Pure Exploration in Kernel and Neural Bandits

Yinglun Zhu[*]
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706
yinglun@cs.wisc.edu

Dongruo Zhou[*]
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
drzhou@cs.ucla.edu

Ruoxi Jiang[*]
Department of Computer Science
University of Chicago
Chicago, IL 60637
roxie62@uchicago.edu

Quanquan Gu
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
qgu@cs.ucla.edu

Rebecca Willett
Department of Statistics and Computer Science
University of Chicago
Chicago, IL 60637
willett@uchicago.edu

Robert Nowak
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706
rdnowak@wisc.edu

Abstract

We study pure exploration in bandits, where the dimension of the feature representation can be much larger than the number of arms. To overcome the curse of dimensionality, we propose to adaptively embed the feature representation of each arm into a lower-dimensional space and carefully deal with the induced model misspecifications. Our approach is conceptually very different from existing works that can either only handle low-dimensional linear bandits or passively deal with model misspecifications. We showcase the application of our approach to two pure exploration settings that were previously under-studied: (1) the reward function belongs to a possibly infinite-dimensional Reproducing Kernel Hilbert Space, and (2) the reward function is nonlinear and can be approximated by neural networks. Our main results provide sample complexity guarantees that only depend on the effective dimension of the feature spaces in the kernel or neural representations. Extensive experiments conducted on both synthetic and real-world datasets demonstrate the efficacy of our methods.

1 Introduction

Pure exploration in bandits [11, 12, 6] has been extensively studied in machine learning. Consider a set of arms, where each arm is associated with an unknown reward distribution. The goal is to

[*]Equal contribution

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

approximately identify the optimal arm using as few samples as possible. Applications of bandit pure exploration range from medical domains [3] to online content recommendation [40].

Despite the popularity of bandit pure exploration, it was previously mainly studied in two relatively restrictive settings: (1) the standard multi-armed bandit setting [22, 19, 9, 18], where the expected rewards among arms are completely unrelated to each other, and (2) the (generalized) linear bandit setting [38, 13, 10, 26], where the expected rewards are assumed to be linearly parameterized by some unknown weight vector. The standard multi-armed bandit setting fails to deal with large arm sets, and the linear bandit setting suffers from both model misspecifications (due to its simplified linear form) and the curse of dimensionality in the high-dimensional setting. Pure exploration is also studied in continuous spaces. However, guarantees therein scale exponentially with dimension [31, 4].

In this paper, we generalize bandit pure exploration to the nonlinear and high-dimensional settings. More specifically, we study the following two settings: (1) the rewards of arms are parameterized by a function belonging to a Reproducing Kernel Hilbert Space (RKHS), and (2) the rewards of arms are nonlinear functions that can be approximated by an overparameterized neural network. Problems in these two settings are often high-dimensional in nature. To overcome the curse of dimensionality, we propose to adaptively embed each arm's feature representation in a lower-dimensional space and carefully deal with the induced misspecifications. Note that our approach is conceptually very different from all existing work dealing with model misspecifications: they assume the existence of misspecifications and address it in the original space (thus dealing with model misspecifications in a passive way) [28, 7]. On the other hand, we deliberately induce (acceptable) misspecifications to embed arms into lower-dimensional spaces and thus overcome the curse of dimensionality.

1.1 Contribution and Outline

We make the following main contributions:

- In Section 3, we introduce the idea of adaptive embedding to avoid the curse of dimensionality. The induced model misspecifications are carefully handled, which is novel in the bandit pure exploration setting. The sample complexity is theoretically analyzed and we relate the instance-dependent sample complexity to the complexity of a closely-related linear bandit problem without model misspecification. As a by-product, our algorithm can also be applied to constrained high-dimensional linear bandit pure exploration to reduce sample complexity.
- In Section 4, we specialize the adaptive embedding scheme to pure exploration in an RKHS. We construct feature mappings from eigenfunctions and eigenvalues of the associated kernel. The effective dimension of the kernel is analyzed, and we provide sample complexity guarantees in terms of the eigenvalue decay of the associated kernel. We rely on a *known* kernel in this setting.
- In Section 5, we further extend our adaptive embedding scheme to pure exploration with a general nonlinear reward function and model the reward function with an over-parameterized neural network. Sample complexity guarantees are provided with respect to the eigenvalue decay of the associated Neural Tangent Kernel. To the best of our knowledge, this provides the first theoretically founded pure exploration algorithm with a neural network approximation.
- In Section 6, we conduct extensive experiments on both synthetic and real-world datesets to confirm the efficacy of our proposed algorithms. We conclude our paper in Section 7 with open problems.

1.2 Related Work

The bandit pure exploration problem has a long history, dating back to the seminal work by Bechhofer [5], Paulson et al. [32]. One classical objective of pure exploration is Best Arm Identification (BAI), where the goal is to identify the best arm using as few samples as possible [22, 19, 9, 15]. To make it applicable to a large action space, the BAI problem is also extensively studied as the good arm identification problem, where the goal is to identify an ϵ-optimal arm [11, 12, 20, 21, 34, 23, 30].

The pure exploration problem in linear bandits is initially analyzed in Soare et al. [38], where optimal experimental design [27] is applied to guide the allocation of samples. Other approaches dealing with linear bandits, with various sample complexity guarantees, include adaptive sampling [45] and an approach called track-and-stop [10]. Constrained linear bandit pure exploration is also commonly studied with additional assumptions on the reward parameters [41, 10]. We note that the

track-and-stop approach only achieves optimal instance-dependent sample complexity in the regime where the confidence parameter approaches 0, but fails to do so in the moderate confidence regime. Fiez et al. [13] propose an elimination-based algorithm (with optimal design) that achieves (nearly) instance-dependent sample complexity. such algorithm is further generalized to the combinatorial bandit setting [24].

Learning with model misspecifications was recently introduced in bandit learning, with the primary emphasis placed on the regret minimization problem [16, 28, 14]. A very recent independent work studies pure exploration in kernel bandits with misspecifications [7]; both their and our algorithms follow the framework of RAGE [13] and draw inspiration from [28]. Camilleri et al. [7] propose a robust estimator that works in high-dimensional spaces and also explore the project-then-round idea through *regularized least squares*. Our algorithms adaptively embed actions into lower dimensional spaces according to some error tolerances (different embeddings from round to round); our rounding and elimination steps are thus computed only with respect to lower-dimensional embeddings. We additionally study the pure exploration problem with an overparameterized neural network. As mentioned before, our approach is also conceptually different from existing ones: rather than passively dealing with model misspecifications in its original representation, we deliberately and adaptively embed arms into a lower-dimensional space to avoid the curse of dimensionality; the induced model misspecifications are also carefully dealt with in our algorithms.

2 Problem setting

We introduce the general setting and notations for pure exploration in bandits. Consider a set of arms $\mathcal{X} \subseteq \mathbb{R}^D$ where the number of arms $|\mathcal{X}| = K$ is possibly very large. We use an unknown function $h : \mathcal{X} \to [-1, 1]$ to represent the true reward of each arm. A noisy feedback $h(\boldsymbol{x}) + \xi$ is observed after each sample arm $\boldsymbol{x}$, where the noise ξ is assumed to be 1-sub-Gaussian. The learner is allowed to allocate her samples based on previously collected information, and the goal is to approximately identify an approximately optimal arm using as few samples as possible. Let $\boldsymbol{x}_\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} h(\boldsymbol{x})$ denote the optimal arm among $\mathcal{X}$. We aim at developing (ϵ, δ)-PAC guarantees: for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the algorithm outputs an ϵ-optimal arm $\widehat{\boldsymbol{x}}$ such that $h(\widehat{\boldsymbol{x}}) \geq h(\boldsymbol{x}_\star) - \epsilon$ using a finite number of samples. The performance of the algorithm is measured by its sample complexity, i.e., the number of samples pulled before it stops and recommends a candidate arm.

Notations. We define $\Delta_{\boldsymbol{x}} = h(\boldsymbol{x}_\star) - h(\boldsymbol{x})$ as the sub-optimality gap of arm $\boldsymbol{x}$. We use the notations $\mathcal{S}_k := \{\boldsymbol{x} \in \mathcal{X} : \Delta_{\boldsymbol{x}} < 4 \cdot 2^{-k}\}$ (with $\mathcal{S}_1 = \mathcal{X}$). We consider feature mappings of the form $\boldsymbol{\psi}_d(\cdot) : \mathcal{X} \to \mathbb{R}^d$, and define $\boldsymbol{\psi}_d(\mathcal{X}) = \{\boldsymbol{\psi}_d(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X}\}$. We use $\boldsymbol{\Lambda}_{\mathcal{X}} = \{\lambda \in \mathbb{R}^{|\mathcal{X}|} : \sum_{\boldsymbol{x}\in\mathcal{X}} \lambda_{\boldsymbol{x}} = 1, \lambda_{\boldsymbol{x}} \geq 0\}$ to denote the $(|\mathcal{X}| - 1)$-dimensional probability simplex over arms in $\mathcal{X}$; and set $\boldsymbol{A}_{\boldsymbol{\psi}_d}(\lambda) = \sum_{\boldsymbol{x}\in\mathcal{X}} \lambda_{\boldsymbol{x}} \boldsymbol{\psi}_d(\boldsymbol{x}) \boldsymbol{\psi}_d(\boldsymbol{x})^\top$.[2] We use $\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}$ to represent the Mahalanobis norm. We also define $\mathcal{Y}(\mathcal{V}) = \{\boldsymbol{v} - \boldsymbol{v}' : \boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}\}$ for any set $\mathcal{V}$. For a matrix $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$, we use $\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{x}')$ to denote the entry of $\boldsymbol{H}$ which locates at row $\boldsymbol{x}$ and column $\boldsymbol{x}'$.

3 Bandit pure exploration with adaptive embedding

We introduce the idea of bandit pure exploration with adaptive embedding, which can be viewed as an approach that actively trades off sample complexity with accuracy guarantees: we adaptively embed the feature representation into lower-dimensional spaces to avoid the curse of dimensionality, and conduct pure exploration with *misspecified* linear bandits. The embedding dimensions are carefully selected so that we can identify an ϵ-optimal arm.

We formalize the idea as follows. For any $d \in \mathbb{N}$, we assume the existence of a feature mapping $\boldsymbol{\psi}_d : \mathcal{X} \to \mathbb{R}^d$ and a unknown reward vector $\boldsymbol{\theta}_d \in \mathbb{R}^d$ such that, for any $\boldsymbol{x} \in \mathcal{X}$,

$$h(\boldsymbol{x}) = \langle \boldsymbol{\psi}_d(\boldsymbol{x}), \boldsymbol{\theta}_d \rangle + \eta_d(\boldsymbol{x}),$$

where $\eta_d(\boldsymbol{x})$ represents the induced approximation error on arm $\boldsymbol{x}$ with respect to the low-dimensional embedding $\boldsymbol{\psi}_d(\cdot)$. Without loss of generality, we assume that the action set $\mathcal{X}$ is rich enough so that $\boldsymbol{\psi}_d(\mathcal{X})$ spans $\mathbb{R}^d$ for d considered in this paper. Otherwise, one can always project feature

[2] A generalized inversion is used for singular matrices. We refer to Appendix A.1 for detailed discussion.

representations $\boldsymbol{\psi}_d(\mathcal{X})$ into an even lower-dimensional space without losing information in the linear component.

We use $\widetilde{\gamma} : \mathbb{N} \to \mathbb{R}$ to represent the misspecification level: an upper bound of the induced approximation error across all arms, i.e., $\max_{\boldsymbol{x}\in\mathcal{X}}|\eta_d(\boldsymbol{x})| \le \widetilde{\gamma}(d)$. We define $g(d,\zeta) \coloneqq (1+\zeta)\inf_{\lambda\in\boldsymbol{\Lambda}_{\mathcal{X}}}\sup_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{\psi}_d(\mathcal{X}))}\|\boldsymbol{y}\|^2_{\boldsymbol{A}_{\boldsymbol{\psi}_d}(\lambda)^{-1}}$, which represents the optimal value of a transductive design among embeddings in $\mathbb{R}^d$. We define $\gamma(d) \coloneqq (16+8\sqrt{g(d,\zeta)})\,\widetilde{\gamma}(d)$, which quantifies the sub-optimality gap of the identified arm in the worst case. One can easily show $\gamma(d) \le O(\widetilde{\gamma}(d)\sqrt{d})$ through Kiefer-Wolfowitz theorem [27].

Remark 1. *We believe such optimality guarantees are un-improvable in general. In fact, a hard instance is constructed in [28] showing that, even with deterministic feedback, identifying a $o(\widetilde{\gamma}(d)\sqrt{d})$-optimal arm requires sample complexity exponential on d. On the other side, identifying a $\Omega(\widetilde{\gamma}(d)\sqrt{d})$-optimal only requires sample complexity polynomially in d. Such a sharp trade-off between optimality and sample complexity motivates our definition of $\gamma(d)$ (and our sample complexities are polynomially in d).*

We assume the knowledge of both the feature mapping $\boldsymbol{\psi}_d(\cdot)$ and the error function $\widetilde{\gamma}(\cdot)$. This assumption is mild since one can explicitly construct/analyze $\boldsymbol{\psi}_d(\cdot)$ and $\widetilde{\gamma}(\cdot)$ in many cases (as discussed in Section 3.2, Section 4 and Section 5). We further assume that $\gamma(d)$ can be made arbitrarily small for large enough d. Such assumption trivially holds if the rewards are perfectly explained for d large enough, i.e., $\widetilde{\gamma}(d) = 0$. We now define the *effective dimension* with respect to $\gamma(d)$ (induced from feature mapping $\boldsymbol{\psi}_d(\cdot)$) as follows.

Definition 1. *For any $\epsilon > 0$, we define the effective dimension as $d_{\text{eff}}(\epsilon) \coloneqq \min\{d \ge 1 : \gamma(d) \le \epsilon\}$.*

In general, the effective dimension $d_{\text{eff}}(\epsilon)$ captures the smallest dimension one needs to explore in order to identify an ϵ-optimal arm. Similar notions have been previously used in regret minimization settings [42, 43]. One can easily see that $d_{\text{eff}}(\epsilon_1) \le d_{\text{eff}}(\epsilon_2)$ as long as $\epsilon_1 \ge \epsilon_2$.

3.1 Algorithm and analysis

Algorithm 1 follows the framework of RAGE [13] to eliminate arms with sub-optimality gap $\ge O(2^{-k})$ at the k-th iteration. It runs for $n = O(\log(1/\epsilon))$ iterations and identifies an ϵ-optimal arm. We use optimal experimental design to select arms for faster elimination. For any fixed design $\lambda \in \boldsymbol{\Lambda}_{\mathcal{X}}$, with $N \ge r_d(\zeta)$ samples and an approximation factor ζ (with default value $\zeta \in [1/10, 1/4]$), the rounding procedure in $\mathbb{R}^d$, i.e., $\texttt{ROUND}(\lambda, N, d, \zeta)$, outputs a discrete allocation $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ satisfying

$$\max_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{\psi}_d(\mathcal{X}))}\|\boldsymbol{y}\|^2_{\left(\sum_{i=1}^N \boldsymbol{\psi}_d(\boldsymbol{x}_i)\boldsymbol{\psi}_d(\boldsymbol{x}_i)^\top\right)^{-1}} \le (1+\zeta)\max_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{\psi}_d(\mathcal{X}))}\|\boldsymbol{y}\|^2_{\boldsymbol{A}_{\boldsymbol{\psi}_d}(\lambda)^{-1}}/N. \tag{1}$$

Efficient rounding procedures exist with $r_d(\zeta) = \frac{d^2+d+2}{\zeta}$ [33] or $r_d(\zeta) = \frac{180d}{\zeta^2}$ [1, 13]. We refer reads to [13, 33, 1] for detailed rounding algorithms and the associated computational complexities.

Unlike RAGE that directly works in the original high-dimensional space, Algorithm 1 adaptively embeds arms into lower-dimensional spaces and carefully deals with the induced misspecifications. More specifically, the embedding dimension d_k is selected as the smallest dimension such that the induced error term ϵ_k is well controlled, i.e., $\epsilon_k \le O(2^{-k})$. The embedding is more aggressive at initial iterations due to larger error tolerance; The embedding dimension selected at the last iteration is (roughly) $d_{\text{eff}}(\epsilon)$ to identify an ϵ-optimal arm. The number of samples required for each iteration N_k is with respect to an experimental design in the lower-dimensional space *after embedding*. The ROUND procedure also becomes more efficient due to the embedding. Before stating our main theorem, we introduce the following complexity measure [38, 13, 10], which quantifies the hardness of the pure exploration problem (with respect to mapping $\boldsymbol{\psi}_d(\cdot)$).

$$\rho^\star_d(\epsilon) \coloneqq \inf_{\lambda\in\boldsymbol{\Lambda}_{\mathcal{X}}}\sup_{\boldsymbol{x}\in\mathcal{X}\setminus\{\boldsymbol{x}_\star\}}\frac{\|\boldsymbol{\psi}_d(\boldsymbol{x}_\star)-\boldsymbol{\psi}_d(\boldsymbol{x})\|^2_{\boldsymbol{A}_{\boldsymbol{\psi}_d}(\lambda)^{-1}}}{\max\{h(\boldsymbol{x}_\star)-h(\boldsymbol{x}),\epsilon\}^2}.$$

Theorem 1. *With probability of at least $1-\delta$, Algorithm 1 correctly outputs an ϵ-optimal arm with sample complexity upper bounded by*

$$640\sum_{k=1}^{\lceil\log_2(2/\epsilon)\rceil}\Big(\Big(k\,\rho^\star_{d_k}(2^{2-k})\log(k^2|\mathcal{X}|^2/\delta)\Big)+(r_{d_k}(\zeta)+1)\Big) \le \widetilde{O}\big(d_{\text{eff}}(\epsilon)\cdot\max\{\Delta_{\min},\epsilon\}^{-2}\big),$$

Algorithm 1 Arm Elimination with Adaptive Embedding and Induced Misspecification

Input: Action set $\mathcal{X}$, confidence parameter δ, accuracy parameter ϵ and rounding approximation factor ζ.

Set $n = \lceil \log_2(2/\epsilon) \rceil$ and $\widehat{\mathcal{S}}_1 = \mathcal{X}$.
for $k = 1, 2, \ldots, n$ **do**
  Set $\delta_k = \delta/k^2$, $d_k = d_{\text{eff}}(4 \cdot 2^{-k})$.
  Select feature representation $\boldsymbol{\psi}_{d_k}(\cdot)$, and calculate the induced misspecification level $\widetilde{\gamma}(d_k)$.
  Set $r_{d_k}(\zeta) = O(d_k/\zeta^2)$ as the number of samples needed for ROUND in $\mathbb{R}^{d_k}$.
  Set λ_k and τ_k be the design and the value of the following optimization problem

$$\inf_{\lambda \in \mathbf{\Lambda}_{\mathcal{X}}} \sup_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{\psi}_{d_k}(\widehat{\mathcal{S}}_k))} \|\boldsymbol{y}\|^2_{\boldsymbol{A}_{\boldsymbol{\psi}_{d_k}}(\lambda)^{-1}}.$$

  Set $\epsilon_k = 2\widetilde{\gamma}(d_k) + \widetilde{\gamma}(d_k)\sqrt{(1+\zeta)\,\tau_k}$,
  and $N_k = \max\{\lceil (2^{-k} - \epsilon_k)^{-2} 2(1+\zeta)\,\tau_k \log(|\widehat{\mathcal{S}}_k|^2/\delta_k) \rceil, r_{d_k}(\zeta)\}$.
  Get $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_k}\} = \texttt{ROUND}(\lambda_k, N_k, d_k, \zeta)$.
  Pull arms $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_k}\}$ and receive rewards $\{y_1, \ldots, y_{N_k}\}$.
  Set $\widehat{\boldsymbol{\theta}}_k = \boldsymbol{A}_k^{-1}\boldsymbol{b}_k$, where $\boldsymbol{A}_k = \sum_{i=1}^{N_k} \boldsymbol{\psi}_{d_k}(\boldsymbol{x}_i)\boldsymbol{\psi}_{d_k}(\boldsymbol{x}_i)^\top$ and $\boldsymbol{b}_k = \sum_{i=1}^{N_k} \boldsymbol{\psi}_{d_k}(\boldsymbol{x}_i) y_i$.
  Eliminate arms with respect to criteria

$$\widehat{\mathcal{S}}_{k+1} = \widehat{\mathcal{S}}_k \setminus \{\boldsymbol{x} \in \widehat{\mathcal{S}}_k : \exists \boldsymbol{x}' \text{ such that } (\boldsymbol{\psi}_{d_k}(\boldsymbol{x}') - \boldsymbol{\psi}_{d_k}(\boldsymbol{x}))^\top \widehat{\theta}_k \geq \omega_k(\boldsymbol{\psi}_{d_k}(\boldsymbol{x}') - \boldsymbol{\psi}_{d_k}(\boldsymbol{x}))\},$$

  where $\omega_k(\boldsymbol{y}) = \epsilon_k + \|\boldsymbol{y}\|_{\boldsymbol{A}_k^{-1}} \sqrt{2\log(|\widehat{\mathcal{S}}_k|^2/\delta_k)}$.
end for

Output: Output any arm in $\widehat{\mathcal{S}}_{n+1}$.

where $d_k = d_{\text{eff}}(4 \cdot 2^{-k}) \leq d_{\text{eff}}(\epsilon)$ *since* $4 \cdot 2^{-k} \geq \epsilon$ *when* $k \leq \lceil \log_2(2/\epsilon) \rceil$.

The rounding term $r_d(\zeta)$ commonly appears in the sample complexity of linear bandits [13, 24]; and our rounding term is with respect to the lower-dimensional space after embedding, which only scales with d_k rather than the ambient dimension. To further interpret the complexity, we define another complexity measure of a closely related linear bandit problem in the low-dimensional space and without model misspecifications.

$$\widetilde{\rho}^\star_d(\epsilon) := \inf_{\lambda \in \mathbf{\Lambda}_{\mathcal{X}}} \sup_{\boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{x}_\star\}} \frac{\|\boldsymbol{\psi}_d(\boldsymbol{x}_\star) - \boldsymbol{\psi}_d(\boldsymbol{x})\|^2_{\boldsymbol{A}_{\boldsymbol{\psi}_d}(\lambda)^{-1}}}{\max\{\langle \boldsymbol{\psi}_d(\boldsymbol{x}_\star) - \boldsymbol{\psi}_d(\boldsymbol{x}), \boldsymbol{\theta}_d \rangle, \epsilon\}^2},$$

where $\langle \boldsymbol{\psi}_d(\boldsymbol{x}_\star) - \boldsymbol{\psi}_d(\boldsymbol{x}), \boldsymbol{\theta}_d \rangle$ on the denominator represents the sub-optimality gap characterized by the linear component rather than the true sub-optimality gap $h(\boldsymbol{x}_\star) - h(\boldsymbol{x})$. The relation between $\rho^\star(\epsilon)$ and $\widetilde{\rho}^\star(\epsilon)$ is discussed as follows.

Proposition 1. *Suppose* $\max_{\boldsymbol{x} \in \mathcal{X}} |h(\boldsymbol{x}) - \langle \boldsymbol{\psi}_d(\boldsymbol{x}), \boldsymbol{\theta}_d \rangle| \leq \widetilde{\gamma}(d)$. *For any* $\epsilon \geq \widetilde{\gamma}(d)$, *we have* $\rho^\star_d(\epsilon) \leq 9\widetilde{\rho}^\star_d(\epsilon)$. *When* $\widetilde{\gamma}(d) < \Delta_{\min}/2$, $\widetilde{\rho}^\star_d(0)$ *represents the sample complexity of a closely-related linear bandit problem without model misspecifications, i.e.,* $\widetilde{h}(\boldsymbol{x}) = \langle \boldsymbol{\psi}_d(\boldsymbol{x}), \boldsymbol{\theta}_d \rangle$.

Remark 2. *When* $\widetilde{\gamma}(d) < \Delta_{\min}/2$, *our sample complexity upper bound is relevant to the sample complexity of closely-related linear bandit problems without model misspecifications in lower-dimensional spaces. In fact,* $\widetilde{\rho}^\star_d(0)\log(1/2.4\delta)$ *is the lower bound of the corresponding linear bandit problem in* $\mathbb{R}^d$ *[38, 13, 10].*

Remark 3. *Although the misspecification levels are generally known for situations considered in this paper, we also provide an algorithm that deals with unknown misspecification levels in Appendix D. Similar sample complexity guarantees are provided, but only in an unverifiably way (due to unknown misspecification levels): the algorithm starts to output* ϵ*-good arms after* N *samples, yet it doesn't know when to stop. We refer readers to [23] for details on the unverifiable sample complexity.*

3.2 Application to high-dimensional linear bandits

We apply the idea of adaptive embedding to high-dimensional linear bandits. We consider linear bandit problem of the form $\boldsymbol{h} = \boldsymbol{X}\boldsymbol{\theta}_\star$ where $\boldsymbol{X} \in \mathbb{R}^{K \times D}$ and the i-th row of $\boldsymbol{X}$ represents the

feature vector of arm $\boldsymbol{x}_i$. We assume that $\|\boldsymbol{\theta}_\star\|_2 \leq C$, which is commonly studied as the constrained linear bandit problem [41, 10].

Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the singular value decomposition (SVD) of $\boldsymbol{X}$, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ for some $r \leq \min\{K, D\}$. Let $u_{i,j}$ denote the (i,j)-th entry of matrix $\boldsymbol{U}$ and $\boldsymbol{u}_{:,i}$ denote the i-th column of $\boldsymbol{U}$ (similar notations for $\boldsymbol{V}$). We have

$$\boldsymbol{h} = \boldsymbol{X}\boldsymbol{\theta}_\star = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{\theta}_\star = \sum_{i=1}^{d} \sigma_i \boldsymbol{u}_{:,i}\boldsymbol{v}_{:,i}^\top\boldsymbol{\theta}_\star + \sum_{i=d+1}^{D} \sigma_i \boldsymbol{u}_{:,i}\boldsymbol{v}_{:,i}^\top\boldsymbol{\theta}_\star =: \sum_{i=1}^{d} \sigma_i \boldsymbol{u}_{:,i}\boldsymbol{v}_{:,i}^\top\boldsymbol{\theta}_\star + \boldsymbol{\eta},$$

where $\|\boldsymbol{\eta}\|_\infty \leq C\sum_{i=d+1}^{D}\sigma_i$. As a result, for any $d \leq r$, we can construct the feature mapping $\boldsymbol{\psi}_d(\boldsymbol{x}_i) = [\sigma_1 u_{i,1}, \ldots, \sigma_d u_{i,d}]^\top \in \mathbb{R}^d$ such that $h(\boldsymbol{x}_i) = \langle \boldsymbol{\psi}_d(\boldsymbol{x}_i), \widetilde{\boldsymbol{\theta}}_\star \rangle + \boldsymbol{\eta}(\boldsymbol{x}_i)$, where $\widetilde{\boldsymbol{\theta}}_\star = \left[\boldsymbol{V}^\top\boldsymbol{\theta}\right]_{[1:d]} \in \mathbb{R}^d$ is the associated reward parameter.[3] The upper bound of the induced misspecification can be expressed as $\widetilde{\gamma}(d) = C\sum_{i=d+1}^{D}\sigma_i$, which allows us to calculate $\gamma(d)$. We can then apply Algorithm 1 to identify an ϵ-optimal arm. A high-dimensional linear bandit instance is provided in Appendix B.3 showing that: Algorithm 1 takes $\widetilde{O}(1/\epsilon^2)$ samples to identify an ϵ-optimal arm, while the sample complexity upper bound of RAGE scales as $\widetilde{O}(D/\epsilon^2)$.

4 Pure exploration in RKHS

We consider a kernel function $\mathcal{K} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ over a compact set $\mathcal{Z}$; we assume the kernel function satisfies condition stated in the Mercer's Theorem (see Appendix E.1) and has eigenvalues decay fast enough (see Assumption 1). Let $\mathcal{H}$ be the Reproducing Kernel Hilbert Space (RKHS) induced from $\mathcal{K}$. We assume $\mathcal{X} \subseteq \mathcal{Z}$ and the true reward of any arm $\boldsymbol{x} \in \mathcal{X}$ is given by an unknown function $h \in \mathcal{H}$ such that $\|h\|_{\mathcal{H}} \leq 1$.

Let $\{\phi_j\}_{j=1}^\infty$ and $\{\mu_j\}_{j=1}^\infty$ be sequences of eigenfunctions and non-negative eigenvalues associated with kernel $\mathcal{K}$.[4] A corollary of Mercer's theorem shows that any $h \in \mathcal{H}$ can be written in the form of $h(\cdot) = \sum_{j=1}^\infty \theta_j\phi_j(\cdot)$ for some $\{\theta_j\}_{j=1}^\infty \in \ell^2(\mathbb{N})$ such that $\sum_{j=1}^\infty \theta_j^2/\mu_j < \infty$. We also have $\|h\|_{\mathcal{H}}^2 = \sum_{j=1}^\infty \theta_j^2/\mu_j$. Although functions in RKHS are non-linear in nature, we now can represent them in terms of an infinite-dimensional linear function. We construct feature mappings for the embedding next.

For any $\boldsymbol{x} \in \mathcal{X}$, we have $h(\boldsymbol{x}) = \sum_{j=1}^\infty \theta_j\phi_j(\boldsymbol{x}) = \sum_{j=1}^\infty \frac{\theta_j}{\sqrt{\mu_j}}\sqrt{\mu_j}\phi_j(\boldsymbol{x})$. Let $C_\phi := \sup_{\boldsymbol{x}\in\widetilde{\mathcal{X}}, j\geq 1}|\phi_j(\boldsymbol{x})|$. Since $\sum_{j=1}^\infty \theta_j^2/\mu_j = \|h\|_{\mathcal{H}}^2 \leq 1$ is bounded, for any $d \in \mathbb{N}$, we define feature mapping $\boldsymbol{\psi}_d(\boldsymbol{x}) = [\sqrt{\mu_1}\phi_1(\boldsymbol{x}), \ldots, \sqrt{\mu_d}\phi_d(\boldsymbol{x})]^\top \in \mathbb{R}^d$ such that

$$h(\boldsymbol{x}) = \langle \boldsymbol{\theta}_d, \boldsymbol{\psi}_d(\boldsymbol{x}) \rangle + \eta_d(\boldsymbol{x}),$$

where $\boldsymbol{\theta}_d = [\theta_1/\sqrt{\mu_1}, \ldots, \theta_d/\sqrt{\mu_d}]^\top \in \mathbb{R}^d$ and $|\eta_d(\boldsymbol{x})| \leq \widetilde{\gamma}(d) := C_\phi\sqrt{\sum_{j>d}\mu_j}$. We remark here that the constant C_ϕ is calculable and usually mild, e.g., $C_\phi = 1$ for $\phi_j(x) = \sin((2j-1)\pi x/2)$.

We can then construct $\gamma(d)$ and $d_{\text{eff}}(\epsilon)$ as in Section 3 and specialize Algorithm 1 to the kernel setting. Both $\gamma(d)$ and $d_{\text{eff}}(\epsilon)$ depend on eigenvalues of the associated kernel. Fortunately, fast eigenvalue decay are satisfied by most kernel functions, e.g., Gaussian kernel. We quantify such properties through the following assumption.

Assumption 1. *We consider kernels with the following eigenvalue decay with some absolute constants C_k and β.*

1. *Kernel $\mathcal{K}$ is said to have (C_k, β)-polynomial eigenvalue decay (with $\beta > 3/2$) if $\mu_j \leq C_k j^{-\beta}$ for all $j \geq 1$.*

[3] We note that the embeddings and associated quantities can also be constructed on the fly with respect to the set of uneliminated arms.

[4] With a known kernel, the sequence of eigenfunctions and eigenvalues can be analytically calculated or numeriaclly approximated [36, 35]. We assume the knowledge of eigenfunctions and eigenvalues in this paper.

2. *Kernel $\mathcal{K}$ is said to have (C_k, β)-exponential eigenvalue decay (with $\beta > 0$) if $\mu_j \leq C_k e^{-\beta j}$ for all $j \geq 1$.*

Theorem 2. *Suppose Assumption 1 holds. For any $\epsilon > 0$, the following statements hold when we specialize Algorithm 1 to the kernel setting.*

1. *Suppose $\mathcal{K}$ has (C_k, β)-polynomial eigenvalue decay. We have $d_{\text{eff}}(\epsilon) \leq O(\epsilon^{-2/(2\beta-3)})$, and the sample complexity of identifying an ϵ-optimal arm is upper bounded by $\widetilde{O}(\epsilon^{-2/(2\beta-3)} \max\{\Delta_{\min}, \epsilon\}^{-2})$.*

2. *Suppose $\mathcal{K}$ has (C_k, β)-exponential eigenvalue decay. We have $d_{\text{eff}}(\epsilon) \leq O(\log(1/\epsilon))$, and the sample complexity of identifying an ϵ-optimal arm is upper bounded by $\widetilde{O}(\max\{\Delta_{\min}, \epsilon\}^{-2})$.*

Remark 4. *Our sample complexity guarantees are directly related to the eigenvalue decay of the underlying kernel function, rather than the empirical kernel matrix as studied in previous works [7, 42]. Although one can also provide an instance dependent bound as in Theorem 1, the worst-case sample complexity bound in Theorem 2 provides insightful characterizations of the sample complexity in terms of eigenvalue decay. One should notice that with exponential eigenvalue decay, the sample complexity $\widetilde{O}(\epsilon^{-2})$ essentially matches, up to logarithmic factors, the complexity of distinguishing a two-armed bandit up to accuracy ϵ [25].*

5 Pure exploration with neural networks

In this section we present a neural network-based pure exploration algorithm in Algorithm 2. Our algorithm is inspired by the recently proposed neural bandits algorithms for regret minimization [47, 46]. At the core of our algorithm is to use a neural network $f(\boldsymbol{x}; \boldsymbol{\theta})$ to learn the unknown reward function h. Specifically, following [8, 47], we consider a fully connected neural network $f(\boldsymbol{x}; \boldsymbol{\theta})$ with depth $L \geq 2$

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sqrt{m}\boldsymbol{W}_L \sigma\Big(\boldsymbol{W}_{L-1}\sigma\big(\cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x})\big)\Big), \tag{2}$$

where $\sigma(x) := \max(x, 0)$ is the ReLU activation function, $\boldsymbol{W}_1 \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}_L \in \mathbb{R}^{1\times m}$, and for $2 \leq i \leq L-1$, $\boldsymbol{W}_i \in \mathbb{R}^{m\times m}$. Moreover, we denote $\boldsymbol{\theta} = [\text{vec}(\boldsymbol{W}_1)^\top, \ldots, \text{vec}(\boldsymbol{W}_L)^\top]^\top \in \mathbb{R}^p$, where $p = m + md + m^2(L-2)$ is the number of all the network parameters. We use $\boldsymbol{g}(\boldsymbol{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta})$ to denote the gradient of the neural network output with respect to the weights.

In detail, at k-th iteration, Algorithm 2 firstly applies its current gradient mapping $\boldsymbol{g}(\boldsymbol{x}; \boldsymbol{\theta}_{k-1})$ over the whole action set $\mathcal{X}$, and obtains the collection of gradients $\boldsymbol{G} \in \mathbb{R}^{|\mathcal{X}|\times p}$. Then Algorithm 2 does SVD over $\boldsymbol{G}$ and constructs a d_k-dimensional feature mapping $\boldsymbol{\psi}_{d_k}$, which can be regarded as the projection of the gradient feature mapping $\boldsymbol{g}(\boldsymbol{x}; \boldsymbol{\theta}_{k-1})$ to the most informative d_k-dimensional eigenspace. Here we choose d_k such that the summation of the eigenvalues of the remaining eigenspace be upper bounded by some error $\bar{\epsilon}$. Algorithm 2 then computes the optimal design λ_k over $\boldsymbol{\psi}_k(\mathcal{X})$ and pulls arms $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_k}\}$ based on both the design λ_k and the total number of allocations N_k. Finally, Algorithm 2 trains a new neural network $f(\boldsymbol{x}; \boldsymbol{\theta}_k)$ using gradient descent starting from the initial parameter $\boldsymbol{\theta}_0$ (details are deferred to Appendix F), then eliminates the arms $\boldsymbol{x}$ in the current arm set $\widehat{\mathcal{S}}_k$ which are sub-optimal with respect to the neural network function value $f(\boldsymbol{x}; \boldsymbol{\theta}_k)$.

The main difference between Algorithm 2 and its RKHS counterpart is as follows. Unlike Algorithm 1 which works on known feature mappings $\boldsymbol{\psi}_d$ (derived from a known kernel $\mathcal{K}$), Algorithm 2 does not have information about the feature mapping, and thus it constructs the feature mapping from the raw high-dimensional up-to-date gradient mapping $\boldsymbol{g}(\boldsymbol{x}; \boldsymbol{\theta}_{k-1})$. The feature mapping is constructed with respect to a *trained* neural work, which leverages the great representation power of neural networks. This makes Algorithm 2 a more general and flexible algorithm than Algorithm 1.

Now we present the main theorem of Algorithm 2. Let $\boldsymbol{H}^{|\mathcal{X}|\times|\mathcal{X}|}$ be the Neural Tangent Kernel (NTK)[17] gram matrix over all arms $\mathcal{X}$ (the detailed definition of $\boldsymbol{H}$ is deferred to Appendix F). We define the effective dimension for the neural version as below. The definition is similar to Definition 1.

Algorithm 2 Neural Arm Elimination

Input: Action set $\mathcal{X}$, initial parameter $\boldsymbol{\theta}_0$, neural network $f(\boldsymbol{x};\boldsymbol{\theta})$, gradient mapping $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\theta})$, width of the matrix m, parameter of the number of allocations A, approximation parameter ζ, regularization parameter α, error parameter $\bar{\epsilon},\epsilon$, confidence level $\delta_k=\delta/(8k^2)$

Set $\widehat{\mathcal{S}}_1=\mathcal{X}$.

for $k=1,2,\ldots,n$ **do**

Construct the truncated feature representation $\boldsymbol{\psi}_k(\mathcal{X})$ based on gradient mapping $\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta}_{k-1})$. In detail, let $\boldsymbol{G}\in\mathbb{R}^{|\mathcal{X}|\times p}$ be the collection of gradients such that

$$\boldsymbol{G}=[\boldsymbol{g}(\boldsymbol{x}_1;\boldsymbol{\theta}_{k-1})^\top;\ldots;\boldsymbol{g}(\boldsymbol{x}_{|\mathcal{X}|};\boldsymbol{\theta}_{k-1})^\top]/\sqrt{m}\in\mathbb{R}^{|\mathcal{X}|\times p} \tag{3}$$

Let $[\boldsymbol{U},\boldsymbol{\Sigma},\boldsymbol{V}]$ be the SVD of $\boldsymbol{G}$, where $\boldsymbol{U}=(u_{i,j})\in\mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$, $\boldsymbol{\Sigma}=[\mathrm{diag}(e_1,\ldots,e_{|\mathcal{X}|}),0]\in\mathbb{R}^{|\mathcal{X}|\times p}$, $\boldsymbol{V}\in\mathbb{R}^{p\times p}$. Get $d_k=\min\{d\in[|\mathcal{X}|]:\sum_{i=d+1}^{|\mathcal{X}|}e_i\le\bar{\epsilon}\}$, and set $\boldsymbol{\psi}_{d_k}(\boldsymbol{x}_i)=(e_1u_{i,1},\ldots,e_{d_k}u_{i,d_k})\in\mathbb{R}^{d_k}$.

Set λ_k and τ_k be the experimental design and the value of the following optimization problem

$$\inf_{\lambda\in\boldsymbol{\Lambda}_{\mathcal{X}}}\sup_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{\psi}_{d_k}(\widehat{\mathcal{S}}_k))}\|\boldsymbol{y}\|^2_{\boldsymbol{A}_{\boldsymbol{\psi}_{d_k}}(\lambda)^{-1}}. \tag{4}$$

Set $N_k=\max\left\{2^{2k}\,A(1+\zeta)\log\big(|\mathcal{X}|^2/\delta_k\big),r_{d_k}(\zeta)\right\}$.

Get $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{N_k}\}=\texttt{ROUND}(\lambda_k,N_k,d_k,\zeta)$.

Pull arms $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{N_k}\}$ and receive rewards $\{y_1,\ldots,y_{N_k}\}$.

Using J_k step η_k-step size gradient descent to optimize the following loss function to obtain $\boldsymbol{\theta}_k$,

$$\boldsymbol{\theta}_k=\arg\min L(\boldsymbol{\theta}):=\sum_{j=1}^{N_k}(f(\boldsymbol{x}_j;\boldsymbol{\theta})-y_j)^2+\frac{m\alpha}{2}\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2. \tag{5}$$

Set $\boldsymbol{A}_k=\alpha\boldsymbol{I}+\sum_{i=1}^{N_k}\boldsymbol{\psi}_{d_k}(\boldsymbol{x})\boldsymbol{\psi}_{d_k}(\boldsymbol{x})^\top$ and eliminate arms with respect to criteria

$$\widehat{\mathcal{S}}_{k+1}=\widehat{\mathcal{S}}_k\setminus\{\boldsymbol{x}\in\widehat{\mathcal{S}}_k:\exists\boldsymbol{x}'\text{ such that }f(\boldsymbol{x}';\boldsymbol{\theta}_k)-f(\boldsymbol{x};\boldsymbol{\theta}_k)\ge 2^{-k}/8+3\epsilon/8\}.$$

end for

Output: Output any arm in $\widehat{\mathcal{S}}_{n+1}$.

Definition 2 (Neural version)**.** *For any $\epsilon>0$, we define the effective dimension as $d_{\mathrm{eff}}(\epsilon):=\min\{d\in[|\mathcal{X}|]:\sum_{i=d+1}^{|\mathcal{X}|}\lambda_i(\boldsymbol{H})\le\epsilon\}$, where $\lambda_i(\boldsymbol{H})$ is the i-th eigenvalue of $\boldsymbol{H}$.*

Next, we make standard assumptions for the initialization of neural networks and the arms $\boldsymbol{x}\in\mathcal{X}$.

Assumption 2 ([47])**.** *There exists $\lambda_0>0$ such that $\boldsymbol{H}\succeq\lambda_0\boldsymbol{I}$. For any $\boldsymbol{x}\in\mathcal{X}$, the arm $\boldsymbol{x}$ satisfies $\|\boldsymbol{x}\|_2=1$ and that its j-th coordinate is identical to its $j+d/2$-th coordinate. Meanwhile, the initial parameter $\boldsymbol{\theta}_0=[vec(\boldsymbol{W}_1)^\top,\ldots,vec(\boldsymbol{W}_L)^\top]^\top$ is initialized as follows: for $1\le l\le L-1$, $\boldsymbol{W}_l$ is set to $\begin{pmatrix}\boldsymbol{W}&0\\0&\boldsymbol{W}\end{pmatrix}$, where each entry of $\boldsymbol{W}$ is generated independently from $\mathcal{N}(0,4/m)$; $\boldsymbol{W}_L$ is set to $(\boldsymbol{w}^\top,-\boldsymbol{w}^\top)$, where each entry of $\boldsymbol{w}$ is generated independently from $\mathcal{N}(0,2/m)$.*

We now present our main theorem for pure exploration with neural network approximation. The formal version of our theorem is deferred to Appendix F.

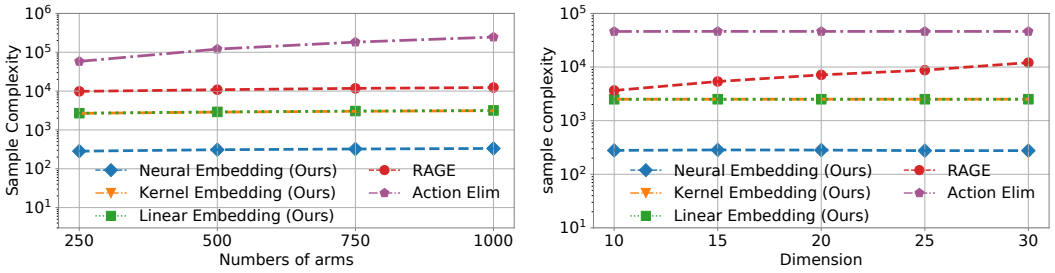
Theorem 3 (Informal)**.** *Under Assumption 2, with proper selection of parameters $\alpha,n,\bar{\epsilon},A,\eta_k,J_k$, then when $m=poly(|\mathcal{X}|,L,\lambda_0^{-1},\log(|\mathcal{X}|/\delta_k),N_k,\alpha,\bar{\epsilon}^{-1})$, with probability at least $1-\delta$, $\widehat{\mathcal{S}}_{K+1}$ only includes arm $\boldsymbol{x}$ satisfying $\Delta_{\boldsymbol{x}}\le\epsilon$, and the total sample complexity of Algorithm 2 is bounded by*

$$N=\widetilde{O}\Big((1+\zeta)d_{\mathrm{eff}}(\bar{\epsilon}^2/|\mathcal{X}|)/\epsilon^2+r_{d_{\mathrm{eff}}(\bar{\epsilon}^2/|\mathcal{X}|)}(\zeta)\Big)=\widetilde{O}\Big(d_{\mathrm{eff}}(\bar{\epsilon}^2/|\mathcal{X}|)\epsilon^{-2}\Big).$$

Remark 5. *For the case where the effective dimension can be well bounded, e.g., $d_{\mathrm{eff}}(\bar{\epsilon}^2/|\mathcal{X}|)=O(\log(|\mathcal{X}|/\bar{\epsilon}^2))$, Theorem 7 suggests that Algorithm 2 is able to identify an ϵ-optimal arm within $\tilde{O}(\epsilon^{-2})$ samples. That suggests that our neural network-based algorithm is efficient without constructing a low-dimensional linear approximation of h in prior, like the previous linear or RKHS approaches.*

6 Experiments

We conduct four experiments on synthetic and real-world datasets. We specialize our embedding idea to the neural, kernel, and linear regimes, and denote the algorithms as `NeuralEmbedding` (Algorithm 2), `KernelEmbedding` (Algorithm 1 with Gaussian kernel), and `LinearEmbedding` (Algorithm 1 with linear representation), respectively. We compare our algorithms with two baselines: `RAGE` and `ActionElim`. `RAGE` [13] conducts pure exploration in linear bandits and `ActionElim` [18, 11] ignores all feature representations. The (empirical) sample complexity of each algorithm is calculated as the number of samples needed so that the uneliminated set contains only ϵ-optimal arms. Unsuccessful runs, i.e., those terminate with non-ϵ-optimal arms, are reported as failures. In our experiments, we set $\epsilon = 0.1$ and $\delta = 0.05$. All results are averaged over 50 runs.[5]



(a) (b)

Figure 1: Experiments on synthetic datasets. (a) linear reward function with $D = 20$; (b) nonlinear reward function with $K = 200$.

Synthetic datasets. We first generate the feature matrix $\widetilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{E} \in \mathbb{R}^{K \times D}$ where $\boldsymbol{X}$ is constructed as a rank-2 matrix and $\boldsymbol{E}$ is a perturbation matrix with tiny spectral norm (See Appendix G for details). Each row of $\widetilde{\boldsymbol{X}}$ represents the feature representation of an arm, and those features can be grouped into two clusters with equal size. In the case with linear rewards, we let $\boldsymbol{\theta}_\star$ equal to the first row of $\widetilde{\boldsymbol{X}}$. For nonlinear rewards, the reward of each arm is set as the 2-norm of its feature representation. We vary the number of arms K and the ambient dimension D in our experiments.

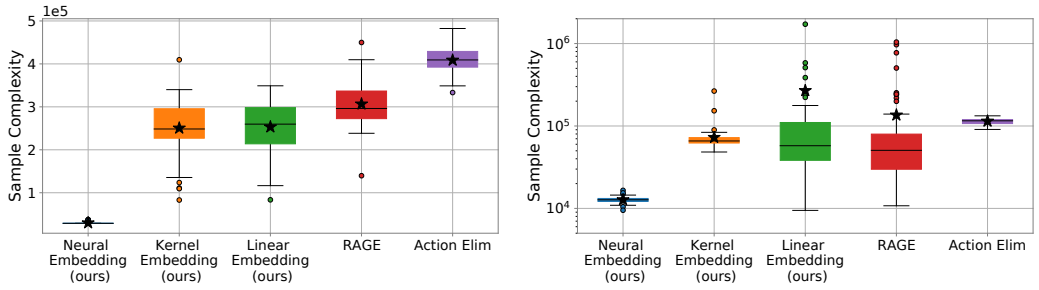
Fig. 1 shows experimental results on synthetic datasets. All algorithms successfully identify ϵ-optimal arms with zero empirical failure probability (due to the simplicity of the datasets). In terms of sample complexity, `NeuralEmbedding` outperforms all other algorithms in most cases, and `KernelEmbedding` and `LinearEmbedding` significantly outperform `RAGE` and `Action Elim`. The sample complexities of `NeuralEmbedding`, `KernelEmbedding` and `LinearEmbedding` are not affected when increasing number of arms or dimensions since they first identify the important subspace and then conduct elimination. On the other side, the sample complexity of `ActionElim` gets larger with increasing number of arms and the sample complexity of `RAGE` gets larger with increasing dimensions.

MNIST dataset. The MNIST dataset [29] contains hand-written digits from 0 to 9. We view each digit as an arm, and set their rewards according to the confusion matrix of a trained classifier. Digit 7 is chosen as the optimal arm with reward 1; the reward of digits $1, 2$ and 9 are set to be 0.8, and all other digits have reward 0.5. In each experiment, we randomly draw 200 samples (20 samples each digit) from the dataset. We project the raw feature matrix $\boldsymbol{X} \in \mathbb{R}^{200 \times 784}$ into a lower-dimensional space $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{200 \times 200}$ so that it becomes full rank (but without losing any information) and feasible for `RAGE`. Our goal is to correctly identify a digit 7.

Yahoo dataset. The Yahoo! User Click Log Dataset R6A[6] contains users' click-through records. Each record consists of a 36-dimensional feature representation (obtained from an outer product), and a binary outcome stating whether or not a user clicked on the article. We view each record as

[5] All algorithms are elimination-styled for fair comparison. Other implementation details are deferred to Appendix G.

[6] `https://webscope.sandbox.yahoo.com`



(a) MNIST (b) Yahoo

Figure 2: Experiments on real-world datasets. The mean sample complexity is represented by a black star. The mean sample complexities of `LinearEmbedding` and `RAGE` are heavily affected by outliers in the Yahoo dataset.

an arm, and set the reward as 0.8 (if clicked) or 0.3 (if not clicked) to makes the problem harder. In each experiment, we randomly draw 200 arms from the dataset, where 5 of them having rewards 0.8 (proportional to true click-through ratio), Our goal is to identify an arm with rewards 0.8. Our experimental setup is similar to the one used in Fiez et al. [13]. However, their true rewards are obtained from a least square regression. We do not enforce linearity in rewards in our experiment.

Box plots in Fig. 2 show the sample complexity of each algorithm on real-world datasets. `NeuralEmbedding` significantly outperforms all other algorithms thanks to (1) the representation power of neural networks and (2) efficient exploration in low-dimensional spaces. `KernelEmbedding` and `LinearEmbedding` have competitive performance on the MNIST dataset. Table 1 shows the success rate of each algorithm. Linear methods such as `RAGE` and `LinearEmbedding` have relatively low success rates on the Yahoo dataset (with nonlinear rewards). Our `NeuralEmbedding` and `KernelEmbedding` methods have high success rates since they are designed for nonlinear setting.

Table 1: Success rates on real-world datasets

	`NeuralEmbedding`	`KernelEmbedding`	`LinearEmbedding`	`RAGE`	`ActionElim`
MNIST	98%	100%	100%	100%	100%
Yahoo	100%	98%	88%	90%	100%

7 Conclusion

We introduce the idea of adaptive embedding in bandit pure exploration. Unlike existing works that passively deal with model misspecifications, we adaptively embed high-dimensional feature representations into lower-dimensional spaces to avoid the curse of dimensionality. The induced misspecifications are carefully dealt with. We further apply our approach to two under-studied settings with the nonlinearity: (1) pure exploration in an RKHS and (2) pure exploration with neural networks. Our sample complexity guarantees depend on the effective dimension of the feature spaces in the kernel or neural representations. We conduct extensive experiments on both synthetic and real-world datasets, and our algorithms greatly outperform existing ones.

Our current analysis with neural representations is in the NTK regime, which can only describe a part of the representation of the neural networks. We leave extending our algorithm to more general settings (beyond the NTK regime) as a future direction.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and area chair for their helpful comments. YZ and RN are partially supported by AFOSR grant FA9550-18-1-0166. DZ and QG are partially supported by the National Science Foundation CAREER Award 1906169, IIS-1904183 and AWS Machine Learning Research Award. RJ and RW are partially supported by AFOSR FA9550-18-1-0166, NSF OAC-1934637, NSF DMS-2023109, and NSF DGE-2022023. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal discrete optimization for experimental design: A regret minimization approach. *Mathematical Programming*, pages 1–40, 2020.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 2019.

[3] Maryam Aziz, Emilie Kaufmann, and Marie-Karelle Riviere. On multi-armed bandit designs for phase i clinical trials. *arXiv preprint arXiv:1903.07082*, 2019.

[4] Peter L Bartlett, Victor Gabillon, and Michal Valko. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Algorithmic Learning Theory*, pages 184–206. PMLR, 2019.

[5] Robert E Bechhofer. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. *Biometrics*, 14(3):408–429, 1958.

[6] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer, 2009.

[7] Romain Camilleri, Julian Katz-Samuels, and Kevin Jamieson. High-dimensional experimental design and kernel bandits. *arXiv preprint arXiv:2105.05806*, 2021.

[8] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, 2019.

[9] Lijie Chen and Jian Li. On the optimal sample complexity for best arm identification. *arXiv preprint arXiv:1511.03774*, 2015.

[10] Rémy Degenne, Pierre Ménard, Xuedong Shang, and Michal Valko. Gamification of pure exploration for linear bandits. In *International Conference on Machine Learning*, pages 2432–2442. PMLR, 2020.

[11] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

[12] Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

[13] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, pages 10667–10677, 2019.

[14] Dylan J Foster, Claudio Gentile, Mehryar Mohri, and Julian Zimmert. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

[15] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027. PMLR, 2016.

[16] Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[17] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[18] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2014.

[19] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439. PMLR, 2014.

[20] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

[21] Hideaki Kano, Junya Honda, Kentaro Sakamaki, Kentaro Matsuura, Atsuyoshi Nakamura, and Masashi Sugiyama. Good arm identification via bandit feedback. *Machine Learning*, 108(5): 721–745, 2019.

[22] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 1238–1246. PMLR, 2013.

[23] Julian Katz-Samuels and Kevin Jamieson. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*, pages 1781–1791. PMLR, 2020.

[24] Julian Katz-Samuels, Lalit Jain, Zohar Karnin, and Kevin Jamieson. An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits. *arXiv preprint arXiv:2006.11685*, 2020.

[25] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1): 1–42, 2016.

[26] Abbas Kazerouni and Lawrence M Wein. Best arm identification in generalized linear bandits. *Operations Research Letters*, 49(3):365–371, 2021.

[27] Jack Kiefer and Jacob Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[28] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR, 2020.

[29] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[30] Blake Mason, Lalit Jain, Ardhendu Tripathy, and Robert Nowak. Finding all ϵ-good arms in stochastic bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] Rémi Munos. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning, 2014. https://hal.archives-ouvertes.fr/hal-00747575v4/document.

[32] Edward Paulson et al. A sequential procedure for selecting the population with the largest mean from k normal populations. *Annals of Mathematical Statistics*, 35(1):174–180, 1964.

[33] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[34] Sivan Sabato. Epsilon-best-arm identification in pay-per-reward multi-armed bandits, 2019. https://openreview.net/forum?id=H1xkvNrlLS.

[35] Gabriele Santin and Robert Schaback. Approximation of eigenfunctions in kernel-based spaces. *Advances in Computational Mathematics*, 42(4):973–993, 2016.

[36] Stewart Schlesinger. Approximating eigenvalues and eigenfunctions of symmetric kernels. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):1–14, 1957.

[37] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[38] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. *arXiv preprint arXiv:1409.6110*, 2014.

[39] Gilbert W Stewart. Perturbation theory for the singular value decomposition. Technical report, University of Maryland, 1998. https://drum.lib.umd.edu/handle/1903/552.

[40] Ervin Tanczos, Robert Nowak, and Bob Mankoff. A kl-lucb bandit algorithm for large-scale crowdsourcing. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5896–5905, 2017.

[41] Chao Tao, Saúl Blanco, and Yuan Zhou. Best arm identification in linear bandits with linear dimension dependency. In *International Conference on Machine Learning*, pages 4877–4886. PMLR, 2018.

[42] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[43] Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *International Conference on Machine Learning*, pages 46–54. PMLR, 2014.

[44] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[45] Liyuan Xu, Junya Honda, and Masashi Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 843–851. PMLR, 2018.

[46] Weitong ZHANG, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. In *International Conference on Learning Representations*, 2020.

[47] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.