Efficient Kernel UCB for Contextual Bandits

Houssam Zenati, Alberto Bietti, Eustache Diemert, Julien Mairal, Matthieu Martin, Pierre Gaillard

Efficient Kernel UCB for Contextual Bandits

Houssam Zenati
Criteo AI Lab, Inria[1]

Alberto Bietti
NYU Center for Data Science

Eustache Diemert
Criteo AI Lab

Julien Mairal
Inria[1]

Matthieu Martin
Criteo AI Lab

Pierre Gaillard
Inria[1]

Abstract

In this paper, we tackle the computational efficiency of kernelized UCB algorithms in contextual bandits. While standard methods require a $\mathcal{O}(CT^3)$ complexity where T is the horizon and the constant C is related to optimizing the UCB rule, we propose an efficient contextual algorithm for large-scale problems. Specifically, our method relies on incremental Nyström approximations of the joint kernel embedding of contexts and actions. This allows us to achieve a complexity of $\mathcal{O}(CTm^2)$ where m is the number of Nyström points. To recover the same regret as the standard kernelized UCB algorithm, m needs to be of order of the effective dimension of the problem, which is at most $\mathcal{O}(\sqrt{T})$ and nearly constant in some cases.

1 Introduction

Contextual bandits for sequential decision making have become ubiquitous in many applications such as online recommendation systems (Li et al., 2010). At each round, an agent observes a *context* vector and chooses an *action*; then, the *environment* generates a *reward* based on the chosen action. The goal of the agent is to maximize the cumulative reward over time, which requires a careful balancing between exploitation (maximizing reward using past observations) and exploration (increasing the diversity of observations).

In this paper, we consider a kernelized contextual bandit framework, where the rewards are modeled by a function in a reproducing kernel Hilbert space (RKHS). In other words, we assume the expected reward to be linear with respect to a joint context-action feature map of possibly infinite dimension. This setup provides flexible modeling choices through the feature map for both discrete and continuous action sets, and exploration algorithms typically rely on constructing confidence sets for the parameter vector and exploring using upper confidence bound (UCB) rules (Li et al., 2010). The extensions to infinite-dimensional feature maps we consider has been introduced by Krause and Ong (2011); Valko et al. (2013) using kernelized variants of UCB, which allow effective exploration even for rich non-parametric reward functions lying in a RKHS, such as smooth functions over contexts and/or actions.

Despite the rich modeling capabilities of such kernelized UCB algorithms, they lack scalability since standard algorithms scale at best as $\mathcal{O}(CT^3)$ where T is the horizon (total number of rounds) and the constant C is the cost of selecting an action according to the UCB optimization rule. This large cost is due to the need to solve linear systems involving a $t \times t$ kernel matrix at each round t, and motivates developing efficient versions of these algorithms for large problems. In supervised learning, a common technique for reducing computation cost is to leverage the fact that the kernel matrix is often approximately low-rank, and to use Nyström approximations (Williams and Seeger, 2001; Rudi et al., 2015). We extend such approximations to the contextual bandit setting, by relying on incremental updates of a dictionary of Nyström anchor points, which allows us to reduce the complexity to $\mathcal{O}(CTm^2)$, where m is the final number of dictionary elements. In order to preserve a small regret comparable to the vanilla kernel UCB method, m is of the order of an *effective dimension* quantity, which is typically much smaller than T, and at most $\sqrt{T}$.

Closely related to our work, Calandriello et al. (2019,

[1]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

2020) recently considered Nyström approximations in the non-contextual setting with finite actions, corresponding to a Bayesian optimization problem. Whereas their algorithm is effective when there are no contexts, a direct extension to the contextual setting yields a complexity of $\mathcal{O}(Tm^3)$, which may be $\mathcal{O}(T^{2.5})$ in the worst case, despite a batching strategy allowing to recompute a new dictionary only about m times. In contrast, our incremental strategy reduces the previous complexity to $\mathcal{O}(Tm^2)$, and thus at most $\mathcal{O}(T^2)$.

Even though adopting an incremental strategy for updating the Nyström dictionary may seem to be a simple idea, achieving the previously-mentioned complexity while preserving a regret that is comparable to the original kernel UCB approach is non-trivial. Nyström approximations cause dependencies in the projected kernel matrix that makes it difficult to use martingale arguments, which led Calandriello et al. (2020) to use other mathematical tools that are compatible with updates resampling a new Nyström dictionary. In contrast, we manage to use martingale arguments for an incremental strategy that is less computationally expensive. For that, we extend the standard analysis of the OFUL algorithm for linear bandits (Abbasi-yadkori et al., 2011; Chowdhury and Gopalan, 2017) to the kernel setting with Nyström approximations. In particular, this requires non-trivial extensions of concentration bounds to infinite-dimensional objects. Our analysis also uses the incremental structure of the projections that Calandriello et al. (2019) do not have. This allows us to prove the complexity of our algorithm. Moreover, unlike previous works, we explicit the regret-complexity trade-off under the capacity condition assumption. Finally, we also provide numerical experiments showing that our theoretical gains are also observed in practice.

2 Related Work

UCB algorithms are commonly used in the bandit literature to carefully balance exploration and exploitation by defining confidence sets on unknown reward functions (Lattimore and Szepesvári, 2020). For stochastic linear contextual bandits, the OFUL algorithm (Abbasi-yadkori et al., 2011) obtains improved guarantees compared to previous analyses (*e.g.*, Li et al., 2010) by providing tighter confidence bounds based on self-normalized tail inequalities.

Extensions of linear contextual bandits and UCB algorithms to infinite-dimensional representations of contexts or actions have been studied by Krause and Ong (2011) and Valko et al. (2013) by using kernels and Gaussian processes. While their analyses involve different concepts of effective dimension, it can be shown that these are closely related (see Section 3.3). Valko et al. (2013) notably achieves a better scaling in the horizon in the regret, but requires a finite action space. Chowdhury and Gopalan (2017) improves the analysis of GP-UCB using tools inspired by Abbasi-yadkori et al. (2011) and similar to our analysis of kernel-UCB, though it considers the non-contextual setting. Tirinzoni et al. (2020) in the contextual linear bandit problem use a primal-dual algorithm to achieve an optimal asymptotical regret bound but does not address the issue of computational complexity nor the kernelized setting. Likewise, Camilleri et al. (2021) propose a new estimator in the non-contextual kernelized bandit problem to achieve a tighter regret bound using an elimination algorithm but does no focus on computational efficiency neither.

In the Bayesian experimental design literature Derezinski et al. (2020) propose an efficient sampling scheme using determinant point processes in the non-kernel case and a non-contextual framework. For improving the computational complexity of kernelized UCB procedures in a non-contextual setting as well, Calandriello et al. (2019) use a Nyström approximation of the kernel matrix which is recomputed at each step. Because the corresponding algorithm is not practical when a large number of steps are needed, Calandriello et al. (2020) consider a batched version, which significantly improves its computation and complexity.

In contrast, we use an incremental construction based on the KORS method (Calandriello et al., 2017a), which has been used previously with full information feedback (see also Jézéquel et al., 2019), allowing us to significantly improve the computational complexity of the contextual GP-UCB algorithm, for the same regret guarantee. Such an incremental approach appears to be a key to achieve better complexity than a natural contextual variant of the algorithm of Calandriello et al. (2020), see Table 1, both in theory and in practice (see Section 5). Such an extension is unfortunately non-trivial and requires a different regret analysis, as discussed earlier.

Mutnỳ and Krause (2019) also study kernel approximations for efficient variants of GP-UCB, focusing on random feature expansions. Nevertheless, the number of random features may need to be very large–often exponential in the dimension–in order to achieve good regret, due to a misspecification error which requires stronger, uniform approximation guarantees. Finally, Kuzborskij et al. (2019) also considers leverage score sampling for computational efficiency, but focuses on linear bandits in finite dimension.

Algorithm	Regret	Space	Time Complexity
CGP-UCB (Krause and Ong, 2011)	$\mathcal{O}(\sqrt{T}d_{\text{eff}}(\lambda,T))$	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
SupKernelUCB (Valko et al., 2013)	$\mathcal{O}(\sqrt{Td_{\text{eff}}(\lambda,T)\log(C)})$	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
C-BKB (Calandriello et al., 2019)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda,T)}+d_{\text{eff}}(\lambda,T))$	$\mathcal{O}\left(Td_{\text{eff}}\right)$	$\mathcal{O}\left(T^2d_{\text{eff}}^2+CTd_{\text{eff}}^2\right)$
C-BBKB (Calandriello et al., 2020)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda,T)}+d_{\text{eff}}(\lambda,T))$	$\mathcal{O}\left(Td_{\text{eff}}\right)$	$\mathcal{O}\left(Td_{\text{eff}}^3+CTd_{\text{eff}}^2\right)$
K-UCB (ours)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda,T)}+d_{\text{eff}}(\lambda,T))$	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
EK-UCB (ours)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda,T)}+d_{\text{eff}}(\lambda,T))$	$\mathcal{O}\left(Td_{\text{eff}}\right)$	$\mathcal{O}(CTd_{\text{eff}}^2)$

Table 1: Comparison of regret bounds (up to logarithmic factors in T) and total time complexity. When the action space is finite, for e.g in SupKernelUCB, we write $C=|\mathcal{A}|$ its cardinality and note that the argmax is obtained in C computations of the UCB rule. Note that the reported regret of CGP-UCB, SupKernel UCB and CBBKB use here the definition of the effective dimension $d_{\text{eff}}(\lambda,T)$ in Eq. (7) which depends on the horizon T and the parameter λ (i.e the inverse of the GP noise in CGP-UCB, BKB and BBKB). This effective dimension d_{eff} is equivalent, up to logarithmic factors, to the information gain used by Srinivas et al. (2010); Calandriello et al. (2020) and the definition used by Valko et al. (2013) (see Appendix D). Moreover, we report the complexities of the contextualized versions of BKB and BBKBs, noting that the non-contextual versions may benefit from certain optimizations when the action space is discrete (Calandriello et al., 2019, 2020).

3 Warm-up: Kernel-UCB for Contextual Bandits

In this section, we introduce stochastic contextual bandits with reward functions lying in a RKHS, and provide an analysis of the Kernel-UCB algorithm (similar to GP-UCB) which will be a starting point for studying the computationally efficient version in Section 4.

Notations. We define here basic notations. Given a vector $v\in\mathbb{R}^d$ we write its entries $[v_i]_{1\le i\le d}$ and we will write $v^\top w$ or $\langle v,w\rangle$ the dot product for elements in $\mathbb{R}^d$ and in the Hilbert space $\mathcal{H}$. We denote by $\|\cdot\|$ the Euclidean norm and the norm in $\mathcal{H}$. The conjugate transpose for a linear operator L on $\mathcal{H}$ is denoted by L^*. For two operators L,L' on $\mathcal{H}$, we write $L\preccurlyeq L'$ when $L-L'$ is positive semi-definite and we use $\lesssim$ for approximate inequalities up to logarithmic multiplicative or additive terms. A summary of the notations is provided in Appendix A.

3.1 Setup

In the contextual bandit problem, at each time t in $1,\dots,T$, where T is the horizon, for each context x_t in $\mathcal{X}$, an action a_t in $\mathcal{A}$ is chosen by an agent and induces a reward r_t in $\mathbb{R}$. The input and action spaces $\mathcal{X}$ and $\mathcal{A}$ can be arbitrary (e.g., finite or included in $\mathbb{R}^d$ for some $d\ge 1$). Note that $\mathcal{A}$ may change over time, but we keep it fixed here for simplicity.

In this paper, we focus on stochastic kernel contextual bandits and assume that there exists a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ such that

$$r_t=\langle\theta^*,\phi(x_t,a_t)\rangle+\varepsilon_t\,,$$

where ε_t are i.i.d. centered subGaussian noise, $\theta^*\in\mathcal{H}$ is an unknown parameter, and $\phi:\mathcal{X}\times\mathcal{A}\to\mathcal{H}$ is a known feature map associated to $\mathcal{H}$. It satisfies

$$\langle\phi(x,a),\phi(x',a')\rangle=k\big((x,a),(x',a')\big)\,,$$

where k is a positive definite kernel associated to $\mathcal{H}$. We assume k to be bounded, i.e., there exists $\kappa>0$ such that $k(s,s)\le\kappa^2$ for any $s\in\mathcal{X}\times\mathcal{A}$.

Thus, the goal of the agent is, given the previously observed contexts, actions and rewards $(x_s,a_s,r_s)_{s=1\dots t-1}$ and the current context x_t, to choose an action a_t in order to minimize the following regret after T rounds

$$R_T:=\mathbb{E}\left[\sum_{t=1}^{T}\max_{a\in\mathcal{A}}\langle\theta^*,\phi(x_t,a)\rangle-\sum_{t=1}^{T}r_t\right]. \quad (1)$$

3.2 Algorithm: Kernel-UCB

Upper confidence algorithm (UCB) algorithms maintain for each possible action an estimate of the mean reward as well as a confidence interval around that mean, and then chooses at each time the highest upper confidence bound. Formally, if we have a confidence set $\mathcal{C}_t\subset\mathcal{H}$ based on samples $(x_{t'},a_{t'},y_{t'})$, for $t'\in\{1,\dots,t-1\}$ that contains the unknown parameter vector θ^* with high probability, we may define

$$\text{K-UCB}_t(a)=\max_{\theta\in\mathcal{C}_t}\langle\theta,\phi(x_t,a)\rangle \quad (2)$$

as an upper bound on the mean pay-off $\langle\theta^*,\phi(x_t,a)\rangle$ of a. To choose the highest upper confidence bound from the confidence set at time t, the algorithm then selects:

$$a_t\in\arg\max_{a\in\mathcal{A}}\text{K-UCB}_t(a). \quad (3)$$

We then build an empirical estimate of the unknown quantity θ^* using regression. More precisely in the

kernelized setting, we use the regularized least square estimator with

$$\hat{\theta}_t \in \arg\min_{\theta\in\mathcal{H}} \Big\{ \sum_{s=1}^{t} \left(\langle \theta, \phi(x_s, a_s)\rangle - r_s\right)^2 + \lambda\|\theta\|^2 \Big\}. \quad (4)$$

Rearranging the terms $\varphi_s = \phi(x_s, a_s)$ and writing $V_t = \sum_{s=1}^{T} \varphi_s \otimes \varphi_s + \lambda I$, we obtain that the analytical solution for Eq. (4) is $\hat{\theta}_t = V_t^{-1} \sum_{s=1}^{t} \varphi_s r_s$. The previous solution from time $t-1$ then defines the center of the ellipsoidal confidence set

$$\mathcal{C}_t = \{\theta \in \mathcal{H} : \|\theta - \hat{\theta}_{t-1}\|_{V_{t-1}} \le \beta_t(\delta)\}. \quad (5)$$

where $\|\theta\|_V^2 = \theta^\top V \theta$, and $\beta_t(\delta)$ is its radius (see Lemma 3.1). With $\mathcal{C}_t$ in that form, we can write the solution of Eq. (2) as

$$\text{K-UCB}_t(a) = \langle \hat{\theta}_{t-1}, \phi(x_t, a)\rangle + \beta_t(\delta)^{1/2} \|\phi(x_t, a)\|_{V_{t-1}^{-1}}. \quad (6)$$

Indeed, by defining $B_2 = \{x \in \mathbb{R}^d : \|x\|_2 \le 1\}$ the unit ball with the Euclidean norm, it is easy to see that $\mathcal{C}_t = \hat{\theta}_t + \beta_t(\delta)^{1/2} V_{t-1}^{-1/2} B_2$. Then, for $\theta \in B_2$ maximising the quantity $\langle \theta, \phi(x_t, a)\rangle = \phi(x_t, a)^\top \hat{\theta}_{t-1} + \beta_t(\delta)^{1/2} \phi(x_t, a)^\top V_{t-1}^{-1/2} \theta$ gives Eq. (6).

3.3 Regret analysis

We provide an analysis of the regret of the kernelized UCB rule in Eq. (6) using standard statistical analysis definitions of the effective dimension.

Let us write the operator $\Phi_t : \mathcal{H} \to \mathbb{R}^t$ such that $\Phi_t^* = [\varphi_1, \dots \varphi_t]$, where $\varphi_i = \phi(x_i, a_i)$ for $i \in [1, t]$. Let us define K_t the kernel matrix associated to kernel k and the set of pairs $(x_1, a_1), \dots, (x_t, a_t)$, $K_t = \Phi_t \Phi_t^*$ is a $t \times t$ matrix. We define the effective dimension of a kernel matrix as in Hastie et al. (2001) and will use the following in our work.

Definition 3.1. *The effective dimension of the matrix K_T is defined as,*

$$d_{\text{eff}}(\lambda, T) := \text{Tr}(K_T(K_T + \lambda I_T)^{-1}). \quad (7)$$

In what follows, for simplicity of notation, we abbreviate $d_{\text{eff}}(\lambda, T)$ to d_{eff} unless we use different parameters on d_{eff}. To extend the analysis of OFUL (Abbasi-yadkori et al., 2011) to the contextual kernel UCB algorithm, we will use the following proposition that has been proved and used by Jézéquel et al. (2019).

Proposition 3.1. *For any horizon $T \ge 1, \lambda > 0$ and all input sequences $(x_1, a_1), \dots, (x_T, a_T)$*

$$\sum_{k=1}^{T} \log\left(1 + \frac{\lambda_k(K_T)}{\lambda}\right) \le \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}},$$

where $\lambda_k(K_T)$ denotes the k-th largest eigenvalue of K_T.

We now provide a regret bound extending the analysis of Abbasi-yadkori et al. (2011) to the kernel setting. In particular, we start by providing an upper bound on the ellipsoid greater axis.

Lemma 3.1. *Let $\delta \in (0, 1)$ and define $\beta_{t+1}(\delta)$ by*

$$\sqrt{\lambda}\|\theta^*\| + \sqrt{2\log\frac{1}{\delta} + \log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\text{eff}}}.$$

Then, with probability at least $1 - T\delta$, for all $t \in [T]$

$$\left\|\hat{\theta}_t - \theta^*\right\|_{V_t} \le \beta_{t+1}(\delta). \quad (8)$$

We use this lemma (which relies on Proposition 3.1 whose proof is in Appendix B.1) to bound the distance between the estimated parameter $\hat{\theta}_t$ at each round t and the true parameter θ^*. By combining this result with Proposition 3.1, we then prove the following theorem that extends the LinUCB upper bound result from Lattimore and Szepesvári (2020).

Theorem 3.1. *Let $T \ge 2$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x, a), \theta^*\rangle| \le 1$ for all $a \in \bigcup_{t=1}^{T} \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$. Then, the K-UCB rule defined in Eq. (3) for the choice $\mathcal{C}_t$ as in (5) with parameter $\lambda > 0$, and $\delta = 1/T^2$, satisfies the pseudo-regret bound*

$$R_T \lesssim \sqrt{T}\Big(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\Big),$$

where $\lesssim$ hides logarithmic factors in T.

The proof of Theorem 3.1 and the precise statement of the regret bound are given in Appendix B.1.

In particular, assuming the norm of the true parameter $\theta^\star$ to be bounded, we obtain the following corollary with a capacity condition on the effective dimension.

Corollary 3.1. *Assuming the capacity condition $d_{\text{eff}} \le (T/\lambda)^\alpha$ for $0 \le \alpha \le 1$, the regret of K-UCB is bounded as $R_T \lesssim T^{\frac{1+3\alpha}{2+2\alpha}}$ with an optimal $\lambda \approx T^{\frac{\alpha}{1+\alpha}}$.*

As an example, if we consider a kernel that is a tensor product between a linear kernel on contexts and a Sobolev-type kernel (*e.g.*, a Matern kernel) of order s on actions, with $s > d/2$ (where d is the dimension of the continuous action space), then we may consider that the kernel eigenvalues decay as $i^{-2s/d}$, leading to an effective dimension as above with $\alpha = d/2s$, and a regret of $T^{\frac{1}{2}\frac{2s+3d}{2s+d}}$.

Discussion. We note that this regret is not optimal for such problems, but matches the regret of most other kernel or Gaussian process optimization algorithms (see, *e.g.*, Scarlett et al., 2017). More precisely, our analysis

recovers classical rates of the GP-UCB algorithm (Srinivas et al., 2010; Chowdhury and Gopalan, 2017), and extends them to the contextual bandit setting. We note that the analysis of Chowdhury and Gopalan (2017) further removes some logarithmic factors, and similar improvements may be obtained in our setting since it is based on similar tools. The SupKernelUCB algorithm by Valko et al. (2013) obtains improved dependencies on T in the regret bounds, but requires a finite set of actions, and therefore is not directly comparable to ours. The CGP-UCB algorithm by Krause and Ong (2011) obtains similar results to ours in the contextual setting, but uses a different analysis. Our result is therefore not new, and our analysis is meant as a starting point for the efficient variant based on incremental Nyström approximations, which will be introduced in the sequel.

We note that these works use different notions than our effective dimension d_{eff} to characterize complexity, namely the information gain

$$\gamma(\lambda, t) = \frac{1}{2}\log\left(\det\left(I + \frac{1}{\lambda}K_t\right)\right)$$

used by Krause and Ong (2011) as well as the different effective dimension definition in (Valko et al., 2013)

$$\tilde{d}(\lambda, t) \;=\; \min\{j : j\lambda\log T \geq \sum_{k>j}\lambda_k(K_t)\}.$$

It can be shown that these are equivalent up to logarithmic factors to our definition of the effective dimension d_{eff} (see Appendix D). This allows us to compare up to logarithmic factors the algorithm regrets, as shown in Table 1.

4 Efficient Kernel-UCB

In this section, we introduce our efficient kernelized UCB (EK-UCB) algorithm based on incremental Nyström projections. We begin by extending the ellipsoidal confidence bounds from the previous section to the case with projections on finite-dimensional linear subspaces of the RKHS. Then, we present our main algorithm and analyze its complexity and regret.

4.1 Upper confidence bounds with projections

In this section, we study the UCB updates and corresponding high-probability confidence bounds for our EK-UCB algorithm. Because these steps do not depend on a specific choice of projections, we consider generic projection operators onto subspaces of the RKHS, noting that the next sections will consider specific choices based on Nyström approximations.

At round $t \geq 1$, we consider a generic subspace $\tilde{\mathcal{H}}_t$ of $\mathcal{H}$, and let $P_t : \mathcal{H} \to \tilde{\mathcal{H}}_t$ be the orthogonal projection operator on $\tilde{\mathcal{H}}_t$, so that $P_t\mathcal{H} = \tilde{\mathcal{H}}_t$. For a fixed regularization parameter $\lambda > 0$, we consider the following regularized estimator restricted to $\tilde{\mathcal{H}}_t$:

$$\tilde{\theta}_t \in \operatorname*{arg\,min}_{\theta\in\tilde{\mathcal{H}}_t}\left\{\sum_{s=1}^{t}\left(\langle\theta, \phi(x_s, a_s)\rangle - r_s\right)^2 + \lambda\|\theta\|^2\right\}. \quad (9)$$

Define $\tilde{V}_t = \sum_{s=1}^{t} P_t\varphi_s \otimes P_t\varphi_s + \lambda I$, which may be written $\tilde{V}_t = P_tF_tP_t + \lambda I$ where $F_t = \Phi_t^*\Phi_t : \mathcal{H} \to \mathcal{H}$ is the covariance operator. Recalling the notation $Y_t = (r_1, \ldots, r_t)^\top$, we obtain that $\tilde{\theta}_t = \tilde{V}_t^{-1}P_t\Phi_t^*Y_t$. We may then define the following ellipsoidal confidence set:

$$\tilde{\mathcal{C}}_t := \left\{\theta \in \mathcal{H} : \|\theta - \tilde{\theta}_{t-1}\|_{\tilde{V}_{t-1}} \leq \tilde{\beta}_t(\delta)\right\}, \quad (10)$$

for some radius $\tilde{\beta}_t(\delta)$ to be specified later. Note that the ellipsoid is not necessarily contained inside the projected space $\tilde{\mathcal{H}}_t$, and may in fact include θ^* even if $\theta^* \notin \tilde{\mathcal{H}}_t$. This is a crucial difference with random feature kernel approximations (Mutnỳ and Krause, 2019), for which a standard confidence set would be finite dimensional, and thus generally does not include θ^*; this leads to larger regret due to misspecification, unless the number of random features is very large in order to ensure good *uniform* approximation. We may then define the following upper confidence bounds, which still rely on the original feature map ϕ:

$$\text{EK-UCB}_t(a) := \max_{\theta\in\tilde{\mathcal{C}}_t}\langle\theta, \phi(x_t, a)\rangle. \quad (11)$$

This may again be written in closed form as

$$\text{EK-UCB}_t(a) = \langle\hat{\theta}_{t-1}, \phi(x_t, a)\rangle + \tilde{\beta}_t(\delta)^{1/2}\|\phi(x_t, a)\|_{\tilde{V}_{t-1}^{-1}}.$$

We note that for appropriate choices of $\tilde{\mathcal{H}}_t$, such a quantity can be explicitly computed using the kernel trick, as we discuss in Section 4.3. The following lemma shows that $\tilde{\mathcal{C}}_t$ is a valid confidence set, which contains θ^* with high probability, provided that the projection captures well the dominating directions in the covariance operator.

Lemma 4.1. *Let $\delta \in (0, 1)$. Define $\tilde{\beta}_{t+1}(\delta)$ as*

$$\left(\sqrt{\lambda} + \sqrt{\mu_t}\right)\|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\left(e + \frac{et\kappa^2}{\lambda}\right)d_{\text{eff}}},$$

where $\mu_t := \|(I - P_t)F_t^{1/2}\|^2$. Then, with probability at least $1 - T\delta$, for all $t \in [T]$

$$\left\|\tilde{\theta}_t - \theta^*\right\|_{\tilde{V}_t} \leq \tilde{\beta}_{t+1}(\delta). \quad (12)$$

Algorithm 1: Incremental KORS subroutine

Input: Time t, past dictionary $\mathcal{Z}$, context-action s_t, regularization μ, accuracy ε, budget γ

Compute the leverage score $\tilde{\tau}_t$ from $\mathcal{Z}, s_t, \mu, \varepsilon$;

Compute $\tilde{p}_t = \min\{\gamma\tilde{\tau}_t, 1\}$;

Draw $z_t \sim \mathcal{B}(\tilde{p}_t)$ and if $z_t = 1$, add s_t to $\mathcal{Z}$;

Result: Dictionary $\mathcal{Z}$

The quantity μ_t controls how well the projection operator P_t captures the dominating eigen-directions of the covariance operator, and should be at most of order λ in order for the confidence bounds to be nearly as tight as for the vanilla K-UCB algorithm. The next section further discusses how this quantity is controlled with incremental Nyström projections.

4.2 Learning with incremental Nyström projections

We now consider specific choices of the projections P_t and subspaces $\tilde{\mathcal{H}}_t$ obtained by Nyström approximation (Williams and Seeger, 2001; Rudi et al., 2015). In particular, the spaces $\tilde{\mathcal{H}}_t$ now take the form

$$\tilde{\mathcal{H}}_t = \mathrm{Span}\big\{\phi(s), s \in \mathcal{Z}_t\big\}, \tag{13}$$

where $\mathcal{Z}_t \subset \{(x_1, a_1), \dots (x_t, a_t)\}$ is a dictionary of anchor points taken from the previously observed data. Our approach consists of constructing the dictionaries $\mathcal{Z}_t$ *incrementally*, by adding new observed examples (x_t, a_t) on the fly when deemed important, so that we have $\mathcal{Z}_1 \subset \mathcal{Z}_2 \cdots \subset \mathcal{Z}_t$. We achieve this using the Kernel Online Row Sampling (KORS) algorithm of Calandriello et al. (2017a), shown in Algorithm 1, which decides whether to include a new sample $s_t = (x_t, a_t)$ by flipping a coin with probability proportional to its *leverage score* (Mahoney and Drineas, 2009). More precisely, an estimate $\tilde{\tau}$ of the leverage score that uses the state feature φ_t and parameters μ, ε is used to assess how a given state is useful to characterize the dataset. More details on the KORS algorithm are given in Appendix E.

We state the following proposition of Calandriello et al. (2017a, Theorem 1, with $\varepsilon = 1/2$), which will be useful for our regret and complexity analyses.

Proposition 4.1. *Let $\delta > 0$, $n \geq 1$, $\mu > 0$. Then the sequence of dictionaries $\mathcal{Z}_1 \subset \mathcal{Z}_2 \subset \dots \mathcal{Z}_T$ learned by KORS with parameters $\mu > 0, \varepsilon = 1/2$ and $\gamma = 12\log(T/\delta)$ satisfies with probability $1-\delta$, $\forall t \geq 1$*

$$\|(I - P_t)F_t^{1/2}\|^2 \leq \mu \quad \text{and} \quad |\mathcal{Z}_t| \leq 9 d_{\mathrm{eff}}(\mu, T)\log(2T/\delta)^2 .$$

Additionally, the algorithm runs in $\mathcal{O}(d_{\mathrm{eff}}(\mu, T)^2)$ time and $\mathcal{O}(d_{\mathrm{eff}}(\mu, T)^2 \log(T)^4)$ space per iteration.

This result shows that when choosing $\mu \approx \lambda$, then KORS will maintain dictionaries of size at most d_{eff} (up to log factors), while guaranteeing that the confidence bounds studied in Section 4.1 are nearly as good as for the case of K-UCB.

4.3 Implementation and complexity analysis

Here, we analyze the complexity of the algorithm and describe its practical implementation. Recall that at each round t the agent chooses an action a that maximises the UCB rule $\mu_{t,a} + \tilde{\beta}_t \sigma_{t,a}$ where we use Eq. (11) to reformulate the mean term $\mu_{t,a} = \langle \hat{\theta}_{t-1}, \phi(x_t, a)\rangle$ and the variance term $\sigma^2_{t,a} = \|\phi(x_t, a)\|^2_{\tilde{V}_{t-1}^{-1}}$. We use the representer theorem on the projection space $\mathcal{H}_t$ to derive efficient computations of the latter two terms instead of using a kernel trick with $t \times t$ gram matrices. Indeed, in the next proposition, we prove that the two terms can be expressed with $m_t \times m_t$ matrices instead, where $m_t = |\mathcal{Z}_t|$ is the size of the dictionary at time t. We use the notations $K_{\mathcal{S}_t}(s')$ for the kernel column vector $[k(s_1, s'), \dots, k(s_t, s')]^\top$, where $\mathcal{S}_t = \{s_i\}_{i=1\dots t}$ are the past states, and $K_{\mathcal{A},\mathcal{B}}$ for the matrix of kernel evaluations $[k(s, s')]_{s\in\mathcal{A}, s'\in\mathcal{B}}$.

Proposition 4.2. *At any round t, by considering $s_{t,a} = (x_t, a)$, the mean and variance term of the EK-UCB rule can be expressed with:*

$$\begin{aligned}
\Gamma_t &= K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1} \\
\Lambda_t &= \left(K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} K_{\mathcal{S}_{t-1}\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right)^{-1} \\
\tilde{\mu}_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \Lambda_t \Gamma_t \\
\Delta_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right) K_{\mathcal{Z}_{t-1}}(s_{t,a}) \\
\tilde{\sigma}^2_{t,a} &= \frac{1}{\lambda} k(s_{t,a}, s_{t,a}) + \Delta_{t,a}.
\end{aligned}$$

The algorithm then runs in a space complexity of $\mathcal{O}(Tm)$ and a time complexity of $\mathcal{O}(CTm^2)$.

In our algorithm, the incremental updates of the projections allow us to derive rank-one updates of the expressions $\Lambda_t, \Gamma_t, K^{-1}_{\mathcal{Z}_t\mathcal{Z}_t}$ in all cases. First, when the dictionary does not change (i.e $P_t = P_{t-1}$), the update of the $m_t \times m_t$ matrix Λ_t can be performed with Sherman-Morrison updates, and the term $\Gamma_t = K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1}$ can also benefit from a rank-one update given the latest reward and state. Both updates are performed in no more than $\mathcal{O}(m_t^2)$ time and space. Second, when the dictionary changes (i.e $P_t \succ P_{t-1}$), the matrix Λ_t can be updated in two stages with a rank-one update using Sherman-Morrison on the states as if the dictionary did not change, in $\mathcal{O}(m_t^2)$ time and space, and second rank-one update on the dictionary using the Schur complement in $\mathcal{O}(tm_t + m_t^2)$ time and space. Similarly, we can update $\Gamma_t = K_{\mathcal{Z}_t\mathcal{S}_{t-1}} Y_{t-1}$ with a first

Algorithm 2: Efficient Kernel UCB

Input: T the horizon, λ regularization and exploration parameters, k the kernel function, $\varepsilon > 0, \gamma > 0$

Initialization;
Context x_0, a_0 chosen randomly and reward r_0 ;
$\mathcal{S} = \{(x_0, a_0)\}, Y_{\mathcal{S}} = [r_0]$ $\mathcal{Z} = \{(x_0, a_0)\}$;
$\Lambda_t = (K_{\mathcal{Z}\mathcal{S}}K_{\mathcal{S}\mathcal{Z}} + \lambda K_{\mathcal{Z}\mathcal{Z}})^{-1}$ $\Gamma_t = K_{\mathcal{Z}\mathcal{S}}Y_{\mathcal{S}}$;
for $t = 1$ *to* T **do**
  Observe context x_t ;
  Choose $\tilde{\beta}_t$ (e.g as in Lem. 4.1, and $\delta = \frac{1}{T^2}$) ;
  Choose $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\mu}_{t,a} + \tilde{\beta}_t \tilde{\sigma}_{t,a}$;
    $\tilde{\mu}_{t,a} \leftarrow K_{\mathcal{Z}}(s_{t,a})^\top \Lambda_t \Gamma_t$;
    $\Delta_{t,a} = K_{\mathcal{Z}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K_{\mathcal{Z}\mathcal{Z}}^{-1}\right) K_{\mathcal{Z}}(s_{t,a})$
    $\tilde{\sigma}_{t,a}^2 \leftarrow \frac{1}{\lambda} k(s_{t,a}, s_{t,a}) + \Delta_{t,a}$;
  Observe reward r_t and $s_t \leftarrow (x_t, a_t)$;
  $Y_{\mathcal{S}} \leftarrow [Y_{\mathcal{S}}, r_t]^\top, \mathcal{S} \leftarrow \mathcal{S} \cup \{s_t\}$;
  $\mathcal{Z}' \leftarrow \mathrm{KORS}(t, \mathcal{Z}, K_{\mathcal{Z}}(s_t), \lambda, \varepsilon, \gamma)$;
  if $\mathcal{Z}' = \mathcal{Z}$ **then**
    Incremental inverse update Λ_t with s_t;
    $\Gamma_{t+1} \leftarrow \Gamma_t + r_t K_{\mathcal{Z}}(s_t)$;
  end
  else
    $z = \mathcal{Z}' \setminus \mathcal{Z}$;
    Incremental inverse update Λ_t with s_t, z ;
    Incremental inverse update $K_{\mathcal{Z}\mathcal{Z}}^{-1}$ with z;
    $\Gamma_{t+1} \leftarrow [\Gamma_t + r_t K_{\mathcal{Z}}(s_t),\ K_{\mathcal{S}}(z)^\top Y_{\mathcal{S}}]^\top$
  end
end

update on the states and stacking a block of size $1 \times t$ in $\mathcal{O}(tm_t)$ space and time. Eventually, the inverse of the dictionary gram matrix $K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}$ is updated with Schur complement in $\mathcal{O}(m_t^2)$. Besides, the second case when the projection is updated occurs at most m times and the first case at most T times. When the UCB rule is computed on C discrete actions or when we assume that it can be optimized using $\mathcal{O}(C)$ evaluations, given that the KORS algorithm runs in $\mathcal{O}(m^2)$ time and space, our algorithm has a total complexity of $\mathcal{O}(CTd_{\text{eff}}^2)$ in time and $\mathcal{O}(Td_{\text{eff}})$ in space, using that $m \approx d_{\text{eff}}$. Note that, as in all UCB algorithms, including ours, the theoretical value for $\tilde{\beta}_t$ in Lemma 4.1 is hard to estimate and often too pessimistic and leads to over-exploration, as discussed by Calandriello et al. (2020). In practice, choosing a fixed value has shown to perform well in our experiments.

In contrast, the non-incremental approach of Calandriello et al. (2020) in the BBKB algorithm needs to recompute a new dictionary about d_{eff} times. Each update involves the computation of a new covariance matrix $K_{\mathcal{Z}\mathcal{S}}K_{\mathcal{S}\mathcal{Z}}$ which costs $\mathcal{O}(tm_t^2)$ operations for its contextual variant[1], yielding an overall $\mathcal{O}(Td_{\text{eff}}^3)$ with $m \approx d_{\text{eff}}$, as illustrated in Table 1.

4.4 Regret analysis

We now analyze the regret of the EK-UCB algorithm, using Proposition 4.1 as well as Lemma 4.1.

Theorem 4.1. *Let $T \geq 1$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x,a), \theta^* \rangle| \leq 1$ for all $a \in \bigcup_{t=1}^T \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$. Then, the EK-UCB algorithm with regularization λ along with KORS updates with parameter μ satisfies the regret bound*

$$R_T \lesssim \sqrt{T}\left(\sqrt{\frac{\mu m}{\lambda}} + \sqrt{d_{\text{eff}}}\right)\left(\|\theta^*\|(\sqrt{\lambda}+\sqrt{\mu}) + \sqrt{d_{\text{eff}}}\right),$$

where $m := |\mathcal{Z}_T|$. In particular, the choice $\mu = \lambda$ yields $m \lesssim d_{\text{eff}}$ and the bound

$$R_T \lesssim \sqrt{T}\left(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\right).$$

Furthermore, the algorithm runs in $O(Tm)$ space complexity and $O(CTm^2)$ time complexity.

The regret bound is again given up to logarithmic factors and we detail the proof as well as the precise bound in Appendix C. As for K-UCB, one may analyze the resulting regret under a capacity condition, and when $\mu \approx \lambda$, we obtain the same guarantees as in Corollary 3.1. Note that our analysis leverages the fact that the dictionary is constructed incrementally, in particular using a condition $P_t \succeq P_{t-1}$, which yields the approximation term $\sqrt{\mu m/\lambda}$. Had we used fixed projections with some operator P, this approximation term would instead be $\sqrt{\mu/\lambda}$ with $\mu = \|(I-P)F_T^{1/2}\|^2$.

As a consequence of this theorem, the following corollary analyzes when the approximation terms dominate the regret, i.e when the dictionary size does not suffice to recover the original regret bound.

Corollary 4.1. *Assuming the capacity condition $d_{\text{eff}} \leq (T/\lambda)^\alpha$ for $0 \leq \alpha \leq 1$. Let $m \geq 1$, under the assumptions of Thm. 4.1, the regret of EK-UCB satisfies*

$$R_T \lesssim \begin{cases} Tm^{\frac{\alpha-1}{2\alpha}} & \text{if } m \leq T^{\frac{\alpha}{1+\alpha}} \\ T^{\frac{1+3\alpha}{2+2\alpha}} & \text{otherwise} \end{cases}$$

for the choice $\lambda = \mu = Tm^{-1/\alpha}$.

The proof is postponed to Appendix C. In a practical setting, the dictionary size is controlled by the choice of the projection parameter μ. When μ is too high,

[1] The original BBKB algorithm does not involve contexts and consider a finite set of actions, allowing to compute the covariance matrix in $\mathcal{O}(\min(t, |\mathcal{A}|)m_t^2)$.

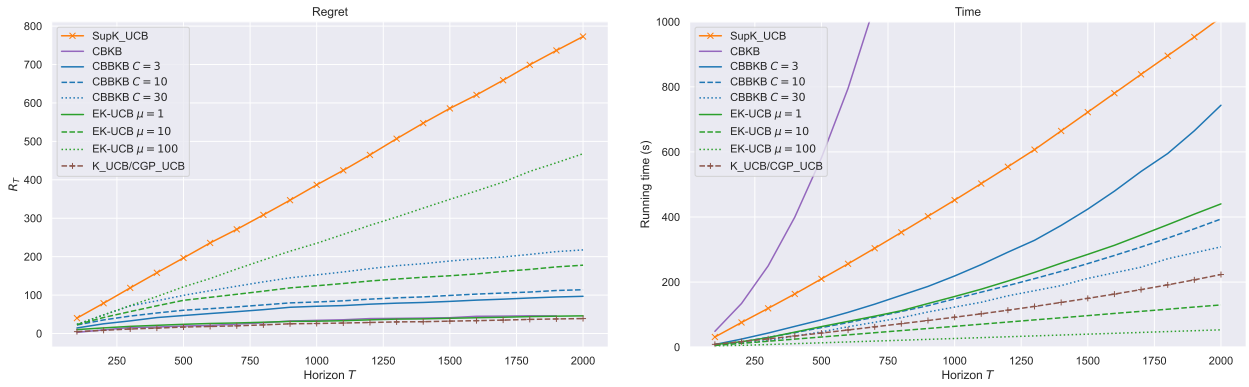

Figure 1: 'Bump' setting: Regret and running times of EK-UCB, CBBKB, CBKB, SupKUCB and K-UCB, with $T = 1000$ and $\lambda = 10$ (see Corollary 3.1 and 4.1). EK-UCB matches the best theoretical regret-time compromise when the projection parameter $\mu = \lambda$. We show other values of μ: higher μ ($\mu = 100$) leads to faster computational time but worse regret, and reciprocally ($\mu = 1$) leads to worse computational time and better regret. Additional results where λ and μ change simultaneously are available in the Appendix F.

it induces a smaller dictionary size m but thus linear regret as indicated in the previous corollary. However, by choosing a low μ, we still recover the original regret but increase the size of the dictionary and thus pay a higher computation time. To recover the original regret, the regularization parameter λ must be set to μ in all cases to recover the original regret, and both values have a theoretical optimal value which depends on the horizon to recover the best convergence rate under the capacity condition assumption.

5 Numerical Experiments

We now evaluate our proposed EK-UCB approach empirically on a synthetic scenario, in order to illustrate its performance in practice. All algorithms have been carefully optimized for fair comparisons.[2] More experimental details, discussions, and additional experimental results are provided in Appendix F.

Experimental setup. We consider a 'Bump' synthetic environment with contexts uniformly distributed in $[0, 1]^p$, with $p = 5$, and actions in $[0, 1]$. The rewards are generated using the function $r(x, a) = \max(0, 1 - \|a - a^*\|_1 - \langle w^*, x - x^* \rangle)$ for some a^*, w^* and x^* picked randomly and fixed. We also consider additional 2D synthetic settings 'Chessboard' and 'Step Diagonal' presented in Appendix F.2.2. We use a Gaussian kernel in this setting. We run our algorithms for $T = 2000$ steps and average our results over different 3 random runs.

[2]The code with open-source implementations for experimental reproducibility is available at `https://github.com/criteo-research/Efficient-Kernel-UCB`.

Baselines. In our experiments, we chose to compare to K-UCB, SupK-UCB and to works which focus on improving the $\mathcal{O}(T^3)$ time-complexity for the kernel case. We implemented K-UCB, SupK-UCB (SupKernelUCB, Valko et al. (2013)), EK-UCB (our efficient version of the K-UCB algorithm) as well as our contextual adaptation of the BKB (Calandriello et al., 2019) and BBKB (Calandriello et al., 2020) algorithms; we will refer to these respectively as CBKB and CBBKB. Specifically, we use the same accumulation criteria as Calandriello et al. (2020) for the "resparsification" strategies (i.e., the resampling of the dictionary) with a threshold parameter C. We also proceed to the same sampling and equation updates as the original algorithms while using our joint kernel on context-action pairs. Note also that CGP-UCB/K-UCB only differ from their parameter β_t and match the same algorithm in our implementation (see second last paragraph in Sec. 4.3).

Results. We report the average regret and running times of the algorithms over different runs in Fig. 1 and Fig. 2 to analyze how the the different algorithms perform. In particular, our algorithm (EK-UCB) achieves low regret while running in low computational time.

In the first example for the 'Bump' environment in Fig. 1, for $T = 2000$, we have set $\lambda = 10$ (of the order of $\sqrt{T}$) and see that the value of $\mu = \lambda$ indeed achieves a good tradeoff between regret and time. The parameter μ determines the quality of the projection required in the algorithm. Thus, for a smaller μ, the algorithm achieves a better regret but pays a higher time complexity. We note that a similar role is played by the parameter C in the BBKB algorithm. The smaller C, the more frequent the dictionary updates,

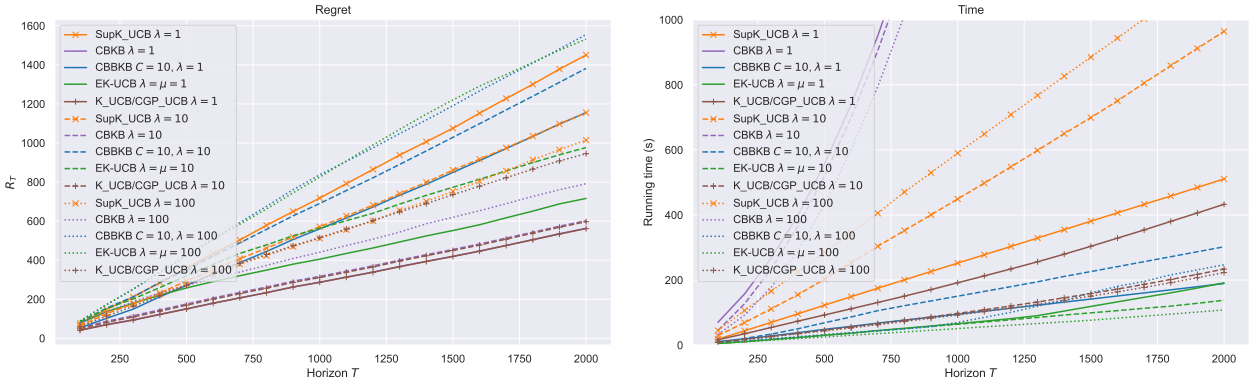

Figure 2: 'Chessboard' setting: Regret and running times of EK-UCB ($\lambda = \mu$), CBBKB ($C = 10$) and CBKB, with $T = 2000$ and with varying λ. We notice that low λ values have better regrets but higher computatinal times. Overall EK-UCB achieves the best regret-time compromise for all parameters of λ while CBBKB sometimes improves upon the K-UCB complexity but has both higher regret than EK-UCB and higher computational time.

and thus the slower is the algorithm. While the CGP-UCB/K-UCB obtains the best regret, we note also, that EK-UCB ($\mu = 1$), CBKB (which is CBBKB with $C = 1$) essentially take the full dictionary $m \approx T$ and thus also match K-UCB, but with dictionary building computational overheads which make them more computationally intensive than K-UCB itself. In the Appendix F we provide additional results that show that consistently EK-UCB provides the best time-regret compromise with regards to K-UCB.

Second, in Fig. 2 we show for the 'Chessboard' setting the influence of varying λ for all methods (fixing $\mu = \lambda$ for EK-UCB). Both CBBKB and EK-UCB improve upon the K-UCB computational time in this case, but EK-UCB achieves lower computational times while also having lower regrets than CBBKB for all settings. We also notice that the CBKB algorithm runs much slower than the CBBKB algorithm in all experiments, as expected due to its costly dictionary update at every round which requires processing all previous points. The computational overheads of its dictionary building therefore makes it not practical despite its theoretical guarantees. Note also that CBBKB uses scores based on the variance estimates on past states for its "resparfication" strategy and EK-UCB uses leverage scores to build its dictionary thus looking for directions that are orthogonal to the previous anchor points; both approaches are more effective than updating the dictionary at each round. Eventually, recall that our incremental projection scheme allows us to perform rank-one updates of the dictionary. This also contributes to the practical speedup of our EK-UCB algorithm, as compared to the CBBKB strategy.

Moreover, SupK-UCB performs poorly in our experiments due to its over-exploring elimination strategy that might be beneficial only for large T and makes it unpractical in its current time-complexity. Note that the main author of SupK-UCB co-authored Calandriello et al. (2019) where it is mentioned that it indeed has "tighter analysis than GP-UCB [but] does not work well in practice".

6 Discussions

In this work, we proposed a method for contextual kernel UCB algorithms in large-scale problems. The EK-UCB algorithm runs in $\mathcal{O}(Td_{\text{eff}})$ space and $\mathcal{O}(CTd_{\text{eff}}^2)$ time complexity, which significantly improves over the standard contextual kernel UCB method. Note that while previous efficient Gaussian process algorithms allow to scale up the learning problems in non contextual and discrete action environments, we have shown how the incremental projection updates were crucial to perform efficient approximations in the joint context-action space, providing the same regret guarantees for a smaller computational cost. We note that the batching strategy of BBKB may still be useful even under our incremental updates, and thus provides an interesting avenue for future work. Another natural question is whether we may obtain algorithms with better regret guarantees similar to Valko et al. (2013) in the finite action case, while also achieving gains in computational efficiency as in our work.

Acknowledgments

JM was supported by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes, (ANR-19-P3IA-0003).

References

Y. Abbasi-yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Adv. Neural Information Processing Systems (NIPS)*, 2011.

D. Calandriello, A. Lazaric, and M. Valko. Efficient second-order online kernel learning with adaptive embedding. In *Adv. Neural Information Processing Systems (NIPS)*, 2017a.

D. Calandriello, A. Lazaric, and M. Valko. Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning (ICML)*, 2017b.

D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory (COLT)*, 2019.

D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco. Near-linear time Gaussian process optimization with adaptive batching and resparsification. In *International Conference on Machine Learning (ICML)*, 2020.

R. Camilleri, J. Katz-Samuels, and K. Jamieson. High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning (ICML)*, 2021.

S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2017.

M. Derezinski, F. Liang, and M. Mahoney. Bayesian experimental design using regularized determinantal point processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. 2001.

R. Jézéquel, P. Gaillard, and A. Rudi. Efficient online learning with kernels for adversarial large scale problems. In *Adv. in Neural Information Processing Systems (NeurIPS)*. 2019.

A. Krause and C. S. Ong. Contextual gaussian process bandit optimization. In *Adv. Neural Information Processing Systems (NIPS)*, 2011.

I. Kuzborskij, L. Cella, and N. Cesa-Bianchi. Efficient linear bandits through matrix sketching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web (WWW)*, 2010.

M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences (PNAS)*, 106(3): 697–702, 2009.

M. Mutnỳ and A. Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Adv. in Neural Information Processing Systems (NeurIPS)*, 2019.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Adv. in Neural Information Processing Systems (NIPS)*, 2015.

J. Scarlett, I. Bogunovic, and V. Cevher. Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory (COLT)*, 2017.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

A. Tirinzoni, M. Pirotta, M. Restelli, and A. Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. In *NeurIPS*, 2020.

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

C. K. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Adv. Neural Information Processing Systems (NIPS)*, 2001.

Appendix

This appendix is organized as follows:

- Appendix A: summary of the notations used in the analysis
- Appendix B: proofs of Section 3 – Kernel-UCB
- Appendix C: proofs of Section 4 – Efficient Kernel-UCB
- Appendix E: details on the implementation of the algorithms
- Appendix F: additional experiment details, discussions and results

A List of notations

In this appendix, we recall useful notations that are used throughout the paper.

Below are generic notations and notations related to RKHS:

- $[n] := \{1, \dots, n\}$
- $\lesssim$ denotes an approximate inequality up to logarithmic multiplicative or additive terms
- $\mathcal{X}$ is the input space
- $\mathcal{A}$ is the action set
- $k : (\mathcal{X} \times \mathcal{A}) \times (\mathcal{X} \times \mathcal{A}) \to \mathbb{R}$ is a bounded positive definite Kernel
- $\kappa > 0$ is an upper-bound on the kernel $\kappa^2 \geq \sup_{s \in \mathcal{X} \times \mathcal{A}} k(s, s)$.
- $\mathcal{H}$ is the reproducing kernel Hilbert space associated to k
- $\phi : \mathcal{X} \times \mathcal{A} \to \mathcal{H}$ is the feature map such that $k(s, s') = \langle \phi(s), \phi(s') \rangle$ for any $s, s' \in \mathcal{X} \times \mathcal{A}$
- $\langle \varphi, \varphi' \rangle := \varphi^\top \varphi'$ denotes the inner product for any $\varphi, \varphi' \in \mathcal{H}$
- $\| \cdot \|$ denotes the norm associated to $\mathcal{H}$. It is the one induced by the inner product, i.e., $\|\varphi\|^2 = \langle \varphi, \varphi \rangle$
- $\| \cdot \|_V$ denotes for any symmetric positive semi-definite operator $V : \mathcal{H} \to \mathcal{H}$ the norm such that $\|\varphi\|_V = \|V^{1/2}\varphi\|$ for all $\varphi \in \mathcal{H}$
- $L \preccurlyeq L'$ means that $L - L'$ is positive semi-definite for two operators L, L' on $\mathcal{H}$
- $\varphi \otimes \varphi' : \mathcal{H} \to \mathcal{H}$ is the tensor product of φ and $\varphi' \in \mathcal{H}$

Below are notations related to the sequential setting. Here, $t \in [T]$ denotes the index of the round:

- $\theta^* \in \mathcal{H}$ is the unknown parameter
- $x_t \in \mathcal{X}, a_t \in \mathcal{A}$ are the context and action played at round t
- $s_t := (x_t, a_t) \in \mathcal{X} \times \mathcal{A}$
- $\mathcal{S}_t := \{s_1, \dots, s_t\}$ denotes the history
- $\varepsilon_1, \dots, \varepsilon_T$ are independent centered sub-Gaussian noise
- $H_t := (\varepsilon_1, \dots, \varepsilon_t)^\top$
- $\mathcal{F}_t := \sigma(\varepsilon_1, \dots, \varepsilon_t)$ is the natural filtration with respect to $(\varepsilon_i)_{i \geq 1}$
- $r_t := \langle \theta^*, \phi(x_t, a_t) \rangle + \varepsilon_t$ is the reward
- $Y_t := (r_1, \dots, r_t)^\top \in \mathbb{R}^t$ is the vector of rewards
- $\varphi_t := \phi(x_t, a_t) \in \mathcal{H}$
- $\lambda > 0$ is the regularization parameter
- $F_t := \sum_{s=1}^t \varphi_s \otimes \varphi_s$ is the covariance operator
- $V_t := \sum_{s=1}^t \varphi_s \otimes \varphi_s + \lambda I : \mathcal{H} \to \mathcal{H}$ is the regularized covariance operator
- $\Phi_t : \mathcal{H} \to \mathbb{R}^t$ is the operator such that $[\Phi_t \varphi]_i = \varphi(x_i, a_i) = \langle \varphi, \phi(x_i, a_i) \rangle$ for any $\varphi \in \mathcal{H}$ and $i \in [t]$
- Φ^* denotes the conjugate transpose of a linear operator Φ on $\mathcal{H}$
- $K_t := \Phi_t \Phi_t^* : \mathbb{R}^t \to \mathbb{R}^t$ is the kernel matrix at time $t \geq 1$. Note that $[K_t]_{ij} = k((x_i, a_i), (x_j, a_j))$.
- $\lambda_i(K_t)$ is the i-th largest eigenvalue of K_t
- $d_{\text{eff}}(\lambda, t) := \text{Tr}(K_t(K_t + \lambda I_t)^{-1})$ is the effective dimension of the matrix K_t

Below are notations related to the Kernel-UCB algorithm without projections:

- $\hat{\theta}_t := V_t^{-1} \Phi_t^* Y_t$ is the estimator of the algorithm
- $\delta > 0$ is the confidence level
- $\beta_t(\delta)$ is the radius of the confidence ellipsoid of the algorithm

– $\mathcal{C}_t := \left\{\theta \in \mathcal{H} : \left\|\theta - \hat{\theta}_{t-1}\right\|_{V_{t-1}} \leq \beta_t(\delta)\right\}$ is the confidence ellipsoid played by the algorithm

Below are notations related to the Kernel-UCB algorithm with projections. All along the analysis, the notation $\tilde{x}$ corresponds to the projected version of the object x.

– $\mathcal{Z}_t \subset \left\{(x_1, a_1), \ldots, (x_t, a_t)\right\}$ is a dictionary
– $\tilde{\mathcal{H}}_t := \mathrm{Span}\left\{\phi(s), s \in \mathcal{Z}_t\right\}$ is a linear subspace of $\mathcal{H}$ and is used at round t.
– $P_t : \mathcal{H} \to \tilde{\mathcal{H}}_t$ is the Euclidean projection onto $\mathcal{H}$ so that $\tilde{\mathcal{H}}_t = \{P_t\varphi, \varphi \in \mathcal{H}\}$
– $\tilde{V}_t := \sum_{s=1}^{t}(P_t\varphi_s) \otimes (P_t\varphi_s) + \lambda I = P_t F_t P_t + \lambda I$ is the regularized projected covariance operator
– $\tilde{\theta}_t := P_t\tilde{V}_t^{-1}P_t\Phi_t^* Y_t$ is the projected estimator of the algorithm
– $\tilde{C}_t := \left\{\theta \in \mathcal{H} : \|\theta - \tilde{\theta}_{t-1}\|_{\tilde{V}_{t-1}} \leq \tilde{\beta}_t(\delta)\right\}$ is the confidence ellipsoid related to the projected estimator
– $\mu_t := \left\|(I - P_t)F_t^{1/2}\right\|^2$ is the approximation error of the projection

Eventually, we provide notations related to the kernel matrix computations when we write the update rules of the efficient algorithm.

– $K_{\mathcal{S}}(s')$ is the kernel column vector $[k(s_1, s'), \ldots, k(s_l, s')]^\top$ of size $|\mathcal{S}| = l$. Note that $K_{\mathcal{S}_t}(s) = \Phi_t\phi(s)$.
– $K_{\mathcal{Z}\mathcal{S}}$ is the kernel matrix vector $[k(z, s)]_{z\in\mathcal{Z}, s\in\mathcal{S}}$ of size $|\mathcal{Z}| \times |\mathcal{S}|$.
– $s_{t,a} = (x_t, a)$ refers to the pair of context x_t and any action $a \in \mathcal{A}_t$ that can be chosen in the UCB rule.

B Proofs of Section 3: Kernel UCB

In this appendix we prove of Lemma 3.1 and Theorem 3.1.

B.1 Proof of Lemma 3.1

We first prove Lemma 3.1, which controls the size of the confidence intervals considered by the algorithm. It states that with probability $1 - \delta$, for all $t \geq 1$:

$$\theta^* \in C_t, \quad \text{where} \quad C_t = \left\{\theta \in \mathbb{R}^d, \|\theta - \hat{\theta}_{t-1}\|_{V_{t-1}} \leq \beta_t(\delta)\right\}. \tag{14}$$

Lemma 3.1. *Let $\delta \in (0, 1)$. Assume $\kappa^2 \geq \sup_{s\in\mathcal{X}\times\mathcal{A}} k(s, s)$. Then with probability at least $1 - T\delta$, for all $t \in [T]$*

$$\begin{aligned}\|\hat{\theta}_t - \theta^*\|_{V_t} &\leq \sqrt{\lambda}\|\theta^*\| + \sqrt{2\log\frac{1}{\delta} + \log\left(\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)\right)} \\ &\leq \sqrt{\lambda}\|\theta^*\| + \sqrt{2\log\frac{1}{\delta} + \log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\mathrm{eff}}(\lambda, T)} \quad =: \quad \beta_{t+1}(\delta).\end{aligned}$$

Proof. The analysis is inspired by the one of Abbasi-yadkori et al. (2011) for linear bandits and uses inequality tails on vector valued martingales. We introduce $M_t = \sum_{s=1}^{t}\varphi_s\varepsilon_s \in \mathcal{H}$, which is a martingale with regards to the natural filtration $\mathcal{F}_t := \sigma(\varepsilon_1, \ldots, \varepsilon_t)$. Solving the least-square optimization problem (4), $\hat{\theta}_t$ equals

$$\hat{\theta}_t = V_t^{-1}\sum_{s=1}^{t}\varphi_s Y_s = V_t^{-1}\sum_{s=1}^{t}\varphi_s(\varphi_s^\top\theta^* + \varepsilon_s) = V_t^{-1}((V_t - \lambda I_d)\theta^* + M_t) = \theta^* - \lambda V_t^{-1}\theta^* + V_t^{-1}M_t\,.$$

Multiplying by the square root of V_t and using the triangle inequality

$$\left\|V_t^{1/2}\left(\hat{\theta}_t - \theta^*\right)\right\| = \left\| - \lambda V_t^{-1/2}\theta^* + V_t^{-1/2}M_t\right\| \leq \lambda\left\|V_t^{-1/2}\theta^*\right\| + \left\|V_t^{-1/2}M_t\right\|.$$

On the other hand, given that $V_t = F_t + \lambda I$ where F_t is positive semi-definite, $V_t^{-1/2} \preccurlyeq \lambda^{-1/2}I$ and thus

$$\lambda\|V_t^{-1/2}\theta^*\| \leq \lambda\frac{1}{\sqrt{\lambda}}\|\theta^*\| = \sqrt{\lambda}\|\theta^*\|\,.$$

We now prove for the other term that with probability at least $1 - \delta$

$$\|V_t^{-1/2}M_t\| \leq \sqrt{2\log\frac{1}{\delta} + \log\det\frac{1}{\lambda}(K_t + \lambda I)}\,.$$

Step 1: Martingales For all $\nu \in \mathcal{H}$, we define the random-variable

$$S_{t,\nu} = \exp\Big(\nu^\top M_t - \frac{1}{2}\nu^\top V_t \nu\Big)$$

and now show that it is a $\mathcal{F}_t$-super-martingale. First, note that the common distribution of the $\varepsilon_1, \dots, \varepsilon_t$ is 1-sub Gaussian, i.e., for all $\mathcal{F}_{t-1}$-measurable real-valued random variable ν_{t-1}, we have

$$\mathbb{E}\left[\exp(\nu_{t-1}\varepsilon_t)|\mathcal{F}_{t-1}\right] \leq \exp\Big(\frac{\nu_{t-1}^2}{2}\Big). \tag{15}$$

Thus, using that $M_t = M_{t-1} + \varphi_t \varepsilon_t$ and $V_t = V_{t-1} + \varphi_t \otimes \varphi_t$,

$$\begin{aligned}
\mathbb{E}\left[S_{t,\nu}|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\exp\big(\nu^\top M_t - \frac{1}{2}\nu^\top V_t \nu\big)|\mathcal{F}_{t-1}\right] \\
&= \mathbb{E}\left[S_{t-1,\nu}\exp\big(\nu^\top \varphi_t \varepsilon_t - \frac{1}{2}\nu^\top(\varphi_t \otimes \varphi_t)\nu\big)|\mathcal{F}_{t-1}\right] \\
&= S_{t-1,\nu}\,\mathbb{E}\left[\exp\big(\nu^\top \varphi_t \varepsilon_t - \frac{1}{2}(\nu^\top \varphi_t)^2\big)\big|\mathcal{F}_{t-1}\right] \leq S_{t-1,\nu}\,,
\end{aligned}$$

where the last inequality is by applying (15) with $\nu_{t-1} = \nu^\top \varphi_t$ since $\varphi_t = \phi(x_t, a_t)$ is $\mathcal{F}_{t-1}$-measurable. Therefore, $S_{t,\nu}$ is a $\mathcal{F}_t$-super-martingale for any $\nu \in \mathcal{H}$, and

$$\mathbb{E}\left[S_{t,\nu}\right] \leq \mathbb{E}\left[S_{0,\nu}\right] = \exp\Big(-\frac{\lambda}{2}\|\nu\|^2\Big). \tag{16}$$

Rewriting $S_{t,\nu}$ in its vertex form with $m = V_{t-1}M_t$ yields

$$S_{t,\nu} = \exp\left(-\frac{1}{2}(\nu - m)^\top V_t(\nu - m)\right) \times \exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\,,$$

which substituted into (16) entails

$$\mathbb{E}\left[\exp\Big(-\frac{1}{2}(\nu - m)^\top V_t(\nu - m)\Big) \times \exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \leq \exp\Big(-\frac{\lambda}{2}\|\nu\|^2\Big), \qquad \forall \nu \in \mathcal{H}\,. \tag{17}$$

Step 2: Laplace's method integrating

Now, following Laplace's method which is standard for the proof of LinUCB, the goal is to integrate both sides of the above expression. Let us first rewrite it in order to consider finite dimensional objects thanks to the Kernel trick.

Recalling $V_t := \Phi_t^*\Phi_t + \lambda I$ and $K_t := \Phi_t\Phi_t^*$, following (Valko et al., 2013), we will use the following identities:

$$\begin{aligned}
&(\Phi_t^*\Phi_t + \lambda I)\Phi_t^* = \Phi_t^*(\Phi_t\Phi_t^* + \lambda I) && (18)\\
\Rightarrow \quad &V_t\Phi_t^* = \Phi_t^*(K_t + \lambda I) && (19)\\
\Rightarrow \quad &\Phi_t^*(K_t + \lambda I)^{-1} = V_t^{-1}\Phi_t^*\,. && (20)
\end{aligned}$$

Let $x \in \mathbb{R}^t$ and write $\nu = V_t^{-1}\Phi_t^* x \in \mathcal{H}$ and recall that $m = V_t^{-1}M_t = V_t^{-1}\Phi_t^* H_t$, where $H_t = (\varepsilon_1, \dots, \varepsilon_t)^\top$. We have

$$\begin{aligned}
\exp\Big(-\frac{1}{2}(\nu - m)^\top V_t(\nu - m)\Big) &= \exp\Big(-\frac{1}{2}(x - H_t)^\top \Phi_t V_t^{-1} V_t V_t^{-1}\Phi_t^*(x - H_t)\Big) \\
&= \exp\Big(-\frac{1}{2}(x - H_t)^\top \Phi_t\Phi_t^*(K_t + \lambda I)^{-1}(x - H_t)\Big) \qquad \leftarrow \text{ by (20)} \\
&= \exp\Big(-\frac{1}{2}(x - H_t)^\top K_t(K_t + \lambda I)^{-1}(x - H_t)\Big) \qquad \leftarrow K_t = \Phi_t\Phi_t^* \\
&= \exp\Big(-\frac{1}{2}(x - H_t)^\top K_t^{1/2}(K_t + \lambda I)^{-1}K_t^{1/2}(x - H_t)\Big)\,, \qquad (21)
\end{aligned}$$

where the last equality is because $(K_t + \lambda I)^{-1}$ and $K_t^{1/2}$ commute. Similarly,

$$\exp\Big(-\frac{\lambda}{2}\|\nu\|^2\Big) = \exp\Big(-\frac{\lambda}{2}x^\top K_t^{1/2}(K_t+\lambda I)^{-2}K_t^{1/2}x\Big)\,.$$

Combining with (16) and (21) thus gives for any $x \in \mathbb{R}^t$,

$$\mathbb{E}\left[\exp\Big(-\frac{1}{2}(x-H_t)^\top K_t^{1/2}(K_t+\lambda I)^{-1}K_t^{1/2}(x-H_t)\Big)\times\exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \leq \exp\Big(-\frac{\lambda}{2}x^\top K_t^{1/2}(K_t+\lambda I)^{-2}K_t^{1/2}x\Big)\,. \quad (22)$$

Now, that we are back to finite dimensional space, the idea would consists in integrating both parts over $x \in \mathbb{R}^t$. But the matrix K_t may be non-invertible, we thus need a few more steps to integrate over $\mathrm{Im}(K_t)$ only.

Let $d_t = \mathrm{rank}(K_t)$ and $Q_t \in \mathbb{R}^{t\times d_t}$ the matrix formed by the orthonormal eigenvectors of K_t with non-zero eigenvalues. Let $u \in \mathbb{R}^{d_t}$ then $Q_t u \in \mathrm{Im}(K_t)$ and there exists $x \in \mathbb{R}^t$ such that $K_t^{1/2}x = Q_t u$. Defining $z \in \mathbb{R}^{d_t}$ such that $Q_t z = K_t^{1/2}H_t$ and substituting into Inequality (22) yields, for any $u \in \mathbb{R}^{d_t}$

$$\mathbb{E}\left[\exp\Big(-\frac{1}{2}(u-z)^\top Q_t^\top(K_t+\lambda I)^{-1}Q_t(u-z)\Big)\times\exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \leq \exp\Big(-\frac{\lambda}{2}u^\top Q_t^\top(K_t+\lambda I)^{-2}Q_t u\Big)\,. \quad (23)$$

Now, we integrate both sides over $u \in \mathbb{R}^{d_t}$, recognizing a multidimensional Gaussian density, we have

$$\int_{\mathbb{R}^d}\exp\Big(-\frac{1}{2}(u-z)^\top Q_t^\top(K_t+\lambda I)^{-1}Q_t(u-z)\Big)d\mu(u) = \sqrt{\det\big(2\pi(Q_t^\top(K_t+\lambda I)^{-1}Q_t)^{-1}\big)} = \sqrt{(2\pi)^{d_t}\prod_{i=1}^{d_t}\big(\lambda_i(K_t)+\lambda\big)}\,,$$

where $\lambda_i(K_t)$ is the i-th largest eigenvalue of K_t. Similarly

$$\int_{\mathbb{R}^d}\exp\Big(-\frac{\lambda}{2}u^\top Q_t^\top(K_t+\lambda I)^{-2}Q_t u\Big)d\mu(u) = \sqrt{\det\Big(2\pi\lambda^{-1}\big(Q_t^\top(K_t+\lambda I)^{-2}Q_t\big)^{-1}\Big)} = \sqrt{\Big(\frac{2\pi}{\lambda}\Big)^{d_t}\prod_{i=1}^{d_t}\big(\lambda_i(K_t)+\lambda\big)^2}\,.$$

Therefore, by the Fubini-Tonelli theorem, plugging the last two equations into Inequality (23) entails

$$\sqrt{(2\pi)^{d_t}\prod_{i=1}^{d_t}\big(\lambda_i(K_t)+\lambda\big)}\,\mathbb{E}\left[\exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \leq \sqrt{\Big(\frac{2\pi}{\lambda}\Big)^{d_t}\prod_{i=1}^{d_t}\big(\lambda_i(K_t)+\lambda\big)^2}\,,$$

which, after reorganizing the terms, yields

$$\mathbb{E}\left[\exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \leq \sqrt{\prod_{i=1}^{d_t}\Big(1+\frac{\lambda_i(K_t)}{\lambda}\Big)} = \sqrt{\frac{\det(K_t+\lambda I)}{\lambda^t}}\,.$$

Step 3: Markov-Chernov bound. It remains to upper-bound the above expectation using concentration inequalities. For $u > 0$,

$$P\Big(\|V_t^{-1/2}M_t\| > u\Big) = P\left(\frac{\|V_t^{-1/2}M_t\|^2}{2} > \frac{u^2}{2}\right) \leq \exp\Big(-\frac{1}{2}u^2\Big)\,\mathbb{E}\left[\exp\Big(\frac{1}{2}\|V_t^{-1/2}M_t\|^2\Big)\right] \leq \exp\left(-\frac{u^2}{2}+\frac{1}{2}\log\frac{\det(K_t+\lambda I)}{\lambda^t}\right) = \delta \quad (24)$$

for the claimed choice

$$u = \sqrt{2\log\frac{1}{\delta}+\log\det\frac{1}{\lambda}(K_t+\lambda I)}\,.$$

The proof then concludes by using Prop. 3.1 on the $\log\det\frac{1}{\lambda}(K_t+\lambda I)$ term and by applying a union bound. □

B.2 Proof of Theorem 3.1

We are now ready to prove Theorem 3.1, which upper-bounds the regret of K-UCB.

Theorem 3.1. *Let $T \geq 2$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x,a), \theta^* \rangle| \leq 1$ and $\|\phi(x,a)\| \leq \kappa$ for all $a \in \bigcup_{t=1}^T \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$. Then, the K-UCB rule defined in Eq. (3) for the choice $\mathcal{C}_t$ as in (5) satisfies the pseudo-regret bound*

$$\begin{aligned} R_T &\leq 2 + 2\sqrt{T\left(\log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)\right)}\left[\sqrt{\lambda}\|\theta^*\| + \sqrt{2\log(T) + \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda)}\right] \\ &\lesssim \sqrt{T}\Big(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}(\lambda, T)} + d_{\text{eff}}(\lambda, T)\Big)\,. \end{aligned}$$

Proof. Let $\delta \in (0, 1/2)$. By Lemma 3.1, with probability $1 - T\delta$,

$$\forall t \in [T], \quad \theta^* \in C_t\,. \tag{25}$$

Step 1: Small instantaneous regrets under the event (25). Assume that (25) holds. Let

$$a_t^* := \max_{a \in \mathcal{A}_t} \langle \phi(x_t, a), \theta^* \rangle \quad \text{and} \quad \Delta_t := \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle$$

be respectively the optimal decision and the instantaneous regret at round t. We also define

$$\rho_t \in \arg\max_{\theta \in C_t} \big\{ \langle \phi(x_t, a_t), \theta \rangle \big\}\,.$$

Since $\theta^* \in C_t$, we have

$$\langle \phi(x_t, a_t^*), \theta^* \rangle \leq \max_{\theta \in C_t} \{\langle \phi(x_t, a_t^*), \theta \rangle\} = \text{K-UCB}_t(a_t^*) \leq \text{K-UCB}_t(a_t) = \max_{\theta \in C_t} \{\langle \phi(x_t, a_t), \theta \rangle\} = \langle \phi(x_t, a_t), \rho_t \rangle\,,$$

which entails because θ^* and $\tilde{\theta}_{t-1}$ belong to C_t,

$$\Delta_t = \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle \leq \langle \phi(x_t, a_t), \rho_t - \theta^* \rangle \leq \|\phi(x_t, a_t)\|_{V_{t-1}^{-1}} \|\rho_t - \theta^*\|_{V_{t-1}} \leq 2\|\phi(x_t, a_t)\|_{V_{t-1}^{-1}} \beta_t(\delta)\,.$$

Recall that $\varphi_t := \phi(x_t, a_t)$. Then, summing over $t = 1, \ldots, T$ and using that by assumption

$$|\Delta_t| \leq \big|\langle \phi(x_t, a_t^*), \theta^* \rangle\big| + \big|\langle \phi(x_t, a_t), \theta^* \rangle\big| \leq 2 \sup_{x \in \mathcal{X}, a \in \mathcal{A}_t} |\langle \phi(x, a), \theta^* \rangle| \leq 2\,,$$

we can write the cumulative regret as

$$\begin{aligned} \sum_{t=1}^T \Delta_t &\leq \sqrt{T \textstyle\sum_{t=1}^T \Delta_t^2} && \leftarrow \quad \text{Jensen's inequality} \\ &\leq 2\sqrt{T \textstyle\sum_{t=1}^T \min\{\|\varphi_t\|_{V_{t-1}^{-1}}^2 \beta_t(\delta)^2, 1\}} \\ &\leq 2\beta_T(\delta)\sqrt{T \textstyle\sum_{t=1}^T \min\{\|\varphi_t\|_{V_{t-1}^{-1}}^2, 1\}} && \leftarrow \quad 1 \leq \beta_t(\delta) \leq \beta_T(\delta) \\ &\leq 2\beta_T(\delta)\sqrt{T \textstyle\sum_{t=1}^T \log\left(1 + \|\varphi_t\|_{V_{t-1}^{-1}}^2\right)} && \leftarrow \quad \min(u, 1) \leq 2\log(1+u),\ \forall u > 0\,. \end{aligned} \tag{26}$$

Now we will use the kernel trick to obtain a formulation of $\varphi_t^\top V_{t-1}^{-1} \varphi_t$ using gram matrices. Define $s_t := (x_t, a_t)$ and $\mathcal{S}_t := (s_i)_{1 \leq i \leq t}$ the historical data. For any $l \geq 1$ and $\mathcal{S} \in (\mathcal{X} \times \mathcal{A})^l$, we also denote by $K_{\mathcal{S}}(s')$ the kernel column vector $[k(s_1, s'), \ldots, k(s_l, s')]^\top$ of size $|\mathcal{S}| = l$. Specifically, we have $K_{\mathcal{S}_{t-1}}(s_t) := [k(s_1, s_t), \ldots, k(s_{t-1}, s_t)]^\top = \Phi_{t-1}\varphi_t \in \mathbb{R}^t$. When multiplying $V_{t-1} := \Phi_{t-1}^*\Phi_{t-1} + \lambda I$ by φ_t on the right, we can express

$$\begin{aligned} & V_{t-1}\varphi_t = \Phi_{t-1}^* K_{\mathcal{S}_{t-1}}(s_t) + \lambda\varphi_t, \\ \Rightarrow \quad & \varphi_t = V_{t-1}^{-1}\Phi_{t-1}^* K_{\mathcal{S}_{t-1}}(s_t) + \lambda V_{t-1}^{-1}\varphi_t \\ \Rightarrow \quad & \varphi_t = \Phi_{t-1}^*(K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) + \lambda V_{t-1}^{-1}\varphi_t\,, \end{aligned}$$

where the last equation is by Eq. (20). Thus, multiplying now by $\varphi_t^\top$ on the left and using $\varphi_t^\top \Phi_{t-1}^* = K_{\mathcal{S}_{t-1}}(s_t)$ entails

$$\varphi_t^\top \varphi_t = K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) + \lambda \varphi_t^\top V_{t-1}^{-1} \varphi_t \,.$$

Therefore, reorganizing the terms ans recognizing $\|\varphi_t\|^2_{V_{t-1}^{-1}} = \varphi_t^\top V_{t-1}^{-1}\varphi_t$ and $k(s_t, s_t) = \varphi_t^\top \varphi_t$, we can write

$$\begin{aligned}
1+\|\varphi_t\|^2_{V_{t-1}^{-1}} &= 1 + \varphi_t^\top V_{t-1}^{-1}\varphi_t \\
&= \frac{\lambda + k(s_t,s_t)}{\lambda} - \frac{1}{\lambda} K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1}+\lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) \\
&= \frac{\lambda + k(s_t,s_t)}{\lambda} \left(1 - K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1}+\lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) \big(\lambda + k(s_t,s_t)\big)^{-1}\right) \\
&= \frac{\lambda + k(s_t,s_t)}{\lambda} \det\left(1 - K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1}+\lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) \big(\lambda + k(s_t,s_t)\big)^{-1}\right) \\
&= \frac{\lambda + k(s_t,s_t)}{\lambda} \det\left(I - (K_{t-1}+\lambda I)^{-1/2} K_{\mathcal{S}_{t-1}}(s_t) \big(\lambda + k(s_t,s_t)\big)^{-1} K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1}+\lambda I)^{-1/2}\right),
\end{aligned}$$

where the last equality follows by the matrix determinant lemma $\det(I + AB^\top) = \det(I + B^\top A)$ if A and B are n-by-m matrices. Then, $1 + \|\varphi_t\|^2_{V_{t-1}^{-1}}$ equals

$$\begin{aligned}
&\frac{\lambda + k(s_t,s_t)}{\lambda} \det\left((K_{t-1}+\lambda I)^{-1/2}\Big(K_{t-1} + \lambda I - K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + k(s_t,s_t)\big)^{-1} K_{\mathcal{S}_{t-1}}(s_t)^\top\Big)(K_{t-1}+\lambda I)^{-1/2}\right) \\
&\qquad = \frac{\lambda + k(s_t,s_t)}{\lambda} \frac{\det\left(K_{t-1} + \lambda I - K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + k(s_t,s_t)\big)^{-1} K_{\mathcal{S}_{t-1}}(s_t)^\top\right)}{\det(K_{t-1}+\lambda I)} \,.
\end{aligned}$$

Now, using that

$$K_t + \lambda I = \begin{bmatrix} K_{t-1} + \lambda I & K_{S_{t-1}}(s_t) \\ K_{S_{t-1}}(s_t)^\top & k(s_t,s_t) + \lambda \end{bmatrix},$$

by the block matrix determinant formula

$$\det\big(K_t + \lambda I\big) = (k(s_t,s_t) + \lambda) \det\left(K_{t-1} + \lambda I - K_{S_{t-1}}(s_t)(k(s_t,s_t)+\lambda)^{-1} K_{S_{t-1}}(s_t)^\top\right)$$

we finally get

$$1 + \|\varphi_t\|^2_{V_{t-1}^{-1}} = \frac{1}{\lambda} \frac{\det(K_t + \lambda I)}{\det(K_{t-1} + \lambda I)} \,. \tag{27}$$

Note here that contrary to the proof in Lattimore and Szepesvári (2020), we used here computations using the gram matrix K_t instead of the V_t which lives in the feature space that can be infinite dimensional.

Taking the log and summing over $t = 1, \ldots, T$ telescopes

$$\sum_{t=1}^{T} \log\left(1 + \|\varphi_t\|^2_{V_{t-1}^{-1}}\right) = \log\left(\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)\right) \leq \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T) \,,$$

where we used the Proposition 3.1 for the last inequality and that . Substituting into the regret bound (26) together with $\beta_T(\delta) \leq \beta_{T+1}(\delta)$ entails with probability at least $1 - T\delta$

$$\sum_{t=1}^{T} \Delta_t \leq 2\beta_{T+1}(\delta) \sqrt{T\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)} \,.$$

Choosing $\delta = 1/T^2$, taking the expectation $R_T = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_t\right]$ and using $|\Delta_t| \leq 2$ concludes. □

We now provide a proof for the Corollary that gives out the convergence speed of the K-UCB algorithm with the capacity condition assumption.

B.3 Proof of Corollary 3.1

Corollary 3.1. *Assuming the capacity condition $d_{\text{eff}} \leq (T/\lambda)^\alpha$ for $0 \leq \alpha \leq 1$, the regret of K-UCB is bounded as $R_T \lesssim T^{\frac{1+3\alpha}{2+2\alpha}}$ with an optimal $\lambda \approx T^{\frac{\alpha}{1+\alpha}}$.*

Proof. Starting from $R_T \lesssim \sqrt{T}\Big(\sqrt{\lambda d_{\text{eff}}(\lambda)} + d_{\text{eff}}(\lambda)\Big)$ and assuming the capacity condition $d_{\text{eff}}(\lambda) \lesssim \left(\frac{T}{\lambda}\right)^\alpha$ for some $\alpha \in (0,1)$,

$$R_T \lesssim \sqrt{T}\Big(\sqrt{T^\alpha \lambda^{1-\alpha}} + T^\alpha \lambda^{-\alpha}\Big)\,.$$

Minimizing in $\lambda > 0$ entails

$$\sqrt{T^\alpha \lambda^{1-\alpha}} = T^\alpha \lambda^{-\alpha} \quad \Rightarrow \quad \lambda^* = T^{\frac{\alpha}{1+\alpha}}\,,$$

which yields for $\lambda = \lambda^*$

$$R_T \lesssim T^{\frac{1}{2}+\alpha-\frac{\alpha^2}{1+\alpha}} = T^{\frac{1+3\alpha}{2+2\alpha}}\,.$$

□

C Proofs of Section 4: Efficient Kernel-UCB

Let us start by recalling the setting and the notation of this section. Let $\mathcal{Z}_t \subseteq \mathcal{S}_t$, $\tilde{\mathcal{H}}_t := \text{Span}\big\{\phi(s), s \in \mathcal{Z}_t\big\}$ be the corresponding linear subspace of $\mathcal{H}$, and $P_t : \mathcal{H} \to \tilde{\mathcal{H}}_t$ be the Euclidean projection onto $\mathcal{H}$ so that $\tilde{\mathcal{H}}_t = \{P_t\varphi, \varphi \in \mathcal{H}\}$. The EK-UCB algorithm also builds an estimator

$$\tilde{\theta}_{t-1} \in \operatorname*{arg\,min}_{\theta \in \tilde{\mathcal{H}}_{t-1}} \left\{ \sum_{s=1}^{t-1} \left(\langle \theta, \phi(x_s, a_s)\rangle - r_s\right)^2 + \lambda \|\theta\|^2 \right\} \in \tilde{\mathcal{H}}_{t-1}\,, \tag{28}$$

and uses the confidence set $\tilde{C}_t := \big\{\theta \in \mathcal{H} : \|\theta - \tilde{\theta}_{t-1}\|_{\tilde{V}_{t-1}} \leq \tilde{\beta}_t(\delta)\big\}$. We define $\tilde{V}_t := \sum_{s=1}^t (P_t\varphi_s) \otimes (P_t\varphi_s) + \lambda I$, that we rewrite $\tilde{V}_t = P_t F_t P_t + \lambda I$ where $\Phi_t^* = [\varphi_1, \dots, \varphi_t]$ and $F_t = \Phi_t^*\Phi_t$. Recalling the notation, $Y_t := (r_1, \dots, r_t)^\top$, we then obtain that $\tilde{\theta}_t = P_t \tilde{V}_t^{-1} P_t \Phi_t^* Y_t$. We recall the definition $\mu_t := \big\|(I - P_t)F_t^{1/2}\big\|^2$.

C.1 Proof of Lemma 4.1

The following lemma serves to compute the distance of the center $\tilde{\theta}_t$ to any point in the ellipsoid in the projected space $\tilde{\mathcal{H}}_t$. Note that the norm uses the geometry induced by the direction matrix $\tilde{V}_t$.

Lemma 4.1. *Let $\delta \in (0,1)$. Assume that $\sup_{s \in \mathcal{X}\times\mathcal{A}} k(s,s) \leq \kappa^2$. Then, with probability $1-\delta$, for all $t \geq 1$*

$$\begin{aligned}
\|\tilde{\theta}_t - \theta^*\|_{\tilde{V}_t} &\leq \left(\sqrt{\lambda} + \sqrt{\mu_t}\right)\|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\det\Big(\frac{K_t + \lambda I}{\lambda}\Big)} \\
&\leq \left(\sqrt{\lambda} + \sqrt{\mu_t}\right)\|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)} \quad := \quad \tilde{\beta}_{t+1}(\delta)\,,
\end{aligned}$$

where $\|\theta\|_V^2 = \theta^\top V \theta$.

Proof. Let $t \geq 1$. Note that $P_t V_t P_t = P_t(F_t + \lambda I)P_t = P_t\tilde{V}_t = \tilde{V}_t P_t$ and consequently as well $P_t\tilde{V}_t^{-1} = \tilde{V}_t^{-1}P_t$. We can write with $H_t := (\varepsilon_1, \dots, \varepsilon_t)^\top$,

$$\begin{aligned}
\tilde{\theta}_t &= P_t\tilde{V}_t^{-1}P_t\Phi_t^* Y_t \\
&= \tilde{V}_t^{-1}P_t\Phi_t^* Y_t && \leftarrow \quad P_t\tilde{V}_t^{-1} = \tilde{V}_t^{-1}P_t \\
&= \tilde{V}_t^{-1}P_t\Phi_t^*(\Phi_t\theta^* + H_t) \\
&= \tilde{V}_t^{-1}P_t F_t P_t\theta^* + \tilde{V}_t^{-1}P_t F_t(I - P_t)\theta^* + \tilde{V}_t^{-1}P_t\Phi_t^* H_t \\
&= \theta^* - \lambda\tilde{V}_t^{-1}\theta^* + \tilde{V}_t^{-1}P_t F_t(I - P_t)\theta^* + \tilde{V}_t^{-1}P_t\Phi_t^* H_t\,.
\end{aligned}$$

To obtain later on the norm $\|\tilde{\theta}_t - \theta^*\|_{\tilde{V}_t}$, we multiply by $\tilde{V}_t^{1/2}$ on the left

$$\tilde{V}_t^{1/2}(\tilde{\theta}_t - \theta^*) = -\underbrace{\lambda \tilde{V}_t^{-1/2}\theta^*}_{(i)} + \underbrace{\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*}_{(ii)} + \underbrace{\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t}_{(iii)} . \tag{29}$$

We then compute each norm separately.

(i) Since $\tilde{V}_t = P_t F_t P_t + \lambda I$, all its eigenvalues are larger than λ. Thus, $\tilde{V}_t^{-1/2} \preccurlyeq \lambda^{-1/2} I$, which implies

$$\left\|\lambda \tilde{V}_t^{-1/2}\theta^*\right\| \le \sqrt{\lambda}\|\theta^*\| . \tag{30}$$

(ii) We write $\left\|\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*\right\| = \left\|\tilde{V}_t^{-1/2} P_t F_t^{1/2} F_t^{1/2} (I - P_t)\theta^*\right\|$ and recall $\tilde{V}_t = P_t F_t^{1/2} F_t^{1/2} P_t + \lambda I$ therefore $\tilde{V}_t^{1/2} \succcurlyeq P_t F_t^{1/2}$, which entails

$$\left\|\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*\right\| \le \left\|F_t^{1/2}(I - P_t)\theta^*\right\| \le \sqrt{\mu_t}\|\theta^*\| , \tag{31}$$

where we recall that $\mu_t := \left\|(I - P_t)F_t^{1/2}\right\|^2$.

(iii) Let us upper-bound the norm of the last term

$$\begin{aligned}\left\|\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t\right\| &\le \left\|\tilde{V}_t^{-1/2} P_t V_t^{1/2}\right\| \left\|V_t^{-1/2}\Phi_t^* H_t\right\| \\ &\le \left\|\tilde{V}_t^{-1/2} P_t V_t^{1/2}\right\| \sqrt{2\log\frac{1}{\delta} + \log\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)} ,\end{aligned} \tag{32}$$

with probability at least $1 - \delta$, where the last inequality follows from the same analysis as (24). Then, using that $P_t V_t P_t = P_t F_t P_t + \lambda P_t = \tilde{V}_t + \lambda(P_t - I)$, we have

$$\begin{aligned}\left\|\tilde{V}_t^{-1/2} P_t V_t^{1/2}\right\|^2 = \left\|\tilde{V}_t^{-1/2} P_t V_t P_t \tilde{V}_t^{-1/2}\right\| &= \left\|\tilde{V}_t^{-1/2}\big(\tilde{V}_t + \lambda(P_t - I)\big)\tilde{V}_t^{-1/2}\right\| \\ &= \left\|I + \lambda \tilde{V}_t^{-1/2}(P_t - I)\tilde{V}_t^{-1/2}\right\| \le 1 + \lambda\left\|\tilde{V}_t^{-1/2}\right\|^2 \left\|P_t - I\right\| \le 2 ,\end{aligned}$$

where the last inequality is because $\left\|P_t - I\right\| \le 1$ and $\left\|\tilde{V}_t^{-1/2}\right\| \le \lambda^{-1/2}$. Therefore, substituting into Inequality (32) yields

$$\left\|\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t\right\| \le \sqrt{4\log\frac{1}{\delta} + 2\log\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)} , \tag{33}$$

with probability at least $1 - \delta$.

Finally, combining (30), (31), and (33) with Equation (29) concludes

$$\begin{aligned}\left\|\tilde{\theta}_t - \theta\right\|_{\tilde{V}_t} &\le \lambda\left\|\tilde{V}_t^{-1/2}\theta^*\right\| + \left\|\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*\right\| + \left\|\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t\right\| \\ &\le \big(\sqrt{\lambda} + \sqrt{\mu_t}\big)\left\|\theta^*\right\| + \sqrt{4\log\frac{1}{\delta} + 2\log\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)} .\end{aligned}$$

The second line of the statement follows from Proposition 3.1. □

C.2 Proof of Theorem 4.1

Theorem 4.1. *Let $T \ge 1$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x,a), \theta^* \rangle| \le 1$ for all $a \in \bigcup_{t=1}^T \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$ then the EK-UCB rule in Eq. (4.1) with $\tilde{\mathcal{C}}_t$ defined in Eq. (10), with $m = |\mathcal{Z}_t|$ dictionary updates, satisfies the pseudo-regret bound*

$$R_T \lesssim \sqrt{T}\left(\sqrt{\frac{\mu m}{\lambda}} + \sqrt{d_{\text{eff}}}\right)\left(\sqrt{\lambda} + \sqrt{\mu} + \sqrt{d_{\text{eff}}}\right) .$$

In particular, the choice $\mu = \lambda$ yields $m \lesssim d_{\text{eff}}$ and

$$R_T \lesssim \sqrt{T}\big(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\big) .$$

Proof. Let $\delta > 0$. By Lemma 4.1, with probability $1-\delta$,

$$\forall t \geq 1, \quad \theta^* \in \tilde{C}_t \,. \tag{34}$$

Let us recall and start from the definition of the regret

$$R_T := \mathbb{E}\left[\sum_{t=1}^{T} \Delta_t\right], \qquad \text{where} \quad \Delta_t := \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle \quad \text{and} \quad a_t^* := \max_{a \in \mathcal{A}_t} \langle \phi(x_t, a), \theta^* \rangle \,.$$

Step 1: Small instantaneous regrets under the event (34). Assume that (34) holds and define

$$\tilde{\rho}_t \in \arg\max_{\theta \in \tilde{C}_t} \{\langle \phi(x_t, a_t), \theta \rangle\} \,.$$

Note here that the use of the original feature map allows us to not have any misspecified term that would have been incurred if the projected feature map was used instead instead with $\langle \phi(x_t, a_t^*), \theta^* \rangle = \langle P_t \phi(x_t, a_t^*), \theta^* \rangle + \langle (I - P_t)\phi(x_t, a_t^*), \theta^* \rangle)$ in the upper bound expression.

Now given that $\theta^* \in \tilde{C}_t$ and $a_t \in \arg\max_{a \in \mathcal{A}} \text{EK-UCB}_t(a)$, we have

$$\langle \phi(x_t, a_t^*), \theta^* \rangle \leq \max_{\theta \in \tilde{C}_t} \{\langle \phi(x_t, a_t^*), \theta \rangle\} = \text{EK-UCB}_t(a_t^*) \leq \text{EK-UCB}_t(a_t) = \max_{\theta \in \tilde{C}_t} \{\langle \phi(x_t, a_t), \theta \rangle\} = \langle \phi(x_t, a_t), \tilde{\rho}_t \rangle \,.$$

Therefore,

$$\Delta_t := \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle \leq \langle \phi(x_t, a_t), \tilde{\rho}_t - \theta^* \rangle \leq \|\varphi_t\|_{\tilde{V}_{t-1}^{-1}} \|\tilde{\rho}_t - \theta^*\|_{\tilde{V}_{t-1}} \leq 2\|\varphi_t\|_{\tilde{V}_{t-1}^{-1}} \tilde{\beta}_t(\delta) \,. \tag{35}$$

Then, summing over $t = 1, \ldots, T$ and using $|\Delta_t| \leq 2$ and $\tilde{\beta}_T(\delta) \geq \beta_t(\delta) \geq 1$, we get

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t &\leq \sqrt{T \sum_{t=1}^{T} \Delta_t^2} && \leftarrow \quad \text{Jensen's inequality} \\
&\leq 2\sqrt{T \sum_{t=1}^{T} \min\left\{\|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} \tilde{\beta}_t(\delta)^2, 1\right\}} && \leftarrow \quad |\Delta_t| \leq 2 \text{ and (35)} \qquad (36) \\
&\leq 2\tilde{\beta}_T(\delta)\sqrt{T \sum_{t=1}^{T} \min\left\{\|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1\right\}} && \leftarrow \quad 1 \leq \beta_t(\delta) \leq \beta_T(\delta) \,. \qquad (37)
\end{aligned}$$

Note now that

$$\begin{aligned}
\min\left\{\|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1\right\} &\leq 2\min\left\{\|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} + \|(I-P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1\right\} \\
&\leq 2\min\left\{\|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1\right\} + 2\|(I-P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} \\
&\leq 4\log\left(1 + \|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}\right) + 2\|(I-P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} \,. \qquad (38)
\end{aligned}$$

The first term can be upper-bounded similarly to (27). First, note that since $P_s = P_s P_{t-1}$ for any $1 \leq s \leq t-1$,

$$\tilde{V}_{t-1} := \sum_{s=1}^{t-1} (P_{t-1}\varphi_s) \otimes (P_{t-1}\varphi_s) + \lambda I \succcurlyeq \sum_{s=1}^{t-1} (P_s\varphi_s) \otimes (P_s\varphi_s) + \lambda I =: \tilde{W}_{t-1}$$

which implies $\tilde{V}_{t-1}^{-1} \preccurlyeq \tilde{W}_{t-1}^{-1}$ and thus

$$\log\left(1 + \|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}\right) \leq \log\left(1 + \|P_t\varphi_t\|^2_{\tilde{W}_{t-1}^{-1}}\right) \,. \tag{39}$$

Now, recalling that $V_{t-1} := \sum_{s=1}^{t-1} \varphi_s \otimes \varphi_s + \lambda I$, following the same analysis as for (27), replacing φ_s with $P_s\varphi_s$ for all $s = 1, \dots, t$, we get

$$1 + \|P_t\varphi_t\|^2_{\tilde{W}^{-1}_{t-1}} = \frac{1}{\lambda}\frac{\det(\tilde{K}_t + \lambda I)}{\det(\tilde{K}_{t-1} + \lambda I)}\,,$$

where $\tilde{K}_t \in \mathbb{R}^{t\times t}$ is the kernel matrix such that $\left[\tilde{K}_t\right]_{ij} = \langle P_i\varphi_i, P_j\varphi_j\rangle$ for all $1 \le i, j \le t$. Together with Inequalities (38) and (39), and summing over $t = 1, \dots, T$, it yields

$$\begin{aligned}\sum_{t=1}^{T} \min\left\{\|\varphi_t\|^2_{\tilde{V}^{-1}_{t-1}}, 1\right\} &\le 4\sum_{t=1}^{T} \log\left(\frac{1}{\lambda}\frac{\det(\tilde{K}_t + \lambda I)}{\det(\tilde{K}_{t-1} + \lambda I)}\right) + 2\sum_{t=1}^{T} \|(I - P_t)\varphi_t\|^2_{\tilde{V}^{-1}_{t-1}} \\ &\le 4\log\left(\frac{\det(\tilde{K}_T + \lambda I)}{\lambda^t}\right) + 2\sum_{t=1}^{T} \|(I - P_t)\varphi_t\|^2_{\tilde{V}^{-1}_{t-1}}\,. \end{aligned} \tag{40}$$

We now upper-bound the second term in the right-hand-side. Denoting by $1 = \tau_1 < \tau_2 < \cdots < \tau_m \le T$ the indexes in time when the projection is updated, i.e., $P_t = P_{\tau_i}$ for all $t \in \{\tau_i, \dots, \tau_{i+1} - 1\}$, we can write

$$\begin{aligned}\sum_{t=1}^{T} \left\|(I - P_t)\varphi_t\right\|^2 &= \sum_{i=1}^{m}\sum_{t=\tau_i}^{\tau_{i+1}-1} \left\|(I - P_{\tau_i})\varphi_t\right\|^2 \\ &= \sum_{i=1}^{m}\sum_{t=\tau_i}^{\tau_{i+1}-1} \mathrm{Tr}\left((I - P_{\tau_i})\varphi_t \otimes \varphi_t(I - P_{\tau_i})\right) \\ &= \sum_{i=1}^{m} \mathrm{Tr}\left((I - P_{\tau_i})\Big(\sum_{t=\tau_i}^{\tau_{i+1}-1} \varphi_t \otimes \varphi_t\Big)(I - P_{\tau_i})\right) \\ &= \sum_{i=1}^{m} \mathrm{Tr}\left((I - P_{\tau_{i+1}-1})\Big(\sum_{t=\tau_i}^{\tau_{i+1}-1} \varphi_t \otimes \varphi_t\Big)(I - P_{\tau_{i+1}-1})\right) \\ &\le \sum_{i=1}^{m} \mathrm{Tr}\left((I - P_{\tau_{i+1}-1})\Big(\sum_{t=1}^{\tau_{i+1}-1} \varphi_t \otimes \varphi_t\Big)(I - P_{\tau_{i+1}-1})\right) \\ &\le \sum_{l=1}^{m} \mu_{\tau_{i+1}-1} \le m\mu\,,\end{aligned}$$

where the last inequality follows from Prop. 4.1. Therefore, using that $\tilde{V}^{-1}_{t-1} \preccurlyeq \lambda^{-1} I$, from (38) we get

$$\sum_{t=1}^{T} \min\left\{\|\varphi_t\|^2_{\tilde{V}^{-1}_{t-1}}, 1\right\} \le 4\log\left(\frac{\det(\tilde{K}_T + \lambda I)}{\lambda^t}\right) + \frac{2m\mu}{\lambda}\,.$$

Substituting into Inequality (37) entails

$$\begin{aligned}\sum_{t=1}^{T} \Delta_t &\le 2\tilde{\beta}_T(\delta)\sqrt{T\left(4\log\left(\frac{\det(\tilde{K}_T + \lambda I)}{\lambda^t}\right) + \frac{2m\mu}{\lambda}\right)} \\ &\le 2\tilde{\beta}_T(\delta)\sqrt{T\left(4\log\det\left(\frac{K_T + \lambda I}{\lambda}\right) + \frac{2m\mu}{\lambda}\right)} \\ &\le 2\tilde{\beta}_T(\delta)\sqrt{T\left(4\log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\mathrm{eff}} + \frac{2m\mu}{\lambda}\right)}\end{aligned}$$

where the last inequality is by Prop. 3.1 and where we recall

$$\tilde{\beta}_T(\delta) \le \left(\sqrt{\lambda} + \sqrt{\mu}\right)\|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\mathrm{eff}}}\,.$$

Choosing $\delta = 1/T$ and taking the expectation concludes

$$R_T \leq 2 + 2\tilde{\beta}_T(1/T)\sqrt{T\left(4\log\left(e + \frac{eT\kappa^2}{\lambda}\right)d_{\text{eff}} + \frac{2m\mu}{\lambda}\right)}$$
$$\lesssim \left((\sqrt{\lambda} + \sqrt{\mu})\|\theta^*\| + \sqrt{d_{\text{eff}}}\right)\sqrt{T\left(d_{\text{eff}} + \frac{m\mu}{\lambda}\right)}\,.$$

In particular, for the choice $\mu = \lambda$, by Prop. 4.1, the dictionary is at most of size $m \lesssim d_{\text{eff}}$ with high probability. □

C.3 Proof of Cor. 4.1

Corollary 4.1. *Assuming the capacity condition $d_{\text{eff}} \leq (T/\lambda)^\alpha$ for $0 \leq \alpha \leq 1$. Let $1 \leq m \leq T^{\alpha/(1+\alpha)}$, under the assumptions of Thm. 4.1, the regret of EK-UCB satisfies*

$$R_T \lesssim \begin{cases} Tm^{\frac{\alpha-1}{2\alpha}} & \text{if } m \leq T^{\frac{\alpha}{1+\alpha}} \\ T^{\frac{1+3\alpha}{2+2\alpha}} & \text{otherwise} \end{cases}$$

for the choice $\lambda = \mu = Tm^{-1/\alpha}$. Furthermore, the algorithm runs in $O(Tm)$ space complexity and $O(CTm^2)$ time complexity.

We start from the regret bound of Theorem 4.1, which, forgetting all dependencies that do not depend on T, for the choice $\mu = \lambda$ yields

$$R_T \lesssim \sqrt{T}\left(\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\right).$$

Under the capacity condition $d_{\text{eff}}(\lambda, T) \leq (T/\lambda)^\alpha$, it entails

$$R_T \lesssim \sqrt{T}\left(\lambda^{\frac{1-\alpha}{2}}T^{\frac{\alpha}{2}} + \lambda^{-\alpha}T^\alpha\right) = T^{\frac{1}{2}}\left(T^{1/2}m^{\frac{\alpha-1}{2\alpha}} + m\right) = Tm^{\frac{\alpha-1}{2\alpha}} + \sqrt{T}m\,,$$

where we replaced $\lambda = Tm^{-1/\alpha}$. Optimizing in m, we retrieve the original rate $R_T \lesssim T^{\frac{1+3\alpha}{2+2\alpha}}$ for a dictionary of size $m = T^{\frac{\alpha}{1+\alpha}} \ll T$. Note that a larger dictionary is not necessary in theory since it only hurts both the theoretical rate and the computational complexity. For a smaller dictionary, the first term is predominant and yields a regret of order $\mathcal{O}\left(Tm^{\frac{\alpha-1}{2\alpha}}\right)$, highlighting a trade-off between the complexity which increases with m and the regret which decreases.

D Details on the comparison of the regret bounds of CGP-UCB, SupKernelUCB, and K-UCB

In this appendix, we first detail why we can compare the regrets of CGP-UCB (Krause and Ong, 2011), SupKernelUCB (Valko et al., 2013) and K-UCB as shown in Table 1. We compare the quantities $\tilde{d}, \gamma$ and d_{eff} that appear in the regret bound of the literature (Valko et al., 2013; Calandriello et al., 2019; Krause and Ong, 2011). We show that they are essentially equivalent up to logarithmic factors. We recall first their definitions: for any $t \geq 0$ and $\lambda > 0$

$$\gamma(\lambda, t) = \frac{1}{2}\log\left(\det\left(I + \frac{1}{\lambda}K_t\right)\right)$$
$$\tilde{d}(\lambda, t) = \min\{j : j\lambda\log T \geq \sum_{k>j}\lambda_k(K_t)\}$$
$$d_{\text{eff}}(\lambda, T) = \text{Tr}(K_T(K_T + \lambda I_T)^{-1})\,.$$

We start by proving the first equality (up to logarithmic factors) $d_{\text{eff}}(\lambda, t) \lesssim \gamma(\lambda, t) \lesssim d_{\text{eff}}(\lambda, t)$. We first obtain that $\gamma(\lambda, t) \lesssim d_{\text{eff}}$ with Proposition 3.1. Next, to prove that $d_{\text{eff}} \lesssim \gamma(\lambda, t)$, we prove that for all $x > -1$ $\frac{x}{x+1} \leq \log(1+x)$ by writing for $x > -1$, $h(x) = \frac{x}{x+1} - \log(1+x)$ studying h' and $h(0)$. Therefore,

$$d_{\text{eff}}(\lambda, t) = \text{Tr}(K_t(Kt + \lambda I)^{-1}) = \sum_{k=1}^{t}\frac{\frac{\lambda_k}{\lambda}}{\frac{\lambda_k}{\lambda} + 1} \leq \sum_{k=1}^{t}\log(1 + \frac{\lambda_k}{\lambda}) = \gamma(\lambda, t)$$

Next we detail that $\tilde{d}(\lambda,t) \lesssim \gamma(\lambda,t) \lesssim \tilde{d}(\lambda,t)$. First, Valko et al. (2013) shows that $\tilde{d}(\lambda,t) \lesssim \gamma(\lambda,t)$. Second, to prove $\gamma(\lambda,t) \lesssim \tilde{d}(\lambda,t)$, we write

$$\sum_{k=1}^{t} \log\left(1+\frac{\lambda_k}{\lambda}\right) \leq \sum_{k>\tilde{d}} \frac{\lambda_k}{\lambda} + \sum_{k\leq\tilde{d}} \log(1+\frac{\lambda_k}{\lambda}) \leq \tilde{d}(\lambda,t)\log(t) + \tilde{d}(\lambda,t)\log\left(\frac{\lambda_1}{\lambda}\right),$$

where we used $\log(1+x)\leq x$ on the first term of the sum decomposition, and λ_1 the first and larger eigenvalue of the matrix K_t. Then, using $\lambda_1(K_t) \leq \mathrm{Tr}(K_t) = \sum_{k=1}^{t} \|\varphi_k\|^2 \leq t\kappa^2$, we subsequently obtain $\gamma(\lambda,t) \leq \tilde{d}\left(\log(T) + \log\left(\frac{t\kappa^2}{\lambda}\right)\right)$ which concludes the inequality.

E Algorithm Implementations

Here we give details on the implementations of the contextual kernel UCB algorithms as well as our EK-UCB.

E.1 Kernel UCB algorithm – Implementation details

Let us write $s_{t,a} := (x_t, a)$ and by abbreviation $s_i := (x_i, a_i)$, let us write the historical data $\mathcal{S}_t = (s_i)_{1\leq i\leq t}$. Let us recall $\Phi_t^* = [\varphi_1, \ldots, \varphi_t]$ where $\varphi_i = \phi(x_i,a_i) = \phi(s_i)$ and $K_{\mathcal{S}_t}(s) = \Phi_t\phi(s) = [k(s_1,s), \ldots k(s_t,s)]^\top$. We write $F_t = \Phi_t^*\Phi_t$ and the gram matrix $K_t = \Phi_t\Phi_t^*$. As in (Valko et al., 2013):

$$\begin{aligned}(\Phi_t^*\Phi_t + \lambda I)\Phi_t^* &= \Phi_t^*(\Phi_t\Phi_t^* + \lambda I)\\ (F_t + \lambda I)\Phi_t^* &= \Phi_t^*(K_t + \lambda I)\\ \Phi_t^*(K_t+\lambda I)^{-1} &= (F_t+\lambda I)^{-1}\Phi_t^*.\end{aligned}$$

Expression of the mean $\hat{\mu}_{t,a} = \langle\hat{\theta}_t, \varphi_{t,a}\rangle$. For the mean expression recall that we have: $\hat{\mu}_{t,a} = \langle\hat{\theta}_{t-1}, \varphi_{t,a}\rangle = \varphi_{t,a}^\top\hat{\theta}_{t-1}$ and $\hat{\theta}_t = V_t^{-1}\Phi_t^*Y_t$. Therefore,

$$\hat{\mu}_{t,a} = \varphi_{t,a}^\top\hat{\theta}_{t-1} = \varphi_{t,a}^\top\Phi_{t-1}^*(K_{t-1}+\lambda I)^{-1}Y_{t-1} = K_{\mathcal{S}_{t-1}}(s_{t,a})^\top(K_{t-1}+\lambda I)^{-1}Y_{t-1}.$$

Expression of the standard deviation $\hat{\sigma}_{t,a} = \|\varphi_{t,a}\|_{V_{t-1}^{-1}}$. When multiplying by $\varphi_{t,a} := \phi(x_t,a)$ on the right and then by $\varphi_{t,a}^\top$ on the left

$$\begin{aligned}(\Phi_{t-1}^*\Phi_{t-1}+\lambda I)\varphi_{t,a} &= \Phi_{t-1}^*K_{\mathcal{S}_{t-1}}(s_{t,a}) + \lambda\varphi_{t,a}\\ \varphi_{t,a} &= \Phi_{t-1}^*(K_{t-1}+\lambda I)^{-1}K_{\mathcal{S}_{t-1}}(s_{t,a}) + \lambda(\Phi_{t-1}^*\Phi_{t-1}+\lambda I)^{-1}\varphi_{t,a}\\ \varphi_{t,a}^\top\varphi_{t,a} &= K_{\mathcal{S}_{t-1}}(s_{t,a})^\top(K_{t-1}+\lambda I)^{-1}K_{\mathcal{S}_{t-1}}(s_{t,a}) + \lambda\varphi_{t,a}^\top V_{t-1}^{-1}\varphi_{t,a}\\ \hat{\sigma}_{t,a} = \|\varphi_{t,a}\|_{V_{t-1}^{-1}} &= \frac{1}{\lambda}k(s_{t,a},s_{t,a}) - \frac{1}{\lambda}K_{\mathcal{S}_{t-1}}(s_{t,a})^\top(K_{t-1}+\lambda I)^{-1}K_{\mathcal{S}_{t-1}}(s_{t,a})\end{aligned}$$

This allows to compute the UCB rule with kernel representations as illustrated in Alg. 3.

Algorithm 3: Kernel UCB

Input: T the horizon, λ regularization and exploration parameters, k the kernel function
initialization;
$K_\lambda = \lambda, Y_0 = [r_0]$ where $r_0 = r(x_0,a_0)$ and a_0 is chosen randomly ;
for $t = 1$ *to* T **do**
  Observe context x_t ;
  Compute β_t ;
  Choose $a_t \leftarrow \arg\max_{a\in\mathcal{A}} \hat{\mu}_{t,a} + \beta_t\hat{\sigma}_{t,a}$;
    $\hat{\mu}_{t,a} \leftarrow K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1}Y_{t-1}$;
    $\hat{\sigma}_{t,a}^2 \leftarrow \frac{1}{\lambda}k(s_{t,a},s_{t,a}) - \frac{1}{\lambda}K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1}K_{\mathcal{S}_{t-1}}(s_{t,a})$;
  Observe reward r_t and update $Y_t \leftarrow [r_1, \ldots r_t]$;
  Update the translated gram matrix $K_\lambda \leftarrow [k(s_i,s_j)]_{1\leq i,j\leq t} + \lambda I$;
end

Since the kernel matrices are used instead of estimating and computing directly $\hat{\theta}_t$ and $\phi(x_t, a)$, we can use first-rank updates of the matrices K_t, since:

$$K_t = \begin{bmatrix} K_{t-1} & K_{\mathcal{S}_{t-1}}(s_{t,a}) \\ K_{\mathcal{S}_{t-1}}(s_{t,a})^\top & K(s_{t,a}, s_{t,a}) \end{bmatrix}.$$

It is then easy to use the Schur complement on the inverse K_λ^{-1}. Specifically, the update is performed as the following, with

$$\begin{aligned}
s &\leftarrow k(s_{t,a}, s_{t,a}) + \lambda - K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) \\
Z_{12} &\leftarrow -\frac{1}{s} K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} \\
Z_{21} &\leftarrow -\frac{1}{s} K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) \\
Z_{11} &\leftarrow K_\lambda^{-1} + \frac{1}{s} K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} \\
K_\lambda^{-1} &\leftarrow [Z_{11}, Z_{12}, Z_{21}, \frac{1}{s}].
\end{aligned}$$

Therefore, while inverting the full matrices would induce as full cost of $\mathcal{O}(CT^4)$, using first order updates with Schur complement allows to run the algorithm in $\mathcal{O}(CT^3)$, while using $\mathcal{O}(T^2)$ in space.

E.2 Efficient Kernel UCB algorithm – Implementation details

Instead of using the kernel trick as in the standard algorithm, the efficient Kernel UCB algorithm uses computations in the projected feature space. The key high-level idea is to use as much as possible computations in the projected space $\mathcal{H}_t = \text{span}\{\phi(z)\}_{z \in \mathcal{Z}_t}$ which is of dimension m_t and does not use implicit kernel representation of the whole data which are of size $t \times t$. Here, we detail the computations of the predicted mean and variance bound in the projected space.

At the time t we define the dictionary $\mathcal{Z}_t = \{z_1, \ldots, z_{m_t}\}$ of size $|\mathcal{Z}_t| = m_t$ and the $m_t \times m_t$ kernel matrix $K_{\mathcal{Z}_t} = [k(z_i, z_j)]_{1 \le i,j \le m_t}$, we also write $K_{\mathcal{Z}_t \mathcal{S}_t} = [k(z_i, s_j)]_{1 \le i \le m_t, 1 \le j \le t}$ the $m_t \times t$ matrix on anchor points and historical data $\mathcal{S}_t = \{s_i\}_{1 \le i \le t}$.

The following proposition provides closed-form formulas to implement EK-UCB (Alg. 2).

Proposition 4.2. *At any round t, by considering $s_{t,a} = (x_t, a)$, the mean and variance term of the EK-UCB rule (Alg 2) can be expressed as*[3]

$$\begin{aligned}
\Gamma_t &= K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1} \\
\Lambda_t &= \left(K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} K_{\mathcal{S}_{t-1}\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right)^{-1} \\
\tilde{\mu}_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \Lambda_t \Gamma_t \\
\Delta_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}^{-1}\right) K_{\mathcal{Z}_{t-1}}(s_{t,a}) \\
\tilde{\sigma}_{t,a}^2 &= \frac{1}{\lambda} k(s_{t,a}, s_{t,a}) + \Delta_{t,a}.
\end{aligned}$$

The algorithm then runs in a space complexity of $\mathcal{O}(Tm)$ and a time complexity of $\mathcal{O}(CTm^2)$.

Expression of the mean $\tilde{\mu}_{t+1,a} = \langle \tilde{\theta}_t, \varphi_{t+1,a} \rangle$. At a time $t+1$, we look for $\tilde{\theta} \in \tilde{\mathcal{C}}_{t+1}$ that we write $\tilde{\theta} = \alpha^\top K_{\mathcal{Z}_t}$ where $\alpha \in \mathbb{R}^{m_t}$. We can rewrite the optimization process in Eq. (9) as

$$\arg\min_{\alpha \in \mathbb{R}^{m_t}} \left\{ (K_{\mathcal{S}_t \mathcal{Z}_t} \alpha - Y_t)^\top (K_{\mathcal{S}_t \mathcal{Z}_t} \alpha - Y_t) + \lambda \alpha^\top K_{\mathcal{Z}_t \mathcal{Z}_t} \alpha \right\}$$

[3]Erratum: Note that the proposition slightly differs from the original one in the main document due to typos in the indexes that will be corrected in the final version of the manuscript.

which can be rewritten as

$$\arg\min_{\alpha\in\mathbb{R}^{m_t}} \Big\{ \alpha^\top K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t}\alpha - 2\alpha^\top K_{\mathcal{Z}_t\mathcal{S}_t} Y_t + \lambda \alpha^\top K_{\mathcal{Z}_t\mathcal{Z}_t}\alpha \Big\},$$

and can be solved in closed-form as

$$\alpha^* = (K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}})^{-1} K_{\mathcal{Z}\mathcal{S}_t} Y_t.$$

This eventually gives the expression $\tilde{\mu}_{t+1,a} = \alpha^\top K_{\mathcal{Z}_t s_{t+1}}$

$$\tilde{\mu}_{t+1,a} = K_{\mathcal{Z}_t}(s_{t+1,a})^\top (K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t})^{-1} K_{\mathcal{Z}_t\mathcal{S}_t} Y_t\,.$$

Expression of the standard deviation $\tilde{\sigma}_{t+1,a} = \|\varphi_{t+1,a}\|_{\tilde{V}_t^{-1}}$. When we look for the value of EK-UCB in Eq. (11), it is equivalent to have:

$$\text{EK-UCB}_{t+1}(a) = \max_{\theta\in\mathcal{H},\ \text{s.t}\ \|\theta-\tilde{\theta}_t\|_{\tilde{V}_t}\leq\beta} \langle\theta, \phi(s_{t+1,a})\rangle = \tilde{\mu}_{t+1,a} + \beta\tilde{\sigma}_{t+1,a}.$$

where the variance term $\tilde{\sigma}_{t+1,a}$ is solution to

$$\max_{\theta\in\mathcal{H}}\ \theta^\top\phi(s_{t+1,a})\,.$$
$$\text{s.t}\ \|\theta\|_{\tilde{V}_t}\leq 1.$$

Below, we abbreviate $s = s_{t+1,a} := (x_{t+1}, a)$ for simplicity of notation. We advocate that at each time t when we solve this maximization problem, θ lives in the finite dimensional space

$$\theta\in\mathcal{H}_{t+1,s} =: \text{Span}\left(K_{z_1},\dots K_{z_{m_t}}, K_s\right)\,,$$

where $K_z, K_s\in\mathcal{H}$ such that $K_z(z') = k(z,z')$ and $K_s(s') = k(s,s')$. To prove the above statement, following the Representer theorem proof, and $\mathcal{H}_{t+1,s}$ be the linear span of $K_{z_1},\dots K_{z_{m_t}}, K_s\in\mathcal{H}$. $\mathcal{H}_{t+1,s}$ is a finite dimensional subspace of $\mathcal{H}$, therefore any $\theta\in\mathcal{H}$ can be uniquely decomposed as

$$\theta = \theta_{\mathcal{H}_{t+1,s}} + \theta_\perp$$

with $\theta_{\mathcal{H}_{t+1,s}}\in\mathcal{H}_{t+1,s}$ and $\theta_\perp\perp\mathcal{H}_{t+1,s}$. $\mathcal{H}$ being a RKHS it holds that $\langle\theta_\perp,\phi(s)\rangle = \langle\theta_\perp, K_s\rangle = 0$ because $K_s\in\mathcal{H}_{t+1,s}$. Therefore, $\langle\theta, K_s\rangle = \langle\theta_{\mathcal{H}_{t+1,s}}, K_s\rangle$.

Now writing $\tilde{V}_t = P_tV_tP_t + \lambda(I-P_t)$, we have that $\|\theta\|_{\tilde{V}_t}$ can be written as

$$\|\theta\|_{\tilde{V}_t} = \theta^\top_{\mathcal{H}_{t+1,s}} P_tV_tP_t\theta_{\mathcal{H}_{t+1,s}} + \lambda\theta^\top_{\mathcal{H}_{t+1,s}}(I-P_t)\theta_{\mathcal{H}_{t+1,s}} + \lambda\theta^\top_\perp(I-P_t)\theta_\perp.$$

Therefore, $\|\theta_{\mathcal{H}_{t+1,s}}\|_{\tilde{V}_t}\leq\|\theta\|_{\tilde{V}_t}\leq 1$. The maximization domain $\{\theta\in\mathcal{H}\ \text{s.t}\ \|\theta\|_{\tilde{V}_t}\leq 1\}$ is thus included in $\{\theta\in\mathcal{H}_{t+1,s}\ \text{s.t}\ \|\theta_{\mathcal{H}_{t+1,s}}\|_{\tilde{V}_t}\leq 1\}$, while $\langle\theta,K_s\rangle = \langle\theta_{\mathcal{H}_{t+1,s}}, K_s\rangle$. Therefore, $\max_{\theta\in\mathcal{H}}\langle\theta,K_s\rangle = \max_{\theta\in\mathcal{H}_{t+1,s}}\langle\theta_{\mathcal{H}_{t+1,s}}, K_s\rangle$. Hence we can write the solution of the problem from Eq. (4.1) as

$$\theta_{\mathcal{H}_{t+1,s}} = \sum_{i=1}^{m_t}\alpha_i K_{z_i} + \alpha_{m_t+1}K_s,\ \alpha\in\mathbb{R}^{m_t},\quad \alpha_{m_t+1}\in\mathbb{R}.$$

We will write $\bar{K}_{\mathcal{Z}_t}\alpha = \sum_{i=1}^{m_t}\alpha_i K_{z_i}$ and therefore $\bar{K}^\top_{\mathcal{Z}_t}\bar{K}_{\mathcal{Z}_t} = K_{\mathcal{Z}_t,\mathcal{Z}_t}$ or even $\bar{K}^\top_{\mathcal{Z}_t}K_s = K_{\mathcal{Z}_t s}\in\mathbb{R}^{m_t}$.

Using this notation allows us to write $P_t\varphi_{t+1} = \sum_{i=1}^{m_t}\beta_i(s_{t+1,a})K_{z_i} = \bar{K}_{\mathcal{Z}_t}(K^{-1}_{\mathcal{Z}_t,\mathcal{Z}_t}K_{\mathcal{Z}_t}(s_{t+1,a}))$ where the β coefficient is obtained by solving with the minimization problem defined in the Nyström projection. Therefore when taking the projection $P_t:\mathcal{H}\to\mathbb{R}^{m_t}$ and the operator $\Phi_t:\mathbb{R}^t\to\mathcal{H}$ we can write $P_t\Phi_t = \bar{K}_{\mathcal{Z}_t}(K^{-1}_{\mathcal{Z}_t,\mathcal{Z}_t})K_{\mathcal{Z}_t\mathcal{S}_t}$.

Therefore when writing $\tilde{V}_t = P_tF_tP_t + \lambda I$ we can express $\|\theta\|_{\tilde{V}_t}$ as

$$\|\theta\|_{\tilde{V}_t} = [\bar{K}_{\mathcal{Z}_t}\alpha + \alpha_{m_t+1}K_{s_t}]^\top[\bar{K}_{\mathcal{Z}_t}K^{-1}_{\mathcal{Z}_t,\mathcal{Z}_t}K_{\mathcal{Z}_t,\mathcal{S}_t}K_{\mathcal{S}_t,\mathcal{Z}_t}K^{-1}_{\mathcal{Z}_t,\mathcal{Z}_t}\bar{K}^\top_{\mathcal{Z}_t} + \lambda I][\bar{K}_{\mathcal{Z}_t}\alpha + \alpha_{m_t+1}K_s].$$

This can be reformulated as

$$\begin{bmatrix}\alpha & \alpha_{m_t+1}\end{bmatrix} Q_t \begin{bmatrix}\alpha \\ \alpha_{m_t+1}\end{bmatrix},$$

where $Q_t = \begin{bmatrix}A_t & b_t \\ b_t^\top & c_t\end{bmatrix}$ and for which we have $A_t = K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}$, $b_t^\top = K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{s\mathcal{Z}_t}$ and eventually $c_t = K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} + \lambda K_{ss}$

Next to find the variance term, we note $q_t = [K_{\mathcal{Z}_t s}, K_{ss}]^\top$ and reformulate the optimization process above as

$$\max_{\alpha\in\mathbb{R}^{m_t+1}} \alpha^\top q_t$$
$$\text{s.t } \alpha^\top Q_t \alpha \leq 1$$

gives the solution $\alpha' = \dfrac{Q_t^{-1/2}q_t}{\|Q_t^{-1/2}q_t\|}$ which gives $\sigma_{t,a}$ the maximum value: $\sqrt{q_t^\top Q_t^{-1}q_t}$. We will now express the squared maximum $\sigma_{t+1,a}^2 = q_t^\top Q_t^{-1}q_t$ using the Schur complement on the Q_t matrix.

Defining $A_t = K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}$ and the Schur complement $l_t = c_t - b_t^\top A_t^{-1}b_t$.

We start by simplifying the expression of the Schur complement. For this we reformulate

$$\begin{aligned}
A_t &= K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t} \\
b_t^\top &= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}(A_t - \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}) + \lambda K_{s\mathcal{Z}_t} \\
&= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_t \\
c_t &= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}(A_t - \lambda K_{\mathcal{Z}_t\mathcal{Z}_t})K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} + \lambda K_{ss} \\
&= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_tK_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} - \lambda K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} + \lambda K_{ss}.
\end{aligned}$$

Thus we obtain:

$$\begin{aligned}
l_t &= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_tK_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} - \lambda K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} + \lambda K_{ss} - K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_tA_t^{-1}A_tK_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} \\
&= \lambda(K_{ss} - K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s}).
\end{aligned}$$

Then we write the product between Q_t^{-1} and q_t as:

$$\begin{aligned}
\tilde{\sigma}_{t+1,a}^2 &= \begin{bmatrix}K_{s\mathcal{Z}_t} & K_{ss}\end{bmatrix}\begin{bmatrix}A_t^{-1} + \frac{1}{l}A_t^{-1}b_tb_t^\top A_t^{-1} & -\frac{1}{l}A_t^{-1}b_t \\ -\frac{1}{l}b_t^\top A_t^{-1} & \frac{1}{l}\end{bmatrix}\begin{bmatrix}K_{\mathcal{Z}_t s} \\ K_{ss}\end{bmatrix} \\
&= \begin{bmatrix}K_{s\mathcal{Z}_t}A_t^{-1} + \frac{1}{l}K_{s\mathcal{Z}_t}A_t^{-1}b_tb_t^\top A_t^{-1} - \frac{1}{l}K_{ss}b_t^\top A_t^{-1} & -\frac{1}{l}\lambda K_{s\mathcal{Z}_t}A_t^{-1}b_t + \frac{1}{l}K_{ss}\end{bmatrix}\begin{bmatrix}K_{\mathcal{Z}_t s} \\ K_{ss}\end{bmatrix} \\
&= K_{s\mathcal{Z}_t}A_t^{-1}K_{\mathcal{Z}_t s} + \frac{1}{l}K_{s\mathcal{Z}_t}A_t^{-1}b_tb_t^\top A_t^{-1}K_{\mathcal{Z}_t s} - \frac{1}{l}K_{ss}b_t^\top A_t^{-1}K_{\mathcal{Z}_t s} - \frac{1}{l}K_{s\mathcal{Z}_t}A_t^{-1}b_tK_{ss} + \frac{1}{l}K_{ss}^2 \\
&= K_{s\mathcal{Z}_t}A_t^{-1}K_{\mathcal{Z}_t s} + \frac{1}{l}\left(K_{s\mathcal{Z}_t}A_t^{-1}b_t - K_{ss}\right)^2 \\
&= K_{s\mathcal{Z}_t}A_t^{-1}K_{\mathcal{Z}_t s} + \frac{1}{l}\left(K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} - K_{ss}\right)^2 \\
&= K_{s\mathcal{Z}_t}A_t^{-1}K_{\mathcal{Z}_t s} + \frac{1}{\lambda}K_{ss} - \frac{1}{\lambda}K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t s} \\
&= \frac{1}{\lambda}k(s_{t,a}, s_{t,a}) + \Delta_{t+1,a}\,,
\end{aligned}$$

where $\Delta_{t+1,a} := K_{\mathcal{Z}_t}(s_{t+1,a})^\top\left(\Lambda_{t+1} - \frac{1}{\lambda}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}\right)K_{\mathcal{Z}_t}(s_{t+1,a})$ and $\Lambda_{t+1} := A_{t+1}^{-1}$.

This proves the first of Prop. 4.2.

Algorithm 4: Efficient Kernel UCB

Input: T the horizon, λ regularization and exploration parameters, k the kernel function, $\varepsilon > 0, \gamma > 0$

Initialization;

Context x_0, a_0 chosen randomly and reward r_0 ;

$\mathcal{S} = \{(x_0, a_0)\}, Y_{\mathcal{S}} = [r_0]$;

$\mathcal{Z} = \{(x_0, a_0)\}$;

$\Lambda_t = (K_{\mathcal{Z}\mathcal{S}} K_{\mathcal{S}\mathcal{Z}} + \lambda K_{\mathcal{Z}\mathcal{Z}})^{-1}$ $\Gamma_t = K_{\mathcal{Z}\mathcal{S}} Y_{\mathcal{S}}$;

for $t = 1$ *to* T **do**

  Observe context x_t ;

  Choose $\tilde{\beta}_t$ (e.g as in Lem. 4.1, and $\delta = \frac{1}{T^2}$) ;

  Choose $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\mu}_{t,a} + \tilde{\beta}_t \tilde{\sigma}_{t,a}$;

    $\tilde{\mu}_{t,a} \leftarrow K_{\mathcal{Z}}(s_{t,a})^\top \Lambda_t \Gamma_t$;

    $\Delta_{t,a} = K_{\mathcal{Z}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K_{\mathcal{Z}\mathcal{Z}}^{-1}\right) K_{\mathcal{Z}}(s_{t,a})$;

    $\tilde{\sigma}^2_{t,a} \leftarrow \frac{1}{\lambda} k(s_{t,a}, s_{t,a}) + \Delta_{t,a}$;

  Observe reward r_t and $s_t \leftarrow (x_t, a_t)$;

  $Y_{\mathcal{S}} \leftarrow [Y_{\mathcal{S}}, r_t]^\top, \mathcal{S} \leftarrow \mathcal{S} \cup \{s_t\}$;

  $\mathcal{Z}' \leftarrow \mathrm{KORS}(t, \mathcal{Z}, K_{\mathcal{Z}}(s_t), \lambda, \varepsilon, \gamma)$;

  if $\mathcal{Z}' = \mathcal{Z}$ **then**

    Incremental inverse update Λ_t with s_t;

    $\Gamma_{t+1} \leftarrow \Gamma_t + r_t K_{\mathcal{Z}}(s_t)$;

  end

  else

    $z = \mathcal{Z}' \setminus \mathcal{Z}$;

    Incremental inverse update Λ_t with s_t, z ;

    Incremental inverse update $K_{\mathcal{Z}\mathcal{Z}}^{-1}$ with z;

    Update $\Gamma_{t+1} \leftarrow [\Gamma_t + r_t K_{\mathcal{Z}}(s_t),\ K_{\mathcal{S}}(z)^\top Y_{\mathcal{S}}]^\top$

  end

end

Discussion on practical implementation and time and space complexities The efficient implementation of the algorithm requires to perform efficient updates of the quantities (defined in Prop 4.2) $\Lambda_t = (K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} K_{\mathcal{S}_{t-1}\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}})^{-1}$ and $\Gamma_t = K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1}$.

(i) When the dictionary is not updated $\mathcal{Z}_t = \mathcal{Z}_{t-1}$. For the matrix Γ_t we can perform the update $\Gamma_{t+1} \leftarrow \Gamma_t + r_t K_{\mathcal{Z}_t}(s_t)$ which requires m_t kernel evaluations. As for the matrix Λ_t we can use the first rank Shermann-Morrison formula on it by adding updates on s_t in $\mathcal{O}(m_t^2)$ operations where $\Lambda_{t+1} = (K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t})^{-1}$. Here we only store $K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}$ and do not update it.

(ii) When the dictionary is updated $\mathcal{Z}_t \neq \mathcal{Z}_{t-1}$ and we can write $\mathcal{Z}_t = \mathcal{Z}_{t-1} \cup \{z_{m_t}\}$,

Regarding Γ_t, we do two updates, one on the state s_t by adding $r_t K_{\mathcal{Z}_{t-1}}(s_t)$ and a second on the new anchor point z_{m_t} so that we have

$$\Gamma_{t+1} \leftarrow [\Gamma_t + r_t K_{\mathcal{Z}_{t-1}}(s_t), K_{\mathcal{S}_t}(z_{m_t})^\top Y_t]^\top .$$

The first update is performed in $\mathcal{O}(m_t)$ kernel evaluations as in the (i) case, and the second update requires $\mathcal{O}(t)$ kernel evaluations and then $\mathcal{O}(t)$ computations. Note that the (ii) is only visited at most m times which is the size of the dictionary at $t = T$.

Regarding Λ_t, we note that we can write $K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}$ as

$$\begin{bmatrix} K_{\mathcal{Z}_{t-1}\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} & K_{\mathcal{Z}_{t-1}\mathcal{S}_t} K_{\mathcal{S}_t}(z) + \lambda K_{\mathcal{Z}_{t-1}}(z) \\ K_{\mathcal{S}_t}(z)^\top K_{\mathcal{S}_t\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}}(z)^\top & K_{\mathcal{S}_t}(z)^\top K_{\mathcal{S}_t}(z) + \lambda k(z,z) \end{bmatrix}.$$

We perform the update in two stages by first computing the inverse $(K_{\mathcal{Z}_{t-1}\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}})^{-1}$ by using a first-rank Sherman Morrison on the state update s_t, as if the dictionary did not change, and we then perform a Schur complement update using the latter inverse. Both updates are done in $\mathcal{O}(m_t^2)$ operations.

As for the inverse of the projection gram matrix, we use a Schur complement update in $\mathcal{O}(m_t^2)$ operations that we detail here for $K^{-1}_{\mathcal{Z}_{t+1}\mathcal{Z}_{t+1}}$:

$$K^{-1}_{\mathcal{Z}_t\mathcal{Z}_t} = \begin{bmatrix} K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} + \frac{1}{\omega} w_t w_t^\top & -\frac{1}{\omega} w_t \\ -\frac{1}{\omega} w_t^\top & \frac{1}{\omega} \end{bmatrix}$$

where $\omega = k(z_{m_t}, z_{m_t}) - K_{\mathcal{Z}_{t-1}}(z_{m_t})^\top K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} k_{\mathcal{Z}_{t-1}}(z_{m_t})$ and with $w_t = K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} k_{\mathcal{Z}_{t-1}}(z_{m_t})$.

E.3 Kernel Online Row Sampling (KORS) Subroutine

As in Calandriello et al. (2017b), let us define a projection dictionary $\mathcal{Z}_t$ as a collection of indexed anchor points $\{(z_{t_i}\}_{1 \le i \le m_t}$ where $m_t = |\mathcal{Z}_t|$ as well as the rescaling diagonal matrix $S_{\mathcal{Z}_t}$ with $1/\sqrt{\tilde{p}_{z_s}}$ corresponding to the past sampling probabilities of points $z \in \mathcal{Z}_t$, this matrix is of size $m_t \times m_t$. At each time step, KORS temporarily adds t with weight 1 to the temporary dictionary $\mathcal{Z}_t^*$ and accordingly augments the corresponding matrix $S_{\mathcal{Z}_t^*}$. The augmented dictionary is then used to compute the ridge leverage score (RLS) estimator:

$$\tilde{\tau}_t = \frac{1+\varepsilon}{\mu} \left(k(s_t, s_t) - K_{\mathcal{Z}_t^*}(s_t)^\top S_{\mathcal{Z}_t^*} (S^\top_{\mathcal{Z}_t^*} K_{\mathcal{Z}_t^* \mathcal{Z}_t^*} S_{\mathcal{Z}_t^*} + \mu I)^{-1} S^\top_{\mathcal{Z}_t^*} K_{\mathcal{Z}_t^*}(s_t) \right) . \tag{41}$$

Afterward, it draws a Bernoulli random variable z_t proportionally to $\tilde{\tau}_t$, if it succeeds, $(z_t = 1)$ the point is deemed relevant and added to the dictionary, otherwise it is discarded and never added.

Algorithm 5: Incremental Kernel Online Row Sampling (KORS) subroutine

Input: Time t, past dictionary $\mathcal{Z}$, context-action s_t, regularization μ, accuracy ε, budget γ
Compute the leverage score $\tilde{\tau}_t$ from $\mathcal{Z}, s_t, \mu, \varepsilon$;
Compute $\tilde{p}_t = \min\{\gamma \tilde{\tau}_t, 1\}$;
Draw $z_t \sim \mathcal{B}(\tilde{p}_t)$ and if $z_t = 1$, add s_t to $\mathcal{Z}$;
Result: Dictionary $\mathcal{Z}$

Here, all rows and columns for which $S_{t,*}$ is zero (all points outside the temporary dictionary $\mathcal{I}_{t,*}$) do not influence the estimator, so they can be excluded from the computation. As a consequence, the RLS score $\tilde{\tau}_t$ can be computed efficiently in $\mathcal{O}((m_t+1)^2)$ space and $\mathcal{O}((m_t+1)^2)$ time, using an incremental update in Eq. (41).

As a side note, the quantity $\tilde{\tau}$ is an estimator of the exact RLS quantity τ_t (see Calandriello et al. (2017b)):

$$\tau_t = \varphi_t^\top (K_t + \mu I)^{-1} \varphi_t \, . \tag{42}$$

Here, leverage scores are used to measure the correlation between the new point φ_t w.r.t. the previous $t-1$ points $\{\varphi_i\}_{i \le t-1}$, and therefore how essential it is in characterizing the dataset. In particular, if φ_t is completely orthogonal to the other points, its RLS is maximized, while in the opposite case it would be minimal. In the incremental strategy of the Nyström dictionary building, we use the RLS estimates to add anchor points that are as informative as possible.

F Experiment details

In this section we provide further details as well as additional discussions and numerical results on our proposed method.

F.1 Reproducibility and Implementations

We provide code that is accessible at the link `https://github.com/criteo-research/Efficient-Kernel-UCB`. All experiments were run on a single CPU core (2 x Intel(R) Xeon(R) Gold 6146 CPU@ 3.20GHz).

Baseline implementations We implemented the BKB and BBKB algorithms in (Calandriello et al., 2019) and (Calandriello et al., 2020) by introducing modifications in their implementation to handle contextual information.

For both methods, in the contextual variant, each update involves the computation of a new covariance matrix $K_{\mathcal{Z}\mathcal{S}}K_{\mathcal{S}\mathcal{Z}}$ while the original algorithms do not involve contexts and consider a finite set of actions, allowing to compute the covariance matrix on the finite set of actions (which is done for computational efficiency and is impossible in the joint context-action space). The baselines were carefully optimized using the Jax library (`https://github.com/google/jax`) to allow for just in time compilations of similar blocks in every methods.

Empirical setting In our empirical setting we aimed at showing the regret/computational complexity compromise that is achieved by each method. In particular, both the CBBKB method (Calandriello et al., 2020) and our EK-UCB algorithm use additional hyperparameters than the CBKB. As a matter of fact, CBBKB uses an accumulation threshold C and is used for the 'resparsification' step, with dictionary updates based on all historical states. EK-UCB also uses the hyperparameter μ in KORS that is set to λ for optimal regret-time compromise (see Theorem 4.1). The KORS algorithm uses a budget parameter γ, for which we found empirically good performances when $\gamma \approx \lambda$. We tried our method with a grid on hyperparameters and discuss their influence in the next subsection.

F.2 Additional Results

In this section we provide additional numerical experiment discussions.

F.2.1 Additional discussions on the setting of Section 5

We present additional results on the synthetic setting presented in Section 5 that we call 'Bump' in Figures 3, 4, 5. Here we fix $\lambda = \mu$ for EK-UCB and report the performances of the baselines with the same hyperparameters and make the accumulation threshold C of CBBKB vary through the Figures 3, 4, 5. We provide more discussion on the methods we evaluated.

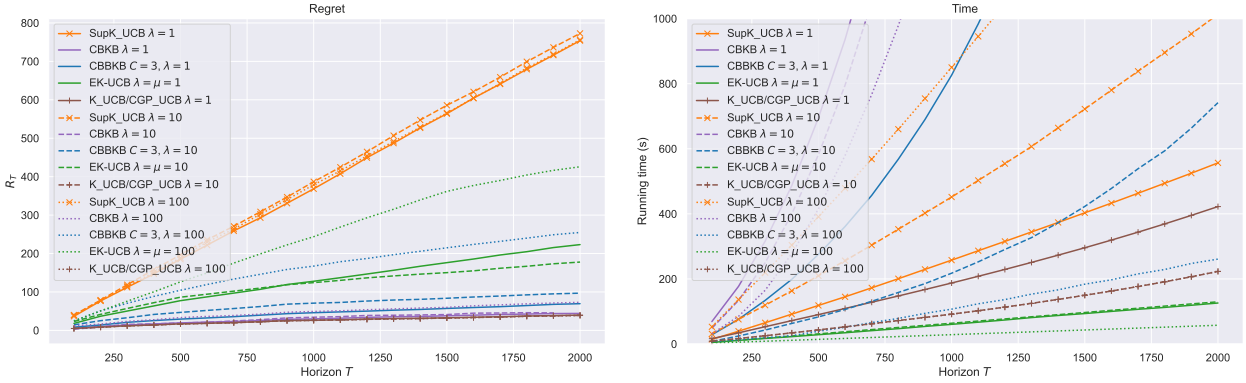

Figure 3: 'Bump' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ (see Corollary 3.1 and 4.1) with varying λ and $C = 3$ for CBBKB. EK-UCB matches the best regret-time compromise.

More dictionary updates lead to better regret but a higher computational complexity We note that the CBKB baseline achieves satisfactory regret but with a drastically higher computational time. This is due to the fact that it resamples the dictionary at each step and therefore resamples a dictionary at the price of a higher time complexity. As for CBBKB, throughout the Figures 3, 4, 5, we can see that the accumulation threshold C that controls the anchor point update frequency determines the regret-time compromise. The lower C, the better is the regret but the higher is the computational time. We can see through the figures that for all values of C, our EK-UCB method achieves similar or better (especially when $C = 30$) regret than CBBKB while always being both faster than CBBKB but more importantly faster than K-UCB. Overall, EK-UCB proposes the most satisfactory regret-time compromise. Moreover, we see that the SupK-UCB method also performs poorly even with different parameters λ and that the optimized K-UCB method also performs better than efficient strategies when the computational overheads of dictionary buildings overtake the efficient kernel approximations.

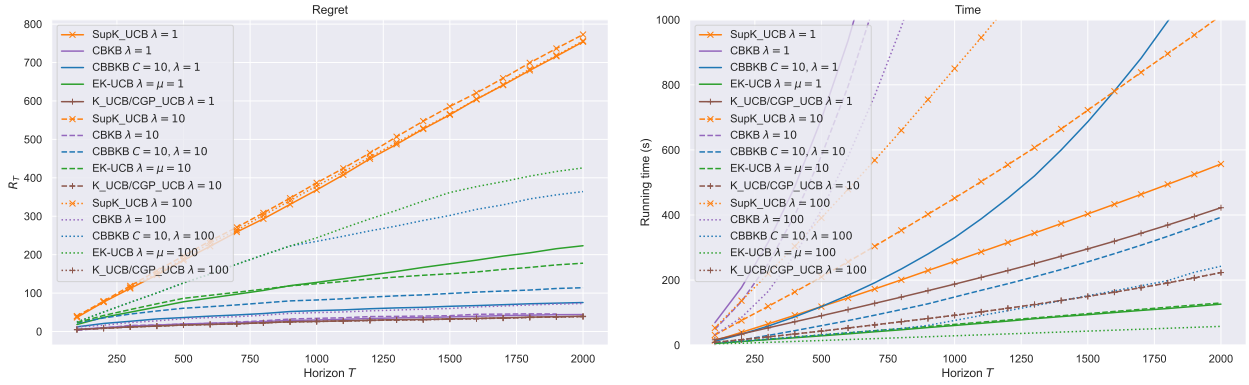

Figure 4: 'Bump' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 10$ for CBBKB. EK-UCB matches the best regret-time compromise.

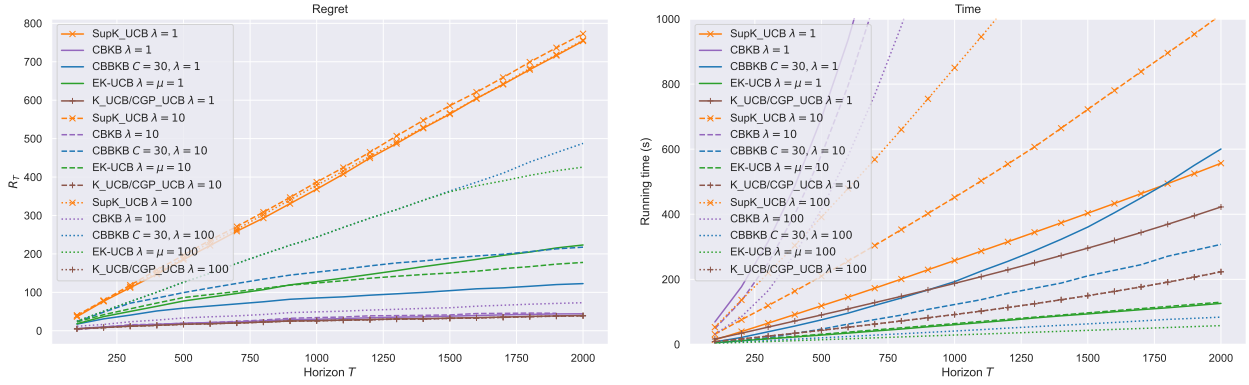

Figure 5: 'Bump' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 30$ for CBBKB. EK-UCB matches the best regret-time compromise

The regularization parameter controls the regret-time comprise in EK-UCB In our method, we can see that the higher λ (with $\lambda = \mu$) the faster the algorithm is but the worse is its regret. As discussed in Corollary 3.1 and 4.1, we use the heuristic to take $\lambda \approx \sqrt{T}$ and set $\mu = \lambda$ afterwards to enjoy the optimal guarantees of our algorithm.

F.2.2 Additional synthetic settings

In this section we introduce additional settings that we call the 'Chessboard' setting as well as the 'Step Diagonal' setting. The two settings lead to similar numerical conclusions as the previous one. We provide more discussions here.

Chessboard and Step Diagonal synthetic setups. The 'Chessboard' synthetic setup is a contextual environment with a piecewise reward function over the joint context-action space $\mathcal{X} \times \mathcal{A} = [0, 1] \times [0, 1]$. More precisely, the joint 2D space is cut into a grid where the values are either 1, 0.5 or 0 according to the part of the grid. Results are shown in Figures 7, 8, 9. The 'Step diagonal' synthetic setup is a contextual environment with a diagonal reward function over the joint context-action space $\mathcal{X} \times \mathcal{A} = [0, 1] \times [0, 1]$. More precisely, the joint 2D space has values of 0 everywhere except along two bands along the diagonal where the action and context values are identical with values 0.5 and 1 respectively on the sub diagonal and the above diagonal. Results are shown in Figures 10, 11, 12. See the code for more details and an illustration of the settings in Fig 6.

Regret-time compromise for CBBKB and EK-UCB. The two settings show what both algorithms CBBKB and EK-UCB achieve as a regret-time compromise. In cases where C is lower (note that CBKB

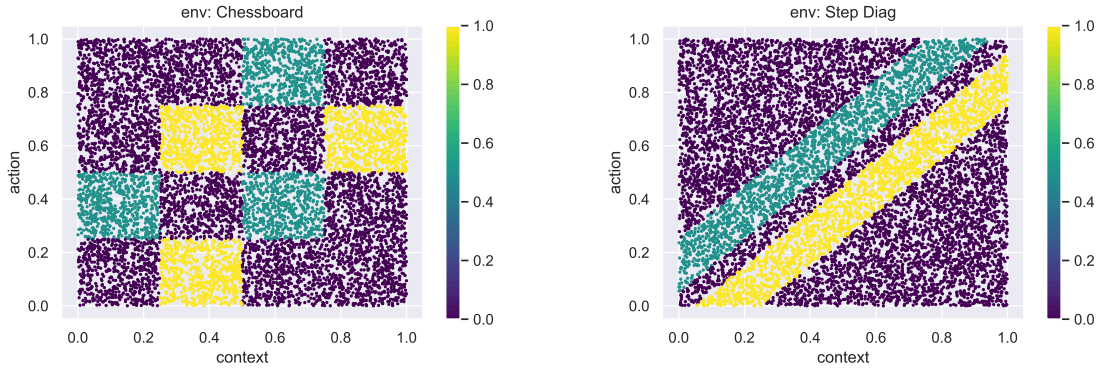

Figure 6: Chessboard (left) and Step Diagonal (right) synthetic setups.

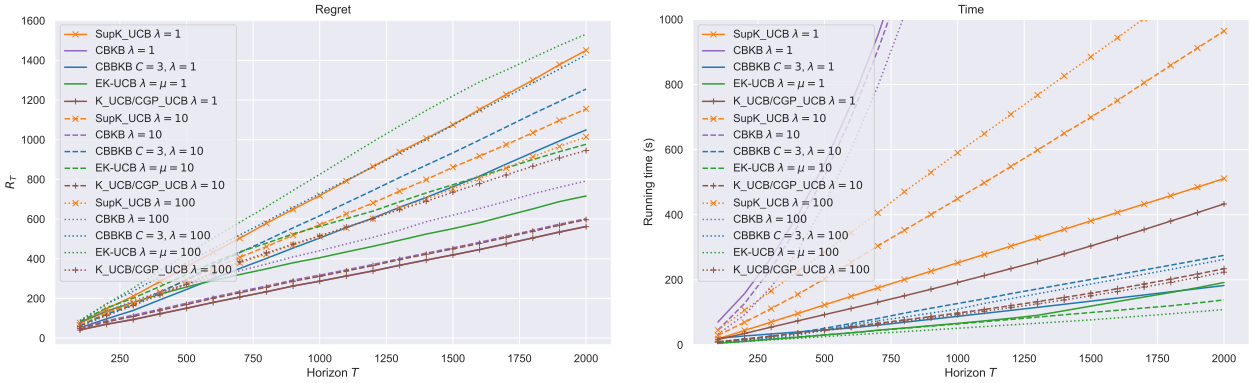

Figure 7: 'Chessboard' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 3$ for CBBKB.

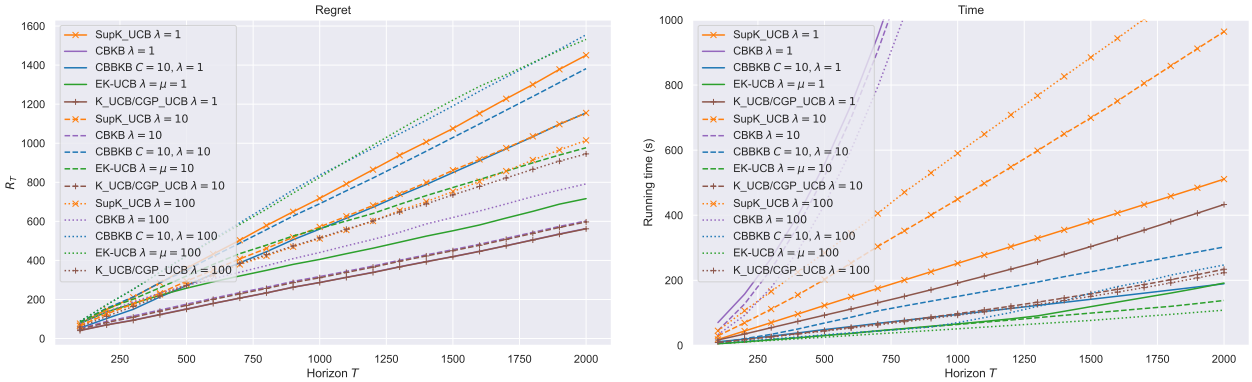

Figure 8: 'Chessboard' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 10$ for CBBKB.

corresponds to CBBKB with $C = 1$) the regret often decreases at the price of higher computational time complexity. Similarly, we can notice that our method has better regrets when λ is low, but with higher computational times, while still providing a benefit over to the K-UCB method, unlike CBBKB. We therefore note again that in practice, dictionary building computational overheads may influence the global computational complexity. Overall, our method with its incremental dictionary building strategy achieves the best satisfactory time-regret compromises in the Chessboard and Step Diagonal settings compared to both K-UCB and the efficient

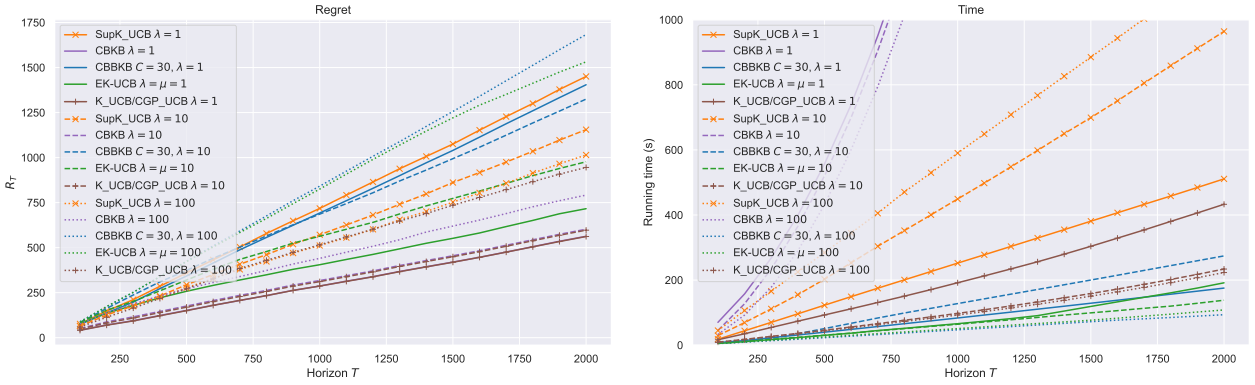

Figure 9: 'Chessboard' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 30$ for CBBKB.

algorithms CBKB and CBBKB.

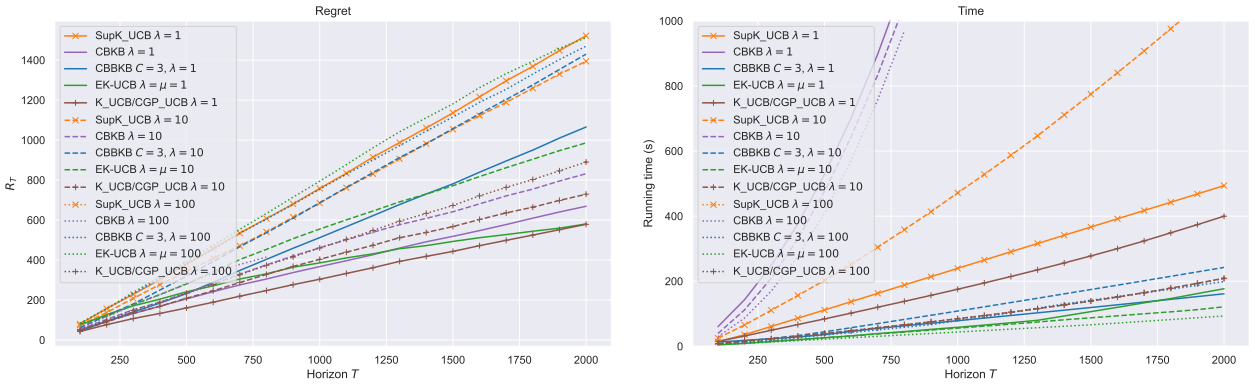

Figure 10: 'Step Diagonal' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 3$ for CBBKB.

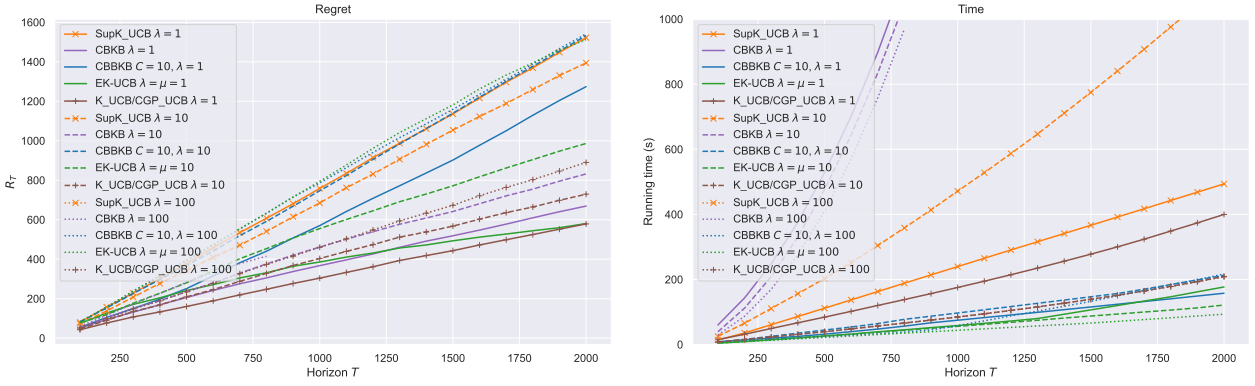

Figure 11: 'Step Diagonal' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 10$ for CBBKB.

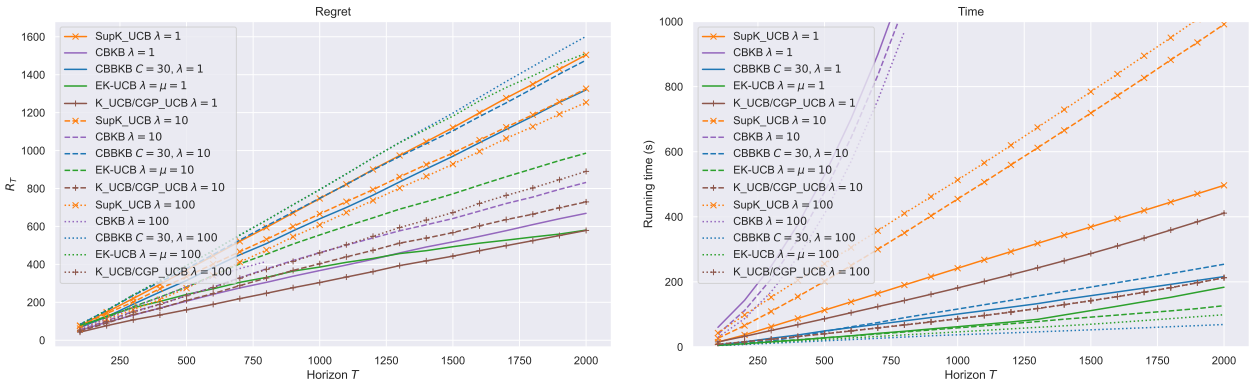

Figure 12: 'Step Diagonal' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 30$ for CBBKB.