

Non-Stationary Representation Learning in Sequential Linear Bandits

YUZHEN QIN ¹ (Member, IEEE), TOMMASO MENARA ² (Member, IEEE),
SAMET OYMAK ³ (Member, IEEE), SHINUNG CHING ⁴ (Member, IEEE),
AND FABIO PASQUALETTI ¹ (Member, IEEE)

¹Department of Mechanical Engineering, University of California, Riverside, CA 92521 USA

²Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093 USA

³Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA

⁴Department of Electrical and Systems Engineering and Biomedical Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA

CORRESPONDING AUTHOR: YUZHEN QIN (e-mail: yuzh.qin@gmail.com)

This work was supported in part by Awards ARO-78259-NS-MUR and AFOSR-FA9550-20-1-0140.

ABSTRACT In this paper, we study representation learning for multi-task decision-making in non-stationary environments. We consider the framework of sequential linear bandits, where the agent performs a series of tasks drawn from different environments. The embeddings of tasks in each environment share a low-dimensional feature extractor called *representation*, and representations are different across environments. We propose an online algorithm that facilitates efficient decision-making by learning and transferring non-stationary representations in an adaptive fashion. We prove that our algorithm significantly outperforms the existing ones that treat tasks independently. We also conduct experiments using both synthetic and real data to validate our theoretical insights and demonstrate the efficacy of our algorithm.

INDEX TERMS Linear bandits, non-stationary representations, representation learning.

I. INTRODUCTION

Humans are naturally endowed with the ability to learn and transfer experience to later unseen tasks. The key mechanism enabling such versatility is the abstraction of past experience into a ‘basis set’ of simpler representations that can be used to construct new strategies much more efficiently in future complex environments [1], [2].

Inspired by this observation, recent years have witnessed an increasing interest in the study of *representation learning* [3]. Representation learning is an important tool to perform transfer learning, wherein common low-dimensional features shared by tasks are inferred and generalized. It underlies major advances in a variety of fields including language processing [4], drug discovery [5], and reinforcement learning [6]. Due to its promising seminal impact, there are many recent theoretical studies on representation learning (e.g., see [7]–[11]). Yet, existing literature focuses on representation learning for batch tasks and is restricted to static representations, thus relying on the working assumption that *one representation fits all tasks*.

Most realistic decision-making scenarios feature two challenges: i) the learning agent faces tasks that appear in sequence, and ii) the agent may encounter distinct environments sequentially (see Fig. 1(a)), where learning a single representation is no longer sufficient. Humans can perform extraordinarily well in such scenarios because of their flexibility to adapt to new environments. For instance, in the Wisconsin Card Sorting Task (WCST, see Fig. 1(b)), participants are asked to match a sequence of stimulus cards to one of the four cards on the table according to some sorting rule—number, shape, or color. The sorting rule changes every now and then without informing the participants. Earlier studies have shown that, in general, humans perform very well on this task (e.g., [12]). By contrast, some classical learning algorithms, such as the tabular-Q learning and the deep-Q learning, struggle in WCST (as we show in Section VI). Unlike humans, these algorithms can neither abstract succinct information from experience nor adapt to new environments. This observation reveals the need to develop more human-like reasoning and a more fluid approach in representation learning.

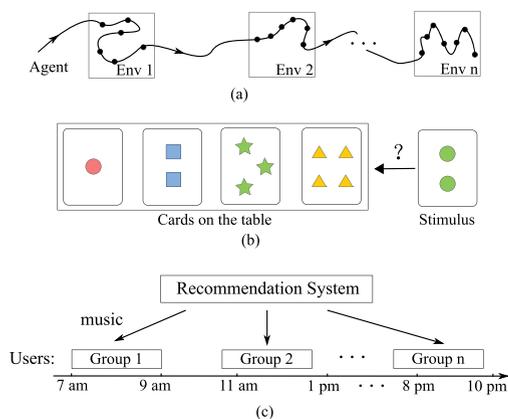


FIGURE 1. Scenarios with non-stationary environments. (a) Changing environments that the agent enters to perform tasks in sequence. (b) The Wisconsin Card Sorting Task, a classical test to assess cognitive capabilities that features a changing sorting rule; (c) A recommendation system that recommends music to distinct users at different times of the day based on the users' tastes.

This paper takes an important step towards a deeper theoretical understanding of representation learning in non-stationary environments. As a prototypical sequential decision-making scenario, we consider a series of linear bandits, where each bandit represents a task, and the objective is to maximize the cumulative reward by interacting with these tasks. Moreover, sequential tasks are drawn from different environments. Importantly, tasks in the same environment share a low-dimensional linear representation. Our modeling choice can be used in a wide range of applications. For instance, consider an adaptive system that recommends music to users of a streaming platform. This system naturally fits our model: non-stationary environments arise due to the fact that distinct groups of active users can have different preferences at different times of the day (see Fig. 1(c)). The goal of this paper is to analytically study representation learning in dynamical environments akin to the above recommendation system.

Related Work: As a well-known model to capture the exploration-exploitation dilemma in decision-making, multi-armed bandits have attracted extensive attention (see [13] for a survey). A variety of generalized bandit problems have been investigated, where the situations with non-stationary reward functions [14], restless arms [15], satisficing reward objectives [16], heavy-tailed reward distributions [17], risk-averse decision-makers [18], and multiple players [19] are considered. Distributed algorithms have also been proposed to tackle bandit problems (e.g., see [20]–[27]). Recently, some studies take into account the nature that sequentially collected data is adaptive and propose novel algorithms to further improve the performance [28], [29].

Besides the above work that focuses on single-task bandits, some efforts have also been made to study multi-task problems. The core of multi-task bandits is to learn and transfer interrelationships across multiple tasks, aiming to improve

decision-making efficiency compared to treating tasks independently. Various types of interrelationships can be leveraged to boost the learning agent's performance, including the mean of tasks drawn from a stationary distribution [30], [31], similarity of task coefficients in linear bandits [32], and resemblance in contexts of arms in contextual bandits [33]. Recently, learning low-dimensional subspaces shared by task coefficients has also been proven to improve performance in simultaneous linear bandits [34]–[36].

Contribution: This paper seeks to develop methods to learn and transfer non-stationary representations across sequential bandits. In contrast to recent studies that play bandits simultaneously [34], [35], the sequential setting is more realistic, and representation learning in this context is more challenging. First, there does not exist a low-dimensional representation that fits all bandits. The agent needs to adapt to dynamical environments. Second, the agent does not know when environment changes happen, thus has no knowledge of the number of tasks drawn from an environment. It is therefore challenging to strike the balance between learning and transferring the representation.

We propose an adaptive algorithm to overcome these challenges. Within each environment, the algorithm alternates between representation exploration and exploitation, making it flexible to different durations of the environments. Meanwhile, we incorporate a change-detection strategy into our algorithm to adapt to non-stationary environments. We further obtain a regret upper bound $\tilde{O}(dr\sqrt{mSN} + Sr\sqrt{N})$, with d being the task dimension, r the representation dimension, m the number of environments, S the number of tasks, and N the number of rounds for each task. Our regret significantly outperforms the baseline $\Theta(Sd\sqrt{N})$ of algorithms treating tasks independently. To demonstrate our theoretical results, we perform some experiments using synthetic data and LastFM data. Simulation results also show that our algorithm considerably outperforms classical reinforcement learning algorithms in WCST.

Our preliminary work [37] presents limited theoretical findings on representation learning in the sequential setting, but we go well beyond that in this paper by considering non-stationary environments and providing a comprehensive account of the results. Further, we demonstrate the broad applications of our results by presenting more experiments.

Organization: The problem setup is in Section II. In Section III, we present an algorithm that performs representation learning in a single environment. An environment-change-detection algorithm is provided in Section IV. In Section V, the main algorithm is presented by putting together Sections III and IV. Illustrative experiments are reported in Section VI. Concluding remarks appear in Section VII.

Notation: Given a matrix $A \in \mathbb{R}^{d \times r}$, $r < d$, $\text{Span}(A)$ denotes its column space, $A_{\perp} \in \mathbb{R}^{d \times (d-r)}$ the orthonormal basis of the complement of $\text{Span}(A)$, $[A]_i$ its i th column, $\sigma_i(A)$ its i th largest singular value, and $\|A\|_F$ its Frobenius norm. We use $\|x\|$ to denote the L_2 norm if x is a vector and the

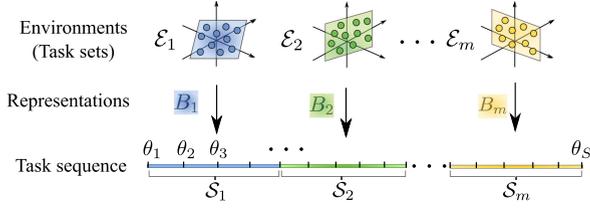


FIGURE 2. Sequential tasks in non-stationary environments. The tasks are taken from different environments $\mathcal{E}_1, \dots, \mathcal{E}_m$, forming subsequences $\mathcal{S}_1, \dots, \mathcal{S}_m$. The tasks in each environment share a low-dimensional representation (described by $B_k, k = 1, \dots, m$). In each subsequence \mathcal{S}_k , there are τ_k tasks, but τ_k 's are not known *a priori*.

spectral norm if x is a matrix. Let A and B be two orthonormal basis of two subspaces $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^{d \times r}$. Define $\sin \theta_1(A, B) := \sin(\theta_1)$ and $\sin \theta_r(A, B) := \sin(\theta_r)$, where θ_1 and θ_r are computed by the singular value decomposition $A^\top B = UDV^\top$ with $D = \text{diag}(\cos \theta_1, \dots, \cos \theta_r)$ satisfying $0 \leq \theta_r \leq \dots \leq \theta_1 \leq \frac{\pi}{2}$. Following [38], the distance between \mathcal{A} and \mathcal{B} is defined as $\text{dist}(\mathcal{A}, \mathcal{B}) = \|\text{diag}(\sin \theta_1, \dots, \sin \theta_r)\|_F = \|A^\top B_\perp\|_F$. Given a positive number x , $\lceil x \rceil$ denotes the smallest integer that is greater than or equal to x . Given two functions $f, g: \mathbb{R}^+ \rightarrow \mathbb{R}^+$, we write $f(x) = O(g(x))$ if there is $M_o > 0$ and $x_0 > 0$ such that $f(x) \leq M_o g(x)$ for all $x \geq x_0$, and $f(x) = \tilde{O}(g(x))$ if $f(x) = O(g(x) \log^k(x))$. Also, we denote $f(x) = \Omega(g(x))$ if there is $M_\Omega > 0$ and $x_0 > 0$ such that $f(x) \geq M_\Omega g(x)$ for all $x \geq x_0$, and $f(x) = \Theta(g(x))$ if $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$.

II. PROBLEM SETUP

In this paper, we consider the following multi-task sequential linear bandits model:

$$y_t = x_t^\top \theta_{q(t)} + \eta_t, \quad (1)$$

where $x_t \in \mathcal{A} \subseteq \mathbb{R}^d$ is the action taken by the agent at round t , $\theta_{q(t)} \in \mathbb{R}^d$ is the bandit coefficient, and η_t is the additive noise that is assumed to be zero-mean 1-sub-Gaussian, i.e., $\mathbb{E}[e^{\lambda \eta_t}] \leq \exp(\frac{\lambda^2}{2})$ for any $\lambda > 0$.

Notice that the coefficient vector $\theta_{q(t)}$ is time-varying. We assume that $q(t) = \lceil \frac{t}{N} \rceil$, where $t = 1, 2, \dots, SN$. That is, the agent plays S bandits in sequence and interacts with each bandit for N rounds.¹ Then, the task sequence can be denoted as $\mathcal{S} := \{\theta_1, \theta_2, \dots, \theta_S\}$.

We further assume that these tasks are drawn from m different environments, $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_m$ (each \mathcal{E}_k is a set of tasks), independent of the agent's strategy. Specifically, the task sequence \mathcal{S} can be divided into m consecutive subsequences (see Fig. 2), i.e.,

$$\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m\},$$

such that $\mathcal{S}_k \subseteq \mathcal{E}_k$ for $k = 1, 2, \dots, m$. Denote τ_k as the number of tasks in \mathcal{S}_k ; it satisfies $\sum_{i=k}^m \tau_k = S$. We assume that, for

each environment \mathcal{E}_k , there exists a matrix $B_k \in \mathbb{R}^{d \times r_k}$ with orthonormal columns such that for any $\theta \in \mathcal{E}_k$,

$$\exists \alpha \in \mathbb{R}^{r_k} : \theta = B_k \alpha.$$

This assumption is motivated by the fact that real-world tasks often share low-dimensional structures, called *representations* [3]. As for the examples in Fig. 1, the representation can describe the common preferences of a user group on a music streaming platform or a certain sorting rule of the Wisconsin Sorting Task. With a slight abuse of terminology, we refer to B_k as the representation of the k th environment, $k = 1, \dots, m$. For simplicity, we assume that the representations have the same dimension,² i.e., $r_k = r$ for all k .

The goal is to maximize the cumulative reward $\sum_{t=1}^{SN} y_t$ by interacting with the sequential bandits in the non-stationary environments. The agent knows $\sigma(t), N, d$, and r , but has no knowledge of θ_i, B_k , and τ_k for any $i = 1, \dots, S$ and $k = 1, \dots, m$. To measure the agent's performance in the $T := SN$ rounds, we introduce the (pseudo-)regret

$$R_T = \sum_{t=1}^T (x_t^* - x_t)^\top \theta_{q(t)}, \quad (2)$$

where $x_t^* = \arg \max_{x \in \mathcal{A}} x^\top \theta_{q(t)}$ is the optimal action that maximizes the reward at round t . Maximizing the cumulative reward is then equivalent to minimizing the regret R_T .

Following existing studies (e.g., see [39], [40]), we assume that the following assumptions on the action set \mathcal{A} and the task coefficients θ_i hold throughout this paper.

Assumption 1 (Linear bandits): We assume that: (a) the action set \mathcal{A} is the ellipsoid of the form $\{x \in \mathbb{R}^d : x^\top M^{-1} x \leq 1\}$, where M is a symmetric positive definite matrix, and (b) there are positive constants θ_{\min} and θ_{\max} so that $\theta_{\min} \leq \|\theta_s\| \leq \theta_{\max}$ for all $s \in \{1, 2, \dots, S\}$.

Limitation of Independent Strategies: Under the stated assumptions, previous work (e.g., see [39]–[41]) shows that the regret of single-task bandits is lower bounded by $\Omega(d\sqrt{d})$. Intuitively, if the sequential bandits $\mathcal{S} = \{\theta_1, \theta_2, \dots, \theta_S\}$ are independently played by those standard algorithms, the best performance is optimally $\Theta(Sd\sqrt{N})$.

Potential Benefits of Representation Learning: In the case of an oracle where the representations B_1, \dots, B_m are known, it holds that $\theta_i = B_k \alpha_i$ for any $\theta_i \in \mathcal{S}$. Letting $z_t = B_k^\top x_t$, the d -dimensional bandit $y_t = x_t^\top \theta_i + \eta_t$ becomes a r -dimensional one $y_t = z_t^\top \alpha_i + \eta_t$ with $\alpha_i \in \mathbb{R}^r$. Following [39]–[41], one can show that the best performance of an algorithm can be optimally $\Theta(Sr\sqrt{N})$, which indicates a significant performance improvement compared to the standard algorithms if $r \ll d$. The reason is that learning to make decisions is accomplished in much lower-dimensional subspaces. The above observation implies potential benefits of representation learning, that is, exploring and exploiting the

¹We make this assumption for simplicity. Our results can be readily generalized to the case where bandits are played for different rounds.

²Our results can be applied to the situation with heterogeneous r_k by simply letting $r = \max_k r_k$.

Algorithm 1: Representation Exploration (RepE).

```

1: Input:  $N, N_1, \{a_1, \dots, a_d\}$ .
2: for  $t = 1 : N_1$  do
3:   Take the action  $x_t = a_i, i = (t - 1 \bmod d) + 1$ 
4: end for
5: Compute  $\hat{\theta} = (X_E X_E^\top)^{-1} X_E Y_E$ 
6: for  $t = N_1 + 1 : N$  do
7:   Take the action  $x_t = \arg \max_{x \in \mathcal{A}} x^\top \hat{\theta}$ 
8: end for
    
```

Algorithm 2: Representation Transfer (RepT(\hat{B})).

```

1: Input:  $N, N_2, \hat{B} \in \mathbb{R}^{d \times r}, \{a_1, \dots, a_r\}$ 
2: for  $t = 1 : N_2$  do
3:   Take the action  $x_i = a'_i, i = (t - 1 \bmod r) + 1$ 
4: end for
5: Compute  $\hat{\alpha} = (\hat{B}^\top X_T X_T^\top \hat{B})^{-1} \hat{B}^\top X_T Y_T$  and  $\hat{\theta} = \hat{B} \hat{\alpha}$ 
6: for  $t = N_2 + 1 : N$  do
7:   Take the action  $x_t = \arg \max_{x \in \mathcal{A}} x^\top \hat{\theta}$ 
8: end for
    
```

Algorithm 3: SeqRepL.

```

1: Input:  $L, N_1, N_2, n$ .           Initialize:  $\hat{P} = 0_{d \times d}$ .
2: for cycle  $n = 1, 2, \dots$  do
3:   Rep Exploration: play  $L$  tasks in  $\mathcal{T}$  using RepE,
4:      $\hat{P} = \hat{P} + \hat{\theta}_i \hat{\theta}_i^\top \triangleright$  fixed duration:  $L$ 
5:    $\hat{B} \leftarrow$  the left-singular vectors associated with the
     largest  $r$  singular values of  $\hat{W} = \frac{1}{nL} \hat{P}$ 
6:   Rep Transfer: play  $nL$  tasks in  $\mathcal{S}_\tau$  using RepT( $\hat{B}$ )
7:      $\triangleright$  increasing duration:  $nL$ 
8: end for
    
```

underlying low-dimensional structure between bandit tasks can facilitate more efficient decision-making.

In our setting, representation learning has two main challenges. First, how can the agent explore and exploit representation in the sequential setting? Particularly, striking the balance between exploration and exploitation becomes more challenging than the situation where bandits are played concurrently [34], [35]. There is a trade-off between the need to explore more sequential tasks (more data samples) to obtain a more accurate representation estimate and the incentive to exploit the learned representation for more efficient learning and higher immediate rewards. Second, how can the agent deal with environment changes? The remainder of this paper aims to address these challenges.

III. REPRESENTATION LEARNING IN SEQUENTIAL BANDITS: WITHIN-ENVIRONMENT POLICY

In this section, we show how the agent can improve its performance by using representation learning in the sequential setting. The main result is the sequential representation

learning algorithm (SeqRepL, see Algorithm 3), a within-environment policy that deals with individual segments of tasks drawn from the same environment.

A. SEQUENTIAL REPRESENTATION LEARNING ALGORITHM

The key feature of SeqRepL is to balance representation exploration and exploitation without knowing the duration of each environment. To show how SeqRepL works, we first restrict our attention to a series of bandit tasks in this section:

$$\mathcal{T} = \{\theta_1, \dots, \theta_\tau\}, \quad (3)$$

where the number of tasks τ is unknown, and the representation shared by the tasks is $B \in \mathbb{R}^{d \times r}$.

SeqRepL operates in a cyclic manner, which is inspired by the PEGE algorithm [39]. It alternates between two sub-algorithms—*representation exploration* (RepE, see Algorithm 1) and *representation transfer* (RepT, see Algorithm 2). Both RepE and RepT are explore-then-commit (ETC) algorithms, consisting of two stages, i.e., exploration and commitment.

RepE: At the exploration stage of N_1 rounds, d actions, a_1, \dots, a_d , are repeatedly taken in sequence. These actions can be arbitrarily chosen but need to be linearly independent such that they span the action space defined by \mathcal{A} . In this paper, we simply let $a_i = \lambda_0 e_i$, where e_i is the i th canonical vector of \mathbb{R}^d and $\lambda_0 > 0$ is such that $a_i \in \mathcal{A}$ for all i . Then, the coefficient θ is estimated by the least-squares regression

$$\hat{\theta} = (X_E X_E^\top)^{-1} X_E Y_E,$$

where $X_E = [x_1, \dots, x_{N_1}]$ and $Y_E = [y_1, \dots, y_{N_1}]^\top$ respectively collect the actions and rewards at this stage. At the commitment stage, the greedy action $x_t = \arg \max_{x \in \mathcal{A}} x^\top \hat{\theta}$ is taken for $N - N_1$ times.

RepT: Different from RepE, RepT utilizes $\hat{B} \in \mathbb{R}^{d \times r}$ as a plug-in surrogate for the unknown representation B to learn the coefficient θ . The exploration stage is accomplished in the r -dimensional space $\text{Span}(\hat{B})$. Specifically, r actions a'_1, \dots, a'_r , are repeatedly taken for N_2 rounds. These actions can be arbitrarily chosen in $\text{Span}(\hat{B})$ such that they are linearly independent. In this paper, we let $a'_i = \lambda_0 [\hat{B}]_i$, where $\lambda_0 > 0$ is such that $a'_i \in \mathcal{A}$. To estimate θ , RepT computes α first. Observe that $y_i = x_i^\top \theta + \eta_i = x_i^\top B \alpha + \eta_i$. Then, α is estimated with \hat{B} by the least-squares regression

$$\hat{\alpha} = (\hat{B}^\top X_T X_T^\top \hat{B})^{-1} \hat{B}^\top X_T Y_T,$$

where $X_T = [x_1, x_2, \dots, x_{N_2}]$ and $Y_T = [y_1, y_2, \dots, y_{N_2}]^\top$. Subsequently, θ is estimated by $\hat{\theta} = \hat{B} \hat{\alpha}$. Similar to RepE, $x_t = \arg \max_{x \in \mathcal{A}} x^\top \hat{\theta}$ is taken at the commitment stage.

SeqRepL: As shown in Algorithm 3, there are two phases in each cycle of SeqRepL. Specifically, in each cycle $n \in \{1, 2, \dots\}$, these phases are:

1) *Representation Exploration Phase:* L tasks (L is to be designed) are played using RepE. Let $\hat{W} = \frac{1}{nL} \sum_i \hat{\theta}_i \hat{\theta}_i^\top$ where $\hat{\theta}_i$'s are all the estimated coefficients obtained by RepE in all the previous n cycles. Then, the representation B is estimated

by performing the singular value decomposition (SVD) to \hat{W} . Specifically, \hat{B} takes the singular vectors associated with the r largest singular values of \hat{W} , i.e., $\hat{B} = \hat{U}_1$ with $\hat{U}_1 \in \mathbb{R}^{d \times r}$ taken from the SVD: $\hat{W} = [\hat{U}_1, \hat{U}_2] \hat{\Sigma} \hat{V}^\top$.

2) *Representation Transfer Phase*: the latest representation estimate \hat{B} is transferred. Specifically, nL sequential tasks are played using $\text{RepT}(\hat{B})$.

Notice that for any cycle n , L more tasks are played using RepT than the previous $(n-1)$ th cycle. We next show that this alternating scheme balances representation exploration and exploitation excellently.

Assumption 2 (Task diversity): Suppose that there exist an integer ℓ and a constant $\nu > 0$ such that any subsequence of length ℓ in the sequence in Eq. (3) satisfies $\sigma_r(W_s W_s^\top / \ell) \geq \frac{\nu}{r} > 0$ for any s , where $W_s = [\theta_{s+1}, \dots, \theta_{s+\ell}]$.

This assumption states that the sequential tasks well spread the entire r -dimensional subspace. It ensures that this subspace can be reconstructed before all the bandit tasks are played, which is crucial to allow for transfer learning in the sequential setting. A similar assumption is found in [34], wherein bandits are played concurrently. Representation learning in the sequential setting is more challenging, thus our assumption is slightly stronger.

Theorem 1 (Upper bound of SeqRepL): Let the agent play the series of bandits in Eq. (3) using SeqRepL in Algorithm 3. Select an L such that³ $L = \Theta(\ell)$, and let $N_1 = dr\sqrt{N/L}$ and $N_2 = r\sqrt{N}$. Then, the regret $R_{\tau N}$ of SepRepL satisfies

$$\mathbb{E}[R_{\tau N}] = \tilde{O}\left(dr\sqrt{\tau N} + \frac{dr}{\lambda_0^2 \nu^2} \sqrt{\tau N} + \frac{d\ell}{r} \sqrt{\tau N} + \tau r \sqrt{N}\right). \quad (4)$$

The fourth term in (4) is the regret incurred when transferring the oracle representation, and the first three terms include the regret that results from representation exploration and transferring the estimated representation with errors.

Remark 1 (Performance comparison): Recall that if the same series of bandits are played using standard algorithms that play bandits independently, e.g., UCB [41], PEGE [39], and ETC [42], the best regret is optimally $\Theta(\tau d \sqrt{N})$. Compared to these algorithms, SeqRepL can have better or worse performance (i.e., ‘‘positive’’ or ‘‘negative’’ transfer), depending on the properties of the bandit tasks in (3):

- If $\tau \gg \max\{r^2, \frac{r^2}{\lambda_0^4 \nu^4}, \frac{\ell^2}{r^2}\}$ and $d \gg r$, using SeqRepL can significantly improve the performance;⁴
- If $\tau \ll \max\{r^2, \frac{r^2}{\lambda_0^4 \nu^4}, \frac{\ell^2}{r^2}\}$, the bound in (4) implies that the cost of learning the representation can overwhelm

³Here, the exact knowledge of ℓ is not required; instead, knowing the order of ℓ is sufficient. In practice, the assumption can be further relaxed. In Fig. 5, we will show that a wide range of L can be chosen without knowing ℓ , while still guaranteeing the performance of our algorithms.

⁴We note that positive transfer can still occur in practice without satisfying this inequality. In Figs. 4 and 5, we let $\tau = 400$ for $\lambda_0 = 1$, $\ell = r = 3$, and $\nu \approx 0.01$, much smaller than 9×10^8 required by the inequality here. Nevertheless, our algorithms still outperform the standard ones significantly.

the possible benefits of transfer learning. Then, using SeqRepL may result in a situation of negative transfer.

Our algorithm is particularly advantageous over the standard ones when there are a large number of tasks in the sequence. Also, in sharp contrast to existing bandit algorithms using representation learning (e.g., [34], [35]), our algorithm requires no knowledge of τ . \triangle

The following corollary provide an upper bound of SeqRepL if ℓ in Assumption 2 is of the order of r^2 .

Corollary 1: Assume that ℓ in Assumption 2 is of the order of r , i.e., $\ell = \Theta(r^2)$. Let $N_1 = d\sqrt{rN}$ and $N_2 = r\sqrt{N}$. Then, the regret of SepRepL for the sequential bandits in Eq. (3) satisfies $\mathbb{E}[R_{\tau N}] = \tilde{O}(dr\sqrt{\tau N} + \frac{dr}{\lambda_0^2 \nu^2} \sqrt{\tau N} + \tau r \sqrt{N})$.

If $\frac{1}{\lambda_0^2 \nu^2} = \Theta(1)$, the regret bound becomes $\mathbb{E}[R_{\tau N}] = \tilde{O}(dr\sqrt{\tau N} + \tau r \sqrt{N})$. For the series of bandits in Eq. (3), it follows from [34] that the lower bound is $\Omega(d\sqrt{r\tau N} + \tau r \sqrt{N})$. Note that there is just a gap of $\tilde{O}(\sqrt{r})$ between our upper bound and this lower bound.

B. PROOF OF THEOREM 1

We first provide some instrumental results, and we refer the readers to Appendix A–C for their proofs.

Lemma 1 (Regret of RepE): Given a bandit task $\theta \in \mathcal{T}$, let the agent play it using RepE in Algorithm 1 for N rounds. Then, the regret R_N of RepE satisfies $\mathbb{E}[R_N] = O(N_1 + \frac{N}{N_1} d^2)$.

Lemma 2 (Regret of RepT): Given a bandit task $\theta \in \mathcal{T}$, assume that there exists $B \in \mathbb{R}^{d \times r}$ with orthonormal columns such that $\theta = B\alpha$ for some $\alpha \in \mathbb{R}^r$. Assume that an estimate \hat{B} is known and satisfies $\|\hat{B}^\top B_\perp\|_F \leq \varepsilon$. Let the agent play this task for N rounds using $\text{RepT}(\hat{B})$ in Algorithm 2, then the regret satisfies $\mathbb{E}[R_N] = O(N_2 + \frac{N}{N_2} r^2 + N\varepsilon^2)$.

Theorem 2 (Accuracy of learned representation): Let the agent play the series of bandits in Eq. (3) using SeqRepL in Algorithm 3. Then, for any cycle n , the estimate \hat{B} at the end of the representation exploration phase satisfies

$$\|\hat{B}^\top B_\perp\|_F \leq \tilde{O}\left(\frac{dr}{\lambda_0 \nu} \sqrt{\frac{1}{nLN_1}}\right), \quad (5)$$

with probability at least $1 - \frac{1}{kN_1}$.

Recall that $\|\hat{B}^\top B_\perp\|_F$ measures the distance between \hat{B} and the true representation B . This distance decreases with n , implying that \hat{B} becomes progressively more accurate as more tasks are explored by the RepE algorithm.

Proof of Theorem 1: In the n th cycle of SepRepL, L tasks are played in the representation exploration phase. Then, it follows from Lemma 1 that the regret in this phase, denoted as $R_{\text{RepE}}(n)$, satisfies $\mathbb{E}R_{\text{RepE}}(n) = O(LN_1 + L\frac{N}{N_1} d^2)$. From Theorem 2, we have

$$\|\hat{B}^\top B_\perp\|_F \leq \tilde{O}\left(\frac{dr}{\lambda_0 \nu} \sqrt{\frac{1}{nLN_1}}\right).$$

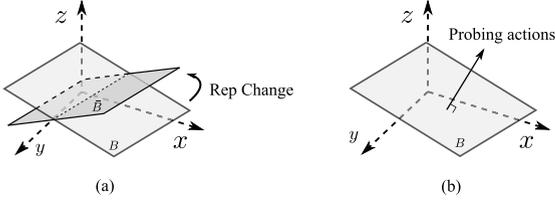


FIGURE 3. Illustration of environment change detection. (a) Along with the environment change, the representation switches from B to \bar{B} . (b) Probing actions are randomly generated in the orthogonal complement of the current representation.

Then, nL bandit tasks are played utilizing the $\text{RepT}(\hat{B})$ algorithm. It follows from Lemma 2 that the regret in the RepT phase, denoted as $R_{\text{RepT}}(n)$, satisfies

$$\begin{aligned} \mathbb{E}[R_{\text{RepT}}(n)] &= \tilde{O}\left(nLN_2 + nL\frac{N}{N_2}r^2 + nLN\frac{d^2r^2}{\lambda_0^2v^2}\frac{1}{nLN_1}\right) \\ &= \tilde{O}\left(nLN_2 + nL\frac{N}{N_2}r^2 + \frac{d^2r^2}{\lambda_0^2v^2}\frac{N}{N_1}\right). \end{aligned}$$

Observe that there are at most $\bar{L} = \lceil\sqrt{2\tau/L}\rceil$ cycles in the series (3) since $L\bar{L} + L\bar{L}(\bar{L} + 1)/2 \geq \tau$. Summing up the regret in the representation exploration and exploitation phases in all the cycles, we obtain

$$\begin{aligned} \mathbb{E}[R_{\tau N}] &= \sum_{n=1}^{\bar{L}} (R_{\text{RepL}}(n) + R_{\text{RepT}}(n)) \\ &\leq \tilde{O}\left(\bar{L}\left(LN_1 + L\frac{N}{N_1}d^2\right) + \sum_{n=1}^{\bar{L}}\left(nLN_2 + nL\frac{N}{N_2}r^2 + \frac{d^2r^2}{\lambda_0^2v^2}\frac{N}{N_1}\right)\right). \end{aligned}$$

Since $N_1 = dr\sqrt{N/L}$, $N_2 = r\sqrt{N}$ and $\bar{L} = \lceil\sqrt{2\tau/L}\rceil$, we have $\mathbb{E}[R_{\tau N}] = \tilde{O}(dr\sqrt{\tau N} + \frac{dr}{\lambda_0^2v^2}\sqrt{\tau N} + \frac{dL}{r}\sqrt{N\tau} + \tau r\sqrt{N})$.

Then, (4) follows from $L = \Theta(r^2)$. \blacksquare

IV. ENVIRONMENT CHANGE DETECTION

To handle environment changes, we propose the representation change detection algorithm (RepCD, see Algorithm 4). It is the key to endow the agent with adaptability.

A. REPRESENTATION CHANGE DETECTION ALGORITHM

To show how RepCD works, we consider the environment change where the representation switches from $B \in \mathbb{R}^{d \times r}$ to $\bar{B} \in \mathbb{R}^{d \times r}$ (see Fig. 3(a)). To detect this environment change, we seek for the tasks that do not belong to the subspace $\text{Span}(B)$. To infer whether a task is an outlier to $\text{Span}(B)$, RepCD takes some *probing actions* and monitors the rewards. The probing actions need to ensure: (1) when a task is not in $\text{Span}(B)$, it can be detected as an outlier with high probability;

Algorithm 4: Rep Change Detection (RepCD(B)).

- 1: **Input:** $B \in \mathbb{R}^{d \times r}$, n_{det} .
- 2: take n_{det} probing actions in \mathcal{A}_{det} ,
 $Y_{\text{det}} = [y_1, \dots, y_{n_{\text{det}}}]^\top$
- 3: **if** $Y_{\text{det}} \notin \mathcal{C}_{n_{\text{det}}}$ **then**
- 4: Rep Change indicator $\mathbb{I}_{\text{det}} = 1$
- 5: **end if**

and (2) when a task is in $\text{Span}(B)$, it can be falsely detected as an outlier with low probability.

The key idea is to select probing actions in the *orthogonal complement* $\text{Span}(B_\perp)$, which is illustrated in Fig. 3(b). A task θ is in the subspace $\text{Span}(B)$ if and only if $B_\perp^\top \theta = 0$. To generate an accurate test, all the directions defined by the columns of B_\perp need to be covered by the probing actions. A naive strategy is to choose $d - r$ actions by simply exhausting the columns of B_\perp , i.e., let $x_i = \lambda_0[B_\perp]_i^\top$, $i = 1, \dots, d - r$, where $\lambda_0 > 0$ is such that $\lambda_0[B_\perp]_i \in \mathcal{A}$. Taking these actions, the agent is expected to receive rewards satisfying $y_i = \lambda_0[B_\perp]_i^\top \theta + \eta_i = \eta_i$ if $\theta \in \text{Span}(B)$. If the agent receives some rewards that exceed the noise level, the task θ is likely an outlier of the current representation.

However, one may not need as many as $d - r$ probing actions if the environment change happens between two very different representations. It is also possible that more than $d - r$ probing actions are required to detect a more subtle environment change. Next, we show how to choose probing actions by taking both of these situations into account.

First, let n_{det} be the number of probing actions (we will show how to select n_{det} soon). Observe that any n_{det} can be rewritten into $n_{\text{det}} = k(d - r) + \bar{n}$, where k can be $0, 1, 2, \dots$. The first $k(d - r)$ probing actions simply take the actions $\{x_i = \lambda_0[B_\perp]_i, i = 1, \dots, d - r\}$ for k times. How to choose the remainder of \bar{n} actions is more interesting.

We require all the $d - r$ directions in $\text{Span}(B_\perp)$ to be covered, which ensures that informed decisions are made for representation change detection, especially when $k = 0$. To do that, we use the idea of *random projection* [43]. First, we generate a projection matrix P that projects from \mathbb{R}^{d-r} onto a random \bar{n} -dimensional subspace uniformly distributed in the Grassmann manifold $G_{(d-r), \bar{n}}$ (which consists of all \bar{n} -dimensional subspaces in \mathbb{R}^{d-r}). One can obtain a matrix $Q \in \mathbb{R}^{(d-r) \times \bar{n}}$ with orthonormal columns that satisfies $P = QQ^\top$. Then, the \bar{n} remaining actions are generated by taking the columns out from the matrix $M = B_\perp Q$.

Finally, we have completed selecting all the n_{det} probing actions, which are included in the set

$$\mathcal{A}_{\text{det}} := \underbrace{\{\hat{A}, \dots, \hat{A}\}}_k, \lambda_0[M]_1, \dots, \lambda_0[M]_{\bar{n}}, \quad (6)$$

where $\hat{A} := \{\lambda_0[B_\perp]_1, \dots, \lambda_0[B_\perp]_{(d-r)}\}$, and λ_0 is a scalar such that all the actions in \mathcal{A}_{det} is in \mathcal{A} .

Let $Y_{\text{det}} = [y_1, \dots, y_{n_{\text{det}}}]^\top$ collect the rewards. We build a confidence interval for Y_{det} , which is

$$\mathcal{C}_{\text{det}} = \left\{ Y \in \mathbb{R}^{n_{\text{det}}} : \left| \frac{1}{\sqrt{n_{\text{det}}}} \|Y\|_2 - 1 \right| \leq \xi_{\text{det}} \right\},$$

where ξ_{det} is the *detection threshold*. For the task θ , if $Y_{\text{det}} \in \mathcal{C}_{\text{det}}$ is observed, we say θ is not an outlier; if $Y_{\text{det}} \notin \mathcal{C}_{\text{det}}$ is observed, we say θ is an outlier and, subsequently, there is an environment change.

The next lemma shows how to select the detection threshold ξ_{det} and the number of probing actions n_{det} such that an outlier can be detected with high probability.

Lemma 3 (Outlier detection: oracle representation): Consider two representations B and \bar{B} and any task θ satisfying $\theta = \bar{B}\alpha$ for some α . Assume that $\sin \theta_r(B, \bar{B}) = \kappa_1$. Let

$$n_{\text{det}} = \left\lceil \frac{9(d-r) \log(2S^2N)}{\theta_{\min}^2 \lambda_0^2 \kappa_1^2} \right\rceil, \quad \xi_{\text{det}} = \sqrt{\frac{\log(2S^2N)}{4n_{\text{det}}}}. \quad (7)$$

Then, the task θ can be detected as an outlier to B by RepCD(B) in Algorithm 4 with probability at least $1 - O(\frac{1}{S^2N})$.

Note that the distance of the two subspaces B and \bar{B} is measured by the smallest angle $\sin \theta_r(B, \bar{B})$. From Lemma 3, it can be seen that fewer probing actions (smaller n_{det}) are needed to detect a change between two representations with a larger distance. Notice that, in Lemma 3, we have used the oracle B in RepCD to detect outliers. However, the agent usually only has access to an estimate \hat{B} . The following lemma states that if \hat{B} is sufficiently accurate, an outlier can still be detected with high probability.

Lemma 4 (Outlier detection: estimated representation): Consider the representation change in Lemma 3. Assume that \hat{B} , which satisfies $\|\hat{B}^\top B_\perp\|_F \leq \varepsilon \leq \frac{1}{3}\kappa_1$, is known. Let n_{det} and ξ_{det} be as in Eq. (7). Then, the task θ can be detected as an outlier to B by RepCD(\hat{B}) in Algorithm 4 with probability at least $1 - O(\frac{1}{S^2N})$.

V. THE MAIN ALGORITHM: CD-REPL

In this section, we provide the main results in this paper.

A. CD-REPL

We present the main algorithm, i.e., the change-detection representation learning algorithm (CD-RepL). CD-RepL uses the strategies that we have presented in the previous two sections to perform representation learning and to adapt to changing environments.

CD-RepL proceeds as follows. If a new environment is detected by RepCD (the first environment is also regarded as a new one), the agent first performs *initial representation exploration*. In this period, bL sequential tasks are played and an initial representation estimate \hat{B} is constructed. Then, using this \hat{B} , the agent starts to test every task to infer whether there is an environment change using RepCD. If there is no representation change, the agent plays the sequential bandits using SeqRepL in Algorithm 3. Note that SeqRepL starts from the $(b+1)$ th cycle instead of the first one. Meanwhile, \hat{B} is

Algorithm 5: CD-RepL.

- 1: **Input:** b, L **Initialize:** $\hat{B} = I_d, \hat{P} = \mathbf{0}$, the new Rep indicator $\mathbb{I}_{\text{det}} = 1$
- 2: **if** $\mathbb{I}_{\text{det}} = 1$ (i.e., a new environment detected) **then**
- 3: reset $\hat{P} = \mathbf{0}$;
- 4: play bL tasks using RepE with $N_1 = dr\sqrt{\frac{N}{\ell}}$,
 $\hat{P} = \hat{P} + \hat{\theta}_i \theta_i^\top$;
- 5: $\hat{B} \leftarrow$ top r singular value decomposition of
 $\hat{W} = \frac{1}{bL} \hat{P}$;
- 6: set $n_{\text{st}} = b + 1$.
- 7: **else**
- 8: invoke RepCD, update \mathbb{I}_{det}
- 9: **end if**
- 10: **if** $\mathbb{I}_{\text{det}} = 0$ **then**
- 11: invoke SeqRepL that starts from the n_{st} th cycle.
- 12: **end if**

constantly updated. Once detected a new environment, the agent restarts the above processes.

Notice that the initial representation exploration plays an important role in CD-RepL. It provides the agent with a rough but acceptable estimate of the underlying representation such that the agent can avoid false detection with high probability (see Lemma 5). Moreover, after the initial exploration SeqRepL does not need to start from the first cycle because of the initial estimate \hat{B} . By starting from $(b+1)$ th cycle, it avoids the unnecessary exploration phases in the first b cycles. Let us explain how to select b .

First, we choose the number of probing actions n_{det} and the detection threshold ξ_{det} since the choice of b depends on them. In the previous section, they are chosen in the case of two representations. For the case of m representations, we use the same idea. Let $\kappa := \min\{\sin \theta_r(B_i, B_{i+1}), i = 1, \dots, m-1\}$. Then, we let

$$n_{\text{det}} = \left\lceil \frac{9(d-r) \log(2S^2N)}{\theta_{\min}^2 \lambda_0^2 \kappa^2} \right\rceil, \quad \xi_{\text{det}} = 2\sqrt{\frac{\log(2S^2N)}{n_{\text{det}}}}. \quad (8)$$

Subsequently, we let

$$b = \left\lceil \frac{9dr\theta_{\max}^2 n_{\text{det}}}{4v^2 \sqrt{\ell N} (d-r) \log(2S^2N)} \right\rceil. \quad (9)$$

The expressions in (8) and (9) are guided by the following underlying ideas:

1) Fewer probing actions (n_{det}) and a larger detection threshold (ξ_{det}) are sufficient to detect representation changes if the distance between representations are larger.

2) The choice of b needs to balance between the need to explore more tasks such that the initial estimate of \hat{B} is sufficiently accurate to avoid false detection and the incentive to explore fewer tasks to incur less regret.

The following theorem provides an upper bound for CD-RepL, which also justifies our choice of n_{det} , ξ_{det} , and b .

Theorem 3 (Upper bound of CD-RepL): Let the agent play the sequential bandits in \mathcal{S} using the CD-RepL in Algorithm 5. Let $L = \Theta(\ell)$, n_{det} and ξ_{det} be as in Eq. (8), and b be as in

Eq. (9). Then, the regret of the CD-RepL satisfies

$$\begin{aligned} \mathbb{E}[R_T] = & \\ & \tilde{O} \left(\underbrace{\sum_{i=1}^m \left(dr\sqrt{\tau_i N} + \frac{dr}{\lambda_0^2 v^2} \sqrt{\tau_i N} + \frac{d\ell}{r} \sqrt{\tau_i N} \right)}_{(a)} + Sr\sqrt{N} \right. \\ & \left. + \underbrace{Sn_{\text{det}} + m b d r \sqrt{N} \ell}_{(b)} + \underbrace{2m}_{(c)} \right). \end{aligned} \quad (10)$$

The regret incurred by CD-RepL can be decomposed into three parts. Term (a) follows from Theorem 1, which is the regret incurred by the SepRepL for each environment. Term (b) is the regret incurred by the probing actions for each task and the initial representation exploration for each environment. Term (c) is due to unsuccessful detection and false detection. Similar to Remark 1, whether CD-SepL outperforms the existing algorithms that treat bandits independently depends on the relationship between the parameters $d, r, \ell, \lambda_0, v, \kappa, \tau, S$ and m . To clarify the dependence, we provide the following corollary.

Corollary 2: Assume that $\ell = \Theta(r)$, $1/(\lambda_0^2 v^2) = \Theta(1)$, and κ satisfies

$$\kappa \geq \max \{ \min\{p_1, p_2\}, \min\{p_3, p_4\} \}, \quad (11)$$

where

$$\begin{aligned} p_1 &= \frac{3S^{\frac{1}{4}}(d-r)^{\frac{1}{2}}\sqrt{\log(2S^2N)}}{2d^{\frac{1}{2}}r^{\frac{1}{2}}N^{\frac{1}{4}}\theta_{\min}\lambda_0}, \\ p_2 &= \frac{3(d-r)^{\frac{1}{2}}\sqrt{\log(2S^2N)}}{2r^{\frac{1}{2}}N^{\frac{1}{4}}\theta_{\min}\lambda_0}, \\ p_3 &= \frac{3(d-r)^{\frac{1}{2}}q^{\frac{1}{2}}m^{\frac{1}{2}}}{4r^{\frac{1}{2}}N^{\frac{1}{4}}S^{\frac{1}{4}}v\theta_{\min}\lambda_0}, \quad p_4 = \frac{3d^{\frac{1}{2}}(d-r)^{\frac{1}{2}}q^{\frac{1}{2}}m^{\frac{1}{2}}}{4r^{\frac{1}{2}}N^{\frac{1}{4}}S^{\frac{1}{4}}v\theta_{\min}\lambda_0}. \end{aligned}$$

Then, the regret of CD-RepL for the sequential tasks in \mathcal{S} satisfies

$$\mathbb{E}[R_{SN}] = \tilde{O} \left(\sum_{i=1}^m dr\sqrt{\tau_i N} + Sr\sqrt{N} \right). \quad (12)$$

The reason that the upper bound in Theorem 3 reduces to the one in Eq. (12) is because Terms (b) and (c) in Eq. (10) are dominated by Term (a) if κ is lower bounded as in Eq. (11). Further, observing that $\sum_{i=1}^m dr\sqrt{\tau_i N} \leq dr\sqrt{SmN}$, the upper bound of the regret in Eq. (12) can be rewritten into $\tilde{O}(dr\sqrt{mSN} + Sr\sqrt{N})$. Recall that algorithms like UCB, ETC, and PEGE (e.g., see [39]–[41]) that play the sequential bandits independently have a regret bound $\Theta(Sd\sqrt{N})$. Under the assumption $r \ll d$, CD-RepL outperforms these algorithms considerably if $S/m > r^2$. In other words, our algorithm is advantageous over the existing ones if: (a) the dimension of linear representation is much smaller than the task dimension; (b) the underlying representation does not change too fast. Notice that if the agent plays the S tasks

simultaneously, the regret upper bound can be up to $O(Sd\sqrt{N})$ even if the idea of representation learning is used. This is because the entire set of the S tasks may not share a common representation, although some of its subsets do (this point will be demonstrated in Fig. 4 in Section VI).

Remark 2: We remark that the assumption of κ in Eq. (11) is mild. The right-hand side of Eq. (11) becomes very small if $d \ll N$. \triangle

B. ANALYSIS OF THEOREM 3

Let us provide an instrumental result first, whose proof is in Appendix E.

Lemma 5 (Probability of a false detection): Consider the case where there is only one environment (i.e., $m = 1$) in the task sequence $\{\theta_1, \theta_2, \dots, \theta_S\}$, and the underlying representation is $B \in \mathbb{R}^{d \times r}$. Let the agent play this sequence of tasks using CD-RepL. Let \hat{B} be the estimated representation after the initial sample phase of bL tasks, the probability that a task θ is detected by RepCD(\hat{B}) as an outlier, denoted by $\Pr[Y_{\text{det}} \notin \mathcal{C}_{\text{det}}(\theta = B\alpha)]$, is less than $O(\frac{1}{S^2N})$.

Proof of Theorem 3: Recall that τ_i sequential tasks are taken from each set \mathcal{S}_i . Therefore, the representations change after the $(\sum_{i=1}^{p-1} \tau_i)$ -th task is played, where $p = 2, \dots, m$. For the simplicity of nation, denote $v_i = (\sum_{j=1}^i \tau_j + 1)$, $i = 1, \dots, m-1$ as the instants when the environment switches happen. Let μ_i be the detection time of the i th switch (i.e., the μ_i th task is detected as an outlier to the i th representation). Therefore, the event $D_i = \{\mu_i = v_i\}$ is a good event, which describes the situations that the i th representation switch is detected immediately after it happens. The event $G_i = \{\mu_i > v_i\}$ denotes a late or an unsuccessful detection. The event $F_i = \{\mu_i < v_i\}$ denotes a false detection, which describes the situation that an alarm is triggered when there is no representation switch.

First, we consider the case with two representations (i.e., 1 representation switch), and the sequential tasks $\{\theta_1, \dots, \theta_{\tau_1}, \theta_{\tau_1+1}, \dots, \theta_{\tau_1+\tau_2}\}$ are played using CD-RepL in

Algorithm 5. The regret of CD-RepL given D_1 satisfies

$$\begin{aligned} \mathbb{E}[R_{(\tau_1+\tau_2)N}|D_1] &\lesssim \\ &\underbrace{\sum_{i=1}^2 dr\sqrt{\tau_i N} + \frac{dr}{\lambda_0^2 v^2} \sqrt{\tau_i N} + \frac{d\ell}{r} \sqrt{N\tau_i} + \tau_i r \sqrt{N}}_{(a)} \\ &+ \underbrace{(\tau_1 + \tau_2)n_{\text{det}}}_{(b)} + \underbrace{2bdr\sqrt{N}\ell}_{(c)}. \end{aligned} \quad (13)$$

Note that term (a) is the regret incurred by SeqRepL in individual subsequence, which follows from Theorem 1; term (b) is incurred by the probing actions for each task; and term (c) results from the initial representation exploration.

If the event G_1 happens, the regret of CD-RepL satisfies

$$\mathbb{E}[R_{(\tau_1+\tau_2)N}|G_1] \lesssim$$

$$\underbrace{\left(dr\sqrt{\tau_1 N} + \frac{dr}{\lambda_0^2 v^2} \sqrt{\tau_1 N} + \frac{d\ell}{r} \sqrt{N\tau_1} + \tau_1 r \sqrt{N} \right)}_{(i)} + \underbrace{(\tau_1 + \tau_2)n_{\text{det}} + bdr\sqrt{N\ell}}_{(ii)} + \underbrace{\tau_2 N}_{(iii)} \quad (14)$$

Here, the terms (i) and (ii) are the regret in the first subsequence, which follows from Eq. (13). If the representation switch is detected after v_i or not detected at all, the regret for the second subsequence would be bounded by $O(\tau_2 N)$, which results in the term (iii). This is intuitive since there are τ_2 tasks in the second subsequence and the regret of each task is bounded by $O(N)$.

If the event F_1 happens, the regret of CD-RepL satisfies

$$\mathbb{E}[R_{(\tau_1+\tau_2)N}|F_1] \lesssim (\tau_1 + \tau_2)N. \quad (15)$$

From Lemma 3, it can be derived that $\Pr[D_1] \geq 1 - O(\frac{1}{S^2N})$. The event G_1 means that D_1 does not happen. It can be calculated that $\Pr[G_1] \leq O(\frac{1}{S^2N})$. The event F_1 means that there is at least 1 false detection in the first subsequence. From Lemma 5, we know that the probability that a task is detected as an outlier falsely is less than $O(\frac{1}{S^2N})$. Then, the probability of F_1 can be calculated as

$$\Pr[F_1] \leq 1 - \left(1 - O\left(\frac{1}{S^2N}\right) \right)^{\tau_1} \leq O\left(\frac{\tau_1}{S^2N}\right).$$

Putting together (13)–(15) and using the law of total expectation, it can be computed that the regret of CD-RepL satisfies

$$\begin{aligned} \mathbb{E}[R_{(\tau_1+\tau_2)N}] &\lesssim \\ &\sum_{i=1}^2 \left(dr\sqrt{\tau_i N} + \frac{dr}{\lambda_0^2 v^2} \sqrt{\tau_i N} + \frac{d\ell}{r} \sqrt{N\tau_i} + \tau_i r \sqrt{N} \right) \\ &+ (\tau_1 + \tau_2)n_{\text{det}} + 2bdr\sqrt{N\ell} + 2, \end{aligned} \quad (16)$$

where the last term on the right-hand side follows from $\tau_2 N \cdot O(\frac{1}{S^2N}) + (\tau_1 + \tau_2)N \cdot O(\frac{\tau_1}{S^2N}) \leq O(2)$.

The upper bound in Eq. (16) can be generalized to the case where there are $m - 1$ representation switches. In this case, the upper bound of the regret of CD-RepL becomes

$$\begin{aligned} \mathbb{E}[R_{SN}] &\lesssim \sum_{i=1}^m \left(dr\sqrt{\tau_i N} + \frac{dr}{\lambda_0^2 v^2} \sqrt{\tau_i N} + \frac{d\ell}{r} \sqrt{N\tau_i} + \tau_i r \sqrt{N} \right) \\ &+ \sum_{i=1}^m \tau_i n_{\text{det}} + mbdr\sqrt{N\ell} + 2(m - 1) \\ &\leq \sum_{i=1}^m \left(dr\sqrt{\tau_i N} + \frac{dr}{\lambda_0^2 v^2} \sqrt{\tau_i N} + \frac{d\ell}{r} \sqrt{\tau_i N} \right) + Sr\sqrt{N} \\ &+ Sn_{\text{det}} + mbdr\sqrt{N\ell} + 2m. \end{aligned}$$

which completes the proof. \blacksquare

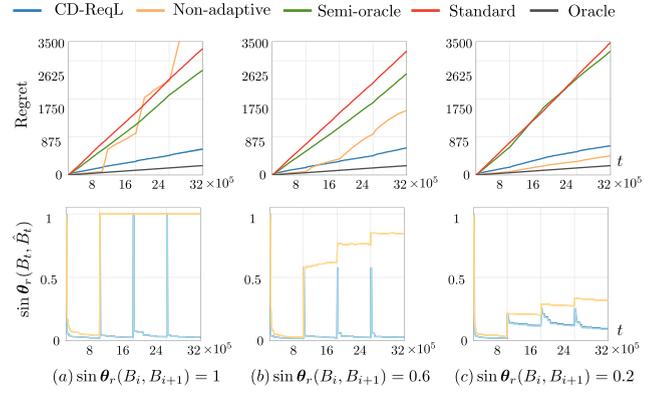


FIGURE 4. Performance comparison between different algorithms using synthetic data. Upper panels: regret; lower panels: distance between \hat{B} to the changing true representation B . In (a), (b), and (c), environment changes involving different representation distances are considered. Shaded areas contain 10 random realizations of trials.

TABLE 1 Baselines.

Algorithms	Description
Standard	play bandits independently (e.g., ETC [42], PEGE [39] and UCB [40], [41])
Semi-oracle	the single representation for all bandits is known (i.e., B such that satisfies $\theta_i = B\alpha_i$ for all i is known ⁵)
Non-adaptive	disabled environment change detection
Oracle	representations and change times are known

VI. ILLUSTRATIVE EXAMPLES

We perform some experiments to validate our theoretical results and demonstrate the efficacy of our algorithm.

Synthetic Data: We first synthesize a set of data to demonstrate our algorithm. Specifically, we consider a series of 1600 bandit tasks of dimension 20. There are four segments in this sequence, and each has 400 tasks. In each segment, there is a representation $B_i \in \mathbb{R}^{20 \times 3}$. The parameters in Assumption 2 are $\ell = 3$ and $v = 0.01$. The action set is the unit ball defined by $\mathcal{A} = \{x \in \mathbb{R}^{20} : \|x\| \leq 1\}$. The noise in the reward-generating function is assumed to be Gaussian $\mathcal{N}(0, 0.3)$. Each task is played for 2000 rounds.

We compare our algorithm (CD-RepL) with four other algorithms in Table 1. We consider several situations where environment changes involve different representation distances (which is measured by $\sin \theta_r(B_i, B_{i+1})$, $i = 1, 2, 3$).

The Oracle outperforms CD-RepL as expected since the latter needs to pay the cost of learning the representations and detecting the environment changes.

From the upper panels in Fig. 4, CD-RepL always outperforms Standard, consistent with our theoretical results. Compared with Semi-oracle, CD-RepL outperforms the existing algorithms that learn the single representation shared by all the bandits. The reason is that this representation may not

⁵Such B always has no smaller dimension than those of within-environment representations, and it may not exist if θ_i 's span the entire \mathbb{R}^d . Note that existing algorithms (e.g., [34], [35]) that play bandits simultaneously and exploit representation learning cannot outperform the semi-oracle algorithm since they need to estimate the representation.

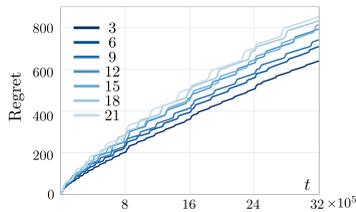


FIGURE 5. Performance of CD-SepL when L is set to different values. The parameters are the same as in Fig. 4(b).

exist or have a high dimension even if the subsets of tasks have low-dimensional representations.

Further, we disable the environment change detection of our algorithm (i.e., Non-adaptive) and compare it with CD-RepL. CD-RepL is particularly advantageous over Non-adaptive when representations change drastically (see Fig. 4(a)); the advantage decreases when it comes to more subtle changes (see Fig. 4(b)). For sufficiently small changes, Non-adaptive can even perform better (see Fig. 4(c)). This is because the price of detecting the changes and re-learning each representation may overwhelm the potential benefits of transferring the learned representations. However, CD-RepL have a much more stable performance in all situations. The lower panels in Fig. 4 illustrate that CD-SepL can detect environment changes in different situations involving drastic or subtle representation changes, which is the key to endow CD-SepL with the adaptability. By contrast, Non-adaptive can no longer learn the true representations accurately after the first environment change.

Recall that our theoretical results rely on fact that the order of ℓ in Assumption 2 is known since the number of tasks in each representation cycle is set to be $L = \Theta(\ell)$. Yet, this assumption is not required in practice. As shown in Fig. 5, CD-RepL has similar performance for a wide range of L . This implies that, even if ℓ is unknown, one can always choose an L that is likely larger than ℓ without compromising the performance much compared to the case of letting $L = \ell$.

LastFM: We use this dataset to demonstrate that our algorithm can be used to design an adaptive recommendation system as depicted in Fig. 1(b). This dataset is extracted from the music streaming service Last.fm. It contains 1892 users, 17632 artists, and a listening count of user-artist pairs. We first remove the artists that have fewer than 40 listeners and the users who listened fewer than 10 artists, and obtain a matrix $M = [m_{ij}]$ of size 411×1565 with each row representing an artists and each column a user. To generate arms and users, we use the non-negative matrix factorization for M and keep the first 20 latent features. In other words, $M \approx AU$, where $A \in \mathbb{R}^{411 \times 20}$ and $U \in \mathbb{R}^{20 \times 1565}$ are non-negative and describe the features of the artists and users, respectively. From U , we select 3 groups of users that approximately lie in distinct subspaces, consisting 11, 6, and 6 users, respectively. These users form a series of bandits $\theta_1, \dots, \theta_{23}$ that have different 2-dimensional representations. We then recommend music

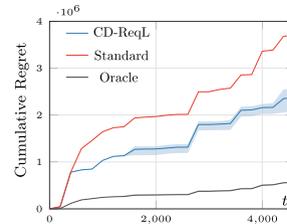


FIGURE 6. Performance comparison between different algorithms as recommendation systems using LastFM data. Shaded areas contain 10 random realizations of trials.

items to these users for 200 times from the action set \mathcal{A} that is composed of the row vectors of A . The reward is generated by $y_t = x_t^\top \theta_i + \eta_t$, where $x_t \in \mathcal{A}$ and η_t is Gaussian noise $\mathcal{N}(0, 0.2)$. In Fig. 6 it can be observed that, by learning and exploiting the representation shared by users and detecting environment changes our algorithm outperforms the existing ones that treat bandits independently.

Wisconsin Card Sorting Task (WCST): WCST, see Fig. 1(b), is typically utilized to assess human abstraction and shift of contexts [44]. Participants need to match a series of stimulus cards to one of the four cards on the table based on a sorting rule. For the stimulus card in Fig. 1(b), if the rule is color, the correct sorting action is the third card. The participants only receive feedback about whether their actions are correct. For convenience, we assume that they receive reward 1 for a correct action, and 0 otherwise. Participants do not know the current sorting rule, thus need to infer it by trial and error. The sorting rule changes every now and then, which makes the task challenging.

WCST can actually be described by a bandit problem with environment changes, where representations define the sorting rules. Specifically, we use a matrix of size 4×3 , $A = [a_1, a_2, a_3]$, to describe each card. Here, the vectors a_1, a_2 and a_3 define the number, color, and shape, respectively, and they take value from the 4-dimensional standard basis $\{e_1, e_2, e_3, e_4\}$. The matrix $A = [e_i, e_j, e_m]$ describes a card that has the i th number, the j th color, and the m th shape of the 4 cards on the table. For instance, the stimulus card in Fig. 1(b) can be described by the matrix $A = [e_2, e_3, e_1]$. Further, we use a vector B to describe the sorting rule, which takes value from the basis $\{b_1, b_2, b_3\}$ of \mathbb{R}^3 , representing the sorting rule is number, color, and shape, respectively.

As a consequence, the reward of WCST is generated by $y_t = x_t^\top \theta_t$ with $\theta_t = A_t B_{\sigma(t)}$, where x_t is the action that takes value from the 4-dimensional standard basis $\{e_1, e_2, e_3, e_4\}$, A_t is the card at round t . Notice that $B_{\sigma(t)}$ that describes the sorting rule can be regarded as the time-varying representation in the reward function.

For standard reinforcement learning algorithms, such as tabular-Q-learning and deep-Q-learning, WCST is challenging. There are in total 4^3 possible stimulus cards (4 colors, 4 numbers, 4 shapes), and for each stimulus card, there are 4 possible categories. To find the best policy, the standard

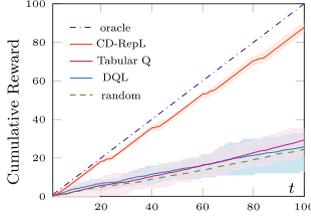


FIGURE 7. Performance comparison for different algorithms in WCST, where the sorting rule changes after ever 20 rounds. Shaded areas contain 10 realizations.

tabular-Q-learning needs many samples to construct the Q table for a single sorting rule. It is then impossible for the standard Q learning algorithm to find the optimal policy if the sorting rule changes in a few trials (in Fig. 7 it changes every 20 rounds). Being unaware of rule changes results in an even worse performance. The deep-Q-learning algorithm⁶ does not perform better since, similarly, it always needs a large number of samples to train the weights in the neural network to find the optimal policy. It can be seen from Fig. 7 that these two algorithms perform barely better than the one that takes a random action at each round.

However, by describing the WCST as a linear bandit model, we find that the problem can reduce to learning the representation $B_{\sigma(t)}$, as the correct action can be computed as $x_t^* = A_t B_t$. The problem then reduces to learn the underlying representation B_{σ} , a task that is much easier than constructing the Q table or training the weights in a Deep-Q network. Remarkably, one does not even need to learn individual θ_t to construct B_{σ} . Instead, B_{σ} can be recovered as $B_{\sigma} = (\sum_{t=1}^k A_t^T x_t x_t^T A_t)^{-1} \sum_{t=1}^k A_t^T x_t y_t$ immediately after $\sum_{t=1}^k A_t^T x_t x_t^T A_t$ becomes invertible. This indicates that our idea in this paper can apply to more general situations. As shown in Fig. 7, our algorithm significantly outperforms the other two, approaching the oracle that makes correct choice at every round. This experiment suggests that the ability to learn representations and shift attention [2], [45] to adapt to environment changes facilitates efficient learning.

VII. CONCLUDING REMARKS

In this paper, we exploit representation learning for decision-making in non-stationary environments using the framework of multi-task sequential linear bandits. We propose an efficient decision-making algorithm that learns and transfers representations online. Employing a representation-change-detection strategy, our algorithm also has the flexibility to adapt to new environments. We further obtain an upper bound

⁶Here, we formalize each input state by a 3-dimensional vector (shape, number, color)^T $\in \{1, 2, 3, 4\}^3$. The result in Fig. 7 considers a three layer neural network with 3, 12, and 4 nodes in the input, hidden, and output layers, respectively. Deeper or wider structures were also considered, but similar performances were obtained.

for the algorithm, analytically showing that it significantly outperforms the existing ones that treat tasks independently. Moreover, we perform some experiments using synthetic data to demonstrate our theoretical results. Using the LastFM data, we show that our algorithm can be applied to designing adaptive recommendation systems. In the Wisconsin Card Sorting Task, experimental results show that our algorithm considerably outperforms some classic reinforcement learning algorithms.

Directions of future work include nonlinear representation learning, representation-based clustering, and task-tailored representation generation from experience in bandit and reinforcement learning problems.

APPENDIX

A. PROOF OF LEMMA 1

Proof: Without loss of generality, we assume that N_1 is a multiple of d . Following similar steps as those in Lemma 3.4 of [39], we can obtain that after N_1 steps of exploration

$$\mathbb{E}[\|\hat{\theta} - \theta\|^2] \leq \frac{d^2}{N_1}.$$

From [39], it holds that $\max_{x \in \mathcal{A}} x^T \theta - \max_{x \in \mathcal{A}} x^T \hat{\theta} \leq J \|\theta - \hat{\theta}\|^2 / \|\theta\|$, where J is a constant that exists since the action set \mathcal{A} is an ellipsoid. Since $\|\theta\| \geq \theta_{\min}$, it follows that

$$\mathbb{E} \left[\max_{x \in \mathcal{A}} x^T \theta - \max_{x \in \mathcal{A}} x^T \hat{\theta} \right] \leq J \frac{\mathbb{E} \|\theta - \hat{\theta}\|^2}{\theta_{\min}} \leq J \frac{d^2}{N_1 \theta_{\min}}. \quad (17)$$

Further, at the exploration phase it holds with $g(\theta) := \arg \max_{x \in \mathcal{A}} x^T \theta$ that

$$g^T(\theta)\theta - x_t^T \theta \leq \max_{x \in \mathcal{A}} x^T \theta - \max_{x \in \mathcal{A}} x^T (-\theta) \leq 2J\theta_{\max}.$$

Therefore, the total regret in N steps satisfies

$$\begin{aligned} \mathbb{E}[R_N] &\leq 2J\theta_{\max}N_1 + (N - N_1)J \frac{d^2}{N_1\theta_{\min}} \\ &= O\left(N_1 + \frac{N}{N_1}d^2\right), \end{aligned}$$

which completes the proof. \blacksquare

B. PROOF OF LEMMA 2

Proof: Without loss of generality, we assume that N_2 is a multiple of r . Recall that, at the end of the exploration phase, $\hat{\alpha}$ is computed by $\hat{\alpha} = (\hat{B}^T X_T X_T^T \hat{B})^{-1} \hat{B}^T X_T Y_T$. Since x_t repeatedly takes actions from a'_1, \dots, a'_r , it holds that $X_T X_T^T = N_2 A A^T / r$ with $A = [a'_1, \dots, a'_r]$. Therefore, we have $\hat{\alpha} = (N_2 \hat{B}^T A A^T \hat{B} / r)^{-1} \hat{B}^T X_T Y_T$. Since $a'_i = \lambda_0 [\hat{B}]_i$, it holds that $A = \lambda_0 \hat{B}$, which implies that $(N_2 \hat{B}^T A A^T \hat{B} / r)^{-1} = \frac{r}{\lambda_0^2 N_2} I_r$. Consequently,

$$\hat{\alpha} = \frac{r}{\lambda_0^2 N_2} \hat{B}^T X_T Y_T.$$

As $Y_T = X_T^\top B\alpha + \eta$ with $\eta = [\eta_1, \dots, \eta_{N_2}]^\top$, we have

$$\begin{aligned}\hat{\alpha} &= \frac{r}{\lambda_0^2 N_2} \hat{B}^\top X_T (X_T^\top B\alpha + \eta) \\ &= \frac{r}{\lambda_0^2 N_2} \frac{N_2}{r} \hat{B}^\top A A^\top B\alpha + \frac{r}{\lambda_0^2 N_2} \hat{B}^\top X_T \eta \\ &= \hat{B}^\top B\alpha + \frac{r}{\lambda_0^2 N_2} \hat{B}^\top X_T \eta.\end{aligned}$$

As $\hat{\theta} = \hat{B}\hat{\alpha}$ and $\theta = B\alpha$, it follows that

$$\hat{\theta} - \theta = \hat{B}\hat{\alpha} - B\alpha = \underbrace{\hat{B}\hat{B}^\top B\alpha - B\alpha}_{s_1} + \underbrace{\frac{r}{\lambda_0^2 N_2} \hat{B}\hat{B}^\top X_T \eta}_{s_2}.$$

Next, we evaluate $\mathbb{E}\|\hat{\theta} - \theta\|^2$. Since $\mathbb{E}s_1^\top s_2 = 0$, it holds that $\mathbb{E}\|\hat{\theta} - \theta\|^2 \leq \mathbb{E}\|s_1\|^2 + \mathbb{E}\|s_2\|^2$.

Observe that $I_d = \hat{B}\hat{B}^\top + \hat{B}_\perp \hat{B}_\perp^\top$. Thus, we have

$$\mathbb{E}\|s_1\|^2 = \|(I_d - \hat{B}_\perp \hat{B}_\perp^\top)B\alpha - B\alpha\|^2 = \|\hat{B}_\perp \hat{B}_\perp^\top B\alpha\|^2.$$

Because $\|\hat{B}_\perp^\top B\|_F \leq \varepsilon$, we have

$$\mathbb{E}\|s_1\|^2 \leq \|\hat{B}_\perp\|^2 \cdot \|\hat{B}_\perp^\top B\|_F^2 \cdot \|\alpha\|^2 \leq \mu \varepsilon^2, \quad (18)$$

where $\mu > 0$, which is such that $\|\alpha\|^2 \leq \mu$, exists since $\theta = B\alpha$ satisfies $\theta_{\min} \leq \theta \leq \theta_{\max}$.

Now, we evaluate $\mathbb{E}\|s_2\|^2$, which satisfies

$$\begin{aligned}\mathbb{E}\|s_2\|^2 &= \frac{r^2}{\lambda_0^4 N_2^2} \mathbb{E}(\hat{B}\hat{B}^\top X_T \eta)^\top \hat{B}\hat{B}^\top X_T \eta \\ &= \frac{r^2}{\lambda_0^4 N_2^2} \sum_{t=1}^{N_2} x_t^\top \hat{B}\hat{B}^\top x_t \mathbb{E}\eta_t^2.\end{aligned}$$

Since η_t is sub-Gaussian with variance proxy variable 1, we have

$$\mathbb{E}\|s_2\|^2 \leq \frac{r^2}{\lambda_0^4 N_2^2} \sum_{t=1}^{N_2} x_t^\top \hat{B}\hat{B}^\top x_t = \frac{r^2}{\lambda_0^2 N_2}. \quad (19)$$

Putting (18) and (19) together, we have $\mathbb{E}[\|\hat{\theta} - \theta\|^2] \leq \frac{r^2}{\lambda_0^2 N_2} + \mu \varepsilon^2$. Similar to (17), one can derive that

$$\mathbb{E}[\max_{x \in \mathcal{A}} x^\top \theta - \max_{x \in \mathcal{A}} x^\top \hat{\theta}] \leq J \frac{r^2}{\theta_{\min} \lambda_0^2 N_2} + J \frac{1}{\theta_{\min}} \mu \varepsilon^2. \quad (20)$$

For the commitment phase, there are $N - N_2$ steps. Thus, the overall regret satisfies

$$\mathbb{E}[R_N] \leq 2J\theta_{\max} N_2 + (N - N_2) \mathbb{E}\left(\max_{x \in \mathcal{A}} x^\top \theta - \max_{x \in \mathcal{A}} x^\top \hat{\theta}\right).$$

Substituting (20) into the right side yields $\mathbb{E}[R_N] = O(N_2 + \frac{N}{N_2} r^2 + N \varepsilon^2)$, which completes the proof. \blacksquare

C. PROOF OF THEOREM 2

Lemma 6 (Matrix Bernstein's inequality [43]): Let X_1, X_2, \dots, X_k be independent zero-mean $d \times d$ symmetric

random matrices so that there exists $M > 0$ such that $\|X_i\| \leq M$ almost surely for all $i = 1, 2, \dots, k$. Then, for any $t \geq 0$, it holds that $\Pr[\|\sum_{i=1}^k X_i\| \geq t] \leq 2d \exp(\frac{-2t^2}{\sigma^2 + Mt/3})$, where $\sigma^2 = \|\sum_{i=1}^k \mathbb{E}X_i\|$.

Proof: Denote the nL tasks that are played using RepE as $\theta_1, \theta_2, \dots, \theta_{nL}$, and let $k := nL$. Then, \hat{W} becomes $\frac{1}{k} \sum_{i=1}^k \hat{\theta}_i \hat{\theta}_i^\top$. Let $W = \frac{1}{k} \sum_{i=1}^k \theta_i \theta_i^\top$ be the true counterpart of \hat{W} . The proof is constructed in two steps. In Step 1, we use Lemma 6 to estimate $\|\hat{W} - (W + D)\|$ with D being a scaled identity matrix; in Step 2, we use the Davis-Kahan sin θ Theorem [46] to evaluate the distance between the top- r singular values of \hat{W} and $W + D$, which is the distance between \hat{B} and the true B (notice that $W + D$ and W share the same singular vectors).

Step 1: Let $\eta := [\eta_1, \eta_2, \dots, \eta_{N_1}]^\top$, and it holds that $Y_E = X_E^\top \theta_i + \eta$. It follows that

$$\hat{\theta}_i = (X_E X_E^\top)^{-1} X_E (X_E^\top \theta_i + \eta) = \theta_i + (X_E X_E^\top)^{-1} X_E \eta.$$

Some algebraic computations yield

$$\begin{aligned}\hat{\theta}_i \hat{\theta}_i^\top &= \theta_i \theta_i^\top + \theta_i \eta^\top X_E^\top (X_E X_E^\top)^{-1} + (X_E X_E^\top)^{-1} X_E \eta \theta_i^\top \\ &\quad + (X_E X_E^\top)^{-1} X_E \eta \eta^\top X_E^\top (X_E X_E^\top)^{-1}.\end{aligned}$$

Since η_i are independent zero mean 1-sub-Gaussian random variables, the expectation of $\hat{\theta}_i \hat{\theta}_i^\top$ can be computed as

$$\begin{aligned}\mathbb{E}\hat{\theta}_i \hat{\theta}_i^\top &= \theta_i \theta_i^\top + (X_E X_E^\top)^{-1} X_E \mathbb{E}\eta \eta^\top X_E^\top (X_E X_E^\top)^{-1} \\ &= \theta_i \theta_i^\top + (X_E X_E^\top)^{-1}.\end{aligned}$$

Since $[a_1, a_2, \dots, a_d]$ be the standard basis of \mathbb{R}^d , it holds that $\sum_{i=1}^d a_i a_i^\top = I_d$. Without loss of generality, we consider N_1 as a multiple of d , then it follows that $X_E X_E^\top = \frac{N_1}{d} \lambda_0^2 I_d$. Therefore, we have $(X_E X_E^\top)^{-1} = \frac{d}{\lambda_0^2 N_1} I_d$. Denote $D := \frac{d}{\lambda_0^2 N_1} I_d$, then it follows that

$$\mathbb{E}\hat{\theta}_i \hat{\theta}_i^\top = \theta_i \theta_i^\top + D,$$

and

$$\hat{\theta}_i \hat{\theta}_i^\top = \theta_i \theta_i^\top + D \underbrace{(\theta_i \eta^\top X_E^\top + X_E \eta \theta_i^\top)}_A + D^2 \underbrace{(X_E \eta \eta^\top X_E^\top)}_C. \quad (21)$$

Define a set of new variables $z_i = \frac{1}{k} \hat{\theta}_i \hat{\theta}_i^\top - \frac{1}{k} (\theta_i \theta_i^\top + D)$. From (21), we have $z_i = \frac{1}{k} (DA + D^2 C - D)$. Then, the expectation of z_i^2 satisfies

$$\begin{aligned}\mathbb{E}z_i^2 &= \frac{1}{k^2} \mathbb{E}[D^2 A^2 + D^4 C^2 + D^2 + D^3 (AC + CA) \\ &\quad - 2D^2 A - 2D^3 C],\end{aligned} \quad (22)$$

where the fact that D commutes with any matrix has been used. Since D is deterministic, to compute $\mathbb{E}z_i^2$, it suffices to calculate $\mathbb{E}A^2, \mathbb{E}C^2, \mathbb{E}(AC + CA), \mathbb{E}A$, and $\mathbb{E}C$.

For $\mathbb{E}A^2$, it holds that

$$\begin{aligned}\mathbb{E}A^2 &= \mathbb{E}(\theta_i \eta^\top X_E^\top)^2 + \mathbb{E}(X_E \eta \theta_i^\top)^2 + \mathbb{E}(\theta_i \eta^\top X_E^\top X_E \eta \theta_i^\top) \\ &\quad + \mathbb{E}(X_E \eta \theta_i^\top \theta_i \eta^\top X_E^\top) \\ &= 2 \frac{\lambda_0^2 N_1}{d} \theta_i \theta_i^\top + N_1 \lambda_0^2 \theta_i \theta_i^\top + \theta_i^\top \theta_i \frac{\lambda_0^2 N_1}{d} I_d.\end{aligned}$$

For $\mathbb{E}C^2$, we have

$$\begin{aligned}\mathbb{E}C^2 &= \mathbb{E}[X_E \eta \eta^\top X_E^\top X_E \eta \eta^\top X_E^\top] \\ &= \mathbb{E}[\eta^\top X_E^\top X_E \eta \cdot X_E \eta \eta^\top X_E^\top] = \mathbb{E}\left[\sum_{i=1}^{N_1} \eta_i^4 x_i^\top x_i x_i x_i^\top\right] \\ &= \psi_4 \lambda_0^2 \left(\sum_{i=1}^{N_1} x_i x_i^\top\right) = \psi_4 \lambda_0^2 X_E X_E^\top = \frac{\psi_4 \lambda_0^4 N_1}{d} I_d,\end{aligned}$$

where $\psi_4 = \mathbb{E}\eta_i^4$ (ψ_4 always exists since η_i is a sub-Gaussian random variable). For $\mathbb{E}(AC + CA)$, it holds that

$$\begin{aligned}\mathbb{E}(AC + CA) &= \mathbb{E}[(\theta_i \eta^\top X_E^\top + X_E \eta \theta_i^\top) X_E \eta \eta^\top X_E^\top \\ &\quad + X_E \eta \eta^\top X_E^\top (\theta_i \eta^\top X_E^\top + X_E \eta \theta_i^\top)] \\ &= \mathbb{E}[\eta^\top X_E^\top X_E \eta \cdot \theta_i \eta^\top X_E^\top] \\ &\quad + \mathbb{E}[\theta_i^\top X_E \eta \cdot X_E \eta \eta^\top X_E^\top] \\ &\quad + \mathbb{E}[\eta^\top X_E^\top \theta_i \cdot X_E \eta \eta^\top X_E^\top] \\ &\quad + \mathbb{E}[\eta^\top X_E^\top X_E \eta \cdot X_E \eta \theta_i^\top] \\ &= \frac{\lambda_0^3 \psi_3 N_1}{d} \theta_i \mathbf{1}_d^\top + \frac{\lambda_0^3 \psi_3 N_1}{d} \text{diag} \theta_i \\ &\quad + \frac{\lambda_0^3 \psi_3 N_1}{d} \text{diag} \theta_i + \frac{\lambda_0^3 \psi_3 N_1}{d} \mathbf{1}_d \theta_i^\top \\ &= \frac{\lambda_0^3 \psi_3 N_1}{d} (\theta_i \mathbf{1}_d^\top + \mathbf{1}_d \theta_i^\top) + \frac{2\lambda_0^3 \psi_3 N_1}{d} \text{diag} \theta_i.\end{aligned}$$

Notice that $\mathbb{E}A = 0$. For $\mathbb{E}C$, it holds that $\mathbb{E}C = \mathbb{E}X \eta \eta^\top X^\top = \frac{\lambda_0^2 N_1}{d} I_d$.

Overall, substituting all the above terms into Eq. (22) we have

$$\begin{aligned}\mathbb{E}z_i^2 &= \frac{1}{k^2} \frac{d^2}{\lambda_0^4 N_1^2} \left[2 \frac{\lambda_0^2 N_1}{d} \theta_i \theta_i^\top + N_1 \lambda_0^2 \theta_i \theta_i^\top \right. \\ &\quad \left. + \theta_i^\top \theta_i \frac{\lambda_0^2 N_1}{d} I_d + D^2 \frac{\psi_4 \lambda_0^4 N_1}{d} I_d + I_d + D \left(\frac{\lambda_0^3 \psi_3 N}{d} \right. \right. \\ &\quad \left. \left. \cdot (\theta_i \mathbf{1}_d^\top + \mathbf{1}_d \theta_i^\top) + \frac{2\lambda_0^3 \psi_3 N}{d} \text{diag} \theta_i \right) - 2D \frac{\lambda_0^2 N}{d} I_d \right] \\ &\leq \frac{d^2}{k^2 \lambda_0^2 N_1} \left(\frac{2}{d} + 1 \right) \theta_i \theta_i^\top + O\left(\frac{d^2}{k^2 \lambda_0^4 N_1^2} I_d \right).\end{aligned}$$

Let $\sigma^2 = \|\sum_{i=1}^k \mathbb{E}z_i^2\|_F$, and it satisfies

$$\sigma^2 \lesssim \left\| \frac{d^2}{k \lambda_0^2 N_1} \frac{1}{k} \sum_{i=1}^k \theta_i \theta_i^\top \right\|_F \leq O\left(\frac{d^2}{k \lambda_0^2 N_1} \text{Trace}(W_k) \right).$$

Since for any $\theta \in \mathcal{T}$, it holds that $\theta_{\min} \leq \|\theta\| \leq \theta_{\max}$ with $\theta_{\min} = \Theta(1)$ and $\theta_{\max} = \Theta(1)$, it follows that $\text{Trace}(W_k) = \Theta(1)$. Therefore, it holds that $\sigma^2 \leq O\left(\frac{d^2}{k \lambda_0^2 N_1}\right)$.

Applying Lemma 6 with $t = 2c_1 \log(2dkN_1) + c_2 \sqrt{4\sigma^2 \log(2dkN_1)}$ for sufficiently large $c_1, c_2 > 0$, we have

$$\left\| \sum_{i=1}^k z_i \right\|_F \lesssim \frac{d}{\lambda_0} \sqrt{\frac{1}{kN_1}} \left(\sqrt{\log(kdN_1)} + \log(kdN_1) \right),$$

with probability at least $1 - \frac{1}{kN_1}$.

Notice that $\sum_{i=1}^k z_i = \hat{W} - (W + D)$. Let $W' = W + D$, we have

$$\begin{aligned}\|\hat{W} - W'\|_F &:= \|\Delta\|_F \\ &\lesssim \frac{d}{\lambda_0} \sqrt{\frac{1}{kN_1}} \left(\sqrt{\log(kdN_1)} + \log(kdN_1) \right).\end{aligned}$$

Step 2: From the Davis-Kahan sin θ Theorem, we have

$$\|\hat{B}^\top B_\perp\|_F \leq \frac{\|\hat{B}_\perp^\top (W - W') B\|}{\omega} \leq \frac{\|\hat{W} - W'\|_F}{\omega}, \quad (23)$$

where $\omega = \inf_{1 \leq i \leq r, r < j \leq d} |\lambda_i(W') - \lambda_j(\hat{W})|$. From the Weyl's Theorem, $|\lambda_i(W') - \lambda_i(\hat{W})| \leq \|\hat{W} - W'\|_F = \|\sum_{j=1}^k z_j\|_F$ for any $i = 1, \dots, d$. Since $\lambda_i(W') = 0$ for all $i \geq r + 1$, it holds that $|\lambda_i(\hat{W})| \leq \|\Delta\|_F$ for all $i \geq r + 1$. Recall that σ_r is the r -th largest eigenvalue of W , therefore $\omega \geq \sigma_k - \|\Delta\|_F$. From the Assumption 2, we know $\sigma_r \geq \nu/r$, therefore we obtain

$$\begin{aligned}\|\hat{B}^\top B_\perp\|_F &\lesssim \frac{\|\Delta\|_F}{\sigma_r - \|\Delta\|_F} \\ &\lesssim \frac{dr}{\lambda_0 \nu} \sqrt{\frac{1}{nLN_1}} \left(\sqrt{\log dnLN_1} + \log dnLN_1 \right),\end{aligned}$$

where $k = nL$ have been used. The proof is complete. \blacksquare

D. ANALYSIS OF LEMMAS 3 AND 4

Let us present some instrumental results first.

Lemma 7 (Random projection, Chap. 5, [43]): Let P be a projection from \mathbb{R}^n onto a random m -dimensional subspace uniformly distributed in the Grassmann manifold $G_{n,m}$. Let $x \in \mathbb{R}^n$ be a fixed point and $\beta > 0$. Then, with probability at least $1 - \exp(-c\beta^2 m)$, we have

$$(1 - \beta) \sqrt{\frac{m}{n}} \|x\|_2 \leq \|Px\|_2 \leq (1 + \beta) \sqrt{\frac{m}{n}} \|x\|_2.$$

Lemma 8: Denote $z = B_\perp^\top \theta$. Let P be a projection matrix from \mathbb{R}^{d-r} onto a random m -dimensional subspace uniformly distributed in the Grassmann manifold $G_{(d-r),m}$.

Then, it holds with probability at least $1 - \exp(-c\beta^2m)$ that $\frac{1}{2}\sqrt{\frac{m}{d-r}}\|z\|_2 \leq \|Pz\|_2 \leq \frac{3}{2}\sqrt{\frac{m}{d-r}}\|z\|_2$.

The proof of Lemma 8 directly follows from Lemma 7 by letting $\beta = 1/2$.

Lemma 9 (Concentration of the norm, Chap. 3, [43]): Suppose that $X = [X_1, X_2, \dots, X_n]^\top$ is a random vector, where X_1, \dots, X_n are independent δ -sub-Gaussian random variable. Then, for any $\xi > 0$ it holds that $\Pr[\frac{1}{\sqrt{n}}\|X\|_2 - \delta] \geq \xi] \leq 2\exp(-\frac{cn\xi^2}{K^2})$, where c is an absolute constant and $K = \max_i \|X_i\|_{\psi_2}$ is assumed $K < 1$.

We are now ready to prove Lemmas 3 and 4.

Proof of Lemma 3: We construct the proof by showing that Y_{det} goes beyond C_{det} with high probability when the task θ is played by RepCD.

Denote $\rho = \|B_\perp^\top \theta\|$, and from Lemma 8 we have $\|Q^\top B_\perp^\top \theta\| = \|QQ^\top B_\perp^\top \theta\| \geq \frac{1}{2}\rho\sqrt{\bar{n}/(d-r)}$ with probability at least $1 - \exp(-c\bar{n}/4)$ since $QQ^\top B_\perp^\top \theta$ can be taken as projecting $B_\perp^\top \theta$ onto the random subspace spanned by Q . The reward vector Y_{det} satisfies $Y_{\text{det}} = \lambda_0 G^\top B_\perp^\top \theta + \eta$ with $G := \underbrace{[I_{d-r}, \dots, I_{d-r}, Q]}_k$. It follows that

$$\begin{aligned} & \Pr[Y_{\text{det}} \in C_{\text{det}}] \\ & \leq \Pr\left[\left|\frac{1}{\sqrt{n_{\text{det}}}}\|\lambda_0 G^\top B_\perp^\top \theta + \eta\| - 1\right| \leq \xi_{\text{det}}\right] \\ & \leq \Pr\left[\left|\frac{1}{\sqrt{n_{\text{det}}}}\|\eta\| - 1\right| \geq \frac{\lambda_0 \rho}{2\sqrt{d-r}}\sqrt{1 + \frac{3k(d-r)}{n_{\text{det}}}} - \xi_{\text{det}}\right] \\ & \leq \Pr\left[\left|\frac{1}{\sqrt{n_{\text{det}}}}\|\eta\| - 1\right| \geq \frac{\lambda_0 \rho}{2\sqrt{d-r}} - \xi_{\text{det}}\right]. \end{aligned}$$

Observe that $\|B_\perp^\top \theta\| = \|B_\perp^\top \bar{B}\alpha\| \geq \sigma_{\min}(B_\perp^\top \bar{B})\|\alpha\|$. Since $\sigma_{\min}(B_\perp^\top \bar{B}) = \sin \theta_r(B, \bar{B})$ and $\theta \geq \theta_{\min}$, we have

$$\|B_\perp^\top \theta\| \geq \sin \theta_r(B, \bar{B})\theta_{\min} = \kappa_1 \theta_{\min}.$$

Therefore, we have $\rho \geq \kappa_1 \theta_{\min}$. From lemma 9, one can derive that

$$\Pr[Y_{\text{det}} \in C_{n_{\text{det}}}] \leq 2\exp\left(-\frac{cn_{\text{det}}(\frac{1}{2}\kappa_1\lambda_0\theta_{\min}\sqrt{\frac{1}{d-r}} - \xi_{\text{det}})^2}{K^2}\right).$$

Let $n_{\text{net}} = \frac{9(d-r)\log(2S^2N)}{\kappa_1^2\theta_{\min}^2\lambda_0^2}$ and $\xi_{\text{det}} = \sqrt{\frac{\log(2S^2N)}{4n_{\text{det}}}}$. Then, it can be calculated that $\Pr[Y_{\text{det}} \in C_{\text{det}}] \leq O(\frac{1}{S^2N})$, which means that the outlier θ to B can be detected with probability at least $1 - O(\frac{1}{S^2N})$. The proof is complete. \blacksquare

For Lemma 3, the only difference is that only an estimate \hat{B} satisfying $\|\hat{B}_\perp^\top B\| \leq \varepsilon$ is known. It can be derived that $\|\hat{B}_\perp^\top \theta\| = \|B_\perp^\top \bar{B}\alpha\| \geq (\kappa_1 - \varepsilon)\theta_{\min}$. Then, following similar steps as those for Lemma 3, one can prove lemma 4.

E. PROOF OF LEMMA 5

Proof: It follows from the proof of Theorem 2 that, after the initial bL tasks, the estimated representation \hat{B} satisfies

$$\|\hat{B}^\top B_\perp\|_F \lesssim \frac{dr}{\lambda_0\nu\sqrt{bLN_1}}\left(\sqrt{\log(dblN_1)} + \log(dblN_1)\right).$$

For the simplicity of notation, let $q = \sqrt{\log(dblN_1)} + \log(dblN_1)$. For θ , there is $\alpha \in \mathbb{R}^r$ such that $\theta = B\alpha$, which implies that $Y_{\text{det}} = \lambda_0 G^\top \hat{B}_\perp^\top \theta + \eta = \lambda_0 G^\top \hat{B}_\perp^\top B\alpha + \eta$, where $G = [I_{d-r}, \dots, I_{d-r}, Q]$. Since $\|Q^\top \hat{B}_\perp^\top \theta\| = \|QQ^\top \hat{B}_\perp^\top \theta\|$, it follows from Lemma 8 that $\|Q^\top \hat{B}_\perp^\top \theta\| \leq \frac{3}{2}\|\hat{B}_\perp^\top \theta\|\sqrt{\bar{n}/(d-r)}$ with probability at least $1 - \exp(-c\bar{n}/4)$. Denote $\varepsilon = \|\hat{B}_\perp^\top B\|_F$, and it can be observed that $\|\hat{B}_\perp^\top \theta\| \leq \|\hat{B}_\perp^\top B\|_F \cdot \|\alpha\| \leq \varepsilon\theta_{\max}$. Subsequently, since $n_{\text{det}} = k(d-r) + \bar{n}$, it holds that

$$\begin{aligned} \frac{1}{\sqrt{n_{\text{det}}}}\|Y_{\text{det}}\|_2 & \leq \frac{1}{\sqrt{n_{\text{det}}}}\left(\frac{\lambda_0\varepsilon\theta_{\max}}{\sqrt{d-r}}\sqrt{k(d-r) + \frac{9}{4}\bar{n}} + \|\eta\|\right) \\ & \leq \frac{3\lambda_0\varepsilon\theta_{\max}}{2\sqrt{d-r}} + \frac{1}{\sqrt{n_{\text{det}}}}\|\eta\| \leq \frac{1}{\sqrt{n_{\text{det}}}}\|\eta\| + \underbrace{\frac{3dr\theta_{\max}q}{2\nu\sqrt{bLN_1}(d-r)}}_s. \end{aligned}$$

Then, from Lemma 9, we have

$$\Pr[Y_{\text{det}} \notin C_{\text{det}}|\theta = B\alpha] \leq 2\exp\left(-\frac{cn_{\text{det}}(\xi_{\text{det}} - s)^2}{K^2}\right).$$

Substituting (9) and (8) into the right-hand side yields $\Pr[\theta \notin \text{Span}(\hat{B})|\theta = B\alpha] \leq \frac{1}{S^2N}$ given $N_1 = dr\sqrt{N/\ell}$. \blacksquare

REFERENCES

- [1] N. T. Franklin and M. J. Frank, "Generalizing to generalize: Humans flexibly switch between compositional and conjunctive structures during reinforcement learning," *PLoS Comput. Biol.*, vol. 16, no. 4, 2020, Art. no. e1007720.
- [2] A. Radulescu, Y. S. Shin, and Y. Niv, "Human representation learning," *Annu. Rev. Neurosci.*, vol. 44, no. 1, pp. 253–273, 2021.
- [3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [4] J. D. Lee, Q. Lei, N. Saunshi, and J. ZHUO, "Predicting what you already know helps: Provable self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 309–323.
- [5] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," 2015, *arXiv:1502.02072*.
- [6] C. D'Eramo, D. Tateo, A. Bonarini, M. Restelli, and J. Peters, "Sharing knowledge in multi-task deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–18.
- [7] M.-F. Balcan, M. Khodak, and A. Talwalkar, "Provable guarantees for gradient-based meta-learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 424–433.
- [8] S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei, "Few-shot learning via learning the representation, provably," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–29.
- [9] N. Tripuraneni, C. Jin, and M. Jordan, "Provable meta-learning of linear representations," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10434–10443.
- [10] N. Tripuraneni, M. Jordan, and C. Jin, "On the theory of transfer learning: The importance of task diversity," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 7852–7862.

- [11] Q. Bouniot, I. Redko, R. Audigier, A. Loesch, Y. Zotkin, and A. Habrard, "Towards better understanding meta-learning methods through multi-task representation learning theory," 2020, *arXiv:2010.01992*.
- [12] A. Borkowska, W. Drożdż, P. Jurkowski, and J. K. Rybakowski, "The wisconsin card sorting test and the n-back test in mild cognitive impairment and elderly depression," *World J. Biol. Psychiatry*, vol. 10, no. 4/3, pp. 870–876, 2009.
- [13] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and non-stochastic multi-armed bandit problems," *Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.
- [14] L. Wei and V. Srivastava, "Nonstationary stochastic multiarmed bandits: UCB policies and minimax regret," 2021, *arXiv:2101.08980*.
- [15] T. Gafni and K. Cohen, "Learning in restless multiarmed bandits via adaptive arm sequencing rules," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 5029–5036, Oct. 2021.
- [16] P. Reverdy, V. Srivastava, and N. E. Leonard, "Satisficing in multi-armed bandit problems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3788–3803, Aug. 2017.
- [17] L. Wei and V. Srivastava, "Minimax policy for heavy-tailed bandits," *IEEE Contr. Syst. Lett.*, vol. 5, no. 4, pp. 1423–1428, Oct. 2021.
- [18] M. Malekipirbazari and Z. Çavuş, "Risk-averse allocation indices for multiarmed bandit problem," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5522–5529, Nov. 2021.
- [19] M. K. Hanawal and S. J. Darak, "Multiplayer bandits: A trekking approach," *IEEE Trans. Autom. Control*, vol. 67, no. 5, pp. 2237–2252, May 2022.
- [20] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multiplayer multiarmed bandits," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2331–2345, Apr. 2014.
- [21] P. Landgren, V. Srivastava, and N. E. Leonard, "Social imitation in cooperative multiarmed bandits: Partition-based algorithms with strictly local information," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 5239–5244.
- [22] D. Martínez-Rubio, V. Kanade, and P. Rebeschini, "Decentralized cooperative stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 4529–4540.
- [23] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision making in multi-agent multi-armed bandits," *Automatica*, vol. 125, 2021, Art. no. 109445.
- [24] U. Madhushani and N. E. Leonard, "A dynamic observation strategy for multi-agent multi-armed bandit problem," in *Proc. Eur. Control Conf.*, 2020, pp. 1677–1682.
- [25] J. Zhu and J. Liu, "A distributed algorithm for multi-armed bandit with homogeneous rewards over directed graphs," in *Proc. Amer. Control Conf.*, 2021, pp. 3038–3043.
- [26] U. Madhushani and N. Leonard, "When to call your neighbor? Strategic communication in cooperative stochastic bandits," 2021, *arXiv:2110.04396*.
- [27] U. Madhushani, A. Dubey, N. Leonard, and A. Pentland, "One more step towards reality: Cooperative bandits with imperfect communication," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 7813–7824.
- [28] M. Dimakopoulou, Z. Ren, and Z. Zhou, "Online multi-armed bandits with adaptive inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 1939–1951.
- [29] R. Zhan, Z. Ren, S. Athey, and Z. Zhou, "Policy learning with adaptively collected data," 2021, *arXiv:2105.02344*.
- [30] M. G. Azar, A. Lazaric, and E. Brunskill, "Sequential transfer in multi-armed bandit with finite set of models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2220–2228.
- [31] L. Cella, A. Lazaric, and M. Pontil, "Meta-learning with stochastic linear bandits," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1360–1370.
- [32] M. Soare, O. Alsharif, A. Lazaric, and J. Pineau, "Multi-task linear bandits," in *Proc. Workshop Transfer Multi-task Learn.: Theory Meets Pract.*, 2014, pp. 106.
- [33] A. A. Deshmukh, U. Dogan, and C. Scott, "Multi-task learning for contextual bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 4851–4859.
- [34] J. Yang, W. Hu, J. D. Lee, and S. S. Du, "Impact of representation learning in linear bandits," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–25.
- [35] J. Hu, X. Chen, C. Jin, L. Li, and L. Wang, "Near-optimal representation learning for linear bandits and linear RL," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4349–4358.
- [36] L. Cella, K. Lounici, and M. Pontil, "Multi-task representation learning with stochastic linear bandits," 2022, *arXiv:2202.10066*.
- [37] Y. Qin, T. Menara, S. Oymak, S. Ching, and F. Pasqualetti, "Representation learning for context-dependent decision-making," in *Proc. Amer. Control Conf.*, Atlanta, GA, USA, 2022, pp. 2130–2135.
- [38] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. III," *SIAM J. Numer. Anal.*, vol. 7, no. 1, pp. 1–46, 1970.
- [39] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Math. Operations Res.*, vol. 35, no. 2, pp. 395–411, 2010.
- [40] Y. Li, Y. Wang, X. Chen, and Y. Zhou, "Tight regret bounds for infinite-armed linear contextual bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 370–378.
- [41] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proc. Conf. Learn. Theory*, 2008, pp. 355–367.
- [42] Y. Abbasi-Yadkori, A. Antos, and C. Szepesvári, "Forced-exploration based algorithms for playing in stochastic linear bandits," in *Proc. COLT Workshop On-line Learn. Limited Feedback*, 2009, vol. 92, pp. 236–241.
- [43] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*. Cambridge, U.K.: Cambridge Univ. Press, 2018.
- [44] D. A. Grant and E. Berg, "A behavioral analysis of degree of reinforcement and ease of shifting to new responses in a Weigl-type card-sorting problem," *J. Exp. Psychol.*, vol. 38, no. 4, pp. 404–411, 1948.
- [45] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–17.
- [46] R. Bhatia, *Matrix Analysis*. New York, NY, USA: Springer-Verlag, 2013.



tems, network neuroscience, and reinforcement learning theory.

YUZHEN QIN (Member, IEEE) received the B.E. degree in control engineering from Hohai University, Nanjing, China, in 2012, the M.E. degree in control engineering from Wuhan University, Wuhan, China, in 2015, and the Ph.D. degree in systems and control from the Engineering and Technology Institute, University of Groningen, Groningen, The Netherlands, in 2019. He is currently a Postdoctoral Researcher with the University of California, Riverside, CA, USA. His main research interests include nonlinear control systems, network neuroscience, and reinforcement learning theory.



TOMMASO MENARA (Member, IEEE) received the Laurea degree (equivalent to B.Sc.) in mechatronics engineering from the University of Padova, Italy, in 2013, the Laurea Magistrale degree (equivalent to M.Sc.) in robotics and automation engineering from the University of Pisa, Pisa, Italy, in 2016, and the Ph.D. degree in mechanical engineering from the University of California, Riverside, CA, USA, in 2021. He is currently a Postdoctoral Scholar with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA. His research interests are in the intersection of control theory, neuroscience, and machine learning. Dr. Menara was the recipient of the IEEE CSS 2020 Roberto Tempo Award and the Best Student Paper Award at the 2019 American Control Conference.



SAMET OYMAK (Member, IEEE) received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 2009, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2011 and 2015, respectively. He was a Postdoctoral Scholar with the University of California, Berkeley, Berkeley, CA, USA, from 2014 to 2015. He is currently an Assistant Professor of electrical and computer engineering with the University of California, Riverside, CA, USA.



SHINUNG CHING (Member, IEEE) received the B.Eng. (Hons.) degree from McGill University, Montreal, QC, Canada, the M.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA. He is currently an Associate Professor with the Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA. He was subsequently a Postdoctoral Associate with the Harvard Medical School. He is author of more than 90 refereed papers and the textbook *Quasilinear Control*. His research interests are in the intersection of systems and control theory, theoretical neuroscience, and neural engineering.



FABIO PASQUALETTI (Member, IEEE) received the Laurea degree (equivalent to B.Sc.) in computer engineering and the Laurea Magistrale degree (equivalent to M.Sc.) in automation engineering from the University of Pisa, Italy, in 2004 and 2007, respectively, and the Doctor of Philosophy degree in mechanical engineering from the University of California, Santa Barbara, CA, USA, in 2012. He is currently a Professor of mechanical engineering with the University of California, Riverside, CA, USA. His main research interests

include the analysis and control of network systems, security of cyber-physical systems, and network neuroscience. He has received several awards, including a Young Investigator Research Program Award from AFOSR in 2020, a Young Investigator Program Award from ARO in 2017, and the 2016 TCNS Outstanding Paper Award from IEEE CSS. He is a Member of SIAM.