Distributed Linear Bandits under Communication Constraints

Sudeep Salgia [1] **Qing Zhao** [1]

Abstract

We consider distributed linear bandits where M agents learn collaboratively to minimize the overall cumulative regret incurred by all agents. Information exchange is facilitated by a central server, and both the uplink and downlink communications are carried over channels with fixed capacity, which limits the amount of information that can be transmitted in each use of the channels. We investigate the regret-communication trade-off by (i) establishing information-theoretic lower bounds on the required communications (in terms of bits) for achieving a sublinear regret order; (ii) developing an efficient algorithm that achieves the minimum sublinear regret order offered by centralized learning using the minimum order of communications dictated by the information-theoretic lower bounds. For sparse linear bandits, we show a variant of the proposed algorithm offers better regret-communication trade-off by leveraging the sparsity of the problem.

1. Introduction

The tension between learning efficiency and communication cost is evident in many distributed learning problems. If distributed agents can share all their locally obtained information and fully coordinate their actions, the problem effectively reduces to a centralized problem, and the greatest learning efficiency defined by the centralized counterpart is trivially achieved at the price of high communication cost.

What is the minimum amount (in terms of bits) of communications needed to achieve the learning efficiency offered by centralized learning? How one might design a distributed learning algorithm that operates at such an optimal point in the communication-learning efficiency trade-off curve? These fundamental questions have not been adequately addressed in the literature.

1.1. Main Results

In this paper, we address the above questions within the scope of distributed linear bandits. We consider a system of M distributed agents whose actions generate random rewards governed by a common unknown mean $\theta^* \in \mathbb{R}^d$. The agents aim to optimize their actions over time to minimize the overall cumulative regret incurred by all agents over a horizon of length T. Communications across agents are facilitated by a central server. To quantify the communication cost to the bit level, we assume that both the uplink and downlink channels have a capacity of R bits per channel use.

Our main results are twofold. First, we establish an information-theoretic lower bound on the required communications for achieving a sublinear regret order. Second, we develop an efficient algorithm that achieves the optimal regret order offered by centralized learning using the minimum order of communications dictated by the information-theoretic lower bound. For sparse linear bandits, we show a variant of the proposed algorithm offers better regret-communication trade-off by leveraging the sparsity of the problem.

For the distributed linear bandit problem, to achieve the optimal regret order of $\Omega(d\sqrt{MT})$ in both M and T as offered by centralized learning, the agents need to cooperate in learning the underlying reward vector θ^*. In addition to a policy for choosing reward-generating actions at each time, a distributed learning algorithm also includes a communication strategy that governs *when* to communicate and *what* and *how* to send it (i.e., quantization and encoding) over the finite-capacity channels. To minimize the total regret that is accumulating over time and aggregating over the agents while using a minimum amount of communications, the communication strategy needs to work in tandem with action selection to ensure a continual flow of information available at all agent for decision-making.

The key idea of the proposed algorithm is an *progressive learning and sharing* (PLS) structure that systematically coordinates the collective exploration of θ^* and the information sharing of the estimates of θ^* over finite-capacity channels. Specifically, the PLS algorithm progresses as each agent learns and shares one-bit information about θ^*

[1] Department of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. Correspondence to: Sudeep Salgia <ss3827@cornell.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

(per dimension) at a time, starting from the most significant bit in the binary expansion of θ^* in each dimension. This bit-by-bit learning and sharing is structured in interleaving exploration and exploitation epochs with carefully controlled epoch lengths, to achieve both the minimum order of channel usage and the minimum order of cumulative regret.

1.2. Related Work

Communication cost is commonly partitioned into two parts: the size of the message at each information exchange and the frequency of information exchange. As detailed below, these two sides of the same coin have largely been dealt with separately in the literature when developing communication-efficient algorithms for distributed bandit problems. Our work departs from existing studies by taking a holistic view on communication cost and making an initial attempt at characterizing the information-theoretic limit on the communication-learning trade-off in distributed linear bandits.

In the group of work focusing on reducing communication frequency through intermittent information exchange, it is often assumed that the information being transmitted, typically a vector in $\mathbb{R}^d$, can be communicated with infinite precision, which requires a channel with infinite capacity. See, for example, (Hillel et al., 2013; Tao et al., 2019; Agarwal et al., 2021) on discrete bandit problems and (Wang et al., 2019; Ghosh et al., 2021; Huang et al., 2021; Chawla et al., 2022; Amani et al., 2022) on linear bandits. In particular, Wang et al. (2019); Huang et al. (2021) and Amani et al. (2022) proposed algorithms based on batched elimination of arms where poorly performing arms are eliminated at the end of each batch. The batched structure offers a natural way to limit communication to one message exchange per batch. However, there is no constraint on the amount of information that can be transmitted in each batch, allowing numbers to be sent with infinite precision. Furthermore, the downlink cost in Wang et al. (2019) and Huang et al. (2021) grows as $\mathcal{O}((M + d \log \log d) \log(MT))$ and $\mathcal{O}((M + d^2) \log(MT))$ *scalars* respectively. The linear scaling with the number of agents makes this cost significantly worse than the $\mathcal{O}(d \log(MT))$ *bits* offered by PLS. We provide a more detailed comparison with these results in Appendix A.

The other group of work focuses on reducing the size of the message at each information exchange. The frequency of communication is not an concern. The objective is to best *approximate* the information being exchanged via techniques such as quantization and sparsification (see, for example, Konečný et al. (2016); Hanna et al. (2021); Mitra et al. (2022); Suresh et al. (2017)). In particular, Hanna et al. (2021) proposed a quantization scheme to reduce the communication overhead in discrete multi-armed bandit problems at the cost of a small multiplicative constant in the regret. Recently, Mitra et al. (2022) proposed an algorithm for decentralized linear bandits with a finite-capacity uplink channel and an infinite-capacity downlink channel. They developed an adaptive encoding scheme for communication that ensured order-optimal regret for their proposed algorithm. However, with a linear order of message exchanges in T, the total uplink communication cost is $\mathcal{O}(dT)$ bits as opposed to the $\mathcal{O}(d \log T)$ bits of communication cost in our proposed algorithm. Moreover, the results in Mitra et al. (2022) hold only for single agent while ours hold for a distributed setup with multiple agents.

In the context of developing communication-efficient algorithm, another line of related work is Federated Learning (FL) (McMahan et al., 2017). FL aims to collaboratively learn a model by leveraging the data available at all the agents with a focus on ensuring privacy of the data for the participating agents. Developing communication efficient FL algorithms is an active area of research (see Konečný et al. (2016); Liu et al. (2019); Sun et al. (2019); Reisizadeh et al. (2020); Haddadpour et al. (2021); Jhunjhunwala et al. (2021); Hönig et al. (2022) and references therein). Detailed surveys can be found in Tang et al. (2020) and Zhao et al. (2022). These studies focus on the first-order stochastic optimization, which is different from the (zeroth-order) stochastic linear bandits considered in this work, in terms of both action selection strategies and the relevant information that needs to be exchanged.

It is impossible to do full justice to the vast literature on communication-efficient distributed learning. We present above existing studies on distributed bandits that are most relevant to this work. Additional discussions of related work is provided in Appendix A, albeit remaining to be incomplete.

2. Problem Formulation

Consider a system of M distributed agents indexed by $\{1, 2, \ldots, M\}$. The agents face a common stochastic linear bandit model characterized by an unknown mean reward vector $\theta^* \in \mathbb{R}^d$. Specifically, each agent $j \in \{1, 2, \ldots, M\}$ has access to an action set $\mathcal{A} = \{a \in \mathbb{R}^d : \|a\|_2 \leq 1\}$ and chooses to play an action $a_t^j \in \mathcal{A}$ at every time instant t during a time horizon of T instants. When an action $a_t^j \in \mathcal{A}$ is played by agent j at time t, it receives a reward

$$y_t^j = \langle \theta^*, a_t^j \rangle + \eta_t^j,$$

where η_t^j is zero-mean noise that is i.i.d. across time instants and across the agents and satisfies $\log(\mathbb{E}[\exp(\lambda \eta_t^j)]) \leq \lambda^2 \sigma^2 / 2$ for all $\lambda \in \mathbb{R}$, i.e., the noise is σ^2-sub-Gaussian. WLOG, we assume $\|\theta^*\|_2 \leq 1$. It is straightforward to extend it to the case $\|\theta^*\|_2 \leq B$, where B is a known constant, by appropriately scaling the obtained rewards. We point out that the unit ball assumption on the action space is adopted to facilitate a simpler exposition with greater focus on the impact of communication constraints. The algorithm can be easily extended to general action spaces (see Appendix E.1).

Information exchange across the agents goes through a central server. Both the uplink channel (from the agents to the server) and downlink channel (from the server to the agents) have a finite capacity of R bits per channel use, which limits the message size in each information exchange. This model quantifies the cumulative communication cost to the bit level, and represents a more challenging problem than those considered in the literature where communication channel between the server and the agent is assumed to have infinite capacity in at least one direction, if not both.

The design objective is a distributed learning policy consisting of (i) a decision strategy that governs the selection of actions $\{a_t^j\}$ of each agent j at each time t and (ii) a communication strategy that determines when to communicate what and how to send it over the channel via quantization and encoding. The performance of a learning policy is measured in terms of the overall cumulative regret $R(T)$ and the cumulative communication cost $C(T)$ incurred by the policy. The overall cumulative regret is given by

$$R(T) = \sum_{j=1}^{M} \sum_{t=1}^{T} \left[\max_{a \in \mathcal{A}} \langle \theta^*, a \rangle - \langle \theta^*, a_t^j \rangle \right]. \qquad (1)$$

The communication cost $C(T)$ is measured using $C_{\mathrm{u}}(T)$ and $C_{\mathrm{d}}(T)$, the number of bits transmitted on the uplink channel (i.e., by any agent to the server) and that on the downlink channel (i.e., the average number of bits transmitted by the server to an agent), respectively.

The learning and communication efficiency of a learning policy is measured against the benchmarks. In particular, the cumulative regret is lower bounded by $\Omega(d\sqrt{MT})$, which is the optimal regret order in a centralized setting with total MT reward observations centrally available for learning. In Sec. 5, we establish an information-theoretic lower bound on the communication cost required for achieving a sublinear regret order.

3. Progressive Learning and Sharing

In this section, we present the Progressive Learning and Sharing (PLS) algorithm. We start in Sec. 3.1 with the basic structure of the algorithm followed by a detailed implementation in Sec. 3.2.

3.1. The Basic Structure of PLS

In PLS, learning and information sharing progress along the binary expansion of θ^* in each dimension, starting from the most significant bit. Below we present separately the information sharing and learning components of PLS.

3.1.1. Progressive Information Sharing

In PLS, the unknown reward vector θ^* is learnt with increasing accuracy, one bit at a time, as the algorithm progresses. Once the next bit in the binary expansion[1] of θ^* in each of the d coordinates is learnt with sufficient accuracy, the agents transmit their estimates of this bit to the central server, which aggregates the estimates and broadcast the aggregated estimate of the new bit to the agents for subsequent exploitation and further exploration of θ^*.

This progressive sharing mechanism can be seamlessly integrated with regret minimization to achieve both minimum regret order and minimum channel usage. Specifically, since only 1 bit of information is shared per coordinate in each information exchange, it suffices to send d bits in each transmission, achieving the benefit of having small messages. Furthermore, one can note that any reward-maximizing action taken based on an estimate $\hat{\theta}$ incurs a regret proportional to the estimation error $\|\hat{\theta} - \theta^*\|_2$. Consequently, an estimation error of $\mathcal{O}(1/\sqrt{T})$ is sufficient to ensure an order-optimal regret, implying that it is sufficient to estimate each coordinate of θ^* up to an accuracy of $\mathcal{O}(1/\sqrt{T})$. Since sending the first r bits of the binary representation ensures an error of no more than 2^{-r}, transmitting the first $\mathcal{O}(\log T)$ bits of the binary representation is sufficient to achieve the required accuracy. As a result, infrequent communication with a total of $\mathcal{O}(\log T)$ rounds can transmit all relevant information about θ^*.

3.1.2. Progressive Collaborative Learning

The progressive learning component of PLS is carried out in two stages: an initial stage for estimating the norm of θ^* followed by a refinement stage with interleaving exploration and exploitation.

In the initial norm estimation stage, the goal is to estimate, within a multiplicative factor, the norm $\|\theta^*\|_2$ of the underlying mean reward. This procedure is purely exploratory in nature. The collaborative exploration across agents is carried out in epochs with exponentially growing epoch lengths. At the end of each epoch, a threshold-based termination test is employed to determine whether the required estimation accuracy has been reached, which terminates the norm estimation stage. Information exchange occurs at the end of each epoch, and the exponentially growing epoch length ensures a low communication frequency.

The norm estimation stage serves multiple purposes. First,

[1] While the algorithm learns an one bit at a time, it does not necessarily imply that the bit sequence learnt corresponds to the binary expansion.

it allows PLS to be adaptive to the norm of θ^* through the threshold-based termination rule. Specifically, it provides sufficient initial exploration with sufficiency automatically adapted to $\|\theta^*\|_2$ to ensure the estimation error is small enough for subsequent exploitation in the refinement stage. Second, this initial norm estimate sets the dynamic range of subsequent estimates of θ^* to be used in the differential quantization for subsequent information sharing. Third, the estimate of $\|\theta^*\|_2$ is also used to control the length of the exploitation epoch in the refinement stage to balance the exploration-exploitation trade-off.

In the refinement stage, the estimate of θ^* obtained in the norm estimation stage is further refined. Similar to the norm estimation stage, the refinement stage also proceeds in epochs. The difference is that each epoch in the refinement stage consists of an exploration sub-epoch followed by an exploitation one. The exploration sub-epochs are for continual learning of θ^*, one bit in each sub-epoch. The exploitation sub-epochs are to maximize rewards at each agent by playing the best action based on the current estimate of θ^*. The lengths of the exploration and exploitation sub-epochs are both growing exponentially, but at different rates to carefully balance the exploration-exploitation trade-off. Information sharing is carried out only at the end of each exploration sub-epoch for the newly learned bit of θ^*. The refinement stage with its interleaved exploration and exploitation continues until the end of the time horizon.

3.2. Detailed Description of PLS

In this section, we dive into the details of PLS.

3.2.1. Progressive Collaborative Learning

Norm Estimation Stage: This stage proceeds in purely exploratory epochs. During an epoch k, each agent plays each unit vector in an orthonormal basis[2] of $\mathbb{R}^d$ for s_k times. Each agent j computes the sample mean of the observed rewards for each basis vector to obtain an estimate $\hat{\theta}_k^{(j)}$ of the underlying vector θ^*. This estimate $\hat{\theta}_k^{(j)}$ is clipped to within a radius of $R_k + B_k$, quantized using a stochastic quantizer with resolution α_k and sent to the server by the agent. The process is repeated in every epoch until the agents receive a message from the server to terminate. We defer the details of the clipping and quantization steps to the Sec. 3.2.2 that describes the communication strategy. All policy parameters are specified at the end of the section.

At the server, upon receiving the estimates from the agents, the server averages them to obtain a combined estimate $\hat{\theta}_k^{(\text{SERV})}$. The server compares the norm of this estimate to a threshold $4\tau_k$. If the norm exceeds the threshold, the server sends a message to the agents to terminate the norm estimation stage. Otherwise, the server and the agents proceed into the next epoch. The value of τ_k is chosen to be an upper bound on the estimation error of θ^* at the end of the k^{th} epoch, allowing PLS to estimate $\|\theta^*\|_2$ within a multiplicative factor at the end of the norm estimation stage.

The pseudo code for the norm estimation stage is given in Algorithms 1 and 2.

Algorithm 1 Norm Estimation: Agent $j \in \{1, 2, \ldots, M\}$

```
Set $k \leftarrow 1$
while True do
    Play each basis vector $s_k$ times and compute the sample mean $\hat{\theta}_k^{(j)}$
    $\tilde{\theta}_k^{(j)} \leftarrow \textsc{Clip}(\hat{\theta}_k^{(j)}, R_k + B_k)$
    $Q(\tilde{\theta}_k^{(j)}) \leftarrow \textsc{StoQuant}(\tilde{\theta}_k^{(j)}, \alpha_k, R_k + B_k)$
    Send $Q(\tilde{\theta}_k^{(j)})$ to the server
    if received terminate from server then
        break
    else
        $k \leftarrow k + 1$
    end if
end while
```

Algorithm 2 Norm Estimation: The Server

```
Set $k \leftarrow 1$
while True do
    Compute $\hat{\theta}_k^{(\text{SERV})} = \frac{1}{M}\sum_{j=1}^{M} Q(\tilde{\theta}_k^{(j)})$
    if $\tau_k \leq \frac{1}{4}\|\hat{\theta}_k^{(\text{SERV})}\|$ then
        Server sends terminate to all agents
        break
    else
        $k \leftarrow k + 1$
    end if
end while
```

Refinement Stage: This stage also proceeds in epochs, starting with the epoch index at which Norm Estimation stage terminated. Each epoch k during Refinement begins with an exploration sub-epoch where, similar to Norm Estimation, each agent obtains their estimate of θ^*, $\hat{\theta}_k^{(j)}$, by playing each of the basis vectors s_k times. At the end of the sub-epoch, the agents share the next bit learnt during this time by transmitting $\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1}$ to the server after appropriate clipping and quantization. Here $\bar{\theta}_{k-1}$ denotes the current estimate of θ^* available to the agents after $k-1$ epochs. This differential quantization allows the agents to share only the "new bit" learnt during the exploration sub-epoch. As a response, the agents receive $Q(\hat{\theta}_k^{(\text{SERV})})$, a quantized version of the udpate, from the server which is used to refine their

[2] The basis is chosen *a priori* and known to all the agents and the server. It can be any orthonormal basis of $\mathbb{R}^d$.

estimate of θ^* to $\bar{\theta}_k = \bar{\theta}_{k-1} + Q(\hat{\theta}_k^{(\text{SERV})})$. The exploitation sub-epoch follows the communication round where all agents play the unit vector along $\bar{\theta}_k$ throughout the t_k time steps of the sub-epoch. The refinement stage continues by proceeding into the next epoch and repeating the process until the end of the time horizon.

The steps at the server in this stage are similar to those in the norm estimation stage. In particular, the server collects estimates from the agents at the end of the exploration sub-epoch, computes the mean $\hat{\theta}_k^{(\text{SERV})}$ and then broadcasts the differential update $\hat{\theta}_k^{(\text{SERV})} - \bar{\theta}_{k-1}$ after passing it through a deterministic quantizer with resolution β_k. A pseudo code for the refinement stage is given in Algorithms 3 and 4.

Algorithm 3 Refinement: Agent $j \in \{1, 2, \ldots, M\}$

```
Input: The epoch index at the end of Norm Estimation stored as $k_0$, $\bar{\theta}_{k_0-1} \leftarrow 0, k \leftarrow k_0$
while budget is not exhausted do
    Play each basis vector $s_k$ times and compute the sample mean $\hat{\theta}_k^{(j)}$
    $\tilde{\theta}_k^{(j)} \leftarrow \textsc{Clip}(\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1}, R_k + B_k)$
    $Q(\tilde{\theta}_k^{(j)}) \leftarrow \textsc{StoQuant}(\tilde{\theta}_k^{(j)}, \alpha_k, R_k + B_k)$
    Send $Q(\tilde{\theta}_k^{(j)})$ to the server
    Receive $Q(\hat{\theta}_k^{(\text{SERV})})$ from the server
    $\bar{\theta}_k \leftarrow \bar{\theta}_{k-1} + Q(\hat{\theta}_k^{(\text{SERV})})$
    if $k = k_0$ then
        Set $\mu_0 \leftarrow \|\bar{\theta}_k\|_2$
    end if
    Play the action $a = \bar{\theta}_k / \|\bar{\theta}_k\|$ for the next $t_k$ rounds.
    $k \leftarrow k + 1$
end while
```

Algorithm 4 Refinement: The Server

```
Input: The epoch index at the end of Norm Estimation stored as $k_0$, $\bar{\theta}_{k_0-1} \leftarrow 0, k \leftarrow k_0$
while time horizon $T$ is not reached do
    Receive $Q(\tilde{\theta}_k^{(j)})$ from all the agents
    Compute $\hat{\theta}_k^{(\text{SERV})} = \bar{\theta}_{k-1} + \frac{1}{M} \sum_{j=1}^{M} Q(\tilde{\theta}_k^{(j)})$
    $Q(\hat{\theta}_k^{(\text{SERV})}) \leftarrow \textsc{DetQuant}(\hat{\theta}_k^{(\text{SERV})} - \bar{\theta}_{k-1}, \beta_k, B_k + \tau_k)$ and broadcasts it to all agents
    $k \leftarrow k + 1$
end while
```

Setting Policy Parameters: We now specify the values of parameters used in PLS. For an epoch k, the length of the exploration (sub-)epoch, s_k, is set to $\lceil 40\sigma^2 d \log(8MK/\delta) 4^k \rceil$ and that of the exploitation one is set to $t_k := \lceil M s_k^2 \mu_0^2 \rceil$. In the above definitions, K denotes the maximum possible number of epochs in the algorithm and is defined as $K := \max\{k \in \mathbb{N} : 40\sigma^2 d \log(8Mk/\delta)(4^k - 4) \leq T\} = \mathcal{O}(\log T)$. In the definition of t_k, μ_0 is the estimate of $\|\theta^*\|_2$ obtained at the end of the norm estimation stage. The exponential lengths of the epoch designed for the bit by bit progressive learning are evident from the above choices. This choice of the lengths also allows PLS to address the exploration-exploitation trade-off by balancing the regret incurred during the exploration and exploitation sub-epochs.

The threshold τ_k and sequence R_k are set based upon high probability bounds on $\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\|_2$ and $\|\hat{\theta}_k^{(j)} - \theta^*\|_2$ respectively that simultaneously hold for all agents. In particular, τ_k is set to $3 \cdot 2^{-(k+1)}/\sqrt{M}$ and $R_k := 2^{-k}$. Notice that this choice of R_k echoes the progressive learning feature, allowing the agents to learn θ^*, one bit at a time. The sequence B_k bounds the error $\|\bar{\theta}_{k-1} - \theta^*\|_2$ and is set to $5\tau_k$ for $k \geq 2$ with $B_1 = 1$. Lastly, the resolution parameter sequences α_k and β_k are defined as $\alpha_k := \alpha_0 \sigma \sqrt{d}/\sqrt{s_k}$ and $\beta_k = \beta_0 \tau_k$ for some numerical constants $\alpha_0, \beta_0 < 1$. The constants α_0 and β_0 control the message size associated with uplink and downlink communication respectively.

3.2.2. Progressive Information Sharing

The communication protocol of PLS consists of two steps : clipping and quantizing the vector to be sent to reduce the size of message being transmitted and encoding the quantized value to send it over the communication channel.

Clipping and Quantization: This step employs well-known sub-routines described below to map a vector to a low resolution, quantized version of itself.

- $\textsc{Clip}(x, r)$ is a simple routine that takes input a vector x and clips it within a ℓ_2 ball of radius r. Mathematically, the routine returns the value $x \cdot \min\{1, r/\|x\|\}$.
- $\textsc{StoQuant}(y, \varepsilon, r)$ returns the quantized version of a scalar y using the popular approach of stochastic quantization. Specifically, the interval $[-r, r]$ is divided into $l_\varepsilon = \lceil 2r/\varepsilon \rceil$ intervals of equal length and indexed from 1 to l_ε. The value y is quantized to one of the end points of the intervals to which it belongs in a randomized manner with the probability inversely proportional to the distance from y. In particular, the stochastic quantizer outputs $Q_s(y)$ given by

$$Q_s(y) = \begin{cases} b_{l-1} & \text{w.p. } b_l - y, \\ b_l & \text{otherwise.} \end{cases}$$

In the above expression, $b_m = r\left(\dfrac{2m}{l_\varepsilon} - 1\right)$ for $m = 1, 2, \ldots, l_\varepsilon$ and $l = \{m : b_{m-1} \leq y < b_m\}$. With a slight abuse of notation, we also use $\textsc{StoQuant}(x, \varepsilon', r)$ to denote stochastic quantization of a vector x. In the case of a vector, all the coordinates are quantized as mentioned above with $\varepsilon = \varepsilon'/\sqrt{d}$.

- $\textsc{DetQuant}(y, \varepsilon, r)$ is also a quantization routine similar to $\textsc{StoQuant}(y, \varepsilon, r)$ with the only difference that the output of this routine is deterministic. Specifically, the deterministic quantizer outputs

$$Q_d(y) = \begin{cases} b_{l-1} & \text{if } |b_l - y| > |b_{l-1} - y|, \\ b_l & \text{otherwise.} \end{cases}$$

Once again we overload the definition with a vector analogue $\textsc{DetQuant}(x, \varepsilon', r)$ where each coordinate is deterministically quantized with $\varepsilon = \varepsilon'/\sqrt{d}$.

The two different quantization schemes used in this step serve their own, different purposes in algorithm. PLS employs stochastic quantization for uplink communication which allows it to exploit the concentration properties of the zero mean noise added by the quantization. Consequently, it enables to completely leverage the access to observations from M different agents to achieve the speed up proportional to the number of agents. A deterministic quantization scheme in its place would have resulted in accumulation of errors and consequently prevented the speedup with the number of agents. On the other hand, the deterministic quantization used for downlink transmission ensures that all the agents have the same estimate of θ^* and consequently the same value of μ_0, the estimate of $\|\theta^*\|_2$. Since μ_0 governs the length of the exploitation sub-epoch, a common value shared between the agents helps maintain the synchronization across different epochs and agents.

Encoding: As described above, both the quantization routines used in PLS when run with accuracy parameter ε, quantize each coordinate axis into l_ε intervals resulting in $l_\varepsilon + 1$ possible values for the quantized version of each coordinate. The encoding step maps these $(l_\varepsilon + 1)^d$ possible values of quantized vectors to different messages that can be sent over the communication channel. We use a common encoding strategy for the uplink and downlink channels.

PLS sends each coordinate of the quantized version, one by one, using the variable-length encoding strategy, unary coding (Cover & Thomas, 2006). In particular, each coordinate is represented by a header followed by a sequence of 1's whose length is equal to the absolute value of the coordinate. The header is 3 bits long where the first and the third bit are both 0 and the second bit represents the sign of the value, where 0 & 1 imply negative and positive values respectively. As an example, in this encoding scheme, the numbers -3 and 5 are represented as 000111 and 01011111 respectively.

4. Performance Analysis

In this section, we show that PLS achieves the order-optimal regret of $\mathcal{O}(d\sqrt{MT})$ up to logarthmic factors with a communication cost that matches the order of information-theoretic lower bound established in Sec. 5.

4.1. Regret Analysis

The following theorem characterizes the regret of PLS.

Theorem 4.1. *Consider the distributed stochastic linear bandit setting described in Sec. 2. If PLS is run with parameters as described in Sec. 3.2 for T time instants, then the following relation holds with probability at least $1 - \delta$,*

$$R_{PLS}(T) \leq Cd\sqrt{MT} \log\left(\frac{8MK}{\delta}\right) \log\left(9\sqrt{MT}\right) + \mathcal{O}(\log T),$$

for some constant $C > 0$, independent of d, M, δ and T.

Theorem 4.1 establishes that the regret incurred by PLS is $\tilde{\mathcal{O}}(d\sqrt{MT})$ which matches the lower bound for a centralized setting with MT queries up to logarithmic factors. This implies that PLS explores the achievability of communication-learning trade-off at the frontier of optimal learning performance.

We here provide a sketch of the proof for Theorem 4.1. We begin the proof by decomposing the regret incurred by PLS as follows : $R_{\text{PLS}}(T) = R_{\text{NE}}(T) + R_{\text{REF}}(T)$, where $R_{\text{NE}}(T)$ and $R_{\text{REF}}(T)$ denote the regret incurred during the norm estimation and the refinement stages respectively. The bound on regret incurred by PLS is obtained by separately bounding each of above the two terms in the decomposition. The following lemma provides a bound on the regret incurred during the norm estimation stage.

Lemma 4.2. *Consider the Norm Estimation stage described in Alg. 1 and 2. If it is run for at most T time instants in a distributed setup with M agents and $\|\theta^*\| \leq 1$, then the regret incurred during this stage satisfies the following relation with probability at least $1 - \delta$,*

$$R_{\text{NE}}(T) \leq Cd(1 + \sigma^2)\sqrt{MT} \log(1/\delta') \log_2\left(9\sqrt{MT}\right) + 2d \log_2(9\sqrt{MT}),$$

where $\delta' = \delta/8MK$ and $C > 0$ is a constant independent of d, M, δ and T.

Since the Norm Estimation stage is based on pure exploration, we upper bound $R_{\text{NE}}(T)$ by $2\|\theta^*\|_2$ times the duration of the norm estimation stage, where $2\|\theta^*\|_2$ corresponds to the trivial bound on the instantaneous regret. The central step in the proof of the above lemma is bounding the duration of the norm estimation stage. For this part, we first establish a $\mathcal{O}(1/\|\theta^*\|_2^2)$ bound on the duration based on our threshold test which determines when the stage terminates. However, the fixed length of the time horizon dictates a hard upper bound of T on the duration of this stage. The proof is completed by taking a minimum of these two bounds to bound the duration of the norm estimation stage followed optimizing over $\|\theta^*\|_2$ to obtain the tightest bound.

To bound $R_{\text{REF}}(T)$, we separately bound the regret incurred during the exploration and exploitation sub-epochs. The regret incurred during the exploration sub-epoch of the k^{th} epoch is bounded by $2\|\theta^*\|_2$ times ds_k, the duration of the exploration sub-epoch, similar to the proof of Lemma 4.2. The regret incurred during the exploitation sub-epoch is bounded using the following lemma, which is similar to Lemma 3.6 in Rusmevichientong & Tsitsiklis (2010).

Lemma 4.3. *If $\hat{\theta}$ is an estimate of a vector θ such that $\|\hat{\theta} - \theta\|_2 \leq \tau \leq \|\theta\|_2$, then instantaneous regret incurred by an algorithm that plays the action $a = \hat{\theta}/\|\hat{\theta}\|_2$ on a stochastic linear bandit instance with true underlying vector θ can be bounded by $\tau^2/\|\theta\|$.*

Lemma 4.3 implies that even when the estimation error is $\nu\|\theta^*\|_2$, for $\nu \in [0, 1]$ the regret incurred by the algorithm scales as $\nu^2\|\theta^*\|_2$. This is a crucial fact that helps balance the exploration-exploitation trade-off. In particular, the lengths of the exploration and exploitation epochs in PLS are designed based on the result of this lemma. The estimation error at the end of exploration sub-epoch in epoch k satisfies $\mathcal{O}(s_k^{-1/2})$ while the regret incurred during the sub-epoch satisfies $\mathcal{O}(s_k\|\theta^*\|_2)$. Based on the above lemma, the regret incurred during the corresponding exploitation sub-epoch of length $t_k = \mathcal{O}(s_k^2\|\theta^*\|_2^2)$ is $\mathcal{O}(\frac{1}{s_k\|\theta^*\|_2} \cdot t_k) = \mathcal{O}(s_k\|\theta^*\|_2)$ matching the regret incurred during the exploration phase. This is the fundamental mechanism that provides the necessary balance between exploration and exploitation in PLS allowing it to achieve optimal-order regret. Moreover, this also provides additional insight into the novel choice of the length exploitation epoch based on the norm of θ^* in PLS.

The final bound on $R_{\text{REF}}(T)$ is obtained by noting each epoch is $\mathcal{O}(s_k^2 + s_k) = \mathcal{O}(16^k)$ steps long implying a total of $\mathcal{O}(\log T)$ epochs. The detailed proofs of all the Lemmas and the Theorem can be found in Appendix B.

4.2. Communication Cost

The communication cost incurred by PLS is characterized in the following theorem.

Theorem 4.4. *Consider the distributed stochastic linear bandit setting described in Sec. 2. If PLS is run with parameters as described in Sec. 3 for a time horizon of T, then the uplink and the downlink communication costs (in bits) incurred by PLS, i.e. $C_u(T)$ and $C_d(T)$, satisfy $\mathcal{O}\left(\frac{d}{\alpha_0}\log T\right)$ and $\mathcal{O}\left(\frac{d}{\beta_0}(\log M + \log T)\right)$ respectively.*

Thus, Theorem 4.4 in conjunction with Theorem 4.1 shows that PLS incurs the order-optimal regret while simultaneously achieving the order-optimal communication cost matching the lower bound in Sec. 5. Hence, PLS further reduces the communication cost as compared to communication-efficient algorithms proposed in Wang et al. (2019); Huang et al. (2021); Amani et al. (2022) while maintaining the optimal regret performance. The proof of the above theorem revolves around the following lemma.

Lemma 4.5. *Consider the communication scheme of PLS outlined in Sec. 3.2.2. If PLS is run with parameters as described in Sec. 3.2, then any message exchanged between the server and an agent during PLS is at most $\mathcal{O}(d)$ bits.*

The bound on uplink communication cost $C_{\text{u}}(T)$ immediately follows by noting that there are at most $K = \mathcal{O}(\log T)$ epochs, or equivalently, communication rounds in PLS. The additional $\mathcal{O}(d \log M)$ term in $C_{\text{d}}(T)$ is incurred when the server sends the initial estimate of θ^* to all the agents. The details of the proof along with proof of Lemma 4.5 are provided in Appendix B.

Remark 4.6. Based on Lemma 4.5, it can immediately concluded that a capacity of $R = \mathcal{O}(d)$ bits for both the uplink and the downlink channel suffices for PLS to achieve order-optimal regret performance. Some other studies like Suresh et al. (2017) and Mitra et al. (2022) have also proposed encoding schemes which result in message sizes of $\mathcal{O}(d)$ bits. Suresh et al. (2017) proposed an encoding scheme for distributed mean estimation using the well-known variable length encoding schemes such as Huffman and Arithmetic encoding. It first constructs a histogram over the different $l_\varepsilon + 1$ values in the qunatized version of a vector followed by a Huffman tree based on that histogram. During each communication round, the sender first sends the corresponding Huffman tree followed by the message encoded using that. Compared to such variable-length schemes, our proposed scheme is easier to implement, both in terms of memory and computation, and also avoids overheads like sending the Huffman tree. The IC-Lin-UCB algorithm proposed in Mitra et al. (2022) uses an encoding scheme based on constructing ε-cover of hyperspheres in $\mathbb{R}^d$ at every time instant. The computational cost of constructing such covering sets grows exponentially with the dimension, rendering the approach infeasible even for problems of moderate dimensionality. On the other hand, the computational cost of the encoding scheme in PLS grows linearly with dimension, making PLS an attractive option even for high dimensional problems.

Remark 4.7. The $\mathcal{O}(d \log M)$ term in the downlink communication cost can be interpreted as the cost incurred by the server to facilitate information exchange since the data is distributed across M agents. In other words, the server provides each agent with the additional information learnt from the other agents through these $\mathcal{O}(d \log M)$ bits, leading to *collaboration* among them. In absence of transmission of these bits, the problem would reduce to M independent agents trying to learn a linear bandit model and incurring an overall regret of $\mathcal{O}(dM\sqrt{T})$ that grows linearly with M. Thus, it can also be interpreted as the cost associated with having a sublinear regret with respect to the number of agents M. Similarly, $\mathcal{O}(d \log T)$ term corresponds to the cost associated with having a sublinear regret with respect to the time horizon.

5. Lower Bound on Communication Cost

In this section, we explore the converse result for the communication-learning trade-off. In particular, the following theorem establishes information theoretic lower bounds in terms of actual number of bits that need to be transmitted by the clients and the server over the channel for any distributed algorithm to achieve sublinear regret.

Theorem 5.1. *Consider the distributed linear bandit instance with M agents described in Section 2. Any distributed algorithm that incurs an overall cumulative regret that is order optimal in both T and M with probability at least $2/3$ needs to transmit at least $\Omega(d \log(MT))$ bits of information over the downlink channel and $\Omega(d \log T)$ bits of information over the uplink channel.*

The lower bounds established in the above theorem match the achievability results for PLS shown in the previous section. We would like to emphasize that PLS is the first algorithm for distributed linear bandits for which communication cost incurred matches the order of information theoretic lower bounds in terms of actual number of bits transmitted over the channel. Additionally, PLS simultaneously attains order-optimal regret guarantees. Thereby, the above theorem along with the performance guarantees of PLS provides a holistic view of the communication-learning efficiency trade-off in distributed linear bandits. The proof of the theorem follows from an application of Fano's inequality along with bounds on metric entropy. A detailed proof of the above theorem is provided in Appendix C.

6. Leveraging Sparsity

We now consider a variant of the original problem where the underlying reward vector θ^* is known to satisfy an additional sparsity constraint. In particular, it is known that the number of non-zero elements in θ^* are no more than $s \ll d$, i.e., $\|\theta^*\|_0 \leq s$. For this setup, we propose Sparse-PLS, a variant of PLS, to leverage the sparsity of θ^* to further reduce the communication cost.

Sparse-PLS makes two modifications to the original PLS algorithm to leverage the sparsity of θ^*. The first modification is made to the set of actions played during an exploration epoch. Specifically, Sparse-PLS replaces the orthonormal basis of R^d with a set of actions, $\mathcal{B}_s$, that spans only a subspace of R^d. $\mathcal{B}_s$ consists of $m = \mathcal{O}(s \log(d/\delta)) \ll d$ vectors drawn independently from the set $\{-1/\sqrt{d}, +1/\sqrt{d}\}^d$. It is ensured that all the agents use the same random set $\mathcal{B}_s$ via a common random seed. This modification offers a two-fold advantage. First, it helps reduce the message size required for uplink communication as it is sufficient to transmit these noisy projections in $\mathbb{R}^m$, where $m \ll d$, in order to recover the original sparse vector θ^* (Candès et al., 2006; Candes & Tao, 2006; 2007). Second, since the regret incurred in PLS is proportional to length of the exploration epochs, this modification allows Sparse-PLS to replace a factor of the actual dimension of the vector with the level of sparsity in the regret bounds, making it smaller (See Theorem 6.1). The second modification is made to the process to estimate θ^* at the server. Sparse-PLS employs the LASSO estimator (Tibshirani, 1996; Bickel et al., 2009) at the server to obtain a sparse estimate of θ^* from the noisy projections sent by the agents. Please refer to the supplementary material for a pseudo-code and additional details about Sparse-PLS.

The following theorem characterizes the performance of Sparse-PLS,in terms of both regret and communication cost.

Theorem 6.1. *Consider the distributed stochastic linear bandit setting described in Sec. 2 with an additional assumption of sparsity on the underlying mean reward, i.e., $\|\theta^*\|_0 \leq s$. If Sparse-PLS is run for a time horizon of T, then the regret incurred by Sparse-PLS satisfies*

$$R_{\textit{Sparse-PLS}}(T) \leq C\sqrt{sdMT} \log\left(MK/\delta\right) \log\left(\sqrt{MT}\right),$$

with probability at least $1 - \delta$ for some $C > 0$, independent of s, d, M and T. Moreover, the uplink and downlink communication costs, $C_u(T)$ and $C_d(T)$ are no more than $\mathcal{O}(s \log T)$ and $\mathcal{O}(d(\log M + \log T))$ bits respectively.

As it can be noted from the above theorem, Sparse-PLS replaces a factor of dimension d with the sparsity level s, or equivalently the effective dimension, in the regret bound matching the optimal-order regret bounds (Abbasi-Yadkori et al., 2012). Furthermore, it also reduces $C_{\mathrm{u}}(T)$ from $\mathcal{O}(d \log T)$ to $\mathcal{O}(s \log T)$ demonstrating its ability to leverage the inherent sparsity to reduce communication and regret. We refer the reader to Appendix D for a detailed proof.

7. Conclusion

In this work, we investigated the communication-learning trade-off in distributed learning setups within the scope of distributed linear bandits. We proposed a novel algorithm, called Progressive Learning and Sharing (PLS), that learns and shares the information about the unknown reward vector progressively, one bit at a time. We showed that PLS incurs order-optimal regret using a uplink communication of $\mathcal{O}(d \log T)$ bits and downlink communication of $\mathcal{O}(d(\log M + \log T))$ bits. We also established matching information-theoretic lower bounds on the communication cost for any algorithm with sublinear regret. Lastly, for sparse linear bandits, we showed that a variant of the proposed algorithm offers better communication-learning trade-off by leveraging the sparsity of the problem.

8. Acknowledgements

This work was supported in part by by the Government of Israel - Israeli Ministry of Defense, Mission to the USA. We would also like to thank the anonymous reviewers for their constructive comments that have helped improve the exposition of this work.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 2011. ISBN 9781618395993.

Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. Online-to-confidence-set conversions and application to sparse stochastic bandits. In Lawrence, N. D. and Girolami, M. (eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 1–9, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. URL https://proceedings.mlr.press/v22/abbasi-yadkori12.html.

Acharya, J., Canonne, C. L., Sun, Z., and Tyagi, H. The role of interactivity in structured estimation. In *Conference on Learning Theory*, pp. 1328–1355. PMLR, 2022.

Agarwal, M., Aggarwal, V., and Azizzadenesheli, K. Multi-Agent Multi-Armed Bandits with Limited Communication, 2021. URL http://arxiv.org/abs/2102.08462.

Amani, S., Lattimore, T., György, A., and Yang, L. F. Distributed Contextual Linear Bandits with Minimax Optimal Communication Cost, 2022. URL http://arxiv.org/abs/2205.13170.

Ball, K. *An Elementary Introduction to Modern Convex Geometry*. Cambridge, 1997.

Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008. ISSN 01764276. doi: 10.1007/s00365-007-9003-x.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009. ISSN 00905364. doi: 10.1214/08-AOS620.

Bistritz, I. and Leshem, A. Game of Thrones: Fully Distributed Learning for Multi-Player Bandits, 2021.

Candes, E. and Tao, T. Near Optimal Signal Recovery From Random Projections : Universal Encoding Strategies. In *IEEE Transactions on Information Theory*, volume 52, pp. 5406–5425, 2006. URL https://statweb.stanford.edu/$\sim$candes/papers/OptimalRecovery.pdf.

Candes, E. and Tao, T. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, dec 2007. ISSN 00905364. doi: 10.1214/009053606000001523.

Candès, E. J., Romberg, J. K., and Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59 (8):1207–1223, 2006. ISSN 00103640. doi: 10.1002/cpa.20124.

Carpentier, A. and Munos, R. Bandit Theory meets Compressed Sensing for high-dimensional Stochastic Linear Bandit. In *Journal of Machine Learning Research*, volume 22, pp. 190–198, 2012.

Chawla, R., Sankararaman, A., Ganesh, A., and Shakkottai, S. The Gossiping Insert-Eliminate Algorithm for Multi-Agent Bandits, 2020. URL http://arxiv.org/abs/2001.05452.

Chawla, R., Sankararaman, A., and Shakkottai, S. Multi-Agent Low-Dimensional Linear Bandits. *IEEE Transactions on Automatic Control*, 2022. ISSN 15582523. doi: 10.1109/TAC.2022.3179521.

Chen, Y., Wang, Y., Fang, E. X., Wang, Z., and Li, R. Nearly Dimension-Independent Sparse Linear Bandit over Small Action Spaces via Best Subset Selection. *Journal of the American Statistical Association*, pp. 1–31, 2022. ISSN 0162-1459. doi: 10.1080/01621459.2022.2108816.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214. JMLR Workshop and Conference Proceedings, 2011.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Dani, V., Hayes, T. P., and Kakade, S. M. The price of bandit information for online optimization. In *Advances in*

Neural Information Processing Systems 20 - Proceedings of the 2007 Conference, 2008. ISBN 160560352X.

Diakonikolas, I., Grigorescu, E., Li, J., Natarajan, A., Onak, K., and Schmidt, L. Communication-efficient distributed learning of discrete distributions. *Advances in Neural Information Processing Systems*, 30, 2017.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Zhang, Y. Optimality guarantees for distributed statistical estimation, 2014. URL http://arxiv.org/abs/1405.0782.

Ghosh, A., Sankararaman, A., and Ramchandran, K. Adaptive Clustering and Personalization in Multi-Agent Stochastic Linear Bandits, 2021. URL http://arxiv.org/abs/2106.08902.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pp. 2350–2358. PMLR, 2021.

Hanna, O. A., Yang, L. F., and Fragouli, C. Solving Multi-Arm Bandit Using a Few Bits of Communication. *International Conference on . . .*, 2021. URL http://arxiv.org/abs/2111.06067.

Hao, B., Lattimore, T., and Wang, M. High-dimensional sparse linear bandits. In *Advances in Neural Information Processing Systems*, volume 2020-Decem, 2020.

Hillel, E., Karnin, Z., Koren, T., Lempel, R., and Somekh, O. Distributed exploration in Multi-Armed Bandits. In *Advances in Neural Information Processing Systems*, 2013.

Hönig, R., Zhao, Y., and Mullins, R. DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8852–8866. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/honig22a.html.

Huang, R., Wu, W., Yang, J., and Shen, C. Federated Linear Contextual Bandits. In *Advances in Neural Information Processing Systems*, volume 32, pp. 27057–27068, 2021. ISBN 9781713845393.

Jhunjhunwala, D., Gadhikar, A., Joshi, G., and Eldar, Y. C. Adaptive quantization of model updates for communication-efficient federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3110–3114. IEEE, 2021.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. A short note on concentration inequalities for random vectors with subgaussian norm, 2019. URL https://arxiv.org/abs/1902.03736.

Kalathil, D., Nayyar, N., and Jain, R. Decentralized learning for multi-player multi-armed bandits. In *Proceedings of the IEEE Conference on Decision and Control*, pp. 3960–3965, 2012. doi: 10.1109/CDC.2012.6426587.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency, 2016. URL http://arxiv.org/abs/1610.05492.

Korda, N., Szorenyi, B., and Li, S. Distributed clustering of linear bandits in peer to peer networks. In *33rd International Conference on Machine Learning, ICML 2016*, volume 3, pp. 1966–1980, 2016. ISBN 9781510829008.

Landgren, P., Srivastava, V., and Leonard, N. E. On distributed cooperative decision-making in multiarmed bandits. In *2016 European Control Conference, ECC 2016*, pp. 243–248, 2017. ISBN 9781509025916. doi: 10.1109/ECC.2016.7810293.

Liu, K. and Zhao, Q. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010. doi: 10.1109/TSP.2010.2062509.

Liu, Y., Kang, Y., Zhang, X., Li, L., Cheng, Y., Chen, T., Hong, M., and Yang, Q. A Communication Efficient Collaborative Learning Framework for Distributed Features, 2019. URL http://arxiv.org/abs/1912.11187.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Mitra, A., Hassani, H., and Pappas, G. Exploiting Heterogeneity in Robust Federated Best-Arm Identification, 2021. URL http://arxiv.org/abs/2109.05700.

Mitra, A., Hassani, H., and Pappas, G. J. Linear Stochastic Bandits over a Bit-Constrained Channel, 2022.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031. PMLR, 2020.

Rigollet, P. and Hütter, J.-C. High Dimensional Statistics Lecture Notes, 2017.

Rosenski, J., Shamir, O., and Szlak, L. Multi-player bandits - A musical chairs approach. In *33rd International Conference on Machine Learning, ICML 2016*, volume 1, pp. 276–298, 2016. ISBN 9781510829008.

Rusmevichientong, P. and Tsitsiklis, J. N. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010. ISSN 0364765X. doi: 10.1287/moor.1100.0446.

Salgia, S., Gabay, T., Zhao, Q., and Cohen, K. A communication-efficient adaptive algorithm for federated learning under cumulative regret, 2023.

Sankararaman, A., Ganesh, A., and Shakkottai, S. Social learning in multi agent multi armed bandits. *Proc. ACM Meas. Anal. Comput. Syst.*, 3(3), dec 2019. doi: 10.1145/3366701. URL https://doi.org/10.1145/3366701.

Shahrampour, S., Rakhlin, A., and Jadbabaie, A. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2786–2790, 2017. doi: 10.1109/ICASSP.2017.7952664.

Shi, C., Shen, C., and Yang, J. Federated Multi-armed Bandits with Personalization, 2021. URL http://arxiv.org/abs/2102.13101.

Sun, J., Chen, T., Giannakis, G., and Yang, Z. Communication-efficient distributed learning via lazily aggregated quantized gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *34th International Conference on Machine Learning, ICML 2017*, volume 7, pp. 5119–5128, 2017. ISBN 9781510855144.

Tang, Z., Shi, S., Chu, X., Wang, W., and Li, B. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.

Tao, C., Zhang, Q., and Zhou, Y. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-Armed bandits. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, volume 2019-Novem, pp. 126–146, 2019. ISBN 9781728149523. doi: 10.1109/FOCS.2019.00017.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. URL http://www.jstor.org/stable/2346178.

Tsitsiklis, J. N. and Luo, Z. Q. Communication complexity of convex optimization. *Journal of Complexity*, 3(3):231–243, 1987. ISSN 10902708. doi: 10.1016/0885-064X(87)90013-6.

Wang, Y., Hu, J., Chen, X., and Wang, L. Distributed Bandit Learning: Near-Optimal Regret with Efficient Communication, 2019. URL http://arxiv.org/abs/1904.06309.

Yang, J., Hu, W., Lee, J. D., and Du, S. S. Impact of representation learning in linear bandits. In *International Conference on Learning Representations*, 2021.

Zhao, Z., Mao, Y., Liu, Y., Song, L., Ouyang, Y., Chen, X., and Ding, W. Towards efficient communications in federated learning: A contemporary survey. *arXiv preprint arXiv:2208.01200*, 2022.

Zhu, Z., Zhu, J., Liu, J., and Liu, Y. Federated Bandit: A Gossiping Approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):1–29, 2021. doi: 10.1145/3447380. URL https://doi.org/10.1145/3447380.

A. Additional Related Work

In this section, we discuss some more works related to ours broadly in the context of distributed learning.

Distributed Bandits: The multi-armed bandit (MAB) problem with a finite number of arms has been extensively studied in various distributed learning setups. A line of work (Chawla et al., 2020; Landgren et al., 2017; Shahrampour et al., 2017; Korda et al., 2016; Sankararaman et al., 2019; Mitra et al., 2021; Shi et al., 2021; Zhu et al., 2021) focuses on developing algorithms for cooperative learning in MAB under different structures of the underlying network. Another line of work (Liu & Zhao, 2010; Kalathil et al., 2012; Rosenski et al., 2016; Bistritz & Leshem, 2021) explores a collision based approach with no explicit communication inspired by cognitive radio networks. There are several works that also consider the impact of communication and develop communication efficient algorithms by either reducing the frequency of communication (Agarwal et al., 2021; Hillel et al., 2013; Tao et al., 2019) or proposing quantization approaches (Hanna et al., 2021). The setup of linear bandits considered in this work is more challenging than the MAB setup considered in the above works, especially for the communication constrained setting.

For the problem of distributed linear bandits, the results closest to our are by Wang et al. (2019); Huang et al. (2021) and Amani et al. (2022). Since all algorithms, including PLS and the ones proposed in these papers, achieve order-optimal regret performance, we focus on the communication cost incurred by these algorithms. The DELB algorithm proposed in Wang et al. (2019) has an uplink cost of $\mathcal{O}(d \log(MT))$ *scalars* and a downlink cost of $\mathcal{O}((M + d \log \log d) \log(MT))$ *scalars*. Assuming it is sufficient to represent each scalar using $\mathcal{O}(\log(MT))$ *bits* to ensure accuracy over MT samples, the uplink and downlink cost incurred by the algorithm can be written as $\mathcal{O}(d \log^2(MT))$ *bits* and $\mathcal{O}((M + d \log \log d) \log^2(MT))$ *bits* respectively. The Fed-PE algorithm proposed in Huang et al. (2021) has an uplink cost of $\mathcal{O}(dK \log(MT))$ *scalars* and a downlink cost of $\mathcal{O}((M \log K + d^2) \log(MT))$ *scalars*, which can be equivalently written as $\mathcal{O}(dK \log^2(MT))$ *bits* and $\mathcal{O}((M \log K + d^2) \log^2(MT))$ *bits* respectively. In the above expressions, K refers to the number of actions in the action set. The DisBE-LUCB algorithm (Amani et al., 2022) has an uplink cost of $\mathcal{O}(d \log \log(MT))$ *scalars* and a downlink cost of $\mathcal{O}(d \log \log(MT))$ *scalars*. Once again using a sufficient accuracy bit representation, both these costs are equivalent to $\mathcal{O}(d \log(MT) \log(\log(MT)))$ *bits*. However, these results hold only for $T = \Omega(d^{22})$, which also almost never encountered in practice. Furthermore, the DisBE-LUCB algorithm is most computationally intensive among all four algorithms. Lastly, the uplink and downlink costs incurred by PLS are $\mathcal{O}(d \log(T))$ *bits* and $\mathcal{O}(d \log(MT))$ *bits* which match the information theoretic lower bounds, making the results in PLS tight. It clearly improves upon the d^2 dependence in Huang et al. (2021). Moreover, the linear scaling of the downlink cost with the number of agents in Wang et al. (2019) and (Huang et al., 2021) makes them significantly worse than that of PLS. This is also corroborated by the numerical results (see Appendix F). Moreover, the uplink communication cost of PLS is independent of the number of agents, unlike other algorithms, which is a useful property as individual requirements should not scale with the size of the network. Furthermore, it might seem that the reduction in logarithmic factor is rather incremental. We would like to point out that such logarithmic factors can only be considered insignificant in the presence of a dominating polynomial factor. However, in this case of communication cost, the leading order itself is a polylogarthmic term, in which case ignoring the logarithmic factors is no longer justified. PLS improves the communication cost (in *bits*) from $\mathcal{O}(\log^2(MT))$ to $\mathcal{O}(\log T)$. This reduces the cost to the square root of the original value which certainly has a practical importance. We would like to point out that in such cases, the relative reduction is what matters in practice more than the absolute reduction, and which is a major contribution offered by PLS. Furthermore, in practice, one usually encounters finite capacity channels. The use of $\mathcal{O}(\log T)$ bits to represent scalars imposes a constraint on capacity which requires it to depend on time horizon and is not a desirable characteristic in practice. Lastly, the reduction of the $\log T$ factor requires non-trivial algorithm design and analysis, which is a contribution of PLS over existing results.

There are several other studies in addition to the ones discussed previously. Ghosh et al. (2021) et al. study the problem under heterogeneity assumptions on the agents and propose a new algorithms under personalization and clustering frameworks, the two different ways adopted to tackle heterogeneity. Chawla et al. (2022) consider a high dimensional linear bandit setting where the underlying mean reward lies in a low-dimensional space chosen from a known, finite collection. They propose a decentralized algorithm based on communication over a network to quickly identify this subspace to ensure sub-linear regret. While there is some focus on communication in this study, the proposed algorithm does not offer a linear speed up with respect to the number of agents, although per agent regret improves under collaboration. Korda et al. (2016) et al. consider the problem in a peer-to-peer communication model instead of a star topology. They propose an algorithm that achieves order-optimal regret guarantees with limited communication. However, their proposed policy requires communication of $\mathcal{O}(d^2)$ bits per message over the network which is worse than the $\mathcal{O}(d)$ required by PLS.

Sparse Linear Bandits: The problem of sparse linear bandits has been studied mainly in the centralized setting.Abbasi-Yadkori et al. (2012) proposed an algorithm for sparse linear bandits using a online to batch conversion of the confidence intervals. Borrowing techniques from compressed sensing, Carpentier & Munos (2012) proposed an algorithm that incurs an overall regret of $\mathcal{O}(s\sqrt{T})$. Chen et al. (2022) proposed the sparse variants of the famous Lin-UCB (Abbasi-Yadkori et al., 2011) and Sup-Lin-UCB (Chu et al., 2011) algorithms for the contextual bandit problem. Hao et al. (2020) proposed an algorithm with dimension-independent regret bounds for very high dimensional problems where the dimension is much larger than the time horizon. As mentioned previously, all these works consider the centralized setting which is different from the distributed setting considered in this work. To the best knowledge of the authors, this is the first work considering sparse linear bandits under a distributed setting with communication constraints.

Lower Bounds in Distributed Settings: Several studies have attempted to characterize information theoretic lower bounds on communication under for various statistical estimation tasks. The classical paper of Duchi et al. (2014) derived guarantees on communication requirements for distributed mean estimation for both independent and interactive protocols. Similarly, Tsitsiklis & Luo (1987) have studied the communication complexity in convex optimization while Diakonikolas et al. (2017) study the effect of communication constraints for the task of distribution estimation. For linear bandits, most of the papers that study lower bounds have been in context of lower bounds on regret (Dani et al., 2008; Rusmevichientong & Tsitsiklis, 2010), typically under centralized setting. In the distributed setting, to the best knowledge of the authors, the only work that discusses lower bounds especially in context of communication is Amani et al. (2022). They derive a lower bound on the regret for the contextual linear bandit problem under communication constraints. However, in our work we study the lower bounds on communication costs under regret constraints, which is different from the problem considered in their work and requires significantly different analysis techniques. Recently, Acharya et al. (2022) also analyzed the benefit of interactive protocols in distributed estimation. However, we restrict ourselves to non-interactive settings in this work and extension to such interactive settings is left as a future work.

Explore-then-Commit: The algorithms proposed in Rusmevichientong & Tsitsiklis (2010) and Yang et al. (2021) are based on explore-then-commit type of approach which shares some features with PLS. While there is a high-level similarity in the general approach of explore-then-commit, the structure of the norm estimation followed by refinement in PLS differentiates our work. This structure leads to adaptivity to the unknown $\|\theta^*\|$ and the same regret order even when $\|\theta^*\|$ is arbitrarily small. Differing from the self-adaptivity in PLS, the algorithm in Rusmevichientong & Tsitsiklis (2010) resorts to prior knowledge on the distribution of θ^*, i.e., a Bayesian setting.

B. Analysis for PLS

Before providing the detailed proofs of the Theorems regarding the performance of PLS, we state and prove two supplementary lemmas that will be used in the proofs.

Lemma B.1. *Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ be a collection of n i.i.d. random vectors in $\mathbb{R}^d$ with mean $\boldsymbol{\mu}$. Each coordinate of every random vector is assumed to be an independent sub-Gaussian random variable with variance proxy σ^2. Let $\{\mathbf{Y}_i\}_{i=1}^n$ be another collection of random vectors such that $\mathbf{Y}_i = \mathbf{X}_i \mathbb{1}\{x : \|x\| \leq R + B\}$ for all $i \in \{1, 2, \ldots, n\}$. Then the following relation holds for any $t > 0, R > \sigma\sqrt{2\log(4n)}$ and $B > \|\boldsymbol{\mu}\|_\infty$,*

$$\Pr\left(\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_n - \boldsymbol{\mu}\right\| \geq t\right) \leq 2 \cdot 5^d \cdot \exp\left(-\frac{nt^2}{8\sigma^2}\right).$$

Consequently, $\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_n - \boldsymbol{\mu}\right\| \leq \frac{2\sigma}{\sqrt{n}}(2\sqrt{d} + \sqrt{2\log(2/\delta)})$ *with probability at least* $1 - \delta$.

Proof. Let $\mathbf{v} \in \mathbb{R}^d$ be any unit vector. Since the entries of $\mathbf{X}_i$ are independent, $Z_i = \mathbf{v}^\top \mathbf{X}$ is a sub-Gaussian random variable with mean $\mu_Z = \mathbf{v}^T \boldsymbol{\mu}$ and variance proxy σ^2 for all $i \in \{1, 2, \ldots, n\}$. Let $W_i = Z_i \mathbb{1}\{[-(R+B), (R+B)]\}$. Since $B \geq \|\boldsymbol{\mu}\|_\infty$, $\mu_Z \leq B$. We also define the event $\mathcal{A} := \bigcap_{i=1}^n \mathcal{A}_i$ where $\mathcal{A}_i = \{|Z_i| \leq R + B\}$. For any $\lambda \in \mathbb{R}$, we have,

$$\begin{aligned}\mathbb{E}\left[\exp\left(\lambda\left(\frac{1}{n}\sum_{i=1}^{n}W_i-\mu_Z\right)\right)\right] &= \int_{-\infty}^{\infty}\exp\left(\lambda\left(\frac{1}{n}\sum_{i=1}^{n}w_i-\mu_Z\right)\right)f_{W_1,\ldots,W_n}(w_1,\ldots,w_n)\,\mathrm{d}w_1\,\ldots\,\mathrm{d}w_n\\
&= \int_{\infty}^{\infty}\exp\left(\lambda\left(\frac{1}{n}\sum_{i=1}^{n}w_i-\mu_Z\right)\right)\frac{1}{\Pr(\mathcal{A})}f_{Z_1,\ldots,Z_n}(w_1,\ldots,w_n)\mathbb{1}\{\mathcal{A}\}\,\mathrm{d}w_1\,\ldots\,\mathrm{d}w_n\\
&\leq \int_{\infty}^{\infty}\exp\left(\lambda\left(\frac{1}{n}\sum_{i=1}^{n}w_i-\mu_Z\right)\right)\frac{1}{\Pr(\mathcal{A})}f_{Z_1,\ldots,Z_n}(w_1,\ldots,w_n)\,\mathrm{d}w_1\,\ldots\,\mathrm{d}w_n\\
&\leq \frac{1}{\Pr(\mathcal{A})}\mathbb{E}\left[\exp\left(\lambda\left(\frac{1}{n}\sum_{i=1}^{n}Z_i-\mu_Z\right)\right)\right]\\
&\leq \frac{\exp(\lambda^2\sigma^2/2n)}{\Pr(\mathcal{A})}.\end{aligned}$$

Let us bound the term $\Pr(\mathcal{A})$. We have,

$$\begin{aligned}\Pr(\mathcal{A}) &= \Pr\left(\bigcap_{i=1}^{n}\mathcal{A}_i\right)\\
&= \Pr\left(\bigcap_{i=1}^{n}\{|Z_i|\leq R+B\}\right)\\
&= 1-\Pr\left(\bigcup_{i=1}|Z_i|>R+B\right)\\
&\geq 1-n\Pr(|Z_1|>R+B),\end{aligned}$$

Since Z_1 is a sub-Gaussian random variable,

$$\begin{aligned}\Pr(Z_1>R+B) &= \Pr(Z_1-\mu_Z>R+B-\mu_Z)\\
&\leq \Pr(Z_1-\mu_Z>R)\\
&\leq \exp\left(-\frac{R^2}{2\sigma^2}\right).\end{aligned}$$

Similarly,

$$\begin{aligned}\Pr(Z_1<-(R+B)) &= \Pr(Z_1-\mu_Z<-R-B-\mu_Z)\\
&\leq \Pr(Z_1-\mu_Z<-R)\\
&\leq \exp\left(-\frac{R^2}{2\sigma^2}\right).\end{aligned}$$

On combining the two, we obtain,

$$\Pr(\mathcal{A})\geq 1-2n\exp\left(-\frac{R^2}{2\sigma^2}\right).$$

If $R\geq\sigma\sqrt{2\log(4n)}$, then $\Pr(\mathcal{A})\geq 1/2$. Consequently,

$$\mathbb{E}\left[\exp\left(\lambda\left(\frac{1}{n}\sum_{i=1}^{n}W_i-\mu_Z\right)\right)\right]\leq 2\exp(\lambda^2\sigma^2/2n),$$

and

$$\Pr\left(\frac{1}{n}\sum_{i=1}^{n}W_i-\mu_Z>t\right)\leq 2\exp(-nt^2/2\sigma^2).$$

To obtain concentration bounds for $\left\| \frac{1}{n} \sum_{i=1}^{n} \mathbf{Y}_n - \boldsymbol{\mu} \right\|$, we use the same technique used for unbounded sub-Gaussian random variables (Jin et al., 2019). We reproduce the proof here for completeness. For brevity of notation, we define $\mathbf{W} := \frac{1}{n} \sum_{i=1}^{n} \mathbf{Y}_n$.

Let $\mathcal{N}$ denote a minimal $1/2$-cover of $\mathcal{B}_d(1)$, where $\mathcal{B}_d(r) = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq r\}$. This implies that for all $\mathbf{x} \in \mathcal{B}_d(1)$, $\exists \mathbf{y} \in \mathcal{N}$ such that $\|\mathbf{x} - \mathbf{y}\|_2 \leq 1/2$. Using the standard volumetric bounds, the cardinality of $\mathcal{N}$ can be bounded as $|\mathcal{N}| \leq 5^d$.

Once again, let $\mathbf{v} \in \mathcal{B}_d(1)$ denote a unit vector. For every $\mathbf{v}$, we have a $\mathbf{z}_\mathbf{v} \in \mathcal{N}$ such that $\mathbf{v} = \mathbf{z}_\mathbf{v} + \mathbf{u}$ with $\|\mathbf{u}\| \leq 1/2$. Thus, for any vector $\mathbf{X}$, we have,

$$\begin{aligned}
\max_{\mathbf{v} \in \mathcal{B}_d(1)} \mathbf{v}^\top \mathbf{X} &= \max_{\mathbf{v} \in \mathcal{B}_d(1)} (\mathbf{z}_\mathbf{v} + \mathbf{u})^\top \mathbf{X} \\
&\leq \max_{\mathbf{z} \in \mathcal{N}} \mathbf{z}^\top \mathbf{X} + \max_{\mathbf{u} \in \mathcal{B}_d(1/2)} \mathbf{u}^\top \mathbf{X} \\
&\leq \max_{\mathbf{z} \in \mathcal{N}} \mathbf{z}^\top \mathbf{X} + \frac{1}{2} \max_{\mathbf{u} \in \mathcal{B}_d(1)} \mathbf{u}^\top \mathbf{X}.
\end{aligned}$$

Thus, $\max_{\mathbf{v} \in \mathcal{B}_d(1)} \mathbf{v}^\top \mathbf{X} \leq 2 \max_{\mathbf{z} \in \mathcal{N}} \mathbf{z}^\top \mathbf{X}$. Moreover, since $\|\mathbf{v}\|_2 = 1$, $\max_{\mathbf{v} \in \mathcal{B}_d(1)} \mathbf{v}^\top \mathbf{X} = \|\mathbf{X}\|$. Consequently,

$$\begin{aligned}
\Pr\left(\|\mathbf{Z} - \boldsymbol{\mu}\| \geq t\right) &\leq \Pr(\max_{\mathbf{v} \in \mathcal{B}_d(1)} \mathbf{v}^\top (\mathbf{W} - \boldsymbol{\mu}) \geq t) \\
&\leq \Pr(\max_{\mathbf{z} \in \mathcal{N}} \mathbf{z}^\top (\mathbf{W} - \boldsymbol{\mu}) \geq t/2) \\
&\leq 2|\mathcal{N}| \exp\left(-\frac{nt^2}{8\sigma^2}\right).
\end{aligned}$$

Plugging in the value of $|\mathcal{N}|$ yields the final result. □

Lemma B.2. *For any epoch k in PLS, the estimate $\hat{\theta}_k^{(\text{SERV})}$ satisfies*

$$\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\| \leq \tau_k,$$

with probability at least $1 - \delta$.

Proof. We prove the claim using induction. Firstly, note that if we define $\bar{\theta}_{k-1} := 0$ for all epochs k during the norm estimation stage, then we can rewrite the method to compute the estimate at the server, $\hat{\theta}_k^{(\text{SERV})}$, during the norm estimation stage in the same way as it is computed during the refinement stage. Thus, we consider the definition of $\hat{\theta}_k^{(\text{SERV})}$ as used in Algorithm 4 while implicitly assuming $\bar{\theta}_{k-1} = 0$ for all epochs k during the norm estimation stage.

Let $\eta_k^{(j)} \in \mathbb{R}^d$ denote the quantization noise added by jth client during the kth epoch, i.e., the quantized version received by the server can be written as $Q(\tilde{\theta}_k^{(j)}) = \tilde{\theta}_k^{(j)} + \eta_k^{(j)}$. Since each coordinate is quantized independently, each coordinate of $\eta_k^{(j)}$ is an independent zero mean sub-Gaussian random variable with variance proxy $\alpha_k^2/4d$. At the end of the kth epoch, we have

$$\begin{aligned}
\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\| &= \left\| \bar{\theta}_{k-1} + \frac{1}{M}\sum_{j=1}^{M} Q(\tilde{\theta}_k^{(j)}) - \theta^* \right\| \\
&= \left\| \bar{\theta}_{k-1} + \frac{1}{M}\sum_{j=1}^{M} (\tilde{\theta}_k^{(j)} + \eta_k^{(j)}) - \theta^* \right\| \\
&= \left\| \bar{\theta}_{k-1} + \frac{1}{M}\sum_{j=1}^{M} (\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1})\mathbb{1}_{R_k+B_k} + \frac{1}{M}\sum_{j=1}^{M} \eta_k^{(j)} - \theta^* \right\| \\
&\leq \left\| \frac{1}{M}\sum_{j=1}^{M} (\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1})\mathbb{1}_{R_k+B_k} - (\theta^* - \bar{\theta}_{k-1}) \right\| + \left\| \frac{1}{M}\sum_{j=1}^{M} \eta_k^{(j)} \right\|.
\end{aligned}$$

The second term can be bounded using the concentration of sub-Gaussian random vectors as

$$\left\| \frac{1}{M}\sum_{j=1}^{M} \eta_k^{(j)} \right\| \leq \frac{2\alpha_k}{\sqrt{M}} \left(1 + \sqrt{\frac{1}{2d}\log\left(\frac{2K}{\delta}\right)} \right).$$

For the first term, we will employ Lemma B.1. However, we first need to ensure that the conditions of the lemma are satisfied. For any epoch k, the particular choice of R_k and s_k in PLS satisfies the assumption in Lemma B.1 for all k. To bound the mean, we need to consider the epochs with $\bar{\theta}_{k-1} = 0$ (i.e., the norm estimation stage) and $\bar{\theta}_{k-1} \neq 0$ (the refinement stage) separately. We begin with the case of $\bar{\theta}_{k-1} = 0$. Under this scenario, note that $\|\mathbb{E}[\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1}]\| = \|\mathbb{E}[\hat{\theta}_k^{(j)}]\| = \|\theta^*\|$. For the first epoch, we know that $\|\theta^*\| \leq 1 = B_1$. This implies, we can apply the lemma for the base case. For any epoch $k \geq 2$, we use the inductive hypothesis to establish the bound on the mean. In particular, we use a better estimate of $\|\theta^*\|$ by noting that the norm estimation stage had not been terminated by epoch $k-1$ which implies that $\|\hat{\theta}_{k-1}^{(\text{SERV})}\| \leq 4\tau_{k-1}$. Along with the induction hypothesis, $\|\hat{\theta}_{k-1}^{(\text{SERV})} - \theta^*\| \leq \tau_{k-1}$, we can conclude that $\|\theta^*\| \leq 5\tau_{k-1} \leq B_k$, as required.

Now we are ready to invoke Lemma B.1. Using Lemma B.1, we can conclude that with probability at least $1 - \delta/K$,

$$\left\| \frac{1}{M}\sum_{j=1}^{M} (\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1})\mathbb{1}_{R_k+B_k} - (\theta^* - \bar{\theta}_{k-1}) \right\| \leq \frac{4\sigma\sqrt{d}}{\sqrt{Ms_k}} \left(1 + \sqrt{\frac{1}{2d}\log\left(\frac{2K}{\delta}\right)} \right).$$

On plugging in the value of s_k and α_k and combining the equations, we obtain,

$$\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\| \leq \frac{2^{-k}}{\sqrt{M}} + \frac{2^{-(k+1)}}{\sqrt{M}} = \frac{3}{\sqrt{M}} \cdot 2^{-(k+1)} = \tau_k,$$

holds with probability at least $1 - \delta/K$.

We similarly consider the case where in epoch k, $\bar{\theta}_{k-1} \neq 0$. For this case, we apply Lemma B.1 conditioned on the value of $\bar{\theta}_{k-1}$ and hence all expectations and probabilities are computed with respect to the conditional measure. For brevity of notation and ease of understanding, we will assume the conditioning has been implicitly applied. The condition on R_k holds using the argument as in the previous case and hence we focus only on showing the bound on the mean. If ζ_{k-1} denotes the quantization error during the $k-1$ epoch, then we can write $\bar{\theta}_{k-1} = \hat{\theta}_{k-1}^{(\text{SERV})} + \zeta_{k-1}$. Consequently, $\|\mathbb{E}[\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1}]\| = \|\theta^* - \bar{\theta}_{k-1}\| = \|\theta^* - \hat{\theta}_{k-1}^{(\text{SERV})} - \zeta_{k-1}\| \leq \|\theta^* - \hat{\theta}_{k-1}^{(\text{SERV})}\| + \|\zeta_{k-1}\| \leq \|\theta^* - \hat{\theta}_{k-1}^{(\text{SERV})}\| + \beta_{k-1} \leq \tau_{k-1} + \beta_{k-1} \leq B_k$. For the last step, the bound on $\|\theta^* - \hat{\theta}_{k-1}^{(\text{SERV})}\|$ holds due to the hypothesis in the induction step. In the base case, i.e., k_0, the index of the epoch when the norm estimation stage terminates, the bound holds from the result obtained for the case of $\bar{\theta}_{k-1} = 0$. This implies that we can apply the lemma also for the case of $\bar{\theta}_{k-1} \neq 0$, thereby obtaining the bound on $\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\|$ for all k.

We would like to point that to avoid repeating steps in the proof, we have directly shown the result after invoking Lemma B.1 for all epochs k. The proof actually follows by first invoking Lemma B.1 for the base case and establishing the result. For any epoch k, we then use the inductive hypothesis of the relation for $k-1$ to establish the conditions on the mean after which Lemma B.1 is invoked to complete the induction step.

Lastly, consider the events given by $\mathcal{E}_k := \{\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\| \leq \tau_k\}$ for $k = 1, 2, \dots K$ and the event $\mathcal{E} := \bigcap_{k=1}^{K} \mathcal{E}_k$. From the above analysis, we know that $\Pr(\mathcal{E}_k | \mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_{k-1}) \geq 1 - \delta/K$ for $k = 1, 2, \dots, K$, where $\Pr(\mathcal{E}_0) = 1$. Thus, we have

$$\begin{aligned} \Pr(\mathcal{E}) &= \Pr\left(\bigcap_{k=1}^{K} \mathcal{E}_k\right) \\ &= \prod_{i=1}^{k} \Pr(\mathcal{E}_k | \mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_{k-1}) \\ &\geq \prod_{i=1}^{K} \left(1 - \frac{\delta}{K}\right)^K \\ &\geq 1 - \delta, \end{aligned}$$

as required.

□

We now proceed to provide detailed proofs of the various theorems and lemmas that characterize the regret and communication cost of PLS.

B.1. Proof of Lemma 4.2

Let $R_{\text{NE}}(T)$ denote the regret incurred during the norm estimation stage. We assume that the stage continues until a `terminate` is received from the server or each client has played a total of T actions, whichever happens first.

Under the event $\mathcal{E}$ as defined in Proof of Lemma B.2, the algorithm will terminate at the end of epoch k_0 where $k_0 := \min\{k \in \mathbb{N} : 4\tau_k \leq \|\hat{\theta}_k^{(\text{SERV})}\|\}$. We note that for all k for which the inequality $4\tau_k \leq \|\hat{\theta}_k^{(\text{SERV})}\|$ holds, we also have the relation

$$\begin{aligned} \|\hat{\theta}_k^{(\text{SERV})} - \theta^*\| \leq \tau_k &\leq \frac{1}{4} \cdot \|\hat{\theta}_k^{(\text{SERV})}\| \\ \implies \|\theta^*\| &\in \left[\frac{3}{4}\|\hat{\theta}_k^{(\text{SERV})}\|, \frac{5}{4}\|\hat{\theta}_k^{(\text{SERV})}\|\right] \\ \implies \|\hat{\theta}_k^{(\text{SERV})}\| &\in \left[\frac{4}{5}\|\theta^*\|, \frac{4}{3}\|\theta^*\|\right]. \end{aligned}$$

Consequently, we also have $\tau_k \leq \frac{1}{3} \cdot \|\theta^*\|$. On plugging in the value of τ_k, we obtain

$$\tau_k \leq \frac{1}{3} \cdot \|\theta^*\| \implies 2^{-k} \leq \frac{2\sqrt{M}}{9}\|\theta^*\| \implies k \geq \log_2\left(\frac{9}{2\|\theta^*\|\sqrt{M}}\right).$$

Since k_0 is the smallest natural number satisfying this relation, $k_0 = \max\left\{\left\lceil \log_2\left(\frac{9}{2\|\theta^*\|\sqrt{M}}\right)\right\rceil, 1\right\}$.

We know that $a_t, a^* \in \mathcal{A} = \{x : \|x\|_2 \leq 1\}$ for all $t \in \{1, 2, 3, \dots, T\}$. Hence, the instantaneous regret at time instant can be bounded as

$$r_t = \langle \theta^*, a^* \rangle - \langle \theta^*, a_t \rangle \leq \|\theta^*\|_2 \|a^* - a_t\|_2 \leq 2\|\theta^*\|_2.$$

Consequently, we can bound $R_{\text{NE}}(T)$ as follows. If $k_0 = 1$, then $R_{\text{NE}}(T)$ can be simply upper bounded as $Mds_1 = \mathcal{O}(1)$. If $k_0 \geq 2$, we have

$$\begin{aligned} R_{\text{NE}}(T) &\leq \sum_{k=1}^{k_0} 2M\|\theta^*\|_2 \cdot ds_k \\ &\leq 2Md\|\theta^*\|_2 \cdot s_{k_0} \cdot k_0 \\ &\leq 2Md\|\theta^*\|_2 \cdot \left(32d\sigma^2 \log\left(\frac{8MK}{\delta}\right) 4^{\log_2(9/2\|\theta^*\|_2)+1} + 1\right) \cdot \left(\log_2\left(\frac{9}{2\|\theta^*\|_2}\right) + 1\right) \\ &\leq 2Md\|\theta^*\|_2 \cdot \left(2592\frac{d\sigma^2}{\|\theta^*\|_2^2} \log\left(\frac{8MK}{\delta}\right) + 1\right) \cdot \left(\log_2\left(\frac{9}{2\|\theta^*\|_2}\right) + 1\right) \\ &\leq 5184\frac{d^2\sigma^2}{\|\theta^*\|_2} \log\left(\frac{8MK}{\delta}\right)\left(\log_2\left(\frac{9}{2\|\theta^*\|_2}\right) + 1\right) + 2d\left(\log_2\left(\frac{9}{2\|\theta^*\|_2}\right) + 1\right). \end{aligned}$$

A trivial bound on $R_{\text{NE}}(T)$ is $2\|\theta^*\|_2 MT$. Note that for larger values of $\|\theta^*\|_2$, the former bound is tighter. However, as $\|\theta^*\|_2$ gets smaller, the latter bound becomes a stronger one. To obtain the optimal bound on $R_{\text{NE}}(T)$, we choose the minimum of the two. Thus,

$$R_{\text{NE}}(T) \leq \min\left\{5184\frac{d^2\sigma^2}{\|\theta^*\|_2} \log\left(\frac{8MK}{\delta}\right)\left(\log_2\left(\frac{9}{2\|\theta^*\|_2}\right) + 1\right) + 2d\left(\log_2\left(\frac{9}{2\|\theta^*\|_2}\right) + 1\right), 2\|\theta^*\|_2 MT\right\}.$$

On setting $\|\theta^*\|_2 = d/\sqrt{MT}$, we obtain $R_{\text{NE}}(T)$ is $\mathcal{O}(d\sqrt{MT} \cdot \log(MT) \cdot \log(\log T/\delta))$, as required.

B.2. Proof of Lemma 4.3

We are given access to estimate, $\hat{\theta}$, of the true vector, θ, such that $\|\hat{\theta} - \theta\|_2 \leq \tau \leq \|\theta\|_2$. This implies that $\|\hat{\theta}\|_2 \in [\|\theta\|_2 - \tau, \|\theta\|_2 + \tau]$. Consider the relation,

$$\begin{aligned} &\|\hat{\theta} - \theta\|_2^2 \leq \tau^2 \\ \implies &\|\hat{\theta}\|_2^2 + \|\theta^*\|_2^2 - 2\langle\hat{\theta}, \theta\rangle \leq \tau^2 \\ &\implies -\langle\hat{\theta}, \theta\rangle \leq \frac{1}{2}\left(\tau^2 - \|\hat{\theta}\|_2^2 - \|\theta\|_2^2\right) \\ &\implies \|\theta\| - \frac{1}{\|\hat{\theta}\|_2}\langle\hat{\theta}, \theta\rangle \leq \frac{1}{2\|\hat{\theta}\|_2}\left(\tau^2 - \|\theta\|_2^2 - \|\theta\|_2^2 + 2\|\theta\|_2\|\theta\|_2\right) \\ &\implies \|\theta\| - \frac{1}{\|\hat{\theta}\|_2}\langle\hat{\theta}, \theta\rangle \leq \frac{1}{2\|\theta\|_2}\left(\tau^2 - (\|\theta\|_2 - \|\theta\|_2)^2\right). \end{aligned}$$

Note that the LHS in the last equation is an upper bound on the regret incurred by playing the action $a = \hat{\theta}/\|\hat{\theta}\|_2$. If we denote the regret incurred by playing the action a by Reg(a) and let $\|\hat{\theta}\|_2 = \|\theta\|_2 + \upsilon$, for some $\upsilon \in [-\tau, \tau]$, then

$$\text{Reg}(a) \leq \frac{\tau^2 - \upsilon^2}{2(\|\theta\|_2 + \upsilon)}.$$

To bound the above expression, we consider the function $f(\upsilon) = \dfrac{\tau^2 - \upsilon^2}{(\|\theta\|_2 + \upsilon)}$. Since f is rational function of two polynomials, it is differentiable. On differentiating f, we obtain $f'(\upsilon) = -\dfrac{\upsilon^2 + 2\upsilon\|\theta\|_2 + \tau^2}{(\|\theta\|_2 + \upsilon)^2}$. The solutions for $f'(\upsilon) = 0$ are given by $\upsilon_+ = -\|\theta\|_2 + \sqrt{\|\theta\|_2^2 - \tau^2}$ and $\upsilon_- = -\|\theta\|_2 - \sqrt{\|\theta\|_2^2 - \tau^2}$. By double differentiating f, we can verify that it is indeed maximized at υ_+ for $\upsilon \in [-\tau, \tau]$. Moreover, note that solutions for $f'(\upsilon) = 0$ are real numbers only for $\tau \leq \|\theta\|_2$. This explains the need for the constraint on τ. On setting $\upsilon = \upsilon_+$ in the expression for Reg(a), we obtain the following bound.

$$\begin{aligned}
\mathrm{Reg}(a) &\leq \frac{\tau^2 - (-\|\theta^*\|_2 + \sqrt{\|\theta\|_2^2 - \tau^2})^2}{2\sqrt{\|\theta\|_2 - \tau^2}} \\
&\leq \frac{\tau^2 - \|\theta\|_2^2 - \|\theta\|_2^2 + \tau^2 + 2\|\theta\|_2\sqrt{\|\theta\|_2^2 - \tau^2}}{2\sqrt{\|\theta\|_2 - \tau^2}} \\
&\leq \|\theta\|_2 - \sqrt{\|\theta\|_2^2 - \tau^2} \\
&\leq \frac{\tau^2}{\|\theta\|}.
\end{aligned}$$

B.3. Proof of Theorem 4.1

Let $R_{\text{REF}}(T)$ denote the regret incurred during the refinement stage. To obtain a bound on $R_{\text{REF}}(T)$, we consider an epoch k during the refinement stage. Similar to the norm estimation stage, we bound the regret incurred during the exploration sub-epoch as $2\|\theta^*\|_2 \cdot Mds_k$. Using Lemma 4.3 and the fact that $\|\bar{\theta}_k - \theta^*\| \leq 2\tau_k$ (Ref. Lemma B.2), we can bound the regret during the exploitation sub-epoch as $\dfrac{4\tau_k^2}{\|\theta^*\|_2} \cdot Mt_k$. If $R^{(k)}$ denotes the regret incurred during epoch k, then we have the following relation.

$$\begin{aligned}
R^{(k)} &\leq 2\|\theta^*\|_2 \cdot Mds_k + \frac{4M\tau_k^2}{\|\theta^*\|_2} \cdot (Ms_k^2\mu_0^2 + 1) \\
&\leq 2\|\theta^*\|_2 \cdot Mds_k + \frac{100(M\tau_k s_k\|\theta^*\|_2)^2}{9\|\theta^*\|_2} + \frac{4M\tau_k^2}{\|\theta^*\|_2} \\
&\leq 64\|\theta^*\|_2 Md^2\sigma^2 \log\left(\frac{8MK}{\delta}\right) 4^k + 2\|\theta^*\|_2 \cdot Md + 8M\|\theta^*\|_2 \left(6400\sigma^4 d^2 \log^2\left(\frac{8MK}{\delta}\right) 4^k + 4^{-k}\right) + M\|\theta^*\|_2 \\
&\leq 51200 \cdot \|\theta^*\|_2 Md^2\sigma^2(1+\sigma^2)\log^2\left(\frac{8MK}{\delta}\right) 4^k + M\|\theta^*\|_2\left(8 \cdot 4^{-k} + 2d + 1\right)
\end{aligned}$$

Note that the regret incurred during the exploration sub-epoch is the same as that incurred during the exploitation sub-epoch upto constant factors. This echoes the discussion in Sec. 4 on the careful choice of the lengths of exploration and exploitation epochs in PLS using the estimated norm of $\|\theta^*\|_2$.

Let k_1 denote the index of the epoch when the query budget ends. Also, recall that k_0 was defined to be the epoch index during which the norm estimation stage terminates. If $k_1 = k_0$, then it implies that the exploitation sub-epoch of the k_0^{th} epoch was not completed. Consequently, the regret incurred during the partial sub-epoch is bounded by the regret incurred during the corresponding exploration sub-epoch (upto a constant factor) which in turn is bounded by the regret incurred during the norm estimation stage. Hence, the overall regret has the same order as the of the norm estimation stage, that is, $\tilde{\mathcal{O}}(d\sqrt{MT})$, as required. For the rest of the proof we focus on the case $k_1 \geq k_0 + 1$.

The regret incurred during the refinement stage can be be bounded as

$$R_{\text{REF}}(T) \leq \sum_{k=k_0}^{k_1} R^{(k)}.$$

Since we already have a bound on k_0, we can bound the above expression by finding an upper bound on k_1. We can bound k_1 by using the length of the time horizon. We have,

$$\sum_{k=1}^{k_1-1} ds_k + \sum_{k=k_0}^{k_1-1} t_k \leq T.$$

The first term corresponds to the length of all the exploration (sub-)epochs, including the ones during the norm estimation procedure, while the second accounts for the length of all the exploitation sequences. On plugging in the values of s_k and t_k, we obtain,

$$
\begin{aligned}
T &\geq \sum_{k=1}^{k_1-1} ds_k + \sum_{k=k_0}^{k_1-1} t_k \\
&\geq \sum_{k=k_0}^{k_1-1} M s_k^2 \mu_0^2 \\
&\geq \sum_{k=k_0}^{k_1-1} \frac{16384}{25} M d^2 \|\theta^*\|^2 \sigma^4 \log^2\left(\frac{8MK}{\delta}\right) 16^k \\
&\geq \frac{1024}{9} M d^2 \|\theta^*\|^2 \sigma^4 \log^2\left(\frac{8MK}{\delta}\right) \frac{16^{k_1} - 16^{k_0}}{15} \\
&\geq 100 M d^2 \|\theta^*\|^2 \sigma^4 \log^2\left(\frac{8MK}{\delta}\right) 16^{k_1}.
\end{aligned}
$$

where the last step follows by noting that $k_1 \geq k_0 + 1$. This implies that,

$$
k_1 \leq \log_{16}\left(\frac{T}{100 M d^2 \|\theta^*\|_2^2 \sigma^4}\left(\log\left(\frac{8MK}{\delta}\right)\right)^{-2}\right)
$$

.

Consequently,

$$
\begin{aligned}
R_{\text{REF}}(T) &\leq \sum_{k=k_0}^{k_1} R^{(k)} \\
&\leq \sum_{k=k_0}^{k_1} 51200 \cdot \|\theta^*\|_2 M d^2 \sigma^2 (1+\sigma^2) \log^2\left(\frac{8MK}{\delta}\right) 4^k + M\|\theta^*\|_2 \left(8 \cdot 4^{-k} + 2d + 1\right) \\
&\leq 69000 \cdot \|\theta^*\|_2 M d^2 \sigma^2 (1+\sigma^2) \log^2\left(\frac{8MK}{\delta}\right) 4^{k_1} + M k_1 \|\theta^*\|_2 \left(2d + 9\right) \\
&\leq 69000 \cdot \|\theta^*\|_2 M d^2 \sigma^2 (1+\sigma^2) \log^2\left(\frac{8MK}{\delta}\right) \cdot \frac{1}{d\sigma^2 \|\theta^*\|_2} \left(\log\left(\frac{8MK}{\delta}\right)\right)^{-1} \times \\
&\qquad\qquad \sqrt{\frac{T}{100M}} + M k_1 \|\theta^*\|_2 \left(2d + 9\right) \\
&\leq 6900 (1+\sigma^2) \log\left(\frac{8MK}{\delta}\right) \cdot d\sqrt{MT} + M k_1 \|\theta^*\|_2 \left(2d + 9\right).
\end{aligned}
$$

Adding this to the regret incurred during the norm estimation stage, we can conclude that $R_{\text{PLS}}(T)$ satisfies $\mathcal{O}(d\sqrt{MT} \cdot \log(MT) \cdot \log(\log T/\delta))$.

B.4. Proof of Lemma 4.5

Consider a vector $x \in \mathbb{R}^d$ with $\|x\| \leq r$ and let $Q(x)$ denote its quantized version achieved by a quantizer (deterministic or stochastic) upto a precision of ε. This implies that $\|x - Q(x)\|_2 \leq \varepsilon \implies \|Q(x)\|_2 \leq r + \varepsilon$. Note that $Q(x)$ can be represented as $(q_1, q_2, \dots, q_d)^\top$ where $q_i \in \{-l_\varepsilon/2, \dots, 0, \dots, l_\varepsilon/2\}$ represents the corresponding quantized index for all

$i \in \{1, 2, \dots, d\}$. Thus, we have,

$$\begin{aligned}
&\sum_{i=1}^{d} q_i^2 \left(\frac{2r}{l_\varepsilon}\right)^2 \leq (r+\varepsilon)^2 \\
\implies &\sum_{i=1}^{d} q_i^2 \leq \left(\frac{(r+\varepsilon)l_\varepsilon}{2r}\right)^2 \\
&\leq 4d\left(\frac{r}{\varepsilon}+1\right)^2 .
\end{aligned}$$

Consequently,

$$\sum_{i=1}^{d} |q_i| \leq \sqrt{d\sum_{i=1}^{d} q_i^2} \leq 2d\left(\frac{r}{\varepsilon}+1\right).$$

It can be noted that under the encoding scheme based on unary encoding used in PLS, the message size in bits in PLS is given by $3d + \sum_{i=1}^{d} |q_i|$, where $3d$ corresponds to the sum of lengths of the headers. As a result, the message size is bounded by $d(3 + 2(r/\varepsilon + 1))$. We use this relation to establish the message sizes on both the uplink and the downlink channels.

Let us first consider the uplink communication. In epoch k, r corresponds to $R_k + B_k$ and ε to α_k. On plugging in the prescribed values of the above parameters, we note that $(R_k + B_k)/\alpha_k$ is C/α_0, where C is a constant independent of d, M and T. Hence, any message sent on the uplink channel is no more than $\mathcal{O}(d)$ bits. Similarly, for the downlink communication, the ratio $(B_k + \tau_k)/\beta_k \leq C'/\beta_0$ where once again C' is a constant independent of d, M and T. Hence, any message sent on the downlink channel also has a size of $\mathcal{O}(d)$ bits, as required.

B.5. Proof of Theorem 4.4

The bound on the uplink communication cost follows immediately from Lemma 4.5. Since the agents send a message of $\mathcal{O}(d/\alpha_0)$ bits only once every epoch and there are no more than $K = \mathcal{O}(\log T)$ epochs in PLS, the uplink communication cost of PLS, $C_{\text{u}}(T)$, satisfies $\mathcal{O}((d/\alpha_0)\log T)$. The $\mathcal{O}((d/\beta_0)\log T)$ term in $C_{\text{d}}(T)$, the downlink cost, also follows using the same argument.

The $\mathcal{O}(d \log M)$ term in the downlink communication cost corresponds to sending the initial estimate of θ^* to the clients, specifically in the event that the norm estimation stage ends within the first epoch. Note that, if the norm estimation stage ends in the first epoch, the estimation error in θ^* becomes $\tau_1 = \mathcal{O}(\sqrt{1/M})$ from the initial estimate of $\mathcal{O}(1)$. As a result, PLS requires $\mathcal{O}(d \log M)$ bits to transmit the estimate, adding to the downlink communication cost. This $\mathcal{O}(d \log M)$ cost is what facilitates information exchange between the agents that leads to the linear speedup with respect to the number of agents.

Remark B.3. This estimate is also sent using the unary encoding scheme used in PLS over $\mathcal{O}(\log M)$ rounds with each round sending one additional bit of information per coordinate. Depending on the synchronization requirements as dictated by the hardware, the learner may not be allowed multiple uses of the channel. In the event that it is possible to use the channel several times between two actions, this information can be easily sent with $\mathcal{O}(\log M)$ uses of the channel. However, if the server is allowed to use the channel only once between two actions of the agent, this transmission of the initial estimate can be accommodated within PLS as follows. At the end of the exploration sub-epoch of the first epoch, instead of starting with the exploitation sub-epoch, the agents begin with the exploration sub-epoch of the second epoch. During this time, the server broadcasts the initial estimate of θ^* over the next $\mathcal{O}(\log M)$ time instants. The agents continue to play the actions as dictated by the exploration sub-epoch until the communication is completed. Once the communication is completed, the agents now start with the exploitation sub-epoch of the first epoch. At the end of the first epoch, they restart the exploration sub-epoch, starting from where they had left at the end of the communication. Thus, if needed, PLS can accommodate the additional transmission requirements by a minor reordering of the explorative and exploitative actions.

C. Lower Bounds

In this section, we discuss the details of the proofs to establish the lower bounds on the communication cost under regret constraints.

C.1. Proof of Theorem 5.1

The proof of this theorem consists of two main steps. In the first step, we show that all algorithms achieving a sub-linear cumulative regret need to solve the problem of distributed mean estimation. This reduction allows us to leverage several existing techniques and results for information-theoretic lower bounds, especially in the case of distributed statistical estimation. The second step is to establish lower bound on communication cost (both uplink and downlink) for a given estimation error based on the above reduction. The primary idea for this step is to use to classical reduction to identification from a specifically constructed hard instance. Specifically, we show that for any estimator with a small error it is necessary to solve the identification problem. The final bound on the communication cost is then obtained by using Fano's inequality to bound the error of this identification problem.

We first establish how the constraint of a sub-linear cumulative regret for linear bandit algorithm can be translated to having a small simple regret. Consider any policy π that achieves a sub-linear regret $R_\pi(T)$ under the setup described in Sec. 2. Consequently, the policy π also guarantees a sub-linear regret of $R_\pi(T)/MT$. This follows immediately by choosing the final action to be the average of all the actions chosen by all the agents. Note that this is a permissible action since $\mathcal{A}$ is a convex set. As a result, a lower bound of $\underline{r}(T)$ on simple regret achievable by any policy immediately implies a lower bound of $\underline{R}(T) = MT\underline{r}(T)$ on the cumulative regret achievable by any policy. This relation is used later in the proof to draw parallels with the problem of distributed mean estimation.

For the second step, we separately establish the lower bounds for the uplink and downlink communication cost, by constructing two different hard instances. We begin with the lower bound on the downlink cost. We would like to point out that we prove a more general case for the downlink cost where we allow algorithms that incur a sublinear w.r.t to both T and M, as opposed to just being order optimal.

C.1.1. Proof of downlink cost

For the lower bound on the downlink cost, recall that for a policy a π with sub-linear cumulative regret of $R_\pi(T)$, the average of all the actions chosen by all the agents, denoted by $\bar{a}_\pi$, achieves a simple regret of $R_\pi(T)/MT$. This implies that using the actions taken by π, one can estimate θ^* to a reasonable accuracy dictated by $R_\pi(T)$. In particular, let $\hat{\theta}(A;\pi)$ denote an estimator of θ^* based on all the actions taken by a policy π, which we denote by the random variable A, and $\|\theta^*\|_2 = 1$. Then,

$$\begin{aligned}\inf_{\hat{\theta}} \|\hat{\theta}(A;\pi) - \theta^*\|_2^2 &\leq \left\| \frac{\bar{a}_\pi}{\|\bar{a}_\pi\|_2} - \theta^* \right\|_2^2 \\ &\leq \frac{\|\bar{a}_\pi\|_2^2}{\|\bar{a}_\pi\|_2^2} + \|\theta^*\|_2^2 - 2\langle \theta^*, \frac{\bar{a}_\pi}{\|\bar{a}_\pi\|_2}\rangle \\ &= 2(1 - \langle \theta^*, \bar{a}_\pi \rangle) \\ &\leq 2\frac{R_\pi(T)}{MT}.\end{aligned}$$

We use the above relation to obtain a bound on the downlink communication cost based on the information carried in A, the actions taken by a policy, about the underlying vector θ^*.

Let $\mathcal{V}_\varepsilon$ denote a maximal 2ε-packing of $\mathcal{S}^d$ using the $\|\cdot\|_2$ norm, where $\mathcal{S}^d$ denotes the surface of a unit sphere in $\mathbb{R}^d$. Equivalently, $\mathcal{S}^d = \partial\mathcal{B}_d(1) \in \mathbb{R}^{d-1}$, where $\mathcal{B}_d(r)$ denotes a ball (in $\|\cdot\|_2$-norm) of radius r in $\mathbb{R}^d$. Let V be a random variable chosen uniformly at random from $\mathcal{V}_\varepsilon$. Consider a linear bandit instance instantiated with $\theta^* = V$. Note that this construction implicitly ensures $\|\theta^*\|_2 = 1$. As before, let $\hat{\theta}(A;\pi)$ denote an estimator of θ^* based on all the actions taken by a policy π. This problem of estimating θ^* can be mapped to that of identifying V using classical techniques by defining a testing function $\hat{V}$ for each estimator $\hat{\theta}$. In particular, define $\hat{V} := \arg\min_{v \in \mathcal{V}_\varepsilon} \|\hat{\theta}(A;\pi) - v\|_2$, that is, it maps $\hat{\theta}$ to the closest point in the set $\mathcal{V}_\varepsilon$.

Since $\mathcal{V}_\epsilon$ is 2ε-packing, $\|\hat{\theta}(A;\pi) - V\|_2 > \varepsilon$ whenever $\hat{V} \neq V$. Consequently, $\Pr\left(\|\hat{\theta}(A;\pi) - \theta^*\|_2^2 > \varepsilon^2/2\right) =$

$\Pr\left(\|\hat{\theta}(A;\pi) - V\|_2^2 > \varepsilon^2/2\right) \geq \Pr(\hat{V} \neq V)$. Since $V \to A \to \hat{V}$ from a Markov chain, using Fano's inequality (Cover & Thomas, 2006), we can conclude that

$$\Pr(\hat{V} \neq V) \geq 1 - \frac{I(V;A) + \log 2}{|\mathcal{V}_\varepsilon|},$$

where $I(V;A)$ denotes the mutual information between V and A. Note that the event $\{\|\hat{\theta}(A;\pi) - \theta^*\|_2^2 > \varepsilon^2/2\}$ implies that no estimator based on the actions of π can estimate θ^* within an error of $\varepsilon^2/2$ implying that π incurs a cumulative regret of at least $\varepsilon^2 MT/4$. Hence, to ensure a cumulative regret of $R_\pi(T)$ with probability at least $2/3$, we need to ensure $\Pr(\hat{V} \neq V) \leq \Pr\left(\|\hat{\theta}(A;\pi) - \theta^*\|_2^2 > \varepsilon^2/2\right) < 1/3$ for $\varepsilon = 2\sqrt{R_\pi(T)/MT}$. Equivalently, $I(V;A) \geq 2\log|\mathcal{V}_\varepsilon|/3 - \log 2$ with $\varepsilon = 2\sqrt{R_\pi(T)/MT}$. Using the standard bounds on packing numbers (Ball, 1997) and noting that $R_\pi(T)$ is sub-linear in both M and T, we can conclude that $I(V;A) \geq \Omega(d\log(MT))$. The bound on the communication cost is obtained using the data processing inequality. Let Z denote all the messages broadcast by the server. Then $I(V;Z)$ is a lower bound on the downlink communication cost. Notice that Z obeys the Markov chain $V \to Z \to A \to \hat{V}$ since the actions taken by the agents change with V through the messages broadcast by the server. From data processing inequality, we have, $I(V;Z) \geq I(V;A)$. In other words, since the messages Z transfer the information about the actions of other agents to any given agent, they should at least have as much information about θ^* (or equivalently V) as much as A, the set of all actions, does. Combining this with the bound on $I(V;A)$, we arrive the required lower bound on the downlink communication cost.

C.1.2. Proof for uplink cost

We establish bounds on the uplink cost by drawing parallels to the distributed mean estimation problem. Consider the problem of distributed mean estimation where X denotes the random variable corresponding to the observations by the agents and Y denotes the messages sent by the agents to the server. Based on the messages Y, if no estimator can recover θ^* to within a mean squared error of ε^2, then no linear bandit algorithm can achieve a simple regret of ε^2. This is similar to the argument shown for the downlink case. This implies that, only if θ^* can be estimated to within an accuracy of ε^2, can there exist a linear bandit algorithm with a simple regret ε^2 and hence potentially with a cumulative regret of $2\varepsilon^2 MT$. Thus, we consider the problem of distributed mean estimation and show that unless all agents send $\Omega(d\log T)$ bits of information to the server, no estimator can estimate θ^* within an accuracy of $1/MT$ with probability at least $2/3$. For this case, we only consider the optimal rates instead of any sublinear rates.

The proof borrows ideas and techniques from the classical results in (Duchi et al., 2014). In particular, the proof is similar to those of Proposition 2 and Theorem 1 in (Duchi et al., 2014). However, it is different in its own sense as it builds upon those results and strengthens the lower bounds with the missing logarithmic factors. Thus, this proof might be of independent interest to the larger community.

The basic idea in the proof is to construct a Markov chain $\mathbf{V} \to X \to Y$, where X represents the collection of all the observations at all the agents and Y denotes the messages sent by the agents. In particular, $X = (X^{(1)}, X^{(2)}, \ldots, X^{(M)})$ and $Y = (Y_1, Y_2, \ldots, Y_M)$, where $X^{(j)}$ denotes the set of observations at agent j and Y_j denotes the messages sent by agent j to the server. Since the agents cannot communicate with each other directly, we assume Y_j depends only on $X^{(j)}$.

We begin with construction the hard instance for the random variable $\mathbf{V}$. Let $\mathcal{V} = \{\pm 1\}^{d-1}$ and $\boldsymbol{\upsilon} = (\upsilon_1, \upsilon_2, \ldots, \upsilon_\ell) \in \mathcal{V}^\ell$ for some $\ell \in \mathbb{N}$ specified later. For each $\boldsymbol{\upsilon} \in \mathcal{V}^\ell$, we define a vector $\mu_{\boldsymbol{\upsilon}} \in R^{d-1}$ given by

$$\mu_{\boldsymbol{\upsilon}} := \sum_{r=1}^{\ell} \frac{2^{-r}}{\sqrt{d-1}} \upsilon_r.$$

Note that $\|\mu_{\boldsymbol{\upsilon}}\|_2 \leq \sum_{r=1}^{\ell} \frac{2^{-r}}{\sqrt{d-1}} \|\upsilon_r\| \leq \sum_{r=1}^{\ell} 2^{-r} \leq 1$. Thus, $\mu_{\boldsymbol{\upsilon}} \in \mathcal{B}_{d-1}(1)$ for all $\boldsymbol{\upsilon} \in \mathcal{V}^\ell$. We also define a lifting map $f : \mathcal{B}_{d-1}(1) \to \mathcal{S}_{\geq 0}^d$, where $\mathcal{S}_{\geq 0}^d$ denotes the hemisphere where the last coordinate only takes on non-negative values. It maps a vector $x \in \mathcal{B}_{d-1}(1)$ to a vector $x' \in \mathcal{S}_{\geq 0}^d$, where $x'_{1:d-1} = x$ and $x'_d = \sqrt{1 - \|x\|^2}$. In other words, f lifts a point in unit hypersphere in $d-1$ to the corresponding point on the unit hyper(hemi)sphere in $\mathbb{R}^d$ by adding the last component. From the definition, one can note that f is a bijection. Furthermore, one can note that for any two points $x, x' \in \mathcal{B}_{d-1}$, we have $\|x - x'\|_2 \leq \|f(x) - f(x')\|_2 \leq \sqrt{2}\|x - x'\|_2$.

Let $\mathbf{V}$ be a random variable drawn from the set $\mathcal{V}^\ell$. We construct a linear bandit instance with $\theta^* = f(\delta\mu_{\mathbf{V}})$ for some $\delta \in (0,1)$ whose value is specified later. Since, f is a bijection that preserves distances upto a constant, it equivalent to consider the problem of estimating $\theta^* = \theta_{\mathbf{V}} = \delta\mu_{\mathbf{V}}$, where the operation f is assumed to have been carried out implicitly. Hence, for the remainder of the proof, we focus only of the problem on estimating $\theta_{\mathbf{V}}$.

To specify the observation model, we first need to set up some notations and definitions. Given a $u \in \{-1, 1\}$ and $\rho \in [0, 1]$, we define $P_{u,\rho}$ to be the distribution that assigns a mass of $(1 + \rho)/2$ to u and $(1 - \rho)/2$ to $-u$. We overload the definition of $P_{u,\rho}$ for when $u = (u_1, u_2, \dots, u_p)^\top \in \{-1, 1\}^p$ is a vector. In this case, a sample from $P_{u,\rho}$ is a vector whose i^{th} coordinate is drawn according to $P_{u_i,\rho}$, independently of others. For a given value of $\mathbf{V} = \boldsymbol{v}$, each agent j, independent of other agents, receives an collection of T i.i.d. samples, denoted by $X^{(j)} = \{X^{(j,k)}\}_{k=1}^n$ for $j = 1, 2, \dots, M$. Each sample $X^{(j,k)}$ is obtained by first drawing $\tilde{V}^{(j,k)}(\boldsymbol{v}) = (\tilde{V}^{(j,k,1)}(\boldsymbol{v}), \dots, \tilde{V}^{(j,k,\ell)}(\boldsymbol{v})) \in \mathcal{V}^\ell$, where $\tilde{V}^{(j,k,r)}(\boldsymbol{v}) \sim P_{v_r,\rho_0}$ for $r = 1, 2, \dots, \ell$ and then setting $X^{(j,k)} = \mu_{\tilde{V}^{(j,k)}(\boldsymbol{v})}$. In the above definition $\rho_0 := \delta/(T\ell)$. If $X_i^{(j)}$ denotes the vector obtained by taking the i^{th} coordinate of all the T samples at agent j, then from the above definition, one can note that $X_i^{(j)}$'s are independent across i, that is, each coordinate is independent of others.

Having specified the underlying mean vector and the observation model, the next step is to use strong data processing inequality (Duchi et al., 2014) to quantitatively relate $I(\mathbf{V}; Y_j)$ and $I(X^{(i)}; Y_j)$. To establish this relation, we make use of Lemma 5 from (Duchi et al., 2014). Before invoking the Lemma, we first need to establish a bound on the likelihood ratio of any realization x_i, of the random variable $X_i^{(j)}$, under two different values of $\mathbf{V}$, say $\boldsymbol{v}, \boldsymbol{v}'$ and also specify the measurable sets over which the bound holds. Throughout this part, we carry out all the analysis for a fixed agent j.

Note that any realization x_i can be mapped to the realizations $\{\tilde{v}_i^{(j,k)}\}_{k=1}^T$, where $\tilde{v}_i^{(j,k)} = (\tilde{v}_i^{(j,k,1)}, \dots, \tilde{v}_i^{(j,k,\ell)})$. For any fixed value of $r \in \{1, 2, \dots, \ell\}$, consider the set $\tilde{v}_i^{(j,1:T,r)} = \{\tilde{v}_i^{(j,1:T,r)}\}_{k=1}^T$ which only contains values from $\{-1, 1\}$ drawn from $P_{v_{r,i},\rho_0}$. Here $v_{r,i}$ denotes the i^{th} coordinate of v_r. Suppose this collection contains T_1 instances of $+1$ and $T - T_1$ of -1, then the likelihood ratio any under any pair $(\boldsymbol{v}, \boldsymbol{v}')$ can be bounded as $\left(\frac{1+\rho_0}{1-\rho_0}\right)^{|T - 2T_1|}$. If S_r denotes the absolute value of sum of the elements in $\tilde{v}_i^{(j,1:T,r)}$, then this bound on the likelihood ratio can be rewritten as $\left(\frac{1+\rho_0}{1-\rho_0}\right)^{S_r}$. Since all ℓ sequences in $\{\tilde{v}_i^{(j,1:T,r)}\}_{r=1}^\ell$ are independent of each other, the likelihood ratio can be bounded as

$$\sup_{x_i} \sup_{\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}^\ell} \frac{\Pr(x_i | \boldsymbol{v})}{\Pr(x_i | \boldsymbol{v}')} \leq \left(\frac{1 + \delta/(T\ell)}{1 - \delta/(T\ell)}\right)^{S_1} \cdots \left(\frac{1 + \delta/(T\ell)}{1 - \delta/(T\ell)}\right)^{S_\ell} = \left(\frac{1 + \delta/(T\ell)}{1 - \delta/(T\ell)}\right)^{S_1 + \cdots + S_\ell}.$$

To construct a measurable set G_i corresponding to each coordinate i on which the likelihood ratio can be appropriately bounded, consider the set $\mathcal{Z}$ defined as

$$\mathcal{Z} = \left\{ (v_1, \dots, v_T) \in \{-1, +1\}^T : \left| \sum_{r=1}^T v_r \right| \leq a\sqrt{2T} + \delta/\ell \right\}$$

for some $a > 0$ to be determined later. We set $G_i = \{x_i : \tilde{v}_i^{(j,1:T,r)} \in \mathcal{Z} \; \forall \; r \in \{1, 2, \dots, \ell\}\}$ for all coordinates i. That is, G_i consists of all sequences x_i such that all its corresponding $\tilde{v}$ sequences belong to $\mathcal{Z}$. When $X_i^{(j)}$ belongs to the σ-field generated by the elements of G_i, we can bound the likelihood ratio as

$$\begin{aligned} \sup_{x_i \in \sigma(G_i)} \sup_{\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}^\ell} \frac{\Pr(x_i | \boldsymbol{v})}{\Pr(x_i | \boldsymbol{v}')} &\leq \left(\frac{1 + \delta/(T\ell)}{1 - \delta/(T\ell)}\right)^{\ell(a\sqrt{2T} + \delta/\ell)} \\ &\leq \exp\left(3\ell(a\sqrt{2T} + \delta/\ell) \cdot \frac{\delta}{(T\ell)}\right) \\ &\leq \exp\left(3(a + \delta/\ell)\delta\sqrt{\frac{2}{T}}\right) := \exp(\varphi). \end{aligned}$$

for all $T \geq 4\ell/3$. As the last step to invoke the Lemma, we define $E_i^{(j,r)} = \mathbb{1}\{\tilde{v}_i^{(j,1:T,r)} \in \mathcal{Z}\}$ and $E_i^{(j)} = \prod_{r=1}^\ell E_i^{(j,r)}$, where $\mathbb{1}\{A\}$ denotes the indicator variable for the event A. Using the definition of G_i, we can also define $E_i^{(j)}$ as $\mathbb{1}\{X_i^{(j)} \in G_i\}$. Lastly, we also define $E^{(j)} = \prod_{i=1}^{d-1} E_i^{(j)}$.

We are now all set to invoke Lemma 5 from (Duchi et al., 2014). Using the Lemma, we can conclude that for the message sent by agent j, Y_j, the following inequality holds

$$I(\mathbf{V}; Y_j) \leq 2(e^{4\varphi} - 1)^2 I(X^{(j)}; Y_j | E^{(j)} = 1) + \sum_{i=1}^{d-1} H(E_i^{(j)}) + \sum_{i=1}^{d-1} H(\mathbf{V}_i) \Pr(E_i^{(j)} = 0),$$

where $H(W)$ denotes the entropy of the random variable W and $\mathbf{V}_i$ refers to the tuple formed by taking the i^{th} coordinate of all the elements in tuple represented by $\mathbf{V}$.

For the first term, note that for $T \geq 200(a+1)^2$, $\varphi \leq 5/16$ implying that $2(\exp(4\varphi) - 1)^2 \leq 2(8\varphi)^2 = 128\varphi^2$. Using the standard concentration bounds for Bernoulli random variables, we can conclude that $\Pr(E_i^{(j,r)} = 0) \leq 2\exp(-a^2)$ and consequently $\Pr(E_i^{(j)} = 0) \leq 2\ell \exp(-a^2)$. On plugging these results into the previous equation, we obtain,

$$\begin{aligned}
I(\mathbf{V}; Y_j) &\leq \frac{2304(a + \delta/\ell)^2 \delta^2}{T} I(X^{(j)}; Y_j | E^{(j)} = 1) + \sum_{i=1}^{d-1} \sum_{r=1}^{\ell} H(E_i^{(j,r)}) + 2\sum_{i=1}^{d-1} \ell^2 \exp(-a^2) \\
&\leq \frac{2304(a + \delta/\ell)^2 \delta^2}{T} \sum_{k=1}^{T} I(X^{(j,k)}; Y_j | E^{(j)} = 1, X^{(j,1:(k-1))}) + (d-1)\ell(h_2(2e^{-a^2}) + 2\ell e^{-a^2}) \\
&\leq \frac{2304(a + \delta/\ell)^2 \delta^2}{T} \sum_{k=1}^{T} \min\{H(X^{(j,k)} | E^{(j)} = 1, X^{(j,1:(k-1))}), H(Y_j | E^{(j)} = 1, X^{(j,1:(k-1))})\} \\
&\qquad + (d-1)\ell(h_2(2e^{-a^2}) + 2\ell e^{-a^2}) \\
&\leq \frac{2304(a + \delta/\ell)^2 \delta^2}{T} \sum_{k=1}^{T} \min\{H(X^{(j,k)}), H(Y_j)\} + (d-1)\ell(h_2(2e^{-a^2}) + 2\ell e^{-a^2}) \\
&\leq \frac{2304(a + \delta/\ell)^2 \delta^2}{T} \sum_{k=1}^{T} \min\{(d-1)\ell, H(Y_j)\} + (d-1)\ell(h_2(2e^{-a^2}) + 2\ell e^{-a^2})) \\
&\leq 2304(a + \delta/\ell)^2 \delta^2 \min\{(d-1)\ell, H(Y_j)\} + (d-1)\ell(h_2(2e^{-a^2}) + 2\ell e^{-a^2}),
\end{aligned}$$

where we used the relation $I(W; W') \leq \min\{H(W), H(W')\}$ for two random variables along with the fact that conditioning reduces entropy. In the above equations $h_2(p) := -p \log_2(p) - (1-p)\log_2(1-p)$, denotes the entropy of a Bernoulli random variable with mean p, for $p \in [0, 1]$. Since each message Y_j depends only on $X^{(j)}$, we have,

$$\begin{aligned}
I(\mathbf{V}; Y) &\leq \sum_{j=1}^{M} I(\mathbf{V}; Y_j) \\
&\leq \sum_{j=1}^{M} \left[2304(a + \delta/\ell)^2 \delta^2 \min\{(d-1)\ell, H(Y_j)\} + (d-1)\ell(h_2(e^{-a^2}) + \ell e^{-a^2})\right].
\end{aligned}$$

This gives us the bound on $I(\mathbf{V}; Y)$ in terms of $H(Y_j)$, which is a lower bound on the required communication cost to send the message corresponding to agent j. The last step is use Fano's inequality to translate this bound to a bound on the estimation error.

Let $\hat{\theta}(Y)$ denote the estimate obtained at the server using the received messages Y. Similar to the previous case, let $\hat{\mathbf{V}} := \arg\min_{\boldsymbol{v} \in \mathcal{V}^\ell} \|\hat{\theta}(Y) - \theta_{\boldsymbol{v}}\|_2$ denote the corresponding estimate of $\mathbf{V}$. For $\hat{\mathbf{V}} \neq \mathbf{V}$, the estimation error $\|\hat{\theta}(Y) - \theta_{\boldsymbol{v}}\|_2^2$ satisfies $\|\hat{\theta}(Y) - \theta_{\boldsymbol{v}}\|_2^2 \geq \|\theta_{\hat{\mathbf{V}}} - \theta_{\mathbf{V}}\|_2^2/4 \geq \delta^2 \|\mu_{\hat{\mathbf{V}}} - \mu_{\mathbf{V}}\|_2^2 \geq \delta^2 4^{-\ell-1}/(d-1)$. The last bound follows by noting that $\|\mu_{\boldsymbol{v}} - \mu_{\boldsymbol{v}'}\| \geq \delta 2^{-\ell}/\sqrt{d-1}$ for $\boldsymbol{v} \neq \boldsymbol{v}'$.

We set $a := 2\sqrt{\log(4M\ell)}$, $\delta^2 := \left[9216(a + 1/\ell)\sum_{j=1}^{M} \min\left\{1, \frac{H(Y_j}{(d-1)\ell}\right\}\right]^{-1}$ and $\ell = \log_4(T)$. Since $\mathbf{V} \to Y \to \hat{\mathbf{V}}$

forms a Markov chain, Fano's inequality tells us that

$$\begin{aligned}
\Pr(\hat{\mathbf{V}} \neq \mathbf{V}) &\geq 1 - \frac{I(\mathbf{V}; Y) + \log 2}{\log |\mathcal{V}^\ell|} \\
&\geq 1 - \frac{1}{(d-1)\ell} \left\{ \frac{(d-1)\ell}{4} + \sum_{j=1}^{M} \left[(d-1)\ell (h_2(e^{-a^2}) + \ell e^{-a^2}) \right] + \log 2 \right\} \\
&\geq 1 - \left\{ \sum_{j=1}^{M} \left[\frac{1}{4M} \min\left\{1, \frac{H(Y_j)}{(d-1)\ell}\right\} + \frac{6}{80M^2\ell^2} + \frac{1}{256M^4\ell^3} \right] + \frac{\log 2}{(d-1)\ell} \right\}.
\end{aligned}$$

In the last step, we used the value of a along with the fact that $h_2(p) \leq 1.2\sqrt{p}$ for all $p \in [0,1]$. For $T \geq 2048$, we have that $\Pr(\hat{\mathbf{V}} \neq \mathbf{V}) \geq \frac{4}{5} - \frac{1}{4} = \frac{11}{20} \geq \frac{1}{3}$. Hence, we can conclude that the estimation error is at least $\frac{C}{T \sum_{j=1}^{M} \min\left\{1, \frac{H(Y_j}{(d-1)\ell}\right\}}$ for some universal constant $C > 0$ with probability at least $1/3$. Consequently, unless $\sum_{j=1}^{M} \min\left\{1, \frac{H(Y_j}{(d-1)\ell}\right\}$ is $\Omega(M(d-1)\ell)$, the estimation error will satisfy $\Omega(1/MT)$ with probability at least $1/3$. This implies that for at least $\Omega(M)$ agents $H(Y_j)$ should be at least as large as $(d-1)\ell$ bits. In other words, the uplink cost (per agent) should be at least $\Omega(d \log T)$ bits since $\ell = \log_4 T$.

D. Sparse PLS

In this section, provide additional details and analysis of Sparse-PLS. We begin with the pseudo codes.

Algorithm 5 Sparse-PLS Norm Estimation: Agent $j \in \{1, 2, \ldots, M\}$

```
Input: The set of actions $\mathcal{B}_s$
Set $k \leftarrow 1$
while True do
    Play each vector in $\mathcal{B}_s$ for $s_k$ times and compute the sample mean $\hat{\theta}_k^{(j)}$
    $\tilde{\theta}_k^{(j)} \leftarrow \textsc{Clip}(\hat{\theta}_k^{(j)}, R_k + B_k)$
    $Q(\tilde{\theta}_k^{(j)}) \leftarrow \textsc{StoQuant}(\tilde{\theta}_k^{(j)}, \alpha_k, R_k + B_k)$
    Send $Q(\tilde{\theta}_k^{(j)})$ to the server
    if received terminate from server then
        break
    else
        $k \leftarrow k + 1$
    end if
end while
```

Algorithm 6 Sparse-PLS Norm Estimation: The Server

```
Input: The set of actions $\mathcal{B}_s$
Set $k \leftarrow 1$
while True do
    Compute $\hat{\theta}_k^{(\text{SERV})} := \arg\min_\theta \frac{d}{m} \left\| \frac{1}{M} \sum_{j=1}^{M} Q(\tilde{\theta}_k^{(j)}) - X\theta \right\|_2^2 + \lambda_k \|\theta\|_1$
    if $\tau_k \leq \frac{1}{4} \|\hat{\theta}_k^{(\text{SERV})}\|$ then
        Server sends terminate to all agents
        break
    else
        $k \leftarrow k + 1$
    end if
end while
```

In the pseudo codes for Sparse-PLS, X refers to the $\mathbb{R}^{m \times d}$ matrix obtained by stacking the vectors in $\mathcal{B}_s$ one under the other. The parameters for Sparse-PLS are set as $s_k := \lceil 40\sigma^2 d \log(16MK/\delta)4^k \rceil$, $t_k := \lceil mMs_k^2/d \rceil$, $R_k := 2^{-k}\sqrt{m/d}$, $B_k = 7\tau_k\sqrt{m/d}$, $\tau_k := 3 \cdot 2^{(-(k+1))}/\sqrt{M}$, $\alpha_k = \alpha_0\sigma\sqrt{s/s_k}$, $\beta_k = \beta_0\tau_k$ and $\lambda_k := 4\sigma\sqrt{\frac{3}{2ms_k}}(\sqrt{\log(2d)} + \sqrt{\log(4/\delta)})$. The underlying philosophy behind the choice of these parameters is similar to that of PLS. The only additional parameters in Sparse-PLS is the regularization constant λ_k and the size of the set $\mathcal{B}_s$, m. λ_k is chosen based on analysis of the LASSO estimator (Rigollet & Hütter, 2017) while m is chosen to ensure that X satisfies the restricted eigenvalue condition (Bickel et al., 2009). In particular, based on the result in Baraniuk et al. (2008), we set $m = 80(s\log(150d/s) + \log(4/\delta))$ which ensures that with probability at least $1 - \delta/2$ over the randomness of $\mathcal{B}_s$ the following holds for any $\theta \in \mathcal{C}_s$,

$$\frac{3}{4}\|\theta\|_2 \leq \sqrt{\frac{d}{m}}\|X\theta\|_2 \leq \frac{5}{4}\|\theta\|_2.$$

In the above definition $\mathcal{C}_s = \{\theta : \|\theta_{S^c}\|_1 \leq 3\|\theta_S\|_1$ for all $S \subset \{1, 2, \ldots, d\}$ with $|S| \leq s\}$ where θ_S refers to the sub-vector of θ corresponding to the coordinates indexed by S.

Algorithm 7 Sparse-PLS Refinement: Agent $j \in \{1, 2, \ldots, M\}$

```
Input: The epoch index at the end of Norm Estimation stored as $k_0$, The set of actions $\mathcal{B}_s$
$\bar{\theta}_{k_0-1} \leftarrow 0, k \leftarrow k_0$
while time horizon $T$ is not reached do
    Play each vector in $\mathcal{B}_s$ for $s_k$ times and compute the sample mean $\hat{\theta}_k^{(j)}$
    $\tilde{\theta}_k^{(j)} \leftarrow \textsc{Clip}(\hat{\theta}_k^{(j)} - \bar{\theta}_{k-1}, R_k + B_k)$
    $Q(\tilde{\theta}_k^{(j)}) \leftarrow \textsc{StoQuant}(\tilde{\theta}_k^{(j)}, \alpha_k, R_k + B_k)$
    Send $Q(\tilde{\theta}_k^{(j)})$ to the server
    Receive $Q(\hat{\theta}_k^{(\mathrm{SERV})})$ from the server
    $\bar{\theta}_k \leftarrow \bar{\theta}_{k-1} + Q(\hat{\theta}_k^{(\mathrm{SERV})})$
    if $k = k_0$ then
        Set $\mu_0 \leftarrow \|\bar{\theta}_k\|_2$
    end if
    Play the action $a = \bar{\theta}_k/\|\bar{\theta}_k\|$ for the next $t_k$ rounds.
    $k \leftarrow k + 1$
end while
```

Algorithm 8 Sparse-PLS Refinement: The Server

```
Input: The epoch index at the end of Norm Estimation stored as $k_0$, The set of actions $\mathcal{B}_s$
$\bar{\theta}_{k_0-1} \leftarrow 0, k \leftarrow k_0$
while time horizon $T$ is not reached do
    Receive $Q(\tilde{\theta}_k^{(j)})$ from all the agents
    Compute $\hat{\theta}_k^{(\mathrm{SERV})} = \bar{\theta}_{k-1} + \arg\min_\theta \frac{d}{m}\left\|\frac{1}{M}\sum_{j=1}^M Q(\tilde{\theta}_k^{(j)}) - X(\theta - \bar{\theta}_{k-1})\right\|_2^2 + \lambda_k\|\theta\|_1$
    $Q(\hat{\theta}_k^{(\mathrm{SERV})}) \leftarrow \textsc{DetQuant}(\hat{\theta}_k^{(\mathrm{SERV})} - \bar{\theta}_{k-1}, \beta_k, B_k + \tau_k)$ and broadcasts it to all agents
    $k \leftarrow k + 1$
end while
```

Before the proof of Theorem 6.1, we state and prove a lemma analogous to Lemma B.2 for sparse bandits.

Lemma D.1. *For any epoch k in Sparse-PLS, the estimate $\hat{\theta}_k^{(\mathrm{SERV})}$ satisfies*

$$\|\hat{\theta}_k^{(\mathrm{SERV})} - \theta^*\| \leq \tau_k,$$

with probability at least $1 - \delta/2$.

Proof. Similar to the proof of Lemma B.2, we use induction and also define $\bar{\theta}_{k-1} := 0$ for all epochs k during the norm estimation stage.

In Sparse-PLS, the estimate $\hat{\theta}_k^{(\text{SERV})}$ is obtained by minimizing

$$
\begin{aligned}
\mathcal{L}_k(\theta) &:= \frac{d}{m}\left\|\frac{1}{M}\sum_{j=1}^{M} Q(\tilde{\theta}_k^{(j)}) - X(\theta - \bar{\theta}_{k-1})\right\|_2^2 + \lambda_k\|\theta\|_1 \\
&= \frac{d}{m}\left\|\frac{1}{M}\sum_{j=1}^{M}(\tilde{\theta}_k^{(j)} + \eta_k^{(j)}) - X(\theta - \bar{\theta}_{k-1})\right\|_2^2 + \lambda_k\|\theta\|_1 \\
&= \frac{d}{m}\left\|\frac{1}{M}\sum_{j=1}^{M}(\hat{\theta}_k^{(j)} - X\bar{\theta}_{k-1})\mathbb{1}_{R_k+B_k} + \frac{1}{M}\sum_{j=1}^{M}\eta_k^{(j)} - X(\theta - \bar{\theta}_{k-1})\right\|_2^2 + \lambda_k\|\theta\|_1 \\
&= \frac{d}{m}\left\|\frac{1}{M}\sum_{j=1}^{M}(\hat{\theta}_k^{(j)} - X\bar{\theta}_{k-1})\mathbb{1}_{R_k+B_k} + \frac{1}{M}\sum_{j=1}^{M}\eta_k^{(j)} - X(\theta - \bar{\theta}_{k-1})\right\|_2^2 + \lambda_k\|\theta\|_1 \\
&= \frac{d}{m}\left\|X\theta^* + \frac{1}{M}\sum_{j=1}^{M}\Delta_k - X\bar{\theta}_{k-1} + \frac{1}{M}\sum_{j=1}^{M}\eta_k^{(j)} - X(\theta - \bar{\theta}_{k-1})\right\|_2^2 + \lambda_k\|\theta\|_1 \\
&= \frac{d}{m}\left\|X\theta^* + \frac{1}{M}\sum_{j=1}^{M}\Delta_k + \frac{1}{M}\sum_{j=1}^{M}\eta_k^{(j)} - X\theta\right\|_2^2 + \lambda_k\|\theta\|_1,
\end{aligned}
$$

where Δ_k corresponds to clipped sub-Gaussian observation noise. Using the result of Theorem 2.18 in (Rigollet & Hütter, 2017), we can conclude that

$$
\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\|_2^2 \leq \frac{1024 sd\log(8K/\delta)}{mM}\left(\frac{\sigma^2}{s_k} + \frac{\alpha_k^2}{4s}\right).
$$

Plugging in the choice of s_k and α_k yields,

$$
\|\hat{\theta}_k^{(\text{SERV})} - \theta^*\|_2 \leq \frac{3}{\sqrt{M}} \cdot 2^{-(k+1)} = \tau_k,
$$

with probability at least $1 - \delta/K$. Here, similar to proof of Lemma B.2 the choice of R_k and B_k allows us to use the clipped sub-Gaussian concentration which is implicitly used while invoking the result from Rigollet & Hütter (2017).

Using an argument similar to the one in Lemma B.2, we can conclude that the above inequality holds for epochs during the algorithm with probability at least $1 - \delta/2$.

□

D.1. Proof of Theorem 6.1

The analysis for the regret performance of Sparse-PLS is almost identical to that of PLS, as described in Appendix B. Once again, the regret is decomposed into the sum of regret incurred during the norm estimation stage and the refinement stage.

Recall that the regret during the norm estimation stage of PLS is bounded using the result in Lemma B.2. Since Lemma D.1 is identical to Lemma B.2, the proof of Lemma 4.2 follows through almost unchanged for the case of Sparse-PLS. The only difference in the case of Sparse-PLS is that there are m actions in the basis set instead of d as in the case of PLS. This scales the corresponding term in the regret incurred during the norm estimation stage by a factor of m/d resulting in an overall regret of $\mathcal{O}(\sqrt{sdMT\log(d/\delta)}\log(MT)\log(\log T/\delta))$.

Similarly, the analysis in the refinement stage also follows through identically but for a couple of minor changes. The regret during a exploration sub-epoch is also scaled by a factor of m/d for the same reason as in the case of the norm estimation stage. The analysis for exploration sub-epoch follows through as is with the different value of t_k which is also scaled by a factor of s/d. Carrying out the same steps with these updated values result in an overall regret of $\mathcal{O}(\sqrt{sdMT\log(d/\delta)}\log(MT)\log(\log T/\delta))$, as required.

For the communication cost, the bound on the downlink cost follows as in case for PLS. For the case of the uplink cost, note that in Sparse-PLS, the transmitted vector lies in $\mathbb{R}^m$. Using the same argument in used on the proof of Lemma 4.5 for vectors in $\mathbb{R}^m$ along with the updated choice of R_k, B_k and α_k, we can conclude that each uplink message is at most $\mathcal{O}(m)$ bits long, resulting in an overall uplink cost of $\mathcal{O}(m \log T) = \mathcal{O}(s \log dT)$.

E. Extensions

E.1. Beyond the Unit Ball

The unit ball assumption on the action space can be relaxed to smooth action spaces without any change to the algorithm. We refer an action space $\mathcal{A}$ to be smooth if for any θ, θ' in the unit ball, $\|a(\theta) - a(\theta')\| \leq L\|\theta - \theta'\|$ for some $L > 0$, where $a(\theta) = \arg\max_{v \in \mathcal{A}} \langle v, \theta \rangle$. The smoothness condition allows one to directly use the estimate of θ^* as a proxy for θ^* for pure exploitation. In particular, for $\hat{\theta}$ satisfying $\|\theta^* - \hat{\theta}\| \leq \tau$, the regret incurred by using $\hat{\theta}$ as a proxy can be written as $\langle a(\theta^*), \theta^* \rangle - \langle a(\hat{\theta}), \hat{\theta} \rangle \leq \|a(\theta^*) - a(\hat{\theta})\| \|\theta^* - \hat{\theta}\| \leq L\tau^2$. This result is generalization of Lemma 4.3 to smooth action spaces. The rest of the analysis follows as is, yielding the result.

The assumption on the action space can be completely done away with to allow for general arbitrary actions by making two minor modifications to the PLS algorithm. Firstly, during the exploration sub-epoch, the orthogonal basis of $\mathbb{R}^d$ is replaced by a $\mathcal{B} \subseteq \mathcal{A}$, where the set $\mathcal{B}$ satisfies $|\mathcal{B}| = d$, $\text{span}(\mathcal{B}) = \mathbb{R}^d$ and that all the eigenvalues of $\sum_{x \in \mathcal{B}} xx^\top$ are greater than some $\lambda > 0$. The existence of such a set $\mathcal{B}$ is a very mild assumption on the action set. We also set the length of the exploration sub-epoch $s_k = \lceil 40(\sigma/\lambda)^2 d \log(8MK/\delta) 4^k \rceil$ and modify the remaining parameters appropriately. Secondly, on account of the possible non-smoothness of the action set, pure exploitation using $\hat{\theta}$ as a proxy may no longer yield optimal regret guarantees as the optimal action may change by a lot even by a small perturbation in θ^*. The non-smoothness necessitates a small amount of continuous exploration in the "exploitation" phase to avoid sticking on one action, which we ensure using the uncertainty ellipsoid technique. In particular, for the r^{th} instant during the exploitation sub-epoch of the k^{th} epoch, take the action $A_r = \arg\max_{a \in \mathcal{A}_k} \langle a, \breve{\theta}_{r-1} \rangle + \Gamma a^\top C_{r-1}^{-1} a$. In the above description, $\mathcal{A}_k = \{a \in \mathcal{A} : \langle a, \bar{\theta}_k \rangle \geq \max_{a' \in \mathcal{A}} \langle a', \bar{\theta}_k \rangle - 2\tau_k\}$, $C_r = C_{r-1} + A_r A_r^\top$, $\breve{\theta}_{r-1} = C_{r-1}^{-1} \sum_{l=1}^{r} A_l(y_l - \langle A_l, \bar{\theta}_k \rangle)$, where A_l denotes the action taken at the l^{th} instant, y_l denotes the corresponding reward, $C_0 = I$ (the identity matrix) and $\Gamma > 0$ is an appropriately chosen constant.

The analysis for the modified version of PLS is very similar to that of the original one. Since $\mathcal{B}$ also spans $\mathbb{R}^d$, the error bounds on $\hat{\theta}^{(\text{SERV})}$ follow as is. For the regret analysis, the regret for the exploration sub-epochs remains unchanged. In the "exploitation" sub-epoch, we can use result from Rusmevichientong & Tsitsiklis (2010); Chu et al. (2011) to conclude that the regret incurred during the sub-epoch is $\mathcal{O}(d\sqrt{t_k})$, which upon substituting the value of t_k yields the same bound as in the current version, upto constants. The final bound on the regret then follows exactly as described in Appendix B.3.

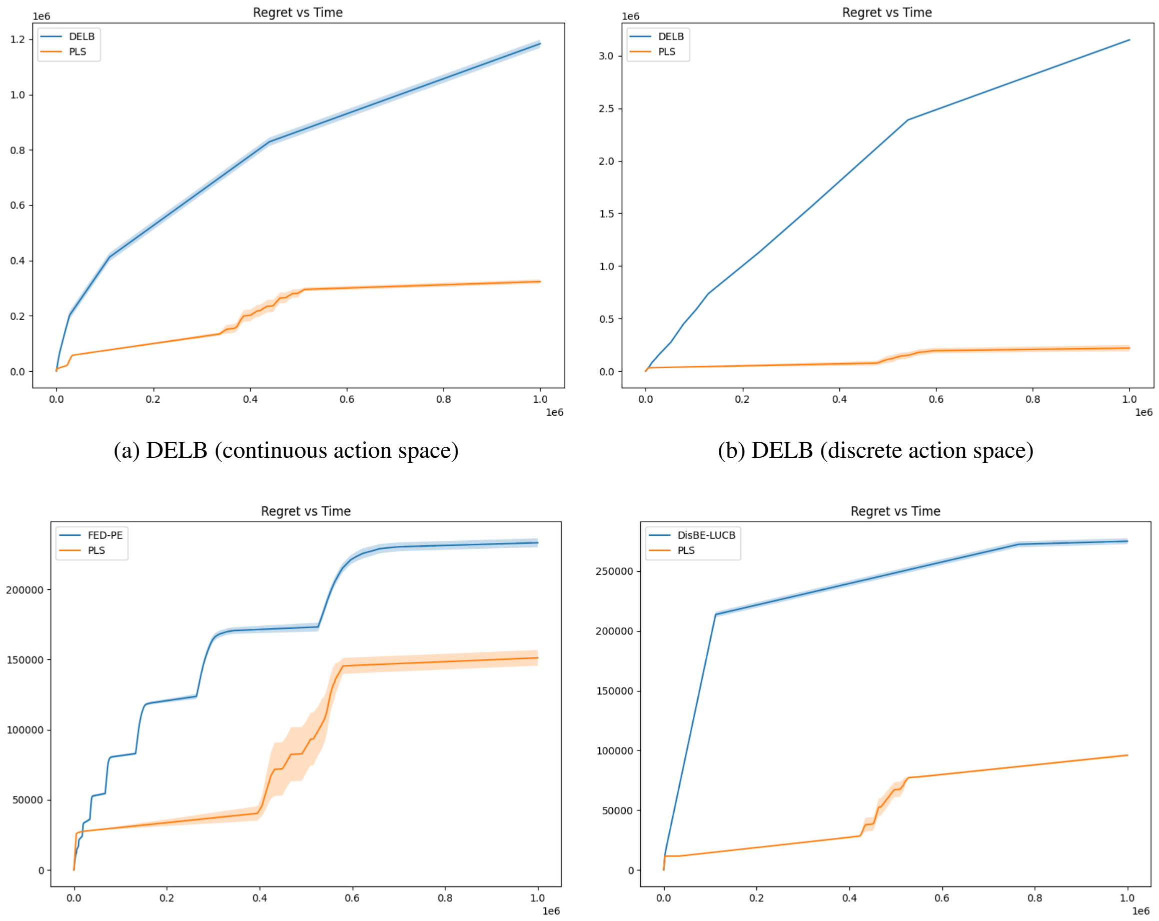
E.2. Beyond Linear Bandits

The idea of progressive learning and sharing and be extended to convex optimization to achieve optimal accuracy-communication trade-off. In a follow up work (Salgia et al., 2023), we show how this idea of PLS can be used to design algorithm for distributed stochastic convex optimization that achieves optimal regret and communication guarantees. The primary idea is to progressively learn and share information about x^*, the minimizer, as opposed to about θ^*. The proposed approach, called Communication Efficient Adaptive Learning (CEAL), builds upon minibatch gradient descent, where the batch size is adaptively tuned to the gradient at that iterate using the norm estimation routine developed for PLS.

F. Empirical Studies

In this section, we provide empirical evidence that corroborates our theoretical findings. We compare our proposed PLS algorithm with three popular distributed linear bandit algorithms, namely, Distributed Elimination for Linear Bandits (DELB) (Wang et al., 2019), Federated Phased Elimination (Fed-PE) (Huang et al., 2021) and Distributed Batch Elimination Linear Upper Confidence Bound (DisBE-LUCB) (Amani et al., 2022).

We consider a distributed linear bandit instance with $d = 20$, $M = 10$ agents which is run for a time horizon of $T = 10^6$ steps. The underlying mean reward vector is drawn uniformly from the surface of a unit ball. The rewards are corrupted with a zero mean Gaussian with unit variance. We plot the averaged cumulative regret for different algorithms considered over the time horizon of T in Fig. 1 and report the uplink and downlink communication costs in Table 1. Recall that the uplink



(a) DELB (continuous action space)

(b) DELB (discrete action space)

(c) Fed-PE

(d) DisBE-LUCB

Figure 1: Cumulative Regret vs Time for different algorithms. The bold line represents the mean obtained over 10 Monte Carlo runs and the shaded region represents the region of error bars corresponding to one standard deviation.

communication cost is defined as the number of bits sent by *one* agent to the server and the downlink cost is defined as the number of bits broadcast by the server. All the results are reported after averaging over 10 Monte Carlo runs. For a fair comparison, we assume that the real numbers are represented by $\log(MT)$ bits, which comes out to 24 bits in our setting as opposed to 32 for regular floats. We, however, for the purpose of implementation transfer the full float representation, which works in favor of the other algorithms.

F.1. Experimental Setup

Since DELB, Fed-PE and DisBE-LUCB are designed for different settings, we perform a pairwise comparison of PLS with each of these algorithms based on the setting for which they are original designed for. We describe each of experimental setups in detail below.

- DELB: Since the underlying setting in DELB is the same as that considered in PLS, we carry out two sets of experiment to compare the performance of PLS against that of DELB. In the first experiment, we consider a linear bandit instance with unit ball as the action space. In the second experiment, we consider an action space consisting of $K = 120$ actions drawn from a unit ball at random.

- FED-PE: We consider the shared parameter setting described in Huang et al. (2021). For each agent, we choose $K = 120$ actions randomly from the unit ball, independent of other agents. The phase lengths for FED-PE are set to be

Experiment	Algorithm	Uplink Cost	Downlink Cost
DELB continuous action space	DELB PLS	531.8 103.0	7400.2 139.6
DELB discrete action space	DELB PLS	336.0 112.9	4700 155.1
Federated homogeneous setting	Fed-PE PLS	72960 301.1	249900 402.5
Stochastic Contexts	DisBE-LUCB PLS	2000 188.9	2000 309.5

Table 1: Communication cost (in bits) for various algorithms against PLS in their corresponding experimental setup. Reported values are obtained after averaging over 10 Monte Carlo runs.

growing in powers of 2, similar to the choice adopted in Huang et al. (2021).

- DisBE-LUCB: We adopt the same experimental setup as considered in Amani et al. (2022) for the evaluation of DisBE-LUCB. However, instead of a distribution over 100 instances, we consider a distribution over 50 instances with $K = 40$ actions in each of them.

Depending on the experimental setup, we either use the implementation of PLS as outlined in the main paper or that of its extension to general action spaces as described in Appendix E.1.

F.2. Results

PLS offers a significantly lower cumulative regret as compared to DELB in both the cases and the significant improvement of PLS over DELB in terms of communication cost also evident from Table 1. In particular, the difference in downlink cost is significant due to the linear scaling with the number of agents for DELB. For the case for FED-PE, PLS outperforms it in terms of regret incurred despite not being designed for this heterogeneous setting. In terms of communication cost, the scaling with respect to the number of actions significantly deteriorates the uplink cost for FED-PE and the dependence on d^2 worsens the downlink cost. On the other hand, PLS continues to enjoy both small uplink and downlink costs. Lastly, PLS also offers superior performance over DisBE-LUCB, both in terms of regret and communication cost, even within the stochastic contextual bandit setup. The above experimental results demonstrate the improved performance of PLS over existing distributed linear bandit algorithms across a variety of setups.