QuACK: A Multipurpose Queuing Algorithm for Cooperative k-Armed Bandits

Benjamin Howson
Imperial College London

Sarah Filippi
Imperial College London

Ciara Pike-Burke
Imperial College London

Abstract

This paper studies the cooperative stochastic k-armed bandit problem, where m agents collaborate to identify the optimal action. Rather than adapting a specific single-agent algorithm, we propose a general-purpose black-box reduction that extends any single-agent algorithm to the multi-agent setting. Under mild assumptions, we prove that our black-box approach preserves the regret guarantees of the chosen algorithm, and is capable of achieving minimax-optimality up to an additive graph-dependent term. Our method applies to various bandit settings, including heavy-tailed and duelling bandits, and those with local differential privacy. Empirically, it is competitive with or outperforms specialized multi-agent algorithms.

1 INTRODUCTION

The stochastic multi-armed bandit problem is a fundamental model for sequential decision-making. Here, a single agent sequentially interacts with the environment over a series of rounds. In each round, the agent selects an action and receives a reward drawn from an unknown distribution associated with that action. (Lattimore and Szepesvári, 2020).

There are numerous provably efficient algorithms for this setting, and there are various mechanisms for balancing the exploration-exploitation trade-off. Examples include optimism, posterior sampling, bootstrapping, and softmax exploration (Auer et al., 2002a; Thompson, 1933; Kveton et al., 2019; Bian and Jun, 2022).

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

However, decision-making tasks naturally arise in distributed settings, such as recommender systems and sensor networks (Tekin et al., 2014; Tran-Thanh et al., 2011). Here, there exist multiple decision-makers interacting with the environment. For example, large-scale recommender systems are often a distributed system of servers, where each server hosts a decision-maker. Every time a user arrives to one of the servers, the corresponding decision-maker chooses an item to recommend, and the user will provide a reward signal in response. The decision-makers can update their own knowledge on the quality of the item with this feedback. They can also share this information with neighbouring servers to improve future decision-making at all servers.

Motivated by distributed applications, we study an extension of the traditional multi-armed bandit model where there is a network of m agents who each interact with the same k-armed bandit environment. This extension allows each agent to communicate over the network and possibly collaborate with other agents to speed up the learning process.

1.1 Related Work

Designing provably efficient algorithms for the multi-agent setting is more challenging than the single-agent setting. Previous work has extended *specific* single-agent algorithms to the multi-agent setting using either a gossip-based or a leader-based approach.

Gossip-Based. The gossiping protocol is a popular technique from distributed computing. The main idea is to use an iterative averaging procedure to aggregate information from neighbouring agents to approximate the full network information (Xiao and Boyd, 2004; Duchi et al., 2012). This technique has been combined with two well-known single-agent bandit algorithms. Landgren et al. (2016) and Martínez-Rubio et al. (2019) focus on the upper confidence bound algorithm (Auer et al., 2002a). They each use variants of the gossiping protocol to approximate the number of plays and the sum of rewards across the entire network,

which the agents use to compute their upper confidence bounds. Lalitha and Goldsmith (2021) combine the gossiping protocol, used to approximate the full network posterior, with Thompson Sampling in bandit problems with Bernoulli rewards (Thompson, 1933). These algorithms are all asymptotically optimal. However, the analyses of these algorithms are complex, and specific to the algorithm and variant of the gossiping protocol considered. Furthermore, their performance depends on the choice of communication matrix that the gossip procedure uses for iterative averaging. Even when the network is known, choosing this matrix is a non-trivial task, and it is common to use heuristics (Xiao and Boyd, 2004).

Leader-Based. The leader-based approaches nominate one or more leading agents. These agents direct the exploration of their followers. Bar-On and Mansour (2019) consider the non-stochastic setting and analyse the exponential weights algorithm with numerous leaders, who each send their action distributions to the non-leading agents in their neighbourhood. Wang et al. (2020) consider the stochastic setting with limited communication. Their algorithm elects a single leading agent who performs all the exploration, using an upper confidence bound algorithm, and all the communication. The remaining agents act greedily with respect to the empirical means of the leader. Again, these papers analyse extensions of specific single-agent bandit algorithms. Our approach also fits into the category of leader-based approaches. However, we develop a leader-based black-box reduction where the leading agent can use any single-agent bandit algorithm for decision-making. Specifically, we build upon queuing methods commonly used to handle delayed feedback (Joulani et al., 2013; Mandel et al., 2015). However, extending these queuing methods to the multi-agents setting introduces unique theoretical challenges that do not arise in single-agent scenarios.

Additional Related Work. There exist many variations of the multi-agent bandit problem in the literature. Szorenyi et al. (2013) study the multi-armed bandit problem in peer-to-peer networks where each agent communicates with two randomly chosen agents in every round. Yang et al. (2021) and Chen et al. (2023) consider the setting where each agent on the network plays at different and possibly unknown times. Madhushani et al. (2021) study the setting where there is imperfect communication between agents. Shahrampour et al. (2017); Hossain et al. (2021) study the case where every agent has their own distribution over the rewards for each action. Perhaps the best-studied multi-agent problem is where agents receive zero or degraded rewards if they play the same action at the same time. See (Boursier and Perchet, 2024) for an extensive and recent survey on this topic.

1.2 Contributions

This paper proposes and analyses a general purpose algorithm for cooperative stochastic k-armed bandit problems. Our main contributions are as follows:

- We propose a black-box reduction, QuACK, that accepts any single-agent bandit algorithm as input and immediately extends it to the multi-agent setting.
- Theorem 1 shows that we can upper bound the performance over the entire network in terms of the guarantees of the chosen single-agent bandit algorithm. Pairing our reduction with a near optimal algorithm yields a near optimal algorithm for the multi-agent version of the standard k-armed bandit problem, up to an additive graph-dependent quantity. These results are competitive with or better than the case-by-case analyses of previous works.
- Our theoretical guarantees hold under mild assumptions on the bandit environment. We require that the distribution of the reward depends only on the chosen action, and each distribution has a finite first moment. Hence, if there exists a provably efficient algorithm for a single-agent bandit problem, and this bandit problem satisfies the assumptions, our reduction guarantees that it will be provably efficient in the multi-agent setting. This makes developing provably efficient multi-agent algorithms for various bandit problems simple. In particular, in Section 4:
  - We demonstrate the simplicity by using our reduction to design multi-agent algorithms for heavy-tailed bandits and duelling bandits, which have all been considered separately in the literature (Landgren et al., 2016; Dubey and Pentland, 2020; Raveh et al., 2024). The resulting algorithms have comparable guarantees to those developed specifically for each setting.
  - We demonstrate how to use our theoretical findings to develop provably efficient algorithms for new multi-agent settings, such as local differential privacy, by using an appropriate single-player algorithm (Ren et al., 2020).
- Finally, we perform an experimental comparison to existing works which shows that our reduction is competitive or outperforms existing methods

when paired with a comparable single-agent bandit algorithm.

2 PROBLEM SETTING

This paper considers cooperative multi-agent variants of stochastic multi-armed bandit problems where we have a finite set of actions: $\mathcal{A}$ such that $|\mathcal{A}| = k$. Formally, we will represent the network of m agents and their connections through an undirected graph G with:

$$\begin{aligned} &G = (V, E) \\ &V = \{1, 2, \cdots, m\} \\ &E \subseteq \{(v, w) \in V \times V : v \neq w\} \end{aligned}$$

where $v \in V$ and $e \in E$ represent an agent and a communication channel between a pair of agents, respectively. Throughout, we consider the setting where the agents can communicate with any other agent in their *neighbourhood*, which we define for all $v \in V$ as follows:

$$N_v = \{w \in V : (v, w) \in E\}.$$

Each round consists of every agent simultaneously playing their own action, receiving their own reward and communicating with their neighbours. Formally, for each $t \in \{1, 2, \cdots, n\}$:

- Each agent $v \in V$ plays action $A_t^v \in \mathcal{A}$.
- Each agent $v \in V$ receives reward $X_t^v \in \mathbb{R}$.
- Each agent $v \in V$ communicates with all $w \in N_v$.

Throughout, we will make use of a standard and mild assumption on the bandit environment (Lattimore and Szepesvári, 2020).

Assumption 1. *The bandit environment generates feedback that depends only on the chosen action. Letting P_a denote the reward distribution for action a, we assume that:*

$$X_t^v \,|\, A_t^v = a \overset{\text{iid}}{\sim} P_a \;\; \textit{and} \;\; \mu_a = \mathbb{E}_{X \sim P_a}[X] < \infty$$

for all $t \in \mathbb{N}$, $a \in \mathcal{A}$ and $v \in V$.

Additionally, we make a mild assumption on the graph and the communication protocol.

Assumption 2. *Let d_{vw} denote the length of the shortest path between agents $(v, w) \in V \times V$ with $d_{vv} = 0$. Messages can be passed from v to w in d_{vw} rounds and the graph diameter is finite:*

$$d = \max_{(v,w) \in V \times V} d_{vw} < \infty \,.$$

2.1 Measuring Performance

Our focus is on the cooperative stochastic multi-armed bandit problems, where the network of agents collaborate to minimise the regret of the entire network, also known as the *group regret*:

$$R_G(n) = \sum_{a \in \mathcal{A}} \Delta_a \, \mathbb{E}\left[\sum_{v=1}^{m} T_{av}(n)\right]$$

where the expectation is over the network of agents interacting with the bandit environment. Here, $\Delta_a = \mu_\star - \mu_a$ is the sub-optimality gap of action a where $\mu_\star = \max \mu_a$ denotes the expected reward of the optimal action, and

$$T_{av}(n) = \sum_{t=1}^{n} 1\{A_t^v = a\} \tag{1}$$

is the number of times agent $v \in V$ chooses action $a \in \mathcal{A}$ over the course of n rounds. Applying standard arguments gives a minimax lower bound on the group regret when all reward distributions have bounded support (Auer et al., 2002b):

$$R_G(n) \geq \frac{1}{20}\sqrt{mn(k-1)}. \tag{2}$$

This lower bound holds for cases where each agent can communicate immediately with any other agent on the graph. Hence, it may not be tight for graphs with certain structures. However, it does demonstrate that information sharing can be used to improve the group performance in the worst-case.

3 BLACK-BOX REDUCTION

We present QuACK, *a queuing algorithm for cooperative k-armed bandits*. QuACK is a multipurpose black-box reduction that can be paired with any single-agent bandit algorithm to extend it to the multi-agent case.

QuACK is a leader-based algorithm that builds on the queuing approach that was developed for the setting of delayed feedback (Joulani et al., 2013; Mandel et al., 2015). Essentially, the idea is to select a *leader* in the network and provide them with an arbitrary single-agent bandit algorithm to select actions for the whole network. The leader begins by initialising an empty queue for each action. These queues will store the rewards that other agents observe from the environment and have passed to the leader. Intuitively, the leader can use the reward samples in these queues instead of playing actions in the real environment.

Specifically, in each round, the leader begins by asking the bandit algorithm for an action. Whenever there

Algorithm 1 QuACK

Input: Leader $v \in V$
Input for Leader: Bandit Algorithm π
Initialisation for Leader: $Q_a = \emptyset$ for all $a \in \mathcal{A}$
for $t = 1, 2, \cdots, n$ **do**
  Leader (v)
    Receive messages via shortest path:
      $\mathcal{M}_t = \{(A^w_{t-d_{vw}}, X^w_{t-d_{vw}})\}_{w \neq v}$
    Append messages to the queues:
      $Q_a = Q_a \cup \{x : (a', x) \in \mathcal{M}_t \text{ and } a' = a\}$
    Select a with π
    while $Q_a \neq \emptyset$ **do**
      Remove x from Q_a
      Update π with (a, x)
      Select a with π
    end while
    Play $A^v_t = a$
    Observe $X^v_t \sim P_{A^v_t}$
    Send A^v_t to all $w \neq v$ via shortest path
  Follower $(w \neq v)$
    if $t > d_{vw}$ **then**
      Receive $A^v_{t-d_{vw}}$ from v via shortest path
      Play $A^w_t = A^v_{t-d_{vw}}$
    else
      Play A^w_t uniformly at random
    end if
    Observe $X^w_t \sim P_{A^w_t}$
    Send (A^w_t, X^w_t) to v via shortest path.
end for

is a reward in the queue for that action, the leader will take the reward from the queue and use this to update the bandit algorithm. Importantly, the leader does not play this action in the environment. This continues until the bandit algorithm suggests playing an action whose queue is empty. Once this occurs, the leader will play the action in the environment and will use the observed reward to update the bandit algorithm. The leader then communicates this decision to the other, *follower*, agents. Once a follower receives the decision from the leader, they will proceed to play the instructed action and communicate the reward back to the leader. Once the leader receives rewards from their followers, they will place the rewards in the corresponding queues, and begin the process again. Notably, the process of sending decisions and rewards can take several rounds because the message needs to travel across the network.

Algorithm 1 presents the pseudo-code for QuACK. For clarity, we have made two simplifications in the presentation of the algorithm. Firstly, the index of the leading agent is given as input. Secondly, the shortest path between each follower and the chosen leader is known. However, these quantities need not be known in practice. Appendix A explains how to nominate the leader and compute the required shortest paths in a distributed manner, e.g. each agent only knows the structure of the graph locally.

3.1 Message-Passing Protocol

The message a follower $w \neq v$ sends to the leader v in round t contains only the action and the reward:

$$(A^w_t, X^w_t).$$

Sending such simple messages is possible since there is a known fixed route between the leader and any follower. Although multiple routes with the minimum path length might exist, the algorithm determines exactly one and adheres to it throughout. Computing these routes can be done in a distributed manner before the start of the interaction. Specifically, this amounts to constructing the shortest path tree of the graph, which can be done by solving the single-source shortest-path problem before interacting with the bandit environment (Ahuja et al., 1993). Figure 1 illustrates the shortest path tree for the grid graph, where the black vertex represents the leader.

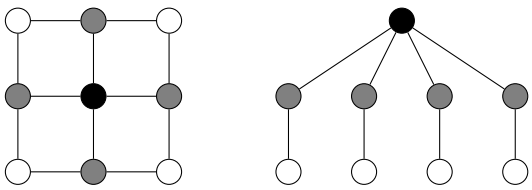

Figure 1: Grid Graph and its Shortest Path Tree.

The Distributed Bellman-Ford algorithm can solve the single-source shortest-path problem exactly and it does so using approximately m iterations, whilst only requiring that each agent knows their neighbours and has a unique identifier (Ford, 1956; Bellman, 1958; Moore, 1959; Elkin, 2020). Appendix A presents full details of this algorithm for completeness.

Sending instructions from the leader to the followers using the shortest path tree is straightforward. The leader communicates the action they played to their children, and the children send this action to their children at the end of the next round, and so forth. Using the shortest path tree also makes passing messages to the leader straightforward. Each follower will forward messages from their children to their parent, and send their own message to their parents. Let, $\mathcal{C}_w$ and $\mathcal{P}_w$ denote the children and the parent of agent w on the shortest path tree, respectively. Then, in the

t-th round, each follower w will perform the following communication:

- Forward $A_t^w = A_{t-d_{vw}}^v$ to each $z \in \mathcal{C}_w$.
- Send $M_t^v = (A_t^w, X_t^w) \cup \{M_{t-1}^z\}_{z \in \mathcal{C}_w}$ to $\mathcal{P}_z$.

The leader v only needs to send the action they choose in each round to their children, which corresponds to just the first item in the above list. Thus, the communication complexity, in terms of the number of messages sent in each round, for each agent w is upper bounded by the diameter of the graph.

Note that we could use any other message-passing protocol that satisfies Assumption 2. For example, each agent could forward everything they have been sent in the past m rounds. However, in this case, all messages would have to contain additional information about who originally sent the message and the time the message was sent, e.g. a unique agent identifier and the round. This extra information is required so that the followers can identify the most recent action sent to them by the leader. Furthermore, this additional information can be used by the leader to ensure that each message is added to the queue only once.

3.2 Theoretical Analysis

This subsection provides an analysis of QuACK, as presented in Algorithm 1. Throughout, we will prove guarantees for an arbitrarily chosen leader and bandit algorithm. This will allow us to show that the leading order term in the group regret depends only on the quality of the chosen algorithm, with the graph-dependence being relegated to a lower order term.

We will now show that, by maintaining the queues, the leader creates a simulated single-player version of the environment for the bandit algorithm. For this, we need to define the number of times the single-agent bandit algorithm chooses each action up to and including their τ-th decision. Letting $\tilde{A}_s$ denote the s-th action chosen by the bandit algorithm and the counter as follows:

$$T'_a(\tau) = \sum_{s=1}^{\tau} 1\{\tilde{A}_s = a\} \tag{3}$$

for all $(a, \tau) \in \mathcal{A} \times \mathbb{N}$.

Define $P = \{P_1, \cdots, P_k\}$ to be the set of reward distributions that characterise the multi-armed bandit environment faced by the network of agents, as in Assumption 1. Then, the cumulative regret of the bandit algorithm after directly interacting with the bandit environment for τ rounds can be defined in the usual manner:

$$S_\pi(\tau) = \sum_{a \neq \star} \Delta_a \, \mathbb{E}_P \left[T'_a(\tau) \right] ,$$

where $\mathbb{E}_P$ is used to make it explicit that the bandit algorithm is interacting directly with the environment. Lemma 1 shows that the cumulative regret of the bandit algorithm interacting directly with the bandit environment for τ rounds is equivalent to its cumulative regret over τ rounds in the queued version of the environment.

Lemma 1. *Under Assumption 1, QuACK guarantees, for all $\tau \in \mathbb{N}$, that:*

$$S_\pi(\tau) = \sum_{a \neq \star} \Delta_a \, \mathbb{E} \left[T'_a(\tau) \right]$$

where $\mathbb{E}$ denotes expectation with respect to the interaction between the network of agents and the multi-armed bandit environment.

Proof. Under Assumption 1, the rewards for each action-agent pair are independent and identically distributed random variables. Properties of exchangeable random variables guarantee that the sequence of rewards the leading agent feeds to the bandit algorithm has the same distribution as in the single-player environment. Therefore, the expectation in the definition of the single-player regret and in the multi-agent setting are the same. See Appendix B.1 for a formal proof. □

Recall that the group regret is defined in terms of the number of times the network of agents plays each action. Thus, Lemma 1 is only going to be useful if we can relate the number of times the network of agents plays each action to the number of times the bandit algorithm plays each action in the queues. This relationship is what we now establish.

In QuACK, the bandit algorithm is primarily updated using rewards from the queues. Occasionally, the leader must play an action in the bandit environment if the corresponding queue is empty. Therefore, Equation (3) accounts for both the number of times the leader performs this action in the bandit environment and the number of samples taken from the queue.

Let s_t denote the number of simulated rounds the bandit algorithm has completed in the queued version of the environment by the end of the t-th round. Algorithm 1 tells us that the action played by the leader in the t-th round of its interaction with the bandit environment will be the action suggested by the bandit algorithm in the corresponding simulated round:

$$\tilde{A}_{s_t} = A_t^v \, .$$

Thus, by the end of the final round, we can write down the number of times the single-agent bandit algorithm chooses each action as follows:

$$T'_a(s_n) = \sum_{t=1}^{n} \sum_{s=s_{t-1}+1}^{s_t} 1\{\tilde{A}_s = a\} . \tag{4}$$

Lemma 2 uses Equation (4) to relate the number of times the network of agents plays each action to the number of times the bandit algorithm chooses each action.

Lemma 2. *Running QuACK with an arbitrary $v \in V$ as the leader guarantees that:*

$$\sum_{w=1}^{m} T_{aw}(t - d_{vw}) \leq T'_a(s_t) + 2\sum_{w=1}^{m} d_{vw}$$

for all $t \in \mathbb{N}$.

Proof. See Appendix B.2 □

Lemmas 1 and 2 provide crucial insights into the properties of Algorithm 1. Namely, that the queuing mechanism simulates the bandit environment and that the decision-making of the bandit algorithm in the queues is directly related to the network of agents in the environment. Given these, the following theorem upper bounds the group regret by the regret of the bandit algorithm in the queues, plus an additive graph-dependent quantity that measures the speed of information flow to and from the leader.

Theorem 1. *Under Assumptions 1 and 2, the group regret of QuACK run with single-player bandit algorithm π and leading agent $v \in V$ is bounded by:*

$$R_G(n) \leq S_\pi(mn) + \left(3\sum_{w=1}^{m} d_{vw}\right)\sum_{a=1}^{k} \Delta_a.$$

Proof. The proof begins by upper bounding the group plays for action $a \in \mathcal{A}$ at the end of the final round as:

$$\sum_{w=1}^{m} T_{aw}(n) \leq \sum_{w=1}^{m} T_{aw}(n - d_{vw}) + \sum_{w \neq v} d_{vw}. \tag{5}$$

From Section 3.1, we know that the leader appends each message to the queue exactly once, which implies that $s_n < mn$ almost surely. Combining Equation (5) with Lemma 2 and $s_n < mn$ allows us to upper bound the number of times the entire network plays action a by the number of times π has played this action:

$$\begin{aligned}\sum_{w=1}^{m} T_{aw}(n) &\leq \sum_{w=1}^{m} T_{aw}(n - d_{vw}) + \sum_{w \neq v} d_{vw} \\ &\leq T'_a(mn) + 3\sum_{w \neq v} d_{vw}. \end{aligned} \tag{6}$$

Plugging Equation (6) into the definition of the group regret gives us:

$$\begin{aligned} R_G(n) &= \sum_{a \neq \star} \Delta_a \, \mathbb{E}\left[\sum_{w=1}^{m} T_{aw}(n)\right] \\ &\leq \sum_{a \neq \star} \Delta_a \, \mathbb{E}\left[T'_a(mn)\right] + \left(3\sum_{w \neq v} d_{vw}\right)\sum_{a \neq \star} \Delta_a \\ &= S_\pi(mn) + \left(3\sum_{w \neq v} d_{vw}\right)\sum_{a \neq \star} \Delta_a \end{aligned}$$

where the final line follows from Lemma 1. Appendix B.3 presents further details. □

Theorem 1 suggests that we should pick the leading agent as follows:

$$v_\star = \arg\min_{v \in V} \sum_{w=1}^{m} d_{vw} .$$

This intuitively makes sense. Indeed, minimising the sum of the shortest paths will minimise the total delay in sending and receiving instructions and feedback from the followers. Appendix A presents an approach for finding this agent in a distributed manner. Nevertheless, Assumption 2 guarantees that the worst-case graph-dependence is given by:

$$\sum_{w=1}^{m} d_{vw} \leq d(m-1),$$

which holds for any choice of leading agent.

Theorem 1 holds for any choice of bandit algorithm and makes weak assumptions on the rewards. This is in contrast to existing works, where it is common to analyse a specific bandit algorithm and make stronger assumptions on the reward distributions, such as having a particular parametric form or subgaussian tails. Section 4 shows that the guarantees presented in this subsection are competitive with the case-specific analyses that exist in the literature when providing QuACK with the corresponding single-agent bandit algorithm.

4 INSTANCES OF QuACK

Theorem 1 holds under the assumption that the rewards are conditionally independent given the actions and the expected reward for each of the actions is finite. Therefore, we can apply our reduction in numerous bandit environments by choosing an appropriate single-agent algorithm. Here, we present a non-exhaustive selection of bandit problems that satisfy Assumption 1.

In each bandit problem, QuACK will be provided with a specific single-agent algorithm. The specific choices are made so that we can draw comparisons with the existing literature wherever possible. However, any other provably efficient single-player algorithm could be combined with our reduction to get similar results.

4.1 Standard Bandits

The standard multi-armed bandit problem consists of playing an action and receiving a reward drawn independently from a distribution that depends only on the chosen action (Lattimore and Szepesvári, 2020). Typically, it is assumed that each distribution has support or has subgaussian tails.

Thus, Assumption 1 holds and we turn our attention to the bandit environments containing distributions with support bounded in $[0, 1]$. Corollaries 1 and 2 presents the group regret bound for our algorithm when the leader uses UCB1 (Auer et al., 2002a) and MOSS (Audibert and Bubeck, 2009), respectively.

Corollary 1. *Running QuACK with UCB1 of Auer et al. (2002a) and an arbitrary leader guarantees that:*

$$R_G(n) \leq \sum_{a \neq \star} \frac{8 \ln(mn)}{\Delta_a} + \left(3 + 3 \sum_{w=1}^{m} d_{vw}\right) \sum_{a \neq \star} \Delta_a.$$

Proof. Theorem 1 of Auer et al. (2002a) provides an upper bound on the regret of the upper confidence bound algorithm over a τ-round interaction with the bandit environment:

$$S_\pi(\tau) \leq \sum_{a \neq \star} \frac{8 \ln(\tau)}{\Delta_a} + 3 \sum_{a \neq \star} \Delta_a. \tag{7}$$

Combining this upper bound with Theorem 1 and taking $\tau = mn$ gives the result. □

Corollary 2. *Running QuACK with MOSS of Audibert and Bubeck (2009) and an arbitrary leader guarantees that:*

$$R_G(n) \leq 49\sqrt{mnk} + \left(3 \sum_{w=1}^{m} d_{vw}\right) \sum_{a \neq \star} \Delta_a.$$

Equation (2) presents the minimax lower bound for the multi-agent version of the standard multi-armed bandit problem. Corollary 2 shows that QuACK is capable of attaining this minimax rate up to an additive term that is independent of the horizon. Conversely, prior works miss the minimax lower bound by at least a poly-logarithmic factor involving the horizon and the number of agents (Landgren et al., 2016; Martínez-Rubio et al., 2019; Lalitha and Goldsmith, 2021).

Martínez-Rubio et al. (2019) analyse a multi-agent extension of UCB1 (Auer et al., 2002a). This is not the only extension of this well-known algorithm to the multi-agent setting (Landgren et al., 2016). However, Martínez-Rubio et al. (2019) offer the best theoretical guarantees, and prove the upper bound on the group regret:

$$R_G(n) \leq \sum_{a \neq \star} \frac{8\eta\left(1 + \frac{\epsilon}{2}\right) \ln(mn)}{\Delta_a} + C_G \sum_{a \neq \star} \Delta_a$$

where $\eta > 1$ and $\epsilon \in (0, 1)$ are tunable hyperparameters of their algorithm. Further, C_G is a graph-dependent quantity with the following definition:

$$C_G = \left\lceil \frac{6m \ln\left(\frac{2m}{\epsilon}\right)}{\sqrt{2 \ln |\lambda|^{-1}}} \right\rceil + m + 4$$

where λ is the second largest singular value of the communication matrix they use for the gossiping procedure. Corollary 1 shows a strict improvement in the leading-order term. Comparing the graph-dependence is a little more challenging and requires specifying the communication matrix. Following their recommendations, we can show that for regular graphs, our graph-dependence is smaller than a function of theirs:

$$3 \sum_{w=1}^{m} d_{vw} < \frac{C_G}{\sqrt{2 \ln |\lambda|^{-1}}}$$

which suggests the dependence in our bounds is better in regular graphs with a small spectral gap. Additionally, we can prove a strict improvement on the graph-dependence for networks with a specific structure, such as the star graph. Generally, the communication complexity of their algorithm and ours comparable but not equivalent. See Appendix C.1 for further details.

4.2 Heavy-Tailed Bandits

Bubeck et al. (2013) introduced the single-agent heavy tailed bandit problem. The decision-making process is exactly the same as the standard bandit setting. However, the reward distributions only have finite moments of order $1+\epsilon$ where $\epsilon > 0$ controls the heaviness of the tails. Consequently, Assumption 1 still holds because the expected values of each distribution are still finite in this setting. Thus, QuACK can be used for the multi-agent heavy-tailed bandit problem

Bubeck et al. (2013) propose a generic solution for the single-agent setting that they call the *robust upper confidence bound algorithm*. This algorithm replaces the empirical mean with a robust estimator of the expected value, such as the truncated-mean, median-of-means, or Catoni's M (Bickel, 1965; Alon et al., 1999; Catoni, 2012).

Indeed, QuACK can be used with any of the robust upper confidence bound strategies, as defined by these robust estimators of the expected rewards. Corollary 3 presents a guarantee on the group regret when the leading agent uses the truncated-mean estimator.

Corollary 3. *Running QuACK with Robust UCB of Bubeck et al. (2013) using the truncated mean and an arbitrary leader guarantees that:*

$$R_G(n) \leq \mathcal{O}\left(\sum_{a \neq \star} \frac{\sigma^{1/\epsilon} \ln(mn)}{\Delta_a^{1/\epsilon}} + \left(\sum_{w=1}^{m} d_{vw}\right) \sum_{a \neq \star} \Delta_a\right)$$

where $\mathbb{E}_{X \sim P_a}[X^{1+\epsilon}] \leq \sigma$ *for all* $a \in \mathcal{A}$.

Dubey and Pentland (2020) study the multi-agent heavy-tailed bandit problem and devise two algorithms based on the truncated-mean estimator. Their best algorithm achieves a similar guarantee. However, their leading order term is scaled by the independence number of the γ-th power of the graph, where γ is a hyperparameter of their algorithm. Our bound is strictly better when this quantity is greater than one, which occurs when $\gamma < d$. Since they also employ a leader-based algorithm that uses a message passing procedure, the additive terms in their bound and their per-round communication are comparable. See Appendix C.2 for further details.

4.3 Duelling Bandits

Yue et al. (2009) introduced the single-agent duelling bandit problem. Here, feedback from the environment is the outcome of a duel between two actions. Modifying Algorithm 1 for the multi-agent duelling bandit problem amounts to maintaining a queue for each pair of actions: $\mathcal{A}_+ = \mathcal{A} \times \mathcal{A}$. Now, each agent will communicate the pair of actions and the outcome of the duel to their neighbours.

Assumption 1 holds since the outcomes of the duels are assumed to depend only on the pair of actions chosen in each round. Recall that we define the group regret relative to a fixed optimal action. Therefore, Theorem 1 applies to solution concepts, such as the Condorcet, Copeland, and Borda winners (Zoghi et al., 2014, 2015; Jamieson et al., 2015). Corollary 4 presents an upper bound on the group regret for an algorithm designed under the assumption that the Condorcet winner exists (Zoghi et al., 2014).

Corollary 4. *Running QuACK with an arbitrary leader following the relative upper confidence bound algorithm of Zoghi et al. (2014) guarantees:*

$$R_G(n) \leq \mathcal{O}\left(\sum_{a \neq \star} \frac{\ln(mn)}{\tilde{\Delta}_a} + \left(\sum_{w=1}^{m} d_{vw}\right) \sum_{a \neq \star} \tilde{\Delta}_a\right)$$

where $\tilde{\Delta}_a = P(a_\star \succ a) - 1/2$ *denotes the sub-optimality gap of action* a *in the duelling bandit setting.*

Raveh et al. (2024) develop several algorithms for the multi-agent duelling bandit problem. Their best algorithm builds on the relative upper confidence bound algorithm and achieves a similar guarantee. However, their leading order term is scaled by the clique covering number of the γ-th power of the graph, where γ is a hyperparameter of their algorithm. Thus, we obtain a strict improvement whenever this quantity is greater than one, which occurs when $\gamma < d$. Since their algorithm employs a message passing procedure, the additive terms in their bound and the per-round communication are comparable to ours. See Appendix C.3 for further details.

4.4 Local Differential Privacy

Privacy is a topic of growing interest in the machine learning community and seeks to protect private information within datasets. Ren et al. (2020) study *local differential privacy* (LDP) in the bandit setting, where the environment uses a mechanism to make the rewards private before sending them to the agent. Formally, in the multi-agent setting, the only difference is that the rewards each agent receives are privatised:

$$X_t^v = f_\epsilon(X) \quad \text{for } X \overset{\text{iid}}{\sim} P_{A_t^v}$$

where f_ϵ denotes the privatising mechanism that ensures ϵ-LDP of the rewards before handing them to the agent. Procedures for privatising rewards include adding Bernoulli or Laplace noise to the rewards, and proving the mechanism provides the specified level of privacy are independent of the bandit algorithm, e.g. Lemmas 2 and 5 of Ren et al. (2020).

Therefore, we can directly use these mechanisms in the multi-agent setting and guarantee that the rewards are ϵ-LDP. Furthermore, Assumption 1 is satisfied in this setting because the curator adds independent and identically distributed noise to the rewards.

Corollary 5. *Running QuACK with LDP-UCB-L of Ren et al. (2020) and an arbitrary leader is* ϵ*-LDP and guarantees:*

$$R_G(n) \leq \mathcal{O}\left(\sum_{a \neq \star} \frac{1}{\epsilon^2} \frac{\ln(mn)}{\Delta_a} + \left(\sum_{w=1}^{m} d_{vw}\right) \sum_{a \neq \star} \Delta_a\right).$$

Corollary 5 uses QuACK to get the first provably efficient algorithm for multi-agent bandits with local differential privacy.

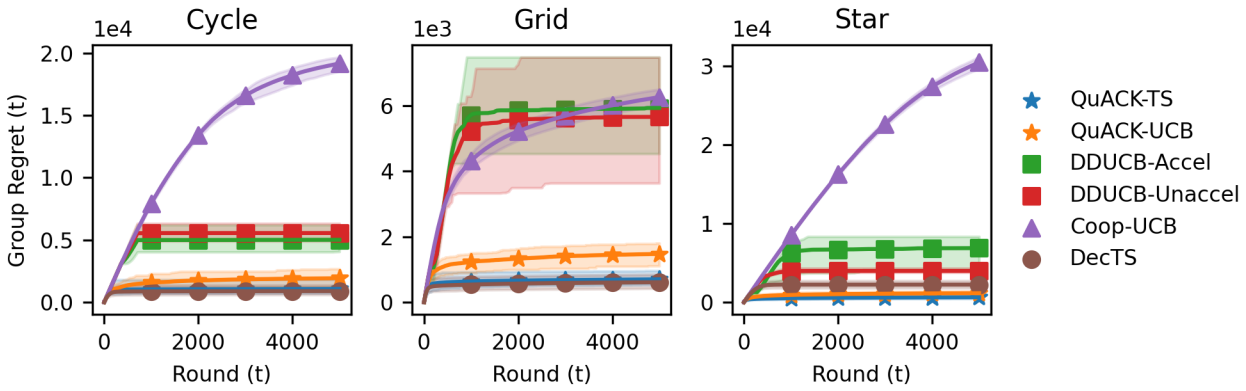

Figure 2: Group Regret for a Network of 196 Agents.

5 EXPERIMENTAL RESULTS

Theorem 1 and its corollaries suggest that our algorithm will be competitive with existing multi-agent approaches when paired with a comparable single-agent bandit algorithm. Naturally, we seek to verify these theoretical insights through experiment.

Here, we consider a simple bandit environment with ten actions where each reward distribution is Bernoulli with $\mu_1 = 0.5$ and $\mu_2 = \cdots = \mu_{10} = 0.45$. Following existing works, we conduct our experiments on cycle and grid graphs. Additionally, we investigate the star graph to verify our theoretical insights. Our results are averaged over 100 independent runs and we represent uncertainty by shading between the 2.5-th and 97.5-th quantiles.

QuACK is initialised with UCB1 (Auer et al., 2002a) and Thompson Sampling (Thompson, 1933). This allows for a fair comparison with existing methods that extend these single-agent algorithms to the multi-agent setting. Specifically, we seek to compare:

- QuACK with UCB1 against Coop-UCB (Landgren et al., 2016) and DDUCB (Martínez-Rubio et al., 2019)
- QuACK with Thompson Sampling against DecTS (Lalitha and Goldsmith, 2021).

Appendix A provides additional details about the hyperparameters used in these algorithms and presents experimental results for bandits with heavy tails.

Figure 2 compares the group regret of each algorithm across various graph structures. For UCB-based algorithms, we observe that QuACK significantly outperforms Coop-UCB and DDUCB. These empirical findings support the claims we made based on the theoretical results in Section 4. For Thompson Sampling algorithms, we observe that our algorithm is competitive with existing work on the cycle and grid structures, and performs significantly better on the star structure, also supporting our theoretical claims.

6 CONCLUSION

This paper proposes a generic black-box algorithm that can extend any single-agent bandit algorithm to the multi-agent setting. Under mild assumptions on the bandit environment, we show that our black-box approach immediately transfers the theoretical guarantees of the chosen bandit algorithm from the single-agent setting to the multi-agent setting.

For the standard bandit setting, we can pair our algorithm with any minimax optimal single agent algorithm and match the minimax lower bound in the multi-agent setting, up to an additive graph-dependent quantity. We suspect that the lower bound is loose because it fails to capture the structure of the graph and the communication delays. Proving graph-dependent lower bounds is an interesting open problem.

Additionally, we assumed that each agent can communicate real-valued messages to each neighbour at the end of every round with arbitrary precision. However, some practical applications impose constraints on the amount of communication in terms of frequency or the number of bits. Relaxing these assumptions are an important next step for this line of research.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback that helped to greatly improve

the clarity and quality of the manuscript.

BH is funded by EPSRC through the Modern Statistics and Statistical Machine Learning CDT. Grant number EP/S023151/1.

References

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1): 137–147, 1999.

J. Audibert and S. Bubeck. Minimax Policies for Adversarial and Stochastic Bandits. In *Proceedings of the 22nd Annual Conference on Computational Learning Theory*, pages 217–226, 2009.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 2002a.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Y. Bar-On and Y. Mansour. Individual Regret in Cooperative Nonstochastic Multi-Armed Bandits. In *Advances in Neural Information Processing Systems*, 2019.

R. Bellman. On a Routing Problem. *Quarterly of Applied Mathematics*, 1958.

D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice-Hall, 1992.

J. Bian and K. Jun. Maillard Sampling: Boltzmann Exploration Done Optimally. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022.

P. J. Bickel. On Some Robust Estimates of Location. *The Annals of Mathematical Statistics*, 36(3):847–858, 1965.

E. Boursier and V. Perchet. A Survey on Multi-player Bandits. *Journal of Machine Learning Research*, 2024.

S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits With Heavy Tail. *IEEE Transactions on Information Theory*, 2013.

O. Catoni. Challenging the empirical mean and empirical variance: A deviation study. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 48(4):1148 – 1185, 2012.

Y. J. Chen, L. Yang, X. Wang, X. Liu, M. Hajiesmaili, J. C. S. Lui, and D. Towsley. On-Demand Communication for Asynchronous Multi-Agent Bandits. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, 2023.

F. R. K. Chung. Diameters and Eigenvalues. *Journal of the American Mathematical Society*, 1989.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

D. L. Cohn. *Measure Theory*. Birkh auser/Springer, New York, 2013.

A. Dubey and A. Pentland. Cooperative Multi-Agent Bandits with Heavy Tails. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Transactions on Automatic Control*, 2012.

M. Elkin. Distributed Exact Shortest Paths in Sublinear Time. *J. ACM*, 2020.

W. Fokkink. *Distributed Algorithms: An Intuitive Approach*. The MIT Press, 2013.

L. Ford. *Network Flow Theory*. RAND Corporation, 1956.

Safwan Hossain, Evi Micha, and Nisarg Shah. Fair Algorithms for Multi-Agent Multi-Armed Bandits. In *Advances in Neural Information Processing Systems*, 2021.

K. Jamieson, S. Katariya, A. Deshpande, and R. Nowak. Sparse Dueling Bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

P. Joulani, Andras A. György, and C. Szepesvári. Online Learning under Delayed Feedback. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

B. Kveton, C. Szepsvári, S. Vaswani, Z. Wen, T. Lattimor, and M. Ghavamzadeh. Garbage In, Reward Out: Bootstrapping Exploration in Multi-Armed Bandits. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

A. Lalitha and A. Goldsmith. Bayesian Algorithms for Decentralized Stochastic Bandits. *IEEE Journal on Selected Areas in Information Theory*, 2021.

P. Landgren, V. Srivastava, and N. E. Leonard. Distributed Cooperative Decision-Making in Multi-Armed Bandits: Frequentist and Bayesian Algorithms. In *IEEE 55th Conference on Decision and Control*, 2016.

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

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

T. Mandel, Y. Liu, E. Brunskill, and Z. Popović. The Queue Method: Handling Delay, Heuristics, Prior Data, and Evaluation in Bandits. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

D. Martínez-Rubio, V. Kanade, and P. Rebeschini. Decentralized Cooperative Stochastic Bandits. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

E. Moore. *The Shortest Path Through a Maze*. Bell Telephone System., 1959.

O. Raveh, J. Honda, and M. Sugiyama. Multi-Player Approaches for Dueling Bandits. Preprint, 2024.

W. Ren, X. Zhou, J. Liu, and N. B. Shroff. Multi-Armed Bandits with Local Differential Privacy. Preprint, 2020.

N. Santoro. *Design and Analysis of Distributed Algorithms*. John-Wiley & Sons, 2006.

Shahin Shahrampour, Alexander Rakhlin, and Ali Jadbabaie. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

B. Szorenyi, R. Busa-Fekete, I. Hegedus, R. Ormandi, M. Jelasity, and B. Balazs. Gossip-Based Distributed Stochastic Bandit Algorithms. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

C. Tekin, S. Zhang, and M. van der Schaar. Distributed Online Learning in Social Recommender Systems. *IEEE Journal of Selected Topics in Signal Processing*, 2014.

W. R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3-4):285–294, 1933.

L. Tran-Thanh, A. Rogers, and N. Jennings. Long-Term Information Collection with Energy Harvesting Wireless Sensors: A Multi-Armed Bandit Based Approach. *Autonomous Agents and Multi-Agent Systems*, 2011.

P. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo. Optimal Algorithms for Multiplayer Multi-Armed Bandits. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020.

L. Xiao and S. Boyd. Fast Linear Iterations for Distributed Averaging. *Systems & Control Letters*, 2004.

L. Yang, Y. J. Chen, S. Pasteris, M. Hajiesmaili, J. C. S. Lui, and D. Towsley. Cooperative Stochastic Bandits with Asynchronous Agents and Constrained Feedback. In *Advances in Neural Information Processing Systems*, 2021.

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed duelling bandits problem. In *Proceedings of the 22nd Conference on Learning Theory*, 2009.

M. Zoghi, S. Whiteson, R. Munos, and M. Rijke. Relative Upper Confidence Bound for the K-Armed Dueling Bandit Problem. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

M. Zoghi, Z. Karnin, S. Whiteson, and M. Rijke. Copeland Dueling Bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes: See Section 2, Assumptions 1 and 2, and Section 3]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes: See Sections 3]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes: Stated within the statement of theoretical results that require the assumptions.]
   (b) Complete proofs of all theoretical results. [Yes: Appendix B contains full proofs and sketches given within the main paper.]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes: See Appendix D]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes: See Appendix D]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes: See Section 5]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes: See Appendix D]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes: Referenced Github links in Appendix D]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [No]

   (d) Information about consent from data providers/curators. [Not Applicable: Publicly available.]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Material

A IMPLEMENTATION DETAILS

Section 3 presents QuACK assuming that we have elected a leader and have computed a shortest path tree for our message passing protocol. However, these quantities need not be known in advance and can be computed in a distributed manner using well-known algorithms.

A.1 Shortest Path Tree

Algorithm 1 requires that each agent knows their parents and children on the shortest path tree, and is presented assuming that these have been pre-computed. Constructing the shortest path tree rooted at a fixed vertex amounts to solving the single-source shortest path problem. Traditionally, we can solve this problem in non-distributed systems using the Bellman-Ford algorithm (Ford, 1956; Bellman, 1958). However, using this algorithm would require the existence of an agent who possesses knowledge of the entire network topology. Therefore, we will present the distributed variant of this algorithm (Bertsekas and Gallager, 1992). Here, each agent knows only their neighbours.

Algorithm 2 presents the pseudo-code for the Distributed Bellman-Ford algorithm. Briefly, this algorithm takes a source agent as input, and each agent keeps track of their distances from the source and their parent on the shortest path tree.

Algorithm 2 Distributed Bellman-Ford (`DBF`)

Input. Index of a source agent v
Set $d_v^1 = 0$ and $d_w^1 = \infty$ for $w \neq v$
Set $\mathcal{P}_w = \emptyset$ for each $w \in V$ to initialise their parent.
for $t \in \{1, 2, \cdots m-1\}$ **do**
Each $w \in V$ sends d_w^t to $z \in N_w$
Each $w \in V$ receives d_z^t from $z \in N_w$
Each $w \in V$ updates their distance from the source

$$d_w^{t+1} = \min_{z \in N_w \cup \{w\}} \left\{ d_z^t + 1\{z \neq w\} \right\}$$

Each $w \in V$ updates their parent:

$$\mathcal{P}_w = \begin{cases} \arg\min_{z \in N_w} d_z^t + 1 & \text{if } d_w^{t+1} < d_w^t \\ \mathcal{P}_w & \text{if } d_w^{t+1} = d_w^t \end{cases}$$

end for
Each $w \in V$ sends their identifier w to their parent $\mathcal{P}_w$.
Each $w \in V$ receives the identifiers from their children and creates their set of children:

$$\mathcal{C}_w = \{z \in V : \mathcal{P}_z = w\}$$

After Algorithm 2 terminates, each agent will know their children and parent on the shortest path tree rooted at the index of the input agent. These are exactly the quantities we require to implement the message-passing scheme described in Section 3.1.

A.2 Leader Election

Leader Election (LE) is a well-known problem in distributed computing (Santoro, 2006; Fokkink, 2013). Here, we want the network to unanimously agree upon a single agent as a leader. Generally, there are impossibility results for anonymous networks (Fokkink, 2013). Therefore, we will assume that each agent has a unique identifier.

Wang et al. (2020) devise a simple leader election scheme that requires at most $d + 1$ iterations, where d is the diameter of the graph. Algorithm 3 presents a slight modification of their procedure that will ease the exposition of finding the optimal leader in the next subsection.

Algorithm 3 Leader Election (`LE`)

Each $v \in V$ sets f_v arbitrarily.
  Section A.3 has each agent set f_v to their sum of shortest paths.
Each $v \in V$ saves their initial value $I_v = (v, f_v)$
Each $v \in V$ creates their state value $S_v^1 = (v, f_v)$
for $t \in \{1, 2, \cdots, m\}$ **do**
  Each $v \in V$ receives S_w^t from $w \in N_v$
  if $\exists w \in N_v$ such that $f_w < f_v$ **then**
    Update $S_v^{t+1} = (w, f_w)$
  else if $\exists w \in N_v$ such that $f_w = f_v$ **then**
    Update $S_v^{t+1} = (\min\{w, v\}, f_v)$
  else
    $S_v^{t+1} = S_v^t$
  end if
end for
if $I_v = S_v^{m+1}$ **then**
  Agent v is the leader.
end if

Running Algorithm 3 requires at most m iterations because every agent will have seen the identifier of every other agent in the graph at the end of this iteration. Furthermore, Algorithm 3 will always terminate with exactly one leader. To see this, we can consider two possible cases.

- *Unique Minimum.* Here, there exists exactly one agent v who possesses the smallest f_v value. Therefore, they never get to update their state, and the network elects them as the leader.
- *Non-Unique Minimum.* Let $f = \min_{v \in V} f_v$ and suppose multiple agents attain this value. Algorithm 3 implements an index-based tie-breaking rule to handle this case. Notably, agents with $f_v = f$ will eventually receive an $f_w = f$ and they will proceed to check if their index is minimal. Since each agent is assumed to have a unique index, there will be exactly one agent who possesses $f_v = f$ with the minimum index. This agent will never update their state, and the network will elect them as the leader.

A.3 Optimising the Leader

Combining Sections A.1 and A.2 allows us to find the best-positioned leading agent in the network according to our theoretical guarantees. Algorithm 4 presents the pseudo-code for finding and electing the leader.

Briefly, this procedure starts with each agent calling Algorithm 2. Afterwards, we perform several rounds of message passing so that agent v can calculate the sum of shortest paths: f_v. Finally, Algorithm 3 performs leader election with f_v equal to the sum of shortest paths.

Algorithm 4 Optimise the Leader

for $v \in V = \{1, 2, \cdots, m\}$ **do**
  Compute Sum of Shortest Paths
  Execute `DBF`(v)
    Each agent now knows d_{vw} (distance from source v)
    Each agent now knows $\mathcal{P}_w$ (their parent on the shortest path tree with source v)
  Each $w \in V$ creates message $M_1^w = (w, d_{vw})$
  for $t \in \{1, 2, \cdots, m\}$ **do**
    Each $w \in V$ sends M_t^w to $\mathcal{P}_w$
    Each $w \in V$ updates $M_{t+1}^w = M_t^w \cup \{M_t^z\}_{z \in \mathcal{C}_w}$
  end for
  Agent v calculates $f_v = \sum_w d_{vw}$ from the messages.
end for
Use Sum of Shortest Paths as f_v in Leader Election
Each agent now knows their value for $f_v = \sum_{w=1}^m d_{vw}$
Execute `LE` to elect the leader with minimal sum of shortest paths.

B MISSING PROOFS

Throughout the main paper we deferred several proofs to the appendix and gave sketches of the proofs for the main results. Here, we present full proofs of the claims made in the main paper and we do so in chronological order. Specifically, the remainder of this section will have the following structure:

- Section B.1 presents a proof of Lemma 1 which establishes the equivalence of playing in the bandit environment and the queued version of the environment created by the feedback from non-leading agents.
- Section B.2 presents a proof of Lemma 2 which tells us that the difference between the group plays and the pseudo plays of the leader is almost surely bounded by a graph-dependent constant.
- Section B.3 presents a full proof of Theorem 1 and fills in steps missing from the proof sketches given in the main paper.

B.1 Proof of Lemma 1

In QuACK, the leader uses rewards from the queues as well as their own observations to *simulate* an environment for updating the bandit algorithm. Recall that the rewards in the queues have been gathered by other agents in the network, who all interact with the same bandit environment. Lemma 1 relates the cumulative regret suffered by the bandit algorithm in the queued environment, created by the leader and the followers, to the cumulative regret that the same algorithm would suffer if it were to just interact bandit environment directly. More precisely it states that:

$$\sum_{a \neq \star} \Delta_a \, \mathbb{E}_P \left[T'_a(\tau) \right] = \sum_{a \neq \star} \Delta_a \, \mathbb{E} \left[T'_a(\tau) \right] \ . \tag{8}$$

where

$$T'_a(\tau) = \sum_{s=1}^{\tau} 1\{\tilde{A}_s = a\}$$

In Equation (8), the first expectation is with respect to the bandit algorithm π interacting directly with the bandit environment. The second expectation is with respect to bandit algorithm π interacting with the *simulated* environment created by the leader and the followers, who all follow Algorithm 1.

For each action $a \in \mathcal{A}$, let P_a be the distribution over possible rewards when playing action a in the bandit environment, as defined in Assumption 1. For each a, we also define $\tilde{P}_a$ which is a probability measure over the possible rewards for playing action a in the simulated environment created for the algorithm by the leader and the followers. In addition let p_π be the density over action-reward pairs up to time τ when the single player bandit algorithm π interacts with the environment, and let $\tilde{p}_\pi$ be the density over action-reward pairs in the

simulated environment up to time n. Following the notation of Lattimore and Szepesvári (2020), we can write down these densities as follows:

$$p_\pi\left(a_1, x_1, \cdots, a_\tau, x_\tau\right) = \prod_{s=1}^{\tau} \pi\left(a_s \,|\, a_1, x_1, \cdots, a_{s-1}, x_{s-1}\right) p_{a_s}\left(x_s\right)$$

$$\tilde{p}_\pi\left(a_1, x_1, \cdots, a_\tau, x_\tau\right) = \prod_{s=1}^{\tau} \pi\left(a_s \,|\, a_1, x_1, \cdots, a_{s-1}, x_{s-1}\right) \tilde{p}_{a_s}\left(x_s\right)$$

where p_a and $\tilde{p}_a$ denote the Radon-Nikodym derivatives of P_a and $\tilde{P}_a$, respectively.

Letting $h = (x_1, a_1, \ldots a_\tau, x_\tau) \in \mathcal{H}_\tau = (\mathcal{A} \times \mathbb{R})^\tau$ represent a possible sequence of τ action-reward pairs, and using these notations, allows us to define the two expectations in Equation (8) more precisely:

$$\mathbb{E}_P\left[T'_a\left(\tau\right)\right] = \int_{h \in \mathcal{H}_\tau} \sum_{(a_s, x_s) \in h} 1\{a_s = a\}\, p_\pi(h)\, dh$$

whereas

$$\mathbb{E}\left[T'_a\left(\tau\right)\right] = \int_{h \in \mathcal{H}_\tau} \sum_{(a_s, x_s) \in h} 1\{a_s = a\}\, \tilde{p}_\pi(h)\, dh$$

Under Assumption 1 we know that the rewards for each action-agent pair are independent and identically distributed random variables. Furthermore, the queuing mechanism reorders the rewards independently of the values they take, e.g. the re-ordering depends on the length of the shortest path. Therefore, using exchangeability of independent and identically distributed random variables, we have that:

$$P_a \overset{d}{=} \tilde{P}_a$$

The Radon-Nikodym Theorem tells us that the p_a is unique, e.g. see Theorem 4.2.2 of Cohn (2013). Combining $P_a = \tilde{P}_a$ with this well-known result tells us that $p_a(x) = \tilde{p}_a(x)$ for all $x \in \mathbb{R}$. Therefore $p_\pi(h) = \tilde{p}_\pi(h)$ for all $h \in \mathcal{H}_\tau = (\mathcal{A} \times \mathbb{R})^\tau$ and hence,

$$\int_{h \in \mathcal{H}_\tau} f\left(h\right) p_\pi\left(h\right)\, dh = \int_{h \in \mathcal{H}_\tau} f\left(h\right) \tilde{p}_\pi\left(h\right)\, dh$$

for any arbitrary function $f : \mathcal{H}_\tau \to \mathbb{R}$. Choosing $f(h) = \sum_{(a_s, x_s) \in h} 1\{a_s = a\}$ completes the proof.

B.2 Proof of Lemma 2

Define the last round that the leading agent plays action a in the bandit environment as follows:

$$z = \max\left\{j \leq t : A^v_j = a\right\}.$$

Recall s_t denotes the pseudo round counter when the leading agent makes their t-th decision in the bandit environment. Using Equation (4) and leveraging the fact that the round in the previous display is associated with a play of action a gives us the following:

$$\begin{aligned}
T'_a\left(s_t\right) &= \sum_{s=1}^{s_t} 1\{\tilde{A}_s = a\} \\
&= \sum_{s=1}^{s_z} 1\{\tilde{A}_s = a\} + \sum_{s=s_z+1}^{s_t} 1\{\tilde{A}_s = a\} \\
&= T'_a\left(s_z\right) + \sum_{s=s_z+1}^{s_t} 1\{\tilde{A}_s = a\} \\
&\overset{(\star)}{=} \sum_{w=1}^{m} T_{aw}\left(z - d_{vw}\right) + \sum_{s=s_z+1}^{s_t} 1\{\tilde{A}_s = a\} \\
&\geq \sum_{w=1}^{m} T_{aw}\left(z - d_{vw}\right) \qquad (9)
\end{aligned}$$

where $(\star)$ follows from the fact that the leader plays action a in round z so the number of times they play this action in the queue must be equivalent to the group plays of this action in the bandit environment.

Now, Equation (9) relates the number of times the bandit algorithm has played action a to the number of times the network has chosen this action. Recall that each follower will play action a for the last time in round:

$$z_w = z + d_{vw}$$

where z is the last round that the leader plays action a in the bandit environment. Using Equation (1) with the definition and properties of z_w allows us to get an upper bound on the number of plays for any non-leading agent w as follows:

$$\begin{aligned}
T_{aw}\left(t - d_{vw}\right) &= \sum_{j=1}^{t-d_{vw}} 1\{A_j^w = a\} \\
&= \sum_{j=1}^{z-d_{vw}} 1\{A_j^w = a\} + \sum_{j=z-d_{vw}+1}^{t-d_{vw}} 1\{A_j^w = a\} \\
&= T_{aw}\left(z - d_{vw}\right) + \sum_{j=z-d_{vw}+1}^{t-d_{vw}} 1\{A_j^w = a\} \\
&\overset{(\star)}{\leq} T_{aw}\left(z - d_{vw}\right) + \sum_{j=z-d_{vw}+1}^{z+d_{vw}} 1\{A_j^w = a\} \\
&\leq T_{aw}\left(z - d_{vw}\right) + 2d_{vw} \qquad (10)
\end{aligned}$$

where $(\star)$ follows from analysing the two possible cases. When $t - d_{vw} \geq z_w = z + d_{vw}$, z_w is the last round follower w plays this action up to and including the t-th round and the summation is empty. Otherwise, $t - d_{vw} < z_w = z + d_{vw}$ and we are extending the summation to include more rounds, which gives us the inequality.

Summing Equation (10) over all agents and employing the inequality from Equation (9) gives us:

$$\begin{aligned}
\sum_{w=1}^{m} T_{aw}\left(t - d_{vw}\right) &\leq \sum_{w=1}^{m} T_{aw}\left(z - d_{vw}\right) + 2\sum_{w=1}^{m} d_{vw} && \text{(Equation (10))} \\
&\leq T'_a\left(s_t\right) + 2\sum_{w=1}^{m} d_{vw} && \text{(Equation (9))}
\end{aligned}$$

as required.

B.3 Proof of Theorem 1

Within Section 3, we provided a nearly complete proof. Here, we present each step for completeness. Firstly, we can rewrite the group plays for action a at the end of the final round as follows:

$$\begin{aligned}
\sum_{w=1}^{m} T_{aw}\left(n\right) &= T_{av}\left(n\right) + \sum_{w \neq v} T_{aw}\left(n\right) \\
&= T_{av}\left(n\right) + \sum_{w \neq v} \sum_{t=1}^{n} 1\{A_t^w = a\} \\
&= T_{av}\left(n\right) + \sum_{w \neq v} \sum_{t=1}^{n-d_{vw}} 1\{A_t^w = a\} + \sum_{w \neq v} \sum_{t=n-d_{vw}+1}^{n} 1\{A_t^w = a\} \\
&= T_{av}\left(n\right) + \sum_{w \neq v} T_{aw}\left(n - d_{vw}\right) + \sum_{w \neq v} \sum_{t=n-d_{vw}+1}^{n} 1\{A_t^w = a\} \\
&\leq T_{av}\left(n\right) + \sum_{w \neq v} T_{aw}\left(n - d_{vw}\right) + \sum_{w \neq v} d_{vw} \qquad (11)
\end{aligned}$$

Secondly, combining Equation (11) with Lemma 2 allows us to upper bound the group plays by the number of times the leader plays the action in the queued version of the environment:

$$\begin{aligned}
\sum_{w=1}^{m} T_{aw}(n) &\leq T_{av}(n) + \sum_{w \neq v} T_{av}(n - d_{vw}) + \sum_{w \neq v} d_{vw} && \text{(Equation (11))} \\
&= \sum_{w=1}^{m} T_{aw}(n - d_{vw}) + \sum_{w=1}^{m} d_{vw} && \text{(Since } d_{vv} = 0\text{)} \\
&\leq T'_a(s_n) + 3 \sum_{w \neq v} d_{vw} && (12)
\end{aligned}$$

where the final line follows from Lemma 2. Plugging Equation (12) allows us to get an upper bound on the group regret:

$$\begin{aligned}
R_G(n) &= \sum_{a \neq \star} \Delta_a \, \mathbb{E}\left[\sum_{w=1}^{m} T_{aw}(n)\right] \\
&\leq \sum_{a \neq \star} \Delta_a \, \mathbb{E}\left[T'_a(s_n) + 3 \sum_{w \neq v} d_{vw}\right] && \text{(Equation (12))} \\
&= \sum_{a \neq \star} \Delta_a \, \mathbb{E}\left[T'_a(s_n)\right] + \left(3 \sum_{w \neq v} d_{vw}\right) \sum_{a \neq \star} \Delta_a \\
&\leq \sum_{a \neq \star} \Delta_a \, \mathbb{E}\left[T'_a(mn)\right] + \left(3 \sum_{w \neq v} d_{vw}\right) \sum_{a \neq \star} \Delta_a && \text{(Since } s_n \leq mn\text{)} \\
&= S_\pi(mn) + \left(3 \sum_{w \neq v} d_{vw}\right) \sum_{a \neq \star} \Delta_a && (13)
\end{aligned}$$

where the final line follows from Lemma 1.

C COMPARISON TO EXISTING WORKS

Our theoretical results span several multi-agent bandit problems, which include: subgaussian rewards, heavy-tailed rewards, duelling bandits and bandits with local differential privacy, amongst any other bandit problems that satisfy Assumption 1. Here, we compare our bounds to existing works in the literature.

C.1 Standard Bandits

Several works design algorithms under the standard assumptions on the reward distributions, such as bounded support, subgaussian tails or specific parametric assumptions (Landgren et al., 2016; Martínez-Rubio et al., 2019; Lalitha and Goldsmith, 2021). These algorithms all fit into the *gossip-based* category. Essentially, these algorithms all use an iterative averaging scheme so that each agent can approximate full network information, which requires the specification of a communication matrix that must possess certain properties.

Suppose $P \in \mathbb{R}^{m \times m}$ is this communication matrix and it has properties:

- $P_{vw} = 0$ if $(v, w) \notin E$
- P must have rows that sum to one.
- P must have columns that sum to one.
- P must have real-valued eigenvalues satisfying:

$$1 = \lambda_1(P) > |\lambda_2(P)| \geq |\lambda_3(P)| \geq \cdots \geq |\lambda_m(P)| \geq 0$$

Xiao and Boyd (2004) show that these proprties imply the following:

$$\lim_{s\to\infty} P^s \to \frac{1}{m}\vec{1}\vec{1}^T$$

Thus, the communication matrix can be used to maintain an approximation of full network information. This is best illustrated through an example. Suppose that we want to approximate the average number of times each agent has played each action. That is, we want to maintain an approximation of the following:

$$\frac{\sum_{w=1}^{m} T_{aw}(t)}{m}$$

Define $\hat{T}_a(t) \in \mathbb{R}^m$ as the vector containing each agent's approximation of the above quantity. Letting $\zeta_a(t) \in \{0,1\}^m$ denote the vector whose entries indicate whether the correspond agent chosen action a in the t-th round. Then, gossip-based algorithms will use a linear iteration of the following form (Xiao and Boyd, 2004):

$$\hat{T}_a(t+1) = P\hat{T}_a(t) + \zeta_a(t) = \sum_{s=1}^{t} P^{t-s+1}\zeta_a(s)$$

where $\hat{T}_a(0) = \vec{0}$. For $s \ll t$ we can use the properties of the communication matrix to argue that:

$$P^{t-s+1}\zeta_a(s) \approx \frac{\sum_{w=1}^{m}\zeta_{aw}(s)}{m}$$

Therefore, summing many such terms will give an approximation of the average plays for each action across the network, and multiplying this by the number of agents gives an approximation of the full network plays. Generally, gossip-based algorithms will use linear iterations of the above form, or a variant thereof, to approximate all quantities required by the chosen bandit algorithm (Landgren et al., 2016; Martínez-Rubio et al., 2019; Lalitha and Goldsmith, 2021). For UCB1, this is the full network plays and the full network sum of rewards for each action.

Landgren et al. (2016) utilise the linear iterations presented above to devise an upper confidence bound algorithm for the multi-agent setting. Lalitha and Goldsmith (2021) apply the same idea to Thompson Sampling. Martínez-Rubio et al. (2019) propose DDUCB, which uses a variant of the scheme that can be viewed as truncating the summation to ensure that it only includes terms where the approximation is sufficiently accurate. They show that this modification leads to improved theoretical guarantees. Thus, DDUCB of Martínez-Rubio et al. (2019) is the focus of our comparison, who prove the following bound on the group regret for reward distributions bounded in $[0,1]$:

$$\sum_{a\neq\star}\frac{8\eta\left(1+\frac{\epsilon}{2}\right)\ln(mn)}{\Delta_a} + (C_G + m + 4)\sum_{a\neq\star}\Delta_a$$

where $\eta > 1$ and $\epsilon \in (0, \eta^{-1}/7(\eta+1))$ are tune-able hyperparameters. Furthermore, C_G is a graph-dependent quantity defined as:

$$C_G = \left\lceil \frac{6m\ln\left(\frac{2m}{\epsilon}\right)}{\sqrt{2\ln|\lambda|^{-1}}} \right\rceil.$$

Corollary 1 shows that QuACK with UCB1 has a comparable but smaller constant pre-multiplying the leading order term. Therefore, we focus on comparing the additive graph dependent terms. These terms are lower order and become negligible for n sufficiently large. However, it is still interesting to provide a comparison, as these terms are informative of how the network topology impacts the performance. Throughout, we will assume that the communication matrix is chosen as-per the recommendations of Martínez-Rubio et al. (2019):

$$P = I - \frac{1}{1+\max_{v\in V}|N_v|}(D - M)$$

where D and M denote the degree matrix and the adjacency matrix of the graph. Formally, D and M have the following element-wise definitions:

$$D_{vw} = \begin{cases} |N_v| & \text{if } w = v \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad M_{vw} = \begin{cases} 1 & \text{if } (v,w) \in E \\ 0 & \text{otherwise} \end{cases}$$

Star Graphs. The star graph consists of $m-1$ vertices connected to a single central vertex. Here, we can exactly evaluate the spectrum of the communication matrix. Notably, the spectrum of $L \coloneqq D - M$ is known for this graph (Chung, 1997):

$$\lambda(L) = \begin{cases} m & \text{with multiplicity } 1 \\ 1 & \text{with multiplicity } m-2 \\ 0 & \text{with multiplicity } 1 \end{cases}$$

Therefore, the spectrum of the communication matrix P can be found by plugging in these quantities:

$$\lambda(P) = 1 - \frac{\lambda(L)}{m} = \begin{cases} 0 & \text{with multiplicity } 1 \\ \frac{m-1}{m} & \text{with multiplicity } m-2 \\ 1 & \text{with multiplicity } 1 \end{cases}$$

which gives us $\lambda = \lambda_2(P) = {}^{m-1}/_{m}$. Plugging this into their graph-dependent term yields:

$$\begin{aligned} C_G + m + 4 &= \left\lceil \frac{6m \ln\left(\frac{m}{\epsilon}\right)}{\sqrt{2 \ln\left(\frac{m}{m-1}\right)}} \right\rceil + m + 4 \\ &\geq \frac{6m \ln(m)}{\sqrt{2 \ln\left(\frac{m}{m-1}\right)}} + m + 4 \\ &> 7m + 4 \end{aligned}$$

where the final inequality follows from the fact that $m \geq 2$ is required for the multi-agent setting. Running Algorithm 4 will choose the central vertex as the leading agent. Therefore, the length of the shortest path from every follower to the leader is 1 and this will give make the graph-dependence in our algorithm:

$$3 + 3 \sum_{w=1}^{m} d_{vw} = 3 + 3(m-1) = 3m$$

which is strictly smaller than that of Martínez-Rubio et al. (2019).

Regular Graphs. Graphs are called δ-regular if every agent has exactly δ neighbours, e.g. $|N_v| = \delta$ for all $v \in V$. For these graphs, the expression for P simplifies and allows us to obtain an expression for the spectral term found in the graph-dependent term of Martínez-Rubio et al. (2019):

$$P = \frac{I + M}{1 + \delta} \implies \lambda \coloneqq \lambda_2(P) = \frac{1 + \lambda_2(M)}{1 + \delta}$$

Chung (1989) provide an upper bound on the diameter d for regular graphs, which we can substitute into the graph-dependent term in the group regret of QuACK-UCB:

$$\begin{aligned}
3 + 3\sum_{w=1}^{m} d_{vw} &\leq 3 + 3d\,(m-1) \\
&\leq 3 + \left\lceil \frac{3\,(m-1)\ln\,(m-1)}{\ln\left(\frac{1+\delta}{|1+\lambda_2(M)|}\right)} \right\rceil \\
&= 3 + \left\lceil \frac{3\,(m-1)\ln\,(m-1)}{\ln|\lambda|^{-1}} \right\rceil \\
&= 3 + \left\lceil \frac{6\,(m-1)\ln\,(m-1)}{\sqrt{2\ln|\lambda|^{-1}}\sqrt{2\ln|\lambda|^{-1}}} \right\rceil \\
&\overset{(a)}{<} 3 + \left\lceil \frac{6\,(m-1)\ln\left(\frac{2m}{\epsilon}\right)}{\sqrt{2\ln|\lambda|^{-1}}\sqrt{2\ln|\lambda|^{-1}}} \right\rceil \\
&= 3 + \frac{C_G - 6\ln(m-1)}{\sqrt{2\ln|\lambda|^{-1}}} \\
&\leq 3 + \frac{C_G}{\sqrt{2\ln|\lambda|^{-1}}}
\end{aligned}$$

where (a) follows from the fact that $\ln(2m/\epsilon) > \ln(m-1)$ for $m \geq 2$ and $\epsilon \in (0,1)$. This establishes that we can upper bound the graph-dependence in our theoretical guarantees by the a function involving the graph-dependent term found in that of Martínez-Rubio et al. (2019). Notably, better dependence for our algorithm is guaranteed whenever:

$$\sqrt{2\ln|\lambda|^{-1}} \geq 1 \implies |\lambda| \geq \exp\left(-\frac{1}{2}\right)$$

Generally, this suggests that the graph-dependence of our leader-based approach is better for δ-regular graphs where the communication matrix has large enough values for $|\lambda|$.

C.2 Heavy-Tailed Rewards

Dubey and Pentland (2020) design a multi-agent extension of the robust upper confidence bound algorithm using the truncated mean estimator. To the best of our knowledge, this is the only work addressing heavy-tailed environments in the multi-agent setting. Before stating their theoretical results, we first define the independence number of a graph:

$$\alpha\,(G) = \max_{S \subseteq V} \{|S| : (v,w) \notin E \text{ for all } (v,w) \in S\}$$

which is the size of the largest subset of vertices where no two vertices are adjacent. Dubey and Pentland (2020) develop several algorithms, and their best guarantee on the group regret up to absolute constants is given by:

$$\alpha\,(G_\gamma)\sum_{a \neq \star} \frac{\sigma^{\frac{1}{\epsilon}}\ln\,(n)}{\Delta_a^{\frac{1}{\epsilon}}} + (m\gamma \cdot \alpha\,(G_\gamma) + \alpha\,(G_\gamma) + m)\sum_{a \neq \star}\Delta_a$$

where ϵ controls the heaviness of the tails for the reward distributions, γ is a parameter their algorithm takes as input that governs how far each agent can communicate, and the γ-th power of the graph is defined as:

$$G_\gamma = (V, E_\gamma) \;\text{ where }\; E_\gamma = 1\{(v,w) \in V \times V : d_{vw} \leq \gamma\}$$

which adds an edge to the graph whenever $d_{vw} \leq \gamma$. From Corollary 3, the group regret of running QuACK with the robust upper confidence bound using the truncated-mean estimator has the following upper bound:

$$R_G\,(n) \leq \mathcal{O}\left(\sum_{a \neq \star}\frac{\sigma^{\frac{1}{\epsilon}}\ln\,(mn)}{\Delta_a^{\frac{1}{\epsilon}}} + \left(\sum_{w=1}^{m} d_{vw}\right)\sum_{a \neq \star}\Delta_a\right)$$

Thus, we obtain a smaller leading order term whenever:

$$n^{\alpha(G_\gamma)-1} \geq m$$

Notably, $\alpha(G_\gamma) \in [1, m]$ for all graphs. Therefore, our graph-dependence is smaller whenever $\alpha(G_\gamma) > 1$ and $m < n$, e.g. the number of agents is smaller than the number of rounds.[1] Specifying $\gamma \geq d$ as the input parameter will yields an independence number of 1 in the bound Dubey and Pentland (2020). However, their analysis is specific to the robust upper confidence bound algorithm and the truncated mean estimator, whereas Theorem 1 is strictly more general.

Dubey and Pentland (2020) have an additive graph-dependent term that displays a trade-off between the input parameter γ and the independent number. Generally, increasing γ will decrease the independence number and decreasing γ will increase the independent number. Therefore, we expect their additive term to be comparable to ours in many cases. For example, choosing $\gamma = d$ will minimise the leading order term in the regret bound and yields an additive graph-dependence of:

$$m\gamma \cdot \alpha(G_\gamma) + \alpha(G_\gamma) + m = md + m + 1$$

Conversely Corollary 3 shows that our additive graph-dependence of the order:

$$\sum_{w=1}^{m} d_{vw} \leq m(d-1)$$

C.3 Duelling Bandits

Raveh et al. (2024) design a multi-agent extension of various duelling bandit algorithms. To the best of our knowledge, this is this only work addressing the multi-agent duelling bandit problem. Their best guarantee is for an extension of the relative upper confidence bound algorithm (Zoghi et al., 2014). Up to absolute constants, they show that this algorithm achieves an upper bound on the group regret given by:

$$\chi(G_\gamma) \sum_{a \neq \star} \frac{\ln(n)}{\tilde{\Delta}_a} + m^2 k^2 \left(1 + \ln\left(1 + |N_\star^\gamma|\right)\right) \tilde{\Delta}_{\max} + (1+\gamma) km \tilde{\Delta}_{\max}$$

where $\chi(G_\gamma) \in [1, m]$ denotes the clique covering number of the γ-th power of the graph and γ is an input parameter. From Corollary 4, we can upper bound the group regret when running QuACK with the relative upper confidence bound algorithm:

$$R_G(n) \leq \mathcal{O}\left(\sum_{a \neq \star} \frac{\ln(mn)}{\tilde{\Delta}_a} + \left(\sum_{w=1}^{m} d_{vw}\right) \sum_{a \neq \star} \tilde{\Delta}_a\right)$$

Comparing the leading-order term reveals that we obtain a strictly better leading order term whenever:

$$n^{\chi(G_\gamma)-1} \geq m$$

Therefore, our graph-dependence is smaller whenever $\chi(G_\gamma) > 1$ and $m < n$, e.g. the number of agents is smaller than the number of rounds. Specifying $\gamma \geq d$ as the input parameter will yield a clique covering number of 1 in the bound of Raveh et al. (2024). However, their analysis is specific to the relative upper confidence bound algorithm, whereas Theorem 1 is strictly more general.

Raveh et al. (2024) have an additive graph-dependent term that is strictly larger. Notably, their graph-dependence is always quadratic in the network size, whereas ours scales as md. Notably, $d < m$ in connected graphs.

[1] Note that $m > n$ implies that all multi-agent bounds on the group regret become linear in the horizon.

D ADDITIONAL EXPERIMENTS

Experimentally, we seek to compare our black-box reduction to existing works, who design and analyse extensions of well-known single agent bandit algorithms. Our simulations were conducted on a personal machine (Intel i5-10310U CPU @ 1.70Ghz x 8). Furthermore, all results are averaged over 100 independent runs.

D.1 Subgaussian Experiments

Here, we describe the missing experimental details for the experiments found in the main paper, such as the hyperparameters for existing algorithms. Firstly, Landgren et al. (2016), Martínez-Rubio et al. (2019) and Lalitha and Goldsmith (2021) are all gossip-based algorithms. Thus, they all require the specification of a communication matrix. All authors recommend choosing this matrix based on the graph Laplacian. Formally, they use the following doubly stochastic matrix:

$$P = I - \frac{1}{1 + \max_{v \in V} |N_v|} (D - M)$$

where M, D and δ denote the adjacency matrix, the degree matrix and the maximum degree of the underlying graph. Below, we provide specific details on additional hyperparameters of each algorithm.

- Landgren et al. (2016) (coop-UCB) has been implemented by other researchers (Martínez-Rubio et al., 2019). Their algorithm requires an exploration hyperparameter $\gamma > 1$; we choose this parameter based on the empirical results of existing work (Martínez-Rubio et al., 2019; Lalitha and Goldsmith, 2021).
  - Exploration Parameter: $\gamma = 1.01$.
- Martínez-Rubio et al. (2019) (DDUCB) provide an implementation of their algorithm on Github. Their algorithm requires specifying two hyperparameters; we use the values stated in their theorems and chosen in their experiments.
  - Exploration Parameter (Theorem 3.2 of Martínez-Rubio et al. (2019)): $\eta = 2$.
  - Approximation Parameter (Theorem 3.2 of Martínez-Rubio et al. (2019)): $\epsilon = 1/22$.
- Lalitha and Goldsmith (2021) (DecTS) provide an implementation of their algorithm on Github. Their algorithm requires the specification of a learning rate. We use the value found in the theoretical results that they use in their experiments.
  - Learning Rate (Theorem 1 of Lalitha and Goldsmith (2021)): $\eta = m$.
- Algorithm 1 requires the specification of a leader and a single-player bandit algorithm.
  - Leader: As Theorem 1 suggests, we select the leader as the solution to the graph median problem for each experimental setting. We do so using Algorithm 4.
  - Bandit Algorithm: Our goal is to compare how well our black-box reduction performs relative to existing works, who design and analyse specific algorithms designed for the setting. Therefore, we consider UCB (Auer et al., 2002a) and Thompson Sampling (Thompson, 1933) to compare with existing literature.

Our experiments consider three network structures, namely cycle, grid and star graphs with varying network sizes. Figure 2 in the main paper shows the experimental results for the largest network size of $m = 196$ agents. Here, we present the experimental results for smaller network sizes.

Cycle Graph. The cycle graph consists of agents organised in a circle who are connected to their clockwise and anti-clockwise neighbours. Thus, the graph is 2-regular. Section 5 presents the empirical results for a cycle with $m = 196$ agents. Figure 3 displays the group regret for cycle graphs with varying numbers of agents.

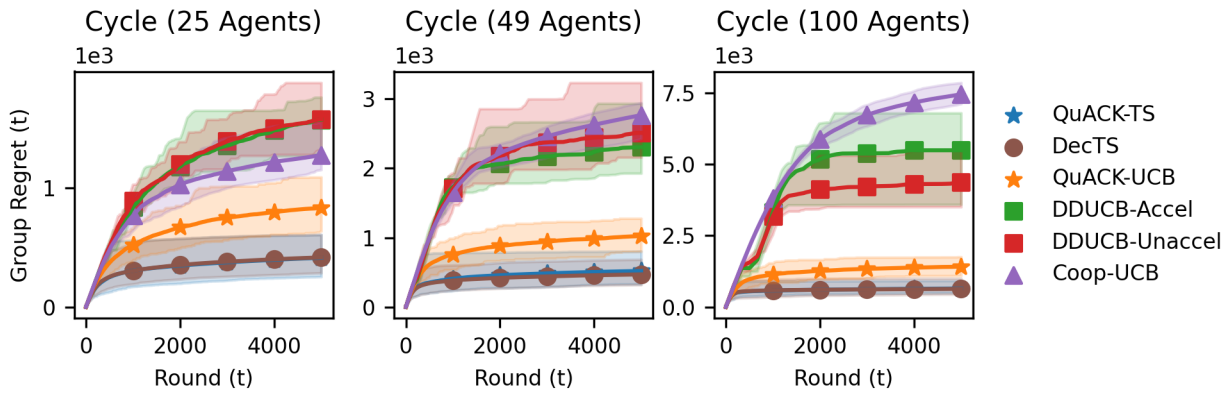

Figure 3: Group Regret for Cycle Graphs.

Figure 3, shows that QuACK-UCB outperforms the other UCB-based algorithms on all network sizes. As Section C.1 suggests, the performance gap increases with the size of the network. Furthermore, QuACK-TS is competitive with DecTS.

Grid Graph. The grid graph consists of agents organised in a square lattice and each agent can have two, three or four neighbours. Section 5 presents the results of our experiments for a grid graph with $m = 196$ agents. Figure 4 displays the group regret for grid graphs with varying numbers of agents.

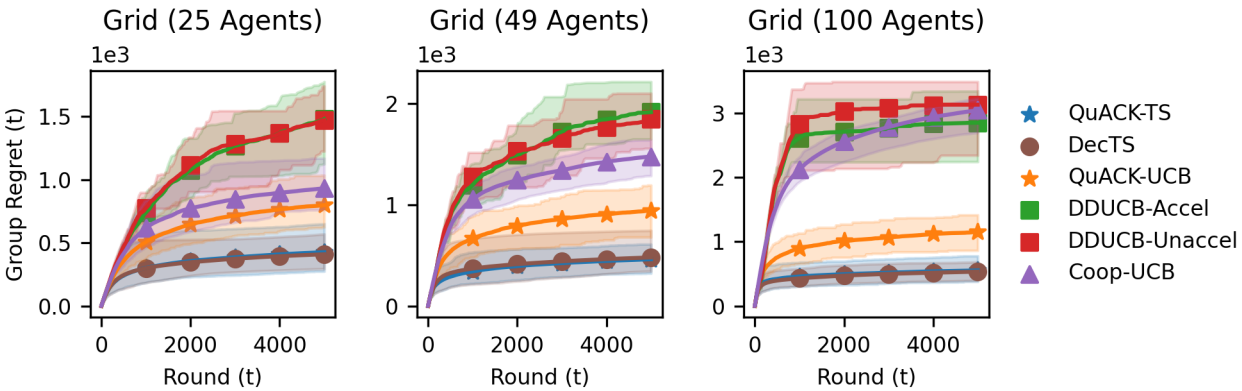

Figure 4: Group Regret for Grid Graphs.

Here, QuACK-UCB out performs the other UCB-based algorithms on all network sizes. Furthermore, QuACK-TS is competitive with DecTS.

Star Graph. The star graph is a common sub-structure that appears in social networks. This particular graph consists of a single central vertex, who is connected to all other vertices. The non-central vertices have exactly one neighbour, which is the central vertex. Section 5 presents the empirical results for a star with $m = 196$ agents. However, we also conduct experiments on smaller stars to verify our theoretical results. Figure 4 displays the group regret for cycle graphs with varying numbers of agents.

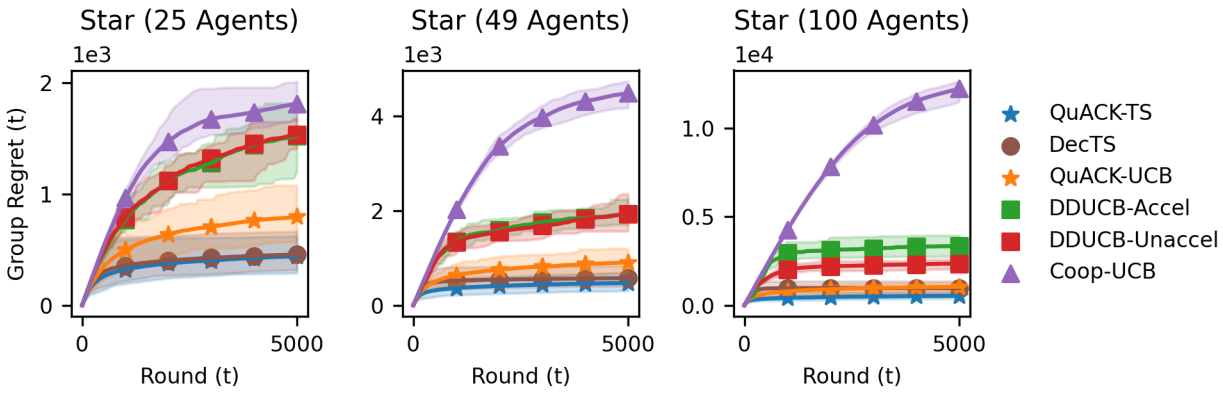

Figure 5: Group Regret for Star Graphs.

Here, QuACK-UCB out performs the other UCB-based algorithms on all network sizes. Furthermore, QuACK-TS is competitive with DecTS on the smallest star network. However, QuACK-TS achieves smaller group regret for larger star networks.

Network Scaling. Finally, Figure 6 visualises the group regret at the end of the final round as a function of the network size for each graph structure.

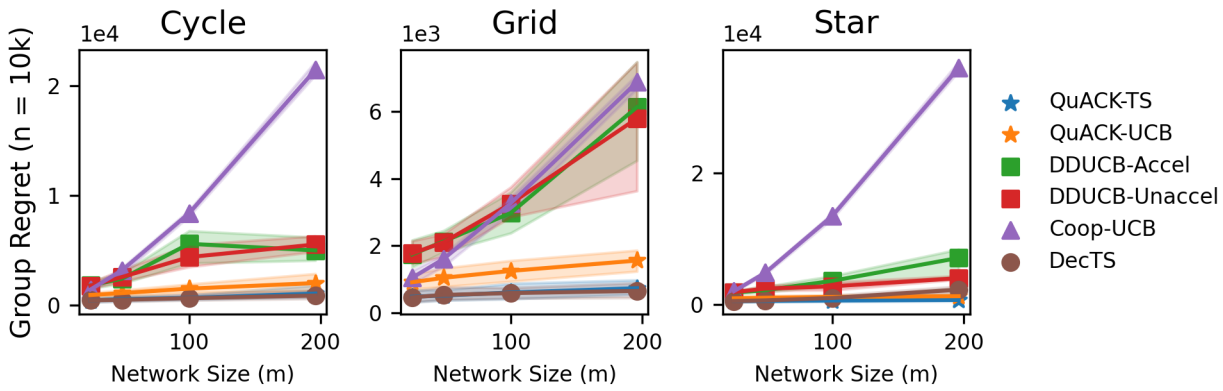

Figure 6: Network Size Scaling of Group Regret.

Notably, we can see the group regret of Coop-UCB and DDUCB scales quickly with the network size for each graph and this validates the theoretical insights from Section C.1. Conversely, the group regret of QuACK-UCB, QuACK-TS and Dec-TS have a shallower gradient, indicating a better dependence on the network size and topology.

D.2 Heavy-Tailed Experiments

Following Dubey and Pentland (2020), we conduct our experiments on standard α-stable densities. Specifically, $\alpha = 1.9$ in our experiments and we select the location of the densities such that:

$$\mu_a = \begin{cases} 0.7 & \text{if } a = 1 \\ 0.4 & \text{if } a \neq 1 \end{cases}$$

where $\mathcal{A} = \{1, 2, \cdots, 5\}$. Below, we provide specific details of the hyperparameters used for each algorithms.

- Dubey and Pentland (2020) obtain the best theoretical guarantees for CMP-UCB and this algorithm effectively partitions the graph into disjoint subsets. Each of these subset has one leader that uses the robust upper confidence bound algorithm. Following their guidance, we select the leaders such that they form the maximal weighted independent set of each graph.
  - To test our theoretical predictions, we run their algorithm with 1, 2 and 4 leaders on each graph. Notably, with 1 leader we expect their algorithm to outperform ours. With ≥ 2 leaders, we expect our algorithm to perform best.
- Algorithm 1 requires a single-agent heavy-tailed bandit algorithm as input. For a direct comparison, we choose the robust upper confidence bound algorithm with the truncated mean estimator (Bubeck et al., 2013). Similarly to the subgaussian setting, we select the leader as the arg min of the graph-median problem.

Our experiments in this setting consider the cycle and grid network structures with $m \in \{16, 36, 64\}$ agents. These experiments use a smaller number of agents and fewer rounds because the computational complexity of computing the truncated mean estimator grows linearly with the number of reward samples for each action.[2]

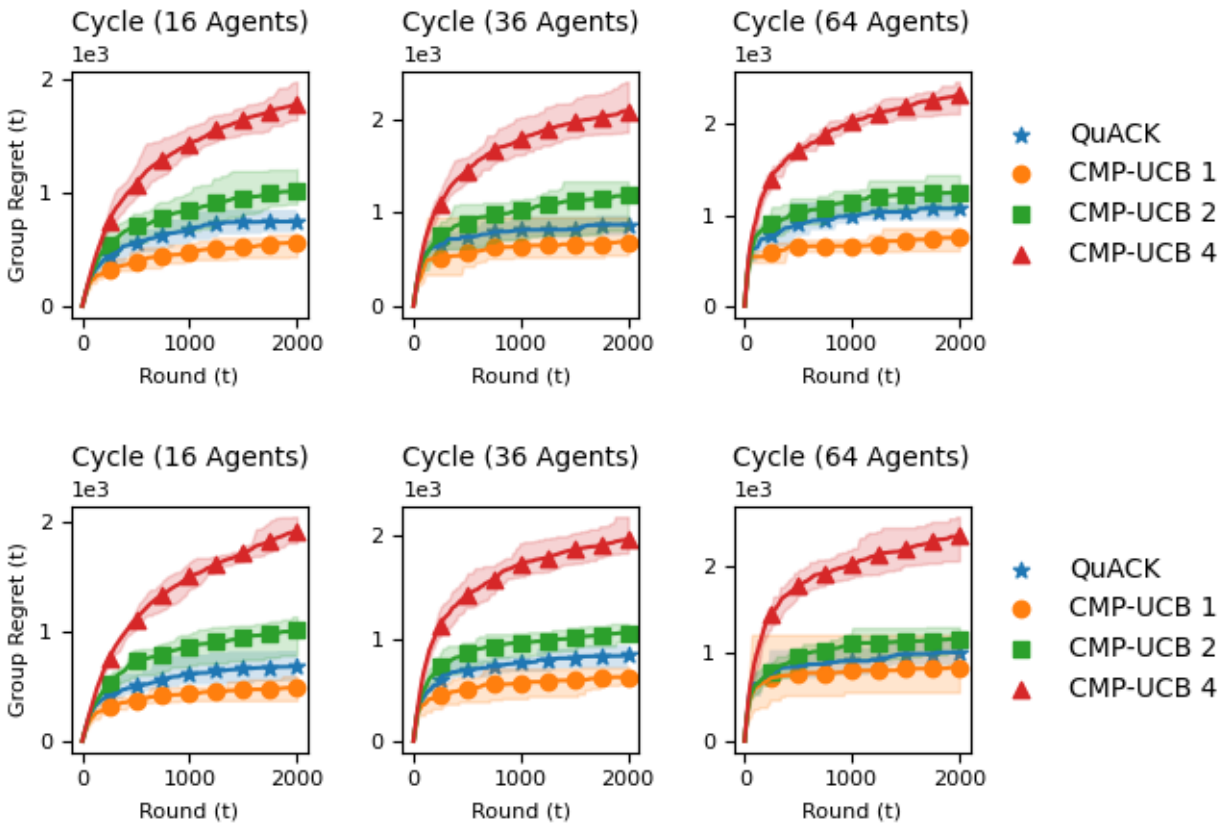

Figure 7: Group Regret for Cycle and Grid Graphs.

Figure 7 displays the group regret for the various cycle and grid graphs. From Section C.2, we expect QuACK will perform better than CMP-UCB when it receives a γ such that the resulting graph has independence number is greater than 1. Notably, our experimental results align exactly with our theoretical predictions.

[2]The number of rewards for each action depends implicitly on the network size in the multi-agent setting.