ORTHOGONAL FUNCTION REPRESENTATIONS FOR CONTINUOUS ARMED BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper addresses the continuous-armed bandit problem, which is a generalization of the standard bandit problem where the action space is a $d-$dimensional hypercube $\mathcal{X} = [-1, 1]^d$ and the reward is an $s-$times differentiable function $f : \mathcal{X} \to \mathbb{R}$. Traditionally, this problem is solved by assuming an implicit feature representation in a Reproducing Kernel Hilbert Space (RKHS), where the objective function is linear in this transformation of $\mathcal{X}$. In addition to this additional intake, this comes at the cost of overwhelming computational complexity. In contrast, we propose an explicit representation using an orthogonal feature map (Fourier, Legendre) to reduce the problem to a linear bandit with misspecification. As a result, we develop two algorithms `OB-LinUCB` and `OB-PE`, achieving state-of-the-art performance in terms of regret and computational complexity.

1 INTRODUCTION

This paper considers the problem of optimizing a reward function $f : \mathcal{X} \to \mathbb{R}$. As in most of the literature, we will always take $\mathcal{X} = [-1, 1]^d$, as the results can be trivially extended to any compact domain with Lipschitz boundary. At each round, the learner chooses an action $x_t \in \mathcal{X}$ and observes a noisy sample for $f(x_t)$. The goal is to minimize the cumulative regret, defined as $\sum_{t=1}^{T} f(x_t) - \sup_{x\in\mathcal{X}} f(x)$, where T is a given time horizon. This is also known as the continuous armed bandit problem, which generalizes, increasing the number of arms to an uncountable set, the finite ”multi-armed bandit problem” (Lattimore & Szepesvári, 2020; Auer et al., 2002).

In the continuous bandit problem (Agrawal, 1995), the algorithm cannot try all the arms even once, so it has to exploit some notion of smoothness of the function f to estimate the mean reward of the majority of arms without pulling them. Without any assumption of smoothness, it can be demonstrated that the problem is non-learnable. In the literature on continuous bandits, two primary families of methods have been introduced, depending on the specific assumptions made about the function f.

1. Lipschitzness: under the assumption that f is Lipschitz or Hölder continuous, algorithms based on discretization (Kleinberg, 2004; Kleinberg et al., 2008) are able to achieve the best possible regret bound.
2. RKHS representation: under the assumption that f belongs to a reproducing kernel Hilbert space with a known kernel, it is possible to solve the problem with kernel methods, and in particular Gaussian processes (Srinivas et al., 2009; de Freitas et al., 2012; Valko et al., 2013; Chowdhury & Gopalan, 2017; Shekhar & Javidi, 2018; Li & Scarlett, 2022).

The idea of the second family of algorithms, which are considered to be state of the art for practical applications, is that there is a feature map $\boldsymbol{\varphi} : [a, b] \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, such that for $x, x' \in [a, b]$ the kernel $k(x, x') = \langle \boldsymbol{\varphi}(x), \boldsymbol{\varphi}(x') \rangle_{\mathcal{H}}$ and an unknown vector $\boldsymbol{v}$ such that $f(x) = \langle \boldsymbol{\varphi}(x), \boldsymbol{v} \rangle_{\mathcal{H}}$. In this way, the problem is reduced to a linear bandit (Abbasi-Yadkori et al., 2011) and, by using the kernel trick (Schölkopf, 2000), it is possible to solve it even without an explicit calculation of $\boldsymbol{\varphi}$. While very elegant, this solution leads to relevant downsides: a rather specific assumption on f and, most importantly, a terrible computational inefficiency since kernel methods are well-known to be very slow in prediction.

In this paper, we propose a different approach to reduce the continuous bandit problem to a linear bandit. We drop the assumption that this feature map is exact, thus admitting to have $f(x) =$

$\langle \boldsymbol{\varphi}(x), \boldsymbol{v} \rangle_{\mathcal{H}} + \varepsilon(x)$ for some error function ε. In this way, we are able to explicitly compute the approximate feature map by means of an orthogonal basis of a Hilbert space and reduce the problem to a misspecified linear bandit (Lattimore et al., 2020; Ghosh et al., 2017).

1.1 Related works

As mentioned earlier, the main comparisons for the algorithms of this paper are kernel methods for the continuous bandit problem, which are able to solve it by viewing it as a linear bandit through an implicit representation given by a feature map in an RKHS. Thanks to their excellent performance in practical scenarios, much research has been done in this direction, starting with (Srinivas et al., 2009), which introduced the popular `GP-UCB` algorithm to (Chowdhury & Gopalan, 2017), which invented its improved version `IGP-UCB`. Improved regret bounds were obtained by (Valko et al., 2013; Li & Scarlett, 2022), while the problem of reducing the overwhelming computational complexity has been tackled in (Calandriello et al., 2019; 2020; 2022). The connection with recent advancements in this field is very deep and will be explained in detail in the appendix E.2. Recently, an idea similar to the one in our paper has been studied by Liu et al. (2021). Their method requires that the objective function f is smooth in the Hölder sense, i.e., it admits continuous derivatives up to a certain order, and the last derivative considered satisfies an Hölder inequality. Their algorithm relies on both discretization of the state space, dividing the domain of f in a number of bins depending on T, and linear bandits, making a *local* linear representation of f in each bin. This is done by Taylor polynomials: function f is locally approximated by its Taylor expansion. Conversely, our paper focuses on a *global* representation of the function f by means of a generalized Fourier series. In this way, we obtain an algorithm that is arguably much simpler and avoids fitting a number of linear bandit instances that diverges with T (thus obtaining also a better computational efficiency).

2 Preliminaries

Let us start by introducing the main definitions and notations used in this work, which will also be summarized in a table in the appendix A. In this paper, we study the continuous armed bandit problem over the space $\mathcal{X} = [-1, 1]^d$ where the objective function f is assumed to be smooth of some order s known to the learner (including the case $s = +\infty$). Precisely, given that $f \in \mathcal{C}^s(\mathcal{X})$, the learner has to choose at each time step an arm $x_t \in \mathcal{X}$, receiving a reward $r_t = f(x_t) + \eta_t$, where $\{\eta_t\}_t$ are i.i.d. samples form a $\sigma-$subgaussian distribution. The goal is to minimize the regret up to a known time horizon T, defined as $R_T := \sum_{t=1}^T \sup_{x \in \mathcal{X}} f(x) - f(x_t)$. In addition to the space of $s-$ times differentiable functions $\mathcal{C}^s(\mathcal{X})$, we will make use of the function space of square-integrable functions $L^2(\mathcal{X})$, defined as

$$L^2(\mathcal{X}) := \left\{ f : \int_{\mathcal{X}} f(x)^2 \, dx < +\infty \right\}.$$

This space (if we identify functions that are equal almost everywhere) is a Hilbert space endowed with the following scalar product $\langle f, g \rangle_{L^2} := \int_{\mathcal{X}} f(x) g(x) \, dx$. In a Hilbert space, two functions are said to be orthogonal if their scalar product is zero. A set of vectors in a Hilbert space is called an *orthogonal* basis if every element of the space can be written as a linear combination of elements in the set and all the elements are pairwise orthogonal. We will see some examples of orthogonal bases in the following subsection.

3 Orthogonal functions and their properties

For simplicity of notation, we start from the case where $d = 1$, so that $\mathcal{X} = [-1, 1]$. The generalization for higher d will come by making a Cartesian product of the basis functions. The most famous basis of $L^2(\mathcal{X})$ is indeed the Fourier basis. We use this basis to define a feature map associating to each $x \in \mathcal{X}$, the application of the basis to that point.

Definition 1 (Fourier feature map). *Define, for any $n \geq 0$, the following functions*

$$\varphi_{F,n}(x) := \begin{cases} 1/\sqrt{2} & n = 0 \\ \cos\left(\frac{n}{2}\pi x\right) & n > 0 \textit{ even} \\ \sin\left(\frac{n+1}{2}\pi x\right) & n \textit{ odd} \end{cases}.$$

Furthermore, for every $N \in \mathbb{N}$, define the following feature maps $\boldsymbol{\varphi}_{F,N} : [-1,1] \to \mathbb{R}^N$

$$\boldsymbol{\varphi}_{F,N}(x) := [\varphi_{F,0}(x), \varphi_{F,1}(x) \dots \varphi_{F,N}(x)]$$

As anticipated, the importance of the Fourier feature map lies in the fact that the set $\{\varphi_{F,n}\}_{n\in\mathbb{N}}$ forms an *orthogonal* basis for the space $L^2(\mathcal{X})$. However, the Fourier basis is not the only well-known function sequence to enjoy this property. Indeed, there are also sequences of polynomials forming an orthogonal basis of $L^2(\mathcal{X})$, the most famous one being the Legendre polynomials.

Definition 2 (Legendre feature map). *(Quarteroni et al., 2010) Calling $\varphi_{L,n}(x)$ the n-th order Legendre polynomial, define, for every $N \in \mathbb{N}$, the following feature map $\boldsymbol{\varphi}_{L,N} : [-1,1] \to \mathbb{R}^N$*

$$\boldsymbol{\varphi}_{L,N}(x) := [\varphi_{L,0}(x), \dots \varphi_{L,N}(x)]$$

Legendre polynomials are currently used in numerical mathematical applications like polynomial interpolation and numerical quadrature.

3.1 Multi-dimensional generalization

Even if these basis functions are all defined on the interval $[-1,1]$, we can generalize them to the case where $\mathcal{X} = [-1,1]^d$ by doing a Cartesian product operation. Precisely, the generalization of a given basis $\boldsymbol{\varphi}_N$ of $L^2([-1,1])$ to $[-1,1]^d$ is given by

$$\boldsymbol{\varphi}_N^d(x_1, \dots x_d) := \left\{ \varphi_{N_1}(x_1) \times \varphi_{N_2}(x_2) \dots \varphi_{N_d}(x_d) : \sum_{i=1}^{d} N_i \leq N \right\}.$$

This formula applies to Fourier and Legendre bases in the same way. Unlike the 1-dimensional case, where we needed exactly N features to get a basis of degree N, this number is significantly bigger here. In fact, it can be proved that the length of the feature vector $\boldsymbol{\varphi}_N^d(x_1, \dots x_d)$ corresponds to $\binom{N+d}{N}$, which is always bounded by N^d. It is much more difficult to visualize this feature map, which goes $\mathcal{X} \to \mathbb{R}^{\binom{N+d}{N}}$ still, the mathematically we are following the very same idea of $d = 1$.

We can use these feature maps to project any function in $L^2(\mathcal{X})$ on the linear subspace generated by the first N elements of the bases. Formally, being $\{\varphi_n\}_{n\in\mathbb{N}}$ the Legendre or Fourier basis functions, for any $f \in L^2(\mathcal{X})$ there are coefficients $\{a_n\}_{n\in\mathbb{N}}$ such that $\sum_{n=0}^{N} a_n\varphi_n \overset{L^2}{\to} f$, and they can be found with a simple scalar product: $a_n = \langle f, \varphi_n \rangle_{L^2}$. The existence of this representation is sufficient to ensure that the function f can be approximated by a linear function in a features space given by our choice of $\{\varphi_n\}_{n\in\mathbb{N}}$. Not only the existence of the sequence a_n is ensured, but it has remarkable properties if the function f is smooth. If $f \in \mathcal{C}^s(\mathcal{X})$, as in our assumptions, the coefficients a_n form a fastly decaying sequence, and the precise way in which $a_n \to 0$ depends on s. These kinds of results are known in Approximation Theory as decay properties.

3.2 Decay properties

The smoothness of f can heavily influence the magnitude of its coefficients a_n in the Fourier of Legendre basis. To give the idea behind our method, we list here two informal theorems in case of $d = 1$, that is, for $\mathcal{X} = [-1,1]$. These decay properties allow us to prove that smooth functions can be very well approximated by orthogonal polynomials or Fourier series.

Theorem 1. ***(Informal)*** *Let $f : [-1,1] \to \mathbb{R}$ be a measurable function. If $f \in \mathcal{C}^s([-1,1])$, and $f^{(s+1)}$ is square-integrable then,*

$$\left\| \sum_{n=0}^{N} a_{L,n}\varphi_{L,n} - f \right\|_{\infty} = \mathcal{O}(N^{-s-1/2}),$$

The same result holds for the Fourier basis $\{\varphi_{F,n}\}_n$ if $f \in \mathcal{C}^s_{per}([-1,1])$[1].

[1] Periodic conditions at the boundary.

Algorithm 1 OB-**LinBand** Algorithm

```
Require: Linear bandit algorithm LinBand, Time horizon $T$, Degree $N$ of the feature map, Error
    probability $\delta$, Basis function to use $\{\varphi_n\}_n$, Upper bound $S$ for $\|f\|_{L^2}$
1: $\boldsymbol{\varphi}_N \leftarrow [\varphi_0(x), \varphi_1(x) \dots \varphi_{\tilde{N}}(x)]\ \forall x \in \mathcal{X}$
2: Instanciate learner $\mathcal{L} \leftarrow$LinBand(arms= $\boldsymbol{\varphi}_N$, $S = S, \delta = \delta$)
3: for $t \in$ Range($T$) do
4:     $x_t \leftarrow \mathcal{L}$.select arm()
5:     Receive reward $r_t$
6:     $\mathcal{L}$.update($r_t$)
7: end for
```

Theorem 2. ***(Informal)*** *Let $f : [-1, 1] \to \mathbb{R}$ be a measurable function. If f is analytic, then,*

$$\left\| \sum_{n=0}^{N} a_{L,n} \varphi_{L,n} - f \right\|_{\infty} = \mathcal{O}(N^{\beta} \rho^{\alpha - N}),$$

for some $\alpha, \beta > 0$ and $\rho > 1$. The same result holds for the Fourier basis if $f \in \mathcal{C}^s_{per}([-1, 1])$.

For the formal statement of these results, see the appendix B.[2] Note that all the results of this section are valid in case $d = 1$. For higher dimensional spaces, we need a different strategy, as no result exists in the literature concerning orthogonal polynomials in more than one dimension. We discuss this technical difficulty in the proof of the main theorems.

4 Algorithms

This section presents our algorithm designed to address the continuous armed bandit setting with smooth objective functions. Our algorithm is presented in two variations, each with a different focus. The first variation is tailored for practical performance, while the second variation emphasizes theoretical guarantees.

The idea of both algorithms is to use the feature maps defined in the previous section to convert our continuous bandit problem into a linear bandit one. Precisely, consider the objective of the bandit algorithm, which is to play arms with high reward to minimize the regret. We have

$$\arg\max_{x \in \mathcal{X}} f(x) \approx \arg\max_{x \in \mathcal{X}} \sum_{n=0}^{N} a_n \varphi_n(x) = \arg\max_{x \in \mathcal{X}} \langle \mathbf{a}_N, \boldsymbol{\varphi}_N(x) \rangle, \tag{1}$$

where $\mathbf{a}_N$ stands for $[a_0, \dots a_N]$, the vector of the first N coefficients of the projection of the function in the vector space generated by the first N basis functions. In fact, what we do in our abstract algorithm 1 is exactly (line 1) to apply a given feature map $\boldsymbol{\varphi}_N$ on $\mathcal{X}$ and then use a generic linear bandit algorithm **LinBand** to choose the actions. The next subsections are devoted to choosing which linear bandit algorithm to use. A first solution will be to use the celebrated `LinUCB` (defined in (Abbasi-Yadkori et al., 2011) as OFUL, the name `LinUCB` was given later (Lattimore & Szepesvári, 2020) chapter 19). This will originate an elegant and practical algorithm, `OB-LinUCB`. To achieve the optimal regret guarantee, we will use a linear bandit algorithm called `phased elimination` Lattimore et al. (2020) falling in the field of *misspecificated linear bandits* (Lattimore et al., 2020; Ghosh et al., 2017). This will originate another version of our algorithm that we call `OB-PE` (see subsection 4.2).

4.1 OB-LinUCB

If, in equation 1 we had a perfect equality instead of the "≈" symbol, the problem would be solved by standard bandit algorithms such as `LinUCB` (defined in (Abbasi-Yadkori et al., 2011) as OFUL, the name `LinUCB` was given later (Lattimore & Szepesvári, 2020) chapter 19), taking as hidden

[2] For the theory behind these results, refer to (Wang & Xiang, 2012; Butzer et al., 1977; Katznelson, 2004).

vector $\mathbf{a}_N$, and as space of arms the subset of $\mathbb{R}^N$ given by $\{\boldsymbol{\varphi}_N(x) : x \in \mathcal{X}\}$. Instead, in this case, we have an approximation error since we are truncating the sum to N, thus falling in the field of *misspecificated linear bandits* (Lattimore et al., 2020; Ghosh et al., 2017). Nonetheless, if the misspecification is very small, solving the problem using LinUCB rather than a specific algorithm for misspecified linear bandits is still more convenient.

For clarity, we call this algorithm, obtained by plugging `LinUCB` in line 1, `OB-LinUCB`. The parameter $\widetilde{N}$, which indicates the dimension of both the unknown vector $\mathbf{a}_N$ and of the feature map $\boldsymbol{\varphi}_N$, is strictly liked with N, which corresponds to the *degree* of the feature map. If $d = 1$, the two numbers coincide, while in the multidimensional case, their relation is described in section 3.1. Note that `LinUCB` requires an upper bound S on the two norm of $\mathbf{a}_N$. Having assumed $\{\varphi_n\}_n$ to be an orthogonal sequence in L^2 reveals crucial in this case. Indeed, from Parseval's theorem (Rudin, 1974)

$$\|\mathbf{a}_N\|_2 \leq \sqrt{\sum_{n=0}^{\infty} a_n^2} = \|f\|_{L^2}.$$

This result answers a natural question: *why do we need the feature map to be an orthogonal basis of L^2?* In appendix E.1, we substantiate this question, also providing empirical evidence for the answer. Since $N < \infty$, we will incur a misspecification, which, for any $x \in \mathcal{X}$, corresponds to $\varepsilon(x) = f(x) - \langle \mathbf{a}_N, \boldsymbol{\varphi}_N(x)\rangle$. Nonetheless, we will show that this algorithm is able to perform very well in practice.

4.2 OB-PE

`OB-LinUCB` cannot achieve competitive regret since it does not explicitly handle the misspecification. For this reason, we propose a different choice for the linear bandit algorithm to be chosen as **`LinBand`**: the algorithm `phased elimination`, presented in Lattimore et al. (2020). With this modification, the algorithm can achieve a better regret guarantee, even if the misspecification is not negligible. We refer as `OB-PE` to the algorithm obtained by plugging `phased elimination` in Algorithm 1.

One feature of `phased elimination` Lattimore et al. (2020) is worth mentioning. This algorithm imposes that the number of arms is $k < \infty$, and its regret grows as $\log(k)$. Even if our set of arms $\boldsymbol{\varphi}_N \leftarrow [\varphi_0(x), \varphi_1(x) \dots \varphi_{\widetilde{N}}(x)]\ \forall x \in \mathcal{X}$ is uncountable, this does not represent an issue. Indeed, we can cover it by balls of radius $T^{-1/2}$ and preserve the same regret guarantee. This holds for two reasons: being f Lipschitz continuous, making an $T^{-1/2}-$cover of $\mathcal{X}$ allows to retain an $LT^{-1/2}-$suboptimal arm. This translates in an additive term $+L\sqrt{T}$ on the regret, which is negligible with respect to the main part. The second reason is that k grows as $\left(T^{1/2}\right)^d$, so that the multiplicative term on the regret corresponds to $(d/2)\log(T)$, which is also negligible.

5 Main results

In this section, we present the proofs of the main results regarding the regret bounds for algorithm 1 in different scenarios. We will always assume that the noise η_t is i.i.d. and $\sigma-$subgaussian $\|f\|_\infty \leq 1$ (one can be replaced by any constant by just rescaling). Both assumption are ubiquitous in the literature, and they strictly include the assumption to have an upper bound for $\|f\|_{L^2}$ done in algorithm 1, as by Hölder's inequality $\|f\|_{L^2} \leq 2\|f\|_\infty \leq 2$.

5.1 Regret bound for $d = 1$

The first theorem provides a regret guarantee for algorithm `OB-PE`, which is optimal if $\mathcal{X} = [-1, 1]$. The same does not hold for higher dimension spaces since proof techniques are slightly different, as we shall see in the next subsection.

Theorem 3. *Fix $\delta > 0$ and assume that $f \in \mathcal{C}^s([-1, 1])$ and its $s + 1-$th derivative is square-integrable. With probability at least $1 - \delta$, algorithm `OB-PE`, when instantiated with Legendre or*

Fourier[3] *feature maps and* $N = T^{\frac{1}{2s+1}}$ *achieves regret*

$$R_T \leq \log(1/\delta)\, \widetilde{\mathcal{O}}(T^{\frac{s+1}{2s+1}}).$$

For the proof, see Appendix C.1. Taking $\delta = T^{-1}$, it follows that our algorithm has an expected regret that is bounded by $\widetilde{\mathcal{O}}(T^{\frac{s+1}{2s+1}})$. In is worth mentioning that the previous result hides in the $\widetilde{\mathcal{O}}(\cdot)$ notation constants depending on f, as in is for every algorithm for the continuous bandit setting. For Zooming, the regret depends on the Lipschitz constant of the function, for UCB-Meta-Algorithm on the Holder constant corresponding to the maximal degree of differentiability. For GP methods, the dependence is on the norm in the RKHS. This dependence on f is unavoidable: since $\mathcal{C}^s(\mathcal{X})$ functions are dense in $\mathcal{C}^0(\mathcal{X})$, if it were possible to have a universal bound of order $T^{\frac{s+1}{2s+1}}$ valid for every $s-$times differentiable function without function-dependent constants, it would also be possible to extend the regret bound to the whole space $\mathcal{C}^0(\mathcal{X})$.

5.2 Regret bound for $d > 1$

In the more general case of $\mathcal{X} = [-1, 1]^d$, we can prove another regret bound for algorithm `OB-PE`. Still, as we shall see, this bound is slightly suboptimal.

Theorem 4. *Fix $\delta > 0$ and assume that $f \in \mathcal{C}^s([-1, 1]^d)$. With probability at least $1 - \delta$, algorithm* `OB-PE`*, when instantiated with multivariate Legendre feature map (see 3.1) and $N = T^{\frac{d}{2s}}$ achieves regret*

$$R_T \leq \log(1/\delta)\, \widetilde{\mathcal{O}}(T^{\frac{2s+d}{4s}}).$$

This result has two relevant drawbacks. Firstly, it only holds for Legendre feature map, and not for Fourier ones. Second, its regret bound of order $T^{\frac{2s+d}{4s}}$ becomes vacuous when $d > 2s$. Still, this result is close to the known lower bound for the setting, telling that the regret cannot be smaller that $\Omega(T^{\frac{s+d}{2s+d}})$, so that in the regime $d > 2s$, even the optimal regret is very close to $\mathcal{O}(T)$. The reason why our regret is not optimal in this case are very deep, and we investigate them in the appendix E.2.

Case of f infinitely differentiable. The previous results apply in the case of a smooth function of an arbitrary but finite degree s. It is also interesting to study the case of infinitely differentiable functions, for which we give a motivating example in Appendix D. In this case, we can achieve the best possible regret $\sqrt{T}$, up to logarithmic terms.

Theorem 5. *Fix $\delta > 0$ and assume that $f \in \mathcal{C}^\infty(\mathcal{X})$, being also analytic with convergence radius $1 + \rho$ for some $\rho > 0$. Then, with probability at least $1 - \delta$, algorithm* `OB-PE`*, when instantiated with multivariate Legendre feature map and $N = \log(T)^{\log(1+\rho)^{-1}}$, satisfies*

$$R_T \leq \log(1/\delta)\, \widetilde{\mathcal{O}}(\sqrt{T}).$$

This result imposes a stronger condition w.r.t. the fact of being just infinitely differentiable. In fact, it is required that the reward function f is analytic in an open set containing $\mathcal{X}$. In fact, it could happen, for a $\mathcal{C}^\infty(\mathcal{X})$ function that is not analytic, that the approximation error relative to an N degree feature map decreases more than polinomially but less than exponentially. Examples of functions of this kind are for example $N^{-\log(N)}$.

5.3 Comparison with state of the art

In this section, we aim at comparing the result of this paper with the state of the art for continuous armed bandits. As anticipated in the introduction, there are many algorithms in the literature that are focused on the continuous-armed bandit problem, with many different ideas coming from different fields. As comparison, we have chosen the ones which achieve the best results for either regret or computational complexity, or are particularly popular. From the literature about Lipschitz bandits, we have chosen the celebrated `Zooming` (Kleinberg et al., 2008) and `UCB-Meta-Algorithm` (Liu et al., 2021), which achieves optimal regret bound. From the literature of Bayesian optimization, we

[3] in case we use Fourier basis, periodicity at the boundary of the interval

Algorithm	$R_T(\mathcal{C}^s, \mathbb{R}^1)$	$R_T(\mathcal{C}^s, \mathbb{R}^d)$	$R_T(\mathcal{C}^\infty, \mathbb{R}^d)$	Complexity
Zooming	$T^{2/3}$	$T^{\frac{d+1}{d+2}}$	$T^{\frac{d+1}{d+2}}$	$k^2 T$
UCB-Meta-algorithm	$T^{\frac{s+1}{2s+1}}$	$T^{\frac{s+d}{2s+d}}$	N/A	kT^2
IGP-UCB	$T^{\frac{s+3/2}{2s+1}}$	$T^{\frac{s+3d/2}{2s+d}}$	$T^{1/2}$	$T^4 + kT^3$
BPE	$T^{\frac{s+1}{2s+1}}$	$T^{\frac{s+d}{2s+d}}$	$T^{1/2}$	$T^4 + kT^3$
BBKB	$T^{\frac{s+3/2}{2s+1}}$	$T^{\frac{s+3d/2}{2s+d}}$	$T^{1/2}$	$T^{\frac{4s+5d}{2s+d}} + kT^{\frac{4s+4d}{2s+d}}$
mini-META	$T^{\frac{s+3/2}{2s+1}}$	$T^{\frac{s+3d/2}{2s+d}}$	$T^{1/2}$	$T^2 + kT^{\frac{2s+4d}{2s+d}}$
OB-PE (ours)	$T^{\frac{s+1}{2s+1}}$	$T^{\frac{s+d/2}{2s}}$	$T^{1/2}$	$T^{\frac{3d}{2s}} + kT^{\frac{d}{s}}$
Lower bounds	$\Omega(T^{\frac{s+1}{2s+1}})$	$\Omega(T^{\frac{s+d}{2s+d}})$	$\Omega(T^{1/2})$	N/A

Table 1: Algorithms and their theoretical performance. In the regret guarantees and in the computational complexity columns we have omitted the $\widetilde{\mathcal{O}}(\cdot)$ notation for readability.

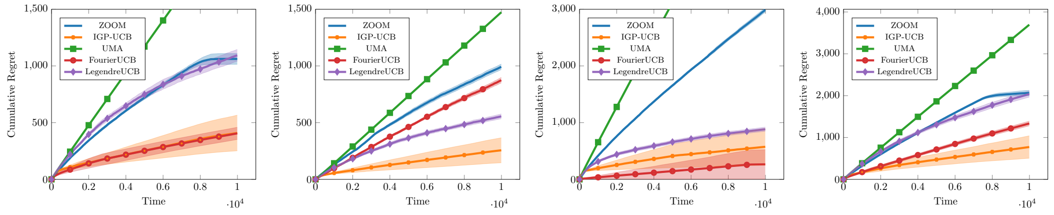
have chosen IGP-UCB(Chowdhury & Gopalan, 2017), which gives the most optimized version of the celebrated GP-UCB algorithm and taken BPE (Li & Scarlett, 2022), which achieves optimal regret. Lastly, we have also considered two very recent algorithm BBKB and mini-META (Calandriello et al., 2020; 2022) from the field of Bayesian optimization which try to improve the computational efficiency of classical methods. We summarize the theoretical guarantees of these algorithms in Table 1. It is important to clarify some aspects that cannot be represented in the table.

Assumptions on the regret guarantees Not all the assumptions on the regret are specified in the table. The algorithms from the literature of Bayesian Optimization (IGP-UCB, BPE, BBKB, mini-META), assume that the reward function f belongs to an RKHS with kernel given by either the RBF (Gaussian) kernel or the Matérn one. To obtain the regret bound, we have used the well-known fact that the $\nu-$Matérn kernel contains functions that are $\lceil \nu \rceil$ times differentiable in the L^2 sense. However, this does *not* mean that the algorithm has a regret guarantee for any $\mathcal{C}^s$ function. Also our algorithm, for the case $R_T(\mathcal{C}^s, \mathbb{R}^1)$ requires to assume that the reward function has an $s+1-$th derivative which is square-integrable.

Computation of the time complexity In order to have a fair comparison between the algorithms, we have fixed some aspects that are shared. First, in the table we have reported, for each algorithm, the total time complexity to perform all the length T episode. Since we are in continuous space, we need to assume that some form of discretization, otherwise it is impossible to choose the candidate arm at each round. We have assumed that all algorithms start from the same discretization of $\mathcal{X}$ which is composed of k elements. Lastly, note that the complexities of BBKB, mini-META and OB-PE (ours) may seem to explode for $d \gg s$ while in fact this regime corresponds to an unfeasible region where the algorithms make linear regret.

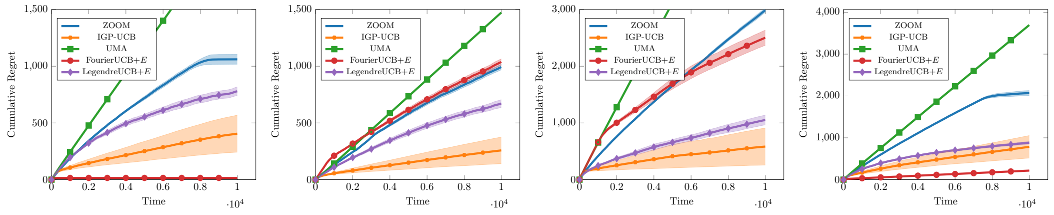
Comparing our algorithm with the state of the art in the family of GP bandits, we can see that our OB-PE enjoys the same regret guarantees of the best of this family BPE, except for the case of a reward function on $\mathcal{X} = \mathbb{R}^d$ with $s < +\infty$, where our algorithm is slightly worse. Still, OB-PE wins with margin on computational complexity, where it is able to outperform even BBKB and mini-META.

UCB-Meta-algorithm (Liu et al. (2021)), unlike those based on Gaussian Processes, works under our same set of assumptions. As we have seen, this algorithm is able to outperform our regret guarantee in case of a reward function on $\mathcal{X} = \mathbb{R}^d$ with $s < +\infty$ but does not have a good regret guarantee in the $s = +\infty$ case, since their approach would require to make a linear bandit of infinite dimension. Furthermore, our algorithm outperfors UCB-Meta-algorithm on the computational side by a significant margin. Finally, OB-PE is arguably much simpler, as it requires to just build one global feature map instead of many local ones (a number which depends on T). This inherent simplicity enhances the algorithm's interpretability, allowing for the incorporation of domain knowledge. For instance, leveraging the fact that Legendre polynomials of even order are even functions, we can intelligently reduce the dimension of the feature map when we know that the objective function f follows even or odd patterns, showcasing the flexibility of OB-PE.



(a) (b) (c) (d)

Figure 1: Regret plots of the algorithms in four environments (base version)



(a) (b) (c) (d)

Figure 2: Regret plots of the algorithms in **the same** four environments (even version)

Overall, algorithm OB−PE is competitive with the state of the art under all aspects. We conclude this comparison with experiments over synthetic data to validate the considerations of this section.

6 Experiments

To test the algorithms introduced in this paper, we have performed numerical simulations showing their performance compared to some baselines in the literature on continuous armed bandit environments. The details of the experiments are shown in Appendix F. Here we limit ourselves to a brief description of the results.

Setting The various environments are characterized by the reward function f since the noise added is always Gaussian with zero mean and unit variance. The choice of f, from left to right, corresponds to (a) a Gaussian density function with $\sigma = 0.3$, (b) An even polynomial of degree 4, (c) A product between a $\sin$ function and a polynomial (not even neither odd), (d) a triangular window function on the interval $[-1/2, 1/2]$. This function is piecewise linear and continuous, but not continuously differentiable. As baselines, both in Figures 1 and 2, we put the same algorithms of the literature appearing in the Table 1; ZOOM standing for Zooming, IGP−UCB for IGP−UCB and UMA for UCB−Meta−algorithm. As a comparison, in Figure 1, we have OB−LinUCB with the two sets of orthogonal functions, with the name FourierUCB and LegendreUCB. Instead, in Figure 2, we are showing the performance of the three algorithms when we only use *even* basis functions. This is done to test the flexibility of our algorithms in case they receive some information on the function f we are optimizing, as it happens very often in practice due to the presence of domain knowledge. In experiment (c), to test what happens when the algorithms receive a wrong advice, we consider a function f that is not even.

Results As we can see, the best-performing baseline is always IGP−UCB. This result comes with no surprise: this method is known to give extremely good results in practice. On the other side, UMA turns out to be the worst algorithm in every setting, despite having the best theoretical guarantees. To ensure the truthfulness of this result, we have conducted an extensive hyperparameter tuning of this baseline F.3, showing that even with the best parameters, the performance is not satisfactory. This can be explained by the fact that this algorithm comes from a very theoretical paper and was designed to prove a regret bound rather than to be valid in practice.

Table 2: Comparison of computation times for experiments on Environment (a)

Algorithm	ZOOM	IGP-UCB	UMA	FourierUCB	LegendreUCB
Time (s)	3.9	14957.3	108.1	3.2	3.4

Coming to our algorithms, we can see that LegendreUCB has stable performance, always performing better than ZOOM and always worse than IGP-UCB. The most surprising algorithm is indeed FourierUCB. This algorithm has weaker theoretical guarantees, as it requires periodic conditions at the boundary. Formally, this condition is satisfied only by Environment (d) (which is $\mathcal{C}^0(\mathcal{X})$ and periodic) and partially by (b) (which is $\mathcal{C}^\infty(\mathcal{X})$, but only $\mathcal{C}^0_{per}(\mathcal{X})$ since the derivative is not continuous at the boundary). Nonetheless, this algorithm is able to outperform the powerful IGP-UCB in half of the environments, both in the standard case and in the case with even basis functions. Its performance is particularly good in Environment (a) of Figure 2. This can be explained by the fact that the Gaussian function with all its derivatives decreases to zero very quickly at the extremes of the interval. Therefore it can be well approximated by a $\mathcal{C}^\infty_{per}$ function with a value of 0 at the boundary.

Last but not least, it is important to consider the computational effort of the algorithms to perform the full experiment. In Table 2, we have reported the time to run all the experiments with Environment (a). This conclusion agrees with our predictions in Table 1: the different time complexity leads to very different orders of magnitude in the actual running time, with IGP-UCB being five thousand times slower than our algorithms.

7 Conclusions

In this paper, we have introduced a new method to face the continuous armed bandit problem when the reward curve f is s times differentiable. To this end, we proposed to project the reward curve on the subspace generated by the first elements of an orthogonal basis of $L^2(\mathcal{X})$ (Fourier, Legendre). This representation allows to reduce the continuous armed bandit problem to a linear bandit with misspecification. This problem can be solved by means of the meta-algorithm 1, from which we have introduced the two algorithms: OB-LinUCB and OB-PE. As we have shown, the former is more oriented to practice, being simple and computationally efficient, while the latter has a regret guarantee that is close to the lower bound for reward functions $f \in \mathcal{C}^s(\mathcal{X})$, being able to match it if either $d = 1$ or $s = +\infty$. Moreover, this algorithm enjoys state-of-the-art computational complexity, even surpassing algorithms specifically designed to be fast. Finally, OB-LinUCB is validated in simulated environments where it achieves performance competitive with the best baseline with a significantly lower computational effort.

Future works. In this paper we have presented an algorithm that bridges the gap between two families of methods, the one based on Gaussian processes and the one based on Lipschtzness/Hölder continuity. Despite having a strong similarity with UCB-Meta-algorithm, the proof techniques for the regret bounds are based on projecting an Hilbert space over a finite dimensional subspace, as the ones of Vakili et al. (2021) (the paper discovering the bound on γ_T allowing for optimal regret in Gaussian process bandits). Therefore, the most interesting question is whether it is possible to reduce the two huge family of algorithms to a meta-algorithm which is able to achieve ”best-of-both-worlds” performance.

References

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

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

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

Thomas Bagby, Len Bos, and Norman Levenberg. Multivariate simultaneous approximation. *Constructive approximation*, 18(4):569–577, 2002.

Harold P Boas. Lecture notes on several complex variables, 2012.

PL Butzer, H Dyckhoff, E Görlich, and RL Stens. Best trigonometric approximation, fractional order derivatives and lipschitz classes. *Canadian Journal of Mathematics*, 29(4):781–793, 1977.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory*, pp. 533–557. PMLR, 2019.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Near-linear time gaussian process optimization with adaptive batching and resparsification. In *International Conference on Machine Learning*, pp. 1295–1305. PMLR, 2020.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Scaling gaussian process optimization by evaluating a few unique candidates multiple times. In *International Conference on Machine Learning*, pp. 2523–2541. PMLR, 2022.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pp. 844–853. PMLR, 2017.

Nando de Freitas, Alex Smola, and Masrour Zoghi. Regret bounds for deterministic gaussian process bandits. *arXiv preprint arXiv:1203.2177*, 2012.

Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Johannes Hahn (https://math.stackexchange.com/users/62443/johannes hahn). When does a multivariate power series define an entire function? Mathematics Stack Exchange. URL `https://math.stackexchange.com/q/2650950`. URL:https://math.stackexchange.com/q/2650950 (version: 2022-08-29).

Matthew O Jackson, Tomas Rodriguez-Barraquer, and Xu Tan. Epsilon-equilibria of perturbed games. *Games and Economic Behavior*, 75(1):198–216, 2012.

Yitzhak Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, 17, 2004.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 681–690, 2008.

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

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*, pp. 5662–5670. PMLR, 2020.

Zihan Li and Jonathan Scarlett. Gaussian process bandit optimization with few batches. In *International Conference on Artificial Intelligence and Statistics*, pp. 92–107. PMLR, 2022.

Yusha Liu, Yining Wang, and Aarti Singh. Smooth bandit optimization: generalization to holder space. In *International Conference on Artificial Intelligence and Statistics*, pp. 2206–2214. PMLR, 2021.

Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.

Walter Rudin. Real and complex analysis, mcgraw-hill. *Inc.,*, 1974.

Bernhard Schölkopf. The kernel trick for distances. *Advances in neural information processing systems*, 13, 2000.

Shubhanshu Shekhar and Tara Javidi. Gaussian process bandits with adaptive discretization. 2018.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 82–90. PMLR, 2021.

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

Haiyong Wang and Shuhuang Xiang. On the convergence rates of legendre approximation. *Mathematics of computation*, 81(278):861–877, 2012.

A Index of the notations

In this section, we leave, for the reader's convenience, a table of the notations introduced in this paper.

$\mathcal{X}$	Space of arms
d	Dimension of $\mathcal{X}$ in the sense of vector spaces
T	Time horizon
T	Cumulative regret
f	The unknown reward function
$\mathcal{H}$	Generic Hilbert space
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Scalar product in the Hilbert space $\mathcal{H}$
$L^2(\mathcal{X})$	Space of square-integrable functions over $\mathcal{X}$
$\varepsilon(x)$	Misspecification evaluated in x
η_t	Random noise
σ	Subgaussianity constant of the noise
$\boldsymbol{\varphi}_N$	generic feature map of degree N
φ_n	$n-$th element of a generic feature map $\boldsymbol{\varphi}_N$
a_n	Scalar product between f and φ_n
$\mathbf{a}_N$	Vector of the coefficients relative to feature map $\boldsymbol{\varphi}_N$
$\boldsymbol{\varphi}_{F,N}$	Fourier feature map of degree N (1 dimension)
$\boldsymbol{\varphi}_{L,N}$	Legendre feature map of degree N (1 dimension)
$\boldsymbol{\varphi}^d_{F,N}$	Fourier feature map of degree N (d dimensions)
$\boldsymbol{\varphi}^d_{L,N}$	Legendre feature map of degree N (d dimensions)
N	Degree of a feature map
$\widetilde{N}$	Lenght of a feature map ($= N$ if $d = 1$)
$\mathcal{C}^s(\mathcal{X})$	$s-$times differentiable functions over $\mathcal{X}$
$\mathcal{C}^s_{per}([-1, 1])$	$s-$times differentiable periodic functions on $[-1, 1]$

B Formal statement of the Decay properties

The following theorems can be found in Quarteroni et al. (2010) Page 348 formula (9.30).

Theorem 6. *Note $\{\varphi_{L,n}\}_n$ as the Legendre feature map. Let $f : [-1, 1] \to \mathbb{R}$ be a measurable function. If $f \in \mathcal{C}^s([-1, 1])$, and $f^{(s+1)} \in L^2([-1, 1])$,*

$$\left\| \sum_{n=0}^{N} a_{L,n}\varphi_{L,n} - f \right\|_\infty = C\|f\|_s N^{-s-1/2},$$

where C is a universal constant, while $\|f\|_s = \sum_{k=0}^{s+1} \|f^{(k)}\|_{L^2}$.

Note that in the previous theorem, the continuity of the $s + 1$ derivative is not required, and in principle not even its boundedness. A similar result applies to the Fourier basis if we assume periodic conditions at the boundaries (Quarteroni et al. (2010) Page 365).

Theorem 7. *Let $f : [-1, 1] \to \mathbb{R}$ be a measurable function. If $f \in \mathcal{C}^s_{per}([-1, 1])$ (space of $s-$times differentiable functions which are periodic at the boundary), and $f^{(s+1)} \in L^2([-1, 1])$,*

$$\left\| \sum_{n=0}^{N} a_{F,n}\varphi_{F,n} - f \right\|_\infty = C\|f\|_s N^{-s-1/2},$$

where C is a universal constant, while $\|f\|_s = \sum_{k=0}^{s+1} \|f^{(k)}\|_{L^2}$

In the case of $f \in \mathcal{C}^\infty([-1,1])$, the bounds given in the previous theorems vanish more than polynomially in n, meaning that for any positive k we have the following convergence property

$$\frac{\left\|f - \sum_{n=0}^{N} a_n \varphi_n\right\|_\infty}{N^{-k}} \to 0.$$

However, it should be noted that this result does not directly guarantee exponential decay in N, as we could encounter scenarios where the decay follows a function such as $N^{-\log(N)}$. To achieve exponential decay, it is necessary to assume analyticity of the function.

Theorem 8. *Wang & Xiang (2012) Let, $f : [-1,1] \to \mathbb{R}$ be a function that admits an analytic extension on the complex ellipse $\mathcal{E}_\rho$ given by*

$$\mathcal{E}_\rho := \left\{ z \in \mathbb{C} : z = \frac{u + u^-1}{2},\ |u| = \rho > 1 \right\}.$$

Then, the approximation with Legendre feature map satisfies

$$\|f - \sum_{n=0}^{N} a_n \varphi_n\|_\infty \leq 2\sqrt{\frac{\rho^2 + \rho^-2}{\rho^2 - 1}} \|f\|_\infty N^{1/2} \rho^{1/2-N}.$$

A similar result holds for the approximation with Fourier feature map.

Theorem 9. *Katznelson (2004)(Exercise I.4.4) If f is periodic and analytic, then there are $K(f), \beta(f) > 0$ such that*

$$a_{n,F} \leq K(f) e^{-\beta(f)n}.$$

From this theorem, a straightforward corollary follows by summing all the terms $a_{F,n}$.

Corollary 10. *If f is periodic and analytic, there are $K(f), \beta(f) > 0$ such that*

$$\left\| f - \sum_{n=0}^{N} a_{n,F} \varphi_{n,F} \right\|_\infty \leq K(f) \frac{e^{-\beta(f)N}}{1 - e^{-\beta(f)}}.$$

This result guarantees that we can achieve exponentially accurate approximation using Fourier basis functions when the function is analytic and periodic.

B.1 Approximation theory for multivariate functions

Regarding the case of Legendre and Fourier feature maps, there is currently no result ensuring that the approximation of f using the usual coefficients $a_{L,n} := \langle f, \varphi_{L,n} \rangle_{L^2}$ enjoys particular decay properties. Still, there are results showing that smooth functions can be well approximated by multivariate polynomials. The following result is a particular case of Theorem 1 by Bagby et al. (2002).

Theorem 11. *Let $f \in \mathcal{C}^s([-1,1]^d)$. Then, for every $N > 0$, there is a polynomial p_N of degree at most N such that*

$$\|f - p_N\|_\infty \leq C(f) N^{-s},$$

where $C(f)$ is a constant only depending on f.

A stronger result holds for infinitely differentiable functions.

Theorem 12. *Let $f \in \mathcal{C}^\infty(\mathcal{X})$, be analytic with convergence radius $1 + \rho$ for some $\rho > 0$. Then, for every $N > 0$, there is a polynomial of degree at most N such that*

$$\|f - p_N\|_\infty \leq \frac{C(f)(1+\rho)^{-N}}{\rho},$$

where $C(f)$ is a constant only depending on f.

Proof. Being f analytic, we know that on the domain $\{x : \|x\|_\infty \leq 1 + \rho\}$ we have

$$f(x) = \sum_{n=0}^{\infty} \sum_{|\alpha|=n} a_\alpha x^\alpha,$$

where α is one $d-$dimensional multi index with degree n. Since the convergence radius is $1 + \rho$, we know by root test (Boas, 2012)[4] that

$$(1 + \rho)^{-1} = \limsup_n \left(\sum_{|\alpha|=n} |a_\alpha| \right)^{1/n}.$$

Using this result, there is a constant $C(f) < \infty$ such that $\sum_{|\alpha|=n} |a_\alpha| < C(f)(1 + \rho)^{-n}$. Therefore, we have

$$\begin{aligned}\left\| f(x) - \sum_{n=0}^{N} \sum_{|\alpha|=n} a_\alpha x^\alpha \right\|_\infty &= \left\| \sum_{n=N}^{\infty} \sum_{|\alpha|=n} a_\alpha x^\alpha \right\|_\infty \\ &\leq C(f) \sum_{n=N}^{\infty} (1 + \rho)^{-n} \sup_{x \in [-1,1]^d} \|x\|_\infty^n \\ &\leq \frac{C(f)(1 + \rho)^{-N}}{\rho}.\end{aligned}$$

Therefore, the $N-$degree polynomial $\sum_{n=0}^{N} \sum_{|\alpha|=n} a_\alpha x^\alpha$ satisfies the thesis.

□

C Missing proofs

In this section, we provide the proof for the regret bounds of the main paper, relative to algorithm `OB-PE` under different assumptions on the reward function.

C.1 Regret bound for `OB-PE` in the $1-$dimensional case

We now proceed to establish a regret bound for our second algorithm, `OB-PE`, which achieves a superior theoretical performance in terms of regret.

Theorem 3. *Fix $\delta > 0$ and assume that $f \in \mathcal{C}^s([-1, 1])$ and its $s + 1-$th derivative is square-integrable. With probability at least $1 - \delta$, algorithm `OB-PE`, when instantiated with Legendre or Fourier[5] feature maps and $N = T^{\frac{1}{2s+1}}$ achieves regret*

$$R_T \leq \log(1/\delta)\, \widetilde{\mathcal{O}}(T^{\frac{s+1}{2s+1}}).$$

Proof. Due to the fact that in `OB-PE` is based on algorithm `phased elimination`, presented in Lattimore et al. (2020), we have to cover the space $[-1, 1]$ to get a finite number of arms.

We can prove that starting form a discretization of the space $[-1, 1]$ in $\lceil \sqrt{T} \rceil$ discrete arms has no effect on the regret.

$$\begin{aligned} R_T &= \sum_{t=1}^{T} \max_{x \in [-1,1]} f(x) - f(x_t) \\ &= \sum_{t=1}^{T} \max_{x \in [-1,1]} f(x) - \max_{j=1,\dots\lceil \sqrt{T} \rceil} f(x_j) + \max_{j=1,\dots\lceil \sqrt{T} \rceil} f(x_j) - f(x_t). \end{aligned}$$

[4] see also this discussion (https://math.stackexchange.com/users/62443/johannes hahn)

[5] in case we use Fourier basis, periodicity at the boundary of the interval

Having assumed smoothness, we have also that f is $L-$Lipschitz continuous for some $L > 0$. Therefore, the first part $\sup_{x\in[-1,1]} f(x) - \sup_{j=1,\dots\lceil\sqrt{T}\rceil} f(x_j)$ will be bounded by $LT^{-1/2}$. This results in

$$\begin{aligned} R_T &\leq T \cdot LT^{-1/2} + \sum_{t=1}^{T} \max_{j=1,\dots\lceil\sqrt{T}\rceil} f(x_j) - f(x_t) \\ &= \underbrace{LT^{1/2}}_{\in\widetilde{\mathcal{O}}(\sqrt{T})} + \sum_{t=1}^{T} \max_{j=1,\dots\lceil\sqrt{T}\rceil} f(x_j) - f(x_t). \end{aligned}$$

Therefore, from now on, we will assume without loss of generality that the best arm belongs to our discretization $j = 1, \dots\lceil\sqrt{T}\rceil$. At this point, the algorithm requires to use `phased elimination` with a representation of the arms given, for every $j = 1, \dots\lceil\sqrt{T}\rceil$, by

$$\boldsymbol{\varphi}_N(x_j) \leftarrow [\varphi_0(x_j), \varphi_1(x_j) \dots \varphi_N(x_j)].$$

Therefore, Proposition 5.1. by Lattimore et al. (2020) ensures the following high probability regret bound. For each $\delta > 0$, we have in the case of k arms, with probability at least $1 - \delta$,

$$R_T \leq \log\left(\frac{1}{\delta}\right) \widetilde{\mathcal{O}}(\sqrt{NT}\log(k) + \varepsilon\sqrt{N}T), \tag{2}$$

where ε is an upper bound on the misspecification, which in our case can be computed as follows:

$$\begin{aligned} \inf_{\boldsymbol{\theta}\in\mathbb{R}^N} \|f(x) - \langle\boldsymbol{\theta}, \boldsymbol{\varphi}_N(x)\rangle\|_\infty &\leq \left\| f(x) - \sum_{n=0}^{N} a_n\varphi_n(x) \right\|_\infty \\ &\leq C\|f\|_s N^{-s-1/2} \\ &= C\|f\|_s T^{\frac{-s-1/2}{2s+1}} = C\|f\|_s T^{-\frac{1}{2}}. \end{aligned}$$

where the second passage comes from theorem 6 (theorem 7 in case of Fourier basis, and C is the universal constant there defined) and the third from the definition $N = T^{\frac{1}{2s+1}}$. Substituting this value to the maximal misspecification ε, in equation equation 2 the regret is bounded with probability $1 - \delta$ by

$$R_T \leq \log\left(\frac{1}{\delta}\right) \widetilde{\mathcal{O}}(\sqrt{NT}\log(k) + \sqrt{N}\sqrt{T}) = \log\left(\frac{1}{\delta}\right) \widetilde{\mathcal{O}}(T^{\frac{s+1}{2s+1}}),$$

where the last passage is valid since, due to our discretization the number of arms corresponds to $\lceil\sqrt{T}\rceil$, so that $\log(k) \approx \log(T)/2$. This step ends the proof. □

C.2 Regret bound for $d > 1$

In this section, we prove the main result for our algorithm in the multivariate case $\mathcal{X} = [-1, 1]^d$.

Theorem 4. *Fix $\delta > 0$ and assume that $f \in \mathcal{C}^s([-1, 1]^d)$. With probability at least $1 - \delta$, algorithm `OB-PE`, when instantiated with multivariate Legendre feature map (see 3.1) and $N = T^{\frac{d}{2s}}$ achieves regret*

$$R_T \leq \log(1/\delta)\, \widetilde{\mathcal{O}}(T^{\frac{2s+d}{4s}}).$$

Proof. We start from the proof of the result for one dimension, changing only what is needed.

1. In order have apply the algorithm `phased elimination` and the desired regret bound, we need to make a $T^{-1/2}-$cover of the state space, so that the number of actions becomes

finite. In the one dimensional case, this cover contained $\lceil\sqrt{T}\rceil$ points, while here we need $k = \lceil\sqrt{T}\rceil^d$. Still, since this number enters in the regret just as $\log(k)$, this only makes it scale by a factor d.

2. In the multivariate case, the dimension $\widetilde{N}$ of the feature vector does not coincide with the degree of the polynomial obtained. In fact, to have a vectors which forming a basis for the vector space of degree $d-$variate polynomials of degree N, we need $\widetilde{N} = \binom{N+d}{N}$. This quantity is always bounded by N^d, which is much easier to compute.

3. In the multivariate case, we have no result about the decay property of the coefficients of Legendre polynomials. Therefore, we cannot bound $\|f(x) - \sum_{n=0}^{N} a_n\varphi_n(x)\|_\infty$ as done for the univariate case. Still, using `phased elimination` we are not forced to choose $\boldsymbol{\theta} = \mathbf{a}_N$, the vector of Legendre coefficients. Instead, we can rely on the following argument: due to theorem 11, we know that for every $N > 0$, there is a polynomial p_N of degree at most N such that

$$\|f - p_N\|_\infty \leq C(d)N^{-s}, \tag{3}$$

where $C(d)$ is a constant only depending on d. Therefore, since the feature map $\boldsymbol{\varphi}^d_{F,N}$ forms a basis for the vector space of all $d-$variate polynomials of degree at most N, this means that there is a vector $\boldsymbol{\theta}_*$ such that $\langle \boldsymbol{\varphi}^d_{L,N}(x), \boldsymbol{\theta}_*\rangle = p_N(x)$, the polynomial defined in equation equation 3. Then, we have

$$\begin{aligned}\inf_{\boldsymbol{\theta}\in\mathbb{R}^{\tilde{N}}} \|f(x) - \langle\boldsymbol{\theta}, \boldsymbol{\varphi}^d_{L,N}(x)\rangle\|_\infty &\leq \|f(x) - \langle\boldsymbol{\theta}_*, \boldsymbol{\varphi}^d_{L,N}(x)\rangle\|_\infty \\ &= \|f - p_N\|_\infty \\ &\leq C(d)N^{-s}.\end{aligned}$$

Once done these three modifications, the proof follows similarly: once chosen $N = T^{\frac{1}{2s}}$ we get $\widetilde{N} \leq N^d = T^{\frac{d}{2s}}$. Having proved that the maximal misspecification ε is bounded by $C(f)N^{-s}$, the regret is bounded with probability $1 - \delta$ by

$$\begin{aligned}R_T &\leq \log\left(\frac{1}{\delta}\right)\widetilde{\mathcal{O}}\left(\sqrt{\widetilde{N}T}\log(k) + \sqrt{\widetilde{N}}TC(f)N^{-s}\right) \\ &\leq \log\left(\frac{1}{\delta}\right)\widetilde{\mathcal{O}}\left(\sqrt{(T^{\frac{d}{2s}})T}\log(k) + (T^{\frac{d}{4s}})TC(f)T^{-\frac{1}{2}}\right) \\ &= \log\left(\frac{1}{\delta}\right)\widetilde{\mathcal{O}}(T^{\frac{2s+d}{4s}}).\end{aligned}$$

where the last passage is valid since, due to our discretization the number of arms corresponds to $\lceil\sqrt{T}\rceil^d$, so that $\log(k) \approx d\log(T)/2$. This step ends the proof.

□

C.3 Regret bounds in the case of $\mathcal{C}^\infty$ functions

Theorem 5. *Fix $\delta > 0$ and assume that $f \in \mathcal{C}^\infty(\mathcal{X})$, being also analytic with convergence radius $1 + \rho$ for some $\rho > 0$. Then, with probability at least $1 - \delta$, algorithm* `OB-PE`*, when instantiated with multivariate Legendre feature map and $N = \log(T)^{\log(1+\rho)^{-1}}$, satisfies*

$$R_T \leq \log(1/\delta)\,\widetilde{\mathcal{O}}(\sqrt{T}).$$

Proof. It is sufficient to repeat the same passages of the previous proof with the bound of the misspecification given by theorem 12. □

D APPLICATION: TREMBLING HAND PROBLEM

The smoothness assumption upon which our algorithm is built may appear overly restrictive. In practice, it becomes challenging to ensure that the function f in the continuous armed bandit problem exhibits a certain degree of smoothness. Hence, we propose a modified setting where the condition naturally holds.

Up until now, our assumption has been that by selecting an arm $x_t \in [-1, 1]$, we could observe a sample $f(x_t) + \eta_t$, where η_t represents a zero-mean noise independent of the past. However, a more realistic scenario involves an additional unobservable noise ζ_t (also i.i.d.) affecting the choice of the arm x_t. In fact, in a continuous state space, it is likely that the choice of actions is affected by some sort of random error due, for example, to measurement errors. A similar model is known in the field of game theory as the ”trembling hand problem” Jackson et al. (2012). Here, we make two key assumptions. First, we assume that the ’horizontal noise’ ζ_t follows a Gaussian distribution. Second, we assume that the function $f \in L^\infty_{per}([-1, 1])$, allowing for the evaluation of the function outside the interval $[-1, 1]$ through periodicity. This assumption is necessary due to the Gaussian nature of the noise ζ_t, which requires the function f to be defined on the entire space $\mathbb{R}$. Importantly, no specific smoothness assumptions are imposed on f.

In this setting, it is meaningless to compare to the fixed and noiseless best arm, since it would lead to linear regret because of the noise. We can instead compare the policy to the best arm when the noise is applied. This leads to the following definition of regret

$$R_T := \sum_{t=1}^{T} \sup_{x\in[-1,1]} \mathbb{E}_{\zeta\sim\mathcal{N}(0,\sigma^2)} f(x+\zeta) - f(x_t+\zeta_t) \qquad \zeta_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2).$$

This kind of problem can be reduced to a standard continuous bandit problem by just substituting the reward curve f with

$$\tilde{f} := f * \mathcal{N}(0, \sigma^2) = \int_0^\infty f(x-y)\frac{e^{-y^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma}\, dy.$$

In this way, the objective function f gets smoothed by the effect of the noise ζ; it is well known that the convolution between a $\mathcal{C}^\infty$ function, such as the Gaussian density function, and a bounded function, such as f, is $\mathcal{C}^\infty$ too. This allows us to prove that the Fourier coefficients a_n of $\tilde{f}$ decay exponentially fast. Moreover, in this particular case, it is possible to provide a more precise quantification of their decay.

Theorem 13. *The Fourier coefficients of $\tilde{f}$ are bounded as follows*

$$|a_n| \le \|f\|_{L^2} \times \begin{cases} e^{-\sigma^2 n^2/8} & n \text{ even} \\ e^{-\sigma^2 (n+1)^2/8} & n \text{ odd} \end{cases}$$

Proof. First of all, note that using the Fourier series in exponential form, we have

$$f(x) = \sum_{n=-\infty}^{\infty} b_n e^{inx},$$

for a sequence b_n, which respects Parseval’s identity:

$$\sum_{n=-\infty}^{\infty} b_n^2 = \|f\|^2_{L^2} < +\infty. \tag{4}$$

Now, note that

$$
\begin{aligned}
\tilde{f}(x) &= \int_{-\infty}^{\infty} f(x-y)\frac{e^{-y^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma}\, dy \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} b_n e^{in(x-y)} \frac{e^{-y^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma}\, dy \\
&= \sum_{n=-\infty}^{\infty} e^{inx} b_n \int_{-\infty}^{\infty} e^{-iny}\frac{e^{-y^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma}\, dy.
\end{aligned}
$$

Observe that the last term corresponds to the Fourier transform of $\frac{e^{-y^2/(2\sigma^2)}}{\sqrt{2\pi}}$ evaluated at n. Since the Fourier transform of $\frac{e^{-y^2/(2\sigma^2)}}{\sqrt{2\pi}}$ corresponds to $e^{-\sigma^2\xi^2/2}$, this term corresponds to $e^{-\sigma^2 n^2/2}$. Substituting, we get

$$
\begin{aligned}
\tilde{f}(x) &= \sum_{n=-\infty}^{\infty} b_n \int_{-\infty}^{\infty} e^{iny}\frac{e^{-\sigma^2 y^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma}\, dy \\
&= \sum_{n=-\infty}^{\infty} e^{-\sigma^2 n^2/2} b_n e^{inx}.
\end{aligned}
$$

This means that $b'_n = e^{-\sigma^2 n^2/2} b_n$ corresponds to the Fourier coefficients of $\tilde{f}$ in exponential form. By equation equation 4

$$b'_n \le e^{-\sigma^2 n^2/2}\|f\|_{L^2}.$$

To obtain a bound for the Fourier coefficients in sin-cosine form, it is sufficient to apply the trigonometric identities

$$\cos(x) = \frac{e^{ix}+e^{-ix}}{2} \qquad \sin(x) = \frac{e^{ix}-e^{-ix}}{2i}.$$

□

This fact leads to the conclusion that both algorithms OB-LinUCB and OB-MissLinUCB are able to achieve regret $\tilde{O}(\sqrt{T})$ regret in this case. To see this, it is sufficient to substitute the decay rate $e^{-\sigma^2 n^2/8}\|f\|_{L^2}$, which is faster than exponential, in the proof of theorem 5.

E Further considerations

In this section, we address some question of theoretical relevance, which we cannot insert in the main paper due to the limited space.

E.1 Does OB-LinUCB work with nonorthogonal features?

As linear bandit algorithms are designed to work whenever the reward can be written as a scalar product $r_t = \langle \boldsymbol{\theta}, \boldsymbol{\varphi}(x)\rangle + \eta_t$, one could do the following reasoning. Since the vector space generated by the first N Legendre polynomials $\varphi_{L,1}(x), \dots \varphi_{L,N}(x)$ corresponds to the one generated by the basis $1, x, \dots x^N$, we can use the former as feature vector and still achieve the same results.

This reasoning may seem correct, but in fact it is affected by a subtle problem. The LinUCB algorithm, in order to work properly, needs not only that the average rewards have the form $\langle \boldsymbol{\theta}, \boldsymbol{\varphi}(x)\rangle$, but also that an upper bound on $\|\boldsymbol{\theta}\|_2$ is known. While this assumption is natural in the linear bandit setting, in the continuous armed bandit one, our assumption only includes a bound for $\|f\|_\infty$, which is not necessary linked to a bound on $\|\boldsymbol{\theta}\|_2$. While in the main paper it is shown that, for an orthogonal feature map $\|\boldsymbol{\theta}\|_2 \le 2\|f\|_\infty$, the same result does not hold in general.

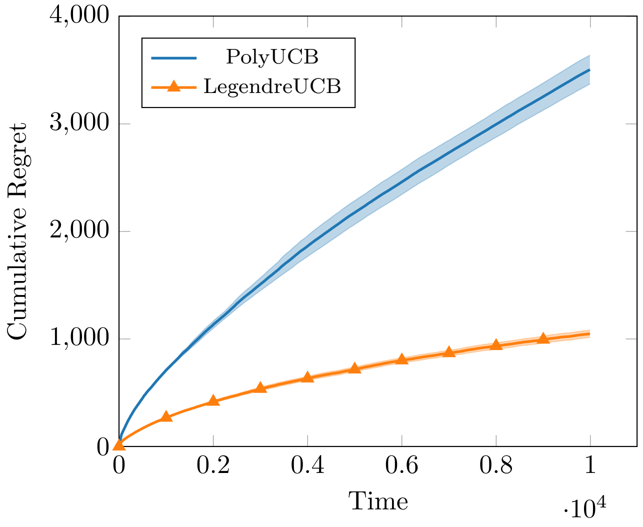


Figure 3: Regret curves of `LegendreUCB` and its trivial version `PolyUCB` which uses the standard polynomial basis instead of the Legendre basis.

Coming back to our specific case, where the feature map $\{1, x, \dots x^N\}$ is proposed, we can see clearly this phenomenon. In fact, there are functions such that $\|f\|_\infty = \sup_{x \in [-1,1]} |f(x)| \le 1$ but its corresponding $\boldsymbol{\theta}$ coefficient has huge norm. For example, consider the polynomial:

$$p(x) = -0.3652 + 12.3076x^2 - 64.974x^4 + 109.5056x^6 - 57.4752x^8.$$

Even having very big coefficients in the base formed by $\{1, x, \dots x^N\}$, it satisfies $\|p\|_\infty = 1$. Using this function as reward function, we can make an experiment to compare our `LegendreUCB` with the algorithm obtained by using the basis $\{1, x, \dots x^N\}$ in `OB-LinUCB`, which we call `PolyUCB`. In figure 3, we have shown the result of this experiment. As we can see, `OB-LinUCB` is able to completely outperform its competitor, and the reason stays in the fact that polynomial $p(x)$ can be represented as a linear combination of the elements of the Legendre basis with $\boldsymbol{\theta}$ is bounded by 2, while this does not hold for the standerd polynomial basis.

E.2 Connections with recent advances in Gaussian Process Bandits and tightness of the regret

Despite our approach resembling that of Liu et al. (2021), one of the most intriguing aspects of our analysis lies in its parallelism with one of the most significant papers in the Gaussian process bandit field, namely Vakili et al. (2021). At a high level, the idea behind our proofs is to project the subspace of s-times differentiable functions onto the vector space formed by fixed-degree algebraic or trigonometric polynomials. Similarly, Vakili et al. (2021) accomplishes, albeit in different terms, a projection of an RKHS onto the vector space generated by the first N eigenfunctions of the kernel.

At this point, both our work and theirs rely on analogous results: while we utilize the results from the theory of approximation B to bound the projection error, they employ the rate of decay of kernel eigenvalues. It is worth noting that, even though the space of s-times differentiable functions does not admit a kernel, we can view sequences of orthogonal functions as analogs of kernel eigenfunctions and the results on the decay of kernel eigenvalues as analogs of the decay properties of Fourier/Legendre coefficients B.

Now the question is as follows: if the results are analogous, how is it possible that in the case of Gaussian process bandits, the article by Vakili et al. (2021) enables achieving the optimal regret of the multidimensional case, namely $T^{\frac{s+d}{2s+d}}$, while our regret bound is slightly worse?

To understand the reason, we need to analyze the regret of Gaussian process bandits:

$$R_T \le \widetilde{\mathcal{O}}(\sqrt{T}\underbrace{\sqrt{\widetilde{N}+\delta_{\widetilde{N}}T}}_{\sqrt{\gamma_T}})$$

This follows from equation (7) by Vakili et al. (2021) (here written in our notation) and the fact that the optimal regret for GP bandit is $\widetilde{\mathcal{O}}(\sqrt{T\gamma_T})$. The final bound follows from the fact that $\delta_{\widetilde{N}}$ can be proved to be of order $\widetilde{N}^{1-\frac{2s+d}{d}}$, so that the optimal regret corresponds to, optimizing the value of $\widetilde{N}=T^a$,

$$R_T \le \widetilde{\mathcal{O}}\left(T^{\frac{1}{2}+\frac{1}{2}\min_a \max\{a,1-\frac{2sa}{d}\}}\right).$$

For $a=\frac{d}{2s+d}$, this achieves the optimal bound $T^{\frac{s+d}{2s+d}}$. Instead, from the proof of our theorem 4 we have that the order of our regret can be written as

$$\begin{aligned} R_T &\le \widetilde{\mathcal{O}}\left(\sqrt{\widetilde{N}T}+\sqrt{\widetilde{N}}TN^{-s}\right) \\ &= \widetilde{\mathcal{O}}\left(T^{\frac{1}{2}+\frac{1}{2}\min_a \max\{a,1+\mathbf{a}-\frac{2sa}{d}\}}\right). \end{aligned}$$

Where the only difference is the additional term a in the last exponent, which is highlighted in red. Not coincidentally, this term arises precisely at the point where our approach and that of Vakili et al. (2021) diverge. While in their case, Gaussian process properties are applied, in ours, we are compelled to use a misspecified bandit algorithm to account for projection error. Presently, the most prominent algorithm known for this problem is the ”phased elimination” algorithm, introduced by Lattimore et al. (2020). This algorithm provides a bound on the regret for bandits with a maximum misspecification ε and dimension D in the form:

$$R_T = \widetilde{\mathcal{O}}\left(\sqrt{DT}+\sqrt{\mathbf{D}}T\varepsilon\right).$$

Where the term in red is precisely what prevents our regret to be optimal. The existence of this term has been recognised as very annoying by the same authors of Lattimore et al. (2020), even if it cannot be eliminated for some feature maps. In the end, there are essentially three possibilities: 1) the optimal regret for Gaussian process bandits was proved to be $T^{\frac{s+d}{2s+d}}$ only thanks to the strong assumption of being in an RKHS and methods based on projection cannot achieve the optimal regret for general $\mathcal{C}^s(\mathcal{X})$ spaces 2) the regret bound of our algortihm can be refined to be $T^{\frac{s+d}{2s+d}}$ by improving the regret bound of Lattimore & Szepesvári (2020) in case of our specific feature maps 3) It is possible to modify our approach in a way that does not involve misspecified bandits to achieve regret $T^{\frac{s+d}{2s+d}}$. We leave this question as an open problem.

F Experiments addenda

In this section, we provide a detailed explanation of the experiments conducted in the main paper. Due to space limitations, we were unable to fully elaborate on the rationale behind the selection of the environments discussed in the main paper. Therefore, this section primarily aims to address this gap and provide a comprehensive understanding of the chosen environments.

F.1 Environments

The four plots depicting the function f for each of the four environments can be observed in plots 4 and 5. It is worth noting that the applied noise is consistently Gaussian with a variance of 1.0. Some consideration on the four choices follow.

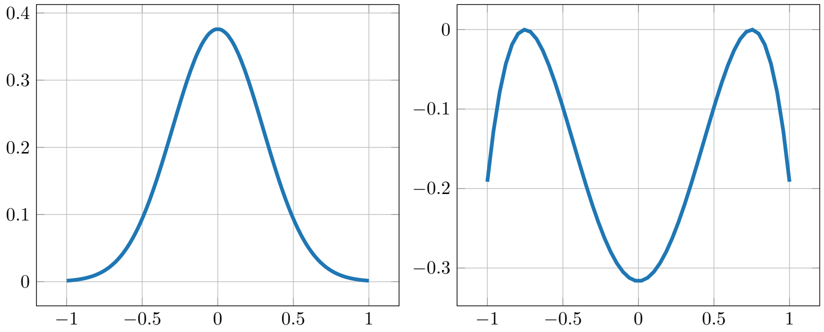


Figure 4: Left: experiment (a), Right: experiment (b).

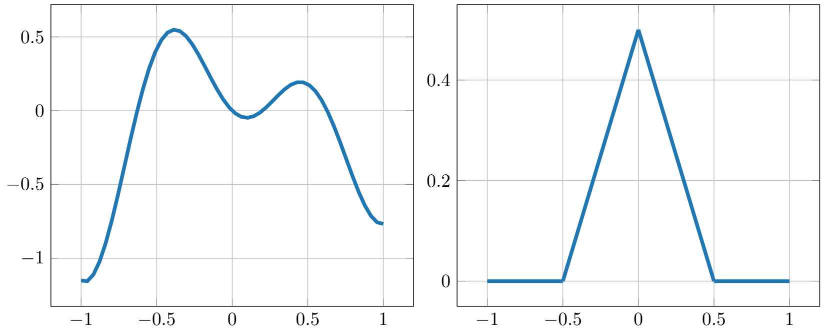


Figure 5: Left: experiment (c), Right: experiment (d).

(a) This curve is a Gaussian, belonging to $\mathcal{C}^\infty([-1,1])$, so it is ideal for algorithms `LegendreUCB` and `ChebyshevUCB`. Instead, if we look at the periodicity at the boundary, this function is only in $\mathcal{C}^0_{per}([-1,1])$, which seems not to be the best scenario for `FourierUCB`. Nonetheless, as the derivatives of any order are small at the boundaries of the interval, we can conjecture that the function is can be well approximated by a $\mathcal{C}^\infty_{per}([-1,1])$. Moreover, this function is even, thus the version of the algorithms with domain knowledge (called $+E$ in the experiments) should perform better.

(b) This curve is a polynomial of degree four, as before $f \in \mathcal{C}^\infty([-1,1])$. As before, if we look at the periodicity at the boundary, this function is only in $\mathcal{C}^0_{per}([-1,1])$, but differently from the Gaussian case, we have $f'(-1) \gg f(1)$ which makes it very hard for `FourierUCB` to work. As the previous one, this function is even.

(c) This curve is a product between a sinusoidal wave and a polynomial, so again $f \in \mathcal{C}^\infty([-1,1])$. This time the function is not in $\mathcal{C}^0_{per}([-1,1])$, which means `FourierUCB` has no performance guarantee. This function, differently from the other ones, is not even, so the algorithms with domain knowledge like `ChebyshevUCB`$+E$ are receiving a wrong information, and are not guaranteed to work.

(d) This curve is piecewise linear, $1-$Lipschtz but only in $\mathcal{C}^0([-1,1])$. However, it is also in $\mathcal{C}^0_{per}([-1,1])$ which means `FourierUCB` has the same performance guarantee of `LegendreUCB` in this case. This function is also even.

As demonstrated in the main paper, the experimental results largely align with the predictions of the theoretical analysis, with one notable exception. Environment (c) stands out due to its discontinuity at the boundaries, which renders `FourierUCB` without a performance guarantee in this particular setting. However, contrary to expectations, plot (c) of Figure 1 illustrates that `FourierUCB` achieves remarkable performance, even surpassing the performance of `IGP-UCB`.

F.2 Hyperparameters of the algorithms

In this section, we provide a comprehensive overview of the hyperparameters utilized for each of the algorithms employed in experiments 1 and 2.

1. `ZOOM`. This algorithm, referred to as Zooming, does not have any specific hyperparameters. Its implementation follows that of Kleinberg et al. (2008), with the exception of the covering oracle, which can be simplified since we are working in one dimension.
2. `IGP-UCB`. For IGP-UCB, we have followed the instructions of Chowdhury & Gopalan (2017). First, $R = 1$, due to the fact that the noise is Gaussian, and in particular $1-$subgaussian. As we are in dimension 1, we have put $\gamma_t = \log(t)^2$, while for the norm in the RKHS, we have $B = 4$. The confidence is set to $\delta = 1/T$, even if choosing $\delta = 1/\sqrt{T}$ gives similar results. Lastly, due to problems of stability we have imposed to evaluate every point 10 times. This procedure is also well known to mitigate computational issues significantly reducing the running time. For the kernel we have used the standard Radial Basis Functions, since most of the benchmarks are infinitely differentiable.
3. `UMA`. For the UCB-Meta-algorithm Liu et al. (2021), we encountered poor performance across all environments. As a result, we dedicated a separate subsection F.3 to thoroughly tune the algorithm and investigate the reasons behind its underperformance.
4. For our algorithms, `FourierUCB`, `LegendreUCB` and `ChebyshevUCB` we have two Hyperparameters to consider. The first hyperparameter is m, which serves as an upper bound on the L^2 norm of f. For `FourierUCB`, we set $m = 0.1$, while for `LegendreUCB` and `ChebyshevUCB`, we set $m = 1.0$. These values are reasonable, taking into account the magnitudes of the benchmark functions. The second hyperparameter is N, which determines the number of features to include. For `FourierUCB`, we set $N = 8$, and for `LegendreUCB` and `ChebyshevUCB`, we set $N = 6$.

F.3 Hyperparameter tuning of `UMA`

To prove the truthfulness of the results of the experiment in the main paper, which show a terrible performance for `UCB-Meta-algorithm`, we have tuned its hyperparameters in a separate experiment on our benchmark (a), for a reduced time horizon $T = 5000$. The parameters used to perform the experiment in the main paper are already the best found.

The `UCB-Meta-algorithm` (Liu et al., 2021) allows for tuning three different hyperparameters:

`alpha`: This parameter represents the degree of the Taylor series utilized by the algorithm. `bins`: It determines the number of bins into which the interval $[-1, 1]$ is divided. `epsilon`: This hyperparameter serves as an upper bound on the value of misspecification.

After fixing a specific set of values for each hyperparameter, namely `alpha` $\in 4, 5, 6, 8, 10, 12$, `bins` $\in 5, 8, 10, 20$, and `epsilon` $\in 0.0, 0.01, 0.05, 0.1$, we conducted a random search by sampling 50 tuples of hyperparameters from the defined sets. For each tuple, we evaluated the algorithm with 5 different random seeds on environment (a). Finally, we selected the tuple that yielded the best performance among the evaluated tuples.

Results The best performing tuple of hyperparameters reveals to be `alpha` $= 4$, `bins` $= 8$ and `epsilon` $= 0.1$. Still, its performance is not significantly different from the others. In the following plot 6, we show the regret obtained by the algorithms as a functions of the three hyperparameters, for the points considered.

We can see that the performance seems to improve for small values of `alpha` and `bins` while it does not change significantly with `epsilon` ($z-$axis). However, the range between the best values in bright red and the worst in blue is narrow, just 1179.7 versus 1317.0.

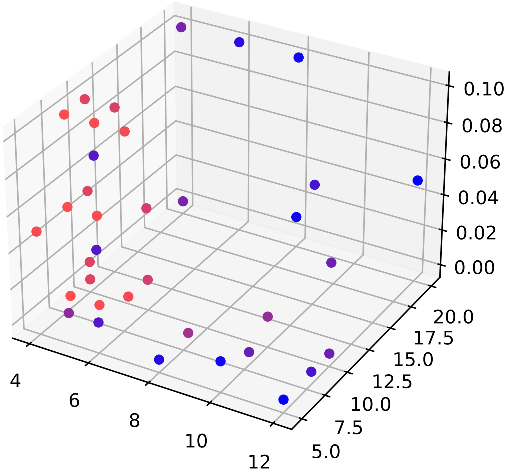
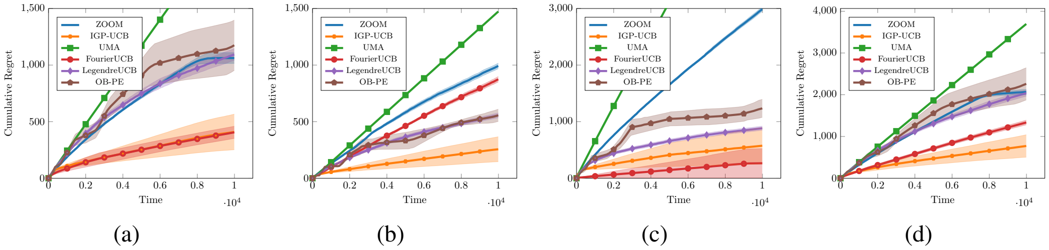


Figure 6: Regret of the algorithm depending on the hyperparamters chosen. On the three axes we have `alpha` (left axis), `bins` (right axis) and `epsilon` (z—axis). Bright red = better, Deep blue = worse.



(a) (b) (c) (d)

Figure 7: Regret plots of the algorithms in four environments (base version)

F.4 Additional experiments

The new experiments, whose regret curves are plotted in 7, are done on the same four environments, but adding the `OB-PE` baseline. This algorithm has only been run with Legendre basis function, as the trigonometric basis is endowed of slightly less theoretical guarantees. As we can see, on one side `OB-PE` surpasses the algorithm with the best theoretical guarantees, `UMA` and, in some environments, `Zooming`. On the other hand, we can see how this algorithm is outperformed by our `LegendreUCB`, which has no theoretical guarantees. As very often happens, in practice the simplicity of `LegendreUCB` seems to win over the most involved, but theoretically grounded, `OB-PE`. Running times of the algorithms can be found in table 3.

Experiment in 2D To be complete, as the experiments in the main paper only deal with continuous bandits on $[-1, 1]$, we performed an experiment with the action space being $[-1, 1]^2$. In this case, the reward function corresponds to $f(\boldsymbol{x}) = e^{-\|\boldsymbol{x}\|_2^2} = e^{x_1^2+x_2^2}$. Results in figure 8 show that, as before, `LegendreUCB` is able to achieve a much better regret than `OB-PE`. Still, the latter shows a regret curve that significantly flattens after roughly 8000 time-steps. This suggests that the former algorithm could be superior for very long horizons. Running times of the algorithms can be found in table 4.

Table 3: Comparison of computation times for experiments of figure 7 on Environment (a)

Algorithm	ZOOM	IGP-UCB	UMA	FourierUCB	LegendreUCB	OB-PE
Time (s)	3.9	14957.3	108.1	3.2	3.4	1.1

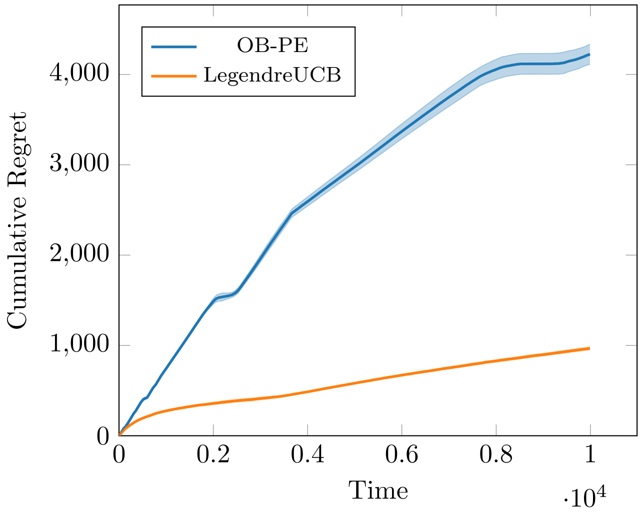


Figure 8: Regret curves of LegendreUCB and OB-PE on an environment with two dimensions.

F.5 Detailed explanation of the experiments

In this section, we report all the details of the experiments performed in the paper. These are important to ensure the truthfullness of the results and the claims based on empirical validation.

Training Details In the main paper, we presented a total of eight experiments, with two experiments conducted for each distinct environment. Each experiment was executed using twenty random seeds, and the computations were distributed across twenty parallel processes using the joblib library. The total computational time for each experiment closely aligns with the running time of the slowest algorithm, which in this case is the IGP-UCB algorithm. As stated in the main paper, the running time for the IGP-UCB algorithm was measured to be 14957.3 seconds, approximately four hours.

Compute We used a server with the following specifications:

- CPU: 88 Intel(R) Xeon(R) CPU E7-8880 v4 @ 2.20GHz cpus
- RAM: 94,0 GB

As mentioned, we parallelized the computing for the 20 different random seeds, therefore only 20 of the 88 cores were actually used.

Reproducibility Given the stochastic nature of the bandit problem, we conducted multiple simulations to account for variability. All experiments were repeated with a total of 20 different random seeds, corresponding to the first 20 natural numbers. The random seed influenced the generation of rewards by the environment, while the proposed algorithms, being deterministic, were unaffected

Table 4: Running times for the experiment in figure 8

Algorithm	LegendreUCB	OB-PE
Time (s)	15.2	85.8

by the seed. This approach allowed us to capture the performance of the algorithms across different random realizations of the bandit problem.