

Multiagent Low-Dimensional Linear Bandits

Ronshee Chawla [✉], Abishek Sankararaman [✉], and Sanjay Shakkottai [✉], *Fellow, IEEE*

Abstract—We study a multiagent stochastic linear bandit with side information, parameterized by an unknown vector $\theta^* \in \mathbb{R}^d$. The side information consists of a finite collection of low-dimensional subspaces, one of which contains θ^* . In our setting, agents can collaborate to reduce regret by sending recommendations across a communication graph connecting them. We present a novel decentralized algorithm, where agents communicate subspace indices with each other and each agent plays a projected variant of LinUCB on the corresponding (low dimensional) subspace. By distributing the search for the optimal subspace across users and learning of the unknown vector by each agent in the corresponding low-dimensional subspace, we show that the per-agent finite-time regret is much smaller than the case when agents do not communicate. We finally complement these results through simulations.

Index Terms—Decentralized learning, gossip, linear bandits, networks, regret minimization.

I. INTRODUCTION

THE multiarmed bandit (MAB) model features a single decision maker making sequential decisions under uncertainty. It has found a wide range of applications: advertising [1], information retrieval [2], and operation of data centers [3] to name a few. See also the books in [4] and [5]. As the scale of applications increases, several decision makers (a.k.a. agents) are involved in making repeated decisions as opposed to just a single agent. For example, in Internet advertising, multiple servers are typically deployed to handle the large volume of traffic [6]. Multiagent MAB models have emerged as a framework to design algorithms accounting for this large scale.

In recent times, there has been a lot of interest in the study of multiagent unstructured bandits [7]–[10]. However, from a practical perspective, the linear bandit framework has shown to

be more appropriate than unstructured bandits in many instances (e.g., recommendations [11] and clinical studies [12]). The linear bandit's framework allows for a continuum of arms with a shared reward structure, thereby modeling many complex online learning scenarios [13], [14]. Despite its applicability, the study of multiagent linear bandits is limited. The key technical challenge arises from the “information leakage”: the reward obtained by playing an arm gives information on the reward obtained by all other arms. In a multiagent scenario, this is further exacerbated, making the design of collaborative algorithms nontrivial.

We take a step in this direction by considering a collaborative multiagent low-dimensional linear bandit problem and propose a novel decentralized algorithm. Agents in our model have side information in the form of subspaces. In our algorithm, agents collaborate by sharing these subspaces as opposed to the linear reward in our algorithm. Our main result shows that, even with minimal communications, the regret of all agents are much lower compared with the case of no collaboration.

Model overview: Our problem consists of a single instance of a stochastic linear bandit with unknown parameter $\theta^* \in \mathbb{R}^d$, played concurrently by N agents. The common side information available to the agents is a collection of K disjoint m -dimensional subspaces, only one of which contains θ^* . However, the agents are not aware of the subspace containing θ^* . At each time t , each agent $i \in [N]^1$ chooses a subspace in $[K]$. Subsequently, it plays an action vector $a_t^{(i)}$ from the action set $\mathcal{A}_t \subset \mathbb{R}^d$ while satisfying the constraints imposed by the chosen subspace and receives a reward $\langle a_t^{(i)}, \theta^* \rangle + \eta_t^{(i)}$, where $\eta_t^{(i)}$ is zero mean sub-Gaussian noise. Thus, the abovementioned problem can be visualized as a two-tier bandit problem, described as follows. The first tier corresponds to the K arms of an unstructured bandit. In the second tier, each arm corresponds to solving the stochastic linear bandit problem (parameterized by unknown θ^*) over one of the K known subspaces. The rewards obtained by the agents are only dependent on their actions and independent of actions of other agents.

The agents in our model are connected through a communication graph over which they can exchange messages to collaborate. Agents are constrained to communicate by exchanging messages for a fixed number of times over any given time span. We seek decentralized algorithms for agents, i.e., the choice of action vector, communication choices, and messages depend only on the observed past history (of action vectors, rewards, and messages) of that agent.

Motivating example: We motivate our model in the context of personalized news recommendation systems. Suppose that a user u can be modeled by a (unknown) vector $\theta_u^* \in \mathbb{R}^d$, which

Manuscript received 15 May 2021; revised 20 February 2022; accepted 10 May 2022. Date of publication 1 June 2022; date of current version 26 April 2023. This work was supported in part by ONR under Grant N00014-19-1-2566, in part by NSF under Grant SATC 1704778, in part by ARO under Grant W911NF-17-1-0359, in part by NSA SoS Lablet under Grant H98230-18-D-0007, and in part by the WNCG Industrial Affiliates Program. Recommended by Senior Editor Tetsuya Iwasaki and Guest Editors George J. Pappas, Anuradha M. Annaswamy, Manfred Morari, Claire J. Tomlin, Rene Vidal, and Melanie N. Zeilinger. (Corresponding author: Ronshee Chawla.)

Ronshee Chawla and Sanjay Shakkottai are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: rcronshee@gmail.com; sanjay.shakkottai@utexas.edu).

Abishek Sankararaman is with the University of California at Berkeley, Berkeley, CA 94720 USA, and also with the Applied Scientist at Amazon AWS, Palo Alto, CA 94303 USA (e-mail: abishek@utexas.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3179521>.

Digital Object Identifier 10.1109/TAC.2022.3179521

¹ $[N]$ denotes the set $\{1, \dots, N\}$.

lies in one of K possible subspaces. These subspaces reflect information from historical data of other users whose feature vectors have been learned (for example, users that have been in the system for a long time) and can be categorized into a collection of low-dimensional subspaces. Thus, when any new user enters the system, it needs to 1) identify the subspace that the user's vector lies in, and 2) determine the corresponding θ_u^* for that user. At any point of time, each user is handled by a single server (agent). However, in large-scale applications, a collection of servers is deployed to handle the large number of users. Even though a user's queries are routed to different servers over time; however, these servers can collaborate by exchanging messages to speed up learning.

Elaborating on the news recommendation example above-mentioned, the subspaces could correspond to political leanings of the user (e.g., social liberal, fiscal conservative, and libertarian). In this model, all users with the same political leanings would share the same subspace; however, their personal vectors θ_u^* would differ (to capture fine-grained individual differences). The two-tier bandit, thus, models a coarse dimensionality reduction through the subspace choice and a finer characterization through θ_u^* in a specific low-dimensional subspace.

The above-mentioned discussion reflects the two-tier bandit from a single user's perspective; the system will run many parallel instances of this—one for each user. Our model abstracts this setup, and our algorithm demonstrates that agents (servers) can indeed benefit from collaborations with minimal communications.

Our main contributions are given as follows.

SubGoss algorithm: We propose SubGoss (see Algorithm 1), which proceeds in phases, such that agents in any phase

- 1) explore the subspaces repeatedly to identify the correct subspace containing θ^* , followed by
- 2) playing Projected LinUCB on that subspace, and
- 3) communicating that subspace whenever requested.

Our algorithm constrains agents to search for θ^* over only a small set (of cardinality $\leq \frac{K}{N} + 2$) of subspaces per agent. Agents use pairwise communications to recommend subspaces (not samples), i.e., agents communicate the ID of the estimated best subspace. This set of subspaces is updated through recommendations: agents accept new recommendations and drop subspace(s) unlikely to contain θ^* , ensuring that the total number of subspaces an agent considers at all times remain small. Agents can communicate $O(\log T)$ times over a span of time T . Nevertheless, the best one *spreads* to all the agents through communications and thus all agents eventually identify the correct subspace.

Asymptotically matching an oracle's regret rate in large systems: Despite playing from a time-varying set of subspaces, every agent incurs a regret of at most $O(m\sqrt{T} \log T) + O(\frac{K}{N} \cdot m\sqrt{T})$ (see Theorem 1). This scaling *does not depend*² on the gossip matrix G , and we show that these communication constraints only affect the constant term in regret.

Note that an oracle that knew the right subspace will only incur regret for finding θ^* in that subspace, while avoiding any regret

due to subspace search. We use this fact at the end of Section IV to informally argue that even if an agent gets the information about the correct subspace whenever it communicates with other agents, it cannot do better than $\Omega(m\sqrt{T})$ regret under our model of information sharing. Consequently, we show in Corollary 3 that for large K and N , SubGoss achieves near-optimal performance, demonstrating that it uses the side information effectively.

Finite-time gains due to faster search of subspaces with collaboration: We quantify the extent to which collaboration helps by analyzing the ratio of regret upper bound achieved by SubGoss without collaborations to that achieved with collaboration. We observe that the benefits occur from the ability of multiple agents to do a faster search for the right subspace containing θ^* as compared with a single agent.

In high-dimensional settings (when d is large and m is a constant) with large number of subspaces and agents ($K = N = O(d)$), we show in Corollary 3 (and the remarks following it) that by time $T = \Omega(d)$, the collaborative gain is of the order of $\Omega(\frac{d}{\log d})$. The key reason for the gain lies in the ability of multiple agents to distribute the search for the right subspace among them, enabling *all* agents to identify the subspace faster, compared with a single agent without collaboration. Finally, these results are corroborated through simulations (see Fig. 1).

Related work: Our work focuses on collaborative multi-agent bandits, where agents jointly accomplish the shared objective of minimizing cumulative regret [6], [9], [10], [15]–[19]. Our work focuses on a setting where agents only share recommendations of actions (e.g., to minimize network traffic due to collaboration, while optimizing for per-agent regret), and do not share the samples themselves [9], [15], [18]. In each of these studies, agents play from a small subset of arms at all times and exchange the arm IDs of what they consider the best arm in their playing set through pairwise gossip-style communications, which is further used to update their playing set. Another approach that focuses on reducing network traffic, while simultaneously optimizing for total cumulative regret (sum over time and users), is based on the follow-the-leader approach [19], wherein a leader among the agents is elected and subsequently becomes the sole player exploring the arms, while other agents act as its followers. However, all of the abovementioned works are adapted to the case of finite-armed unstructured MABs and cannot be applied to a linear bandit setup such as ours. Nevertheless, we adopt some of the broader principles from [9] and [15] regarding the use of the gossiping paradigm for communications to spread the best subspace into our algorithm design.

The stochastic linear bandit framework and the study of LinUCB algorithm was initiated in [13] and [14]. From a practical perspective, the linear bandit framework has been shown to be effective for various applications: for example, Li *et al.* [11] and Agarwal *et al.* [20] applied this framework in the context of Internet advertising and [12], [21] apply in the context of clinical trials. Furthermore, a projected version of LinUCB on low-dimensional subspaces has been recently studied in [22].

To the best of our knowledge, our model has not been studied before, even in a single-agent setting. Our model can be

²We require G to be connected. See Appendix A.

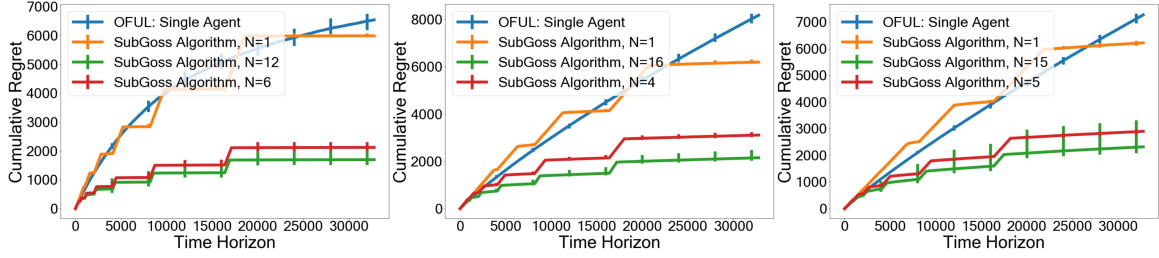


Fig. 1. Illustrating benefit of collaboration. (d, m, K) are $(24, 2, 12)$, $(48, 3, 16)$, and $(60, 4, 15)$, respectively.

viewed as a generalization of the well-studied model of sparse linear bandits [12], [23]–[25]. The sparse linear bandit problem assumes that the unknown vector θ^* is s -sparse, for some known sparsity level $s < d$. In other words, θ^* is assumed to lie in one of the $\binom{d}{s}$ subspaces, where each of these subspaces corresponds to a particular set of s coordinates, i.e., the subspaces are axis aligned. Our model is a generalization where θ^* lies in one of any K given arbitrary disjoint subspaces. The two main algorithmic ideas in sparse bandits are to either use heavy-tailed priors for sampling action vectors and the associated posterior distributions that result in sparse estimates [24], [25], or use a LASSO-type regularizer in the estimator [12]. We cannot use the techniques from sparse linear bandits in our model because even though the unknown θ^* lies in one of the low-dimensional subspaces, all of its d coordinates can have nonzero values. Consequently, the linear bandit suffers from the problem of “information sharing”: the reward obtained by playing an action vector in one subspace reveals information about the rewards of action vectors in other subspaces. Hence, algorithmic ideas from sparse bandits are not directly applicable in our setting.

The study of multiagent linear bandit framework has attracted a lot of attention lately [26]. Multiagent linear bandits have been studied in the context of clustering [27], differentially private federated learning [28], and safety-critical distributed learning [29]. However, all of these works involve agents sharing samples with each other in the absence of side information, unlike our setting where agents have the side information in the form of subspaces and communicate only subspace IDs with each other.

II. PROBLEM SETUP

Our problem setup consists of single instance of stochastic linear bandit (parameterized by unknown θ^*), concurrently played by N agents. All agents play from the same set of action vectors $\{\mathcal{A}_t\}_{t \in \mathbb{N}}$ at any time t , where $\mathcal{A}_t \subset \mathbb{R}^d$. The side information available to all the agents is a collection of K disjoint subspaces in $\mathbb{R}^d$ of dimension $m < d$. These subspaces are denoted by the $d \times m$ orthonormal matrices $\{U_i\}_{i=1}^K$, where $\text{span}(U_i)$ defines a m -dimensional subspace in $\mathbb{R}^d$. One of these subspaces contains θ^* , but agents are unaware of the subspace containing it. Without loss of generality, we assume that $\|\theta^*\|_2 \leq 1$ and K is an integral multiple of N . Let $P_k = U_k U_k^T$ denote the projection matrix of the subspace $\text{span}(U_k)$ for all $k \in [K]$. We also assume that the set of action vectors $\{\mathcal{A}_t\}_{t \in \mathbb{N}}$ contain the orthonormal basis vectors of all the K subspaces (which are columns of U_k for all $k \in [K]$) for all $t \in \mathbb{N}$.

At any time t , an agent i chooses a subspace in $[K]$. Subsequently, it plays an action vector $a_t^{(i)} \in \mathcal{A}_t$ while satisfying the constraints imposed by the chosen subspace and the reward obtained is given by $r_t^{(i)} := \langle a_t^{(i)}, \theta^* \rangle + \eta_t^{(i)}$. Here, $\eta_t^{(i)}$ is a zero mean sub-Gaussian noise, conditional on the actions and rewards accumulated only by agent i , i.e., for all $z \in \mathbb{R}$, $\mathbb{E}[\exp(z\eta_t^{(i)}) | \mathcal{F}_{t-1}^{(i)}] \leq \exp(\frac{z^2}{2})$ a.s. and $\mathcal{F}_{t-1}^{(i)} = \sigma(a_1^{(i)}, r_1^{(i)}, \dots, a_{t-1}^{(i)}, r_{t-1}^{(i)}, a_t^{(i)})$. The noise is independent across agents. Thus, the abovementioned setup can be abstracted as a two-tier bandit problem, where 1) the first tier corresponds to the K arms of an unstructured bandit, and 2) in second tier, each arm corresponds to solving the stochastic linear bandit problem over one of the K subspaces.

Collaboration among agents: Our model builds on gossip-based communication constraints for multiagent finite-armed unstructured bandits in [9] and [15]. The agents collaborate by exchanging messages over a communication network. This matrix is represented through an $N \times N$ gossip matrix G , with rows in this matrix being probability distributions over $[N]$. At each time step, after playing an action vector and obtaining a reward, agents can additionally choose to communicate with each other. Agent i , if it chooses to communicate, will do so with another agent $j \sim G(i, \cdot)$, chosen independently of everything else. However, for any time horizon T , the total number of times an agent can communicate is $O(\log T)$. Each time an agent chooses to communicate, it can exchange at most $\log_2 K + 1$ number of bits. Therefore, every agent communicates $O(\log K \cdot \log T)$ bits over a time horizon of t , for all t .

Decentralized Algorithm: Each agent’s decisions (arm play and communication decisions) in the algorithm depend only on its own history of plays and observations, along with the recommendation messages that it has received from others.

Performance metric: Each agent plays action vectors in order to minimize their individual cumulative regret. At any time t , the instantaneous regret for an agent i is given by $w_t^{(i)} = \langle \theta^*, a_t^* \rangle - \langle \theta^*, a_t^{(i)} \rangle$, where $a_t^* = \arg \max_{a \in \mathcal{A}_t} \langle \theta^*, a \rangle$. The expected cumulative regret for any agent $i \in [N]$ is given by $\mathbb{E}[R_T^{(i)}] := \mathbb{E}[\sum_{t=1}^T w_t^{(i)}]$, where the expectation is with respect to the σ -field generated by action vectors and rewards of agent i up to and including time T .

III. SUBGOSS ALGORITHM

Key ideas and intuition: Our setting considers that the unknown θ^* lies in one of a large number of (low dimensional)

subspaces. In our approach, agents at any time instant identify a small *active set of subspaces* (cardinality $\leq K/N + 2$) and play actions only within this set of subspaces (however, this set is time-varying). At each point of time, an agent first *identifies* among its current active set of subspaces the one likely to contain θ^* . It subsequently plays a projected version of LinUCB on this identified subspace. The communication medium is used for *recommendations*; whenever an agent is asked for information, it sends as message the subspace ID that it thinks most likely contains θ^* , which is then used to update the active set of the receiving agent. Thus, an agent's algorithm has two time-interleaved parts: 1) updating active sets through collaboration, which is akin to a distributed best-arm identification problem, and 2) determining the optimal θ^* from within its active set of subspaces, an estimation problem in low dimensions, similar to the classical linear bandit.

SubGoss algorithm is organized in phases with the active subspaces for each agent fixed during the phase. Within a phase, all agents solve two tasks: 1) identify the most likely subspace among its active subspaces to contain θ^* , and 2) within this subspace, play actions optimally to minimize regret. The first point is accomplished by agents through pure exploration. In pure exploration, agents play the orthonormal basis vectors of all the subspaces in their respective active sets in a round-robin fashion. Agents minimize their regret during pure exploration by considering a *small* active set of subspaces (of cardinality $\leq K/N + 2$) at all times; otherwise, agents play action vectors within their best estimated subspace containing θ^* to minimize regret. This step is achieved by playing a projected version of the LinUCB algorithm. The second step only incurs regret in the dimension of the subspace (once the true subspace is correctly identified) as opposed to the ambient dimension, thereby keeping regret low. Due to communications, the correct subspace *spreads* to all agents, while playing from a small active set of active subspaces at all times (and thus reducing the regret due to explorations).

A. Description

SubGoss algorithm builds on some of the ideas developed for a (noncontextual) collaborative setting for unstructured bandits in [9]. We fix an agent $i \in [N]$ for ease of exposition. SubGoss proceeds in phases, where phase $j \in \mathbb{N}$ is from time slots $\sum_{l=1}^{j-1} \lceil b^{l-1} \rceil + 1$ to $\sum_{l=1}^j \lceil b^{l-1} \rceil$, both inclusive, where $b > 1$ is given as an input. During each phase j , agent i only plays from an active set $S_j^{(i)} \subset [K]$ of subspaces such that $|S_j^{(i)}| \leq (K/N) + 2$. Agents communicate at the end of the phase to update their active set. Note that the phase length is $\lceil b^{j-1} \rceil$, which satisfies the communication constraint of $O(\log T)$ communications for any time horizon T .

Initialization: At the beginning of the algorithm, every agent is assigned a *sticky* set of subspaces, by partitioning the subspaces equally across agents

$$(\widehat{S}^{(i)})_{i=1}^N = \left\{ (i-1)\frac{K}{N} + 1, \dots, i\frac{K}{N} \right\}. \quad (1)$$

We set the initial active set $S_1^{(i)} = \widehat{S}^{(i)}$.

Action vectors chosen in a phase: We play the following two subroutines in every phase $j \in \mathbb{N}$ in the order as described.

- 1) *Explore:* In this subroutine, for every $k \in S_j^{(i)}$, we play the orthonormal basis vectors of the subspace $\text{span}(U_k)$ (which are the columns of U_k) in a round robin fashion for $8m \lceil b^{\frac{j-1}{2}} \rceil$ times.

Let $\tilde{n}_{k,j}^{(i)}$ denote the number of times subspace $\text{span}(U_k)$ has been explored by agent i up to and including phase j . After executing the explore subroutine in a phase, agent i calculates the least squares estimates $\tilde{\theta}_{k,j}^{(i)}$ for every $k \in S_j^{(i)}$ by using only the explore samples of the subspace $\text{span}(U_k)$ up to and including phase j . Mathematically, $\tilde{\theta}_{k,j}^{(i)} = \arg \min_{\theta \in \mathbb{R}^d} \|(\tilde{A}_{k,\tilde{n}_{k,j}^{(i)}}^{(i)})^T \theta - \tilde{\mathbf{r}}_{k,\tilde{n}_{k,j}^{(i)}}^{(i)}\|_2$, where $\tilde{A}_{k,\tilde{n}_{k,j}^{(i)}}^{(i)}$ is a $d \times \tilde{n}_{k,j}^{(i)}$ matrix whose columns are the explore action vectors of the subspace $\text{span}(U_k)$ played up to and including phase j , and $\tilde{\mathbf{r}}_{k,\tilde{n}_{k,j}^{(i)}}^{(i)}$ is a column vector of the corresponding rewards. It is worth noticing that $\tilde{\theta}_{k,j}^{(i)}$ is the estimate of the vector $P_k \theta^*$ (details in the proof of Lemma 4), which is the projection of the unknown vector θ^* in the subspace $\text{span}(U_k)$. We will describe in the proof sketch (see Section IV-B) that this observation is crucial to finding the subspace containing θ^* .

- 2) *Projected LinUCB:* Let $\widehat{\mathcal{O}}_j^{(i)} = \arg \max_{k \in S_j^{(i)}} \|\tilde{\theta}_{k,j}^{(i)}\|_2$. For the remainder of the phase j , agent i chooses the action vector according to the Projected LinUCB [22], played on the subspace $\text{span}(U_{\widehat{\mathcal{O}}_j^{(i)}})$. We set $k = \widehat{\mathcal{O}}_j^{(i)}$ for reducing the clutter while describing Projected LinUCB. For all $\sum_{l=1}^{j-1} \lceil b^{l-1} \rceil + 8m |S_j^{(i)}| \lceil b^{\frac{j-1}{2}} \rceil < t \leq \sum_{l=1}^j \lceil b^{l-1} \rceil$, where t denotes the corresponding time instants after the end of explore subroutine in phase j , let $n_{k,t}^{(i)}$ denote the number of times agent i has played Projected LinUCB on the subspace $\text{span}(U_k)$ up to time t . The action vector chosen is given according to the following equations [22]:

$$a_t^{(i)} \in \arg \max_{a \in \mathcal{A}_t} \max_{\theta \in \mathcal{C}_{k,t}^{(i)}} \langle \theta, P_k a \rangle, \text{ where}$$

$$\mathcal{C}_{k,t}^{(i)} = \left\{ \theta \in \mathbb{R}^d : \|\widehat{\theta}_t^{(i)} - \theta\|_{\bar{V}_{k,t}(\lambda)^{(i)}} \leq \beta_{t,\delta} \right\}$$

$$\beta_{t,\delta} = \sqrt{\lambda} + \sqrt{2 \log \frac{1}{\delta} + m \log(1 + \frac{n_{k,t}^{(i)}}{\lambda m})}, \quad \bar{V}_{k,t}(\lambda)^{(i)} = P_k (A_{k,t-1}^{(i)} (A_{k,t-1}^{(i)})^T + \lambda I_d) P_k, \quad \text{and} \quad \widehat{\theta}_{k,t-1}^{(i)} = \arg \min_{\theta \in \mathbb{R}^d} \| (P_k A_{k,t-1}^{(i)})^T \theta - \mathbf{r}_{k,t-1}^{(i)} \|_2^2 + \lambda \| P_k \theta \|_2^2.$$

$A_{k,t-1}^{(i)}$ is a $d \times n_{k,t}^{(i)}$ matrix whose columns are the Projected LinUCB action vectors played only on the subspace $\text{span}(U_k)$ up to time t , and $\mathbf{r}_{k,t-1}^{(i)}$ is a column vector of the corresponding rewards.

Communications and the active subspaces for the next phase: After phase j gets over, agent i asks for a subspace recommendation from an agent $J \sim G(i, \cdot)$ chosen independently. Denote

Algorithm 1: SubGoss Algorithm (at Agent i).

1: **Input:** K disjoint m -dimensional subspaces $\{U_l\}_{l=1}^K$, $b > 1$, regularization parameter $\lambda > 0$, $\delta \in (0, 1)$.

2: **Initialization:** $\hat{S}^{(i)}, S_1^{(i)}$ [Equation (1)], $j \leftarrow 1$.

3: **while** phase $j \geq 1$ **do**

4: EXPLORE: For each $k \in S_j^{(i)}$, play the orthonormal basis vectors of the subspace ID k in a round robin fashion for $8m \lceil b^{\frac{j-1}{2}} \rceil$ times.

5: Calculate the least squares estimate $\tilde{\theta}_{k,j}^{(i)}$ for each $k \in S_j^{(i)}$ after running the EXPLORE by using only its explore samples collected thus far.

6: $\hat{\mathcal{O}}_j^{(i)} \leftarrow \arg \max_{k \in S_j^{(i)}} \|\tilde{\theta}_{k,j}^{(i)}\|_2$.

7: PROJECTED LINUCB: For the remainder of the phase j , play the Projected LinUCB [22] on the subspace ID $\hat{\mathcal{O}}_j^{(i)}$ by using only its Projected LinUCB samples collected thus far.

8: At the end of phase j , sample an agent from the gossip matrix $ag \sim G(i, \cdot)$ for receiving subspace recommendation.

9: Get the subspace recommendation $\mathcal{O}_j^{(i)} \leftarrow \arg \max_{k \in S_j^{(ag)}} \|\tilde{\theta}_{k,j}^{(ag)}\|_2$.

10: Active set update for the next phase:

11: **if** $\mathcal{O}_j^{(i)} \in S_j^{(i)}$ **then**

12: $S_{j+1}^{(i)} \leftarrow S_j^{(i)}$.

13: **else**

14: **if** $|S_j^{(i)}| < \frac{K}{N} + 2$ **then**

15: $S_{j+1}^{(i)} \leftarrow S_j^{(i)} \cup \mathcal{O}_j^{(i)}$.

16: **else if** $|S_j^{(i)}| = \frac{K}{N} + 2$ **then**

17: $\mathcal{B}_j^{(i)} \leftarrow \arg \max_{k \in S_j^{(i)} \setminus \hat{S}^{(i)}} \|\tilde{\theta}_{k,j}^{(i)}\|_2$.

18: $S_{j+1}^{(i)} \leftarrow \hat{S}^{(i)} \cup \mathcal{B}_j^{(i)} \cup \mathcal{O}_j^{(i)}$.

19: **end if**

20: **end if**

21: $j \leftarrow j + 1$.

22: **end while**

by $\mathcal{O}_j^{(i)} \in [K]$ to be this recommendation. Agent i , if asked for a recommendation at the end of phase j , recommends the subspace ID $\hat{\mathcal{O}}_j^{(i)}$, i.e., using only the explore samples. The next active set is constructed as follows:

- 1) if $\mathcal{O}_j^{(i)} \in S_j^{(i)}$, the active set remains unchanged,
- 2) if $\mathcal{O}_j^{(i)} \notin S_j^{(i)}$ and $|S_j^{(i)}| < \frac{K}{N} + 2$, then $S_{j+1}^{(i)} := S_j^{(i)} \cup \mathcal{O}_j^{(i)}$, and
- 3) if $\mathcal{O}_j^{(i)} \notin S_j^{(i)}$ and $|S_j^{(i)}| = \frac{K}{N} + 2$, then $S_{j+1}^{(i)} := \hat{S}^{(i)} \cup \mathcal{B}_j^{(i)} \cup \mathcal{O}_j^{(i)}$, where $\mathcal{B}_j^{(i)} = \arg \max_{k \in S_j^{(i)} \setminus \hat{S}^{(i)}} \|\tilde{\theta}_{k,j}^{(i)}\|_2$.

Observe that $\hat{S}^{(i)} \subseteq S_j^{(i)} \forall j \in \mathbb{N}$, and thus, $\hat{S}^{(i)}$ is called sticky. Moreover, the update step along with the initialization $S_1^{(i)} = \hat{S}^{(i)}$ also ensures that $|S_j^{(i)}| \leq \frac{K}{N} + 2$ for all phases $j \in \mathbb{N}$.

Please see Algorithm 1 for the pseudocode of the SubGoss algorithm.

Remarks:

- 1) Until phase τ_0 (defined in Theorem 1), the duration of the explore subroutine exceeds $\lceil b^{j-1} \rceil$. In order to make less noisy subspace recommendations until phase τ_0 , the exploration is equally distributed across all the subspaces in $S_j^{(i)}$ for the entire duration of the phase.
- 2) *Choice of $\mathcal{C}_{k,t}^{(i)}$ while playing Projected LinUCB*—The construction and analysis of the confidence set $\mathcal{C}_{k,t}^{(i)}$ is formally described in Theorem 7 in the Appendix. The confidence set is an ellipsoid in the subspace on which Projected LinUCB is played. It is constructed such that: a) it contains θ^* with high probability, and (b) it shrinks in size as the correct sequence of action vectors is played with time.
- 3) *Choice of $a_t^{(i)}$ and its computational complexity while playing Projected LinUCB*—Analogous to the upper confidence bound (UCB) for classical K -armed bandits, an agent playing Projected LinUCB calculates an upper bound for the reward obtained for every $a \in \mathcal{A}_t$ and plays the action vector that maximizes the upper bound. This can be observed for the case when the action set $\mathcal{A}_t$ is finite, given as follows, for a fixed $a \in \mathcal{A}_t$

$$\begin{aligned}
\langle \theta, P_k a \rangle &= \langle \theta - \hat{\theta}_{k,t-1}^{(i)}, P_k a \rangle + \langle \hat{\theta}_{k,t-1}^{(i)}, P_k a \rangle \\
&\leq \|\theta - \hat{\theta}_{k,t-1}^{(i)}\|_{\bar{V}_{k,t}(\lambda)^{(i)}} \cdot \|P_k a\|_{(\bar{V}_{k,t}(\lambda)^{(i)})^\dagger} \\
&\quad + \langle \hat{\theta}_{k,t-1}^{(i)}, P_k a \rangle \\
&\leq \beta_{t,\delta} \|P_k a\|_{(\bar{V}_{k,t}(\lambda)^{(i)})^\dagger} + \langle \hat{\theta}_{k,t-1}^{(i)}, P_k a \rangle
\end{aligned}$$

where the first inequality is obtained by applying Hölder's inequality, and the second inequality follows by using the definition of $\mathcal{C}_{k,t}^{(i)}$. Therefore, for finite action sets, Projected LinUCB plays the action vector

$$a_t^{(i)} = \arg \max_{a \in \mathcal{A}_t} \langle \hat{\theta}_{k,t-1}^{(i)}, P_k a \rangle + \beta_{t,\delta} \|P_k a\|_{(\bar{V}_{k,t}(\lambda)^{(i)})^\dagger}. \quad (2)$$

The first term is the empirical estimate of the reward of the action a , and the second term corresponds to the deviation around that estimate, similar to the UCB value in K -armed bandits. The computational complexity of determining $a_t^{(i)}$ depends on the computational complexity of calculating $(\bar{V}_{k,t}(\lambda)^{(i)})^\dagger$, $\hat{\theta}_{k,t-1}^{(i)}$ and the inner products in (2).

IV. MAIN RESULT

In order to state the result, we assume that the gossip matrix G is connected (detailed definition in Appendix A). We define a random variable $\tau_{\text{spr}}^{(G)}$ denoting the spreading time of the following process: node i initially has a rumor; at each time, an agent without a rumor calls another chosen independently from the gossip matrix G and learns the rumor if the other agent knows the rumor. The stopping time $\tau_{\text{spr}}^{(G;i)}$ denotes the first time when

agent i knows the rumor, and $\tau_{\text{spr}}^{(G)} = \max_{i \in [N]} \tau_{\text{spr}}^{(G;i)}$ is the time by which all agents know the rumor. For ease of exposition, we assume that $\theta^* \in \text{span}(U_1)$, which the agents are unaware of and $\|a\|_2 \leq 1$ for all $a \in \cup_{t=1}^T \mathcal{A}_t$. Let $\Delta = \min_{k \in [K]: k \neq 1} \Delta_k$, where $\Delta_k = \|P_1 \theta^* - P_k \theta^*\|_2$.

Theorem 1: Consider a system consisting of N agents connected by a gossip matrix G , all running SubGoss algorithm with K disjoint m -dimensional subspaces and input parameters $b > 1$, $\lambda \geq 1$, $\delta = \frac{1}{T}$. Then, the expected cumulative regret of any agent $i \in [N]$ after time $T \in \mathbb{N}$ is bounded by

$$\begin{aligned} \mathbb{E}[R_T^{(i)}] \leq & \underbrace{\sqrt{8mT\beta_T^2 \log\left(1 + \frac{T}{m\lambda}\right)} + 2}_{\text{Projected LinUCB Regret}} \\ & + \underbrace{16m\left(\frac{K}{N} + 2\right) \log_b(h_{b,T}) + 16m\left(\frac{K}{N} + 2\right) \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1}}_{\text{Cost of subspace exploration}} \\ & + \underbrace{2g(b) \left([b^{2\tau_0}] + \frac{48b^3}{\log b} \cdot \frac{m^4 N}{\Delta^6} + b\mathbb{E}[b^{2\tau_{\text{spr}}^{(G)}}] \right)}_{\text{Constant cost of pairwise communications}}, \end{aligned} \quad (3)$$

where $\beta_T = \sqrt{\lambda} + \sqrt{2 \log T + m \log(1 + \frac{T-1}{\lambda m})}$, $\tau_0 = \min\{j \in \mathbb{N} : \forall j' \geq j, [b^{j'-1}] \geq 8m(\frac{K}{N} + 2)[b^{\frac{j'-1}{2}}]\}$, $g(b) = (\frac{1}{b-1} + \frac{1}{\log b})$, and $h_{b,T} = b(1 + (T-1)(b-1))$.

Remarks:

- 1) Proposition 5 shows that $\tau_0 \leq 2 \log_b(16m(\frac{K}{N} + 2)) + 1$, and thus, the term $b^{2\tau_0}$ in the constant cost of pairwise communications scales as $O((m \cdot \frac{K}{N})^4)$.
- 2) *Single agent running SubGoss*—In the case of no communication, when a single agent runs Algorithm 1 (without requiring communication graph G), it incurs a higher regret due to subspace exploration (which scales as $O(Km\sqrt{T})$ instead of $O((K/N)m\sqrt{T})$ in the multi-agent case), because it has to search through all the K subspaces to find the subspace containing θ^* . We express this result formally in Theorem 6, which is given in Appendix C.
- 3) Setting $\delta = \frac{1}{T}$ in Theorem 1 requires the knowledge of time horizon in SubGoss algorithm to achieve the corresponding regret guarantee. However, this is not a problem, as a fixed value of the confidence parameter $\delta \in (0, 1)$ achieves the same regret scaling as in Theorem 1 with high probability, which can be proved in a similar manner. Thus, the insights that can be obtained from our results are unaffected by the knowledge of time horizon.
- 4) *Subspace recommendation quality versus network spread*—Observe that $b > 1$ is an input to the algorithm, where agents communicate for the l th time after playing $[b^{l-1}]$ number of times since the last communication. Thus, increasing b will decrease the total number of communications between agents. Theorem 1 shows that there exists an optimal $b^* > 1$, such that

$b^* = \arg \min_{b > 1} \mathbb{E}[R_T^{(i)}]$. This can be seen by observing that as b decreases toward 1, the time between two communication instants reduces. However, each communication is based on fewer samples and, thus, subspace recommendations are noisy. On the other hand, as b becomes large, each recommendation is based on large number of samples and, thus, less noisy. The number of communications, however, is much lower, leading to a large time for the best subspace to spread. The optimal b^* tradeoffs between these two competing effects.

A. Impact of Network Structure on Regret

We can obtain the dependence of regret bound on network-related parameters by expressing the term $\mathbb{E}[b^{2\tau_{\text{spr}}^{(G)}}]$ in terms of the conductance ϕ of the gossip matrix (graph) G . In order to do so, we use a result obtained in [9, Corollary 17], which we reproduce here.

For a d -regular³ graph with adjacency matrix A_G , conductance ϕ , and gossip matrix $G = d^{-1}A_G$, $\mathbb{E}[b^{2\tau_{\text{spr}}^{(G)}}] \leq b^{\frac{2C \log N}{\phi}}$ for all $b \leq \exp(\frac{\phi}{C})$, where C is a universal constant.

Using the previous result, we now consider an illustrative example in which we assume that the agents are connected by a complete graph, i.e., $G(i, j) = \frac{1}{N-1}$ for $j \neq i$, 0, otherwise. In this case, it is easy to see that for all N and $b \leq \exp(\frac{N}{2(N-1)C})$ (where C is an universal constant), $\mathbb{E}[b^{2\tau_{\text{spr}}^{(G)}}] \leq \alpha N^{2(\log_2 b + \log b)}$ for some constant $\alpha > 0$, independent of N (where we substitute the conductance $\phi = \frac{N}{2(N-1)}$ for the complete graph in the result stated in the previous paragraph). This is because for the complete graph, $\tau_{\text{spr}}^{(G)} \leq \log_2 N + \log N$ with high probability. Corollary 2 quantifies the impact of underlying network on regret scaling.

Corollary 2: Suppose the agents are connected by a complete graph and $b = \min\{\exp(\frac{\log 2}{1+\log 2} \cdot \frac{1}{2}), \exp(\frac{N}{2(N-1)C})\}$, i.e., $\log_2 b + \log b \leq \frac{1}{2}$. With the same assumptions for λ and δ as in Theorem 1, the regret scaling of any agent $i \in [N]$ after playing SubGoss algorithm for T time steps is given by

$$\begin{aligned} \mathbb{E}[R_T^{(i)}] \leq & \underbrace{O(m\sqrt{T} \log T)}_{\text{Projected LinUCB Regret}} + \underbrace{O\left(\frac{K}{N} m\sqrt{T}\right)}_{\text{Cost of subspace exploration}} \\ & + \underbrace{O(N)}_{\mathbb{E}[b^{2\tau_{\text{spr}}^{(G)}}]} + O\left(\left(m \cdot \frac{K}{N}\right)^4\right) + O\left(\frac{m^4 N}{\Delta^6}\right) \end{aligned} \quad (4)$$

where $C > 0$ is an universal constant.

In (4), the $O(\cdot)$ notation only hides input constants and universal constants. It is evident from (4) that the network structure does not affect the time scaling in regret.

B. Proof Sketch

We provide the proof of Theorem 1 in Appendix B; however, we summarize its salient ideas here. Similar to the phenomenon

³Standard graph-theoretic notion that has nothing to do with ambient dimension d in $\mathbb{R}^d$.

in unstructured bandits [9], we prove in Proposition 1 that in our linear bandit setup, after a *freezing phase* τ , all agents have the correct subspace containing θ^* . Consequently, for all phases $j > \tau$, all agents will play Projected LinUCB [22] from the correct subspace in the exploit time instants and recommend it at the end of the phase. Therefore, the set of subspaces every agent has does not change after phase τ , and the regret after phase τ can be decomposed into regret due to pure exploration and regret due to Projected LinUCB (see Proposition 2).

The technical novelty of our proof lies in bounding the regret till phase τ , i.e., $\mathbb{E}[\tau + \frac{b^\tau - 1}{b - 1}]$ (see Proposition 3) and in particular showing it to be finite. This follows from two key observations arising from pure exploration in the explore time steps. First, for any agent $i \in [N]$ and subspace ID $k \in S_j^{(i)}$, the estimate $\tilde{\theta}_{k,j}^{(i)}$ of θ^* , using only its explore samples up to and including phase j , concentrates to $P_k \theta^*$ in l_2 norm (see Lemma 4). Therefore, for any subspace ID k , $\|\tilde{\theta}_{k,j}^{(i)}\|_2$ concentrates to $\|P_k \theta^*\|_2$, and eventually, the correct subspace $\text{span}(U_1)$ will achieve the largest value of $\|\tilde{\theta}_{k,j}^{(i)}\|_2$ among all subspaces, if present in the active set. Subsequently, we use the previous fact in Lemma 5 to show that if an agent has the correct subspace containing θ^* in a phase, the probability that it will not be recommended and, hence, dropped at the end of the phase is small.

Combining these two observations, we establish that after a random phase denoted by $\hat{\tau}_{\text{stab}}$ in Appendix B, satisfying $\mathbb{E}[\hat{\tau}_{\text{stab}}] < \infty$, agents never recommend incorrectly at the end of a phase and, thus, play the Projected LinUCB on the correct subspace in the exploit time instants of a phase. To conclude, after random phase $\hat{\tau}_{\text{stab}}$, the spreading time can be coupled with that of a standard rumor spreading [30], as once an agent learns the correct subspace, it is not dropped by the agent. This final part is similar to the one conducted for unstructured bandits in [9], giving us the desired bound on $\mathbb{E}[\tau + \frac{b^\tau - 1}{b - 1}]$.

Remark (Freezing time): The freezing phase τ is a quantity only showing up in the analysis, but is not part of the algorithm. In fact, the algorithm needs all agents to explore and communicate in all phases indefinitely, because τ is a sample-path dependent quantity. Indeed, any bandit algorithm that can achieve sublinear cumulative regret in the stochastic setting inherently has such a freezing time with finite expectation (including in the classical single-agent K -armed bandit); beyond this time, the best arm is identified with high probability. This can be shown by noting that sublinear regret implies that the probability the best arm is not played at time t , decreases to 0 as t goes to infinity. The finite freezing time follows from the simple Hoeffding inequality and Borel–Cantelli lemma. However, this is not useful in the algorithm but serves only as a proof technique. Formally, this random time τ is not a stopping time, and cannot be determined in an online fashion. Moreover, despite the existence of such a freezing time, the lower bounds for regret increase with the time horizon, showing that infinite exploration is necessary.

Remark (Technical differences w.r.t. [9]): The algorithms in [9] and the SubGoss Algorithm appear similar, because of the correspondence between the subspaces in our setup and the arms in a K -armed bandit. However, this correspondence is superficial, because unlike an arm, a subspace represents a continuum of actions, instead of just being an action. In order

to quantify the reward corresponding to a subspace, one has to form an estimate of θ^* in that subspace by playing the sequence of action vectors spanning that subspace.

Furthermore, any given phase in SubGoss bears a superficial resemblance to the *explore-then-commit* (ETC) algorithm, wherein the explore part of the phase that identifies the subspace to commit to in the exploit part, analogous to best arm identification in the standard K -armed bandit. However, the ETC algorithm requires the knowledge of lower bound of arm mean gaps as an input. In our model, this translates to agents needing knowledge of the distance between subspaces (denoted by Δ_k for all $k \neq 1$), which requires knowledge of θ^* . We circumvented this through a phased approach with exponentially increasing lengths, where each phase has an explore part and a commit part. The phases with exponentially increasing lengths ensure that

- 1) the probability of picking a subspace not containing θ^* decreases with every phase,
- 2) the increasing duration of playing Projected LinUCB within a phase as the phases progress minimizes the cumulative regret, and
- 3) does not need knowledge of gap between subspaces.

The consequence of not knowing θ^* is why agents need to continually explore in the explore part of every phase, as opposed to only exploring once in the beginning.

Remark (Discussion on a lower bound): We provide a brief discussion about the fundamental limits of our model (in terms of cumulative regret) to evaluate the effectiveness of SubGoss algorithm. We conjecture that the regret of $\Omega(m\sqrt{T})$ is unavoidable.

The above-mentioned claim can be argued as follows. In our model, an agent can exchange at most $\log_2 K + 1$ number of bits each time it chooses to communicate. One can consider the scenario in which whenever an agent decides to pull subspace recommendation (as a subspace ID in $\{1, \dots, K\}$ can be perfectly described by $\log_2 K + 1$ number of bits) from another agent based on gossip matrix G , suppose it always receives the ID of the correct subspace containing θ^* . In that case, the agent does not have to incur any regret in finding the correct subspace. However, given that there is no sample (action vectors and corresponding rewards) sharing possible between the agents, an agent will still have to search for θ^* in the correct subspace by itself. From [4, Ch. 24], we know that finding θ^* in $\mathbb{R}^d$ without any side information results in $\Omega(d\sqrt{T})$ regret. Given that the subspaces are m -dimensional, we can replace d with m in the previous statement and conclude that finding θ^* in the correct subspace incurs a regret of $\Omega(m\sqrt{T})$.

However, formalizing this argument requires surmounting some technical challenges. First, we need to precisely define the space of allowed communication policies without prescribing the content of the messages, for example, those that communicate at most a fixed number of bits at each time instant and total number of bits that scales as the logarithm of the time horizon. Once this is done, we need to establish that no communication policy under this constraint can encode knowledge of the true underlying θ^* to small enough precision, and show that communicating information other than subspace indices does not yield regret reduction. While the preceding paragraph

provides a plausible intuition for the $\Omega(m\sqrt{T})$ lower bound, a detailed argument is left to future work.

V. BENEFIT OF COLLABORATION IN HIGH DIMENSIONS

In this section, we illustrate how collaboration aids in reducing regret for each agent in the high-dimensional setting. We quantify this by computing the ratio of the regret upper bound achieved by SubGoss Algorithm without collaboration to that achieved with collaboration for any agent, denoted by $r_C(T)$. The *high-dimensional* setting corresponds to a large d , m a constant, and K and N scaling linear in d (system with a large number of agents).

Corollary 3: Consider a high-dimensional system where N agents are connected by a complete graph with $K = N = \frac{d}{m}$, where d is a multiple of m and m is a constant. Assume that $d \geq 3m$. With the choice of b as in Corollary 2,

1) for any agent $i \in [N]$, the regret with collaboration scales as $O(m\sqrt{T} \log T)$, and

2) $r_C(T) = \frac{r_S(T)}{r_M(T)}$, where

$$\begin{aligned} r_S(T) &= \sqrt{8mT\beta_T^2 \log\left(1 + \frac{T}{m\lambda}\right)} + 2 + 16d \log_b(h_{b,T}) \\ &\quad + 16d \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1} + 2g(b) \\ &\quad \times \left(\lceil b(16d)^2 \rceil + \frac{8b^2}{\log b} \cdot \frac{m^2}{\Delta^2} \right) \\ r_M(T) &= \sqrt{8mT\beta_T^2 \log\left(1 + \frac{T}{m\lambda}\right)} + 2 + 48m \log_b(h_{b,T}) \\ &\quad + 48m \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1} + 2g(b) \left(\lceil b^2(48m)^4 \rceil \right. \\ &\quad \left. + \frac{48b^3}{\log b} \cdot \frac{m^3d}{\Delta^6} + \frac{\alpha d}{m} \right). \end{aligned}$$

Proof: The proof follows by substituting $K = N = \frac{d}{m}$ in Theorems 1 and 6, along with the bound for spreading time $\mathbb{E}[b^2\tau_{\text{spr}}^{(G)}]$ from Corollary 2. ■

The following observations can be deduced from point (b) of Corollary 3:

- 1) when $T = \Theta(d^{1+\gamma})$ for all $\gamma \geq 0$, $r_C(T) = \Omega(\frac{d}{\log d})$;
- 2) when $T = \Theta(e^{d^\beta})$ for all $\beta \in (0, 1)$, $r_C(T) = \Omega(d^{1-\beta})$;
- 3) when $T = \Omega(e^d)$, $r_C(T) = \Omega(1)$.

Two remarks are now in order.

Remarks:

- 1) *Matching an oracle's regret rate, asymptotically:* Corollary 3 shows the power of collaboration in a large multi-agent system, as the regret scaling for any agent $i \in [N]$ matches that of a genie who is already aware of the subspace containing θ^* and can play Projected LinUCB on that subspace [22]. This demonstrates that the cost of subspace search can be amortized across agents and

only contributes a lower order term in regret, despite agents communicating infrequently (a total of $O(\log T)$ number of pairwise communications by every agent) and exchanging a limited number of bits in each communication (no sample sharing). Furthermore, the discussion in Section IV implies that in the absence of sample sharing between agents, an agent will incur $\Omega(m\sqrt{T})$ regret for finding θ^* in the correct subspace. Thus, SubGoss Algorithm is *near-optimal* even in high-dimensional settings with large number of subspaces and agents.

- 2) *Finite-time gains due to faster search of subspaces with collaboration:* The observations following the Corollary 3 show that even the single agent running SubGoss without communications is able to utilize the side information and incur lower regret ($O(m\sqrt{T} \log T + d\sqrt{T})$), compared with an agent running optimism in the face of uncertainty linear bandit algorithm (OFUL) [14] without any side information. However, the time taken by the single agent running SubGoss to reap the benefits of the subspace side information is very large in high-dimensional settings ($T = \Omega(e^d)$). In contrast, the ability of a multi-agent system to learn the right subspace faster is what leads to large collaborative gain of $r_C(T) = \Omega(\frac{d}{\log d})$ by time $T = \Omega(d)$. These gains are also observed empirically in Fig. 1. These gains are more pronounced and are observed for large duration of time in settings with large d , which is typical in many modern applications.

VI. NUMERICAL RESULTS

We evaluate the SubGoss algorithm empirically in synthetic simulations. We show the cumulative regret (after averaging across all agents) over 30 random runs of the algorithm with 95% confidence intervals. We compare its performance with two benchmarks: SubGoss algorithm with no collaborations (i.e., a single agent playing SubGoss algorithm) and a single agent playing the OFUL (classical LinUCB) algorithm of [14]. In this section, the number of times a subspace $\text{span}(U_k)$ (where $k \in S_j^{(i)}$) is explored during the Explore subroutine in phase j is set to $m \lceil b^{\frac{j-2}{2}} \rceil$, as the constants in SubGoss algorithm (see Algorithm 1) arise from somewhat loose tail bounds.

In our experiments, the agents are connected through a complete graph. Each m -dimensional subspace is the orthogonal matrix obtained by the singular value decomposition (SVD) of a random $d \times m$ matrix with independent identically distributed (i.i.d.) standard normal entries. The action set $\mathcal{A}$ consists of $5d$ i.i.d. Gaussian vectors on surface of the unit l_2 ball, along with orthonormal basis vectors for each of the K subspaces. The vector θ^* is the projected version of a standard Gaussian vector onto subspace 1 (the true subspace). We set $b = 2$ and $\lambda = 1$ in simulations. Fig. 1 evaluates the performance of SubGoss algorithm for different values of problem parameters (d, m, K, N).

Insights from numerical results: From simulations, we confirm several insights predicted by our theory. First, we see that SubGoss yields lower regret than OFUL for the single agent case, demonstrating that SubGoss can effectively leverage the

side-information provided through the subspaces. Second, we observe the collaboration gains, where any agent in the multi-agent setting incurs far smaller regret compared with a single agent without collaboration. Finally, we also observe that as the number of agents increases, the regret for every agent decreases. These collaborative gains follow as each agent has to search through a smaller set of subspaces to find the true subspace.

VII. CONCLUSION AND OPEN PROBLEMS

We studied a multiagent linear bandit problem with side information (in the form of disjoint m -dimensional subspaces), where only one of the subspaces contains the unknown parameter $\theta^* \in \mathbb{R}^d$, but agents are unaware of the subspace containing it. We proposed a novel decentralized algorithm, where agents collaborate by sending recommendations through pairwise gossip communications across a communication graph connecting them, to minimize their individual cumulative regret. We demonstrated that distributing the search for the subspace containing θ^* across the agents and learning of the unknown vector in the corresponding low-dimensional subspace results in a much smaller per-agent regret, compared with the case when agents do not communicate. However, this article leaves open, some important questions. This article assumed that all agents have exact knowledge of the subspaces. In several practical applications; however, the subspaces are estimated from historical data and as such can only be known noisily at best. Developing algorithms that can leverage benefit from collaboration while being robust to misspecifications is an interesting direction for future work. Another open problem is to establish lower bounds on regret under our model of information sharing. This is nontrivial to define since the communication budget needs to be accounted for in regret. To the best of our knowledge, lower bounds involving both communication and regret minimization have not been established even for the simple unstructured bandits case.

APPENDIX A

TECHNICAL ASSUMPTION FOR THEOREM 1

Building on the communication constraints considered in [9], we make the following mild assumption: The gossip matrix G is *irreducible*, i.e., for any $i \neq j \in [N]$, there exists $2 \leq l \leq N$ and $k_1, \dots, k_l \in [N]$, with $k_1 = i$ and $k_l = j$ such that the product $G(k_1, k_2) \dots G(k_{l-1}, k_l) > 0$. In words, the communication graph among the agents is connected [9].

This assumption is needed because if the communication graph among the agents is not connected, the setup becomes *degenerate*, as there exists at least a pair of agents which cannot participate in information exchange. However, the practical insights that can be obtained from our results are not affected by this assumption.

APPENDIX B

PROOF OF THEOREM 1

In this section and subsequent sections, we assume agents know the parameter S such that $\|\theta^*\|_2 \leq S$. In this article, we

set $S = 1$ for ease of exposition. Before going through the proof, we first provide some definitions and notations.

B1 Definitions and Notations

We adapt the proof ideas developed in [9] for the unstructured bandit case. Recall that for any phase j , agent $i \in [N]$, and subspace $k \in S_j^{(i)}$, $\tilde{n}_{k,j}^{(i)}$ is the number of times agent i explores the subspace $\text{span}(U_k)$ up to and including phase j and $\hat{O}_j^{(i)} = \arg \max_{k \in S_j^{(i)}} \|\tilde{\theta}_{k,j}^{(i)}\|_2$. In words, $\hat{O}_j^{(i)}$ is the ID of the subspace in which every agent $i \in [N]$ plays Projected LinUCB in the exploit time slots of phase j and subsequently, recommends it at the end of phase j . Let $\chi_j^{(i)}$ be the indicator variable for the event $\{1 \in S_j^{(i)}, \hat{O}_j^{(i)} \neq 1\}$, i.e.,

$$\chi_j^{(i)} = 1 \left(1 \in S_j^{(i)}, \hat{O}_j^{(i)} \neq 1 \right)$$

which indicates whether agent i , if it has the subspace $\text{span}(U_1)$, does not recommend it at the end of a phase. Similar to [9], we provide the definitions of certain random times that will aid in the analysis

$$\hat{\tau}_{\text{stab}}^{(i)} = \inf \{j' \geq \tau_0 : \forall j \geq j', \chi_j^{(i)} = 0\}$$

$$\hat{\tau}_{\text{stab}} = \max_{i \in [N]} \hat{\tau}_{\text{stab}}^{(i)}$$

$$\hat{\tau}_{\text{spr}}^{(i)} = \inf \{j \geq \hat{\tau}_{\text{stab}} : 1 \in S_j^{(i)}\} - \hat{\tau}_{\text{stab}}$$

$$\hat{\tau}_{\text{spr}} = \max_{i \in \{1, \dots, N\}} \hat{\tau}_{\text{spr}}^{(i)}$$

$$\tau = \hat{\tau}_{\text{stab}} + \hat{\tau}_{\text{spr}}$$

Here, $\hat{\tau}_{\text{stab}}^{(i)}$ is the earliest phase, such that if agent i has the subspace $\text{span}(U_1)$ in the phases following it, it will recommend the subspace $\text{span}(U_1)$. The number of phases it takes after $\hat{\tau}_{\text{stab}}$ to have the subspace $\text{span}(U_1)$ in its playing set is denoted by $\hat{\tau}_{\text{spr}}^{(i)}$. The following proposition shows that the system is frozen after phase τ , i.e., after phase τ , the set of subspaces of all agents remain fixed in the future.

Proposition 1: For all agents $i \in \{1, \dots, N\}$, we have almost surely

$$\bigcap_{j \geq \tau} S_j^{(i)} = S_\tau^{(i)}$$

$$\hat{O}_l^{(i)} = 1 \quad \forall l \geq \tau, \forall i \in \{1, \dots, N\}.$$

Proof: For any agent $i \in [N]$ and any phase $j \geq \tau$, we have for all $j \geq \tau$

$$\chi_j^{(i)} = 0 \tag{5}$$

as $\tau \geq \hat{\tau}_{\text{stab}}^{(i)}$. However, as $\tau \geq \hat{\tau}_{\text{stab}} + \hat{\tau}_{\text{spr}}^{(i)}$, we know that

$$1 \in S_j^{(i)}. \tag{6}$$

Equations (5) and (6) imply that $\hat{O}_j^{(i)} = 1$. Moreover, $\hat{O}_j^{(i)} = 1$ is true for all phases $j \geq \tau$ and all agents $i \in [N]$, as they are arbitrarily chosen. Furthermore, the update step of the algorithm

along with the abovementioned reasoning tells us that none of the agents will change their subspaces after any phase $j \geq \tau$, as the agents already have the correct subspace in their respective playing sets. Thus, $\bigcap_{j \geq \tau} S_j^{(i)} = S_\tau^{(i)}$ for all agents $i \in [N]$. ■

Proposition 1 also tells us that for all phases $j \geq \tau$, in the exploit time slots, all agents will play Projected LinUCB from the subspace $\text{span}(U_1)$, because the algorithm picks the subspace $\text{span}(U_{\hat{\mathcal{O}}_j^{(i)}})$ in the exploit time slots of phase j and $\hat{\mathcal{O}}_j^{(i)} = 1$ for all $j \geq \tau$.

B2 Intermediate Results

Before stating and proving the intermediate results, we highlight the key pieces needed to prove Theorem 1. We already showed in Proposition 1 that after the freezing phase τ , all agents have the correct subspace containing θ^* and recommend it henceforth. Thus, the expected cumulative regret incurred can be decomposed into two parts: the regret up to phase τ and the regret after phase τ .

The expected cumulative regret incurred up to phase τ is a constant independent of the time horizon (see Proposition 3). It is a consequence of following important observations resulting from pure exploration in the explore time steps.

- 1) For any agent $i \in [N]$ and subspace $k \in S_j^{(i)}$, the estimate $\hat{\theta}_{k,j}^{(i)}$ concentrates to $P_k \theta^*$ in l_2 norm (Lemma 4).
- 2) Subsequently, we show that the probability that an agent will not recommend and, thus, drop the correct subspace containing θ^* is small at the end of a phase (see Lemma 5).

The abovementioned observations imply that after a (random) phase, denoted by $\hat{\tau}_{\text{stab}} \leq \tau$, agents always recommend (and never drop) the correct subspace. After phase $\hat{\tau}_{\text{stab}}$, we stochastically dominate (in Proposition 4) the spreading time of the correct subspace with a standard rumor spreading process [30]. Hence, the expected cumulative regret up to phase τ is bounded by the total number of time steps taken to reach phase $\hat{\tau}_{\text{stab}}$ and the additional number of phases taken to spread the correct subspace.

Post phase τ , the active set of subspaces maintained by agents remains unchanged (as deduced in Proposition 1), and thus, the regret can be decomposed into sum of regret due to pure exploration and regret due to projected LinUCB. The regret due to projected LinUCB is adapted from the analysis of a similar algorithm conducted in [22].

The following intermediate results will precisely characterize the intuition behind the proof of Theorem 1.

Proposition 2: The regret of any agent $i \in \{1, \dots, N\}$ after playing for T steps is bounded by

$$\begin{aligned} \mathbb{E}[R_T^{(i)}] &\leq 2S \left(\mathbb{E} \left[\tau + \frac{b^\tau - 1}{b - 1} \right] \right) + \mathbb{E}[R_{\text{proj}, T}] \\ &\quad + 16mS \left(\frac{K}{N} + 2 \right) \log_b(h_{b,T}) + 16mS \left(\frac{K}{N} + 2 \right) \\ &\quad \times \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1} \end{aligned}$$

where $h_{b,T}$ is defined in Theorem 1.

Proof: We will first show that the instantaneous regret $w_t^{(i)} \leq 2S$ for all $i \in [N]$. In order to obtain this bound, note that for any $a \in \mathbb{R}^d$ such that $\|a\|_2 \leq 1$

$$|\langle \theta^*, a \rangle| \leq \|\theta^*\|_2 \cdot \|a\|_2 \leq S$$

by Cauchy–Schwarz inequality. Therefore, we have $w_t^{(i)} \leq 2S$ for all t .

Let $l \in \mathbb{N}$ such that SubGoss Algorithm is played for t steps by the end of phase l . t and l are related as follows:

$$t = \sum_{p=1}^l \lceil b^{p-1} \rceil. \quad (7)$$

Therefore, $\frac{b^l - 1}{b - 1} \leq t \leq l + \frac{b^l - 1}{b - 1}$. Assume that SubGoss algorithm is played for T steps such that T occurs in some phase E , i.e., $\frac{b^E - 1}{b - 1} + 1 \leq T \leq E + \frac{b^E - 1}{b - 1}$, and it follows that $E \leq \log_b(b(1 + (T - 1)(b - 1))) = \log_b(h_{b,T})$.

Let $e_j = \sum_{l=1}^j \lceil b^{l-1} \rceil$ denote the number of times SubGoss algorithm has been played by the end of phase j and $\text{Reg}_j^{(i)}$ denote the regret incurred by agent i in phase j , i.e., $\text{Reg}_j^{(i)} = \sum_{s=1}^{\lceil b^{j-1} \rceil} w_{e_{j-1}+s}$. From the definition of regret $R_T^{(i)}$

$$\begin{aligned} R_T^{(i)} &= \sum_{t=1}^T w_t^{(i)} \\ &\leq \sum_{j=1}^E \text{Reg}_j^{(i)} \\ &= \sum_{j=1}^{\tau} \text{Reg}_j^{(i)} + \sum_{j=\tau+1}^E \text{Reg}_j^{(i)}. \end{aligned} \quad (8)$$

We will now bound each of the terms in (8) separately. The first term $\sum_{j=1}^{\tau} \text{Reg}_j^{(i)}$ can be bounded as follows:

$$\begin{aligned} \sum_{j=1}^{\tau} \text{Reg}_j^{(i)} &= \sum_{j=1}^{\tau} \sum_{s=1}^{\lceil b^{j-1} \rceil} w_{e_{j-1}+s}^{(i)} \\ &\leq 2S \sum_{j=1}^{\tau} \sum_{s=1}^{\lceil b^{j-1} \rceil} 1 \\ &= 2S \sum_{j=1}^{\tau} \lceil b^{j-1} \rceil \leq 2S \left(\tau + \frac{b^\tau - 1}{b - 1} \right) \end{aligned} \quad (9)$$

where the second step follows from $w_t \leq 2S$ for all $t \in \mathbb{N}$ and the last step follows from the fact that $\lceil x \rceil \leq x + 1$ for all $x \in \mathbb{R}$. We bound the second term $\sum_{j=\tau+1}^E \text{Reg}_j^{(i)}$ in the following steps: let $d_j^{(i)} = 8m|S_\tau^{(i)}| \lceil b^{\frac{j-1}{2}} \rceil$, and $R_{\text{proj}, T}$ denote the regret incurred by playing Projected LinUCB on the subspace containing θ^* after the freezing phase τ , i.e., $R_{\text{proj}, T} = \sum_{j=\tau+1}^E \sum_{d_j^{(i)}+1}^{\lceil b^{j-1} \rceil} w_{e_{j-1}+s}^{(i)}$. Then, we have

$$\sum_{j=\tau+1}^E \text{Reg}_j^{(i)} = \sum_{j=\tau+1}^E \sum_{s=1}^{\lceil b^{j-1} \rceil} w_{e_{j-1}+s}^{(i)}$$

$$\begin{aligned}
&= \sum_{j=\tau+1}^E \left(\sum_{s=1}^{d_j^{(i)}} w_{e_{j-1}+s}^{(i)} + \sum_{d_j^{(i)+1}^{\lceil b^{j-1} \rceil}} w_{e_{j-1}+s}^{(i)} \right) \\
&\stackrel{(a)}{\leq} 2S \sum_{j=1}^E \sum_{s=1}^{8m|S_\tau^{(i)}| \lceil b^{\frac{j-1}{2}} \rceil} 1 + R_{\text{proj},T} \\
&\stackrel{(b)}{\leq} 16mS \left(\frac{K}{N} + 2 \right) \log_b(h_{b,T}) \\
&\quad + 16mS \left(\frac{K}{N} + 2 \right) \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1} + R_{\text{proj},T}. \tag{10}
\end{aligned}$$

Recall that for any agent i , SubGoss algorithm explores in the first $d_j^{(i)} = 8m|S_\tau^{(i)}| \lceil b^{\frac{j-1}{2}} \rceil$ time slots of phase j by playing the orthonormal basis vectors of each of the subspaces in the playing set in a round robin fashion. Therefore, in step (a), we bound the total number of explore steps from phase $j > \tau$ (first term) by the bound on total number of explore steps from $t = 1$ to T . In the remaining time slots of phases $j \in \mathbb{N}$, agents play Projected LinUCB in the subspace $\text{span}(U_{\tilde{\mathcal{O}}_j^{(i)}})$ and $\tilde{\mathcal{O}}_j^{(i)} = 1$ for all $j > \tau$. Thus, the second term in step (a) is bounded previously by the regret incurred by playing Projected LinUCB in the subspace $\text{span}(U_1)$ for T time steps (as the number of times an agent will play Projected LinUCB is less than T). Step (b) follows from the discussion that if time step T occurs in some phase E then $E \leq \log_b(h_{b,T})$, $|S_j^{(i)}| \leq \frac{K}{N} + 2$ for all $j \in \mathbb{N}$, and $\lceil x \rceil \leq x + 1$ for all $x \in \mathbb{R}$.

Substituting (9) and (10) in (8), followed by taking expectation on both sides completes the proof of Proposition 2. ■

The following lemma bounds the probability that for any subspace $k \in S_j^{(i)}$, $\tilde{\theta}_{k,j}^{(i)}$ deviates from $P_k \theta^*$ in l_2 norm after the explore time slots in phase j , which will eventually help us obtain a bound on probability of picking the wrong subspace.

Lemma 4: For any agent $i \in [N]$, phase $j \geq \tau_0$, and $k \in S_j^{(i)}$, we have

$$\mathbb{P} \left(\|\tilde{\theta}_{k,j}^{(i)} - P_k \theta^*\|_2 > \epsilon \right) \leq 2m \exp \left(-\frac{4\epsilon^2}{m} b^{\frac{j-1}{2}} \right)$$

where $\epsilon > 0$.

Proof: We have for any $k \in S_j^{(i)}$

$$\begin{aligned}
\tilde{\theta}_{k,j}^{(i)} &= \arg \min_{\theta \in \mathbb{R}^d} \left\| (\tilde{A}_{k,\tilde{n}_{k,j}}^{(i)})^T \theta - \tilde{r}_{k,\tilde{n}_{k,j}}^{(i)} \right\|_2 \\
&= \arg \min_{\theta \in \mathbb{R}^d} \left\| (P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)})^T \theta - \tilde{r}_{k,\tilde{n}_{k,j}}^{(i)} \right\|_2
\end{aligned}$$

where the last step follows from the fact that during the explore time slots, orthonormal basis vectors for each of the subspaces in $S_j^{(i)}$ are played in a round robin fashion. By squaring the objective function in the last step, taking the gradient and setting it to all zeroes vector, we get

$$\tilde{\theta}_{k,j}^{(i)} = (P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k)^\dagger (P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k) \theta^*$$

$$+ (P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k)^\dagger P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \eta_{k,\tilde{n}_{k,j}}^{(i)} \tag{11}$$

where $M^\dagger$ denotes the Moore–Penrose pseudoinverse of the matrix M . By substituting $P_k = U_k U_k^T$, we get $P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k = U_k \tilde{\Sigma}_{k,\tilde{n}_{k,j}}^{(i)} U_k^T$, where $\tilde{\Sigma}_{k,\tilde{n}_{k,j}}^{(i)} = (U_k^T \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)})(U_k^T \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)})^T$. Note that $\tilde{\Sigma}_{k,\tilde{n}_{k,j}}^{(i)}$ is a symmetric, full-rank $m \times m$ matrix, as $\tilde{A}_{k,\tilde{n}_{k,j}}^{(i)}$ is a matrix whose columns are the orthonormal basis vectors of the subspace $\text{span}(U_k)$ in a round robin fashion and $\tilde{n}_{k,j} > m$. Therefore, $(P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k)^\dagger = U_k (\tilde{\Sigma}_{k,\tilde{n}_{k,j}}^{(i)})^{-1} U_k^T$, and thus, $(P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k)^\dagger (P_k \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)T} P_k) = U_k U_k^T = P_k$. Moreover, as $\tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} = [U_k \dots U_k]_{d \times \tilde{n}_{k,j}}$, $U_k^T \tilde{A}_{k,\tilde{n}_{k,j}}^{(i)} = [I_m \dots I_m]_{m \times \tilde{n}_{k,j}}$ (where I_m denotes the $m \times m$ identity matrix), and thus, $\tilde{\Sigma}_{k,\tilde{n}_{k,j}}^{(i)} = \frac{\tilde{n}_{k,j}}{m} I_m$. Substituting everything above-mentioned in (11), we get

$$\tilde{\theta}_{k,j}^{(i)} = P_k \theta^* + U_k v_{\tilde{n}_{k,j}}^{(i)}$$

where $v_{\tilde{n}_{k,j}}^{(i)}$ is an $m \times 1$ vector whose entries are $v_{\tilde{n}_{k,j},n}^{(i)} = \frac{m}{\tilde{n}_{k,j}} \sum_{p=1: a_{k,p}^{(i)} = u_{k,n}} \eta_p^{(i)}$ for all $n \in [m]$, $a_{k,p}$ denotes the p th column of $\tilde{A}_{k,\tilde{n}_{k,j}}^{(i)}$, and $u_{k,n}$ denotes the n th column of U_k . Hence,

$$\|\tilde{\theta}_{k,j}^{(i)} - P_k \theta^*\|_2^2 = v_{\tilde{n}_{k,j}}^{(i)T} U_k^T U_k v_{\tilde{n}_{k,j}}^{(i)} = \|v_{\tilde{n}_{k,j}}^{(i)}\|_2^2$$

where the abovementioned equality follows from the fact that U_k is an orthonormal matrix. From the assumption that the additive noise is conditionally 1-sub-Gaussian, we know that

$$\mathbb{P} \left(|v_{\tilde{n}_{k,j},n}^{(i)}| > \gamma \right) \leq 2e^{-\frac{\gamma^2 \tilde{n}_{k,j}}{2m}} \tag{12}$$

for all $\gamma > 0$. If $|v_{\tilde{n}_{k,j},n}^{(i)}| \leq \frac{\epsilon}{\sqrt{m}}$ for all $n \in [m]$ and $\epsilon > 0$, then $\|\tilde{\theta}_{k,j}^{(i)} - P_k \theta^*\|_2 \leq \epsilon$. Hence,

$$\begin{aligned}
\mathbb{P} \left(\|\tilde{\theta}_{k,j}^{(i)} - P_k \theta^*\|_2 > \epsilon \right) &\leq \mathbb{P} \left(\exists n \in [m] : |v_{\tilde{n}_{k,j},n}^{(i)}| > \frac{\epsilon}{\sqrt{m}} \right) \\
&= \mathbb{P} \left(\bigcup_{n=1}^m \left(|v_{\tilde{n}_{k,j},n}^{(i)}| > \frac{\epsilon}{\sqrt{m}} \right) \right) \\
&\stackrel{(a)}{\leq} \sum_{n=1}^m \mathbb{P} \left(|v_{\tilde{n}_{k,j},n}^{(i)}| > \frac{\epsilon}{\sqrt{m}} \right) \\
&\stackrel{(b)}{\leq} 2m \exp \left(-\frac{\epsilon^2 \tilde{n}_{k,j}}{2m^2} \right) \\
&\stackrel{(c)}{\leq} 2m \exp \left(-\frac{\epsilon^2}{2m^2} . 8m \lceil b^{\frac{j-1}{2}} \rceil \right)
\end{aligned}$$

$$\stackrel{(d)}{\leq} 2m \exp\left(-\frac{4\epsilon^2}{m} b^{\frac{j-1}{2}}\right).$$

Step (a) is a direct application of union bound. Step (b) uses the result from (12). In step (c), we use the fact that any subspace $k \in S_j^{(i)}$ is explored for at least $8m \lceil b^{\frac{j-1}{2}} \rceil$ times up to and including phase j . Step (d) follows from the inequality $\lceil x \rceil \geq x$ for all $x \in \mathbb{R}$, thus concluding the proof. ■

We will now obtain a bound on probability for choosing a wrong subspace. Since $\theta^* \in \text{span}(U_1)$, any subspace chosen other than $\text{span}(U_1)$ will result in an error. Mathematically, it can be expressed as $\hat{\mathcal{O}}_j^{(i)} \neq 1$, which implies that at least one of the events in the following is true:

$$\left(\|\tilde{\theta}_{\hat{\mathcal{O}}_j^{(i)},j}^{(i)} - P_{\hat{\mathcal{O}}_j^{(i)}}\theta^*\|_2 > \frac{\Delta}{2}\right), \left(\|\tilde{\theta}_{1,j}^{(i)} - P_1\theta^*\|_2 > \frac{\Delta}{2}\right). \quad (13)$$

The abovementioned implication follows from the contrapositive argument by showing that the negation of both the events in (13) must be simultaneously true so that $\hat{\mathcal{O}}_j^{(i)} = 1$ holds. The contrapositive argument can be proved as follows: observe that for any $k (\neq 1) \in S_j^{(i)}$, $(\|\tilde{\theta}_{k,j}^{(i)} - P_k\theta^*\|_2 \leq \frac{\Delta}{2}) \cap (\|\tilde{\theta}_{1,j}^{(i)} - P_1\theta^*\|_2 \leq \frac{\Delta}{2})$ implies $\|P_k\theta^*\|_2 - \frac{\Delta}{2} \leq \|\tilde{\theta}_{k,j}^{(i)}\|_2 \leq \|P_k\theta^*\|_2 + \frac{\Delta}{2}$ and $\|P_1\theta^*\|_2 - \frac{\Delta}{2} \leq \|\tilde{\theta}_{1,j}^{(i)}\|_2 \leq \|P_1\theta^*\|_2 + \frac{\Delta}{2}$, which follows from the triangle inequality. Now, suppose that $\|\tilde{\theta}_{k,j}^{(i)}\|_2 \geq \|\tilde{\theta}_{1,j}^{(i)}\|_2$. Using the bounds on $\|\tilde{\theta}_{k,j}^{(i)}\|_2$ and $\|\tilde{\theta}_{1,j}^{(i)}\|_2$ obtained from the triangle inequality, $\|\tilde{\theta}_{k,j}^{(i)}\|_2 \geq \|\tilde{\theta}_{1,j}^{(i)}\|_2$ implies that $\|P_k\theta^*\|_2 - \frac{\Delta}{2} \geq \|P_1\theta^*\|_2 + \frac{\Delta}{2}$, which after rearranging the terms results in the following inequality: $\|P_1\theta^*\|_2 - \|P_k\theta^*\|_2 \leq -\Delta$. Note that the left-hand side of this inequality is strictly positive, as $P_1\theta^* = \theta^*$. This is a contradiction, as a strictly positive number cannot be less than a strictly negative number, as $\Delta > 0$. Therefore, our initial assertion that $\|\tilde{\theta}_{k,j}^{(i)}\|_2 \geq \|\tilde{\theta}_{1,j}^{(i)}\|_2$ is incorrect and the claim in (13) follows.

The abovementioned discussion results in the following lemma.

Lemma 5: For any agent $i \in [N]$ and phase $j \geq \tau_0$, we have

$$\mathbb{P}\left(1 \in S_j^{(i)}, \hat{\mathcal{O}}_j^{(i)} \neq 1\right) \leq 4m \exp\left(-\frac{\Delta^2}{m} b^{\frac{j-1}{2}}\right).$$

Proof: We have

$$\begin{aligned} & \mathbb{P}\left(1 \in S_j^{(i)}, \hat{\mathcal{O}}_j^{(i)} \neq 1\right) \\ &= \mathbb{P}\left(1 \in S_j^{(i)}, \|\tilde{\theta}_{\hat{\mathcal{O}}_j^{(i)},j}^{(i)}\|_2 \geq \|\tilde{\theta}_{k,j}^{(i)}\|_2 \text{ for all } k \in S_j^{(i)}\right) \\ & \stackrel{(a)}{\leq} \mathbb{P}\left(\left(\|\tilde{\theta}_{\hat{\mathcal{O}}_j^{(i)},j}^{(i)} - P_{\hat{\mathcal{O}}_j^{(i)}}\theta^*\|_2 > \frac{\Delta}{2}\right) \cup \left(\|\tilde{\theta}_{1,j}^{(i)} - P_1\theta^*\|_2 > \frac{\Delta}{2}\right)\right) \\ & \stackrel{(b)}{\leq} \mathbb{P}\left(\|\tilde{\theta}_{\hat{\mathcal{O}}_j^{(i)},j}^{(i)} - P_{\hat{\mathcal{O}}_j^{(i)}}\theta^*\|_2 > \frac{\Delta}{2}\right) \\ & \quad + \mathbb{P}\left(\|\tilde{\theta}_{1,j}^{(i)} - P_1\theta^*\|_2 > \frac{\Delta}{2}\right) \end{aligned}$$

$$\stackrel{(c)}{\leq} 4m \exp\left(-\frac{\Delta^2}{m} b^{\frac{j-1}{2}}\right)$$

where step (a) follows from the fact that $\{\hat{\mathcal{O}}_j^{(i)} \neq 1\}$ implies at least one of the events in (13) must be true, step (b) follows from union bound, and step (c) follows from Lemma 4. This concludes the proof of Lemma 5. ■

Proposition 3: The freezing time $\tau + \frac{b^\tau - 1}{b - 1}$ is bounded by

$$\mathbb{E}\left[\tau + \frac{b^\tau - 1}{b - 1}\right] \leq g(b) \left(\lceil b^{2\tau_0} \rceil + \frac{48b^3}{\log b} \cdot \frac{m^4 N}{\Delta^6} + b\mathbb{E}[b^{2\hat{\tau}_{\text{spr}}}] \right)$$

where τ_0 and $g(b)$ are defined in Theorem 1.

Proof: We follow similar steps as in the proof for [9, Prop. 3] for establishing the abovementioned result. As τ is a nonnegative random variable

$$\begin{aligned} \mathbb{E}[b^\tau] &\leq \mathbb{E}[\lceil b^\tau \rceil] \\ &= \sum_{t \geq 1} \mathbb{P}(\lceil b^\tau \rceil \geq t) \\ &\leq 1 + \sum_{t \geq 2} \mathbb{P}(b^\tau + 1 \geq t) \\ &\leq 1 + \sum_{t \geq 2} \mathbb{P}(\tau \geq \lfloor \log_b(t - 1) \rfloor) \\ &= 1 + \sum_{t \geq 1} \mathbb{P}(\hat{\tau}_{\text{stab}} + \hat{\tau}_{\text{spr}} \geq \lfloor \log_b t \rfloor). \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}[b^\tau] - 1 &\leq \sum_{t \geq 1} \left(\mathbb{P}\left(\hat{\tau}_{\text{stab}} \geq \frac{1}{2} \lfloor \log_b t \rfloor\right) + \mathbb{P}\left(\hat{\tau}_{\text{spr}} \geq \frac{1}{2} \lfloor \log_b t \rfloor\right) \right) \\ &\leq \sum_{t \geq 1} \left(\mathbb{P}\left(\hat{\tau}_{\text{stab}} \geq \frac{1}{2} \lfloor \log_b t \rfloor\right) + \mathbb{P}\left(\hat{\tau}_{\text{spr}} \geq \frac{1}{2} \log_b t - \frac{1}{2}\right) \right) \\ &= \sum_{t \geq 1} \mathbb{P}\left(\hat{\tau}_{\text{stab}} \geq \frac{1}{2} \lfloor \log_b t \rfloor\right) + \sum_{t \geq 1} \mathbb{P}\left(b^{2\hat{\tau}_{\text{spr}}+1} \geq t\right) \\ &\leq \lceil b^{2\tau_0} \rceil + \sum_{t \geq \lceil b^{2\tau_0} \rceil + 1} \mathbb{P}\left(\hat{\tau}_{\text{stab}} \geq \frac{1}{2} \lfloor \log_b t \rfloor\right) + b\mathbb{E}[b^{2\hat{\tau}_{\text{spr}}}]. \end{aligned}$$

Since the spreading time with the standard rumor model dominates $\hat{\tau}_{\text{spr}}$, we use this to bound the term $\mathbb{E}[b^{2\hat{\tau}_{\text{spr}}}]$ in Proposition 4 after the proof of Proposition 3. The summation in the last step is bounded by using Lemma 5, given as follows: for some fixed $x \geq \tau_0$, we have

$$\begin{aligned} \mathbb{P}(\hat{\tau}_{\text{stab}} \geq x) &= \mathbb{P}\left(\bigcup_{i=1}^N (\hat{\tau}_{\text{stab}}^{(i)} \geq x)\right) \\ &\leq \sum_{i=1}^N \mathbb{P}(\hat{\tau}_{\text{stab}}^{(i)} \geq x) \\ &= \sum_{i=1}^N \mathbb{P}\left(\bigcup_{l=x}^{\infty} (\chi_l^{(i)} = 1)\right) \end{aligned}$$

$$\begin{aligned}
&\leq \sum_{i=1}^N \sum_{l \geq x} \mathbb{P} \left(\chi_l^{(i)} = 1 \right) \\
&= \sum_{i=1}^N \sum_{l \geq x} \mathbb{P} \left(1 \in S_j^{(i)}, \hat{\mathcal{O}}_j^{(i)} \neq 1 \right) \\
&\stackrel{(a)}{\leq} \sum_{i=1}^N \sum_{l \geq x} 4m \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right) \\
&= \sum_{l \geq x} 4mN \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right) \quad (14)
\end{aligned}$$

where we use Lemma 5 in step (a). Thus, we obtain the following:

$$\begin{aligned}
&\sum_{t \geq 2^{2\tau_0}} \mathbb{P} \left(\hat{\tau}_{\text{stab}} \geq \frac{1}{2} \lfloor \log_b t \rfloor \right) \\
&\leq \sum_{t \geq \lceil b^{2\tau_0} \rceil + 1} \sum_{l \geq \frac{1}{2} \lfloor \log_b t \rfloor} 4mN \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right) \\
&\leq 4mN \sum_{t \geq \lceil b^{2\tau_0} \rceil + 1} \sum_{l \geq \frac{1}{2} \lfloor \log_b t \rfloor} \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right) \\
&\stackrel{(b)}{\leq} 4mN \sum_{l \geq \tau_0} \sum_{t = \lceil b^{2\tau_0} \rceil + 1}^{b^{2l+1}} \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right) \\
&\leq 4mN \sum_{l \geq \tau_0} b^{2l+1} \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right)
\end{aligned}$$

where step (b) follows by rewriting the range of summations. The sum $\sum_{l \geq \tau_0} b^{2l+1} \exp(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}})$ is bounded as follows:

$$\begin{aligned}
&\sum_{l \geq \tau_0} b^{2l+1} \exp \left(-\frac{\Delta^2}{m} b^{\frac{l-1}{2}} \right) \\
&\leq \int_{x=1}^{\infty} b^{2x+1} \exp \left(-\frac{\Delta^2}{m} b^{\frac{x-1}{2}} \right) dx \\
&\stackrel{(c)}{\leq} \int_{u=1}^{\infty} b^{4 \log_b u + 3} \exp \left(-\frac{\Delta^2}{m} u \right) \frac{2}{u \log b} du \\
&\leq \frac{2b^3}{\log b} \int_{u=0}^{\infty} u^3 \exp \left(-\frac{\Delta^2}{m} u \right) du \\
&= \frac{12b^3}{\log b} \left(\frac{m}{\Delta^2} \right)^3
\end{aligned}$$

where we perform change of variables with $x = 2 \log_b u + 1$ in step (c). Therefore,

$$\mathbb{E}[b^\tau - 1] \leq \lceil b^{2\tau_0} \rceil + \frac{48b^3}{\log b} \cdot \frac{m^4 N}{\Delta^6} + b \mathbb{E}[b^{2\hat{\tau}_{\text{spr}}}]$$

For bounding $\mathbb{E}[\tau]$, notice that $\tau \leq \frac{b^\tau - 1}{\log b}$, where we use the fact that for all $b > 1$, $b^x - x \log b - 1 \geq 0$ for all $x \geq 0$. Thus, $\mathbb{E}[\tau] \leq \frac{1}{\log b} \mathbb{E}[b^\tau - 1]$. Substituting the bound for $\mathbb{E}[b^\tau - 1]$ obtained previously completes the proof of Proposition 3. ■

Proposition 4: The random variable $\hat{\tau}_{\text{spr}}$ is stochastically dominated by $\tau_{\text{spr}}^{(G)}$.

Proof: The proof follows in a similar way as the proof for [9, Prop. 4]. ■

Proposition 5: τ_0 defined in Theorem 1 is bounded by

$$\tau_0 \leq 2 \log_b \left(16m \left(\frac{K}{N} + 2 \right) \right) + 1.$$

Proof: From Theorem 1

$$\tau_0 = \min \left\{ j \in \mathbb{N} : \forall j' \geq j, \lceil b^{j'-1} \rceil \geq 8m \left(\frac{K}{N} + 2 \right) \lceil b^{\frac{j'-1}{2}} \rceil \right\}.$$

As $8m(\frac{K}{N} + 2) \lceil b^{\frac{j'-1}{2}} \rceil \leq 8m(\frac{K}{N} + 2)b^{\frac{j'-1}{2}} + 8m(\frac{K}{N} + 2) \leq 16m(\frac{K}{N} + 2)b^{\frac{j'-1}{2}}$, the minimum value of j that satisfies $b^{j-1} \geq 16m(\frac{K}{N} + 2)b^{\frac{j-1}{2}}$ is an upper bound on τ_0 . Rearranging the terms results in $j \geq 2 \log_b(16m(\frac{K}{N} + 2)) + 1$ and, thus, $\tau_0 \leq 2 \log_b(16m(\frac{K}{N} + 2)) + 1$. ■

B3 Proof of Theorem 1

From Theorem 8 and $\delta = \frac{1}{T}$, $\mathbb{E}[R_{\text{proj}, T}]$ is bounded as follows:

$$\begin{aligned}
&\mathbb{E}[R_{\text{proj}, T}] \\
&= \mathbb{E} \left[R_{\text{proj}, T} \mathbf{1} \left(R_{\text{proj}, T} \leq \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)} \right) \right] \\
&\quad + \mathbb{E} \left[R_{\text{proj}, T} \mathbf{1} \left(R_{\text{proj}, T} > \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)} \right) \right] \\
&\stackrel{(a)}{\leq} \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)} \\
&\quad + 2S \mathbb{E} \left[\sum_{t=1}^T \mathbf{1} \left(R_{\text{proj}, T} > \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)} \right) \right] \\
&= \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)} \\
&\quad + 2S \sum_{t=1}^T \mathbb{E} \left[\mathbf{1} \left(R_{\text{proj}, T} > \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)} \right) \right] \\
&\stackrel{(b)}{\leq} \sqrt{8mT\beta_T^2 \log \left(1 + \frac{T}{m\lambda} \right)} + 2S \cdot \frac{1}{T} \cdot \sum_{t=1}^T 1 \\
&= \sqrt{8mT\beta_T^2 \log \left(1 + \frac{T}{m\lambda} \right)} + 2S. \quad (15)
\end{aligned}$$

In step (a), the first sum is trivially bounded by $\sqrt{8mT\beta_T^2 \log(1 + \frac{T}{m\lambda})}$ and the second sum uses the definition of $R_{\text{proj}, T}$ with $w_t \leq 2S$ for all $t \in \mathbb{N}$. Step (b) uses Theorem 8 with $\delta = \frac{1}{T}$.

Substituting the results of Propositions 3 and 4, along with (15), into Proposition 2 concludes the proof of Theorem 1.

APPENDIX C

REGRET UPPER BOUND FOR SINGLE AGENT RUNNING
SUBGOSS ALGORITHM WITHOUT COMMUNICATIONS

Theorem 6: With the same assumptions as in Theorem 1, when a single agent runs SubGoss algorithm in case of no communication, the regret after any time $T \in \mathbb{N}$ is bounded by

$$\begin{aligned} \mathbb{E}[R_T] &\leq \underbrace{\sqrt{8mT\beta_T^2 \log\left(1 + \frac{T}{m\lambda}\right)}}_{\text{Projected LinUCB Regret}} + 2S \\ &\quad + \underbrace{2Sg(b) \left(\lceil b(16mK)^2 \rceil + \frac{8b^2}{\log b} \cdot \frac{m^2}{\Delta^2} \right)}_{\text{Constant cost of right subspace search}} \\ &\quad + \underbrace{16mKS \log_b(h_{b,T}) + 16mKS \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1}}_{\text{Cost of subspace exploration}}. \end{aligned} \quad (16)$$

Here, β_T , $g(b)$, and $h_{b,T}$ are the same as in Theorem 1.

Proof: Before we prove Theorem 6, we set some notation. Let $\hat{\mathcal{O}}_j = \arg \max_{k \in [K]} \|\hat{\theta}_{k,j}^{(i)}\|_2$, i.e., $\hat{\mathcal{O}}_j$ denotes the ID of subspace in which the single agent plays Projected LinUCB in the exploit time slots of phase j . Let $\bar{\tau}_0 = \min\{j \in \mathbb{N} : \forall j' \geq j, \lceil b^{j'-1} \rceil \geq 8mK \lceil b^{\frac{j'-1}{2}} \rceil\}$. Following the proof of Proposition 5, it can be shown that $\bar{\tau}_0 \leq 2 \log_b(16mK) + 1$.

We also define a random phase τ_{freeze} as follows:

$$\tau_{\text{freeze}} = \inf\{j \geq \bar{\tau}_0 : \forall j' \geq j, \hat{\mathcal{O}}_{j'} = 1\}.$$

Thus, τ_{freeze} is the earliest phase after which the single agent will play the projected LinUCB from the subspace $\text{span}(U_1)$ in the exploit time slots of a phase. Note that the random phase τ_{freeze} plays the same role as $\hat{\tau}_{\text{stab}}$ in the multiagent case. This suggests that the regret analysis in this case must follow the same chain of argument as for the multiagent case.

Following the same steps as for Proposition 2, the bound on the regret of the single agent after T time steps is given by

$$\begin{aligned} \mathbb{E}[R_T] &\leq 2S \left(\mathbb{E} \left[\tau_{\text{freeze}} + \frac{b^{\tau_{\text{freeze}}} - 1}{b - 1} \right] \right) + \mathbb{E}[R_{\text{proj},T}] \\ &\quad + 16mKS \log_b(h_{b,T}) + 16mKS \frac{\sqrt{h_{b,T}} - 1}{\sqrt{b} - 1}. \end{aligned} \quad (17)$$

We have already shown in (15) that

$$\mathbb{E}[R_{\text{proj},T}] \leq \sqrt{8mT\beta_T^2 \log\left(1 + \frac{T}{m\lambda}\right)} + 2S.$$

We will now bound $\mathbb{E}[\tau_{\text{freeze}}]$ and $\mathbb{E}[b^{\tau_{\text{freeze}}}]$ to complete the proof. We first bound $\mathbb{E}[b^{\tau_{\text{freeze}}}]$. From the definition of expectation for positive random variables

$$\mathbb{E}[b^{\tau_{\text{freeze}}}] \leq \mathbb{E}[\lceil b^{\tau_{\text{freeze}}} \rceil]$$

$$\begin{aligned} &= \sum_{t=1}^{\infty} \mathbb{P}(\lceil b^{\tau_{\text{freeze}}} \rceil \geq t) \\ &\leq 1 + \sum_{t=2}^{\infty} \mathbb{P}(b^{\tau_{\text{freeze}}} + 1 \geq t) \\ &\leq 1 + \sum_{t=1}^{\infty} \mathbb{P}(\tau_{\text{freeze}} \geq \lfloor \log_b t \rfloor) \\ &\leq 1 + \lceil b^{\bar{\tau}_0} \rceil + \sum_{t=\lceil b^{\bar{\tau}_0} \rceil+1}^{\infty} \mathbb{P}(\tau_{\text{freeze}} \geq \lfloor \log_b t \rfloor). \end{aligned}$$

We will bound the summation in the last term in the following:

$$\begin{aligned} &\sum_{t=\lceil b^{\bar{\tau}_0} \rceil+1}^{\infty} \mathbb{P}(\tau_{\text{freeze}} \geq \lfloor \log_b t \rfloor) \\ &\stackrel{(a)}{=} \sum_{t=\lceil b^{\bar{\tau}_0} \rceil+1}^{\infty} \mathbb{P} \left(\bigcup_{j \geq \lfloor \log_b t \rfloor} (\hat{\mathcal{O}}_j \neq 1) \right) \\ &\leq \sum_{t=\lceil b^{\bar{\tau}_0} \rceil+1}^{\infty} \sum_{j \geq \lfloor \log_b t \rfloor} \mathbb{P}(\hat{\mathcal{O}}_j \neq 1) \\ &\stackrel{(b)}{\leq} \sum_{t=\lceil b^{\bar{\tau}_0} \rceil+1}^{\infty} \sum_{j \geq \lfloor \log_b t \rfloor} 4m \exp \left(-\frac{\Delta^2}{m} \cdot b^{\frac{j-1}{2}} \right) \\ &\stackrel{(c)}{\leq} 4m \sum_{j \geq \bar{\tau}_0} \sum_{t=\lceil b^{\bar{\tau}_0} \rceil+1}^{b^{j+1}} \exp \left(-\frac{\Delta^2}{m} \cdot b^{\frac{j-1}{2}} \right) \\ &\leq 4m \sum_{j \geq \bar{\tau}_0} b^{j+1} \exp \left(-\frac{\Delta^2}{m} \cdot b^{\frac{j-1}{2}} \right) \\ &\leq 4m \int_{x=1}^{\infty} b^{x+1} \exp \left(-\frac{\Delta^2}{m} \cdot b^{\frac{x-1}{2}} \right) dx \\ &\stackrel{(d)}{=} \frac{8mb^2}{\log b} \int_{u=1}^{\infty} u \exp \left(-\frac{\Delta^2}{m} \cdot u \right) du \\ &\leq \frac{8mb^2}{\log b} \int_{u=0}^{\infty} u \exp \left(-\frac{\Delta^2}{m} \cdot u \right) du \\ &\leq \frac{8b^2}{\log b} \cdot \frac{m^2}{\Delta^2}. \end{aligned}$$

We use the definition of τ_{freeze} in step (a). In step (b), we substitute the bound on probability of choosing the subspace other than $\text{span}(U_1)$ from Lemma 5. We interchange the order of summation in step (c). In step (d), we perform a change of variables with $x = 2 \log_b u + 1$. Therefore,

$$\mathbb{E}[b^{\tau_{\text{freeze}}} - 1] \leq \lceil b(16mK)^2 \rceil + \frac{8b^2}{\log b} \cdot \frac{m^2}{\Delta^2}.$$

For bounding $\mathbb{E}[\tau_{\text{freeze}}]$, note that $\tau_{\text{freeze}} \leq \frac{b^{\tau_{\text{freeze}}} - 1}{\log b - 1}$, where we use the fact that for all $b > 1$, $b^x - x \log b - 1 \geq 0$ for all $x \geq 0$. Thus, $\mathbb{E}[\tau_{\text{freeze}}] \leq \frac{1}{\log b} \mathbb{E}[b^{\tau_{\text{freeze}}} - 1]$. Substituting the

bound for $\mathbb{E}[b^{\tau_{\text{freeze}}} - 1]$ obtained previously completes the proof of Theorem 6. ■

APPENDIX D ANALYSIS OF PROJECTED LINUCB

We now analyze Projected LinUCB as a separate black box, where every agent plays from the correct subspace containing θ^* for T steps. This holds as every agent plays Projected LinUCB on the subspace $\text{span}(U_1)$ during the exploit time slots after the freezing phase τ by only using the Projected LinUCB actions and rewards for $\text{span}(U_1)$, which of course will happen for less than T steps. As this analysis is valid for all agents, we drop the superscript (i) from all the pertinent variables, where $i \in [N]$. Recall that $\bar{V}_t(\lambda) = P_1(V_{t-1} + \lambda I_d)P_1 = U_1 \Sigma_t U_1^T$, where $\Sigma_t = U_1^T V_{t-1} U_1 + \lambda I_m$. Define $\tilde{V}_t(\lambda) = \bar{V}_t(\lambda) - \lambda P_1 = P_1 V_{t-1} P_1 = U_1 \bar{\Sigma}_t U_1^T$, where $\bar{\Sigma}_t = \Sigma_t - \lambda I_m = U_1^T V_{t-1} U_1$.

D1 Confidence Set Construction and Analysis

The construction of the confidence set is done in a similar way as in [4] and [22].

Theorem 7: Let $\delta \in (0, 1)$ and $\beta_{t,\delta} = S\sqrt{\lambda} + \sqrt{2 \log \frac{1}{\delta} + m \log(1 + \frac{t-1}{\lambda m})}$. Then,

$$\mathbb{P}(\forall t \in \mathbb{N}, \theta^* \in \mathcal{C}_t^{(i)}) \geq 1 - \delta$$

where

$$\mathcal{C}_t^{(i)} = \left\{ \theta \in \mathbb{R}^d : \|\hat{\theta}_t^{(i)} - \theta\|_{\bar{V}_t(\lambda)} \leq \beta_{t,\delta} \right\}. \quad (18)$$

The proof of Theorem 7 is adapted from the proof of [22, Th. 8], which considers a different setting: the subspace in which θ^* lies is unknown and needs to be estimated. The error in estimating the correct subspace appears in the construction of projected confidence set in [22, Th. 8]. However, in our setting, agents are aware of the true projection matrices of the subspaces and, thus, do not need to account for subspace estimation error while constructing confidence sets. This necessitates a different definition of the confidence set given in (18).

Thus, when agents play Projected LinUCB on the correct subspace, they do not have to pay the overhead of recovering the actual subspace from the perturbed action vectors. Hence, Theorem 7 is proved by substituting the estimated projection matrix $\hat{P}_t$ equal to the true projection matrix P_1 in the proof of [22, Th. 8] for all $t \in \mathbb{N}$.

D2 Regret Analysis

Before bounding the regret, let us set some notation here. We have

$$a_t \in \arg \max_{a \in \mathcal{A}_t} \max_{\theta \in \mathcal{C}_t^{(i)}} \langle \theta, P_1 a \rangle.$$

Let $a_t^* = \arg \max_{a \in \mathcal{A}_t} \langle \theta^*, a \rangle$ and $\bar{\theta}_t \in \mathcal{C}_t^{(i)}$ such that $\bar{\theta}_t = \arg \max_{\theta \in \mathcal{C}_t^{(i)}} \langle \theta, P_1 a_t \rangle$. The following theorem characterizes the regret after every agent has the right subspace.

Theorem 8: With probability at least $1 - \delta$, the regret incurred after playing Projected LinUCB on the subspace containing θ^*

for T steps satisfies

$$R_{\text{proj},T} \leq \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)}.$$

Proof: The proof of Theorem 8 is contingent on the following lemma. A similar lemma appears as [4, Lemma 19.4].

Lemma 9: Let $a_1, \dots, a_T$ be the sequence of action vectors played up to and including time T . Then,

$$\sum_{t=1}^T \min \left\{ 1, \|a_t\|_{\bar{V}_t(\lambda)^\dagger}^2 \right\} \leq 2 \log \left(\frac{\det(\Sigma_{T+1})}{\det(\Sigma_1)} \right).$$

Proof: The proof is identical to the proof of [4, Lemma 19.4], except that we use the recursive update of $\det(\Sigma_t)$ instead of $\det(V_{t-1})$. ■

We now have the required ingredients to complete the proof of Theorem 8. Using the fact that $\theta^* \in \mathcal{C}_t^{(i)}$ and from the algorithm definitions, the following chain of inequalities is true:

$$\begin{aligned} \langle \theta^*, a_t^* \rangle &= \langle P_1 \theta^*, a_t^* \rangle = \langle \theta^*, P_1 a_t^* \rangle \\ &\leq \max_{\theta \in \mathcal{C}_t^{(i)}} \langle \theta, P_1 a_t^* \rangle \leq \langle \bar{\theta}_t, P_1 a_t \rangle. \end{aligned} \quad (19)$$

Thus, for all $t \in \mathbb{N}$

$$w_t^{(i)} \leq 2\beta_{t,\delta} \|a_t\|_{\bar{V}_t(\lambda)^\dagger} \quad (20)$$

which is shown in a similar way as bounding r_t in [4, Theorem 19.2]. However, we additionally use the fact that $\theta^* = P_1 \theta^*$ and $P_1^2 = P_1$.

While proving Proposition 1, we showed that $w_t^{(i)} \leq 2S$ for all $t \in \mathbb{N}$. Combining this with (20) and $\beta_{T,\delta} \geq S$ results in

$$\begin{aligned} w_t^{(i)} &\leq 2 \min \left\{ S, \beta_{t,\delta} \|a_t\|_{\bar{V}_t(\lambda)^\dagger} \right\} \\ &\leq 2\beta_{T,\delta} \min \left\{ 1, \|a_t\|_{\bar{V}_t(\lambda)^\dagger} \right\}. \end{aligned} \quad (21)$$

Therefore, the cumulative regret incurred by playing projected LinUCB for T can be bounded as follows: let $\mathbb{I}$ denote an all-ones column vector of size T , and $\mathbf{R}$ be a column vector containing the elements $w_1, \dots, w_T$. Then,

$$\begin{aligned} R_{\text{proj},T} &= \sum_{t=1}^T w_t^{(i)} \\ &= \mathbb{I}^T \mathbf{R} \\ &\stackrel{(a)}{\leq} \sqrt{T} \sqrt{\sum_{t=1}^T (w_t^{(i)})^2} \\ &\stackrel{(b)}{\leq} \sqrt{T} \sqrt{4\beta_{T,\delta}^2 \sum_{t=1}^T \min \left\{ 1, \|A_t\|_{\bar{V}_t(\lambda)^\dagger}^2 \right\}} \\ &\stackrel{(c)}{\leq} \sqrt{8T\beta_{T,\delta}^2 \log \left(\frac{\det(\Sigma_{T+1})}{\det(\Sigma_1)} \right)} \\ &\stackrel{(d)}{\leq} \sqrt{8mT\beta_{T,\delta}^2 \log \left(1 + \frac{T}{m\lambda} \right)}. \end{aligned}$$

Step (a) follows from Cauchy–Schwarz inequality. In step (b), we use (21). Step (c) is obtained by the application of Lemma 9. Step (d) results from the fact that $\det(\Sigma_1) = \det(\Lambda) = \lambda^m$ and $\det(\Sigma_t) \leq (\lambda + \frac{t-1}{m})^m$ (which follows from [22, Lemma 11]), thus completing the proof. ■

REFERENCES

- [1] D. Chakrabarti, R. Kumar, F. Radlinski, and E. Upfal, “Mortal multi-armed bandits,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 273–280.
- [2] Y. Yue and T. Joachims, “Interactively optimizing information retrieval systems as a dueling bandits problem,” in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1201–1208.
- [3] E. Hillel, Z. S. Karnin, T. Koren, R. Lempel, and O. Somekh, “Distributed exploration in multi-armed bandits,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 854–862.
- [4] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
- [5] S. Bubeck *et al.*, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.
- [6] S. Baccapatnam, J. Tan, and L. Zhang, “Information sharing in distributed stochastic bandits,” in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2605–2613.
- [7] I. Bistriz and A. Leshem, “Distributed multi-player bandits—A Game of Thrones approach,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7222–7232.
- [8] E. Boursier and V. Perchet, “SIC-MMAB: Synchronisation involves communication in multiplayer multi-armed bandits,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12048–12057.
- [9] R. Chawla, A. Sankararaman, A. Ganesh, and S. Shakkottai, “The gossiping insert-eliminate algorithm for multi-agent bandits,” in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3471–3481.
- [10] D. Martínez-Rubio, V. Kanade, and P. Rebeschini, “Decentralized cooperative stochastic bandits,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4531–4542.
- [11] L. Li, W. Chu, J. Langford, and R. E. Schapire, “A contextual-bandit approach to personalized news article recommendation,” in *Proc. 19th Int. Conf. World Wide Web. ACM*, 2010, pp. 661–670.
- [12] H. Bastani and M. Bayati, “Online decision making with high-dimensional covariates,” *Operations Res.*, vol. 68, no. 1, pp. 276–294, 2020.
- [13] V. Dani, T. P. Hayes, and S. M. Kakade, “Stochastic linear optimization under bandit feedback,” in *Proc. 21st Annu. Conf. Learn. Theory*, 2008, pp. 355–366.
- [14] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, “Improved algorithms for linear stochastic bandits,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2312–2320.
- [15] A. Sankararaman, A. Ganesh, and S. Shakkottai, “Social learning in multi agent multi armed bandits,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 3, pp. 1–35, 2019.
- [16] M. Chakraborty, K. Y. P. Chua, S. Das, and B. Juba, “Coordinated versus decentralized exploration in multi-agent multi-armed bandits,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 164–170.
- [17] R. K. Kolla, K. Jagannathan, and A. Gopalan, “Collaborative learning of stochastic bandits over a social network,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1782–1795, Aug. 2018.
- [18] D. Vial, S. Shakkottai, and R. Srikant, “Robust multi-agent multi-armed bandits,” in *Proc. 22nd Int. Symp. Theory Algorithmic Found. Protocol Des. Mobile Netw. Mobile Comput.*, 2021, pp. 161–170.
- [19] P.-A. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo, “Optimal algorithms for multiplayer multi-armed bandits,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 4120–4129.
- [20] A. Agarwal *et al.*, “Making contextual decisions with low technical debt,” 2016, *arXiv:1606.03966*.
- [21] A. Tewari and S. A. Murphy, “From ads to interventions: Contextual bandits in mobile health,” in *Mobile Health*, Berlin, Germany: Springer, 2017, pp. 495–517.
- [22] S. Lale, K. Azizadenesheli, A. Anandkumar, and B. Hassibi, “Stochastic linear bandits with hidden low rank structure,” 2019, *arXiv:1901.09490*.
- [23] A. Carpentier and R. Munos, “Bandit theory meets compressed sensing for high dimensional stochastic linear bandit,” in *Proc. Artif. Intell. Statist.*, 2012, pp. 190–198.
- [24] S. Gerchinovitz, “Sparsity regret bounds for individual sequences in online linear regression,” in *Proc. 24th Annu. Conf. Learn. Theory. JMLR Workshop Conf.*, 2011, pp. 377–396.
- [25] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvári, “Online-to-confidence-set conversions and application to sparse stochastic bandits,” in *Proc. Artif. Intell. Statist.*, 2012, pp. 1–9.
- [26] Y. Wang, J. Hu, X. Chen, and L. Wang, “Distributed bandit learning: Near-optimal regret with efficient communication,” in *Proc. Int. Conf. Learn. Representations*, 2020.
- [27] N. Korda, B. Szorenyi, and S. Li, “Distributed clustering of linear bandits in peer to peer networks,” in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1301–1309.
- [28] A. Dubey and A. Pentland, “Differentially-private federated linear bandits,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6003–6014, 2020.
- [29] S. Amani and C. Thrampoulidis, “Decentralized multi-agent linear bandits with safety constraints,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6627–6635.
- [30] F. Chierichetti, S. Lattanzi, and A. Panconesi, “Almost tight bounds for rumour spreading with conductance,” in *Proc. 42nd ACM Symp. Theory Comput.*, 2010, pp. 399–408.



Ronshee Chawla received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Indore, India, in 2015, and the M.S. degree in 2019 in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, where he is currently working toward the Ph.D. degree in electrical and computer engineering.

From 2015 to 2016, he was a Scientist with the Defence Research and Development Organisation (DRDO), Ministry of Defence, Government of India. His research focuses on development and analysis of decentralized algorithms for multiagent machine learning.

Mr. Chawla was the recipient of the President of India Gold Medal at IIT Indore in 2015 and DAAD-WISE scholarship for working toward the summer internship with RWTH Aachen University, Aachen, Germany, in 2014.



Abishek Sankararaman received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA, in 2019.

Until August 2020, he was a Postdoc with the University of California Berkeley, Berkeley, CA, USA. He is currently a Scientist with Amazon, Seattle, WA, USA. His research interests include the design and analysis of algorithms for a variety of networked systems, including wireless networks, social networks, online platforms, and distributed computing/cloud networks.



Sanjay Shakkottai (Fellow, IEEE) received the Ph.D. degree in ECE from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2002.

He is currently with The University of Texas at Austin, Austin, TX, USA, where he is currently the Temple Foundation Endowed Professor No. 4, and a Professor with the Department of Electrical and Computer Engineering. His research interests include the intersection of algorithms for resource allocation, statistical learning, and

networks, with applications to wireless communication networks and online platforms.

Dr. Shakkottai was the recipient of the NSF CAREER Award in 2004. He was elected as an IEEE Fellow in 2014.