \geqslant \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\nu_k, \nu_1)}$, where $\mathrm{KL}(\cdot,\cdot)$ denotes the KL-divergence between two distributions, and ν_k denotes the reward distribution of arm k.

It can be simplified as $\liminf_{T\to\infty} \frac{R(T)}{\log T} \geqslant C \cdot \sum_{k>1} \frac{1}{\Delta_k}$, where C is a constant that depends on the specific reward distribution. On the other hand, the regret upper bound of SE-AAC-ODC in (5) can be rewritten as $\limsup_{T\to\infty} \frac{R(T)}{\log T} \leqslant 8(1+\alpha)^2 \cdot \sum_{k>1} \frac{1}{\Delta_k}$. Therefore, comparing the above asymptotic regret lower and upper bounds shows that the regret of SE-AAC-ODC is tight up to a constant factor.

Impact of Parameter α. Theorem 2 shows a trade-off between communication $O(\sum_k \log_\alpha \Delta_k^{-2})$ and regret $O(\alpha^2 \sum_k \Delta_k^{-1} \log T)$, and the parameter α in SE-AAC-ODC controls the trade-off. That is, a larger α leads to a smaller communication cost but a larger regret upper bound. In Appendix E, we conduct additional numerical experiments to investigate the impact of α on the regret and communication cost of SE-AAC-ODC, which reveals another interesting observation that the regret upper bound of SE-AAC-ODC is not sensitive to the choice of α.

Without Known Time Horizon T. One limitation of SE-AAC-ODC is that it requires the knowledge of time horizon T to be known in advance. The time horizon T is used to determine the confidence interval width in (1) in SE-AAC-ODC and guarantee that the arm elimination proceeds properly with high probability. Here, we propose two solutions to address this limitation: (i) Practically, one can replace the T in determining $\mathsf{ECR}_k^{(m)}(t)$ in (2) and arm elimination in (4) with the current decision round number t. This modification would lead to a slightly larger confidence interval width and a slightly higher failure probability of arm elimination. But both would be mitigated as the learning proceeds and the time step t increases, and hence one can expect the algorithm still performs almost the same as the original one. (ii) Theoretically, one can use the doubling trick [10] to estimate the time horizon T in a fully distributed manner. However, this modification, while maintains the near-optimal regret bound, would increase the communication cost by a $\log T$ factor, making the communication cost dependent on the time horizon T. The leader-coordinated algorithm proposed in §4 does not require the knowledge of T.

Proof idea for communication bound. Detail proof of Theorem 2 is presented in §3.2.1. Here, we provide a sketch proof for the communication upper bound in (6). There are three contributions to the communication cost of SE-AAC-ODC: (i) arm elimination notifications due to SE, bounded by KM, (ii) token exchanges (send and return) for the implementation of ODC, and (iii) message sharing according to AAC. Notice that each token sending of (ii) is always together with one message sending triggered by AAC of (iii) (see Lines 15-16), and each token returning is a consequence of the token's previous sending. This implies that the communication cost due to (ii) is upper bounded by at most twice the communication cost of (iii). Therefore, to bounded the total communication cost, we only need to bound the communication costs of AAC, i.e., (iii). For any candidate arm $k \in C(t)$, AAC in Line 11 triggers one communication when the ratio $\mathsf{ECR}_k^{(m)}(\tau_k(t))/\mathsf{ECR}_k^{(m)}(t)$ is greater than the threshold α, and AAC stops communicating about the arm k when the arm is eliminated, which happens when $\mathsf{CR}_k^{(m)}(t) < \Delta_k/2$, that is, $\mathsf{ECR}_k^{(m)}(t) < c\Delta_k$ for some constant $c > 0$. Therefore, the total communication costs for this arm k are upper bounded by $O(M \log_\alpha \Delta_k^{-1})$, where the multiplicative M is because the communication is involved with all agents.

3.2.1 Detail Proof of Theorem 2. Applying Hoeffding's inequality and union bound, we know that for any agent m, the following inequality holds with a probability of at least $1 - MT^{-3}$,

$$|\hat{\mu}_k^{(m)}(t) - \mu_k| \leqslant \mathsf{CR}(n_k(\tau_k(t)) + (n_k^{(m)}(t) - n_k^{(m)}(\tau_k(t)))), \tag{7}$$

where we recall that $\tau_k(t)$ is the last time slot that agents conduct communication to synchronize the observations of arm k before time slot t.

We denote the decision made at a time slot t as a Type-I decision when (7) holds for any agent $m \in \mathcal{M}$ and arm $k \in \mathcal{K}$ at this time slot t; otherwise, we denote it as a Type-II decision.

We note that, as long as the decision is Type-I, the elimination condition in (4) always correctly eliminates suboptimal arms, and, therefore, when there is only one arm remaining in the candidate arm set $C(t)$, it is the optimal arm for sure. In the next two steps, we bound the probability that there are any Type-II decisions and the number of pulling times of any suboptimal arm when there are only Type-I decisions respectively.

Step 1. Upper bound the probability of any Type-II decision occurring. Below we bound the probability of the event that there exists any confidence interval not containing its corresponding true reward mean.

$$\begin{aligned}
&\mathbb{P}(\exists(k,m,t), |\hat{\mu}_k^{(m)}(t) - \mu_k| > \mathsf{CR}(n_{k'}(\tau_k(t)) + (n_k^{(m)}(t) - n_k^{(m)}(\tau_{k'}(t))))) \\
\leqslant\ &\mathbb{P}(\exists(k,m,t,n), |\hat{\mu}_k^{(m)}(t) - \mu_k| > \mathsf{CR}(n)) \\
\leqslant\ &\sum_{(k,m,t,n)\in(\mathcal{K}\times\mathcal{M}\times\mathcal{T}\times\mathcal{T})} \mathbb{P}(|\hat{\mu}_k^{(m)}(t) - \mu_k| > \mathsf{CR}(n)) \\
\leqslant\ &\sum_{(k,m,t,n)\in(\mathcal{K}\times\mathcal{M}\times\mathcal{T}\times\mathcal{T})} MT^{-3} = KM^2T^{-1}.
\end{aligned}$$

Step 2. Upper bound the number of times of pulling suboptimal arms.

Lemma 3. *At any time $t \leqslant T$, if the optimal arm lies in the candidate set and an agent m makes a Type-I decision with pulling a suboptimal arm k, i.e., $I^{(m)}(t) = k$, we have $n_k(t) \leqslant \frac{8(1+\alpha)^2\log T}{\Delta_k^2}$. Therefore, the total number of pulling times of arm i in the whole time horizon is upper bounded as follows,*

$$n_k(T) \leqslant \frac{8(1+\alpha)^2\log T}{\Delta_k^2} + M.$$

Proof. Arm k is pulled at some time t

$$\begin{aligned}
&\overset{(a)}{\Longrightarrow} k \in C(t) \text{ for agent } \ell \text{ fulfills Lemma 1} \\
&\overset{(b)}{\Longrightarrow} \hat{\mu}_k^{(\ell)}(t) + \mathsf{CR}(n_k(\tau_k(t)) + M(n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t)))) \\
&\qquad\geqslant \hat{\mu}_{k'}^{(\ell)}(t) - \mathsf{CR}(n_{k'}(\tau_{k'}(t)) + M(n_{k'}^{(\ell)}(t) - n_{k'}^{(\ell)}(\tau_{k'}(t)))) \text{ for any } k' \in C(t) \\
&\overset{(c)}{\Longleftrightarrow} \hat{\mu}_k^{(\ell)}(t) + 2\mathsf{CR}(n_k(\tau_k(t)) + M(n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t)))) \geqslant \hat{\mu}_{k'}^{(\ell)}(t) \text{ for any } k' \in C(t) \\
&\overset{(d)}{\Longrightarrow} \hat{\mu}_k^{(\ell)}(t) + 2\mathsf{CR}(n_k(t)) \geqslant \hat{\mu}_{k'}^{(\ell)}(t) \text{ for any } k' \in C(t) \\
&\Longrightarrow \hat{\mu}_k^{(\ell)}(t) + 2\mathsf{CR}(n_k(t)) \geqslant \hat{\mu}_1^{(\ell)}(t) \\
&\overset{(e)}{\Longrightarrow} \mu_k + (2+\alpha)\mathsf{CR}(n_k(t)) \geqslant \mu_1 - \alpha\mathsf{CR}(n_1(t)) \\
&\Longrightarrow 2(1+\alpha)\mathsf{CR}(n_k(t)) \geqslant \mu_1 - \mu_k = \Delta_k \qquad (8) \\
&\Longrightarrow n_k(t) \leqslant \frac{8(1+\alpha)^2\log T}{\Delta_k^2},
\end{aligned}$$

where (a) is because the candidate arm sets $C(t)$ are the same for all agents (including arm ℓ), (b) is by the definition of candidate arm set, (c) is because arms in the candidate arm set are evenly explored in a round-robin manner, (d) is from the definition of arm ℓ in (14), and (e) is by applying Lemma 1.

Lastly, since the pulling of arm k in the critical time slot t is not counted, the total pulling times of arm k may be increased by M at most, i.e.,

$$n_k(T) \leqslant \frac{8(1+\alpha)^2 \log T}{\Delta_k^2} + M.$$

□

Combining the results of Steps 1 and 2, the group regret is upper bounded as follows,

$$\begin{aligned}\mathbb{E}[R(T)] &\leqslant \sum_{k>1} n_k(T) \times \Delta_k + KM^2T^{-1} \times T \\ &\leqslant \sum_{k>1} \frac{8(1+\alpha)^2 \log T}{\Delta_k} + \sum_{k>1} M\Delta_k + KM^2.\end{aligned}$$

Step 3. Upper bound communication costs. If there are any Type-II decisions, the total communication times is at most KM^2.

Assume there is no Type-II decision. In the proof of Lemma 3, we have a middle step (8): for any suboptimal arm k, denoting κ_k as the last time slot that the arm was pulled, we have

$$2(1+\alpha)\mathsf{CR}(n_k(\kappa_k)) \geqslant \Delta_k.$$

Recall that $\tau_k(t)$ is the latest communication round about arm k on or before time slot t. Then, $\mathsf{ECR}_k^{(m)}(\tau_k(\kappa_k))$ is the ECR at the latest communication for arm k which can be upper bounded as follows,

$$\mathsf{ECR}_k^{(m)}(\tau_k(\kappa_k)) \overset{(a)}{=} \mathsf{CR}(n_k(\tau_k(\kappa_k))) \geqslant \mathsf{CR}(n_k(\kappa_k)) \geqslant \frac{\Delta_k}{2(1+\alpha)},$$

where equality (a) is because $\tau_k(\kappa_k)$ is a communication time slot in which the estimated confidence radius (ECR) is equal to CR. Recall the initial $\mathsf{ECR}_k^{(m)}(0) = 1$ by definition. The total number of times of communication on arm k is upper bounded as follows,

$$\log_\alpha\left(\frac{\mathsf{ECR}_k^{(m)}(0)}{\mathsf{ECR}_k^{(m)}(\tau_k(\kappa_k))}\right) \leqslant \log_\alpha\left(\frac{2(1+\alpha)}{\Delta_k}\right).$$

Since all arms are pulled in a round-robin manner, the communication cost on the optimal arm is upper bounded by $\log_\alpha\left(\frac{2(1+\alpha)}{\Delta_2}\right)$ where Δ_2 is the smallest reward gap.

Summing the above two type cases yields the communication upper bound as follows,

$$\sum_{k>1} \log_\alpha\left(\frac{2(1+\alpha)}{\Delta_k}\right) + \log_\alpha\left(\frac{2(1+\alpha)}{\Delta_2}\right) + KM^2.$$

As each communication round above needs $2(M-1)$ communications and the notification of arm elimination needs $(K-1)M$ communications in total, the final communication costs are upper bounded by

$$\sum_{k>1} 2M \log_\alpha\left(\frac{2(1+\alpha)}{\Delta_k}\right) + 2M \log_\alpha\left(\frac{2(1+\alpha)}{\Delta_2}\right) + 2KM^3.$$

4 LEADER-COORDINATED ALGORITHMS FOR AMA2B

This section devises leader-coordinated algorithms for AMA2B and analyzes their theoretical performance in terms of regret and communication. In §4.1, we present a leader-follower scheme inspired by the synchronous MA2B [57] and extend it to address a special case of AMA2B where the agent asynchronicity follows stochastic patterns. In §4.2, we propose a leader relay algorithm for the general asynchronous scenario. In §4.3, we combine both algorithms and present a leader-coordinated algorithm for AMA2B as well as its theoretical guarantees.

4.1 LF: A Leader-Follower Scheme

The leader-coordinated algorithm assigns one agent as the leader, denoted as ℓ, and the rest as followers. The leader explores arms and recommends its estimated optimal arm to followers, while the followers keep pulling the arm recommended by the leader. The leader-follower scheme was first introduced by Wang et al. [57] for fully synchronous MA2B. In the synchronous setting, the active decision rounds of all agents are the same, and arbitrarily selecting an agent as the leader would work. However, in the asynchronous case, the leader selection is nontrivial. The main challenge lies in how to select a competent leader.

Algorithm 2 LF: Leader's decision-making procedure

Input: the leader agent index ℓ
Initialize: KL-UCB index $d_k(t) \leftarrow 0$, reward mean estimate $\hat{\mu}_k^{(\ell)}(t) \leftarrow 0$, and estimated optimal arm in previous active time slot $I' \leftarrow 0$, exploration arm set $\mathcal{D}(t) \leftarrow \emptyset$
for all $t \in \mathcal{T}^{(\ell)}$ **do**
  if $\mathcal{D}(t) = \emptyset$ **then**
    Update $\hat{\mu}_k^{(\ell)}(t), n_k^{(\ell)}(t), d_k(t)$ of all arms $k \in \mathcal{K}$
    $I(t) \leftarrow \arg\max_{k\in[K]} \hat{\mu}_k(t)$ ▷ Empirical optimal arm
    if $I(t) \neq I'$ **then** ▷ Empirical optimal arm changes
      Send the new empirical optimal arm $I(t)$ to all followers
      $I' \leftarrow I(t)$
    end if
    $J(t) \leftarrow I(t)$
    $\mathcal{D}(t) \leftarrow \{k \in [K] : d_k(t) > \hat{\mu}_{I(t)}(t)\}$ ▷ Arms with high KL-UCB indices
  else
    $J(t) \leftarrow \arg\max_{k\in\mathcal{D}(t)} d_k(t)$
    $\mathcal{D}(t) \leftarrow \mathcal{D}(t) \setminus \{J(t)\}$
  end if
  Pull arm $J(t)$, observe its reward observation
end for

Leader-coordinated algorithm for stochastic AMA2B. To illustrate how leader selection impacts the performance of the leader-follower scheme, we consider the *stochastic* AMA2B, where the asynchronous activation rounds of each agent m follows a Bernoulli process with known frequency parameter $\theta^{(m)}$, i.e., the agent pulls an arm when the realization of the Bernoulli random variable is equal to 1. Also, without loss of generality, we assume $1/\theta^{(m)} \in \mathbb{N}^+$ for every agent m. Let $n_k^{(\ell)}(t)$ and $\hat{\mu}_k^{(\ell)}(t)$ denote the number of pulls and the reward mean estimate for arm k of leader ℓ at time slot t. Let $d_k(t)$ denote the Kullback-Leibler Upper Confidence Bounds (KL-UCB) index [18] of arm k for leader ℓ at time t (as only leader uses KL-UCB index, we omit the $d_k^{(\ell)}(t)$'s superscript),

Algorithm 3 LF: Follower m's procedure

1: **Initialize:** $J(t) = 0$
2: **for all** $t \in \mathcal{T}^{(m)}$ **do**
3: **if** receive recommended arm $I(t)$ from leader **then**
4: $J(t) \leftarrow I(t)$
5: **end if**
6: Pull arm $J(t)$
7: **end for**

defined as $d_k(t) \coloneqq \sup\{q \geqslant 0 : n_k^{(\ell)}(t)\,\mathrm{kl}(\hat{\mu}_k^{(\ell)}(t), q) \geqslant \log t + 4\log\log t\}$, where $\mathrm{kl}(a, b)$ is the KL-divergence between two Bernoulli distributions with means a and b.

We present the leader's decision-making procedure in Algorithm 2. Based on the KL-UCB index, we construct an exploration arm set $\mathcal{D}(t)$ containing arms with large KL-UCB indices. During each leader active decision round $t \in \mathcal{T}^{(\ell)}$, if the exploration arm set $\mathcal{D}(t)$ is empty (i.e., all arms in the set have been explored once), the leader ℓ updates the empirical mean estimates $\hat{\mu}_k^{(\ell)}(t)$, number of pulling times $n_k^{(\ell)}(t)$, and KL-UCB indices $d_k(t)$ of all arms (Line 5), as well as its empirical optimal arm estimate $I(t)$, which is the arm with the largest reward mean estimate (ties are broken arbitrarily) (Line 6). If the empirical optimal arm $I(t)$ changes (i.e., differs from the empirical optimal arm in the previous active time slot, denoted as I'), the leader updates this new arm recommendation $I(t)$ to all followers (Lines 7-8). Then, leader ℓ pulls the empirical optimal arm $I(t)$ once (Line 11). After that, the leader updates the exploration arm set $\mathcal{D}(t)$ that contains all arms whose KL-UCB indices $d_k(t)$ are greater than the largest empirical reward mean (i.e., arm $I(t)$'s reward mean estimate) (Line 12). Otherwise, if the exploration arm set $\mathcal{D}(t)$ is non-empty, the leader picks an arm with the largest KL-UCB index in $\mathcal{D}(t)$ to explore and then eliminates this explored arm from set $\mathcal{D}(t)$. Meanwhile, the followers keep pulling the most recently recommended arm by the leader during their active decision rounds (Algorithm 3).

Regret and communication analysis for stochastic AMA2B**.** Next, we present regret and communication upper bounds of Algorithm 2 for stochastic AMA2B for picking any agent ℓ as the leader in Proposition 4.

Proposition 4. *For stochastic* AMA2B*, with any agent* $\ell \in \mathcal{M}$ *as the leader, and* $0 < \lambda < \min_{k>1} \frac{\mu_{k-1}-\mu_k}{4}$*, Algorithm 2's regret and communication is upper bounded by*

$$R(T) \leqslant \sum_{k\neq 1} \frac{\Delta_k(\log T + 4\log\log T)}{\mathrm{kl}(\mu_k + \lambda, \mu_1 - \lambda)} + \frac{\sum_{m\in\mathcal{M}} \theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda^{-2} + 57K) + K\lambda^{-2}, \tag{9}$$

$$C(T) \leqslant 4K^2M\lambda^{-2} + 114KM. \tag{10}$$

Proof of Proposition 4 is given in Appendix B.1. The second term of the regret bound in (9) corresponds to the cost due to asynchronicity. The term $(2K^2\lambda^{-2} + 57K)$ upper bounds the number of leader active decision rounds at which the leader recommends the false optimal arm. The multiplicative factor $\sum_{m\in\mathcal{M}} \theta^{(m)}/\theta^{(\ell)}$ — the ratio of the summation of all agent activation frequencies over the leader activation frequency — transfers the number of active decision rounds in which the leader makes false optimal arm recommendations to the expected total number of active rounds that all agents pull these false recommendations. In the stochastic AMA2B scenario, the multiplicative factor is minimized by choosing the agent with the largest activation frequency as the leader, in which case the term $\sum_{m\in\mathcal{M}} \theta^{(m)}/\theta^{(\ell)}$ is upper bounded by M. If the selected leader has a rather low activation frequency, this term may be fairly large.

Algorithm 4 `Relay`: Leader relay

Input: time horizon T, number of agents M, switch budget S, $\eta_t \leftarrow 2/\sqrt{t}$
Initialize: mini-batch $T' \leftarrow \lceil T/S \rceil$, $t \leftarrow 1$, $\ell(t) \leftarrow$ uniformly draw one agent from $\mathcal{M}$
for $t \in \mathcal{T}$ **do**
  Observe the activation status $b^{(\ell(t))}(t)$ of agent $\ell(t)$
  if $t | T'$ **then** ▷ When $t = T', 2T', \ldots$
    ▷ Follow Tsallis-INF to update counters and distribution $\boldsymbol{p}(t+T')$
    $\tilde{b}^{(m)}(t) \leftarrow \begin{cases} \sum_{\tau=t-T'}^{t} b^{(\ell(t))}(\tau)/p^{(\ell(t))}(t), & \text{if } m = \ell(t) \\ 0, & \text{otherwise} \end{cases}$ for all agent m
    $B^{(m)}(t) \leftarrow \sum_{\tau:\tau|T',\tau\leqslant t} \tilde{b}^{(m)}(\tau)/T'$ for all agent m ▷ Scale down by mini-batch T'
    $p^{(m)}(t+T') \leftarrow 4(\eta_t(B^{(m)}(t) - C(t)))^{-2}$ for all agent m,
      where $C(t)$ is a normalization parameter such that $\sum_{m\in\mathcal{M}} p^{(m)}(t+T') = 1$
    Draw next leader $\ell(t+1)$ according to the distribution $\boldsymbol{p}(t+T') = (p^{(m)}(t+T'))_{m\in\mathcal{M}}$
  else
    $\ell(t+1) \leftarrow \ell(t)$ ▷ Fix the leader agent inside a mini-batch
  end if
end for

In order words, to avoid a large second term in the regret upper bound, one should choose a *competent leader* ℓ whose activation frequency is in the same order as those of followers, so that the regret cost due to the leader's false arm recommendation is not large. However, in general `AMA2B`, one does not know agents' activation frequencies a prior, and thus fixing one leader is not a good strategy anymore.

4.2 `Relay`: A Leader Relay Algorithm

In this subsection, we present a *leader relay* algorithm that dynamically switches the leadership among agents such that the sequence of leaders composes a "competent leader". Denote by $b^{(m)}(t)$ the activation status of agent m at time slot t: $b^{(m)}(t) = 1$ if agent m is active at time t; otherwise, $b^{(m)}(t) = 0$. Denote $\ell(t)$ as the leader at time slot t. The aim of leader relay can be roughly interpreted as maximizing the total number of active decision rounds of the leader sequence, i.e., $\max \sum_t b^{(\ell(t))}(t)$, facing agents with adversarial activations. This task can be cast as *adversarial bandits* [5]. The term $b^{(m)}(t)$ corresponds to the reward of a virtual arm (agent) m, and the term $\ell(t)$ as the pulled virtual arm (leader agent) at time slot t. The virtual arm (leader agent) sequence chosen by the leader relay algorithm together composes the "leader". If two consecutive pulled arms (agents) are different, i.e., $\ell(t-1) \neq \ell(t)$, we call it a *leadership switch*. As switches incur communication costs for transferring the contents of algorithm registers, we need to devise an adversarial bandit algorithm with low switches.

To ensure that the leader sequence is competent, the leader relay algorithm needs to guarantee that the activation frequency of the leader sequence is comparable to any single agent's activation frequency. Specifically, the leader sequence needs to have $\Theta(\tau)$ active decision rounds for any time slot $\tau \leqslant T$, that is, $\sum_{t=1}^{\tau} b^{(\ell(t))}(t) = \Theta(\tau)$. To achieve that, we need to pick an adversarial bandits algorithm with an anytime $O(\sqrt{MT})$ regret upper bound (see the Proof sketch of Theorem 5 below). We use the `Tsallis-INF` (implicitly normalized forecaster) algorithm [65] as the base algorithm, and then employ the mini-batch technique [2, 3] to convert `Tsallis-INF` to an algorithm with low switches. Denote by $S \in \mathbb{N}^+$ the number of switches allowed (switch budget) in leader relay, and set the mini-batch length $T' = \lceil T/S \rceil$. The mini-batch technique divides the given time horizon

T into around $\lfloor T/T' \rfloor$ mini-batches, each of which contains T' time slots. Then, the *mini-batched* algorithm treats each mini-batch as one "time slot" (i.e., fixes its action in the mini-batch), where the reward of an action on one "time slot" is the cumulative reward of constantly choosing this action over all T' time slots within the mini-batch in the *original* algorithm. Next, we present the details of the mini-batched `Tsallis-INF` (i.e., leader relay) algorithm.

The leader relay (`Relay`) algorithm, presented in Algorithm 4, is executed by the leader sequence $\{\ell(t)\}_{t\in\mathcal{T}}$ for all time slots in $\mathcal{T}$. In each time slot $t \in \mathcal{T}$, the leader agent $\ell(t)$ collects its activation status $b^{(\ell(t))}(t)$ (Line 4). Only at the end of each mini-batch, i.e., $t|T'$ or when $t = T', 2T', \ldots$, the leadership switch may happen (Line 5). For these time slots $t = T', 2T', \ldots$, the algorithm performs the `Tsallis-INF` algorithm to decide the next leader agent (Lines 6–9). Specifically, the algorithm first updates the estimates $\tilde{b}^{(m)}(t)$ according to the collected activation statuses $\{b^{(\ell(t))}(\tau)\}_{t-T'\leqslant\tau\leqslant t}$ in this past mini-batch for this leader agent $\ell(t)$ and set the estimates of other follower agents $m \neq \ell(t)$ as zero (Line 6). Then, according to all estimates $\{\tilde{b}^{(m)}(\tau)\}_{\tau:\tau|T',\tau\leqslant t}$ in all past mini-batches, the algorithm updates a counter $B^{(m)}(t)$ for all agents $m \in \mathcal{M}$, scaling down by a T' factor for the mini-batch technique (Line 7). Next, following the implicitly normalized forecaster (`INF`) principle [4, 65], the algorithm updates the virtual arm (agent) sample probability in Line 8 and samples the next leader agent according to this probability distribution in Line 9. For other time slots not at the end of a mini-batch, the algorithm fixes on the same leader agent (Line 11).

Next, we provide the leader relay algorithm's theoretical guarantee in Theorem 5.

Theorem 5 (Leader relay guarantees competent leader sequence). *Given $S = C_1M^3$ leadership switch budget where $C_1 > 0$ is a universal constant, for any time slot $\tau \leqslant T$, the total number of active decision rounds of the leader sequence chosen by Algorithm 4 is at least $\tau/2M$, or formally, $\sum_{t=1}^{\tau} b^{(\ell(t))}(t) \geqslant \frac{\tau}{2M}$, where $\ell(t)$ is the leader chosen by Algorithm 4 at time slot t.*

Theorem 5 guarantees that at any time slot τ, the leaders in the sequence are active for at least $\tau/2M$ rounds, composing a competent leader sequence. Hence, the multiplicative factor $\sum_{m\in\mathcal{M}} \theta^{(m)}/\theta^{(\ell)}$ in the second term of (9) is upper bounded by $2M^2$ in the general `AMA2B` setting. The introduction of Algorithm 4 and its corresponding Theorem 5 to leader selection is especially designed to address the asynchronous `AMA2B` setting and thus an unique contribution, which is not covered by existing synchronous settings. The proof of Theorem 5 relies on a novel combination from two distinct areas: adversarial bandits and asynchronous agents.

Proof sketch for Theorem 5. Detail proof of Theorem 5 is presented in §4.2.1. Below, we provide an intuitive proof sketch. The original `Tsallis-INF` algorithm achieves a regret upper bound of $O(\sqrt{MT})$ for adversarial bandits. By extending it with the mini-batch technique, where switches occur at most $S = O(M^3)$ times, the leader-relay algorithm aggregates every $O(T/M^3)$ consecutive time slots into a mini-batch. Within each mini-batch, `Tsallis-INF` makes a single decision at the start and maintains it throughout the mini-batch. This results in a mini-batched `Tsallis-INF` algorithm with $O(T/M)$ regret. By choosing $S = 64M^3$, this regret becomes $T/2M$. Given that the agent with the highest overall activation frequency (the benchmark for the adversarial bandit's regret) will have at least T/M action rounds, the leader-relay algorithm guarantees that the leader (sequence) is active for at least $T/2M$ rounds, which is comparable to the active frequency of any single agent, and therefore, is competent.

4.2.1 Detail Proof of Theorem 5. We first prove a lemma on the leader relay algorithm's "regret" upper bound.

Algorithm 5 LF-Relay: Leader's procedure

for all $t \in \mathcal{T}$ **do**
  if $t \in \mathcal{T}^{(\ell(t))}$ **then** ▷ Decision-making
    Leader $\ell(t)$ runs the decision making procedure (Algorithm 2)
  end if
  Run leader relay (Algorithm 4) to decide the next leader $\ell(t+1)$
  if $\ell(t) \neq \ell(t+1)$ **then** ▷ Leadership switch
    Transfer register contents of Algorithms 2 and 4 to new leader $\ell(t+1)$.
  end if
end for

Lemma 6. *Algorithm 4 guarantees that, for any time horizon T,*

$$\max_{m\in\mathcal{M}} \sum_{t=1}^{T} b^{(m)}(t) - \sum_{t=1}^{T} b^{(\ell(t))}(t) \leqslant \frac{4T\sqrt{M}}{\sqrt{S}}, \tag{11}$$

where $\ell(t)$ is the leader choosing by Algorithm 4 at time slot t.

Proof of Lemma 6. The lemma is proved by applying the mini-batch technique Altschuler and Talwar [2, Theorem 6] to the Tsallis-INF algorithm [65, Theorem 1] for adversarial bandits.

We first recall the original regret upper bound of Tsallis-INF in adversarial bandits [65, Theorem 1], which is

$$\max_{m\in\mathcal{M}} \sum_{t=1}^{T} b^{(m)}(t) - \sum_{t=1}^{T} b^{(\ell'(t))}(t) \leqslant 4\sqrt{MT},$$

where the $\ell'(t)$ is the arm/agent chosen by the original Tsallis-INF algorithm.

The leader Relay algorithm in Algorithm 4 is a T'-mini-batched version of Tsallis-INF. The mini-batch size is $T' = \lceil T/S \rceil$, and the leader may be switched by the Tsallis-INF algorithm every T' decision rounds. According to Altschuler and Talwar [2, Theorem 6], the regret upper bound of the mini-batched algorithm is

$$\max_{m\in\mathcal{M}} \sum_{t=1}^{T} b^{(m)}(t) - \sum_{t=1}^{T} b^{(\ell(t))}(t) \leqslant 4\sqrt{MS} \times \frac{T}{S} \leqslant \frac{4T\sqrt{M}}{\sqrt{S}},$$

where the $4\sqrt{MS}$ is because the mini-batched version has in total S batches, and the $\frac{T}{S}$ is due to that the reward of a mini-batch is scaled up by the mini-batch size T'. □

Notice that at least one active agent exists in each time slot. Hence, the highest number of active decision rounds of a single agent is at least T/M, i.e.,

$$\max_{m\in\mathcal{M}} \sum_{t=1}^{T} b^{(m)}(t) \geqslant \frac{T}{M}.$$

Substituting the above inequality and $S = 64M^3$ to (11) in Lemma 6 yields

$$\sum_{t=1}^{T} b^{(\ell(t))}(t) \geqslant \max_{m\in\mathcal{M}} \sum_{t=1}^{T} b^{(m)}(t) - \frac{4T\sqrt{M}}{\sqrt{S}} \geqslant \frac{T}{2M}.$$

4.3 LF-Relay: Leader-Coordinated Bandit Algorithm & Its Theoretical Analysis

Combining the leader relay policy in Algorithm 4 and the leader decision procedure in Algorithm 2, we propose the leader-coordinated AMA2B algorithm in Algorithm 5. For each time slot t, if active at this time t, this leader $\ell(t)$ runs the leader procedure in Algorithm 2 (Line 3). Then, the leader $\ell(t)$ runs the leader relay algorithm to decide the next leader $\ell(t+1)$ (Line 5). If there is a leadership switch, i.e., $\ell(t) \neq \ell(t+1)$, then leader $\ell(t)$ transfers its data, i.e., all register contents of Algorithms 2 and 4, to new leader $\ell(t+1)$ (Line 7).

In Theorem 7, we present the regret and communication analysis of LF-Relay in Algorithm 5. Proof of Theorem 7 is presented in Appendix B.2.

Theorem 7. *For general* AMA2B*, given* $0 < \lambda < \min_{k>1} \frac{\mu_{k-1}-\mu_k}{4}$ *and the number of leadership switches in Algorithm 4 as* $S = 64M^3$*, Algorithm 5's regret and communication cost satisfy,*

$$R(T) \leqslant \sum_{k>1} \frac{\Delta_k(\log T + 4\log\log T)}{\mathrm{kl}(\mu_k+\lambda, \mu_1-\lambda)} + 2M^2(2K^2\lambda^{-2} + 57K) + K\lambda^{-2}, \tag{12}$$

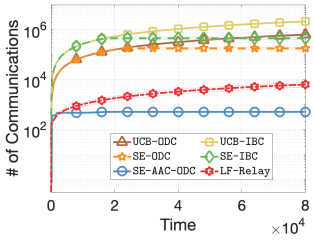
$$C(T) \leqslant 4K^2M\lambda^{-2} + 114KM + 64M^3. \tag{13}$$

Communication and regret bounds comparison between LF-Relay and SE-AAC-ODC. The LF-Relay algorithm achieves a time-independent constant communication cost $O(K^2M\Delta_2^{-2} + M^3)$ (assuming $\lambda = \Delta_2/4$ in (13)). But its order is worse than the $O(KM\log\Delta^{-1})$ cost of the fully distributed SE-AAC-ODC in (6). On the other hand, letting $T \to \infty$ and $\lambda \to 0$, the regret upper bound in (12) becomes $\limsup_{T\to\infty} \frac{R(T)}{\log T} \leqslant \sum_{k>1} \frac{\Delta_k}{\mathrm{kl}(\mu_k,\mu_1)}$. This tightly matches the MA2B's regret lower bound [57, §1.2], which implies that LF-Relay is (asymptotically) optimal. As bounded in terms of the fine-grained KL-divergence, this regret of LF-Relay in (12) is tighter than that of SE-AAC-ODC in (5) based on the coarse-grained reward gaps Δ_k. To sum up, although both algorithms achieve near-optimal regret and constant communications, LF-Relay *has a better regret bound, while* SE-AAC-ODC *enjoys lower communications.*

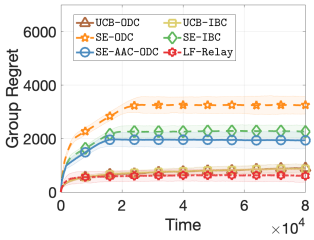
Beyond Bernoulli reward distributions. While in the above analysis of the fully distributed and leader-coordinated algorithms we focus on Bernoulli reward distributions, the proposed algorithms can be extended to general sub-Gaussian reward distributions. The fully distributed algorithm (SE-AAC-ODC) only needs to modify a multiplicative factor of its confidence interval according to the sub-Gaussian scale, and the leader-coordinated algorithm (LF-Relay) needs to replace its KL-UCB index with the *empirical* KL-UCB index [18]. However, the analysis of the leader-coordinated algorithm needs a non-trivial modification as the analysis related to the empirical KL-UCB index is much more involved.

5 NUMERICAL EXPERIMENTS

In this section, we first simulate a multi-server recommendation system application to verify the performance of the proposed algorithms. For recommendation servers, their decision times/rounds depend on when and how often clients visit their corresponding webpage and are often different, unknown, and asynchronous. Specifically, we numerically study M servers (i.e., agents) cooperatively recommending K advertisements (i.e., arms) with Bernoulli rewards whose reward means are uniformly randomly taken from the advertisement recommendation dataset, Ad-Click [35] from Kaggle. We let server m's non-stationary client request arrival times follows a sine function, $\sin(\theta_m + t/30)$, where the phase shifts $\theta_m = m/5$, $m \in \{1, ..., M\}$ differ for different servers. We report the number of communications and regrets after $T = 80\,000$ time slots averaged over 30 independent trials, and we plot the standard deviation as the shaded area.

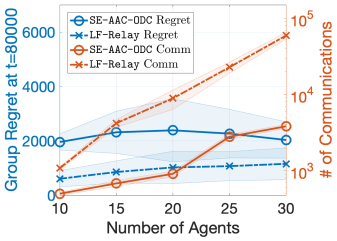


(a) Communications

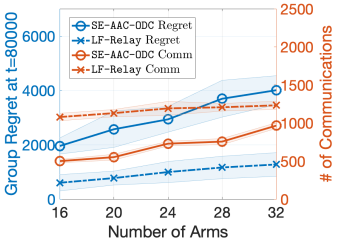


(b) Regrets

Fig. 3. Comparison with baselines

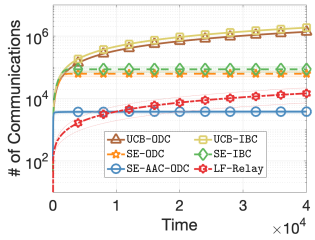


(a) Increase Agents

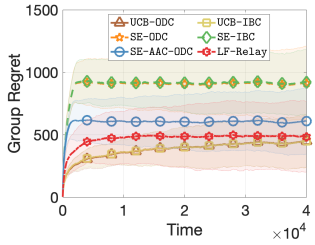


(b) Increase Arms

Fig. 4. Vary AMA2B parameters



(a) Communications



(b) Regrets

Fig. 5. Cognitive Radio Simulation

We compare our SE-AAC-ODC with $\alpha = 5$ and LF-Relay with $S = M^3$ to prior algorithms from [23, 59], namely UCB-ODC, UCB-IBC, SE-ODC, and SE-IBC, with $\alpha = 3$ (the definition of their α parameter is different), when there are $M = 10$ servers recommending $K = 16$ advertisements. The simulation results are presented in Figure 3. Figure 3a shows that our proposed algorithms, LF-Relay and SE-AAC-ODC, incur significantly fewer communication rounds than other baselines, validating our constant communication theoretical upper bounds. Figure 3b shows that LF-Relay achieves the smallest regret among all algorithms, and SE-AAC-ODC has the smallest regret among all SE-based algorithms. Figures 3a and 3b together show that, among the SE-AAC-ODC and LF-Relay, SE-AAC-ODC incurs fewer communications round while has larger regret and LF-Relay incurs a little bit more communications while achieves smaller regret.

We then compare SE-AAC-ODC with $\alpha = 5$ with LF-Relay with $S = M^3$ in two scenarios: 1) the number of advertisements fixed at $K = 16$, and the number of servers increases from $M = 10$ to $M = 30$ with step size 5, and 2) the number of servers fixed at $M = 10$, and the number of advertisements increases from $K = 16$ to $K = 32$ with step size 4. The results are presented in Figure 4. Figure 4a shows that the number of servers (i.e., agents) influences the number of communications more than group regret, especially for LF-Relay. Figure 4b shows that the number of advertisements (i.e., arms) has a moderate impact on both communications and regrets for both SE-AAC-ODC and LF-Relay. Figure 4 further validates that, among the two proposed algorithms, SE-AAC-ODC enjoys lower communications, while LF-Relay wins in regret.

Finally, we simulate a popular multi-agent bandit application: the cognitive radio network [45]. The setup involves $M = 10$ agents and $K = 10$ channels, where the agents' activation patterns follow ON/OFF two-state Markov chains [25] with transition probabilities from Modi et al. [45, Table II]. The arm reward means are determined based on the channel availability rates from Cai et al. [17, Table 1]. Figure 5 presents the regret and communication performance of the proposed SE-AAC-ODC algorithm ($\alpha = 2$) and LF-Relay ($S = 4M^4$), alongside baseline algorithms UCB-ODC, UCB-IBC, SE-ODC, and SE-IBC ($\alpha = 3$). The results are averaged over 60 independent trials, with the

corresponding standard deviations represented as shaded areas. Figure 5a confirms that both LF-Relay and SE-AAC-ODC incur significantly fewer communication rounds than the baseline algorithms. Additionally, Figure 5b indicates that all algorithms exhibit larger regret standard deviations due to increased randomness in agent activation times. Consistent with earlier simulations, among the proposed algorithms, SE-AAC-ODC demonstrates superior communication efficiency, while LF-Relay achieves lower regret.

6 CONCLUSION AND FUTURE DIRECTIONS

This paper addresses the asynchronous multi-agent multi-armed bandits problem by introducing two innovative algorithms: SE-AAC-ODC, designed for fully distributed scenarios, and LF-Relay, intended for leader-coordinated scenarios. Both algorithms achieve near-optimal regret and have horizon-independent communication costs. Specifically, SE-AAC-ODC features a tighter communication upper bound, whereas LF-Relay provides a tighter regret upper bound.

While both algorithms' regret upper bounds align with the asymptotic lower bound (with SE-AAC-ODC being slightly off due to missing constants), evaluating the effectiveness of their communication upper bounds remains challenging due to the absence of corresponding lower bounds. Existing lower bounds are limited to synchronous settings, such as the $\Omega(M)$ bound from Wang et al. [58, Theorem 2], and are also somewhat loose compared to the communication upper bounds even in synchronous settings, as noted by [57, 61]. As synchronous MA2B is a special case of asynchronous AMA2B, it is reasonable to expect that the communication lower bounds for the asynchronous setting will be at least as high as those for the synchronous setting, i.e., a $\Omega(M)$ lower bound is expected for asynchronous AMA2B. Comparing with this lower bound shows that the communication upper bounds of SE-AAC-ODC is tight in terms of the number of agents M, while the communication upper bound of LF-Relay is not tight. A promising direction for future research is to establish more sophisticated communication lower bounds for the asynchronous multi-agent multi-armed bandits problem.

ACKNOWLEDGEMENT

This research is supported by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 (IoBT CRA). The work of John C.S. Lui was supported in part by the RGC SRFS2122-4S02. The work of Mohammad Hajiesmaili is supported by NSF CNS-2325956, CAREER-2045641, CPS-2136199, CNS-2102963, and CNS-2106299. The work of Xutong Liu was partially supported by a fellowship award from the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK PDFS2324-4S04). Lin Yang would like to thank the support from NSFC (no. 62306138), JiangsuNSF (no. BK20230784), and the Inovation Program of State Key Laboratory for Novel Software Technology at Nanjing University (no. ZZKT2024B15). Lin Yang is the corresponding author.

REFERENCES

[1] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. 2006. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer networks* 50, 13 (2006), 2127–2159.

[2] Jason M Altschuler and Kunal Talwar. 2021. Online Learning over a Finite Action Set with Limited Switching. *Mathematics of Operations Research* 46, 1 (2021), 179–203.

[3] Raman Arora, Ofer Dekel, and Ambuj Tewari. 2012. Online Bandit Learning against an Adaptive Adversary: From Regret to Policy Regret *(ICML'12)*. Omnipress, Madison, WI, USA, 1747–1754.

[4] Jean-Yves Audibert and Sébastien Bubeck. 2009. Minimax policies for adversarial and stochastic bandits. In *COLT*. 217–226.

[5] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32, 1 (2002), 48–77.

[6] Maryam Aziz, Emilie Kaufmann, and Marie-Karelle Riviere. 2021. On multi-armed bandit designs for dose-finding clinical trials. *The Journal of Machine Learning Research* 22, 1 (2021), 686–723.
[7] Yogev Bar-On and Yishay Mansour. 2019. Individual regret in cooperative nonstochastic multi-armed bandits. *Advances in Neural Information Processing Systems* 32 (2019).
[8] Debabrota Basu, Christos Dimitrakakis, and Aristide Tossou. 2019. Differential privacy for multi-armed bandits: What is it and what is its cost? *arXiv preprint arXiv:1905.12298* (2019).
[9] Lilian Besson and Emilie Kaufmann. 2018. Multi-player bandits revisited. In *Algorithmic Learning Theory*. PMLR, 56–92.
[10] Lilian Besson and Emilie Kaufmann. 2018. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971* (2018).
[11] Ilai Bistritz and Nicholas Bambos. 2020. Cooperative multi-player bandit optimization. *Advances in Neural Information Processing Systems* 33 (2020), 2016–2027.
[12] Ilai Bistritz and Amir Leshem. 2018. Distributed multi-player bandits-a game of thrones approach. *Advances in Neural Information Processing Systems* 31 (2018).
[13] Ilai Bistritz and Amir Leshem. 2021. Game of thrones: Fully distributed learning for multiplayer bandits. *Mathematics of Operations Research* 46, 1 (2021), 159–178.
[14] Etienne Boursier and Vianney Perchet. 2019. SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits. *Advances in Neural Information Processing Systems* 32 (2019).
[15] Sébastien Bubeck, Yuanzhi Li, Yuval Peres, and Mark Sellke. 2020. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Conference on Learning Theory*. PMLR, 961–987.
[16] Swapna Buccapatnam, Jian Tan, and Li Zhang. 2015. Information sharing in distributed stochastic bandits. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2605–2613.
[17] Kechao Cai, Xutong Liu, Yu-Zhen Janice Chen, and John CS Lui. 2018. An online learning approach to network application optimization with guarantee. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2006–2014.
[18] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. 2013. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics* (2013), 1516–1541.
[19] Nicolò Cesa-Bianchi, Tommaso Cesari, and Claire Monteleoni. 2020. Cooperative online learning: Keeping your neighbors updated. In *Algorithmic learning theory*. PMLR, 234–250.
[20] Nicol'o Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. 2016. Delay and cooperation in nonstochastic bandits. In *Conference on Learning Theory*. PMLR, 605–622.
[21] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba. 2017. Coordinated Versus Decentralized Exploration In Multi-Agent Multi-Armed Bandits.. In *IJCAI*. 164–170.
[22] Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. 2020. The gossiping insert-eliminate algorithm for multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3471–3481.
[23] Yu-Zhen Janice Chen, Lin Yang, Xuchuang Wang, Xutong Liu, Mohammad Hajiesmaili, John C.S. Lui, and Don Towsley. 2023. On-Demand Communication for Asynchronous Multi-Agent Bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3903–3930.
[24] Sayak Ray Chowdhury and Xingyu Zhou. 2022. Shuffle private linear contextual bandits. *arXiv preprint arXiv:2202.05567* (2022).
[25] László Csurgai-Horváth and János Bitó. 2011. Primary and secondary user activity models for cognitive wireless network. In *Proceedings of the 11th International Conference on Telecommunications*. IEEE, 301–306.
[26] Yuval Dagan and Crammer Koby. 2018. A better resource allocation algorithm with semi-bandit feedback. In *Algorithmic Learning Theory*. PMLR, 268–320.
[27] Abhimanyu Dubey et al. 2020. Cooperative multi-agent bandits with heavy tails. In *International Conference on Machine Learning*. PMLR, 2730–2739.
[28] Abhimanyu Dubey and AlexSandy' Pentland. 2020. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems* 33 (2020), 6003–6014.
[29] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
[30] Raphaël Féraud, Réda Alami, and Romain Laroche. 2019. Decentralized exploration in multi-armed bandits. In *International Conference on Machine Learning*. PMLR, 1901–1909.
[31] Jiafan He, Tianhao Wang, Yifei Min, and Quanquan Gu. 2022. A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. *Advances in neural information processing systems* 35 (2022), 4762–4775.
[32] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. 2013. Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems* 26 (2013).

[33] Jiyan Jiang, Wenpeng Zhang, Jinjie Gu, and Wenwu Zhu. 2021. Asynchronous decentralized online learning. *Advances in Neural Information Processing Systems* 34 (2021), 20185–20196.
[34] Pooria Joulani, András György, and Csaba Szepesvári. 2019. Think out of the" box": Generically-constrained asynchronous composite optimization and hedging. *Advances in Neural Information Processing Systems* 32 (2019).
[35] Kaggle. 2015. *Avito Context Ad Clicks*. Retrieved Jul. 22, 2022 from https://www.kaggle.com/c/avito-context-ad-clicks
[36] Kaggle. n.d.. *Avito Context Ad Clicks*. https://www.kaggle.com/c/avito-context-ad-clicks
[37] Ravi Kumar Kolla, Krishna Jagannathan, and Aditya Gopalan. 2018. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking* 26, 4 (2018), 1782–1795.
[38] Tze Leung Lai, Herbert Robbins, et al. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.
[39] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. 2016. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 167–172.
[40] Chuanhao Li and Hongning Wang. 2022. Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 6529–6553.
[41] Fengjiao Li, Xingyu Zhou, and Bo Ji. 2022. Differentially private linear bandits with partial distributed feedback. In *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. IEEE, 41–48.
[42] Udari Madhushani, Abhimanyu Dubey, Naomi Leonard, and Alex Pentland. 2021. One more step towards reality: Cooperative bandits with imperfect communication. *Advances in Neural Information Processing Systems* 34 (2021), 7813–7824.
[43] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. 2017. A survey on mobile edge computing: The communication perspective. *IEEE communications surveys & tutorials* 19, 4 (2017), 2322–2358.
[44] David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. 2019. Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems* 32 (2019).
[45] Navikkumar Modi, Philippe Mary, and Christophe Moy. 2017. QoS driven channel selection algorithm for cognitive radio network: Multi-user multi-armed bandit approach. *IEEE Transactions on Cognitive Communications and Networking* 3, 1 (2017), 49–66.
[46] Sara Momtazmanesh, Hans D Ochs, Lucina Q Uddin, Matjaz Perc, John M Routes, Duarte Nuno Vieira, Waleed Al-Herz, Safa Baris, Carolina Prando, Laszlo Rosivall, et al. 2020. All together to fight COVID-19. *The American journal of tropical medicine and hygiene* 102, 6 (2020), 1181.
[47] Wenbo Ren, Xingyu Zhou, Jia Liu, and Ness B Shroff. 2020. Multi-armed bandits with local differential privacy. *arXiv preprint arXiv:2007.03121* (2020).
[48] Hugo Richard, Etienne Boursier, and Vianney Perchet. 2024. Constant or Logarithmic Regret in Asynchronous Multiplayer Bandits with Limited Communication. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 388–396.
[49] Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. 2019. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 3 (2019), 1–35.
[50] Roshan Shariff and Or Sheffet. 2018. Differentially private contextual linear bandits. *arXiv preprint arXiv:1810.00068* (2018).
[51] Chengshuai Shi and Cong Shen. 2021. Federated multi-armed bandits. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.
[52] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. 2021. Heterogeneous Multi-player Multi-armed Bandits: Closing the Gap and Generalization. *Advances in Neural Information Processing Systems* 34 (2021).
[53] Balazs Szorenyi, Róbert Busa-Fekete, István Hegedus, Róbert Ormándi, Márk Jelasity, and Balázs Kégl. 2013. Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*. PMLR, 19–27.
[54] Jay Tenenbaum, Haim Kaplan, Yishay Mansour, and Uri Stemmer. 2021. Differentially private multi-armed bandits in the shuffle model. *Advances in Neural Information Processing Systems* 34 (2021), 24956–24967.
[55] Aristide Tossou and Christos Dimitrakakis. 2016. Algorithms for differentially private multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.
[56] Antonios Tsourdos, Brian White, and Madhavan Shanmugavel. 2010. *Cooperative path planning of unmanned aerial vehicles*. Vol. 32. John Wiley & Sons.
[57] Po-An Wang, Alexandre Proutiere, Kaito Ariu, Yassir Jedra, and Alessio Russo. 2020. Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4120–4129.
[58] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. 2020. Distributed Bandit Learning: Near-Optimal Regret with Efficient Communication. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[59] Lin Yang, Yu-Zhen Janice Chen, Mohammad Hajiesmaili, John C.S. Lui, and Don Towsley. 2022. Distributed Bandits with Heterogeneous Agents. In *In Proceedings of The IEEE International Conference on Computer Communications 2022*.
[60] Lin Yang, Yu-Zhen Janice Chen, Stephen Pasteris, Mohammad Hajiesmaili, John C.S. Lui, and Don Towsley. 2021. Cooperative stochastic bandits with asynchronous agents and constrained feedback. *Advances in Neural Information Processing Systems* 34 (2021), 8885–8897.
[61] Lin Yang, Xuchuang Wang, Lijun Zhang, Mohammad Hajiesmaili, John C.S. Lui, and Don Towsley. 2023. Cooperative Multi-agent Bandits: Distributed Algorithms with Optimal Individual Regret and Constant Communication Costs. *arXiv preprint arXiv:2308.04314* (2023).
[62] Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. 2020. Locally differentially private (contextual) bandits learning. *arXiv preprint arXiv:2006.00701* (2020).
[63] Xingyu Zhou and Sayak Ray Chowdhury. 2023. On Differentially Private Federated Linear Contextual Bandits. *arXiv preprint arXiv:2302.13945* (2023).
[64] Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. 2021. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*. 3–4.
[65] Julian Zimmert and Yevgeny Seldin. 2021. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research* 22, 28 (2021), 1–49.

A DEFERRED PROOFS FOR FULLY DISTRIBUTED ALGORITHM

A.1 Proof of Lemma 1

We pick an agent ℓ with the highest number of times of pulling arm k in the time slots from $\tau_k(t)$ to time t (break tie arbitrarily), that is,

$$\ell \coloneqq \arg\max_{m' \in \mathcal{M}} \sum_{s=\tau_k(t)}^{t} \mathbb{1}\{I^{(m')}(s) = k\} = \arg\max_{m' \in \mathcal{M}} \left(n_k^{(m')}(t) - n_k^{(m')}(\tau_k(t)) \right), \tag{14}$$

where $I^{(m')}(s)$ denotes the arm that agent m' pulls at time slot s. Recall that the estimate $\hat{\mu}_k^{(\ell)}(t)$ is obtained by averaging $n_k(\tau_k(t)) + n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t))$ samples, where $n_k(\tau_k(t))$ is the number of samples of arm k among all agents at time slot $\tau_k(t)$. Hence, the following equation holds with probability $1 - MT^{-3}$,

$$\begin{aligned}
|\hat{\mu}_k^{(\ell)}(t) - \mu_k| &\overset{(a)}{\leqslant} \mathsf{CR}(n_k(\tau_k(t)) + n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t))) \\
&\overset{(b)}{\leqslant} \mathsf{CR}(n_k(\tau_k(t))) \\
&\overset{(c)}{<} \alpha \mathsf{CR}(n_k(\tau_k(t)) + M(n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t)))) \\
&\overset{(d)}{<} \alpha \mathsf{CR}(n_k(t)),
\end{aligned}$$

where inequality (a) is proved below by Hoeffding's inequality and union bound, inequality (b) is due to that the confidence radius becomes larger with a smaller number of samples, inequality (c) is due to that the condition in Line 11 of Algorithm 1 is false at time slot t ($> \tau_k(t)$), and inequality (d) is because that the agent ℓ has the highest number of times of pulling arm k during $\tau_k(t)$ to t, that is, $n_k(\tau_k(t)) + M(n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t))) \geqslant n_k(t)$.

Below, we present the detailed steps for proving inequality (a) as follows,

$$
\begin{aligned}
&\mathbb{P}\left(|\hat{\mu}_k^{(\ell)}(t)-\mu_k| \leqslant \mathsf{CR}(n_k(\tau_k(t))+n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t)))\right)\\
=&\mathbb{P}\left(|\hat{\mu}_k^{(\ell)}(t)-\mu_k| \leqslant \sqrt{\frac{4\log T}{n_k(\tau_k(t))+n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))}}\right)\\
=&1-\mathbb{P}\left(|\hat{\mu}_k^{(\ell)}(t)-\mu_k| > \sqrt{\frac{4\log T}{n_k(\tau_k(t))+n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))}}\right)\\
\overset{(a1)}{\geqslant}&1-\sum_{n=1}^{M\cdot t}\mathbb{P}\left(|\hat{\mu}_k^{(\ell)}(t)-\mu_k| > \sqrt{\frac{4\log T}{n}}\middle| n_k(\tau_k(t))+n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))=n\right)\\
&\qquad\times\mathbb{P}\left(n_k(\tau_k(t))+n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))=n\right)\\
\geqslant&1-\sum_{n=1}^{M\cdot t}\mathbb{P}\left(|\hat{\mu}_k^{(\ell)}(t)-\mu_k| > \sqrt{\frac{4\log T}{n}}\middle| n_k(\tau_k(t))+n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))=n\right)\\
\overset{(a2)}{\geqslant}&1-\sum_{n=1}^{M\cdot t}T^{-4}\geqslant 1-MT^{-3},
\end{aligned}
$$

where inequality (a1) is due to union bound, and inequality (a2) is by applying Hoeffding's inequality.

B PROOFS FOR LEADER-COORDINATED BANDIT ALGORITHMS

B.1 Proof of Proposition 4

All over this proof, we only consider the reward mean estimates $\hat{\mu}_k^{(\ell)}(t)$ of the leader ℓ, and there is no appearance of other agents. Hence, we omit the (ℓ) superscript.

Step 1. Bound the number of times that the leader recommends the wrong optimal arm. Given $0<\lambda<\min_{k>1}\frac{(\mu_{k-1}-\mu_k)}{4}$, we first define several subsets of time slots that leader ℓ is active as follows,

$$
\begin{aligned}
\mathcal{A} &\coloneqq \{t\in\mathcal{T}^{(\ell)} : I(t)\neq 1\},\\
\mathcal{B} &\coloneqq \{t\in\mathcal{T}^{(\ell)} : |\hat{\mu}_{I(t)}(t)-\mu_{I(t)}|\geqslant\lambda\},\\
\mathcal{G} &\coloneqq \{t\in\mathcal{T}^{(\ell)} : d_1(t)<\mu_1(t)\},\\
\mathcal{H} &\coloneqq \{t\in\mathcal{A}\setminus(\mathcal{B}\cup\mathcal{G}) : |\hat{\mu}_1(t)-\mu_1|\geqslant\lambda\}.
\end{aligned}
$$

Lemma 8. *$\mathcal{A}\cup\mathcal{B}\subset\mathcal{B}\cup\mathcal{G}\cup\mathcal{H}$ and hence, $\mathbb{E}[|\mathcal{A}|]\leqslant\mathbb{E}[|\mathcal{B}|]+\mathbb{E}[|\mathcal{G}|]+\mathbb{E}[|\mathcal{H}|]$.*

Proof of Lemma 8. Let $t\in\mathcal{A}\setminus(\mathcal{B}\cup\mathcal{G})$. To prove this lemma, one only needs to show $t\in\mathcal{H}$, which can be derived as follows,

$$
\hat{\mu}_1(t)\overset{(a)}{\leqslant}\hat{\mu}_{I(t)}\overset{(b)}{\leqslant}\mu_{I(t)}+\lambda\overset{(c)}{\leqslant}\mu_1-\lambda,
$$

where inequality (a) is due to the definition of $I(t)$, inequality (b) is because $t\notin\mathcal{B}$, and inequality (c) is due to the definition of λ. □

Lemma 9. $\mathbb{E}[|\mathcal{B}|]+\mathbb{E}[|\mathcal{G}|]+\mathbb{E}[|\mathcal{H}|]\leqslant 2K^2\lambda^{-2}+57K$

Proof of Lemma 9. We first present two useful lemmas adapted from prior literature as follows,

Lemma 10 (Adapted from [57, Lemma 8]). *Let $k \in \mathcal{K}$, and $\zeta > 0$. Define $\mathcal{F}(t)$ the σ-algebra generated by arm rewards $\{X_k(s)\}_{1\leqslant s\leqslant t, k\in\mathcal{K}}$. Let $\mathcal{E}$ be a random set of rounds such that for all t, $\{t \in \mathcal{E}\} \in \mathcal{F}_{t-1}$. Assume that for all $t \in \mathcal{E}$, we have $n_k(t) \geqslant \zeta \sum_{s=1}^{t} \mathbb{1}\{t \in \mathcal{E}\}$. Then, for all $\lambda > 0$, we have*

$$\mathbb{E}\left[\sum_{t\geqslant 1} \mathbb{1}\{t \in \mathcal{E}, |\hat{\mu}_k(t) - \mu_k| \geqslant \varepsilon\}\right] \leqslant \frac{1}{\zeta\lambda^2}.$$

Lemma 11 (Adapted from [57, Lemma 9]). *Under Algorithm 2, we have*

$$\sum_{t\geqslant 1} \mathbb{P}[d_1(t) \leqslant \mu_1] \leqslant 57K.$$

Next, we respectively upper bound $\mathbb{E}[|\mathcal{B}|]$, $\mathbb{E}[|\mathcal{G}|]$, and $\mathbb{E}[|\mathcal{H}|]$.

Show $\mathbb{E}[|\mathcal{B}|] \leqslant K^2\lambda^{-2}$. We denote $\mathcal{B}_k \coloneqq \{t \in \mathcal{B} : I(t) = k\}$ for all arm $k \in \mathcal{K}$. Due to the exploration design of Algorithm 2, we have $n_k(t) \geqslant (1/K)\sum_{s=1}^{t} \mathbb{1}\{t \in \mathcal{B}_k\}$. Then applying Lemma 10 with $\mathcal{E} = \mathcal{B}_k$ and $\zeta = 1/K$, we have $\mathbb{E}[|\mathcal{B}_k|] \leqslant K\lambda^{-2}$. Therefore, with a union bound, we have $\mathbb{E}[|\mathcal{B}|] \leqslant \sum_{k\in\mathcal{K}} \mathbb{E}[|\mathcal{B}_k|] \leqslant K^2\lambda^{-2}$.

Show $\mathbb{E}[|\mathcal{G}|] \leqslant 57K$. Applying Lemma 11 leads to this upper bound.

Show $\mathbb{E}[|\mathcal{H}|] \leqslant K\lambda^{-2}$. Notice that $t \in \mathcal{H}$ guarantees that

$$d_1(t) \overset{(a)}{\geqslant} \mu_1 \overset{(b)}{\geqslant} \mu_{I(t)} + \lambda \overset{(c)}{\geqslant} \hat{\mu}_{I(k)}(t),$$

where inequality (a) is because $t \notin \mathcal{G}$, inequality (b) is due to the definition of λ, and inequality (c) is because $t \notin \mathcal{B}$. Since $d_1(t) \geqslant \hat{\mu}_{I(t)}(t)$, the optimal arm 1 is inside the exploration arm set $\mathcal{D}(t)$, and Algorithm 2 thus explore this arm at least once every K rounds, i.e., $n_k(t) \geqslant (1/K)\sum_{s=1}^{t} \mathbb{1}\{t \in \mathcal{H}\}$. Applying Lemma 10 with $\mathcal{E} = \mathcal{H}$ and $\zeta = 1/K$ yields $\mathbb{E}[|\mathcal{H}|] \leqslant K\lambda^{-2}$. □

Step 2. Bound the regret of leader exploring suboptimal arms. Denote $\mathcal{Q}_k \coloneqq \{t \in \mathcal{T}^{(\ell)} \setminus (\mathcal{A} \cup \mathcal{B}) : J(t) = k\}$ for suboptimal arm $k \neq 1$. We show that,

$$\mathbb{E}[|\mathcal{Q}_k|] \leqslant \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k + \lambda, \mu_1 - \lambda)} + \lambda^{-2}.$$

Denote $x_k(t) \coloneqq \sum_{s=1}^{t} \mathbb{1}\{t \in \mathcal{Q}_k\}$ as the number of times that $t \in \mathcal{Q}_k$ happens up to time t. We set $x_0 \coloneqq \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k+\lambda, \mu_1-\lambda)}$ as a threshold.

We then define two subset of $\mathcal{Q}_k$ as follows,

$$\mathcal{Q}_{k,1} \coloneqq \{t \in \mathcal{Q}_k : |\hat{\mu}_k(t) - \mu_k| \geqslant \lambda\},$$
$$\mathcal{Q}_{k,2} \coloneqq \{t \in \mathcal{Q}_k : x_k(t) \leqslant x_0\}.$$

Next, we show that $\mathcal{Q}_k \subseteq \mathcal{Q}_{k,1} \cup \mathcal{Q}_{k,2}$. Let $t \in \mathcal{Q}_k \setminus (\mathcal{Q}_{k,1} \cup \mathcal{Q}_{k,2})$. For this t, we have

$$d_k(t) \overset{(a)}{\geqslant} \hat{\mu}_{I(t)}(t) \overset{(b)}{=} \hat{\mu}_1(t) \overset{(c)}{\geqslant} \mu_1 - \lambda \overset{(d)}{>} \mu_k + \lambda \overset{(e)}{>} \hat{\mu}_k(t), \tag{15}$$

where inequality (a) is due to $t \in \mathcal{Q}_k$, inequality (b) is because $t \notin \mathcal{A}$, inequality (c) is due to $t \notin \mathcal{B}$, inequality (d) is due to the definition of λ, and inequality (e) is for $t \in \mathcal{Q}_{k,1}$. Since $t \notin \mathcal{Q}_{k,2}$, we also have

$$n_k(t) \geqslant x_k(t) > x_0. \tag{16}$$

Then, we have

$$x_0\,\mathrm{kl}(\hat{\mu}_k(t), \mu_1 - \lambda) \overset{(a)}{\leqslant} n_k(t)\,\mathrm{kl}(\hat{\mu}_k(t), \mu_1 - \lambda) \overset{(b)}{\leqslant} n_k(t)\,\mathrm{kl}(\hat{\mu}_k(t), d_k(t)) \overset{(c)}{\leqslant} \log T + 4\log\log T,$$

where inequality (a) is by (16), inequality (b) is by (15) and $kl(x, y)$ increases with respect to y when $x < y$, and inequality (c) is by the definition of KL-UCB index $d_k(t)$.

Substituting $x_0 = \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k+\lambda,\mu_1-\lambda)}$ into the above inequality leads to

$$\mathrm{kl}(\hat{\mu}_k(t), \mu_1 - \lambda) \leqslant \mathrm{kl}(\mu_k + \lambda, \mu_1 - \lambda).$$

Noticing that $\mathrm{kl}(x, y)$ decreases with respect to x when $x < y$, we have $\hat{\mu}_k(t) > \mu_k + \lambda$, which contradicts $t \notin \mathcal{Q}_{k,1}$. Therefore, we know $\mathcal{Q}_k \setminus (\mathcal{Q}_{k,1} \cup \mathcal{Q}_{k,2}) = \emptyset$, i.e., $\mathcal{Q}_k \subseteq \mathcal{Q}_{k,1} \cup \mathcal{Q}_{k,2}$.

Next, we upper bound $\mathbb{E}[|\mathcal{Q}_{k,1}|]$ and $\mathbb{E}[|\mathcal{Q}_{k,2}|]$. To bound $\mathbb{E}[|\mathcal{Q}_{k,1}|]$, we apply Lemma 10 with $\mathcal{E} = \mathcal{Q}_{k,1}$ and $\zeta = 1$ (notice that the arm k is played at most once after each $\mathcal{D}(t)$ renewing), then we have $\mathbb{E}[|\mathcal{Q}_{k,1}|] \leqslant \lambda^{-2}$. For $\mathbb{E}[|\mathcal{Q}_{k,2}|]$, we have

$$\mathbb{E}[|\mathcal{Q}_{k,2}|] \leqslant x_0 = \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k + \lambda, \mu_1 - \lambda)}.$$

Combining both upper bound together yields the upper bound for $\mathbb{E}[|\mathcal{Q}_k|]$.

Step 3. Bound the total regret in stochastic case. We note that the regret costs due to Steps 1 and 2 are orthogonal. For Step 1, the total regret cost of the leader ℓ is upper bounded as follows,

$$1 \cdot \mathbb{E}[|\mathcal{A} \cup \mathcal{B}|] \leqslant 2K^2\lambda^{-2} + 57K.$$

When the leader makes wrong arm recommendations, the rest agents (followers) would also pull suboptimal arms and thus pay regret costs. Since the leader makes a wrong recommendation for $|\mathcal{A}|$ active decision rounds, the total arm pulls during these leader active rounds are $\sum_{m\neq\ell}\theta^{(m)}/\theta^{(\ell)}|\mathcal{A}|$. Hence, the total costs due to followers' mistaken pulling are upper bounded as follows,

$$\mathbb{E}\left[\frac{\sum_{m\neq\ell}\theta^{(m)}}{\theta^{(\ell)}}|\mathcal{A}|\right] \leqslant \frac{\sum_{m\neq\ell}\theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda^{-2} + 57K).$$

Summing the above two terms together yields an upper bound for the regret cost in Step 1 as follows,

$$\frac{\sum_{m\in\mathcal{M}}\theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda^{-2} + 57K).$$

For Step 2, the regret cost is only from the leaders' exploration, which is upper bounded as follows,

$$\sum_{k\neq 1}\Delta_k \cdot \mathbb{E}[|\mathcal{Q}_k|] \leqslant \sum_{k\neq 1}\Delta_k \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k + \lambda, \mu_1 - \lambda)} + K\lambda^{-2}.$$

Summing the regret costs from Steps 1 and 2 yields the regret upper bound as follows,

$$R(T) \leqslant \sum_{k\neq 1}\Delta_k \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k + \lambda, \mu_1 - \lambda)} + \frac{\sum_{m\in\mathcal{M}}\theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda^{-2} + 57K) + K\lambda^{-2}.$$

Communication. Communication only happens when the leader updates its arm recommendation to followers. Therefore, the total number times of recommendations is at most $2\mathbb{E}[|\mathcal{A}|]$ times, and in each communication, the leader sends $M-1$ messages to each followers. Hence, the actually communications are upper bounded by

$$C(T) \leqslant 4K^2M\lambda^{-2} + 114KM.$$

B.2 Proof of Theorem 7

The proof procedure of Theorem 7 are almost the same as that of Theorem 4. The only difference appears in bounding the regret cost due to Step 1 (from stochastic case to adversarial case).

For $t \in \mathcal{A}$—time slots in which the leader recommends the wrong arm, the leader agents pay at most $\mathbb{E}[|\mathcal{A} \cup \mathcal{B}|] \leqslant 2K^2\lambda^{-2} + 57K$ regret. Meanwhile, the cost due to followers pulling the suboptimal arm recommendation is upper bounded as follows,

$$(M-1) \cdot 2M \cdot (2K^2\lambda^{-2} + 57K),$$

where $M-1$ is the number of followers, and $2M$ is due to that the leader(s) chosen by Algorithm 4 are active at least every $2M$ decision rounds (see Theorem 5). Together, the regret cost due to step 1 becomes

$$2M^2(2K^2\lambda^{-2}+57K).$$

For communications, besides the leader recommendation, there are additional communication due to leader relay which is $S=64M^3$. Hence, the total communication cost is upper bounded as follows,

$$4K^2M\lambda^{-2}+114KM+64M^3.$$

C PRIVACY PROTECTION MECHANISMS

In multi-agent/distributed learning, it is critical to protect user's sensitive data from privacy risks, so as to encourage users to participate and collaborate with the agents in the learning process. Specifically, using the example mentioned in the introduction, imagine that M hospitals (i.e., agents) are collaborating with each other to conduct a clinical trial (i.e., bandit problem) to study how different treatments (i.e., arms) can affect a disease. Each hospital will choose a specific treatment for participating patients (i.e., users) based on past observations of treatment effects. However, due to privacy concerns about health data leakage, the patients may not be willing to share the actual effects of the treatments with the hospital, which prevents the agents from learning from patient feedback. To ensure privacy, we use the notion of differential privacy (DP), which is a de facto standard for reasoning about information leakage [29]. As defined in Definition 12, DP implies that for any neighboring records, after an ε-DP mechanism, their statistical behaviors are indistinguishable. In this case, it is difficult for any attacker to determine which record is the source of the given output.

Definition 12 (ε-differential privacy (ε-DP)). *For $\varepsilon>0$, a randomized mechanism $Q:\mathcal{D}\to\mathbb{R}^l$ is said to be ε-DP on $\mathcal{D}\subset\mathbb{R}^{l'}$ if for any neighboring $x,x'\in\mathcal{D}$ where $\sum_{i\in[l']}\mathbb{1}\{x_i\neq x'_i\}=1$ and a measurable subset E of $\mathbb{R}^l$, we have $\mathbb{P}\{Q(x)\in E\}\leqslant e^{\varepsilon}\cdot\mathbb{P}\{Q(x')\in E\}$.*

In the above definition of DP, ε is called the privacy budget, where smaller ε implies higher levels of privacy protection. When $\varepsilon=\infty$, there is no privacy protection.

Related works about differential privacy. There is extensive literature on studying differential privacy (DP) in multi-armed bandits [8, 24, 28, 41, 47, 50, 54, 55, 62, 63]. Our work uses the notion of local DP, which has been studied under the single-agent MAB [47], the single-agent linear contextual MAB [62], and distributed linear contextual bandits with partial feedback [41]. Ren et al. [47] studies the DP in single-agent MAB, which largely inspires our privacy protection mechanism. However, we consider a new asynchronous multi-agent MAB setting and prove a series of important privacy/regret/communication guarantees. Other notions for DP, including central DP [8, 55], shuffle DP [24, 54, 63], joint DP [28, 50, 63], etc., pose weaker privacy protections for MAB models, but they may achieve better trade-offs between regret, communication, and privacy. Studying these DP notions for AMA2B will be left as interesting future works.

In this work, we focus on user-level local differential privacy (LDP), which allows the algorithm to be agnostic about privacy. For this reason, this notion is presently adapted by Apple and Google for their large-scale systems[2]. In what follows, we give its formal definitions (§C.1), mechanisms (§C.2), theoretical guarantees (§C.3), and numerical experiments (§C.4).

[2]https://desfontain.es/privacy/real-world-differential-privacy.html

Algorithm 6 Convert-to-Bernoulli(ε) (CTB (ε)) Mechanism

Input: A random reward $r \in [0, 1]$ from the user
Output: An independent sample following $Q(r) \sim \text{Bernoulli}(\frac{re^{\varepsilon}+1-r}{1+e^{\varepsilon}})$

C.1 User-Level Local Differential Privacy (LDP)

To define user-level LDP, we need to specify the user model and the threat model to supplement the model described in §2. Specifically, we denote $U = (u^{(m)}(t))_{t\in\mathcal{T}^{(m)},m\in\mathcal{M}}$ be a sequence of $\sum_{m\in\mathcal{M}} T^{(m)}$ unique users. At each time $t \in \mathcal{T}^{(m)}$, user $u^{(m)}(t)$ comes to be served by agent m, who obtains reward $X_k^{(m)}(t)$ after the agent m recommends arm $k^{(m)}(t)$ to the user. In this context, each user $u^{(m)}(t)$ is identified by their reward responses given to all possible actions recommended to them.

For the threat model, the users do not trust the agent or other users, and the privacy burden lies on the user himself. In this case, each user $u^{(m)}(t)$ needs to (1) release a private version of their reward feedback $X_k^{(m)}(t)$ via a ε-DP mechanism Q, where Q could be privacy protection software or trusted third-party plugins embedded in the user's devices or terminals, and the non-private data will not leave the control of the user unless they are processed and released by these software/plugins; and (2) the agents should make recommendations only based on the private releases. For any random vectors X and Y, we use $X \in \sigma(Y)$ to denote that X is determined by Y plus some random factors independent of Y and the bandit instance. Formally, user-level LDP is defined as follows in Definition 13.

Definition 13 (User-level ε-LDP). *For $\epsilon > 0$, we say the learning process satisfies ε-LDP if (1) there is an ε-DP mechanism $Q : \mathcal{D} \to \mathbb{R}$, and (2) Action $k^{(m)}(t+1) \in \sigma\big(\big(k^{(j)}(s), Q(X^{(j)}_{k^{(j)}(s)}(s))\big)_{s\leqslant t, j\in\mathcal{M}}\big)$ for any agent m and time t.*

Note that user-level LDP is a strong DP guarantee in the sense that it ensures that any attacker (which could be any other user, an agent, or an adversary outside the agents) cannot infer too much about any user's sensitive information (e.g., preference, reward feedback) or determine whether an individual participated in the learning process. Other DP notions, such as agent-level DP (where the agent containing the user can be trusted) [63], will also be satisfied by the post-processing of DP data if user-level LDP is preserved [29].

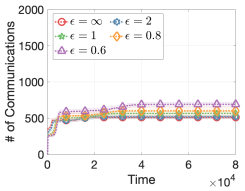
C.2 Convert-to-Bernoulli (CTB) Mechanism

To guarantee user-level LDP, one can adopt the widely-used Laplace mechanism [29] (i.e., adding Laplace noises to data records). However, adding Laplace noise makes the reward unbounded, which significantly increases regret and hinders the algorithm from obtaining constant communication costs. Inspired by [47], we use a simple yet effective Convert-to-Bernoulli (CTB) mechanism, which converts the rewards bounded in $[0, 1]$ to Bernoulli responses. The CTB mechanism is described in Algorithm 6.

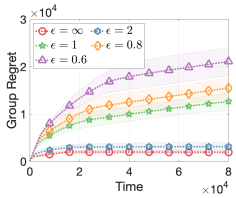
For our algorithm, CTB mechanism Q is agnostic to the algorithm, in the sense that we only need to change the way the agent obtains the reward of Line 9 of Algorithm 1. In particular, agent m now obtains arm k's reward observation $Q(X_k^{(m)}(t))$ with parameter ε as in Algorithm 6. The rest of the algorithm remains exactly the same.

C.3 Privacy, Regret, and Communication Guarantees

Here we present our privacy results, together with the new regret and communication cost. We also compare the non-private and private regret/communication costs.

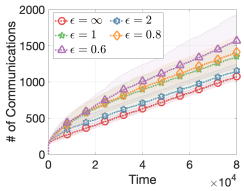


(a) Private comm

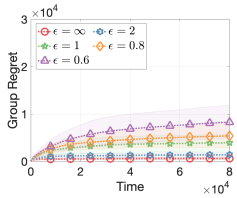


(b) Private regrets

Fig. 6. Performance of SE-AAC-ODC-CTB



(a) Private comm



(b) Private regrets

Fig. 7. Performance of LF-Relay-CTB

Theorem 14. SE-AAC-ODC *(Algorithms 1),* LF *(Algorithm 2) and* LF-Relay *(Algorithm 5) with CTB(ε) (Algorithm 6) all satisfy user-level ε-LDP.*

Theorem 15. *The regret and communication cost of Algorithm 1 with* CTB *(ε) (Algorithm 6) are upper bounded as follows,*

$$R(T) \leqslant \sum_{k>1} \frac{8(1+\alpha)^2 \log T}{\Delta_k} \cdot \left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)^2 + \sum_{k>1} M\Delta_k + KM^2,$$

$$C(T) \leqslant \sum_{k>1} 2M \log_\alpha \left(\frac{2(1+\alpha)}{\Delta_k} \cdot \left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)\right) + 2M \log_\alpha \left(\frac{2(1+\alpha)}{\Delta_2} \cdot \left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)\right) + 2KM^3.$$

Remark 1 (Regret comparison). *Compared with non-private* SE-AAC-ODC *in the fully distributed setting, the regret increases by at most a factor of* $\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)^2$ *factors, which is the cost for privacy protection. Since* $\frac{e^\epsilon+1}{e^\epsilon-1} \leqslant 1 + \frac{2}{\epsilon}$*, this factor approaches* 1 *as* ε *approaches infinity.*

Remark 2 (Communication cost comparison). *When we compare communication cost bounds with non-private algorithms, private* SE-AAC-ODC *algorithm increases by at most a factor of* $\log\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)$*, which still remains independent of time horizon* T*.*

Theorem 16. *Let* $\mu_{k,\varepsilon} = \frac{1}{2} + (\mu_k - \frac{1}{2}) \cdot \frac{e^\epsilon-1}{e^\epsilon+1}$ *and* $0 < \lambda_\varepsilon < \min_{k>1} \frac{(\mu_{k-1,\varepsilon} - \mu_{k,\varepsilon})}{4}$*. For general* AMA2B*, with the number of leadership switches in Algorithm 4 as* $S = 64M^3$*, Algorithm 5 with CTB(ε) (Algorithm 6)'s regret and communication cost satisfy,*

$$R(T) \leqslant \sum_{k>1} \frac{\Delta_k(\log T + 4\log\log T)}{\mathrm{kl}(\mu_{k,\varepsilon} + \lambda_\varepsilon, \mu_{1,\varepsilon} - \lambda_\varepsilon)} + 2M^2(2K^2\lambda_\varepsilon^{-2} + 57K) + K\lambda_\varepsilon^{-2}, \tag{17}$$

$$C(T) \leqslant 4K^2M\lambda_\varepsilon^{-2} + 114KM + 64M^3. \tag{18}$$

C.4 Numerical Experiments for Privacy Protection Mechanism

In this subsection, we report the empirical performance of SE-AAC-ODC-CTB algorithm with different privacy budgets $\epsilon = \infty, 2, 1, 0.8, 0.6$ in (see other detail setup in §5). Figures 6 and 7 show that the number of communications and regrets of SE-AAC-ODC-CTB and LF-Relay-CTB increase as the privacy level increases (i.e., as ϵ decreases) in AMA2B.

D PROOFS FOR PRIVACY PROTECTION OF FULLY-DISTRIBUTED AND LEADER-FOLLOWER ALGORITHMS

D.1 Proof of Theorem 14

Proof of Theorem 14. To prove the privacy guarantee, we need to check the two requirements of the Definition 13.

For requirement i), we rely on the key proposition as follows, which proves that CTB(ε) is ε-DP.

Proposition 17 (Lemma 5 of [47]). *CTB(ε) mechanism (Algorithm 6) is ε-DP on [0,1], and the returned sample follows the Bernoulli distribution with mean* $\mu_{k,\varepsilon} = \frac{1}{2} + (\mu_k - \frac{1}{2}) \cdot \frac{e^\varepsilon - 1}{e^\varepsilon + 1}$.

For requirement ii), since the private version of fully-distributed algorithms and leader-follower algorithms only use the observation from CTB, which satisfies requirement ii). □

D.2 Proof of Theorem 15

Proof of Theorem 15. Notice that after applying CTB, Proposition 17 ensures that the means of the private (Bernoulli) observations become $1 > \mu_{1,\varepsilon} > \mu_{2,\varepsilon} \geqslant \ldots \geqslant \mu_{K,\varepsilon} > 0$, where arm 1 is still the unique optimal arm. For the sub-optimal arm k, the sub-optimality gap becomes $\Delta_{k,\varepsilon} \triangleq \frac{e^\varepsilon - 1}{e^\varepsilon + 1}\Delta_k$ when bounding the number of times $n_k(T)$ until sub-optimal arms will not be pulled. For each pull of arm k, the algorithm will still pay Δ_k regret. Specifically, we slightly adapt the proof from Section A and show the following lemma.

Lemma 18. *Assume M agents independently sample an arm k associated with an i.i.d. reward process with unknown mean μ_k as Algorithm 1 (with threshold parameter $\alpha > 1$), and $n_k(t)$ is the total available samples of all agents. For any t, there* exists *an agent ℓ such that, with probability $1-\delta$, we have*

$$|\hat{\mu}_k^{(\ell)}(t) - \mu_{k,\varepsilon}| \leqslant \alpha \mathsf{CR}(n_k(t), \delta).$$

Proof. We pick an agent ℓ with the highest number of times of pulling arm i in the time slots from $\tau_k(t)$ to time t, that is,

$$\ell \in \arg\max_{m' \in \mathcal{M}} \sum_{s=\tau_k(t)}^{t} \mathbb{1}\{I^{(m')}(s) = i\}. \tag{19}$$

Note that the estimate $\hat{\mu}_k^{(\ell)}(t)$ is obtained by averaging $n_k(\tau_k(t)) + n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t))$ samples. Hence, the following equation holds with probability $1-\delta$.

$$\begin{aligned}
|\hat{\mu}_k^{(\ell)}(t) - \mu_{k,\varepsilon}| &\overset{(a)}{\leqslant} \mathsf{CR}(n_k(\tau_k(t)) + n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t)), \delta) \\
&\overset{(b)}{\leqslant} \mathsf{CR}(n_k(\tau_k(t)), \delta) \\
&\overset{(c)}{<} \alpha \mathsf{CR}(n_k(\tau_k(t)) + M(n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t))), \delta) \\
&\overset{(d)}{<} \alpha \mathsf{CR}(n_k(t), \delta),
\end{aligned}$$

where inequality (a) is by Hoeffding's inequality, inequality (b) is due to that the confidence radius becomes larger with a smaller number of samples, inequality (c) is due to that the condition in Line 11 is false at time slot t ($> \tau_k(t)$), and inequality (d) is because that the agent ℓ has the highest number of times of pulling arm k during $\tau_k^{(\ell)}(t)$ to t, that is, $n_k(\tau_k(t)) + M(n_k^{(\ell)}(t) - n_k^{(\ell)}(\tau_k(t))) \geqslant n_k(t)$. □

Now we follow steps 1-3 in Section A.

Step 1. Upper bound the probability of any Type-II decision occurring.

$$\begin{aligned}
&\mathbb{P}(\exists(k,m,t), |\hat{\mu}_k^{(m)}(t) - \mu_{k,\varepsilon}| > \mathsf{CR}(n_{k'}(\tau_k(t)) + (n_k^{(m)}(t) - n_k^{(m)}(\tau_{k'}(t))), \delta)) \\
\leqslant\ &\mathbb{P}(\exists(k,m,t,n), |\hat{\mu}_k^{(m)}(t) - \mu_{k,\varepsilon}| > \mathsf{CR}(n,\delta)) \\
\leqslant\ &\sum_{(k,m,t,n)\in(\mathcal{K}\times\mathcal{M}\times\mathcal{T}\times\mathcal{T})} \mathbb{P}(|\hat{\mu}_k^{(m)}(t) - \mu_{k,\varepsilon}| > \mathsf{CR}(n,\delta)) \\
\leqslant\ &\sum_{(k,m,t,n)\in(\mathcal{K}\times\mathcal{M}\times\mathcal{T}\times\mathcal{T})} \delta = KMT^2\delta.
\end{aligned}$$

Step 2. Upper bound the number of times of pulling suboptimal arms

Lemma 19. *At any time $t \leqslant T$, if the optimal arm lies in the candidate set and an agent makes a Type-I decision with pulling a suboptimal arm i, i.e., $I^{(m)}(t) = i$, we have $n_k(t) \leqslant \frac{2(1+\alpha)^2\log\delta^{-1}}{\Delta_{k,\varepsilon}^2}$. Therefore, the total number of pulling times of arm i in the whole time horizon is upper bounded as follows,*

$$n_k(T) \leqslant \frac{2(1+\alpha)^2\log\delta^{-1}}{\Delta_{k,\varepsilon}^2} + M.$$

Combining the results of Steps 1 and 2, the regret is upper bounded as follows,

$$\begin{aligned}
\mathbb{E}[R(T)] &\leqslant \sum_{k>1} n_k(T)\times\Delta_k + KMT^2\delta\times T \\
&\leqslant \sum_{k>1}\frac{2(1+\alpha)^2\log\delta^{-1}}{\Delta_k}\cdot\left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)^2 + \sum_{k>1} M\Delta_k + KMT^3\delta \\
&\overset{\delta\leftarrow T^{-3}}{\leqslant} \sum_{k>1}\frac{6(1+\alpha)^2\log T}{\Delta_k}\cdot\left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)^2 + 2KM.
\end{aligned}$$

Step 3. Upper bound communication costs For communication, the bound also holds by replacing Δ_k with $\Delta_{k,\varepsilon}$. Specifically, if there are any Type-II decisions, the total communication times is at most

$$KMT^3\delta.$$

Assume there is no Type-II decision. Following the proof of Lemma 3, we have a similar middle step (20): for any suboptimal arm k, denoting κ_k as the last time slot that the arm was pulled, we have

$$2(1+\alpha)\mathsf{CR}(n_k(\kappa_k),\delta) \geqslant \Delta_{k,\varepsilon}.$$

Hence, the last $\mathsf{ECR}_k^{(m)}(T)$ can be upper bounded as

$$\mathsf{ECR}_k^{(m)}(T) \overset{(a)}{=} \mathsf{CR}(n_k(\kappa_k),\delta) \geqslant \frac{\Delta_{k,\varepsilon}}{2(1+\alpha)},$$

where inequality (a) is because after round κ_k there is no further pulling on arm k. Recall the initial $\mathsf{ECR}_k^{(m)}(0) = 1$. The total number of times of communication on arm i is upper bounded as follows,

$$\log_\alpha\left(\frac{\mathsf{ECR}_k^{(m)}(0)}{\mathsf{ECR}_k^{(m)}(T)}\right) \leqslant \log_\alpha\left(\frac{2(1+\alpha)}{\Delta_{k,\varepsilon}}\right).$$

Since all arms are pulled in a round-robin manner, the communication cost on the optimal arm is upper bounded by $\log_\alpha\left(\frac{2(1+\alpha)}{\Delta_{2,\varepsilon}}\right)$ where $\Delta_{2,\varepsilon}$ is the smallest reward gap.

Summing the above two type cases yields the communication upper bound as follows,

$$\sum_{k>1}\log_\alpha\left(\frac{2(1+\alpha)}{\Delta_k}\cdot\left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)\right)+\log_\alpha\left(\frac{2(1+\alpha)}{\Delta_2}\cdot\left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)\right)+KMT^3\delta.$$

As each communication round above needs $2(M-1)$ communications and the notification of arm elimination needs $(K-1)M$ communications in total, the final communication costs are upper bounded by

$$\sum_{k>1}2M\log_\alpha\left(\frac{2(1+\alpha)}{\Delta_k}\cdot\left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)\right)+2M\log_\alpha\left(\frac{2(1+\alpha)}{\Delta_2}\cdot\left(\frac{e^\varepsilon+1}{e^\varepsilon-1}\right)\right)+2KM^2T^3\delta.$$

□

Proof of Lemma 19.

$$\begin{aligned}
&\text{Arm } k \text{ is pulled at time } t\\
\overset{(a)}{\Longrightarrow}\ & k\in C(t) \text{ for agent } \ell \text{ fulfills Lemma 18}\\
\overset{(b)}{\Longrightarrow}\ & \hat{\mu}_k^{(\ell)}(t)+\mathsf{CR}(n_k(\tau_k(t))+M(n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))),\delta)\\
&\qquad\geqslant \hat{\mu}_{k'}^{(\ell)}(t)-\mathsf{CR}(n_{k'}(\tau_{k'}(t))+M(n_{k'}^{(\ell)}(t)-n_{k'}^{(\ell)}(\tau_{k'}(t))),\delta) \text{ for any } k'\in C(t)\\
\overset{(c)}{\Longrightarrow}\ & \hat{\mu}_k^{(\ell)}(t)+2\mathsf{CR}(n_k(\tau_k(t))+M(n_k^{(\ell)}(t)-n_k^{(\ell)}(\tau_k(t))),\delta)\geqslant \hat{\mu}_{k'}^{(\ell)}(t) \text{ for any } k'\in C(t)\\
\overset{(d)}{\Longrightarrow}\ & \hat{\mu}_k^{(\ell)}(t)+2\mathsf{CR}(n_k(t),\delta)\geqslant \hat{\mu}_{k'}^{(\ell)}(t) \text{ for any } k'\in C(t)\\
\Longrightarrow\ & \hat{\mu}_k^{(\ell)}(t)+2\mathsf{CR}(n_k(t),\delta)\geqslant \hat{\mu}_1^{(\ell)}(t)\\
\overset{(e)}{\Longrightarrow}\ & \mu_{k,\varepsilon}+(2+\alpha)\mathsf{CR}(n_k(t),\delta)\geqslant \mu_{1,\varepsilon}-\alpha\mathsf{CR}(n_1(t),\delta)\\
\Longrightarrow\ & 2(1+\alpha)\mathsf{CR}(n_k(t),\delta)\geqslant \mu_{1,\varepsilon}-\mu_{k,\varepsilon}=\Delta_{k,\varepsilon} \qquad (20)\\
\Longrightarrow\ & n_k(t)\leqslant\frac{2(1+\alpha)^2\log\delta^{-1}}{\Delta_{k,\varepsilon}^2}
\end{aligned}$$

where (a) is because the candidate arm sets $C(t)$ are the same for all agents (including arm ℓ), (b) is by the definition of candidate arm set, (c) is because arms in the candidate arm set are evenly explored in a round-robin manner, (d) is from the definition of arm ℓ in (19), and (e) is by applying Lemma 18.

Lastly, since the pulling of arm i in the critical time slot t is not counted, the total pulling times of arm i may be increased by M at most, i.e.,

$$n_k(T)\leqslant\frac{2(1+\alpha)^2\log\delta^{-1}}{\Delta_{k,\varepsilon}^2}+M.$$

□

D.3 Proof of Theorem 16

Proof of Theorem 16. Since the proof of Theorem 16 is quite similar to that in Section B, we only highlight the key steps that differ. Based on Proposition 17, the mean of private observation is $\mu_{k,\varepsilon}=\frac{1}{2}+(\mu_k-\frac{1}{2})\cdot\frac{e^\varepsilon-1}{e^\varepsilon+1}$. Let $0<\lambda_\varepsilon<\min_{k>1}\frac{\mu_{k-1,\varepsilon}-\mu_{k,\varepsilon}}{4}$, the new events are several subsets of

time slots that leader ℓ is active as follows,

$$\begin{aligned}
\mathcal{A} &\coloneqq \{t \in \mathcal{T}^{(\ell)} : I(t) \neq 1\},\\
\mathcal{B} &\coloneqq \{t \in \mathcal{T}^{(\ell)} : |\hat{\mu}_{I(t),\varepsilon}(t) - \mu_{I(t),\varepsilon}| \geqslant \lambda_\varepsilon\},\\
\mathcal{G} &\coloneqq \{t \in \mathcal{T}^{(\ell)} : d_1(t) < \mu_{1,\varepsilon}(t)\},\\
\mathcal{H} &\coloneqq \{t \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{G}) : |\hat{\mu}_{1(t),\varepsilon} - \mu_{1,\varepsilon}| \geqslant \lambda_\varepsilon\}.
\end{aligned}$$

Lemma 20. *$\mathcal{A} \cup \mathcal{B} \subset \mathcal{B} \cup \mathcal{G} \cup \mathcal{H}$ and hence, $\mathbb{E}[|\mathcal{A}|] \leqslant \mathbb{E}[|\mathcal{B}|] + \mathbb{E}[|\mathcal{G}|] + \mathbb{E}[|\mathcal{H}|]$*

Proof of Lemma 20. Let $t \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{G})$. To prove this lemma, one only needs to show $t \in \mathcal{H}$, which can be derived as follows,

$$\hat{\mu}_1(t) \overset{(a)}{\leqslant} \hat{\mu}_{I(t),\varepsilon} \overset{(b)}{\leqslant} \mu_{I(t),\varepsilon} + \lambda_\varepsilon \overset{(c)}{\leqslant} \mu_{1,\varepsilon} - \lambda_\varepsilon,$$

where inequality (a) is due to the definition of $I(t)$, inequality (b) is because $t \notin \mathcal{B}$, and inequality (c) is due to the definition of λ_ε. □

Lemma 21. *$\mathbb{E}[|\mathcal{B}|] + \mathbb{E}[|\mathcal{G}|] + \mathbb{E}[|\mathcal{H}|] \leqslant 2K^2\lambda_\varepsilon^{-2} + 57K$*

Proof of Lemma 21. We respectively upper bound $\mathbb{E}[|\mathcal{B}|]$, $\mathbb{E}[|\mathcal{G}|]$, and $\mathbb{E}[|\mathcal{H}|]$.

Show $\mathbb{E}[|\mathcal{B}|] \leqslant K^2\lambda_\varepsilon^{-2}$. We denote $\mathcal{B}_k \coloneqq \{t \in \mathcal{B} : I(t) = k\}$ for all arm $k \in \mathcal{K}$. Due to the exploration design of Algorithm 2, we have $n_k(t) \geqslant (1/K)\sum_{s=1}^{t} \mathbb{1}\{t \in \mathcal{B}_k\}$. Then applying Lemma 10 with $\mathcal{E} = \mathcal{B}_k$ and $\zeta = 1/K$, we have $\mathbb{E}[|\mathcal{B}_k|] \leqslant K\lambda_\varepsilon^{-2}$. Therefore, with a union bound, we have $\mathbb{E}[|\mathcal{B}|] \leqslant \sum_{k\in\mathcal{K}} \mathbb{E}[|\mathcal{B}_k|] \leqslant K^2\lambda_\varepsilon^{-2}$.

Show $\mathbb{E}[|\mathcal{G}|] \leqslant 57K$. Applying Lemma 11 leads to this upper bound.

Show $\mathbb{E}[|\mathcal{H}|] \leqslant K\lambda_\varepsilon^{-2}$. Notice that $t \in \mathcal{H}$ guarantees that

$$d_1(t) \overset{(a)}{\geqslant} \mu_{1,\varepsilon} \overset{(b)}{\geqslant} \mu_{I(t),\varepsilon} + \lambda_\varepsilon \overset{(c)}{\geqslant} \hat{\mu}_{I(k)(t),\varepsilon},$$

where inequality (a) is because $t \notin \mathcal{G}$, inequality (b) is due to the definition of λ_ε, and inequality (c) is because $t \notin \mathcal{B}$. Since $d_1(t) \geqslant \hat{\mu}_{I(t)}(t)$, the optimal arm 1 is inside the exploration arm set $\mathcal{D}(t)$, and Algorithm 2 thus explore this arm at least once every K rounds, i.e., $n_k(t) \geqslant (1/K)\sum_{s=1}^{t} \mathbb{1}\{t \in \mathcal{H}\}$. Applying Lemma 10 with $\mathcal{E} = \mathcal{H}$ and $\zeta = 1/K$ yields $\mathbb{E}[|\mathcal{H}|] \leqslant K\lambda_\varepsilon^{-2}$. □

Step 2. Bound the regret of leader exploring suboptimal arms. Denote $Q_k \coloneqq \{t \in \mathcal{T}^{(\ell)} \setminus (\mathcal{A} \cup \mathcal{B}) : J(t) = k\}$ for suboptimal arm $k \neq 1$. We show that,

$$\mathbb{E}[|Q_k|] \leqslant \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_{k,\varepsilon} + \lambda_\varepsilon, \mu_{1,\varepsilon} - \lambda_\varepsilon)} + \lambda_\varepsilon^{-2}.$$

Denote $x_k(t) \coloneqq \sum_{s=1}^{t} \mathbb{1}\{t \in Q_k\}$ as the number of times that $t \in Q_k$ happens up to time t. We set $x_0 \coloneqq \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_{k,\varepsilon}+\lambda_\varepsilon, \mu_{1,\varepsilon}-\lambda_\varepsilon)}$ as a threshold.

We then define two subset of Q_k as follows,

$$\begin{aligned}
Q_{k,1} &\coloneqq \{t \in Q_k : |\hat{\mu}_k(t) - \mu_{k,\varepsilon}| \geqslant \lambda_\varepsilon\},\\
Q_{k,2} &\coloneqq \{t \in Q_k : x_k(t) \leqslant x_0\}.
\end{aligned}$$

Next, we show that $Q_k \subseteq Q_{k,1} \cup Q_{k,2}$. Let $t \in Q_k \setminus (Q_{k,1} \cup Q_{k,2})$. For this t, we have

$$d_k(t) \overset{(a)}{\geqslant} \hat{\mu}_{I(t)}(t) \overset{(b)}{=} \hat{\mu}_1(t) \overset{(c)}{\geqslant} \mu_{1,\varepsilon} - \lambda_\varepsilon \overset{(d)}{>} \mu_{k,\varepsilon} + \lambda_\varepsilon \overset{(e)}{>} \hat{\mu}_k(t), \tag{21}$$

where inequality (a) is due to $t \in \mathcal{Q}_k$, inequality (b) is because $t \notin \mathcal{A}$, inequality (c) is due to $t \notin \mathcal{B}$, inequality (d) is due to the definition of λ_ε, and inequality (e) is for $t \in \mathcal{Q}_{k,1}$. Since $t \notin \mathcal{Q}_{k,2}$, we also have

$$n_k(t) \geqslant x_k(t) > x_0. \tag{22}$$

Then, we have

$$x_0 \,\mathrm{kl}(\hat{\mu}_k(t), \mu_{1,\varepsilon} - \lambda_\varepsilon) \overset{(a)}{\leqslant} n_k(t)\,\mathrm{kl}(\hat{\mu}_k(t), \mu_{1,\varepsilon} - \lambda_\varepsilon) \overset{(b)}{\leqslant} n_k(t)\,\mathrm{kl}(\hat{\mu}_k(t), d_k(t)) \overset{(c)}{\leqslant} \log T + 4\log\log T,$$

where inequality (a) is by (22), inequality (b) is by (21) and $kl(x, y)$ increases with respect to y when $x < y$, and inequality (c) is by the definition of KL-UCB index $d_k(t)$.

Substituting $x_0 = \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_{k,\varepsilon} + \lambda_\varepsilon, \mu_{1,\varepsilon} - \lambda_\varepsilon)}$ into the above inequality leads to

$$\mathrm{kl}(\hat{\mu}_k(t), \mu_{1,\varepsilon} - \lambda_\varepsilon) \leqslant \mathrm{kl}(\mu_{k,\varepsilon} + \lambda_\varepsilon, \mu_{1,\varepsilon} - \lambda_\varepsilon).$$

Noticing that $\mathrm{kl}(x, y)$ decreases with respect to x when $x < y$, we have $\hat{\mu}_k(t) > \mu_k + \lambda_\varepsilon$, which contradicts $t \notin \mathcal{Q}_{k,1}$. Therefore, we know $\mathcal{Q}_k \setminus (\mathcal{Q}_{k,1} \cup \mathcal{Q}_{k,2}) = \emptyset$, i.e., $\mathcal{Q}_k \subseteq \mathcal{Q}_{k,1} \cup \mathcal{Q}_{k,2}$.

Next, we upper bound $\mathbb{E}[|\mathcal{Q}_{k,1}|]$ and $\mathbb{E}[|\mathcal{Q}_{k,2}|]$. To bound $\mathbb{E}[|\mathcal{Q}_{k,1}|]$, we apply Lemma 10 with $\mathcal{E} = \mathcal{Q}_{k,1}$ and $\zeta = 1$ (notice that the arm k is played at most once after each $\mathcal{D}(t)$ renewing), then we have $\mathbb{E}[|\mathcal{Q}_{k,1}|] \leqslant \lambda_\varepsilon^{-2}$. For $\mathbb{E}[|\mathcal{Q}_{k,2}|]$, we have

$$\mathbb{E}[|\mathcal{Q}_{k,2}|] \leqslant x_0 = \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_{k,\varepsilon} + \lambda_\varepsilon, \mu_{1,\varepsilon} - \lambda_\varepsilon)}.$$

Combining both upper bound together yields the upper bound for $\mathbb{E}[|\mathcal{Q}_k|]$.

Step 3. Bound the total regret in stochastic case. We note that the regret costs due to Steps 1 and 2 are orthogonal. For Step 1, the total regret cost of the leader ℓ is upper bounded as follows,

$$1 \cdot \mathbb{E}[|\mathcal{A} \cup \mathcal{B}|] \leqslant 2K^2\lambda_\varepsilon^{-2} + 57K.$$

When the leader makes wrong arm recommendations, the rest agents (followers) would also pull suboptimal arms and thus pay regret costs. Since the leader makes wrong recommendation for $|\mathcal{A}|$ active decision rounds, the total arm pulls during these leader active rounds are $\sum_{m\neq\ell}\theta^{(m)}/\theta^{(\ell)}|\mathcal{A}|$. Hence, the total costs due to followers' mistaken pulling are upper bounded as follows,

$$\mathbb{E}\left[\frac{\sum_{m\neq\ell}\theta^{(m)}}{\theta^{(\ell)}}|\mathcal{A}|\right] \leqslant \frac{\sum_{m\neq\ell}\theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda_\varepsilon^{-2} + 57K).$$

Summing the above two terms together yields an upper bound for the regret cost in Step 1 as follows,

$$\frac{\sum_{m\in\mathcal{M}}\theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda_\varepsilon^{-2} + 57K).$$

For Step 2, the regret cost is only from the leaders' exploration, which is upper bounded as follows,

$$\sum_{k\neq 1}\Delta_k \cdot \mathbb{E}[|\mathcal{Q}_k|] \leqslant \sum_{k\neq 1}\Delta_k \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_k + \lambda_\varepsilon, \mu_1 - \lambda_\varepsilon)} + K\lambda_\varepsilon^{-2}.$$

Summing the regret costs from Steps 1 and 2 yields the regret upper bound as follows,

$$R(T) \leqslant \sum_{k\neq 1}\Delta_k \frac{\log T + 4\log\log T}{\mathrm{kl}(\mu_{k,\varepsilon} + \lambda_\varepsilon, \mu_{1,\varepsilon} - \lambda_\varepsilon)} + \frac{\sum_{m\in\mathcal{M}}\theta^{(m)}}{\theta^{(\ell)}}(2K^2\lambda_\varepsilon^{-2} + 57K) + K\lambda_\varepsilon^{-2}.$$

Communication. Communication only happens when the leader updates its arm recommendation to followers. Therefore, the total number times of recommendations is at most $2\mathbb{E}[|\mathcal{A}|]$ times,

and in each communication, the leader sends $M-1$ messages to each followers. Hence, the actually communications are upper bounded by

$$C(T) \leqslant 4K^2M\lambda_{\varepsilon}^{-2} + 114KM.$$

For the proof for adversarial case, it is the same as the proof for Theorem 5, that is, with an additional $64M^3$ communication costs, and we omit it for simplicity. □

E NUMERICAL EXPERIMENTS FOR THE IMPACT OF α PARAMETER

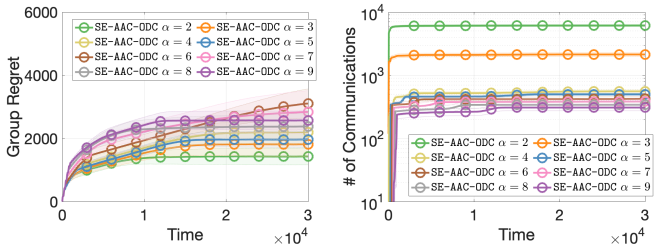
In this section, we conduct additional numerical simulations to investigate the impact of parameter α on our SE-AAC-ODC algorithm.

Please see the figures in our uploaded file. We conduct four sets of experiments evaluating the performance (group regret and total communication costs) of SE-AAC-ODC with $\alpha = 2, 3, 4, 5, 6, 7, 8, 9$. *We found that, in all four sets of experiments, communication costs reduce as α increases; however, group regret has no clear trend of growth as α increases.*

Specifically, in Figure 8, we first consider the setup in the main paper (16 arms with Bernoulli rewards with average rewards uniformly randomly taken from Ad-Click [36], 10 agents with activation frequency of agent m follows a sine function, $\sin(\theta_m + t/30)$, where the phase shifts $\theta_m = m/5$, $m \in \{1, ..., M\}$ differ for different agents) and report the number of communications and regrets after $T = 30\,000$ time slots averaged over 100 independent trials with standard deviation plotted as the shaded area. In Figure 8, we observe that group regret decreases once when α increases from 4 to 5, once when α increases from 6 to 7, and once when α increases from 7 to 8. To verify that this phenomenon is common, we further conduct experiments with the same setting but just fewer agents (5 agents) whose results are reported in Figure 9 and experiments with synchronous agents (all agents are active in all time slots) whose results are reported in Figures 10 and 11.

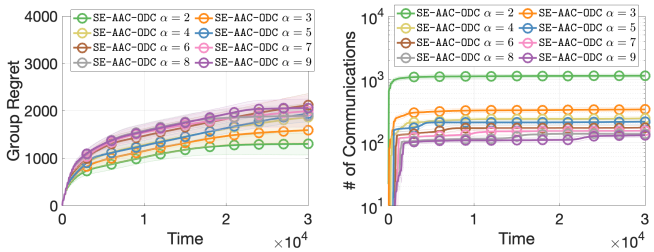
Besides the intuition that reducing communication costs should increase regret, there is another mechanism in our fully distributed algorithm that opposes the increase in regret. This effect is attributed to our successive elimination policy. When an agent eliminates an arm, it broadcasts this decision to all other agents, prompting them to also remove the arm from their candidate set. As α increases, communication frequency decreases, leading to a lack of synchronization in local observations among agents. Consequently, agents may eliminate different suboptimal arms based on their varying stochastic observations, which speeds up the overall arm elimination process and reduces regret. From this perspective, reducing communication may also reduce this regret, contradicting the intuition that reducing communication costs should increase regret. Thus, the interplay between these two factors leads to an unclear trend in group regret.

Received October 2024; revised January 2025; accepted January 2025



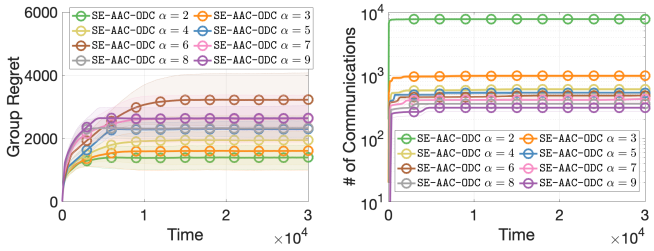
(a) Group Regret (b) Communication

Fig. 8. Asynchronous, 10 agents: 16 arms with Bernoulli rewards with average rewards uniformly randomly taken from Ad-Click [36], 10 agents with activation frequency of agent m follows a sine function, $\sin(\theta_m + t/30)$, where the phase shifts $\theta_m = m/5$, $m \in \{1, ..., M\}$ differ for different agents. We report the number of communications and regrets after $T = 30\,000$ time slots averaged over 100 independent trials, and we plot the standard deviation as the shaded area.



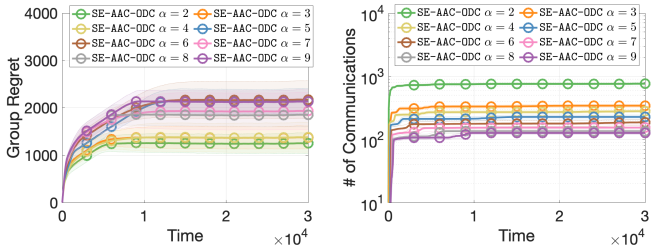
(a) Group Regret (b) Communication

Fig. 9. Asynchronous, 5 agents: 16 arms with Bernoulli rewards with average rewards uniformly randomly taken from Ad-Click [36], 5 agents with activation frequency of agent m follows a sine function, $\sin(\theta_m + t/30)$, where the phase shifts $\theta_m = m/5$, $m \in \{1, ..., M\}$ differ for different agents. We report the number of communications and regrets after $T = 30\,000$ time slots averaged over 100 independent trials, and we plot the standard deviation as the shaded area.



(a) Group Regret (b) Communication

Fig. 10. Synchronous, 10 agents: 16 arms with Bernoulli rewards with average rewards uniformly randomly taken from Ad-Click [36], 10 agents all active in every time slots in time horizon $T = 30\,000$. We report the number of communications and regrets averaged over 100 independent trials, and we plot the standard deviation as the shaded area.



(a) Group Regret (b) Communication

Fig. 11. Synchronous, 5 agents: 16 arms with Bernoulli rewards with average rewards uniformly randomly taken from Ad-Click [36], 5 agents all active in every time slots in time horizon $T = 30\,000$. We report the number of communications and regrets averaged over 100 independent trials, and we plot the standard deviation as the shaded area.