Creator Incentives in Recommender Systems: A Cooperative Game-Theoretic Approach for Stable and Fair Collaboration in Multi-Agent Bandits

Ramakrishnan Krishnamurthy
Courant Institute, New York University

Arpit Agarwal
Indian Institute of Technology Bombay

Lakshminarayanan Subramanian
Courant Institute, New York University

Maximilian Nickel
FAIR, Meta AI

Abstract

User interactions on online recommendation platforms create interdependencies among content creators: feedback on one creator's content influences the system's learning and, in turn, the exposure of other creators' contents. To analyze incentives in such settings, we model collaboration as a multi-agent stochastic linear bandit problem with a transferable utility (TU) cooperative game formulation, where a coalition's value equals the negative sum of its members' cumulative regrets.

We show that, for identical (homogeneous) agents with fixed action sets, the induced TU game is convex under mild algorithmic conditions, implying a non-empty core that contains the Shapley value and ensures both stability and fairness. For heterogeneous agents, the game still admits a non-empty core, though convexity and Shapley value core-membership are no longer guaranteed. To address this, we propose a simple regret-based payout rule that satisfies three out of the four Shapley axioms and also lies in the core. Experiments on MovieLens-100K dataset illustrate when the empirical payout aligns with—and diverges from—the Shapley fairness across different settings and algorithms.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

Collaborative learning has emerged as a powerful paradigm in machine learning, enabling multiple agents to improve overall task performance by data or computational resources [Kairouz et al., 2021, Zhang et al., 2021]. This framework is particularly salient in large-scale online recommender systems, where interactions between users and content creators serve as implicit signals for model optimization. These platforms leverage user feedback to jointly model user preferences and content characteristics, thereby learning latent structures that generalize across the population [Ko et al., 2022, Zhang et al., 2019]. In many of these systems, a creator's revenue is directly tied to user engagement metrics, which in turn depend on how well their content is recommended. Consequently, the learning and reward dynamics are coupled—what the system learns from feedback on one creator's content can affect how another creator's content is recommended, and ultimately, monetized. This introduces a subtle but critical interplay between data-sharing, learning dynamics, and economic outcomes for creators [Qian and Jain, 2024, Hron et al., 2023, Zhu et al., 2023a]. To vividly illustrate this interdependence, consider the following simplified scenario:

Example 1 (Online Recommender Platform)**.** *Let Alice and Bob be content creators, producing content on apples and bananas, respectively. A new user visits the platform, and the system initially recommends Alice's content about apples. Two outcomes are possible:*

1. ***Positive feedback.*** *The user engages positively with Alice's content. The system infers a preference for fruit-related topics and subsequently recommends Bob's content about bananas.*
2. ***Negative feedback.*** *The user disengages or re-*

sponds negatively. The system infers a disinterest in fruit content and refrains from recommending Bob's content to the user.

In both cases, Bob's reward hinges on how the platform generalizes the user's preference from Alice's content:

1. *If the generalization aligns with the user's true interests, Bob either gains a satisfied user or avoids an unsatisfactory impression—both favourable.*
2. *However, if the generalization is inaccurate, Bob either suffers from unwarranted exposure or misses out on a potentially interested user.*

Thus, the system's inference from one creator's data directly impacts another's opportunity and reward, highlighting the entangled fates of collaborating agents in such platforms.

This simplified example abstracts away many practical complexities. In real-world systems, creators may be *homogeneous*—equally capable of producing various content types for the same user population—or *heterogeneous*, with specialized content expertise and/or access to a segmented audience. Nonetheless, the example drives home the central research question:

How does collaboration among learning agents affect one another?

Is it possible to ensure collectively rational participation from all agents?

1.1 Our Contributions

We model this learning scenario as a multi-agent collaborative bandit problem, which captures the exploration-exploitation trade-off (mirroring the tension between learning and revenue generation in recommendation systems). To quantify how an agent's learning affects others, we model the inter-agent dynamics using a transferable utility (TU) coalition game, where the value of a coalition reflects the collective regret reduction achieved through data sharing among the members of the coalition. The formal setup is introduced in Section 2.

Using tools from cooperative game theory, we analyze how properties of the learning problems and algorithmic behaviour/assumptions influence coalition formations and collaborative incentives. To this end, our key contributions are as follows.

1. **Homogeneous Agents with Fixed Action Sets (Section 3):** We consider a symmetric setting where all agents share the same action space across time. Under mild conditions on the learning bandit algorithm, we prove in Theorem 1 that the induced TU game is convex, ensuring the core is non-empty and contains the Shapley value. This implies that full collaboration (i.e., grand coalition formation) is both stable and equitable.
2. **Heterogeneous Agents with Diverse Action Sets (Section 4):** We then analyse more realistic settings where agents differ in their available actions (e.g., content specializations). We show in Theorem 2 that, under some algorithmic assumptions, the resulting game still has a non-empty core, but may not contain the Shapley value. To address this, we propose a simple, regret-based payout scheme and prove in Theorem 3 that, under the assumptions, it satisfies all but one of the Shapley value axioms, providing a practical and principled solution.
3. **Empirical Validation (Section 5):** We conduct numerical simulations on problem instances derived from MovieLens-100k dataset, illustrating how the empirical payout structure aligns with or diverges from the Shapley value in these settings.

1.2 Related Work

For a general survey on linear bandits, see Lattimore and Szepesvári [2020], and for a textbook treatment of cooperative game theory, see Osborne and Rubinstein [1994].

Collaborative Multi-Agent Bandits. In the regret minimization setting, several algorithms have been proposed to upper bound different regret notions across agents. A common goal is minimizing the sum (arithmetic mean) of individual regrets, for which optimal bounds are known [Wang et al., 2019]. Baek and Farias [2021] study 'grouped bandits', where users arriving over time belong to groups (comparable to agents, in our setting) and propose UCB algorithms optimizing Nash Social Welfare (NSW), equivalent to maximizing the product (geometric mean) of regret reduction that the different groups get by collaboration. For identical agents sharing a common action set, Yang et al. [2023] derive optimal bounds on *any* agent's regret (maximum), conservatively bounding the performance of the worst agent. When the agents are rational and self-interested, equilibrium guarantees for their behaviour in both the collaborative setup [Bolton and Harris, 1999, Ramakrishnan et al., 2024] and the competitive setup [Aridor et al., 2024] have been shown.

Heterogeneous Agents. For agents with heterogeneous action sets (non-identical), general instance-dependent bounds remain elusive to the best of our

knowledge. Raghavan et al. [2018] construct an example wherein collaboration using LinUCB [Abbasi-Yadkori et al., 2011] increases regret of an agent. Nonetheless, such heterogeneity is also shown to naturally help in exploration. Specifically, Kannan et al. [2018] show that a myopic greedy algorithm (albeit, after some initial exploration) achieves non-trivial regret bounds, when some heterogeneity is ensured by random perturbations of actions. Wang et al. [2023] further consider agents with 'free' arms—actions that incur no regret—which enables effective exploration for others and yield regret bounds leveraging this heterogeneity.

Shapley Values in Machine Learning and Social Systems. Originating from game theory, Shapley values have been widely adopted to model fairness and marginal impact in ML and social systems. Applications include identifying influential nodes in social networks [Narayanam and Narahari, 2010], incentivizing truthful data sharing [Chessa and Loiseau, 2017], and attributing value in online services like surveys and recommendations [Kleinberg et al., 2001]. In explainable AI, they quantify feature importance in model outputs [Lundberg and Lee, 2017] and evaluate training data contributions [Ghorbani and Zou, 2019, Jia et al., 2019]. Since exact computation is exponential in the number of players/quantities, efficient polynomial-time approximations are used in practice [Mann and Shapley, 1960, Musco and Witter, 2024].

Strategic Aspects in Recommender Systems. There is a rich literature that studies how rational and self-interested content creators (supply side) and viewers/users (demand side) behave in two-sided markets such as recommender system. A line of work models content creation as a strategic *game* in which creators choose/create content to maximize exposure to user demand under a given recommendation rule [Ben-Porat and Tennenholtz, 2018, Jagadeesan et al., 2023, Yao et al., 2023], and how coordinated creators may jointly adjust their strategies [Yu et al., 2025]. These works fix a specific learning/recommendation algorithm and analyze the content creation behaviour and choices. In contrast, we take creators' content as a given, and instead ask how the algorithm should behave so that the resulting utilities satisfy desirable notions of fairness and stability.

Empirical evidence suggests that although the precise recommendation algorithm is typically opaque to creators, they anthropomorphize the algorithm and attribute to it distinct 'personas', adapting their content strategies accordingly to maximize exposure [Wu et al., 2019]. On the demand side, Fedorova et al. [2025] study how groups of users (content viewers) strategically interact with the contents to amplify or counteract algorithmic suppression to tune their future recommendations.

2 SETTING & PRELIMINARIES

Multi-agent bandit problem. There is a set M of agents who all play a common linear bandit instance for a time period of T. (By a slight abuse of notation, we also use M to denote the total number of agents.)

We define a multi-agent linear bandits problem instance by the tuple $I = (\theta^*, \mathrm{X})$, where $\theta^* \in \mathbb{R}^d$ is an unknown model parameter in ambient dimension d, and $\mathrm{X} = (X_{a,t})_{a \in M, t \in [T]}$ denotes the complete profile of action sets across agents and time. At every time step $t \in [T]$, each agent $a \in M$ is presented with a set of actions $X_{a,t} \subset \mathbb{R}^d$. The agent chooses and plays an action $x_{a,t} \in X_{a,t}$, and observes a stochastic reward $y_{a,t} = \langle \theta^*, x_{a,t} \rangle + \eta_{a,t} \in \mathbb{R}$, where $\eta_{a,t}$s are i.i.d. zero-mean sub-gaussian random variables as is standard in the bandit literature.

We introduce some useful notations next. Write $H_{a,t} := (x_{a,s}, y_{a,s})_{s=1}^t$ to be the history of all actions played and rewards observed by agent a up to time t. Write $x^*_{a,t} := \arg\max_{x \in X_{a,t}} \langle \theta^*, x \rangle$ to be the optimal action that maximizes the expected reward for agent a at time t.

Nature of collaboration. We permit agents to communicate and collaborate amongst themselves by forming *coalitions*. Before bandit playing commences, the agents partition themselves into a collection $\mathcal{C}$ of disjoint coalitions. Then, for each coalition $C \in \mathcal{C}$, all agents in the coalition shall reveal all their played actions and observed rewards to all other agents within their coalition. So, each agent $a \in C$ can decide action $x_{a,t}$ at time t using coalition history $H^C_{t-1} := \{H_{b,t-1} : b \in C\}$ up to the previous time step $t-1$. Inversely, no information is shared between any two agents who are not in the same coalition. The coalition shall make use of a multi-agent bandit algorithm $\textsc{Alg}$ that at every time t, takes in the coalition history H^C_{t-1} as input, and outputs a profile of actions $(x_{a,t})_{a \in C}$ for all agents in the coalition to play.

Next, we define the expected pseudo-regret (simply called 'regret' henceforth) of agent a in coalition C that uses $\textsc{Alg}$ on a problem instance $I = (\theta^*, X)$ for a time period T as follows:

$$R^C_a(\textsc{Alg}, I, T) := \sum_{t=1}^{T} \langle \theta^*, x^*_{a,t} \rangle - \mathop{\mathbb{E}}_{I, \textsc{Alg}} \left[\langle \theta^*, x_{a,t} \rangle \right], \tag{1}$$

where the expectation is over both the stochasticity in the rewards and any internal randomness used by

the algorithm. We also use some simpler notations: $R_a(.)$ denotes $R_a^{\{a\}}(.)$ of a singleton coalition, and when the context is clear, the dependence on the problem instance, the algorithm used, and/or time are implicitly baked into the expression R_a^C or $R_a^C(\textsc{Alg}, T)$ in place of $R_a^C(\textsc{Alg}, I, T)$.

Transferable Utility Game. We use the Transferable Utility (TU) game from Co-operative Game Theory to model our setting here. We define the characteristic/value function v—that intrinsically depends on the time horizon T, problem instance I, and the multi-agent bandit algorithm $\textsc{Alg}$ used—as follows: for every coalition $C \in 2^M$,

$$v_{\textsc{Alg},I,T}(C) = -\sum_{a \in C} R_a^C(\textsc{Alg}, I, T) \tag{2}$$

to be the sum of negative (expected) regrets of all agents in the coalition. A higher value $v.(C)$ corresponds to a lower total regret for agents in coalitions C and is therefore preferable. [1] When the context is clear, we simply use $v(C)$ to denote $v_{\textsc{Alg},I,T}(C)$.

We denote $(M, v_{\textsc{Alg},I,T})$ to be the *collaboration game* and shall study this game for different classes of instances I and algorithms $\textsc{Alg}$.

2.1 Mapping our model to Recommender Systems

We provide a part-by-part comparison of our model setting and its real-world interpretation.

Model: Agent 1 takes an action and observes a reward. Agent 1 then shares details about both the action and reward with Agent 2, who uses this information to decide their own action.

Direct translation: Creator 1 (e.g., a YouTube channel) selects and recommends a piece of content to a user, observes the user's engagement (e.g., a like or a click), and shares both the content and feedback with Creator 2 (another channel), who then uses this to choose what content of his to recommend.

Application reality: However, in practice, the platform (e.g., YouTube) is the one that recommends content from Creator 1 to a user, observes user feedback, and internally uses this data to improve recommendations, including recommending content from Creator 2 to another user.

[1] Instead of a value function, one could also define the TU game with a *cost* function that equals the (positive) sum of regrets of all agents, and lower cost is desireable. Nevertheless, we use value functions as it is more widely used.

In our model, we abstract away the platform and represent its centralized learning and coordination as if agents (creators) were directly sharing full information with each other. This abstraction/simplification allows us to study the system's learning and strategic dynamics more transparently (as a multi-agent bandit problem and a cooperative game among the agents), while still capturing the essence of how feedback from one creator's content can influence the recommendations for others.

3 FIXED ACTION SETS

In this section, we restrict our attention to a simple family of multi-agent bandit instances with fixed action sets. Let $\mathcal{I}^{\text{fixed}} \subset \mathcal{I}$ be the set of all problem instances with a single finite action set shared throughout the time horizon by all agents. That is, for all instances $I \in \mathcal{I}^{\text{fixed}}$, there exists some $X' \subset \mathbb{R}^d$ of finite cardinality s.t. $X_{a,t} = X'$ for all agents $a \in M$ and time steps $t \in [T]$.

We describe a multi-agent bandit algorithm $\textsc{Mul}$ to play these instances $I \in \mathcal{I}^{\text{fixed}}$, and analyse the resulting collaboration game $(M, v_{\textsc{Mul},I,T})$.

Algorithm description. We consider $\textsc{Mul}$, a simple collaborative multi-agent bandit algorithm that each coalition shall independently use/execute. It is based on the essence of Howson et al. [2024], it is a meta-algorithm that uses a given single-agent bandit algorithm $\textsc{Sin}$ as a black-box decision-maker to play the multi-agent bandit problem instance.

Described in Algorithm 1, $\textsc{Mul}$ comprises of two conceptual components: First, all of the multiple agents interact with the original multi-agent bandit instance (iterated using real time t), observe rewards, and put the reward into a common reward buffer. Second, the $\textsc{Sin}$ plays a 'simulated' single-agent bandit instance (iterated using virtual time τ) by interacting with this reward buffer.

Specifically, at a given step τ, given the single-agent history $\overline{H}_{\tau-1}$, $\textsc{Sin}$ chooses an action $\overline{x}_\tau \in X$ to play (lines 2,9), seeks and obtains (if available) from the buffer $B_{\overline{x}_\tau}$ a reward $\overline{y}_\tau$ for this action (line 7). It appends this action-reward tuple $(\overline{x}_\tau, \overline{y}_\tau)$ to its single-agent history (line 8) and proceeds to the next step $\tau + 1$. Here, if the reward for action $\overline{x}_\tau$ is not available in the buffer (the condition in line 6 fails), then the first component is triggered, wherein all the agents play *this* action $\overline{x}_\tau$ on the original multi-agent bandit instance (say, in time step t), observe rewards (line 4), and put them into the reward buffer (line 5). After that, the second component resumes.

Algorithm 1 MUL, a multi-agent bandit meta-algorithm.

Input: A set of collaborating agents C, a fixed and finite action set X, a single-agent bandit algorithm SIN.

```
Initialize multi-agent reward buffers : B_x ← ∅, ∀x ∈ X,
  and single-agent history H̄_0 ← ∅, step τ ← 1.
Get initial action to play, x̄_τ ← SIN(H̄_{τ−1}, X).
for time step t ← 1, 2, ..., T do
    Every agent a ∈ C plays same action x_{a,t} ← x̄_τ and
      observes reward y_{a,t}.
    Replenish buffer B_{x̄_τ} ← {y_{a,t} : a ∈ C} with all
      agents' rewards.
    while B_{x̄_τ} ≠ ∅ do
        Remove arbitrarily a reward ȳ_τ from B_{x̄_τ}.
        Append to history H̄_{τ+1} ← H̄_τ ∪ {(x̄_τ, ȳ_τ)},
          move to next step τ ← τ + 1.
        Get next action to play, x̄_τ ← SIN(H̄_{τ−1}, X).
    end while
end for
```

On the choice of the single-agent black-box algorithm SIN to be used, we do not mandate any specific algorithm; instead, we only introduce a mild assumption on the regret behavior (in Assumption 1) it needs to have. Specifically, we assume that the algorithm improves over time: its expected instantaneous regret decreases as time t grows. However, it cannot decrease too rapidly: the cumulative regret curve cannot converge faster than logarithmically at any time.

To formally state the assumption, we introduce additional notation. For brevity, let $R(t) := R_a(\text{SIN}, I, t)$ denote the cumulative regret at time t.

We define two quantities that capture the temporal behaviour of this regret trajectory.

First, for $h > 0$, define the discrete first derivative $R'(t,h) := {}^1/_h\,(R(t+h) - R(t))$, which represents the average regret accumulated over the interval $(t, t+h]$. Second, for $g > 0$, define the discrete second derivative $R''(t,g,h) := {}^1/_g\,(R'(t+g,h) - R'(t,h))$ that measures the rate of change of this average regret. These quantities serve as discrete analogues of the first and second derivatives of the cumulative regret, respectively.

Assumption 1. *[Regret rate achievable] When run on any problem instance $I \in \mathcal{I}^{fixed}$, the expected regret of* SIN *obeys the following:*

1. ***Strict concavity.*** *The second derivative of the regret is strictly negative at all times t, i.e.,*

$$R''(t,g,h) \le -\upsilon_t, \tag{3}$$

for some strictly positive sequence of $\upsilon_t > 0$, for all t, g, h.

2. ***Logarithmic limitation.*** *The second derivative of the regret is bounded from below by that of a logarithmic curve at all times t, i.e.,*

$$R''(t,g,h) \ge -ct^{-2+\varepsilon}, \tag{4}$$

for some arbitrarily small constant $\varepsilon > 0$, for all t, g, h.

We discuss in Section B.1 how these assumptions about concavity of the regret and logarithmic learning limitation are very natural and arise from well known theoretical results and empirical observations that apply to several learning algorithms.

Next, we state in Theorem 1 that TU games induced by such algorithms on problem instances with fixed action sets are *convex*; that is, the marginal contribution that any agent brings to a coalition's value does not decrease as the coalition grows as in Equation (5). For completeness, we refer the reader to Section A for the precise definitions of cooperative game theoretic concepts (such as convex games, balanced games, core of a game, Shapley value axioms etc.) used in the remainder of the paper.

Theorem 1. *When the meta-algorithm* MUL *is run on any problem instance $I \in \mathcal{I}^{fixed}$ for a sufficiently large time horizon T, then, if the black-box single-agent algorithm* SIN *used obeys Assumption 1, the resultant collaboration game $(M, v_{\text{MUL},I,T})$ is convex.*

Deferring the formal proof to Section C, we give a proof sketch here.

Proof sketch. The result is shown in two steps.

Step 1: Relating multi-agent regret to single-agent regret. Fix a coalition C of size m. In Lemma 1, we show that the total regret incurred by the agents in C under MUL over horizon T is tightly controlled on both sides by the regret of the underlying single-agent algorithm SIN run for mT steps.

Lemma 1. *For any time-horizon T and problem instance $I \in \mathcal{I}^{fixed}$, for any agent $a \in C$,*

$$\begin{aligned} R_a(\text{SIN}, I, mT) - mK &\le \sum_{a \in C} R_a^C(\text{MUL}, I, T) \\ &\le R_a(\text{SIN}, I, mT) + mK, \end{aligned}$$

where $m = |C|$ is the number of agents in coalition C, and $K = |X|$ is the size of the action set X.

The key feature of this bound is that the discrepancy between the multi-agent regret and the single-agent regret is an additive term of order mK, which is independent of the time horizon T. Thus, asymptotically in T, the coalition regret behaves like the regret of a single agent run for mT rounds.

Step 2: Supermodularity of the value function. To establish convexity of the game, it suffices to show supermodularity of the value function: for all coalitions $S \subseteq Q \subseteq M$ and any agent $a \notin Q$,

$$v(S \cup \{a\}) - v(S) \;\leq\; v(Q \cup \{a\}) - v(Q). \qquad (5)$$

By definition of our value function (Equation (2)), the above requires comparison of regret quantities across four different coalitions. Using Lemma 1, we re-express the above inequality in terms of single-agent regret quantities that only differ in terms of time horizon for which they are run:

$$\begin{aligned} &R_a(\textsc{Sin}, I, qT{+}T) - R_a(\textsc{Sin}, I, qT) \\ &\leq R_a(\textsc{Sin}, I, sT{+}T) - R_a(\textsc{Sin}, sT) - 4MK, \end{aligned}$$

where $q = |Q|\,, s = |S|$ are the sizes of the coalitions. We complete the proof by showing that the above inequality is satisfied when Sin adheres to Assumption 1. □

The convexity of the collaboration game shown by the Theorem immediately leads to the following corollary:

Corollary 1. *Under the assumptions of Theorem 1, the collaboration game $(M, v_{\textsc{Mul},I,T})$ has a non-empty core. Moreover, the Shapley value of the game lies in the core.*

This is based on standard results about convex games (recapped in Section A.1.1). As the core is non-empty, we have that the grand coalition, i.e., the coalition of the set of all agents M, is stable.

4 HETEROGENEOUS ACTION SETS

In this section, we consider the more general setting of problem instances $I \in \mathcal{I}$ with heterogeneous action sets (non-identical agents). Without postulating specific algorithms, we prescribe some assumptions on the behaviour and performance that a multi-agent bandit algorithm Alg needs to satisfy for our results to hold.

First, we assume the algorithm shall benefit from other agents only through their action-reward tuples (samples) shared:

Assumption 2. *[Anonymized data consumption] The algorithm Alg, when run on a set of agents C, shall determine the action to be played by each agent $a \in C$ at time t as a function of*

1. *the agent's own past action-reward history, $P^a_{t-1} = ((x_{a,s}, y_{a,s}))_{s\in[t-1]}$,*
2. *the sequence of anonymized multisets/pool of past action–reward pairs generated by the other agents in the coalition*

$$P^{-a}_{t-1} = \left(\bigcup_{b\in C\setminus\{a\}} \{(x_{b,s}, y_{b,s})\} \right)_{s\in[t-1]},$$

where no agent identities are retained, and

3. *the set of actions $X_{a,t}$ of the agent at present.*

It can be seen that these data sequences from self-play and from other agents, P^a_t and P^{-a}_t, are functions of the coalition history H^C_t. Second, the (expected) regret of an agent does not increase when more agents join the coalition:

Assumption 3. *[The more (agents) the merrier] For any problem instance $I \in \mathcal{I}$, for any two coalitions $S \subseteq Q \subseteq M$, and any agent $a \in S$, it holds that*

$$R^Q_a(\textsc{Alg}, I, T) \;\leq\; R^S_a(\textsc{Alg}, I, T).$$

In Sections B.2 and B.3, we motivate why these assumptions are reasonable to expect from multi-agent bandit algorithms. We also provide an Explore-Then-Commit algorithm that satisfies these assumptions. We defer it to Section B.4 in the interest of space.

Next, we establish that these two assumptions are sufficient for the collaboration game to have several desirable properties.

Theorem 2. *Consider the collaboration game $(M, v_{\textsc{Alg},I,T})$ induced by any problem instance $I \in \mathcal{I}$. If the bandit algorithm Alg satisfies Assumption 3, then the grand coalition M is* stable*; equivalently, the game has a non-empty core.*

Proof. By the Bondareva-Shapley theorem, the core is non-empty if and only if the game is *balanced*. (Recapped as Theorem 4 in Section A). We shall show our result by showing that our collaboration game is balanced. For brevity, we write v and R^C_a to denote $v_{\textsc{Alg},I,T}$ and $R^C_a(\textsc{Alg}, I, T)$, respectively, for any coalition $C \subseteq M$.

For any balancing mapping w (i.e., $\sum_{S\ni a} w(S) = 1$ for every $a \in M$; see Definition 5), we have

$$\begin{aligned} \sum_{S\subseteq M: S\neq\emptyset} w(S)v(S) \;&=\; \sum_{S\subseteq M: S\neq\emptyset} w(S)\left(-\sum_{a\in S} R^S_a\right) \\ = -\sum_{a\in M}\sum_{S\subseteq M: a\in S} w(S)R^S_a \;&\overset{\text{(a)}}{\leq}\; -\sum_{a\in M}\sum_{S\subseteq M: a\in S} w(S)R^M_a \\ \overset{\text{(b)}}{=} -\sum_{a\in M} R^M_a &= v(M). \end{aligned} \qquad (6)$$

Here, (a) uses $R_a^S \geq R_a^M$ due to Assumption 3, and (b) uses the fact that w is a balancing mapping. Equation (6) shows that the game (M, v) is balanced, and thus it has a non-empty core. □

We have established that even when the action sets are arbitrarily heterogeneous (agents are non-identical), the grand coalition is *stable*, i.e., every agent shall prefer to collaborate together as the set of all agents (the grand coalition). This stability, however, holds only when agents receive a payout/allocation that lies within this non-empty core. In the case of fixed (identical) action sets, we established that the collaboration game is convex (Theorem 1), ensuring that the core contains the Shapley value (Corollary 1), which can serve as both a stable and fair allocation. For heterogeneous agents, by contrast, it remains unclear whether the collaboration game is convex and whether the Shapley value lies in the core. Consequently, the natural question arises: what allocation rule can guarantee both stability and equity/fairness in this more general setting?

4.1 On a Fair Payout Profile

In this subsection, we inquire if there are specific point solutions (allocation profiles) that obey desireable properties of fairness—such as efficiency, dummy-player, symmetry, and linearity—as in the axioms of the Shapley value.

'Grand coalition regret' allocation. For the collaboration game $(M, v_{\text{ALG},I,T})$, consider the allocation profile

$$p = \left(p_a = -R_a^M(\text{ALG}, I, T)\right)_{a \in M}, \tag{7}$$

where each agent is given a payout that equals the negative regret he shall incur as a part of the grand coalition from the said bandit scenario. We investigate what coalitions shall form under this payout structure.

Remark 1 (Practicality)**.** *Before the coalitions of agents form, it might appear impracticable to offer a payout p_a to agent $a \in M$ that depends on the regret in one specific coalition (the grand coalition M) as in Equation (7). However, this is not the case. We shall go on to show that under the promise of this payout, all agents shall indeed form the grand coalition. Thus, payout p is not counterfactual and is a realisable quantity.*

We find out that this payout profile satisfies three of the four Shapley axioms—efficiency, dummy-player, and symmetry—and it also belongs to the core of this game:

Theorem 3. *For any bandit instance $I \in \mathcal{I}$, if* ALG *satisfies Assumptions 2 and 3, then, the allocation $p = \left(p_a = -R_a^M(\text{ALG}, I, T)\right)_{a \in M}$ obeys the axioms of efficiency, dummy-player, and symmetry, and also belongs to the core of the collaboration game $(M, v_{\text{ALG},I,T})$.*

Deferring the formal proof to Section D, we provide a proof sketch here. For brevity, we write v and R_a^C to denote $v_{\text{ALG},I,T}$ and $R_a^C(\text{ALG}, I, T)$, respectively, for any coalition $C \subseteq M$.

Proof sketch. First, the payout satisfies the efficiency axiom by design; the value of the grand coalition $-\sum_{a \in M} R_a^M$ is exhaustively divided among the agents. Building upon the efficiency, we have for any coalition $S \subseteq M$ that

$$\sum_{a \in S} p_a = -\sum_{a \in S} R_a^M \overset{\text{(a)}}{\geq} -\sum_{a \in S} R_a^S = v(S) \tag{8}$$

where (a) is due to Assumption 3. This shows that no coalition 'blocks' the payout p and that it belongs to the core. The dummy-player axiom can also be similarly shown by building from the definition of a dummy player.

The symmetry axiom mandates that the payout of an agent—in our context, the regret of an agent a in the grand coalition—doesn't change when all the agents are relabeled. We shall rely on Assumption 2 and argue that the learning algorithm removes all the agent-specific information during the union operation to pool the data, and it is this pool of anonymized data that is used to determine the actions to play. Conditioned on this pool, the action choice only depends on the agent's action set and not the agent's identity. As a result, the distribution of bandit play trajectories remains unchanged under relabeling of agents, and thus the regret, and by extension, the payout $p_a = -R_a^M$ remains unchanged under relabeling of agents. □

While we have shown that the payout p obeys three of four Shapley axioms, we believe it cannot satisfy the final axiom in general. In Section D.3, we give some thoughts on why this payout p cannot satisfy the final Shapley axiom of Linearity.

Comparison with actual Shapley value. Achieving a payout profile p^* that coincides exactly with the Shapley value is often impractical for two key reasons. First, computing the Shapley value has exponential complexity in the number of agents, making it infeasible beyond small-scale settings. Second, even for a modest number of agents, its computation requires access to each agent's regret under *all* possible coalitions in order to evaluate v in Equation (11). These regret quantities are counterfactual and generally not directly observable.

In that context, Theorem 3 shows, perhaps surprisingly, that the payout profile p satisfies three of the four Shapley axioms without requiring any explicit computation or any redistribution of regret at the end of bandit play. In particular, the allocation $p_a = -R_a^M$ emerges naturally as the payoff when the grand coalition forms, i.e., the regret incurred by agent a when participating in the grand coalition.

5 NUMERICAL SIMULATIONS

We showed in Theorem 3 that under some assumptions, the payout structure $p = (p_a = -R_a^M(\textsc{Alg}, I, T))_{a \in M}$ obeys Shapley's axioms of dummy-player, symmetry, and efficiency, but may not obey linearity. We run numerical simulations to check how much the payout of an agent and his Shapley value align empirically on experiments using the MovieLens dataset below, and present some results on some synthetic instances in Section E.2.

Bandit environment setup. We consider the MovieLens-100k dataset [Harper and Konstan, 2015], motivated by a real-world movie recommendation platform. We create multiple bandit instances $\mathcal{I}' = \{I_{\text{gen}}, I_{\text{age}}, I_{\text{geo}}, I_{\text{occ}}\}$ by creating multiple sets of agents by dividing the set of users along the lines of attributes such as gender, age group, geographic location, and occupation. For each bandit instance, characterized by the attribute, an agent corresponds to the set of users who have a specific value for this attribute.

We consider two algorithms: (i) LinUcb-M(Multi-agent LinUCB)—all agents run an independent copy of the LinUCB algorithm [Abbasi-Yadkori et al., 2011] with all data available so far in the coalition; (ii) Greedy—all agents choose actions myopically so as to maximize their instantaneous expected reward w.r.t. the current parameter estimate (by Ordinary Least Squares) using all data available so far in the coalition.

We experimentally simulate the multi-agent bandit run for all four instances $\mathcal{I}'$ with both the algorithms. Implementation details are given in Section E.1. We present the results for $I_{\text{geo}}, I_{\text{occ}}$ in Figure 1, and defer the other plots to Section E.3.

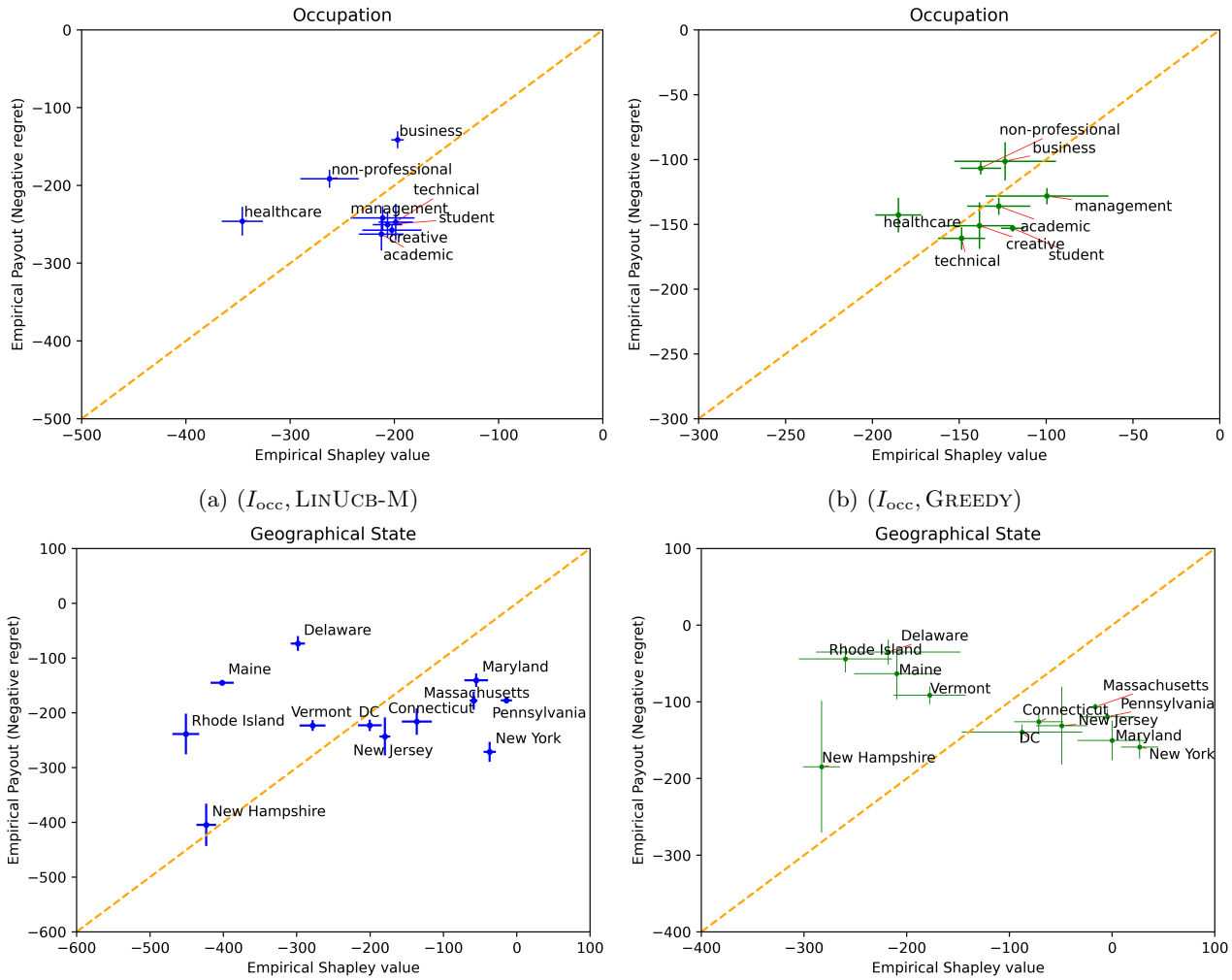
5.1 Experiment Outcomes.

In each bandit scenario, with say M agents in the instance, we run the multi-agent bandit algorithm for all $2^M - 1$ possible coalitions to get the regret of every agent in every coalition. And then, we explicitly compute the empirical Shapley value of the agents $\widehat{\phi} = (\widehat{\phi}_a)_{a \in M}$ from all the coalitional regrets (see Equation (11)). And we scatter-plot for every agent $a \in M$ his empirical payout $\widehat{p}_a$ (in y-axis), which is the negative of the regret from the grand coalition as in Equation (7), against his empirical Shapley value $\widehat{\phi}_a$ (in x-axis). The results are discussed below.

Payouts vs Shapley value. We observe diverse outcomes across the problem instances. In the I_{occ} instance, the payouts and Shapley value are reasonably close to each other (close to the orange identity line) for almost all agents, with both the algorithms (Figures 1a and 1b). This indicates that the regret incurred by an agent is 'fair' (in a Shapley sense) and is a commensurate compensation for the value he brings to the other agents. Further, the Shapley values and payouts show a positive correlation across the agents, more prominently so with the greedy algorithm.

For the I_{geo} instance, for some agents, the payouts and Shapley values are close to each other, but we observe a greater disparity in them for some agents (Figures 1c and 1d). The agents far above (or to the left of) the orange identity line have much higher payout compared to their Shapley value. It is interpreted that these agents benefit more from the other agents than they contribute to them. Correspondingly, the agents far from the identity line on the bottom-right side contribute to the other agents much more than they benefit from other agents. This mismatch highlights that the commonly used reward structure based on user engagement/satisfaction in recommender platforms does not equitably compensate some agents for their contributions to the platform, which are predominantly users from New York and Pennsylvania with LinUcb-M and New York and Maryland with Greedy, in this example. Additionally, however, we recall that these payouts are *not* the actual Shapley values (more discussion on this in Section D.3) and in general are not expected to equal the empirical Shapley values either.

Usefulness of Greedy over LinUCB. Another interesting outcome we see is that the greedy algorithm performs better in minimizing regret than the much acclaimed LinUCB as seen in both I_{geo} and I_{occ} instances. Conventionally, it is known that the delicate exploration-exploitation trade-off handled by LinUCB gives optimal regret guarantees, whereas greedy exploration in general is known to suffer worse regret. However, the agent actions (even greedily chosen ones among the available options) can be diverse enough in their vector directions in $\mathbb{R}^d$ due to the inherent heterogeneity of the agent action sets both across agents and across time. This can result in non-deliberate/inadvertent exploration of different dimensions even when the agent is greedily trying to play actions in a specific direction that it currently thinks is optimal. This result backs up a recent line of work



(a) $(I_{\text{occ}}, \textsc{LinUcb-M})$

(b) $(I_{\text{occ}}, \textsc{Greedy})$

(c) $(I_{\text{geo}}, \textsc{LinUcb-M})$

(d) $(I_{\text{geo}}, \textsc{Greedy})$

Figure 1: **Movielens experiments**

[Kannan et al., 2018, Raghavan et al., 2018] where it is shown that with sufficient heterogeneity in action sets, the greedy algorithm surprisingly achieves non-trivial regret bounds. Further, our experiments demonstrate that the greedy algorithm not only reduces an agent's own regret but also more effectively reduces the regrets of other agents compared to LinUCB. This is reflected in the higher Shapley values achieved by agents following the greedy algorithm in both instances, I_{geo} and I_{occ}.

6 CONCLUSION

In this paper, motivated by creator incentives in Recommender Systems, we considered the setting of multi-agent bandits with both fixed action set and heterogeneous action sets, and investigated the properties of the ensuing TU (transferable utility) coalition games. We explored different bandit algorithm properties, under which we showed that the game with fixed action set is provably convex and thus has a non-empty core containing the Shapley value; and the game with heterogeneous action sets has a non-empty core but may not contain the Shapley value. Further, we proposed a simple payout profile—where each agent's payout is his (negative) regret in the grand coalition—and we established this payout has desireable properties such as belonging to the core, and obeying Shapley axioms of efficiency, dummy-player, and symmetry. A key contribution of our work is to show that these mild and natural assumptions are sufficient to guarantee such strong cooperative-game properties for such a simple payout. An interesting research direction is to investigate whether more sophisticated payout schemes—particularly those that allow transfers of reward or utility at the end of bandit play, a setting well aligned with recommender systems—can yield alternative notions of fairness.

Acknowledgements

Rama Krishnamurthy was supported by the Meta-AI mentorship program from Meta Platforms Inc. Arpit Agarwal was partially supported by an early career research grant awarded by ANRF.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Guy Aridor, Yishay Mansour, Aleksandrs Slivkins, and Zhiwei Steven Wu. Competing Bandits: The Perils of Exploration Under Competition, October 2024. URL http://arxiv.org/abs/2007.10144. arXiv:2007.10144 [cs].

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Jackie Baek and Vivek Farias. Fair exploration via axiomatic bargaining. *Advances in Neural Information Processing Systems*, 34:22034–22045, 2021.

Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Exploiting the natural exploration in contextual bandits. *arXiv preprint arXiv:1704.09011*, 2017.

Omer Ben-Porat and Moshe Tennenholtz. A game-theoretic approach to recommendation systems with strategic content providers. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 1118–1128, Red Hook, NY, USA, December 2018. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/3326943.3327046.

Patrick Bolton and Christopher Harris. Strategic Experimentation. *Econometrica*, 67(2):349–374, 1999. ISSN 1468-0262. doi: 10.1111/1468-0262.00022. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-0262.00022.

Olga N Bondareva. Some applications of linear programming methods to the theory of cooperative games. *Problemy Kibernet*, 10:119, 1963.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24, 2011.

Michela Chessa and Patrick Loiseau. A cooperative game-theoretic approach to quantify the value of personal data in networks. In *Proceedings of the 12th workshop on the Economics of Networks, Systems and Computation*, pages 1–1, 2017.

Ekaterina Fedorova, Madeline Celi Kitch, and Chara Podimata. Altruistic Collective Action in Recommender Systems. November 2025. URL https://openreview.net/forum?id=wKSX4rqRSg.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376. JMLR Workshop and Conference Proceedings, 2011.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Benjamin Howson, Sarah Filippi, and Ciara Pike-Burke. Quack: A multipurpose queuing algorithm for cooperative k-armed bandits. *arXiv preprint arXiv:2410.23867*, 2024.

Jiri Hron, Karl Krauth, Michael Jordan, Niki Kilbertus, and Sarah Dean. Modeling content creator incentives on algorithm-curated platforms. In *The Eleventh International Conference on Learning Representations*, 2023.

Meena Jagadeesan, Nikhil Garg, and Jacob Steinhardt. Supply-side equilibria in recommender systems. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 14597–14608, Red Hook, NY, USA, December 2023. Curran Associates Inc.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Sampath Kannan, Jamie H Morgenstern, Aaron Roth, Bo Waggoner, and Zhiwei Steven Wu. A smoothed

analysis of the greedy algorithm for the linear contextual bandit problem. *Advances in neural information processing systems*, 31, 2018.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*, pages 199–213. Springer, 2012.

Jon Kleinberg, Christos H Papadimitriou, and Prabhakar Raghavan. On the value of private information. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 249–257, 2001.

Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1):141, 2022.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

Irwin Mann and Lloyd S Shapley. *Values of large games, IV: Evaluating the electoral college by Montecarlo techniques*. Rand Corporation, 1960.

Christopher Musco and R Teal Witter. Provably accurate shapley value estimation via leverage score sampling. *arXiv preprint arXiv:2410.01917*, 2024.

Ramasuri Narayanam and Yadati Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE transactions on automation science and engineering*, 8(1):130–147, 2010.

Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

Franco P Preparata and Michael I Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 2012.

Kun Qian and Sanjay Jain. Digital content creation: An analysis of the impact of recommendation systems. *Management Science*, 70(12):8668–8684, 2024.

Manish Raghavan, Aleksandrs Slivkins, Jennifer Vaughan Wortman, and Zhiwei Steven Wu. The externalities of exploration and how data diversity helps exploitation. In *Conference on Learning Theory*, pages 1724–1738. PMLR, 2018.

K Ramakrishnan, Arpit Agarwal, Lakshminarayanan Subramanian, and Maximilian Nickel. Collaborative learning under strategic behavior: Mechanisms for eliciting feedback in principal-agent bandit games. In *Agentic Markets Workshop at ICML 2024*, 2024.

Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6 (1):4–22, 1985.

Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Lloyd S Shapley. On balanced sets and cores. *Naval research logistics quarterly*, 14(4):453–460, 1967.

Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1:11–26, 1971.

Lloyd S Shapley et al. A value for n-person games. 1953.

Xuchuang Wang and Lin Yang. Achieving near-optimal individual regret low communications in multi-agent bandits. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Xuchuang Wang, Lin Yang, Yu-Zhen Janice Chen, Xutong Liu, Mohammad Hajiesmaili, Don Towsley, and John CS Lui. Exploration for free: how does reward heterogeneity improve regret in cooperative multi-agent bandits? In *Uncertainty in Artificial Intelligence*, pages 2192–2202. PMLR, 2023.

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*, 2019.

Eva Yiwei Wu, Emily Pedersen, and Niloufar Salehi. Agent, Gatekeeper, Drug Dealer: How Content Creators Craft Algorithmic Personas. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW): 1–27, November 2019. ISSN 2573-0142. doi: 10.1145/3359321. URL https://dl.acm.org/doi/10.1145/3359321.

Lin Yang, Xuchuang Wang, Mohammad Hajiesmaili, Lijun Zhang, John CS Lui, and Don Towsley. Cooperative multi-agent bandits: Distributed algorithms with optimal individual regret and communication costs. In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Methods Workshop*, 2023.

Fan Yao, Chuanhao Li, Denis Nekipelov, Hongning Wang, and Haifeng Xu. How Bad is Top-K Recommendation under Competing Content Creators? In *Proceedings of the 40th International Conference on Machine Learning*, pages 39674–39701. PMLR, July 2023. URL https://proceedings.mlr.press/v202/yao23b.html. ISSN: 2640-3498.

Yaolong Yu, Fan Yao, and Sinno Jialin Pan. Beyond Self-Interest: How Group Strategies Reshape Content Creation in Recommendation Platforms? June 2025. URL https://openreview.net/forum?id=q0JaH6Ukqb&noteId=1tb6U3J4pX.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38, 2019.

Banghua Zhu, Sai Praneeth Karimireddy, Jiantao Jiao, and Michael I Jordan. Online learning in a creator economy. *arXiv preprint arXiv:2305.11381*, 2023a.

Zheqing Zhu, Rodrigo de Salvo Braz, Jalaj Bhandari, Daniel Jiang, Yi Wan, Yonathan Efroni, Liyuan Wang, Ruiyang Xu, Hongbo Guo, Alex Nikulkov, Dmytro Korenkevych, Ürün Dogan, Frank Cheng, Zheng Wu, and Wanqiao Xu. Pearl: A production-ready reinforcement learning agent. *CoRR*, abs/2312.03814, 2023b.

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.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes.]
   (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]
   (b) Complete proofs of all theoretical results. [Yes, proof sketches are provided in the main paper, and complete proofs are give in the Supplementary Material.]
   (c) Clear explanations of any assumptions. [Yes, there are specific discussions in the Appendix about each of the three assumptions.]

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). [No. Experimental setup is discussed in the Supplementary Material.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (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]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No, experiments are not compute intensive.]

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]
   (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]
   (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 Materials

Contents

A FUNDAMENTALS **14**

A.1 Game theoretic aspects . . . 14

A.1.1 Shapley Value . . . 15

A.2 Mathematical aspects . . . 16

B BANDIT ALGORITHM ASSUMPTIONS **17**

B.1 On Assumption 1 . . . 17

B.2 On Assumption 2 . . . 17

B.3 On Assumption 3 . . . 18

B.4 Algorithm that satisfies Assumptions 2 and 3 . . . 19

C MISSING PROOFS FROM SECTION 3 **23**

C.1 Proof of Theorem 1 . . . 23

C.2 Proof of Lemma 14 . . . 26

D MISSING PROOFS FROM SECTION 4 **28**

D.1 Proof of Theorem 3 . . . 28

D.2 Proof of Claim 20 . . . 29

D.3 On the Linearity axiom. . . . 30

E ADDITIONAL NUMERIC SIMULATIONS **31**

E.1 Notes on Implementation . . . 31

E.2 Synthetic Experiments . . . 32

E.3 MovieLens Experiments . . . 33

E.4 On satisfying Assumption 3 . . . 34

A FUNDAMENTALS

A.1 Game theoretic aspects

In this section, we recap some fundamental definitions and results from co-operative game theory that are referred to in the main paper.

Consider a Transferable Utility (TU) game (N, v), where N is the finite set of players, and $v : 2^N \mapsto \mathbb{R}$ is the characteristic or value function with $v(\emptyset) = 0$.

Definition 4 (Convexity of a TU game)**.** *A transferable utility game (N, v) is said to be* convex *if marginal contributions of agents are non-decreasing with coalition growth. That is, for all players $i \in N$, and for all coalitions C, D such that $C \subseteq D \subseteq N \setminus \{i\}$, it holds that*

$$v(D \cup \{i\}) - v(D) \geq v(C \cup \{i\}) - v(C). \tag{9}$$

Definition 5 (Balanced game)**.** *For a set of players N, a mapping $w : 2^N \mapsto [0, 1]$ is said to be* 'balancing' *if for every player $a \in N$, it holds that*

$$\sum_{S \subseteq N : a \in S} w(S) = 1.$$

A transferable utility game (N, v) is said to be balanced *if for every balancing mapping w it holds that*

$$\sum_{S \subseteq N : S \neq \emptyset} w(S) v(S) \leq v(N).$$

Definition 6 (Core of a TU game)**.** *For a transferable utility game (N, v) with $|N| = n$, its core is defined as the set of payout profiles that are feasible and cannot be improved upon by any coalition. Precisely,*

$$Core(N, v) := \left\{ p = (p_1, \ldots, p_n) \in \mathbb{R}^n : \sum_{i=1}^{n} p_i = v(N); \forall C \subseteq N, \sum_{i \in C} p_i \geq v(C) \right\}. \tag{10}$$

In general, the core of a game is not guaranteed to be non-empty. A necessary and sufficient for it is given next:

Theorem 4 ([Bondareva, 1963, Shapley, 1967])**.** *A transferable utility game (N, v) has a non-empty core if and only if it is balanced.*

A.1.1 Shapley Value

Shapley value is a point solution concept for distributing the value of a coalition among its members in a 'fair' way [Shapley et al., 1953, Roth, 1988].

Definition 7 (Shapley Value)**.** *For a TU game (N, v) with $n = |N|$ players, the Shapley value $\phi = (\phi_i)_{i \in [n]}$ is given by*

$$\phi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} \left(v(S \cup \{i\}) - v(S) \right). \tag{11}$$

The Shapley value is the unique solution that satisfies the following axioms.

Axiom 2 (Symmetry, Anonymity)**.** *Let $\pi : N \mapsto N$ be any (bijection) permutation of the set of players. Overload notation to write $\pi(C) := \{\pi(a) : a \in C\}$ for any coalition $C \subseteq N$. Consider a modified value function πv that is defined as $\pi v(\pi(C)) = v(C)$ for all $C \subseteq N$. Then, the axiom mandates that the Shapley values of the two games obey*

$$\phi_i(v) = \phi_{\pi(i)}(\pi v) \tag{12}$$

for all $i \in [N]$.

Axiom 3 (Carrier)**.** *A subset $C \subseteq N$ of agents is said to be a* 'carrier' *when $v(D) = v(C \cap D)$ for all $D \subseteq N$. Then, the axiom mandates that the cumulative Shapley values of carriers equals the value of grand coalition, i.e.,*

$$\sum_{a \in C} \phi_a(v) = v(N). \tag{13}$$

Axiom 4 (Linearity)**.** *Consider two different TU games (N, v_1) and (N, v_2). Define the combined game $(N, v_1 + v_2)$ such that its value function equals $[v_1 + v_2](C) = v_1(C) + v_2(C)$ for all coalitions $C \subseteq N$. Then, the axiom mandates that the Shapley values of these games shall obey*

$$\phi_i(v_1 + v_2) = \phi_i(v_1) + \phi_i(v_2), \tag{14}$$

for all players $i \in N$.

These Axioms 2 to 4 were used in the original paper (see Chapter 2 of Shapley [1967]) that introduced Shapley value. However, there are some restatements of some of these axioms as given below.

Axiom 5 (Symmetry, "equal treatment of equals"). *For any two players $i, j \in N$ such that $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \in N \setminus \{i, j\}$, their Shapley values $\phi_i(v) = \phi_j(v)$ are equal.*

Axiom 6 (Null player). *A player $i \in N$ is said to be a* 'dummy' *player (or a null player) if the player adds no value to any coalition beyond their individual value, i.e., $v(S \cup \{i\}) = v(S) + v(\{i\})$ for all $S \subseteq N$. The Shapley value of a dummy player is $\phi_i(v) = v(\{i\})$, his individual value.*

The Null player axiom is also popularly stated by additionally assuming $v(\{i\}) = 0$ in the definition of a dummy player.

Theorem 5 (Theorem 7 of Shapley [1971]). *The Shapley value of a convex game belongs to the core.*

The above Theorem also immediately implies that the core of a convex game is non-empty (and the game admits a stable grand coalition).

A.2 Mathematical aspects

Definition 8 (Convex set). *Let V be a Euclidean space. A set $C \subseteq V$ is said to be* convex *if for every $x, y \in C$ and $\lambda \in [0, 1]$, the element $\lambda x + (1 - \lambda)y \in C$.*

We now relax the notion of convexity to the following weaker notion:

Definition 9 (Star-shaped set). *Let V be a Euclidean space. A set $S \subseteq V$ is said to be* star-shaped *if there exists some point/element $x \in S$ such that for every $y \in S$, the line segment $\overline{xy}$ lies within S, i.e., $\forall \lambda \in [0, 1] : \lambda x + (1 - \lambda)y \in S$.*

The set of all such x is called the kernel *of set S.*

A convex set is also star-shaped, and it can be seen that the entire convex set is its own kernel.

These star-shaped sets in the two-dimensional Euclidean plane are known as star-shaped polygons and appear to be well-studied in the computational geometry literature (see e.g., the 1985 textbook of Preparata and Shamos [2012]).

Claim 7 (Kernel-centred Gaussian distributions over star-shaped polytopes). *Consider a star-shaped polytope S with θ^* in its kernel. Let $A \sim \mathcal{N}\left(\theta^*, V_A^{-1}\right)$ and $B \sim \mathcal{N}\left(\theta^*, V_B^{-1}\right)$ be Gaussian random variables with $V_B^{-1} \succeq V_A^{-1}$, i.e., A is more tightly centred around θ^* than B is, in every direction.*

Then, $\mathbb{P}\{A \in S\} \geq \mathbb{P}\{B \in S\}$.

Proof. Write f_A (and f_B) to be the p.d.f of $\mathcal{N}\left(\theta^*, V_A^{-1}\right)$ (sim. $\mathcal{N}\left(\theta^*, V_A^{-1}\right)$) in d dimensions. We show the Claim by comparing the corresponding probability integrals in the polar system. We start with the L.H.S.

$$
\begin{aligned}
\mathbb{P}\{A \in S\} &= \int_{s \in S} f_A(s).ds = \int_{s \in S} (2\pi)^{d/2} \det\left(V_A^{-1}\right)^{-1/2} \exp\left\{\frac{-1}{2}\|s - \theta^*\|_{V_A}\right\}.ds \\
&\overset{(a)}{=} \int_{\phi} \int_{r=0}^{r_\phi} (2\pi)^{d/2} \det(V_A)^{1/2} \exp\left\{\frac{-1}{2}\|r\|_{V_A}\right\}.dr.d\phi \\
&\overset{(b)}{\geq} \int_{\phi} \int_{r=0}^{r_\phi} (2\pi)^{d/2} \det(V_B)^{1/2} \exp\left\{\frac{-1}{2}\|r\|_{V_B}\right\}.dr.d\phi \\
&\overset{(c)}{=} \int_{s \in S} f_A(s).ds = \mathbb{P}\{B \in S\},
\end{aligned}
$$

where (b) is due to $V_A \succeq V_B$. Further, crucially, the conversion to and from the polar systems in (a) and (c) over a continuous $[0, r_\phi]$ is possible because S is star-shaped with θ^* in its kernel, i.e., in any direction ϕ, a ray originating from θ^* exits the polytope S at most once (and never enters again), and we call r_ϕ to be the distance of this exit point from θ^* (which is ∞ if it doesn't exit). □

B BANDIT ALGORITHM ASSUMPTIONS

In our work, we made a string of assumptions about the nature and performance of the bandit algorithms to be used. First, we made Assumption 1 about the black-box algorithm that our Algorithm 1 used for the Homogenous setting (Section 3). Second, we made Assumptions 2 and 3 about the behaviour of the multi-agent algorithm and showed that such an algorithm enjoys nice theoretical properties (Section 4).

In this section, we discuss those three assumptions and their practicality in Sections B.1 to B.3. And finally, in Section B.4 we give a multi-agent bandit algorithm for the heterogeneous setting that obeys the two assumptions assumed for the theoretical results.

B.1 On Assumption 1

Assumption 1. *[Regret rate achievable] When run on any problem instance $I \in \mathcal{I}^{fixed}$, the expected regret of* Sin *obeys the following:*

1. ***Strict concavity.*** *The second derivative of the regret is strictly negative at all times t, i.e.,*

$$R''(t, g, h) \leq -\upsilon_t, \tag{3}$$

for some strictly positive sequence of $\upsilon_t > 0$, for all t, g, h.

2. ***Logarithmic limitation.*** *The second derivative of the regret is bounded from below by that of a logarithmic curve at all times t, i.e.,*

$$R''(t, g, h) \geq -ct^{-2+\varepsilon}, \tag{4}$$

for some arbitrarily small constant $\varepsilon > 0$, for all t, g, h.

The Assumption constrains the nature of regret the single agent bandit algorithm Sin to be used shall incur. Namely, the change of instantaneous regret with time (or the second derivative of the cumulative regret) is upper bounded and lower bounded.

Upper Bound. Equation (3) implies that the total regret $R_{\cdot}(\text{Sin}, I, T)$ is strictly concave as a function of time horizon T, i.e., the instantaneous regret (or the first derivative of the cumulative regret) shall decrease with time. In other words, the algorithm Sin improves over time.

Basic bandit algorithms such as Explore-Then-Commit (also known as Explore-First) and ε-Greedy approaches are known to achieve a regret of $O(T^{2/3})$. More sophisticated algorithms, such as Successive Elimination, UCB1 [Auer et al., 2002], and Thompson Sampling [Kaufmann et al., 2012], all achieve $O(T^{1/2})$ regret rates in finite time. All these functional forms are strictly concave, albeit these bounds are worst-case in nature.

Further, there is ample empirical evidence to suggest that the assumption of concavity is natural. It can be seen that empirical cumulative regret (typically averaged over a few repetitions/runs) follows a strictly concave growth with time. For visual plots, we refer the reader to Russo et al. [2018], Garivier and Cappé [2011], Chapelle and Li [2011] that demonstrate the aforementioned nature of shape of regret curve.

Lower Bound. It is also known that any bandit algorithm cannot perform better than a certain level, i.e., the algorithm will have to incur some minimum rate of (expected) regret. In other words, the cumulative regret curve 'cannot flatten too soon' The seminal work of Robbins [1985] establishes an asymptotic lower bound of $\Omega(\log t)$ to the minimizable regret. And the second derivative of the members of the family of logarithmic curves is of the form $-ct^{-2}$ for some constant c which is used in Equation (4) with a small ε margin.

B.2 On Assumption 2

We recap the space of histories of agent arm play and observations. $H_{a,t} := (x_{a,s}, y_{a,s})_{s=1}^{t}$ denotes the single-agent history, i.e., the sequence of actions played $x_{a,s} \in X_{a,s}$ and rewards observed $y_{a,s} \in \mathbb{R}$, by agent a up to time t. Further, the coalition history $H_{t-1}^{C} := \{H_{b,t-1} : b \in C\}$ comprises of single-agent histories corresponding to all agents in the coalition C up to the time step $t-1$.

Let the space of all histories be defined as follows: $\mathcal{H}_{a,t} = \bigtimes_{s=1}^{t} (X_{a,s} \times \mathbb{R}) \subseteq \left(\mathbb{R}^d \times \mathbb{R}\right)^t$ be the family of single-agent histories of length t. And, let

$$\mathcal{H}_t^C \in \bigtimes_{a \in C} \bigtimes_{s=1}^{t} (X_{a,s} \times \mathbb{R}) \subseteq \left(\mathbb{R}^d \times \mathbb{R}\right)^{|C| \times t}$$

be the family of all coalition histories of length t. And our problem setting permits that a multi-agent algorithm Alg can take as input this coalition history to come up with actions to play for all the agents in the coalition. Precisely, for all $t \in [T]$, $\text{Alg}^t : \mathcal{H}_{t-1}^C \mapsto (\times_{a \in C} X_{a,t})$. [2]

However, we constrain the multi-agent algorithm to use a different quantity as follows:

Assumption 2. *[Anonymized data consumption] The algorithm* Alg*, when run on a set of agents C, shall determine the action to be played by each agent $a \in C$ at time t as a function of*

1. *the agent's own past action-reward history, $P_{t-1}^a = ((x_{a,s}, y_{a,s}))_{s \in [t-1]}$,*
2. *the sequence of anonymized multisets/pool of past action–reward pairs generated by the other agents in the coalition*

$$P_{t-1}^{-a} = \left(\bigcup_{b \in C \setminus \{a\}} \{(x_{b,s}, y_{b,s})\} \right)_{s \in [t-1]},$$

   where no agent identities are retained, and
3. *the set of actions $X_{a,t}$ of the agent at present.*

This Assumption states that the multi-agent algorithm shall not use coalition history $H_t^C \in \mathcal{H}_t^C$ directly, but shall instead use a further processed quantity P_t^C, which is the union of samples observed so far by all agents. It is easy to observe that this pool of samples P_t^C is a deterministic function of the coalition history H_t^C. By data processing inequality, this Assumption doesn't provide the algorithm with additional/improved information in making the choice of actions to play. On the contrary, information is lost. Specifically, the information about the identity of the agent and the time of play is lost in the union operation.

However, it can be argued that this loss of information doesn't affect the performance of the algorithm. This is squarely due to the i.i.d. nature of rewards of the multi-agent linear bandit setting, when conditioned on the action. When agent $a \in M$ at time $t \in [T]$ plays action $x_{a,t} = x \in X_{a,t}$, he observes a reward $y_{a,t} = \langle \theta^*, x \rangle + \eta_{a,t}$. Here, the sequence of additive noises $(\eta_{a,t})_{a \in M, t \in [T]}$ are independent and identically distributed across agents $a \in M$ and time $t \in [T]$. In other words, the reward depends only on the action played and does not depend on the agent who plays it or the time at which it is played. Thus, it is reasonable to lose this information about the identity of agent and time from the samples in the coalition history without impacting the strength of an algorithm.

Examples from literature. In fact, this is a common approach that several multi-agent algorithms use, satisfying this Assumption. For example, the *DisLinUCB* algorithm [Wang et al., 2019] makes all agents run an Upper Confidence Bound (UCB) based algorithm, wherein the parameter estimate is computed by using the aggregate 'design' matrix $V_{a,t} = \sum_{a \in M} \sum_{s=1}^{t} x_{a,s} x_{a,s}^\top$, the sum of outer product of all actions played, and the 'result' matrix $B_{a,t} = \sum_{a \in M} \sum_{s=1}^{t} x_{a,s} y_{a,s}$, the sum of product of action and reward pairs. Both these quantities do not make use of the identity of agents and time, that information is lost in the summation, and these quantities can be seen to be functions of the pool/union of samples P_t^M.

B.3 On Assumption 3

Assumption 3. *[The more (agents) the merrier] For any problem instance $I \in \mathcal{I}$, for any two coalitions $S \subseteq Q \subseteq M$, and any agent $a \in S$, it holds that*

$$R_a^Q(\text{Alg}, I, T) \;\leq\; R_a^S(\text{Alg}, I, T).$$

[2] If the algorithm is non-deterministic, the co-domain becomes a distribution over the joint action space, $\Delta\left(\times_{a \in C} X_{a,t}\right)$, but this distinction is not important to the point we make.

First, we give examples from the literature that argue that given 'enough heterogeneity', collaboration does lower the regret of the agents. Second, we point to some numerical simulations to verify that the assumption is satisfied in most cases in our MovieLens problem instance experiments.

Related Work - Heterogeneity helps implicitly explore. There is a line of work [Bastani et al., 2017, Kannan et al., 2018, Wang and Yang, 2023] which studies how heterogeneity in action sets inherently helps exploration and thus helps minimize regret. Specifically, Kannan et al. [2018] consider the stochastic linear bandit setting as ours, where the action sets can be set by an adversary, but is then perturbed component-wise by i.i.d Gaussian noise before being presented to the agent. That is, at time t, for any given arbitrary action set $X'_t = \{x'_{1,t}, \ldots, x'_{k,t}\}$ with actions in $\mathbb{R}^d$, they are perturbed with i.i.d. noise vectors $\eta_i \sim \mathcal{N}\left(0, \sigma^2 I_d\right)$ for $i \in [k]$ as follows: $X_t = \{x_{i,t} = x'_{i,t} + \eta_i\}_{i \in [k]}$. Thus, at every time t, the action set is made 'diverse' by perturbing it in a random direction in $\mathbb{R}^d$. More importantly, as this perturbation is independent over time, the action sets are also diverse across time. Under such a condition, the authors consider the greedy algorithm, where an action is played from $\arg\max_{x \in X_t} \hat{\theta}_t^\top x$ to myopically maximize current expected reward based on current parameter estimate $\hat{\theta}_t$. They surprisingly show that with high probability, this greedy algorithm attains a regret upper bound of $O\left(\sqrt{dT}/\sigma^2\right)$ where σ^2 is the variance of the component-wise perturbation added to the actions.

Comparing this to our setting, if an agent a has action sets that have good representation in the ambient space $\mathbb{R}^d$, and the other agents' action sets are not correlated with that of agent a, then, agent a benefits from more data as more agents are in the coalition.

Empirical validation. From our MovieLens experiments, we exhaustively check if the Assumption 3 holds. We observe that it holds for most of the agent-coalition pairs. We describe these in Section E.4.

B.4 Algorithm that satisfies Assumptions 2 and 3

In Theorems 2 and 3, we showed that any multi-agent bandit algorithm that satisfies Assumptions 2 and 3 enjoys desirable properties such as the grand coalition being stable, and the payout obeying all but one of the Shapley value axioms. In this section, we give an algorithm (Algorithm 2) that provably satisfies these assumptions (Claim 8 and Theorem 6).

Consider the algorithm M-ETC (Algorithm 2) based on the Explore-Then-Commit (or Explore-First) paradigm extended to accommodate multiple agents interacting with the bandit environment.

M-ETC runs in two stages—an exploration phase, followed by a commit phase. First, the exploration phase happens for a set duration of T' time steps. In each time step in it, the algorithm uses an exploration routine to come up with the determinant-maximizing action $x_{a,t}$ for each agent a to play,

$$x_{a,t} \leftarrow \underset{x \in X_{a,t}}{\arg\max} \det\left(I + V_{a,t-1} + xx^\top\right), \tag{15}$$

where $V_{a,t-1} := \sum_{s<t} x_{a,s} x_{a,s}^\top$. All ties are broken in some deterministic manner agnostic of the agent a. Notably, the exploration action shall depend on the actions recommended to (and played by) the agent so far, but shall be independent of the rewards observed by the agent a, or the actions and rewards of any other agent.

After T' time steps, with the actions and rewards of all agents in A, the algorithm computes an estimate of the linear parameter using Ordinary Least Squares (OLS),

$$\hat{\theta}_A \leftarrow V_A^{-1} B_A, \text{ where } V_A = \sum_{a \in A} \sum_{t \leq T'} x_{a,t} x_{a,t}^\top, \text{ and } B_A = \sum_{a \in A} \sum_{t \leq T'} x_{a,t} y_{a,t}. \tag{16}$$

Second, in the commit phase, each agent plays the action that maximizes his expected reward as if the estimate $\hat{\theta}_A$ is the true parameter value. In other words, the agents 'commit' to the estimate $\hat{\theta}_A$ for the rest of play.

Claim 8. M-ETC *obeys Assumption 2.*

Proof. We prove the claim by showing that at each time $t \in [T]$, the action to be played $x_{a,t}$ depends on the action-reward sequences from self-play or from other agents' play, P_{t-1}^a and P_{t-1}^{-a}.

Algorithm 2 M-ETC, an algorithm for multi-agent linear bandits.

Input : A set of collaborating agents A, time horizon T, action sets $X_{a,t}$ for all agents $a \in A$ and time steps $t \in [T]$, an exploration threshold $T' < T$.

for time step $t = 1, 2, \ldots, T'$ **do** ▷ Exploration phase.
  Each agent $a \in A$ plays action to maximize his exploration 'volume' as in Equation (15) and observes rewards $y_{a,t}$.
end for
Compute parameter estimate $\hat{\theta}_A$ using all agents' statistics until T' as in Equation (16).
for time step $t = T' + 1, \ldots, T$ **do** ▷ Commit phase.
  Each agent $a \in A$ plays action $x_{a,t} \leftarrow \arg\max_{x \in X_{a,t}} x^\top \hat{\theta}_A$.
end for

First, in the exploration phase, we have that the action played

$$\begin{aligned} x_{a,t} &= \arg\max_{x \in X_{a,t}} \det\left(I + V_a^{t-1} + xx^\top\right) = \arg\max_{x \in X_{a,t}} \det\left(I + xx^\top + \sum_{s=1}^{t-1} x_{a,s} x_{a,s}^\top\right) \\ &= \arg\max_{x \in X_{a,t}} \det\left(I + xx^\top + \sum_{(x',y') \in P_{t-1}^a} x' x'^\top\right). \end{aligned} \tag{17}$$

Second, in the commit phase, the action played at any time $t > T'$ depends on the estimate $\hat{\theta}_A$ which in turn is

$$\begin{aligned} \hat{\theta}_A &= V_A^{-1} B_A = \left(\sum_{b \in A} \sum_{s \in [T']} x_{b,s} x_{b,s}^\top\right)^{-1} \left(\sum_{b \in A} \sum_{s \in [T']} x_{b,s} y_{b,s}\right) \\ &= \left(\sum_{(x,y) \in P_{T'}^a} xx^\top + \sum_{(x,y) \in P_{T'}^{-a}}\right)^{-1} \left(\sum_{(x,y) \in P_{T'}^A} xy\right), \end{aligned} \tag{18}$$

where $P_{T'}^a$ (sim. $P_{T'}^{-a}$) is a prefix (a function) of P_{t-1}^a (sim. P_{t-1}^{-a}) as $t > T'$.

From Equations (17) and (18), it is seen that for any agent $a \in A$, time $t \in [T]$, the action played $x_{a,t}$ is a function of the $X_{a,t}$, P_{t-1}^a, and P_{t-1}^{-a}, satisfying the Assumption. □

Theorem 6. M-ETC *obeys Assumption 3. That is, for any problem instance I, two coalitions $S \subseteq Q \subseteq M$, and any agent $a \in S$, it holds that $R_a^Q(\text{M-ETC}, I, T) \leq R_a^S(\text{M-ETC}, I, T)$.*

Proof. Write the regret of agent $a \in A$ (for any generic coalition $A \subseteq M$) to be the sum of regret in the exploration phase and the regret in the commit phase.

$$R_a^A(I, \text{M-ETC}, T) = R_a^A(I, \text{M-ETC}, T') + R_a^A(I, \text{M-ETC}, [T'+1, T]). \tag{19}$$

Regret in Exploration phase. For every agent $a \in M$, the exploration routine in Equation (15) chooses actions $x_{a,t}$ independent of the presence of the other agents. Thus, the actions played and thus the regret incurred by agent a as a part of any coalition is identical. That is, for coalitions $S \subseteq Q$, and any agent $a \in S$, $R_a^Q(I, \text{M-ETC}, T') = R_a^S(I, \text{M-ETC}, T')$.

Regret in Commit phase. To show the Theorem, what remains to be shown is that the regrets in the commit phase obey

$$R_a^Q(I, \text{M-ETC}, [T'+1, T]) \leq R_a^S(I, \text{M-ETC}, [T'+1, T]) \tag{20}$$

We shall show this by arguing that the estimate $\hat{\theta}_Q$ is 'more accurate' than $\hat{\theta}_S$ and with these estimates, the probability of playing the sub-optimal actions in the commit phase is lesser when the agent is part of the bigger coalition Q (Lemma 9).

We set up some notation. W.l.o.g., number the actions in $X_{a,t}$ by decreasing order of optimality as x_1 (optimal), $x_2, \ldots, x_k$, where $k = |X_{a,t}|$. Let $\mathbb{P}^Q\{\cdot\}$ and $\mathbb{P}^S\{\cdot\}$ be the probability measures of the algorithmic trajectory (or history) induced by running M-ETC with coalitions Q and S respectively.

Lemma 9 (Sub-optimal action play probabilities)**.** *At any time $t \in [T'+1, T]$, for agent $a \in S \subseteq Q$ and for any $1 \le i \le k$, the probability of choosing action at least as inferior as i when* M-ETC *is run with bigger coalition Q is at most the corresponding probability when run with smaller coalition S. That is,*

$$\mathbb{P}^Q\{x_{a,t} \in \{x_i, x_{i+1}, \ldots, x_k\}\} \le \mathbb{P}^S\{x_{a,t} \in \{x_i, x_{i+1}, \ldots, x_k\}\}.$$

Proof. Towards proving this, we develop some constructs using a generic coalition A and shall later instantiate them using coalitions S and Q. At any time $t \in [T'+1, T]$, in a generic coalition A, introduce the following collection of sets $\{C_{A,x,t}\}_{x \in X_{a,t}}$, where we define

$$C_{A,x,t} = \{\theta' \in \mathbb{R}^d : x = \arg\max_{x \in X_{a,t}} x^\top \theta'\} \tag{21}$$

to be the set of values that estimate $\hat{\theta}_A$ should belong to for action x to be played by agent a at time t by M-ETC. Precisely,

$$\hat{\theta}_A \in C_{A,x,t} \iff x_{a,t} = x, \tag{22}$$

and we shall study the probability that a certain action x is played by studying the probability that the computed estimate $\hat{\theta}_A$ lies in set $C_{A,x,t}$ corresponding to action x.

Claim 10 (Convex cone)**.** *For any action $x \in X_{a,t}$ of agent a at time t, the set $C_{A,x,t}$ is a convex cone.*

Proof. The Claim follows from the linearity of the $x^\top \theta'$ dot-product function being optimized.

First, it is a cone since for all $\theta' \in C_{A,x,t}$, we have $c\theta' \in C_{A,x,t}$ for any positive constant $c > 0$, as

$$x^\top \theta' > z^\top \theta' \iff x^\top (c\theta') > z^\top (c\theta')$$

for all other actions $z \neq x$.

Second, it is convex since for any $\theta', \theta'' \in C_{A,x,t}$, we have that $c_1\theta' + c_2\theta'' \in C_{A,x,t}$ for positive constants $c_1, c_2 \ge 0$, as

$$\begin{aligned}(x^\top \theta' > z^\top \theta') \wedge (x^\top \theta'' > z) &\implies (c_1.x^\top \theta' + c_2.x^\top \theta'' > c_1.z^\top \theta' + c_2.z^\top \theta'') \\ &\implies (x^\top (c_1 + c_2)\theta' > z^\top (c_1 + c_2)\theta'),\end{aligned}$$

for all other actions $z \neq x$. □

Geometrically, the collection $\{C_{A,x,t}\}_{x \in X_{a,t}}$ partitions $\mathbb{R}^d$ into pie slices with apex/centre at origin.

Next, we claim that the estimate from the bigger coalition $\hat{\theta}_Q$ has a lesser variance than that of the smaller coalition $\hat{\theta}_S$ in all directions:

Claim 11 (Bigger coalition $\Rightarrow$ Tighter estimate)**.** *The estimates obey the Gaussian distributions $\hat{\theta}_S \sim \mathcal{N}\left(\theta^*, V_S^{-1}\right)$ and $\hat{\theta}_Q \sim \mathcal{N}\left(\theta^*, V_Q^{-1}\right)$, with $V_S^{-1} \succeq V_Q^{-1}$.*

Proof. Under the two coalitions S and Q, the OLS estimates computed (Equation (16)) obey the gaussian distributions $\hat{\theta}_S \sim \mathcal{N}\left(\theta^*, V_S^{-1}\right)$ and $\hat{\theta}_Q \sim \mathcal{N}\left(\theta^*, V_Q^{-1}\right)$. And since $S \subseteq Q$, we have that from Equation (16) that $V_Q \succeq V_S$ and thus, the covariances $V_S^{-1} \succeq V_Q^{-1}$. □

To show the Lemma, as observed in Equation (22), we shall show that, for all $1 \le i \le k$, $\hat{\theta}_Q$ falls in the corresponding union of sets with higher probability than $\hat{\theta}_S$ does in the following claim:

Claim 12. $\mathbb{P}^Q\left\{\hat{\theta}_Q \in \bigcup_{j=1}^{i} C_{A,x_j,t}\right\} \geq \mathbb{P}^S\left\{\hat{\theta}_S \in \bigcup_{j=1}^{i} C_{A,x_j,t}\right\}$ *for all* $1 \leq i \leq k$.

Proof. With the nature of distributions of $\hat{\theta}_Q, \hat{\theta}_S$ as in Claim 11, due to Claim 7, it is sufficient to show that the membership sets $C_{A,x_\cdot,t}$s are star-shaped polytopes (Definition 9) with its kernel containing θ^*, the mean of the distributions.

To start, for $i = 1$, $C_{A,x_1,t}$ is a convex set (Claim 10) and is thus star-shaped, so the entire set is its own kernel, and θ^* belongs to this set and is thus in its kernel.

For $i \in [2, k]$, we show this using proof by contradiction. Assume for some i that $\bigcup_{j=1}^{i} C_{A,x_j,t}$ is not a star-shaped polytop w.r.t kernel point θ^*. Then, there exists some ray originating at θ^* and traversing some θ_b and θ_a (in that order) such that $\theta_a \in C_{A,x_a,t}$ and $\theta_b \in C_{A,x_b,t}$ with $b > i \geq a$, i.e., action x_a is more optimal than action x_b. In other words, using observation Equation (22), there is some direction in which as the estimate $\hat{\theta}_A$ moves away from the true θ^*, the action picked by the algorithm ceases to be the optimal action x_1 and changes to the sub-optimal x_b as the estimate gets to θ_b, and then changes to a relatively optimal arm x_a as the estimate moves further away to θ_a.

As $\theta_b \in C_{A,x_b,t}$, by its definition Equation (21),

$$\begin{aligned}
&\theta_b x_b^\top \geq \theta_b x_a^\top \\
&\overset{\text{(a)}}{\implies} (c\theta^* + (1-c)\theta_a)x_b^\top \geq (c\theta^* + (1-c)\theta_a)x_a^\top \\
&\implies c\theta^* x_b^\top \geq c\theta^* x_a^\top + (1-c)\left(\theta_a x_a^\top - \theta_a x_b^\top\right) \\
&\overset{\text{(b)}}{\implies} c\theta^* x_b^\top \geq c\theta^* x_a^\top \\
&\implies \theta^* x_b^\top \geq \theta^* x_a^\top,
\end{aligned} \tag{23}$$

where, (a) is by linear interpolation for some $0 < c < 1$, (b) uses that $\theta_a \in C_{A,x_a,t}$ to have $\theta_a x_a^\top - \theta_a x_b^\top > 0$.

Finally, Equation (23) implies x_b is more optimal than x_a with $b > a$, which is a contradiction. This completes the proof of the Claim. □

The above Claim, in conjunction with Equation (22), completes the proof of the Lemma. □

Finally, to complete the proof of the Theorem, we show in Claim 13, for any time t, the instantaneous regret of M-ETC with coalition Q is no greater than that of M-ETC with coalition S.

Claim 13 (Instantaneous regret inequality)**.** *For any agent* $a \in S \subseteq Q$*, at any time* $t \in [T' + 1, T]$*,*

$$\sum_{j=2}^{k} \mathbb{P}^Q\left\{x_{a,t} = x_j\right\}\Delta_j \leq \sum_{j=2}^{k} \mathbb{P}^S\left\{x_{a,t} = x_j\right\}\Delta_j,$$

where $\Delta_j := \max_{x \in X_{a,t}} \langle \theta^*, x\rangle - \langle \theta^*, x_j\rangle$ *is the sub-optimality 'gap' of action* $x_j \in X_{a,t}$*.*

Proof. Write short-hands $p_i^Q = \mathbb{P}^Q\left\{x_{a,t} = x_i\right\}$, $p_{i,k}^Q = \mathbb{P}^Q\left\{x_{a,t} \in \{x_i, x_{i+1}, \ldots, x_k\}\right\}$ and similarly define $p_i^S, p_{i,k}^S$. We show the claim by induction on action index i.

Hypothesis.

$$H(i) : \sum_{j=i}^{k}\left(p_j^Q - p_j^S\right)\Delta_j \leq \left(p_{i,k}^Q - p_{i,k}^S\right)\Delta_i. \tag{24}$$

Base Case. Statement $H(k)$ holds as

$$p_k^Q \cdot \Delta_k - p_k^S \cdot \Delta_k = \left(p_{k,k}^S - p_{k,k}^Q\right)\Delta_k.$$

Induction Step. Let $H(i+1)$ be true for some $i+1 \leq k$. We show $H(i)$ is true.

$$\begin{aligned}\sum_{j=i}^{k}\left(p_j^Q - p_j^S\right)\Delta_j &= \left(p_i^Q - p_i^S\right)\Delta_i + \sum_{j=i+1}^{k}\left(p_j^Q - p_j^S\right)\Delta_j \\ &\overset{(a)}{\leq} \left(p_i^Q - p_i^S\right)\Delta_i + \left(p_{i+1,k}^Q - p_{i+1,k}^S\right)\Delta_{i+1} \\ &\overset{(b)}{\leq} \left(p_i^Q - p_i^S\right)\Delta_i + \left(p_{i+1,k}^Q - p_{i+1,k}^S\right)\Delta_i = \left(p_{i,k}^Q - p_{i,k}^S\right)\Delta_i. \end{aligned} \tag{25}$$

Here, (a) upper bounds the second term by using the induction assumption $H(i+1)$, then (b) is due to $p_{i+1,k}^Q - p_{i+1,k}^S \leq 0$ by Lemma 9 and $\Delta_i \leq \Delta_{i+1}$. And Equation (25) shows $H(i)$. By mathematical induction, we have that $H(2)$ holds.

Along with the observation that $p_{i,k}^Q - p_{i,k}^S \leq 0$ from Lemma 9, statement $H(2)$ implies the Claim. □

This completes the proof of the Theorem. □

C MISSING PROOFS FROM SECTION 3

C.1 Proof of Theorem 1

Theorem 1. *When the meta-algorithm* Mul *is run on any problem instance $I \in \mathcal{I}^{fixed}$ for a sufficiently large time horizon T, then, if the black-box single-agent algorithm* Sin *used obeys Assumption 1, the resultant collaboration game $(M, v_{\text{Mul},I,T})$ is convex.*

Proof. The result shall be shown in two steps. First, in Lemma 1, we shall neatly bound the cumulative regret of the agents in a coalition running Mul for time T using the analytical regret of the single-agent algorithm Sin (that Mul internally uses) run for a longer time of mT, where m is the size of the coalition. Second, we shall use this neat bound to show that the value function of the collaboration game $v_{\text{Mul},I,T}$ is supermodular for large enough values of T to complete the proof.

Lemma 1. *For any time-horizon T and problem instance $I \in \mathcal{I}^{fixed}$, for any agent $a \in C$,*

$$\begin{aligned} R_a(\text{Sin}, I, mT) - mK &\leq \sum_{a \in C} R_a^C(\text{Mul}, I, T) \\ &\leq R_a(\text{Sin}, I, mT) + mK, \end{aligned}$$

where $m = |C|$ is the number of agents in coalition C, and $K = |X|$ is the size of the action set X.

Proof. We describe the two different regret quantities mentioned in the Lemma statement and introduce a third quantity to connect them.

1. The realized cumulative regret of all agents, $\sum_{a \in C} R_a^C(\text{Mul}, I, \cdot)$, the quantity we try to bound from both sides.

2. The analytical/hypothetical single-agent regret $R_a(\text{Sin}, I, \cdot)$ that is used in the bound terms.

3. To compare the above two quantities, an intermediate quantity (introduced below in Equation (26)) : $\overline{R}(\text{Sin}, B, \cdot)$, the 'regret' of the black-box decision/action sequence $\overline{x}_\tau$s on interacting with the reward buffer B. This is a quantity internal to Mul algorithm that depends on actions chosen by Sin, and not a quantity inherent to the bandit problem.

We set up some notations: Let $\overline{\tau}$ be the final value of τ after MUL terminates, which is also the number of times SIN has been fed with rewards fetched/removed from the buffer (in line 7). Now, we define the 'regret' of the black-box decision-maker up to some step $\tau' \leq \overline{\tau}$ to be

$$\overline{R}(\text{SIN}, B, \tau') := \sum_{\tau=1}^{\tau'} \max_{x \in X} \mathop{\mathbb{E}}_{B} [\overline{y}_\tau | \overline{x}_\tau = x] - \mathop{\mathbb{E}}_{B,\text{SIN}} [\overline{y}_\tau] , \tag{26}$$

where the expectation is over any randomness in the black-box decision-making and in the buffer rewards.

The following Lemma equates this quantity $\overline{R}(\text{SIN}, B, \tau')$ to the hypothetical single-agent regret for a similar time period:

Lemma 14 (Lemma 1 of Howson et al. [2024])**.** *When the bandit rewards obtained depend only on the chosen action (and not the time or agent): for all time steps $t \in [T]$, agents $a \in M$, actions $x \in X$, the observed rewards $y_{a,t} \mid (x_{a,t} = x) \overset{i.i.d.}{\sim} \langle \theta^*, x \rangle + \eta_{a,t}$. Then, for any time $\tau' \leq \overline{\tau}$,*

$$R_a(\text{SIN}, I, \tau') = \overline{R}(\text{SIN}, B, \tau').$$

We present a proof in Section C.2 for the sake of completion. And, our problem setting shares the assumption of the Lemma wherein the reward of an agent at any time depends only on the action played by him at that time, and not on the history, or the identity of agent, or the time of play.

Next, we compare the cumulative regret of the agents in the coalition, $\sum_{a \in C} R_a^C$ to the black-box regret $\overline{R}(\cdot)$ by studying how many steps the black-box decision-maker performs. After m agents in the coalition have played the bandit instance I for T time steps, the number of steps the black-box decision-maker interacts with the buffer, $\overline{\tau}$, is bounded as follows:

Claim 15. *In all algorithmic runs/trajectories, $mT - mK \leq \overline{\tau} \leq mT$.*

Proof. As per the algorithm MUL, the black-box decision-maker necessarily reads (and removes) one sample from the buffer at every step. There are in total mT samples observed by the agents and filled into the buffer, and that is the maximum number of samples the black-box decision-maker could've read out of the buffer. This shows the upper bound.

When black-box decision-maker wants reward for any action x, the agents play this action on the bandit instance *only* when the buffer B_x is empty, and fill it with m samples. No more samples are added to the buffer (the agents don't play this action x again) until the black-box decision-maker reads all the samples and empties the buffer for this action. Thus, at any time, the size (number of yet-to-be-used available samples) of the buffer is at most mK, where K is the number of actions in the action set X. Any sample that has been removed from the buffer has necessarily been consumed by the black-box decision-maker. With a total mT samples filled into the buffer and at most mK unused samples remaining at any time, the black-box decision-maker has interacted with the buffer for at least $mT - mK$ steps. This gives the lower bound. □

In all trajectories/runs of MUL, for every sample $y_{a,t}$ obtained by every agent $a \in M$ at time $t \in [T]$, one of the below two statements hold.

1. It remains in the buffer at the end of time horizon $t = T$. Let B' be the set of agent-time tuples (a, t)-s whose samples $y_{a,t}$-s remain in the buffer at the end. Or,

2. It was consumed by the black-box decision-maker at some step $\tau \in [\overline{\tau}]$. Let $f : [\overline{\tau}] \mapsto (M \times [T]) \setminus B'$ denote this bijective mapping.

Now, we are ready to bound the cumulative regret of all agents in C as follows:

$$\begin{aligned}
\sum_{a\in C} R_a^C(\textsc{Mul}, I, T) &\overset{(a)}{=} \sum_{a\in C}\sum_{t=1}^{T} y^* - \mathop{\mathbb{E}}_{I,\textsc{Mul}}[\langle\theta^*, x_{a,t}\rangle] \\
&\overset{(b)}{=} \sum_{\tau=1}^{\overline{\tau}} y^* - \left\langle\theta^*, x_{f(\tau)}\right\rangle + \sum_{(a,t)\in B'} y^* - \langle\theta^*, x_{a,t}\rangle \\
&\overset{(c)}{=} \sum_{\tau=1}^{\overline{\tau}} y^* - \langle\theta^*, \overline{x}_\tau\rangle + \sum_{(a,t)\in B'} y^* - \langle\theta^*, x_{a,t}\rangle \qquad (27)\\
&\overset{(d)}{\le} \sum_{\tau=1}^{\overline{\tau}} y^* - \langle\theta^*, \overline{x}_\tau\rangle + mK \\
&= \overline{R}(\textsc{Sin}, B, \overline{\tau}) + mK \\
&\overset{(e)}{=} R_a(\textsc{Sin}, I, \overline{\tau}) + mK \overset{(f)}{\le} R_a(\textsc{Sin}, I, mT) + mK. \qquad (28)
\end{aligned}$$

Here, (a) uses $X = X_{a,t}$ for all a, t from problem instance constraint, and introduces short-hand $y^* := \max_{x\in X}\langle\theta^*, x\rangle$, and in (b), we split the regret into two terms based on whether the rewards obtained were consumed by the black-box or not. Then, (c) uses the algorithm property that black-box is given a reward for an action x from the buffer which was originally filled with rewards for action x. Then, (d) uses Claim 15 to bound $|B'| = mT - \overline{\tau} \le mK$ while liberally upper bounding per-time step regret with 1. Then, (e) is by Lemma 14, and finally (f) uses Claim 15 again.

Continuing again from Equation (27), we have

$$\begin{aligned}
\sum_{a\in C} R_a^C(\textsc{Mul}, I, T) &\ge \sum_{\tau=1}^{\overline{\tau}} y^* - \langle\theta^*, \overline{x}_\tau\rangle = \overline{R}(\textsc{Sin}, B, \overline{\tau}) \\
&\overset{(a)}{=} R_a(\textsc{Sin}, I, \overline{\tau}) \overset{(b)}{\ge} R_a(\textsc{Sin}, I, mT) - mK, \qquad (29)
\end{aligned}$$

where (a) is by Lemma 14, and (b) uses $\overline{\tau} \ge mT - mK$ from Claim 15 and liberally upper bounds per-time step regret with 1.

Equations (28) and (29) together show the Lemma. □

Note that the upper and lower bounds in the Lemma only differ by an additive term that is independent of time horizon T.

We next try to show supermodularity of the value function. On any instance I, for any two sets of agents $S \subset Q \subseteq M$, with $|S| = s, |Q| = q$, and any agent $a \in M \setminus Q$, we want to show the inequality

$$\begin{aligned}
& v(S\cup\{a\}) - v(S) \le v(Q\cup\{a\}) - v(Q) \\
\iff\quad & -v(S) - v(Q\cup\{a\}) \le -v(S\cup\{a\}) - v(Q) \\
\iff\quad & \sum_{b\in S} R_b^S(\textsc{Mul}, T) + \sum_{b\in Q\cup\{a\}} R_b^{Q\cup\{a\}}(\textsc{Mul}, T) \le \sum_{b\in S\cup\{a\}} R_b^{S\cup\{a\}}(\textsc{Mul}, T) + \sum_{b\in Q} R_b^Q(\textsc{Mul}, T) \\
\overset{(a)}{\Longleftarrow}\quad & R_a(\textsc{Sin}, sT) + sK + R_a(\textsc{Sin}, (q{+}1)T) + (q{+}1)K \le R_a(\textsc{Sin}, (s{+}1)T) - (s{+}1)K + R_a(\textsc{Sin}, qT) - qK \\
\overset{(b)}{\Longleftarrow}\quad & R_a(\textsc{Sin}, qT{+}T) - R_a(\textsc{Sin}, qT) \le R_a(\textsc{Sin}, sT{+}T) - R_a(\textsc{Sin}, sT) - 4MK
\end{aligned} \qquad (30)$$

Here, (a) is due to Lemma 1, and (b) uses $s + q + 1 \le 2M$.

What remains to be shown is that Equation (30) holds, which postulates that for a given problem instance, the regret of the single-agent algorithm over a time period of T that begins after time qT be lesser than the regret over a similar duration period that begins after time sT by a margin of $4MK$.

Using the notations introduced in Assumption 1, we try to show Equation (30) as follows:

$$\begin{aligned}
&R_a(\textsc{Sin}, sT{+}T) - R_a(\textsc{Sin}, sT) - (R_a(\textsc{Sin}, qT{+}T) - R_a(\textsc{Sin}, qT)) \\
&= T \cdot R'(sT, T) - T \cdot R'(qT, T) = T(qT - sT) \cdot -R^{''}(sT, qT - sT, T) \\
&\overset{(a)}{\geq} T^2 \cdot \upsilon_{sT} \overset{(b)}{\geq} \frac{4MK}{\min_t \upsilon_t} \cdot \upsilon_{sT} \geq 4MK.
\end{aligned} \tag{31}$$

Here (a) uses Equation (3) (Assumption 1), and (b) uses the constraint on T from the Theorem statement. We further ascertain that such a sufficiently large T (dependent on υ_ts, M, and K) is feasible as follows:

$$T \geq \sqrt{\frac{4MK}{\min_t \upsilon_t}} \overset{(a)}{\geq} \sqrt{\frac{4MK}{cT^{-2+\varepsilon}}} = \sqrt{4MK/c} \cdot T^{1-\varepsilon/2} \iff T > \left(\frac{4MK}{c}\right)^{1/\varepsilon},$$

a lower bound (r.h.s.) independent of T. Here, (a) uses Equation (4) (Assumption 1). Equation (31) shows the Theorem. □

C.2 Proof of Lemma 14

This Lemma is originally shown by Howson et al. [2024], and we provide here a proof for the sake of completeness.

Proof. The L.H.S. term can be written from Equation (1) as follows:

$$\begin{aligned}
R_a(\textsc{Sin}, I, \tau') &= \sum_{t=1}^{\tau'} \max_{x \in X_t} \langle \theta^*, x \rangle - \mathop{\mathbb{E}}_{I,\textsc{Sin}} [\langle \theta^*, x_{a,t} \rangle] \\
&\overset{(a)}{=} \sum_{x \in X} \left(\max_{x' \in X} \langle \theta^*, x' \rangle - \langle \theta^*, x \rangle \right) \cdot \mathop{\mathbb{E}}_{I,\textsc{Sin}} \left[\sum_{t=1}^{\tau'} \mathbb{1}\{x_{a,t} = x\} \right] \overset{(b)}{=} \sum_{x \in X} \Delta_x \cdot \mathop{\mathbb{E}}_{I,\textsc{Sin}} \left[\sum_{t=1}^{\tau'} \mathbb{1}\{x_{a,t} = x\} \right]
\end{aligned} \tag{32}$$

Here, (a) is due to action set X being fixed across time, then (b) introduces $\Delta_x = \max_{x' \in X} \langle \theta^*, x' \rangle - \langle \theta^*, x \rangle$ to be the time-independent 'sub-optimality' gap of action x.

The R.H.S term can be written from Equation (26) as

$$\begin{aligned}
\overline{R}(\textsc{Sin}, B, \tau') &\overset{(a)}{=} \sum_{\tau=1}^{\tau'} \max_{x \in X_\tau} \mathop{\mathbb{E}}_{B} [\overline{y}_\tau | \overline{x}_\tau = x] - \mathop{\mathbb{E}}_{B,\textsc{Sin}} [\overline{y}_\tau] \\
&\overset{(b)}{=} \sum_{\tau=1}^{\tau'} \max_{x \in X_\tau} \langle \theta^*, x \rangle - \mathop{\mathbb{E}}_{B,\textsc{Sin}} [\langle \theta^*, \overline{x}_\tau \rangle] \\
&\overset{(c)}{=} \sum_{x \in X} \Delta_x \cdot \mathop{\mathbb{E}}_{B,\textsc{Sin}} \left[\sum_{\tau=1}^{\tau'} \mathbb{1}\{\overline{x}_\tau = x\} \right].
\end{aligned} \tag{33}$$

Here, (a) is by Equation (26), and (b) comes from the nature of algorithm Mul as follows: the reward $\overline{y}_\tau | \overline{x}_\tau = x$ is arbitrarily picked from the buffer B_x, and any reward that was put into this buffer B_x was from playing the action x on bandit instance I whose expected reward is $\langle \theta^*, x \rangle$. Then, (c) is from action set X being fixed across time and uses Δ_x defined in the L.H.S.

From Equations (32) and (33), what remains to be shown is the following claim:

Claim 16 (Equivalence of expected play-counts)**.** *For all actions $x \in X$, and $\tau' \leq \tau^{max}$,*

$$\mathop{\mathbb{E}}_{I,\textsc{Sin}}\left[\sum_{t=1}^{\tau'} \mathbb{1}\left\{x_{a,t} = x\right\}\right] = \mathop{\mathbb{E}}_{B,\textsc{Sin}}\left[\sum_{\tau=1}^{\tau'} \mathbb{1}\left\{\overline{x}_{\tau} = x\right\}\right].$$

Proof. Note that the (random variable) quantities, whose expectations are compared in the Claim, are both functions of the history/trajectory of bandit play. Recollect $H_{a,\tau'} = (x_{a,1}, y_{a,1}, x_{a,2}, y_{a,2}, \cdots, x_{a,\tau'}, y_{a,\tau'})$ (resp. $\overline{H}_{\tau'} = (\overline{x}_1, \overline{y}_q, \cdots, \overline{x}_{\tau'}, \overline{y}_{\tau'})$) to be the histories of hypothetical run of Sin on instance I (resp. Sin on buffer B).

The action spaces $x_{a,\cdot}, \overline{x}_{\cdot} \in X$ are identical between the two. And the space of rewards $y_{a,\cdot}, \overline{y}_{\cdot} \in \mathbb{R}$ in instance I and buffer B are also identical as the buffer B is filled with rewards obtained from multi-agent interaction of agent I. So, the sample spaces of the two histories are the same, say $\mathcal{H} = (X \times \mathbb{R})^{\tau'}$.

We use $h = (x_1, y_2, x_2, y_2, \cdots, x_s, y_s, \cdots, x_{\tau'}, y_{\tau'})$ to denote some complete history in $\mathcal{H}$, and $h_s = (x_1, y_1, \cdots, x_s, y_s)$ for $s \leq \tau'$ to denote a prefix of the history.

We start from R.H.S.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{B,\textsc{Sin}}\left[\sum_{t=1}^{\tau'} \mathbb{1}\left\{\overline{x}_t = x\right\}\right] \overset{\text{(a)}}{=} \int_{h\in\mathcal{H}} f(h) \cdot \mathbb{P}_{B,\textsc{Sin}}\left\{\overline{H}_{\tau'} = h\right\} dh \\
&\overset{\text{(b)}}{=} \int_{h\in\mathcal{H}} f(h) \cdot \prod_{s=1}^{\tau'} \mathbb{P}_B\left\{\overline{y}_s = y_s \mid \overline{x}_s = x_s, \overline{H}_{s-1} = h_{s-1}\right\} \times \mathbb{P}_{\textsc{Sin}}\left\{\overline{x}_s = x_s \mid \overline{H}_{s-1} = h_{s-1}\right\} \cdot dh \\
&\overset{\text{(c)}}{=} \int_{h\in\mathcal{H}} f(h) \cdot \prod_{s=1}^{\tau'} \mathbb{P}_B\left\{\overline{y}_s = y_s \mid \overline{x}_s = x_s\right\} \times \mathbb{P}_{\textsc{Sin}}\left\{\overline{x}_s = x_s \mid \overline{H}_{s-1} = h_{s-1}\right\} \cdot dh \\
&\overset{\text{(d)}}{=} \int_{h\in\mathcal{H}} f(h) \cdot \prod_{s=1}^{\tau'} \mathbb{P}_I\left\{y_{a,s} = y_s \mid x_{a,s} = x_s\right\} \times \mathbb{P}_{\textsc{Sin}}\left\{\overline{x}_s = x_s \mid \overline{H}_{s-1} = h_{s-1}\right\} \cdot dh \\
&\overset{\text{(e)}}{=} \int_{h\in\mathcal{H}} \sum_{t=1}^{\tau'} \mathbb{1}\left\{x_{a,t} = x\right\} \cdot \prod_{s=1}^{\tau'} \mathbb{P}_I\left\{y_{a,s} = y_s \mid x_{a,s} = x_s\right\} \times \mathbb{P}_{\textsc{Sin}}\left\{x_{a,s} = x_s \mid H_{a,s-1} = h_{s-1}\right\} \cdot dh \\
&= \int_{h\in\mathcal{H}} \sum_{t=1}^{\tau'} \mathbb{1}\left\{x_{a,t} = x\right\} \cdot \mathbb{P}_{I,\textsc{Sin}}\left\{H_{a,\tau'} = h\right\} dh = \mathop{\mathbb{E}}_{I,\textsc{Sin}}\left[\sum_{t=1}^{\tau'} \mathbb{1}\left\{x_{a,t} = x\right\}\right].
\end{aligned} \tag{34}$$

Here, (a) introduces short-hand $f(h) := \sum_{\tau=1}^{\tau'} \mathbb{1}\left\{\overline{x}_{\tau} = x\right\}$. Then, (b) splits and introduces two measures: $\mathbb{P}_B\{.\}$ as the randomness in the reward given the action comes from only the buffer B and not the action playing algorithm; and similarly $\mathbb{P}_{\textsc{Sin}}\{.\}$ as the randomness in the choice of action to play given the history comes from only the algorithm Sin and not the buffer environment B. Then, (c) is due to the fact that at any time s, the reward obtained $\overline{y}_s$ from buffer is conditionally independent to the history $\overline{H}_{s-1}$ of actions and rewards, given the action played $\overline{x}_s$ at that time. Then, (d) is from the nature of the buffer usage as follows: As the reward $\overline{y}_s|\overline{x}_s = x_s$ picked/removed from buffer B_{x_s} was originally filled by some agent at some time, say agent b at time s', by interacting with bandit instance I, we have $\mathbb{P}_B\{\overline{y}_s = y_s|\overline{x}_s = x_s\} = \mathbb{P}_I\{y_{b,s'} = y_s|x_{b,s'} = x_s\}$; further, as the rewards only depend on the action played and not the agent who played it or the time at which it was played, we have $\mathbb{P}_I\{y_{b,s'} = y_s|x_{b,s'} = x_s\} = \mathbb{P}_I\{y_{a,s} = y_s|x_{a,s} = x_s\}$. In (e), we replace running variable τ with t, and rename history variable $\overline{H}_s = (\overline{x}_1, \overline{y}_1, \cdots, \overline{x}_s, \overline{y}_s)$ to $H_{a,t} = (x_{a,s}, y_{a,s}, \cdots, x_{a,s}, y_{a,s})$. Finally, Equation (34) shows the Claim. □

This completes the proof of the Lemma. □

D MISSING PROOFS FROM SECTION 4

D.1 Proof of Theorem 3

Theorem 3. *For any bandit instance $I \in \mathcal{I}$, if* ALG *satisfies Assumptions 2 and 3, then, the allocation $p = \left(p_a = -R_a^M(\text{ALG}, I, T)\right)_{a \in M}$ obeys the axioms of efficiency, dummy-player, and symmetry, and also belongs to the core of the collaboration game $(M, v_{\text{ALG},I,T})$.*

Proof. We show that the allocation obeys the different axioms in a series of claims below. We write v and R_a^C in short to denote $v_{\text{ALG},I,T}$ and $R_a^C(\text{ALG}, I, T)$ for any coalition $C \subseteq M$.

Claim 17 (Efficiency)**.** *Payout profile p is efficient. $\sum_{a \in M} p_a = v(M)$.*

Proof. The Claim is immediate as $\sum_{a \in M} p_a = -\sum_{a \in M} R_a^M = v(M)$. □

Claim 18 (Membership in core)**.** *Payout profile p belongs to the core of the collaboration game.*

Proof. A payout profile belongs to the core if it is efficient and it is coalitionally rational. Claim 17 shows p is efficient. Next, we show coalitional rationality. Consider any coalition $S \subseteq M$, we have

$$\sum_{a \in S} p_a = -\sum_{a \in S} R_a^M \overset{\text{(a)}}{\geq} -\sum_{a \in S} R_a^S = v(S),$$

where (a) is due to Assumption 3. □

Claim 19 (Dummy player)**.** *Payout profile p obeys the dummy player (or null player) axiom, i.e., for all agents $a \in M$,*

$$(\forall S \not\ni a : v(S \cup \{a\}) - v(S) = v(\{a\})) \implies p_a = v(\{a\}).$$

Proof. We start with the LHS: for all coalitions $S \not\ni a$,

$$\begin{aligned} v(S \cup \{a\}) - v(S) = v(\{a\}) &\iff \left(\sum_{b \in S} R_b^S - R_b^{S \cup \{a\}}\right) - R_a^{S \cup \{a\}} = -R_a \\ \overset{\text{(a)}}{\implies} -R_a^{S \cup \{a\}} \leq -R_a &\implies R_a \leq R_a^{S \cup \{a\}} \overset{\text{(b)}}{\implies} R_a = R_a^{S \cup \{a\}}, \end{aligned} \tag{35}$$

where (a), (b) use $R_b^S - R_b^{S \cup \{a\}} \geq 0$ and $R_a \geq R_a^{S \cup \{a\}}$ from Assumption 3. We conclude the proof by showing the RHS: $p_a = -R_a^M \overset{\text{(c)}}{=} -R_a = v(\{a\})$, where (c) is due to Equation (35) with a choice of $S = M \setminus \{a\}$. □

Claim 20. *[Symmetry] Payout profile p obeys the symmetry axiom.*

Proof sketch. Deferring the formal proof to Section D.2, we provide a sketch here. The symmetry axiom mandates that the payout of an agent—in our context, the regret of an agent a in the grand coalition—doesn't change when the agents are relabeled. We shall rely on Assumption 2 and argue that all the agent information is removed during the union operation to pool the data, and it is this pool of (anonymized) data that the algorithm uses to determine the actions to play. Conditioned on this pool, the action choice only depends on the agent's action set and not the agent's identity. As a result, the distribution of trajectories remains unchanged under relabeling of agents, and thus the regret, and by extension, his payout $p_a = -R_a^M$ remains unchanged under relabeling of agents. □

This completes the proof of the Theorem. □

D.2 Proof of Claim 20

Claim 20. *[Symmetry] Payout profile p obeys the symmetry axiom.*

Proof. We set up some notation. Consider a permutation/bijection $\pi : [M] \mapsto [M]$. Enumerate the set of agents to be $M = \{a_1, a_2, \ldots, a_M\}$. For the agent $a_i \in M$ with label/id i, let his permuted label/id be $\pi(i) \in [M]$. Then, construct the permuted set of agents to be $\tilde{M} = \{\tilde{a}_{\pi(i)} = a_i : i \in [M]\}$, and extend all notations defined earlier using an overhead $\tilde{}$ for this permuted set of agents. Given some i, write short-hands $a = a_i$, $\tilde{a} = \tilde{a}_{\pi(i)}$. Then, denote by $\tilde{H}_{\tilde{a},t} = \{(\tilde{x}_{\tilde{a},1}, \tilde{y}_{\tilde{a},1}), \ldots, (\tilde{x}_{\tilde{a},t}, \tilde{y}_{\tilde{a},t})\}$ the history/trajectory of actions played and rewards observed by agent $\tilde{a} \in \tilde{M}$ up to time t when algorithm ALG is run with the set of agents $\tilde{M}$.

We want to show that the allocation p_a for any agent $a_i \in M$ doesn't change when he participates in the collaboration game under a different label as a part of a permuted set. Specifically, it requires to be shown that for any permuted agent set $\tilde{M}$ (induced by permutation π), for any agent $a_i \in M$, it holds that $R^{M}_{a_i} = R^{\tilde{M}}_{\tilde{a}_{\pi(i)}}$. We shall show that this condition follows when the algorithm ALG facilitates (carries out) collaboration among agents as per Assumption 2.

First, we claim that the trajectory of an agent $a_i \in M$ when ALG is run on set of agents M is identically distributed to the trajectory of the corresponding relabeled/permuted agent when ALG is run on the relabeled/permuted set of agents $\tilde{M}$. Let $(s_a)_a = ((\bar{x}_{a,1}, \bar{y}_{a,1}), \ldots, (\bar{x}_{a,t}, \bar{y}_{a,t}))_{a \in M} \in S$ be a collection of action-reward sequences, with S the set of all such possible collection of sequences.

Claim 21. *For any $t \in [T]$, for any collection $(s_a)_a \in S$ of action-reward sequences, it holds that*

$$\mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} H_{a,t} = s_a\right] = \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{\tilde{a} \in \tilde{M}} \tilde{H}_{\tilde{a},t} = s_a\right]. \tag{36}$$

Proof. We show this by induction on time step variable t. Let $h(t)$ be the hypothesis that Equation (36) holds for t.

Base Case. At $t = 0$, the only valid action-reward sequence, for any agent $a \in M$ is the empty sequence $s_a = ()$, and both $H_{a,0} = s$ and $\tilde{H}_{\tilde{a},0} = s$ with probability 1. Thus $h(0)$ holds.

Induction Step. We assume $h(t-1)$ holds and shall show $h(t)$ holds. Let $s_{a,t-1} := ((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}))$ be the sequence of first $t-1$ action-reward tuples in s_a. We start from the L.H.S:

$$\begin{aligned}\mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} H_{a,t} = s_a\right] &= \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} y_{a,t} = \bar{y}_{a,t} \middle| \bigcap_{a \in M} x_{a,t} = \bar{x}_{a,t}, \bigcap_{a \in M} H_{a,t\text{-}1} = s_{a,t\text{-}1}\right] \\ &\quad \cdot \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} x_{a,t} = \bar{x}_{a,t} \middle| \bigcap_{a \in M} H_{a,t\text{-}1} = s_{a,t\text{-}1}\right] \cdot \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} H_{a,t\text{-}1} = s_{a,t\text{-}1}\right].\end{aligned} \tag{37}$$

Next, we analyse each of these three terms. First,

$$\begin{aligned}&\mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} y_{a,t} = \bar{y}_{a,t} \middle| \bigcap_{a \in M} x_{a,t} = \bar{x}_{a,t}, \bigcap_{a \in M} H_{a,t\text{-}1} = s_{a,t\text{-}1}\right] \\ &\overset{\text{(a)}}{=} \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{a \in M} y_{a,t} = \bar{y}_{a,t} \middle| \bigcap_{a \in M} x_{a,t} = \bar{x}_{a,t}\right] \overset{\text{(b)}}{=} \prod_{a \in M} \mathop{\mathbb{P}}_{I,\text{ALG}}[y_{a,t} = \bar{y}_{a,t} | x_{a,t} = \bar{x}_{a,t}] \\ &\overset{\text{(c)}}{=} \prod_{\tilde{a} \in \tilde{M}} \mathop{\mathbb{P}}_{I,\text{ALG}}[\tilde{y}_{\tilde{a},t} = \bar{y}_{a,t} | \tilde{x}_{\tilde{a},t} = \bar{x}_{a,t}] \overset{\text{(d)}}{=} \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{\tilde{a} \in \tilde{M}} \tilde{y}_{\tilde{a},t} = \bar{y}_{a,t} \middle| \bigcap_{\tilde{a} \in \tilde{M}} \tilde{x}_{\tilde{a},t} = \bar{x}_{a,t}\right] \\ &\overset{\text{(e)}}{=} \mathop{\mathbb{P}}_{I,\text{ALG}}\left[\bigcap_{\tilde{a} \in \tilde{M}} \tilde{y}_{\tilde{a},t} = \bar{y}_{a,t} \middle| \bigcap_{\tilde{a} \in \tilde{M}} \tilde{x}_{\tilde{a},t} = \bar{x}_{a,t}, \bigcap_{\tilde{a} \in \tilde{M}} \tilde{H}_{\tilde{a},t\text{-}1} = s_{a,t\text{-}1}\right].\end{aligned} \tag{38}$$

The above is due to the independence of the rewards. Specifically, (a),(e) use that given the current action, rewards are independent of the historical actions/rewards, (b),(d) are from the independence of rewards across different agent-play at the same time, and (c) is from the fact that the reward is a function of the specific action $\bar{x}_{a,t}$ played, and independent of the agent who plays the action (or the time at which it is played) for any problem instance I.

Second,

$$\begin{aligned}
&\mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{a\in M} x_{a,t} = \bar{x}_{a,t} \,\middle|\, \bigcap_{a\in M} H_{a,t\text{-}1} = s_{a,t\text{-}1}\right] \\
&\overset{(a)}{=} \mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{a\in M}\left(x_{a,t} = \bar{x}_{a,t} \,\middle|\, \left(P^a_{t-1} = ((\bar{x}_{a,s},\bar{y}_{a,s}))_{s\in[t-1]}\right) \wedge \left(P^{-a}_{t-1} = \left(\cup_{b\in M\setminus\{a\}}\{(\bar{x}_{b,s},\bar{y}_{b,s})\}\right)_{s\in[t-1]}\right)\right)\right] \\
&\overset{(b)}{=} \mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{a\in M}\left(x_{a,t} = \bar{x}_{a,t} \,\middle|\, \left(P^a_{t-1} = ((\bar{x}_{\tilde{a},s},\bar{y}_{\tilde{a},s}))_{s\in[t-1]}\right) \wedge \left(P^{-a}_{t-1} = \left(\cup_{\tilde{b}\in \tilde{M}\setminus\{\tilde{a}\}}\{(\bar{x}_{\tilde{b},s},\bar{y}_{\tilde{b},s})\}\right)_{s\in[t-1]}\right)\right)\right] \\
&\overset{(c)}{=} \mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{\tilde{a}\in \tilde{M}} \tilde{x}_{\tilde{a},t} = \bar{x}_{a,t} \,\middle|\, \bigcap_{\tilde{a}\in \tilde{M}} \tilde{H}_{\tilde{a},t\text{-}1} = s_{a,t\text{-}1}\right],
\end{aligned} \tag{39}$$

where (a) is due to Assumption 2 (first condition), and (b) is due to $\tilde{M}$ is a permutation of M, and (c) uses that action set of an agent $X_{a,t} = X_{\tilde{a},t}$ is the same after permuting/relabeling, and that the algorithm decides the action $\tilde{x}_{\tilde{a},t}$ for each agent $\tilde{a}$ as a function of action-reward sequence over time from self-play $P^{\tilde{a}}_{t-1}$ and an action-reward sequence from an anonymized union from other agents' play $P^{-\tilde{a}}_{t-1}$ (from second and third conditions of Assumption 2). Third,

$$\mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{a\in M} H_{a,t\text{-}1} = s_{a,t\text{-}1}\right] = \mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{\tilde{a}\in \tilde{M}} \tilde{H}_{\tilde{a},t\text{-}1} = s_{a,t\text{-}1}\right] \tag{40}$$

by the induction assumption. Finally, substituting Equations (40), (39) and (38) back into Equation (37) shows us that $h(t)$ holds. □

Now, we are ready to show that symmetry holds. The regret of agent a when Alg is run with agents M is given by

$$\begin{aligned}
R^M_a(\textsc{Alg}, I, T) &= \sum_{t=1}^{T}\left\langle \theta^*, x^*_{a,t}\right\rangle - \mathop{\mathbb{E}}_{I,\textsc{Alg}}\left[\langle \theta^*, x_{a,t}\rangle\right] \\
&= \sum_{t=1}^{T}\left\langle \theta^*, x^*_{a,t}\right\rangle - \sum_{(s_a)_a\in S}\left(\sum_{t=1}^{T}\langle \theta^*, \bar{x}_{a,t}\rangle\right)\mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{a\in M} H_{a,t} = s_a\right] \\
&\overset{(a)}{=} \sum_{t=1}^{T}\left\langle \theta^*, \tilde{x}^*_{\tilde{a},t}\right\rangle - \sum_{(s_a)_a\in S}\left(\sum_{t=1}^{T}\langle \theta^*, \bar{x}_{a,t}\rangle\right)\mathop{\mathbb{P}}_{I,\textsc{Alg}}\left[\bigcap_{\tilde{a}\in M} \tilde{H}_{\tilde{a},t} = s_a\right] \\
&= R^{\tilde{M}}_{\tilde{a}}(\textsc{Alg}, I, T),
\end{aligned} \tag{41}$$

where (a) uses Claim 21. Equation (41) completes the proof of the Claim. □

D.3 On the Linearity axiom.

In Theorem 3, we showed that the payout profile of $p = (p_a = -R^M_a(\textsc{Alg}, I, T)$ obeys the Shapley's axioms of efficiency, null-player, and symmetry. However, we mentioned that the linearity axiom cannot be satisfied. We discuss this further in this sub-section.

If the linearity axiom is satisfied by our payout p, then in conjunction with the other satisfied axioms, p indeed ***is*** the Shapley value. However, given that the Shapley value, by definition, captures the marginal utility of an agent

to all possible 2^{M-1} different coalitions, it is highly unlikely that a simple payout structure like ours, $p_a = -R_a^M$, which only captures the dynamism associated with the grand coalition, could actually be the Shapley value. Thus, we believe the Linearity axiom will not be satisfied by p.

What would Linearity look like? Consider two bandit scenarios $B_1 = (\textsc{Alg}, I, T_1)$ and $B_2 = (\textsc{Alg}, I, T_2)$. Consider the corresponding value functions $v_1(C) := v_{\textsc{Alg},I,T_1}(C) = -\sum_{a\in C} R_a^C(\textsc{Alg}, I, T_1)$ and $v_2 := v_{\textsc{Alg},I,T_2}(C) = -\sum_{a\in C} R_a^C(\textsc{Alg}, I, T_2)$.

The sum of two value functions is traditionally obtained by the 'superposition' of the value functions, i.e., the arithmetic sum of the two individual value function (as if the two games are happening in parallel and do not influence each other) is used to traditionally define the sum of two games, say $w = v_1 + v_2$. Namely, the game $(M, v_1 + v_2)$ has the value/characteristic function

$$w(C) = v_1(C) + v_2(C) = -\sum_{a\in C} R_a^C(\textsc{Alg}, I, T_1) + R_a^C(\textsc{Alg}, I, T_2) \tag{42}$$

for all $C \subseteq M$. Now, how does our definition of payouts p for game $(M, v_1 + v_2)$ relate to the underlying bandit scenario of $B_1 + B_2$? What does '+' mean in this context? There appears to be (at least) two formulations.

1. p is the regret going through bandit scenario B_1 (the specific instance I, with the specific algorithm $\textsc{Alg}$, for the specific time-steps) and *then* going through B_2 with outcome of B_1 in memory. In that case,

$$p_a(v_1 + v_2) = -R_a^C(\textsc{Alg}, I, T_1 + T_2) \tag{43}$$

   is the regret through total time $T_1 + T_2$. Note that this preserves the commutativity of + operation (at least when $\textsc{Alg}$ remains the same).

2. p is the regret of going through bandit scenario B_1 summed up with the regret of going through scenario B_2 *independent* of the execution of B_1. In that case,

$$p_a(v_1 + v_2) = -R_a^C(\textsc{Alg}, I, T_1) - R_a^C(\textsc{Alg}, I, T_2) \tag{44}$$

   Note that this preserves the commutativity of + operation always, also appears more aligned to the notion 'superposition'. However, here, the payout definition assumes it is possible to decompose w back into v_1 and v_2 which may not be permissible.

Between v_1 and v_2, if the instances therein differ with model parameter $\theta_1^* \neq \theta_2^*$, then the first case above matches the second case.

Next, we ask if these two definitions of p satisfy the Linearity axiom. First,

$$\begin{aligned} \textit{Equation (43)} \implies p_a(v_1 + v_2) &= -R_a^C(\textsc{Alg}, I, T_1 + T_2) \\ &\neq -R_a^C(\textsc{Alg}, I, T_1) - R_a^C(\textsc{Alg}, I, T_2) = p_a(v_1) + p_a(v_2), \end{aligned}$$

in general. Thus Linearity axiom does not hold in this case. Second,

$$\textit{Equation (44)} \implies p_a(v_1 + v_2) = -R_a^C(\textsc{Alg}, I, T_1) - R_a^C(\textsc{Alg}, I, T_2) = p_a(v_1) + p_a(v_2),$$

which shows p (the version in Equation (44)) satisfies the Linearity axiom.

But, as discussed above, it is not possible to implement (or realize) such a payout p that requires the decomposition of the sum of two games $v_1 + v_2$ back into the original constituent games v_1, v_2. Thus, we feel that our payout, in its current form, cannot satisfy the Linearity axiom.

E ADDITIONAL NUMERIC SIMULATIONS

E.1 Notes on Implementation

In each bandit instance $I_{attr} \in \mathcal{I}$ (for some attribute), the action sets $X_{a,1}, X_{a,2}, \ldots, X_{a,T}$ for an agent a are generated by jointly embedding users and movies in a $d = 100$ dimensional space such that each X_t corresponds

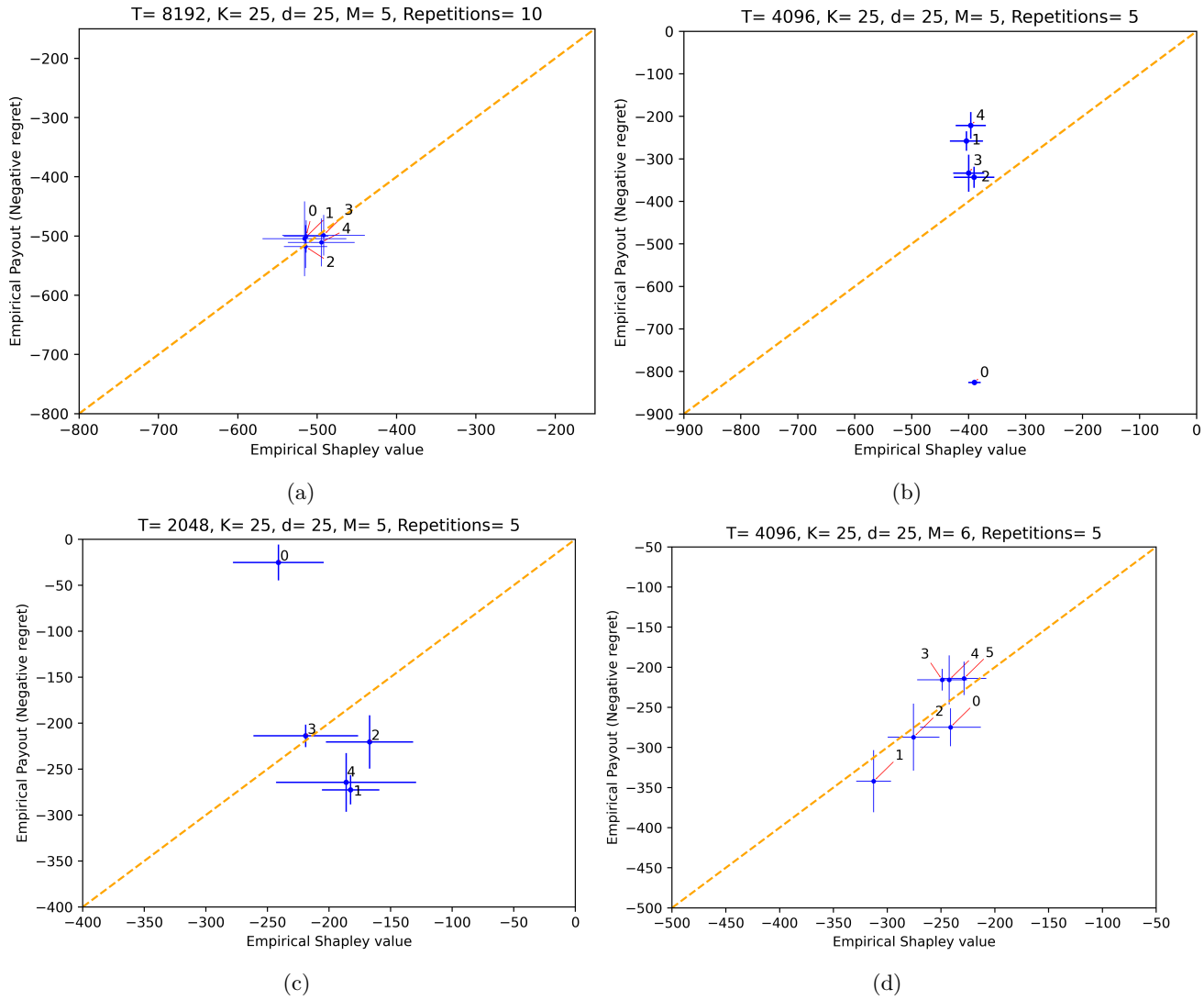


Figure 2: **Synthetic experiments**

to the set of movies that the platform can recommend to a user with attribute value corresponding to that of agent a. We have $|X_{a,t}| = 1682$ for all t for all agents a over all instances I. The parameter $\theta^* \in \mathbb{R}^{100}$ for the linear bandits setup in all instances is the minimizer of the least square error between the predicted ratings and the true ratings. In all experiments, the reward for playing any arm $x \in X_{a,t}$ is $\langle x, \theta^* \rangle + \eta$ where $\eta \sim \mathcal{N}(0, 1)$ independently drawn every time. All experiments are run for a time horizon of $T = 4096$ for 5 repetitions, with mean and error bars (cross-hairs) plotted. For the LinUcb-M, we build upon the LinUCB implementation provided in the Pearl library [Zhu et al., 2023b] and extend it to our multi-agent setting.

E.2 Synthetic Experiments

We perform some synthetic experiments to better elucidate our assumptions and results.

Instance Setup We consider the following 'symmetric' (not to be confused with the Shapley axiom) bandit instance in terms of the action sets of the agents, whereby all agents are identical. With ambient dimension $d = 25$, there are a total of $K = 25$ actions, say $\{1, 2, \ldots, 25\}$, where each action is a unit vector pointing in a unique dimension. This can be equivalently understood as the vanilla multi-armed bandit with d arms. The θ^*

parameter is given by

$$\theta_i^* = \begin{cases} 7/10 & \text{if } i \equiv 1 \pmod 5, \\ 6/10 & \text{if } i \equiv 2 \pmod 5, \\ 5/10 & \text{if } i \equiv 3 \pmod 5, \\ 4/10 & \text{if } i \equiv 4 \pmod 5, \\ 3/10 & \text{if } i \equiv 0 \pmod 5 \end{cases} \tag{45}$$

for all $i \in [d]$. Each θ_i^* can be thought of as the reward means of the action i in the vanilla MAB formulation. From the expression, one can see that these action subsets $A_1 = \{1, 2, \ldots, 5\}, A_2 = \{6, \ldots, 10\}, \ldots, A_5 = \{21, \ldots, 25\}$ all are mutually identical. We then have each agent posses (or be allocated) 2 subsets of actions at all times : for all t, $X_{0,t} = A_1 \cup A_2, X_{1,t} = A_2 \cup A_3, \ldots, X_{4,t} = A_5 \cup A_1$. Individually, each agent posseses an identical set of 10 actions, and viewing the agents arranged in a circle, each agent shares a half of the actions with the neighbour on the left, and shares the other half with the neighbour on the right. All agents are identical w.r.t. their action sets and sharing of action sets with agents.

Outcomes. We run three experiments in this set-up: (i) all agents run LinUCB algorithm; (ii) Agent 0 deviates and plays greedily at all times (maximizing expected reward w.r.t. current parameter estimate without exploration term) while other agents continue to run LinUCB; and (iii) Agent 0 deviates and plays actions uniformly at random while other agents continue to run LinUCB. The comparison of payouts and Shapley value are plotted in Figure 2.

With all identical agents (Figure 2a), every agent's payout is remarkably close to his Shapley value. Further, the payouts (and Shapley values) are also very similar across all agents. In Figure 2c, when there is one agent who plays greedily (that agent is called a 'free-rider' in some contexts), it is seen that the greedy player 0 enjoys incredibly less regret, and has a lower Shapley value than other agents, which implies that he contributes lesser to minimizing the group's regret than other agents do. Further, 1 and 4 are the agents who share actions with free-rider 0, and they have a they have higher regret compared to 2 and 3 who don't share actions with 0. From Figure 2b, the explorer suffers very high regret (very low payout). Agents 1 and 4 benefit by having common actions with this explorer, they enjoy lower regret (higher payout) compared to agents 2 and 3 who don't share actions with the explorer. It is also seen that the Shapley values of all agents appear close to each other, the explorer's shapley value is statistically indistinguishable from that of a normal agent.

Finally, in Figure 2d, we consider a slightly different asymmetric instance. Each agent $a \in \{1, 2, 3, 4, 5\}$ has an identical set of actions $X_{a,t} = A_a$, with no action overlap among each other. And we introduce an asymmetric agent, labeled 0, who shares exactly one action with each of the other five agents. Specifically, $X_{0,t} = \{1, 7, 13, 19, 25\}$. It can be observed from Equation (45) that agent 0 shares with agent 1 their respective optimal arms, with agent 2 their respective second optimal arm, and so forth, and finally with agent 5 their respective least optimal arm.

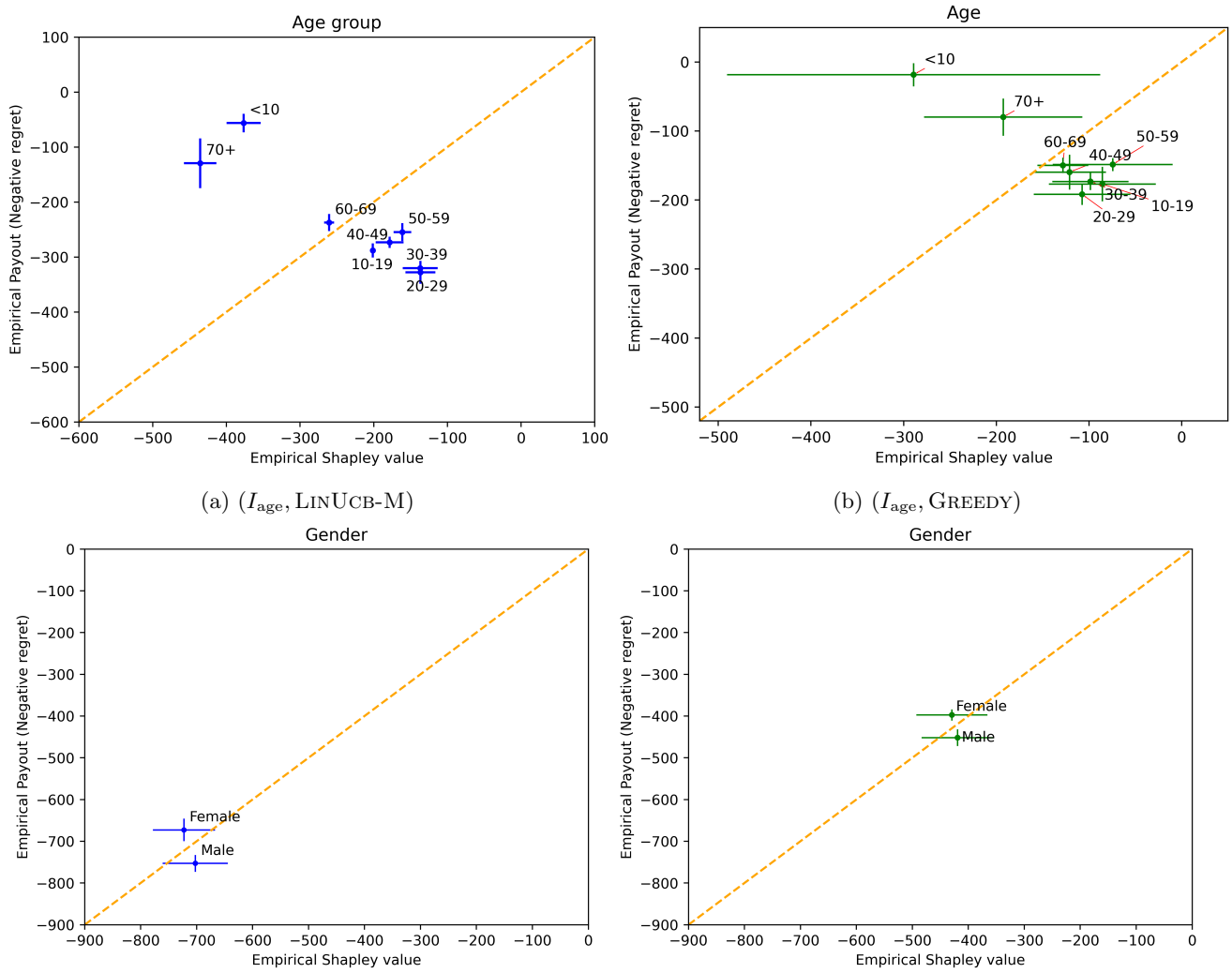
It is seen from Figure 2d that the empirical payouts and Shapley values are well correlated for the agents. The asymmetry in agents 1 to 5 offers an interesting insight. Agent 1 has the least Shapley and most regret. This indicates that sharing and receiving information about the optimal action is not of much use to himself or the receiving agent (which in this case is 0). The reason is that each agent could have very well explored this optimal arm by himself by incurring no regret instead of receiving from other agents. This phenomenon gradually lightens as we move through agents 2,3,4, and 5, who share lesser and lesser optimal arms, and have larger and larger Shapley values and payouts (negative regrets).

E.3 MovieLens Experiments

In this subsection, we present the experimental results on MovieLens problem instances generated by classifying by gender and age group. Note the main paper presents experiments on instances based on classification by occupation and geography.

Figure 3 draws similar insights to the ones mentioned in the main paper. The payouts are close to the Shapley values empirically for several agents, but there exist outlier agents who disproportionately gain or contribute to the other agents.

On another note, the greedy algorithm appears to outpeform the LinUCB, and we attribute this to the inherent



(a) (I_{age}, LinUcb-M)

(b) (I_{age}, Greedy)

(c) (I_{gen}, LinUcb-M)

(d) (I_{gen}, Greedy)

Figure 3: **MovieLens experiments** - Instances based on classification by age and gender.

heterogeneity of the action space (the user representations, here).

E.4 On satisfying Assumption 3

We check if the Assumption—that the regret of an agent doesn't increase if more agents join the coalitoin—holds for our MovieLens experiments. We observe that it is mostly satisfied, with some rare cases where it is not. We present the results for I_{occ} and I_{gen} problem instances with both algorithms LinUcb-M and Greedy.

In the I_{occ} instance, there are 2816 distinct agent-coalition pairs of $(a \in Q, Q \subseteq M)$ with $|M| = 8$. Among them, all but 3 (sim. 22) pairs satisfy the Assumption when run with algorithm LinUcb-M (sim. Greedy), and the details of the pairs that do *not* satisfy are plotted in Table 1 (sim. Table 2). All other pairs satisfy the Assumption and are not presented due to large volume of data. Read agents from $0-7$ in order ['student', 'technical', 'management', 'creative', 'academic', 'business', 'healthcare', 'non-professional'].

Next, we present the full result for I_{gen} instance with both algorithms. It can be seen that the Assumption is satisfied as in Tables 3 and 4. Read agents from $0-1$ in order ['Male', 'Female'].

Table 1: $(I_{\text{occ}}, \text{LinUcb-M})$

Coalition Q	**Agent** $a \in Q$	**Regret** R_a^Q	**Sub-coalition** S	**Reg.** R_a^S	$R_a^Q/R_a^S \geq 1$
(0, 1, 3, 4, 6, 7)	1	354 ± 18	(1, 3, 4, 6, 7)	352 ± 47	1.006
(2, 3, 4, 5, 6, 7)	5	208 ± 23	(3, 4, 5, 6, 7)	208 ± 14	1.0
(0, 1, 2, 3, 5, 6, 7)	5	189 ± 29	(1, 2, 3, 5, 6, 7)	182 ± 4	1.038

Table 2: $(I_{\text{occ}}, \text{Greedy})$

Coalition Q	**Agent** $a \in Q$	**Regret** R_a^Q	**Sub-coalition** S	**Reg.** R_a^S	$R_a^Q/R_a^S \geq 1$
(0, 2, 3, 6)	6	268 ± 37	(2, 3, 6)	252 ± 23	1.063
(0, 3, 6, 7)	7	266 ± 20	(3, 6, 7)	256 ± 27	1.039
(2, 3, 5, 7)	7	230 ± 15	(3, 5, 7)	226 ± 15	1.018
(0, 1, 2, 5, 7)	5	181 ± 25	(1, 2, 5, 7)	172 ± 13	1.052
(0, 1, 4, 5, 7)	7	195 ± 11	(1, 4, 5, 7)	176 ± 19	1.108
(0, 1, 4, 6, 7)	6	205 ± 49	(1, 4, 6, 7)	195 ± 21	1.051
(0, 2, 5, 6, 7)	5	204 ± 45	(2, 5, 6, 7)	184 ± 20	1.109
(0, 4, 5, 6, 7)	6	243 ± 13	(4, 5, 6, 7)	215 ± 24	1.13
(1, 2, 4, 5, 6)	5	193 ± 37	(2, 4, 5, 6)	176 ± 14	1.097
(1, 4, 5, 6, 7)	6	227 ± 20	(4, 5, 6, 7)	215 ± 24	1.056
(0, 1, 2, 3, 6, 7)	6	179 ± 25	(1, 2, 3, 6, 7)	165 ± 27	1.085
(0, 1, 3, 4, 5, 6)	6	190 ± 18	(1, 3, 4, 5, 6)	165 ± 5	1.152
(0, 1, 3, 5, 6, 7)	5	158 ± 10	(1, 3, 5, 6, 7)	152 ± 19	1.039
(0, 3, 4, 5, 6, 7)	7	173 ± 40	(3, 4, 5, 6, 7)	172 ± 17	1.006
(1, 2, 3, 4, 6, 7)	3	210 ± 19	(2, 3, 4, 6, 7)	198 ± 20	1.061
(1, 3, 4, 5, 6, 7)	5	165 ± 34	(3, 4, 5, 6, 7)	154 ± 19	1.071
(1, 3, 4, 5, 6, 7)	6	201 ± 10	(3, 4, 5, 6, 7)	184 ± 54	1.092
(1, 3, 4, 5, 6, 7)	7	174 ± 21	(3, 4, 5, 6, 7)	172 ± 17	1.012
(0, 1, 2, 3, 5, 6, 7)	6	165 ± 33	(1, 2, 3, 5, 6, 7)	145 ± 21	1.138
(0, 1, 2, 4, 5, 6, 7)	6	156 ± 26	(1, 2, 4, 5, 6, 7)	140 ± 19	1.114
(1, 2, 3, 4, 5, 6, 7)	2	162 ± 15	(2, 3, 4, 5, 6, 7)	162 ± 13	1.0
(1, 2, 3, 4, 5, 6, 7)	7	156 ± 18	(2, 3, 4, 5, 6, 7)	152 ± 19	1.026

Table 3: $(I_{\text{gen}}, \text{LinUcb-M})$

Coalition Q	**Agent** $a \in Q$	**Regret** R_a^Q	**Sub-coalition** S	**Reg.** R_a^S	R_a^Q/R_a^S
(0,)	0	1085 ± 84	()	NA ± NA	NA
(1,)	1	1106 ± 48	()	NA ± NA	NA
(0, 1)	0	752 ± 20	(0,)	1085 ± 84	0.693
(0, 1)	1	672 ± 27	(1,)	1106 ± 48	0.607

Table 4: $(I_{\text{gen}}, \text{Greedy})$

Coalition Q	**Agent** $a \in Q$	**Regret** R_a^Q	**Sub-coalition** S	**Reg.** R_a^S	R_a^Q/R_a^S
(0,)	0	673 ± 96	()	NA ± NA	NA
(1,)	1	683 ± 53	()	NA ± NA	NA
(0, 1)	0	451 ± 20	(0,)	673 ± 96	0.67
(0, 1)	1	397 ± 13	(1,)	683 ± 53	0.581