

Projection by Convolution: Optimal Sample Complexity for Reinforcement Learning in Continuous-Space MDPs

Davide Maran

DAVIDE.MARAN@POLIMI.IT

Alberto Maria Metelli

ALBERTOMARIA.METELLI@POLIMI.IT

Matteo Papini

MATTEO.PAPINI@POLIMI.IT

Marcello Restelli

MARCELLO.RESTELLI@POLIMI.IT

Politecnico di Milano, Piazza Leonardo da Vinci, 32-36 - Città Studi, Milano (MI)

Editors: Shipra Agrawal and Aaron Roth

Abstract

We consider the problem of learning an ε -optimal policy in a general class of continuous-space Markov decision processes (MDPs) having smooth Bellman operators. Given access to a generative model, we achieve rate-optimal sample complexity by performing a simple, *perturbed* version of least-squares value iteration with orthogonal trigonometric polynomials as features. Key to our solution is a novel projection technique based on ideas from harmonic analysis. Our $\tilde{\mathcal{O}}(\varepsilon^{-2-d/(\nu+1)})$ sample complexity, where d is the dimension of the state-action space and ν the order of smoothness, recovers the state-of-the-art result of discretization approaches for the special case of Lipschitz MDPs ($\nu = 0$). At the same time, for $\nu \rightarrow \infty$, it recovers and greatly generalizes the $\mathcal{O}(\varepsilon^{-2})$ rate of low-rank MDPs, which are more amenable to regression approaches. In this sense, our result bridges the gap between two popular but conflicting perspectives on continuous-space MDPs.

Keywords: Reinforcement learning; Harmonic analysis; Sample complexity; Continuous spaces

1. Introduction

Reinforcement learning (RL, [Sutton and Barto, 2018](#)) is a paradigm of artificial intelligence where an agent interacts with an environment, receiving a reward and observing transitions in the state of the environment. The agent’s objective is to learn a policy, which is a mapping from the states to the probability distributions on the set of their action, that maximizes the expected return, i.e., the expected long-term cumulative reward. As the environment is stochastic, a way of assessing the performance of an algorithm, having fixed an accuracy threshold ε and error probability δ , is to find the minimum number of interactions n with the environment needed to learn a policy that is ε -optimal with probability at least $1 - \delta$. This kind of result falls under the name of *sample complexity bounds*. For the case of *tabular Markov decision processes* (MDPs, [Puterman, 2014](#)), an optimal result was first proved by [Azar et al. \(2013\)](#), showing a bound on the sample complexity with access to a generative model of order $\tilde{\mathcal{O}}(H^2|\mathcal{S}||\mathcal{A}|\varepsilon^{-2})$ with high probability, where \mathcal{S} is a finite state space, \mathcal{A} is a finite action space and H the time horizon of every episode. This regret is minimax-optimal, in the sense that no algorithm can achieve smaller regret for every arbitrary tabular MDP, as proved by [Azar et al. \(2013\)](#). Assuming that the MDP is tabular, with a finite

number of states and actions, is extremely restrictive. Indeed, most appealing applications of RL like trading (Hambly et al., 2023), robotics (Kober et al., 2013), and autonomous driving (Kiran et al., 2021) do not fall in this setting. For this reason, the search for algorithms with sample complexity bounds for RL in *continuous* spaces is currently one of the most important challenges of the whole field. Obviously, it is not possible to learn an ε -optimal policy for any continuous space MDP¹ and assumptions should be made on the structure of the process.

Previous works have introduced a variety of different settings when the continuous-space RL problem is learnable. The *linear quadratic regulator* (LQR, Bemporad et al., 2002), a model coming from control theory, assumes that the state of the system evolves according to a linear dynamics and that the reward is quadratic. For this problem, when the system matrix is unknown, Abbasi-Yadkori and Szepesvári (2011) obtained a sample complexity bound of order $\tilde{O}(\varepsilon^{-2})$ for a computationally inefficient algorithm. This latter limitation was then removed by Dean et al. (2018); Cohen et al. (2019). *Linear MDPs* (Yang and Wang, 2019; Jin et al., 2020) are another very popular setting where a different form of linearity is assumed. Here, the transition function of the MDP can be factorized as a scalar product between a feature map $\varphi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$ and an unknown vector of signed measures over \mathcal{S} . The reward function is typically assumed to be linear in the same features. For this setting, assuming that the feature map is known, there are algorithms (Jin et al., 2020) achieving sample complexity of order $\tilde{O}(\varepsilon^{-2})$. When the feature map is unknown (Agarwal et al., 2020), the problem gets much more challenging. State-of-the-art results in this field (Uehara et al., 2021) are able to achieve sample complexity $\tilde{O}(|\mathcal{A}|^5 \log(|\mathcal{F}|)\varepsilon^{-2})$, where \mathcal{F} is a known function class containing the true feature map, only if the reward function is known. Despite the optimal dependence on ε , this bound comes with the additional intake that the action space must be finite and “small”, so that this result does not apply to continuous-space RL.

All these settings, by assuming a parametric model, are relatively limited and clearly not able to provide a perspective over general MDPs with continuous spaces. A more general model, building on the often realistic assumption that close state-action pairs are associated with similar reward and transition functions, is that of *Lipschitz MDPs* (Rachelson and Lagoudakis, 2010). Lipschitz MDPs have been applied to several different settings, such as policy gradient methods (Pirota et al., 2015), model-based RL (Asadi et al., 2018), RL with delayed feedback (Liotet et al., 2022), and auxiliary tasks such as imitation learning (Maran et al., 2023). The price for generality is paid with a very inconvenient sample complexity guarantee. State-of-the-art works (e.g., Song and Sun, 2019; Sinclair et al., 2019; Le Lan et al., 2021) in learning theory which focused on this setting where only able to achieve sample complexity of order $\tilde{O}(\varepsilon^{-d-2})$, where d is the dimension of the state-action space, that is assumed to be $\mathcal{S} \times \mathcal{A} = [-1, 1]^d$. Another non-parametric family of continuous spaces MDPs is that of *Kernelized MDPs* (Yang et al., 2020), where both the reward function and the transition function belong to a *reproducing kernel Hilbert space* (RKHS) induced by a known kernel. In the typical application to continuous-state MDPs, the kernel is assumed to come from the Matérn covariance function with parameter $m > 0$. Optimal sample complexity bounds for this problem were very recently proved by Vakili and Olkhovskaya (2023), which showed a sample complexity of order $\tilde{O}(\varepsilon^{-\frac{d+2m}{m}})$. In this result, the presence of d at the exponent is mitigated by m . Indeed, for $m \gg d$, the bounds approach the desired value of $\tilde{O}(\varepsilon^{-2})$ that is achievable for LQRs and Linear MDPs. Still, the Kernelized MDP model is significantly more restrictive than that of

1. Think, for instance, of the case where the reward function is discontinuous.

Lipschitz MDPs. For example, any deterministic MDP is not included in the kernelized family for the Matérn kernel.

Our Contribution. As we saw, the results in the literature of continuous-space RL are affected by a huge gap. Good sample complexity bounds of order $\tilde{\mathcal{O}}(\varepsilon^{-2})$ are known for very *specific* classes of problems, and very unsatisfactory bounds of order $\tilde{\mathcal{O}}(\varepsilon^{-d-2})$ are known for the very *general* Lipschitz MDP case. In this paper, we will focus on a very large class of MDPs known as weakly ν -smooth MDPs, introduced in (Maran et al., 2024), depending on a smoothness parameter $\nu \in \mathbb{N}$. For $\nu = 0$, this class is a generalization of the Lipschitz MDP, while for any ν (including $\nu \rightarrow +\infty$) this class generalizes the most common parametric problems, including LQRs and Linear MDPs. For any ν , this class also generalizes the assumption of kernelized MDP when $m > \nu + 1$. We then define an algorithm based on a novel regression technique, which draws on both kernel theory and Fourier series, able to achieve a sample complexity bound of order $\tilde{\mathcal{O}}(\varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}})$, which is tight in all cases, matching one lower bound proved in the much simpler setting of *stochastic optimization*. In this way, the algorithm is able to achieve the best order in ε both for processes with little smoothness ($\approx \varepsilon^{-d-2}$) and with higher smoothness ($\approx \varepsilon^{-2}$).

Remark 1 *Note that, in the above literature review, some works that we have mentioned do not explicitly show sample complexity bounds, but rather regret bounds. Without giving other details, it is sufficient to say that any bound of the regret of the form $R_T \leq \tilde{\mathcal{O}}(CT^\beta)$ leads to a sample complexity guarantee of $n \leq \tilde{\mathcal{O}}(C/\varepsilon)^{\frac{1}{1-\beta}}$, both holding in high probability (one just runs a regret minimization algorithm and outputs a policy drawn uniformly from the sequence of policies). Using this relation, we have rephrased all the results in the literature in terms of sample complexity.*

2. Preliminaries

We consider a finite-horizon Markov decision process (MDP, Puterman, 2014) $M = (\mathcal{S}, \mathcal{A}, p, R, H)$, where $\mathcal{S} = [-1, 1]^{d_S}$ is the state space, and $\mathcal{A} = [-1, 1]^{d_A}$ is the action space (this choice is without loss of generality, as any compact set could be used instead); $p = \{p_h\}_{h=1}^H$ is the sequence of transition functions, each mapping a pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ to a probability distribution $p_h(\cdot | s, a)$ over \mathcal{S} ; $R = \{R_h\}_{h=1}^H$ is the sequence of reward functions, each mapping a pair (s, a) to a real number $R_h(s, a) \in [0, 1]$, and H is the time horizon. A policy $\pi = \{\pi_h\}_{h=1}^H$ is a sequence of mappings from \mathcal{S} to the probability distributions over \mathcal{A} . For each stage $h \in [H]$, the action is chosen according to $a_h \sim \pi_h(\cdot | s_h)$, the agent gains a deterministic² reward $R_h(s_h, a_h)$, and the environment transitions to the next state $s_{h+1} \sim p(\cdot | s_h, a_h)$. In this setting, it is useful to define the following quantities.

Value functions and Bellman operator. The state-action value function (or *Q-function*) quantifies the expected sum of the rewards obtained under a policy π , starting from a state-stage pair $(s, h) \in \mathcal{S} \times [H]$ and fixing the first action to some $a \in \mathcal{A}$:

$$Q_h^\pi(s, a) := \mathbb{E}_\pi \left[\sum_{\ell=h}^H R_\ell(s_\ell, a_\ell) \mid s_0 = s, a_0 = a \right], \quad (1)$$

2. This is just assumed for simplicity. Results are still valid if we only have access to noisy reward samples, as long as the noise is subgaussian since the stochasticity in the transition function overcomes the one in the reward.

where \mathbb{E}_π denotes expectation w.r.t. to the stochastic process $a_h \sim \pi_h(\cdot|s_h)$ and $s_{h+1} \sim p_h(\cdot|s_h, a_h)$ for all $h \in [H]$. The state value function (or *V-function*) is defined as $V_h^\pi(s) := \mathbb{E}_{a \sim \pi_h(\cdot|s)}[Q_h^\pi(s, a)]$, for all $s \in \mathcal{S}$. It can be proved that, under mild smoothness assumptions (Bertsekas and Shreve, 1996), there is a family of *optimal* policies such that their Q_h^π functions at any stage dominates the one of every other policy for every state-action couple. Their corresponding state-action value function must satisfy the following condition:

$$\forall h \in [H + 1] \quad Q_h^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}, \quad Q_h^* = \begin{cases} 0 & h = H + 1 \\ \mathcal{T}_h Q_{h+1}^* & 1 \leq h \leq H \end{cases},$$

where \mathcal{T} is the *Bellman optimality operator*, defined in this way:

$$\mathcal{T}_h f(s, a) := R_h(s, a) + \mathbb{E}_{s' \sim p_h(\cdot|s, a)} \left[\sup_{a' \in \mathcal{A}} f(s', a') \right].$$

As the operator \mathcal{T} chooses, at each time step, the action maximizing long-term expected reward, it is easy to prove that $Q_h^*(s, a) = \sup_\pi Q_h^\pi(s, a)$ for every s, a, h .

Agent's goal. In this paper, we assume to have access to a *generative model* that, given a state-action-stage tuple (s, a, h) , returns a sample for the next state and one for the reward, $s' \sim p_h(\cdot|s, a)$, $r_h = R_h(s, a)$. The agent's goal is to output an estimate of the optimal policy with the least possible number of queries to the generator. Precisely, one seeks to find a probably approximately correct (PAC) policy $\hat{\pi}$, in the sense that the policy must satisfy:

$$\|V_1(\cdot) - V_1^{\hat{\pi}}(\cdot)\|_{L^\infty} := \sup_{s \in \mathcal{S}} |V_1(s) - V_1^{\hat{\pi}}(s)| \leq \varepsilon,$$

with probability at least $1 - \delta$. This shall be done with the least possible number of interactions with the environment (i.e., calls to the generative model), denoted with n .

Smoothness of real functions. Achieving a PAC guarantee for arbitrary MDPs with continuous spaces is clearly impossible. Indeed, even in the case $H = 1$ and $R_1(s, a) = R_1(a)$ for all $s \in \mathcal{S}$ (a continuous-armed bandit), if $R_1(a) = \mathbf{1}\{a = a_0\}$ (a discontinuous function), no algorithm can identify a_0 , hence the optimal policy. For this reason, assumptions must be enforced to guarantee that the reward and transition functions are endowed with some regularity property. Let $\Omega \subset [-1, 1]^d$ and $f : \Omega \rightarrow \mathbb{R}$. We say that $f \in \mathcal{C}^{\nu, 1}(\Omega)$ if it is ν -times continuously differentiable, and

$$\|f\|_{\mathcal{C}^{\nu, 1}} := \max_{|\alpha| \leq \nu + 1} \|D^\alpha f\|_{L^\infty} < +\infty,$$

where the multi-index derivative is defined as follows

$$D^\alpha f := \frac{\partial^{\alpha_1 + \dots + \alpha_d}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}.$$

The most straightforward case of this definition is given by $\mathcal{C}^{0, 1}(\Omega)$, corresponding to the space of Lipschitz continuous functions, whose increment is bounded by a constant.

Before digging into the application of smooth functions to MDPs, since we are going to draw from the theory of the Fourier series, we also have to introduce the periodic version of the spaces

defined before. For $\Omega = [-1, 1]^d$, we define the space $\mathcal{C}_p^{\nu,1}(\Omega)$ to be the set of functions that are $\mathcal{C}^{\nu,1}(\Omega)$ with periodic conditions at the boundaries (see for example [Wu et al. \(2014\)](#)). For this space, we take same the norm $\|\cdot\|_{\mathcal{C}^{\nu,1}(\Omega)}$ defined above. This assumption is necessary to use methods based on the Fourier series, as is often done in physics and engineering. Dealing with this kind of function, we can define the convolution even if the domain Ω is bounded. In the rest of this paper, we will call, for periodic functions f, g on Ω ,

$$f * g(x) := \int_{\Omega} f(y)g(x-y) dy,$$

where g is extended by periodicity (namely, if any component i of $x-y$ is more than 1, it becomes $x_i - y_i - 2$, and the opposite for $x_i - y_i < -1$) when $x-y$ falls outside of Ω . This operation is usually called *circular convolution*.

Smoothness in MDPs. In previous works on MDPs with continuous spaces, smoothness was assumed on the transition function $p_h(s'|s, a)$ as a function of these three variables. Here, following the same idea of [Maran et al. \(2024\)](#), we introduce a much more general assumption, which directly concerns the Bellman optimality operator.

Assumption 1 (ν -Smooth MDP). *An MDPs is smooth of order ν if, for every $h \in [H]$, the Bellman optimality operator \mathcal{T}_h is bounded on $\mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A}) \rightarrow \mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A})$ with a constant $C_{\mathcal{T}}$.*

Recall that, for an operator $\mathcal{B} : (\mathbb{V}, \|\cdot\|_{\mathbb{V}}) \rightarrow (\mathbb{W}, \|\cdot\|_{\mathbb{W}})$ between two normed vector spaces \mathbb{V}, \mathbb{W} , boundedness means that there is a constant $C_{\mathcal{B}} < +\infty$ such that $\|\mathcal{B}v\|_{\mathbb{W}} \leq C_{\mathcal{B}}(\|v\|_{\mathbb{V}} + 1)$. Therefore, the operator cannot transform a function that has a small norm into one that has one that is arbitrarily large. In our case, we are assuming that $\mathcal{T}_h f \in \mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A})$ with $\|\mathcal{T}_h f\|_{\mathcal{C}^{\nu,1}} \leq C_{\mathcal{T}}(\|f\|_{\mathcal{C}^{\nu,1}} + 1)$, for every function $f \in \mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A})$ and every $h \in [H]$. This assumption coincides with the one of Weakly Smooth MDP by [Maran et al. \(2024\)](#), apart from the fact that we are considering periodic functions (we have added the p subscript in $\mathcal{C}_p^{\nu,1}$ as a reminder). No assumption is strictly stronger than the other, and as the periodicity condition only concerns the boundaries of $\mathcal{S} \times \mathcal{A}$, the two models are essentially equivalent when the interesting part of the MDP lies in the interior of $\mathcal{S} \times \mathcal{A}$.

3. Trigonometric representation

The algorithm we present in this paper is based on the idea of approximating the state-action value function at any time-step h with a trigonometric polynomial. We are going to use the following notation.

Definition 1 *Define the function $\text{soc} : \mathbb{Z} \times [-1, 1] \rightarrow [-1, 1]$ as:*

$$\text{soc}(n, z) := \begin{cases} 1 & n = 0 \\ \sin(n\pi z) & n > 0 \\ \cos(n\pi z) & n < 0. \end{cases}$$

Overloading the notation, for $\mathbf{n} \in \mathbb{Z}^d$ and $z \in [-1, 1]^d$, define the monomial:

$$\text{soc}(\mathbf{n}, z) := \text{soc}(n_1, z_1)\text{soc}(n_2, z_2) \dots \text{soc}(n_d, z_d).$$

Define, for every $N > 0$ and $d > 0$, both integers, a degree N , d -variate trigonometric polynomial as a function $[-1, 1]^d \rightarrow \mathbb{R}$ of the form

$$f(z) = \sum_{\substack{\mathbf{n} \in \mathbb{Z}^d \\ \|\mathbf{n}\|_1 \leq N}} \theta_{\mathbf{n}} \text{soc}(\mathbf{n}, z),$$

for some real coefficients $\theta_{\mathbf{n}}$. We call \mathbb{T}_N^d the vector space of these functions (\mathbb{T}_N when d is clear from the context).

For $d = 1$, the space \mathbb{T}_N corresponds to the vector space generated by the standard basis $1, \sin(\pi z), \dots, \sin(N\pi z), \cos(N\pi z)$. For general d , the vector-space dimension of this space corresponds to the cardinality of the set $\{\mathbf{n} \in \mathbb{Z}^d : \|\mathbf{n}\|_1 \leq N\}$, which can be easily proved to be

$$\tilde{N} := |\{\mathbf{n} \in \mathbb{Z}^d : \|\mathbf{n}\|_1 \leq N\}| = \binom{2N + d}{d} \leq (2N + 1)^d.$$

From this basis, we can build a feature map by just stacking the terms $\mathbf{n} \in \mathbb{Z}^d : \|\mathbf{n}\|_1 \leq N$ in a vector of functions $e_1(z), \dots, e_{\tilde{N}}(z)$. The corresponding feature map is then

$$\varphi_N : [-1, 1]^d \rightarrow \mathbb{R}^{\tilde{N}}, \quad \varphi_N(x) = [e_1(z), \dots, e_{\tilde{N}}(z)]^\top.$$

Indeed, for every $N \in \mathbb{N}$, we can express any function $g \in \mathbb{T}_N$ as $g(z) = \varphi_N(z)^\top \boldsymbol{\theta}_N(g)$, where the feature map depends on z only and the coefficient vector $\boldsymbol{\theta}_N(g)$ on g only. The good part about this is that we can do the same for functions that do not belong to \mathbb{T}_N , at the cost of having a residual term. In this way, any function f may be decomposed as a sum of a term that is linear in the feature map and a residual term $\xi_N[f](z)$:

$$f(z) = \varphi_N(z)^\top \boldsymbol{\theta}_N(f) + \xi_N[f](z). \quad (2)$$

In fact, $\varphi_N(z)$ is a vector of trigonometric functions of z , while $\boldsymbol{\theta}_N(f)$ is the vector stacking all the coefficients, both having length \tilde{N} , both according to the same arbitrary fixed order. Differently from the case of $f \in \mathbb{T}_N$, the coefficient $\boldsymbol{\theta}_N$ is *not* uniquely determined. If $f \notin \mathbb{T}_N$, this coefficient should be chosen so that some norm of the residual term $\xi_N[f](z)$ gets minimized. If we minimize the L^2 -norm of this quantity, Fourier series (Katznelson, 2004) are the way to go, but different projections on the space \mathbb{T}_N result in different values for the parameter $\boldsymbol{\theta}_N$. Crucially, one of the main features that all these projections share with the Fourier series is that the magnitude of the residual term $\xi_N(f)$ decreases with N at a rate that depends on how smooth the function f is, as shown by the following classical result.

Theorem 1 (Theorem 4.1 part (ii) from Schultz (1969)) *There exists an absolute constant $K > 0$ such that, for any $f \in C_p^{\nu,1}([-1, 1]^d)$ we have:*

$$\inf_{t_N \in \mathbb{T}_N} \|t_N(\cdot) - f(\cdot)\|_{L^\infty} \leq \frac{K}{N^{\nu+1}} \|f\|_{C^{\nu,1}},$$

where \mathbb{T}_N denotes the space of trigonometric polynomials of degree not higher than N .

3.1. Our Approach: Foundations

The idea of the algorithm that we will use to tackle this problem is based on the connection between the decay of the Fourier coefficients and the smoothness of the Bellman operator. Indeed, assuming that the Bellman optimality operator is bounded in the space $\mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A})$, as in Assumption 1, entails the following desirable property. Assume that the optimal state-action value function at the step $h + 1$ has the form $Q_{h+1}^*(s, a) = \varphi_N(s, a)^\top \boldsymbol{\theta}_{h+1}$. Then, by assumption on the Bellman operator, we have:

$$Q_h^*(s, a) = \mathcal{T}_h Q_{h+1}^*(s, a) = \mathcal{T}_h \underbrace{\varphi_N(s, a)^\top \boldsymbol{\theta}_{h+1}}_{\in \mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A})} \in \mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A}),$$

as $\varphi_N(s, a)^\top \boldsymbol{\theta}_{h+1}$ is a trigonometric polynomial, so it is $\mathcal{C}_p^{\nu,1}$ a fortiori. Then, also Q_h^* can be written as $Q_h^*(s, a) = \varphi_N(s, a)^\top \boldsymbol{\theta}_h + \xi_N(Q_h^*)(s, a)$, where $\boldsymbol{\theta}_h \in \mathbb{R}^{\tilde{N}}$ and $\xi_N(Q_h^*)(\cdot)$ is small in infinity norm, according to Theorem 1. If $\xi_N(Q_h^*)(\cdot)$ were exactly zero, this MDP would be said to be linearly Bellman complete w.r.t. the feature map $\varphi_N(\cdot)$. Instead, due to the presence of this error term, the feature map is said to suffer from a *misspecification*. Despite similar scenarios have been previously studied in the literature (Zanette et al., 2020), this misspecification significantly impacts the type of guarantees we are able to achieve. In fact, even in the simpler bandit setting (when $H = 1$) it is well known (Lattimore et al., 2020) that learning a ε -optimal policy, even with an arbitrary number of samples, is only possible when $\varepsilon > \sqrt{\tilde{N}} \|\xi_N(Q_1^*)\|_{L^\infty}$. The fact that the misspecification gets multiplied by $\sqrt{\tilde{N}}$ in the sample complexity turns out not to be improvable in the general case (Lattimore et al., 2020). Still, this has catastrophic effects on our setting: in order to decrease $\xi_N(Q_h^*)$ we have to increase N , but this leads to an increase in \tilde{N} , a trade-off that may be unsolvable. For example, we know that $\tilde{N} = \mathcal{O}(N^d)$, while from Theorem 1 $\|\xi_N(Q_h^*)\|_{L^\infty} = \mathcal{O}(N^{\nu+1})$ when $Q_h^* \in \mathcal{C}_p^{\nu,1}(\mathcal{S} \times \mathcal{A})$. So,

$$d > 2\nu + 2 \implies \sqrt{\tilde{N}} \|\xi_N(Q_h^*)\|_{L^\infty} = \mathcal{O}(N^{d/2-\nu-1}) \geq \mathcal{O}(1).$$

In this case, in the product $\sqrt{\tilde{N}} \|\xi_N(Q_h^*)\|_{L^\infty}$ cannot become arbitrarily small, so that learning an ε -optimal policy using algorithms that deal with misspecification for general feature maps is impossible.

For this reason, we have to change our perspective. Instead of trying to learn Q_h^* , we are going to focus on a projected version \tilde{Q}_h . In what follows, we will define \mathcal{P}_N as a projection operator mapping every function f to a trigonometric polynomial of degree N so that \tilde{Q}_h solves, for every $h = 1, \dots, H$,

$$\tilde{Q}_h = \mathcal{P}_N[\mathcal{T}_h \tilde{Q}_{h+1}](s, a),$$

with $\tilde{Q}_{H+1} = Q_{H+1}^* = 0$. In this way, the misspecification vanishes completely, as starting from $\tilde{Q}_{h+1}(s, a)$, that is a trigonometric polynomial, $\tilde{Q}_h(s, a)$ is parametrized in the same way. This means that, with respect to the feature map φ_N , the operator $\mathcal{P}_N[\mathcal{T}_h]$ makes no misspecification, and, hopefully, we can achieve sample complexity bounds by estimating $\tilde{Q}_h(\cdot)$ with classical results for the LSVI algorithm (Munos, 2005). Two problems need now to be addressed: i) Due to the nature of the problem, we only have access to samples of s_h, a_h, s_{h+1}, r_h from the generative model. How can we approximate the projection operator \mathcal{P}_N in this setting? ii) How do we know that \tilde{Q}_h

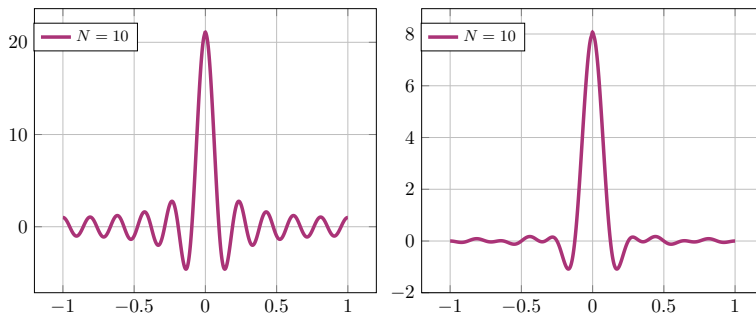


Figure 1: A comparison between Dirichlet Kernel (left) and de la Vallée-Poussin one (right) in one dimension.

is close to Q_h^* for every h ? In the following subsection, we are going to show that both open points can be solved in a satisfactory way due to the particular properties of our trigonometric feature map so that using a different map in place of φ_N would not achieve the same results.

3.2. Projection by Convolution

Let us suppose to have a black-box function $f : \Omega \rightarrow \mathbb{R}$, so that we are able to pick values $x_h \in \Omega$ and receive an unbiased sample $y_h = f(x_h) + \eta_h$, where η_h is a zero-mean noise. One simple trick shows that we can use randomization to obtain samples from functions that are different from f , still using the same generative model. Indeed, let ξ_h be a noise with density function g_ξ . If, after choosing x_h , we perturb it with ξ_h , the resulting sample has a mean satisfying

$$\mathbb{E}[y_h] = \mathbb{E}[f(x_h + \xi_h)] = [f * g_\xi](x_h).$$

This process allows us to obtain unbiased samples from any function \tilde{f} , as long as it can be expressed as a convolution between f and another function. Thus, coming back to our problem, can we use this trick to obtain unbiased samples from $\mathcal{P}_N[f]$, for any choice of the function f ? The answer depends on the choice of the projection operator \mathcal{P}_N that we want to use. If we are interested in the usual Fourier series projection, which minimizes the L^2 -distance from the original function, the solution is provided by the Dirichlet kernel D_N^0 (Katznelson, 2004), a real function that, for $d = 1$ takes the form $D_N^0(x) = \frac{\sin((N+1/2)\pi x)}{\sin(\pi x/2)}$. The main property of this kernel lays in the fact that, for any function f , the convolution product $f * D_N^0$ corresponds exactly to the truncation to the order N of the Fourier series associated to f . This is a possible solution for our main issue: $\|f - g\|_{L^2}$ is indeed minimized across all the trigonometric polynomials of degree N when $g = f * D_N^0$.

We can do even better. As we are interested in a trigonometric polynomial that approximates f uniformly (not necessary in L^2), we may look for a different kernel that corresponds to a projection that is more suited for our case. This can be done by a slight modification of the Dirichlet kernel, which takes the name of de la Vallée Poussin kernel (de la Vallée Poussin, 1918; De La Vallée Poussin et al., 1919; Németh, 2016), that we will denote as D_N . This function is defined, for every positive even integer N , to be an average of Dirichlet Kernels for different orders $D_N(z) := \frac{1}{N/2+1} \sum_{n=N/2}^N D_n^0(z)$. The main feature that makes this kernel superior, for our purposes, to the Dirichlet kernel, is that *its L^1 norm can be bounded regardless of N* . In fact, we

have

$$\|D_N\|_{L^1} \leq \Lambda_d = \mathcal{O}(1), \quad \|D_N^0\|_{L^1} = \mathcal{O}(\log(N)^d), \quad (3)$$

where the first part is rigorously proved in the appendix (Proposition 7). The same proposition allows us to show the following result, which expresses an important property of this kernel D_N .

Theorem 2 *Let $f \in \mathcal{C}_p^{\nu,1}([-1,1]^d)$. Then, $D_N * f \in \mathbb{T}_N$, and we have*

$$\|f - D_N * f\|_{L^\infty} \leq 2\Lambda_d \inf_{t_{N/2} \in \mathbb{T}_{N/2}} \|t_{N/2} - f\|_{L^\infty} \leq \frac{2^{\nu+2}\Lambda_d K}{N^{\nu+1}} \|f\|_{\mathcal{C}^{\nu,1}},$$

where \mathbb{T}_N denotes the space of trigonometric polynomial of degree not higher than N and Λ_d comes from Equation (3).

Clearly, the second inequality comes directly from Theorem 1. This result shows that the operator $D_N * f$ is able to achieve the best approximation of f in \mathbb{T}_N , except for a constant factor $2^{\nu+2}\Lambda_d$. This last result answers both questions of subsection 3.2. Indeed, making the convolution $D_N * f$, which can be estimated by adding to f a noise of density proportional to D_N , has also the role of a projection operator (as $D_N * f \in \mathbb{T}_N$) and good approximation properties ($\|f - D_N * f\|_{L^\infty} = \mathcal{O}(N^{-\nu-1})$).

4. Algorithms

In this section, we introduce the algorithm that we are going to use to achieve optimal sample complexity. Algorithm 1 takes as input the state-action space dimension d , a reference value n_{tot} for the number of interactions with the simulator that we are going to collect at each stage, and the approximation degree N of the feature map. The first part of the algorithm, until Line 10 excluded, is devoted to choosing the state-action couples where to query the generative model. Here, the idea is to define a quasi-optimal design ρ for least-squares regression over the set \mathcal{X}_φ defined in Line 1. Intuitively, a quasi-optimal design is a distribution over \mathcal{X}_φ with few support points ($\mathcal{O}(\tilde{N} \log \log(\tilde{N}))$) such that its linear features "cover" all the ones in \mathcal{X}_φ . As this concept is relatively standard in RL theory, we leave further explanations to Appendix B. Once having chosen the design ρ , we collect in the set $S_{s,a}$ every point in the support of ρ for a number of times given by $\lceil n_{\text{tot}}\rho(s,a) \rceil$. Note that, because of the ceiling, the total number of samples is not exactly n_{tot} , but slightly higher. From Line 10 onward, we have just a perturbed version of the celebrated LSVI algorithm by Munos (2005). The main differences with respect to standard LSVI are highlighted in Algorithm 1 in green and red. As we can see, instead of just querying the generative model for the elements of the buffers \mathcal{D}_h , we perturb them with a noise sampled from the positive and negative parts $\sim D_N^+, \sim D_N^-$ of the Vallée de la Poussin Kernel. Ideally, this is done because we want unbiased samples from $D_N * \mathcal{T}_h \tilde{Q}_{h+1}$, which can be done by adding a perturbation from $D_N(\cdot)$ to the state-action couple. However, we cannot take random samples from something that is not a density function, so we have to decompose D_N in the following way:

$$D_N(\cdot) = \beta_+ D_N^+(\cdot) - \beta_- D_N^-(\cdot), \quad (4)$$

with $D_N^+(z), D_N^-(z)$ being the positive and negative parts of the kernel, both normalized to have integral one, and β^+, β^- are constants such that the equality is verified. Now, we can just

Algorithm 1: Quasi-optimal Design + Perturbed LSVI

Data: State-action space $\mathcal{Z} = [-1, 1]^d$, reference value n_{tot} , approximation degree N

Result: Policy $\hat{\pi}$.

```

1 Let  $\mathcal{X}_\varphi := \{\varphi_N(s, a) : (s, a) \in \mathcal{Z}\}$ 
2 Compute quasi-optimal design  $\rho$  for the set  $\mathcal{X}_\varphi$ 
3 for  $h = 1, \dots, H$  do
4    $\mathcal{D}_h \leftarrow \emptyset$ 
5   for  $s, a \in \text{supp}(\rho)$  do
6     Initialize  $S_{s,a}$  with  $\lceil n_{\text{tot}}\rho(s, a) \rceil$  copies of  $s, a$ 
7      $\mathcal{D}_h \leftarrow \mathcal{D}_h \cup S_{s,a}$ 
8   end
9 end
10 Find  $D_N^+, D_N^-, \beta_+, \beta_-$  from Equation (4)
11 for  $h = H, \dots, 1$  do
12   for  $z_h = (s_h, a_h) \in \mathcal{D}_h$  do
13      $\eta_+ \sim D_N^+(\cdot)$ 
14      $\eta_- \sim D_N^-(\cdot)$ 
15     Query generative model for  $s_{h+1}^+ \sim p_h(\cdot | z_h + \eta_+)$ ,  $r_h^+ \sim R_h(z_h + \eta_+)$ 
16     Query generative model for  $s_{h+1}^- \sim p_h(\cdot | z_h + \eta_-)$ ,  $r_h^- \sim R_h(z_h + \eta_-)$ 
17      $Q_h^{\text{TARGET}}(s_h, a_h) = (\beta_+ r_h^+ - \beta_- r_h^-) + (\beta_+ \hat{V}_{h+1}(s_{h+1}^+) - \beta_- \hat{V}_{h+1}(s_{h+1}^-))$ 
18   end
19   Solve least squares
20     
$$\hat{\theta}_h = \arg \min_{\theta \in \mathbb{R}^N} \sum_{(s_h, a_h) \in \mathcal{D}_h} \left( \varphi_N(s_h, a_h)^\top \theta - Q_h^{\text{TARGET}}(s_h, a_h) \right)^2$$

21   Compute  $\hat{Q}_h(\cdot, \cdot) = \varphi_N(\cdot, \cdot)^\top \hat{\theta}_h$ 
22   Compute next state-value function  $\hat{V}_h(\cdot) = \max_{a \in \mathcal{A}} \{\varphi_N(\cdot, a)^\top \hat{\theta}_h\}$ 
23 end
24 Return  $\hat{\pi}_h(s) := \arg \max_{a \in \mathcal{A}} \hat{Q}_h(s, a)$ 

```

take samples from D_N^+, D_N^- and apply Equation (4) to have an unbiased estimator (Line 17) for $D_N * \mathcal{T}_h \hat{Q}_{h+1}(s_h, a_h)$, which is what we need to keep our Q-function estimates within the vector space spanned by the features $\varphi_N(\cdot, \cdot)$. Note that, despite the modifications, as long as we have a good random number generator, the computational complexity of the algorithm does not increase significantly compared to LSVI: finding ρ is computationally feasible, as shown by Todd (2016, Lemma 3.9), and the second part has the same complexity of LSVI. Additional reasonings of computational efficiency are left to the appendix F.

As anticipated, the reference value n_{tot} does not correspond exactly to the number of samples that we take for each time-step h . In fact, we have that the actual number of samples corresponds to $\sum_{s,a \in \text{supp}(\rho)} \lceil n_{\text{tot}}\rho(s, a) \rceil \leq |\text{supp}(\rho)| + \sum_{s,a \in \text{supp}(\rho)} n_{\text{tot}}\rho(s, a) = |\text{supp}(\rho)| + n_{\text{tot}}$. From this reasoning, we have the following proposition.

Proposition 3 *The total number of samples required by Algorithm 1 in every phase is $2(n_{\text{tot}} + 4\tilde{N} \log \log(\tilde{N}))$.*

Proof The factor $n_{\text{tot}} + 4\tilde{N} \log \log(\tilde{N})$ directly comes from $n \leq n_{\text{tot}} + |\text{supp}(\rho)|$ and the bound on the support given by definition of quasi-optimal design (Proposition 8). The factor 2 comes from the double sampling in Lines 15 and 16. \blacksquare

4.1. Theoretical Guarantees

In this section, we introduce the main result of this paper, which ensures that our algorithm can be used on a Smooth MDP to achieve order-optimal sample complexity.

Theorem 4 *Fix $\varepsilon > 0, \delta > 0$. Apply Algorithm 1 on a ν -Smooth MDP. Then, for a proper choice of $N = \mathcal{O}(\varepsilon^{-\frac{1}{\nu+1}})$, with probability at least $1 - \delta$, we are able to learn a policy $\hat{\pi}$ that is ε -optimal with a number of samples given by*

$$n = \tilde{\mathcal{O}} \left(K(H, d) \varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}} \right).$$

The full proof is very long, and we postpone it to Appendix C. Unfortunately, the constant $K(H, d)$ from the previous theorem is exponential in both H and d . In the next section, after proving that the exponent $-\frac{d+2(\nu+1)}{\nu+1}$ is tight, and no better result can be achieved, we also discuss this drawback.

4.2. Tightness of the result and comparison with state of the art

In this section, we discuss the result of Theorem 4. We start from the question about the tightness of the order of the sample complexity, for which a lower bound from the field of stochastic optimization proves that our exponent is the best possible.

Theorem 5 (*Wang et al. (2018) Theorem 2 + Proposition 3 for $\beta = 0$*) *Consider the optimization of an unknown function $f \in \mathcal{C}^{\nu,1}([-1, 1]^d)$ via an algorithm that can only access f through noisy samples of the form $y_t = f(x_t) + \eta_t$, where x_t are points chosen by the algorithm and $\{\eta_t\}_{t=1}^n$ is a sequence of independent 1-subgaussian noises. Then, the expected number of samples required to find a ε -optimal point \hat{x} satisfies $n = \Omega \left(\varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}} \right)$.*

Note that the setting of the previous theorem is far easier than ours. First, the previous theorem holds in expectation, while our Theorem 4 is valid in high probability. Second, and more importantly, the RL setting with a generative model reduces to stochastic optimization when $H = 1$, so that the reward function $r_1(\cdot)$ corresponds to the function $f(\cdot)$ to be optimized in Theorem 5. Therefore, Theorem 5 implies *a fortiori* a lower bound of order $n = \Omega \left(\varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}} \right)$ in our setting, which matches our result for what concerns the dependence on ε . This result is completely new in the literature of RL with continuous spaces. Indeed, by converting regret bounds into sample complexity bounds, on the one hand, we have algorithms for Lipschitz MDPs (Song and Sun, 2019; Sinclair et al., 2019; Le Lan et al., 2021), which achieve only $\tilde{\mathcal{O}}(\varepsilon^{-d-2})$ whatever the smoothness

of the process, on the other hand, LEGENDRE-ELEANOR (Maran et al., 2024) which only works for $d < 2\nu + 2$ and gives $\tilde{\mathcal{O}}(\varepsilon^{-\frac{d+2(\nu+1)}{-d/2+\nu+1}})$, which is never tight.

Now, we move on to explore the dependence of the sample complexity on the task horizon H and the dimensionality of the state-action space d . The exponential dependence found in our Theorem 4 is not encouraging, but we can prove that this annoying phenomenon cannot be avoided in the whole family of Lipschitz MDPs (defined in the Appendix D), which are a subset of the Smooth MDPs for $\nu = 0$ (Maran et al., 2024). Indeed, for what concerns the time horizon H and the dimension d , the following theorem, which is analogous to Theorem 12 in Maran et al. (2024), shows that for any setting containing Lipschitz MDP, the exponential dependence on both d and H is necessary.

Theorem 6 *Every algorithm for Lipschitz MDPs requires a sample complexity of order $\Omega(C^{dH} \varepsilon^{-d-2})$, for some constant $C > 1$ depending on the process.*

Proof can be found in the Appendix D. Note that all the results in the state of the art obtaining bounds that are sub-exponential in H often hide this dependence in a problem-dependent constant. For example, Theorem 4.1 in Sinclair et al. (2019) provides a regret bound depending on a constant c , that for Lipschitz MDPs reveals in Proposition 2.5 to be exponential in H . For what concerns the dependence on H , this result constitutes a significant problem, as it completely prevents Lipschitz MDPs with long horizons from being solved efficiently without further assumptions.

Extension for misspecified ν . As we have proved in Theorem 4, our algorithm is able to achieve optimal sample complexity for every ν -Smooth MDP. Still, to achieve this result, an exact knowledge of the smoothness parameter ν is required, as it is not possible to compute the optimal number of features N without it. In many practical scenarios, the knowledge of the exact ν is rather unrealistic, so we study what happens in case the algorithm is run for a parameter ν_{miss} that is different from the correct value ν . Taking $\nu_{\text{miss}} > \nu$ leads to the impossibility of guaranteeing the learning of an ε -optimal policy. Since, as we prove in the appendix (theorem 9), the error $\|\tilde{Q}_h - Q_h^*\|_\infty \leq \Psi_h N^{-\nu-1}$, taking $N = \mathcal{O}(\varepsilon^{-\frac{1}{\nu_{\text{miss}}+1}})$ as prescribed by theorem 4, leads to

$$\forall \varepsilon < 1 \quad \|\tilde{Q}_h - Q_h^*\|_\infty = \mathcal{O}\left(\varepsilon^{\frac{\nu+1}{\nu_{\text{miss}}+1}}\right) > \mathcal{O}(\varepsilon).$$

Clearly, this kind of approximation does not allow us to learn a ε -optimal policy, even if \tilde{Q}_h were known exactly. In the opposite case, when $\nu_{\text{miss}} < \nu$, we can just exploit the fact that $\mathcal{C}^{\nu,1}(\mathcal{S}, \mathcal{A}) \subset \mathcal{C}^{\nu_{\text{miss}},1}(\mathcal{S}, \mathcal{A})$ and apply the bound we would obtain if the MDP were only ν_{miss} -Smooth. This writes as

$$n = \tilde{\mathcal{O}}\left(\varepsilon^{-\frac{d+2(\nu_{\text{miss}}+1)}{\nu_{\text{miss}}+1}}\right),$$

which is obviously worse than the one for ν , but still finite.

5. Conclusions and future works

In this work, we have significantly enlarged the class of continuous-spaces MDPs for which sample-efficient learning with a generative model is possible. We focus on the Smooth MDP setting, defined first Maran et al. (2024), which includes the majority of the classes of continuous spaces MDPs for which sample efficient RL is possible. To solve this very general problem, we design an algorithm

that is able to achieve optimal sample complexity in ε for every dimension d and smoothness parameter ν , thus, reaching a lower bound that cannot be improved even in the simpler bandit scenario. This algorithm builds on a novel regression technique that perturbs the sample with a noise distribution coming from the De La Vallée Poussin kernel, which is a contribution of independent interest. The sample complexity bound is exponential in both d and H , but we show a lower bound ensuring that, even in simpler settings, this drawback cannot be avoided.

Future work. This work heavily relies on the assumption of interacting with a simulator during learning. A next advance could be to study if the same sample complexity can be achieved by just interacting online with the MDP. Moreover, a natural question in that setting is if the sample complexity bound can be turned into a regret bound. An intermediate step could be to study the applicability of this method to *local access*—an intermediate setting between generative model and online RL, where the agent can reset the environment to previously visited states (Yin et al., 2023). Another potential application is to *offline RL*, where the dataset is given in advance (Levine et al., 2020). Finally, it remains to be seen whether the dependence on the task horizon can be improved for interesting sub-families of smooth MDPs.

Acknowledgments

Funded by the European Union – Next Generation EU within the project NRPP M4C2, Investment 1.,3 DD. 341 - 15 march 2022 – FAIR – Future Artificial Intelligence Research – Spoke 4 - PE00000013 - D53C22002380006. We thank the prof Maura Elisabetta Salvatori from the Department of Mathematics at Milano University for having introduced the first author of this paper to the de la Vallè Poussin kernel.

References

- Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26. JMLR Workshop and Conference Proceedings, 2011.
- Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.
- Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.
- Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Mach. Learn.*, 91(3): 325–349, 2013.
- Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
- Dimitri Bertsekas and Steven E Shreve. *Stochastic optimal control: the discrete-time case*, volume 5. Athena Scientific, 1996.
- Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only \sqrt{t} regret. In *International Conference on Machine Learning*, pages 1300–1309. PMLR, 2019.
- Ch de la Vallée Poussin. Sur la meilleure approximation des fonctions d’une variable réelle par des expressions d’ordre donné. *CR Acad. Sci. Paris*, 166:799–802, 1918.
- Ch J De La Vallée Poussin et al. *Leçons sur l’approximation des fonctions d’une variable réelle*. Paris, 1919.
- Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. *Advances in Neural Information Processing Systems*, 31, 2018.
- Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.
- Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503, 2023.
- Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.
- Yitzhak Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004.

- Jack Kiefer and Jacob Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.
- B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.
- Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.
- Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR, 2020.
- Charline Le Lan, Marc G Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8261–8269, 2021.
- Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- Pierre Liotet, Davide Maran, Lorenzo Bisi, and Marcello Restelli. Delayed reinforcement learning by imitation. In *International Conference on Machine Learning*, pages 13528–13556. PMLR, 2022.
- Davide Maran, Alberto Maria Metelli, and Marcello Restelli. Tight performance guarantees of imitator policies with continuous actions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9073–9080, 2023.
- Davide Maran, Alberto Maria Metelli, Matteo Papini, and Marcello Restelli. No-regret reinforcement learning in smooth mdps. *arXiv preprint arXiv:2402.03792*, 2024.
- Harsh Mehta. The 11 norms of de la vallée poussin kernels. *Journal of Mathematical Analysis and Applications*, 422(2):825–837, 2015.
- Zak Mhammedi, Adam Block, Dylan J Foster, and Alexander Rakhlin. Efficient model-free exploration in low-rank mdps. *Advances in Neural Information Processing Systems*, 36, 2024.
- Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.
- Zsolt Németh. De la vallée poussin type approximation methods. 2016.
- Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100:255–283, 2015.
- Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

- Emmanuel Rachelson and Michail G Lagoudakis. On the locality of action domination in sequential decision making. 2010.
- Martin H Schultz. l^∞ -multivariate approximation theory. *SIAM Journal on Numerical Analysis*, 6(2):161–183, 1969.
- Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. Adaptive discretization for episodic reinforcement learning in metric spaces. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–44, 2019.
- Zhao Song and Wen Sun. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019.
- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- Michael J Todd. *Minimum-volume ellipsoids: Theory and algorithms*. SIAM, 2016.
- Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. In *International Conference on Learning Representations*, 2021.
- Sattar Vakili and Julia Olkhovskaya. Kernelized reinforcement learning with order optimal regret bounds. *arXiv preprint arXiv:2306.07745*, 2023.
- Yining Wang, Sivaraman Balakrishnan, and Aarti Singh. Optimization of smooth functions with noisy observations: Local minimax rates. *Advances in Neural Information Processing Systems*, 31, 2018.
- Weidong Wu, Joseph Owino, Ahmed Al-Ostaz, and Liguang Cai. Applying periodic boundary conditions in finite element analysis. In *SIMULIA community conference, Providence*, pages 707–719, 2014.
- Yihong Wu. Lecture notes on information-theoretic methods for high-dimensional statistics. *Lecture Notes for ECE598YW (UIUC)*, 16, 2017.
- Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 6995–7004. PMLR, 2019.
- Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I. Jordan. Provably efficient reinforcement learning with kernel and neural function approximations. In *NeurIPS*, 2020.
- Dong Yin, Sridhar Thiagarajan, Nevena Lazic, Nived Rajaraman, Botao Hao, and Csaba Szepesvari. Sample efficient deep reinforcement learning via local planning. *arXiv preprint arXiv:2301.12579*, 2023.
- Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Appendix A. Harmonic Analysis

A.1. Formal definition of de la Vallée-Poussin kernel

Formally, de la Vallée-Poussin kernels indicate a family of functions with two different parameters, n, m , which balance between some properties of the kernel (see [Németh, 2016](#), Equation 2.6). In this paper, we always fix $n = m = N$, so that the kernel can be written as:

$$D_{2N} := \frac{1}{N+1} \sum_{n=N}^{2N} D_n^0, \quad (5)$$

where D_n^0 denotes the order- n standard Dirichlet kernel. This equation shows that the de la Vallée-Poussin kernel is actually just a superposition of Dirichlet kernels of different orders, and some of its properties follow from this. In particular, we need the following result, which is completely based on theorems discovered by [Mehta \(2015\)](#) and [Németh \(2016\)](#).

Proposition 7 *The following properties hold for the de la Vallée-Poussin kernel (Equation 5)*

1. For any periodic function $f : \Omega \rightarrow \mathbb{R}$, $D_N * f$ is a trigonometric polynomial of degree not higher than N .
2. For any trigonometric polynomial f of degree not higher than $N/2$, $D_N * f = f$.
3. There is a constant Λ_d depending only on the dimension d such that $\|D_N\|_{L^1} \leq \Lambda_d$.

Proof We prove the thesis point by point.

1. By linearity of the convolution, we have, from Equation (5):

$$D_N * f = \frac{1}{N/2+1} \sum_{n=N/2}^N D_n^0 * f.$$

By the properties of the Dirichlet kernel, $D_n^0 * f$ is the Fourier series of f truncated on n . As $n \leq N$, all of these functions are trigonometric polynomials of degree not higher than N .

2. Let f be any trigonometric polynomial f of degree not higher than $N/2$, then from Equation (5):

$$\begin{aligned} D_{2N} * f &= \frac{1}{N/2+1} \sum_{n=N/2}^N D_n^0 * f \\ &= \frac{1}{N/2+1} \sum_{n=N/2}^N f = f, \end{aligned}$$

where the second passage comes from the fact that the Fourier series of order $n \geq N$ of f corresponds exactly to f itself.

3. We apply Theorem 2.4 from [Németh \(2016\)](#). Indeed, we have the following chain of inequalities:

$$\begin{aligned}
 \|D_N\|_{L^1} &= \sup_{g \in \mathcal{C}^0(\Omega), \|g\|_\infty \leq 1} \int_{\Omega} D_N(x)g(x) dx \\
 &= \sup_{g \in \mathcal{C}^0(\Omega), \|g\|_\infty \leq 1} \sup_{y \in \Omega} \int_{\Omega} D_N(x)g(y-x) dx \\
 &= \sup_{g \in \mathcal{C}^0(\Omega), \|g\|_\infty \leq 1} \sup_{y \in \Omega} |D_N * g(y)| \\
 &= \sup_{g \in \mathcal{C}^0(\Omega), \|g\|_\infty \leq 1} \|D_N * g\|_\infty.
 \end{aligned}$$

The latter corresponds exactly to the norm of the operator $\cdot \rightarrow D_N * \cdot$ over the space of continuous functions, which is bounded in Theorem 2.4 by [Németh \(2016\)](#) as

$$\sup_{g \in \mathcal{C}^0(\Omega), \|g\|_\infty \leq 1} \|D_N * g\|_\infty \leq c \left(\left(\log \frac{2N}{N+1} \right)^d + 1 \right) \leq c(\log^d 2 + 1).$$

Taking $\Lambda_d := c(\log^d 2 + 1)$ ends the proof. ■

Theorem 2 *Let $f \in C_p^{\nu,1}([-1, 1]^d)$. Then, $D_N * f \in \mathbb{T}_N$, and we have*

$$\|f - D_N * f\|_{L^\infty} \leq 2\Lambda_d \inf_{t_{N/2} \in \mathbb{T}_{N/2}} \|t_{N/2} - f\|_{L^\infty} \leq \frac{2^{\nu+2}\Lambda_d K}{N^{\nu+1}} \|f\|_{C^{\nu,1}},$$

where \mathbb{T}_N denotes the space of trigonometric polynomial of degree not higher than N and Λ_d comes from Equation (3).

Proof The fact that $D_N * f$ is a trigonometric polynomial of degree not higher than N is part one of Proposition 7. Let $t_{N/2} \in \mathbb{T}_{N/2}$, we have:

$$\begin{aligned}
 \|f - D_N * f\|_{L^\infty} &= \|f - t_{N/2}\|_{L^\infty} + \|t_{N/2} - D_N * f\|_{L^\infty} \\
 &= \|f - t_{N/2}\|_{L^\infty} + \|D_N * t_{N/2} - D_N * f\|_{L^\infty}.
 \end{aligned}$$

Here, the last passage comes from part two of Proposition 7. Lastly, we have

$$\|D_N * t_{N/2} - D_N * f\|_{L^\infty} \leq \|D_N\|_{L^1} \|t_{N/2} - f\|_{L^\infty} \leq \Lambda_d \|t_{N/2} - f\|_{L^\infty},$$

by the third part of Proposition 7. In the end, we have proved that, being $\Lambda_d \geq 1$,

$$\begin{aligned}
 \|f - D_N * f\|_{L^\infty} &\leq \|f - t_{N/2}\|_{L^\infty} + \|D_N * t_{N/2} - D_N * f\|_{L^\infty} \\
 &\leq 2\Lambda_d \|t_{N/2} - f\|_{L^\infty}.
 \end{aligned}$$

The proof is completed by applying Theorem 1. ■

Appendix B. Optimal Design for the Least Squares Problem

The construction of the datasets \mathcal{D}_h is one crucial part of Algorithm 1. This will be done in a way that is similar to [Lattimore et al. \(2020\)](#), by applying the concept of optimal design for the least squares estimator. To this aim, we recall the following theorem, which can be found in ([Lattimore et al., 2020](#), Theorem 4.3) and is based on a previous result by [Kiefer and Wolfowitz \(1960\)](#).

Theorem 8 ([Lattimore et al. \(2020\)](#), Theorem 4.4) *Suppose $\mathcal{X} \subset \mathbb{R}^{\tilde{N}}$ is a compact set spanning $\mathbb{R}^{\tilde{N}}$. Then, there is a probability distribution ρ on \mathcal{X} such that $|\text{supp}(\rho)| \leq 4\tilde{N} \log \log(\tilde{N})$ and, once defined*

$$\Sigma = \mathbb{E}_{\mathbf{x} \sim \rho} [\mathbf{x}\mathbf{x}^\top],$$

we have, for all $\mathbf{x} \in \Omega$, $\|\mathbf{x}\|_{\Sigma^{-1}}^2 \leq 2\tilde{N}$ (here the notation $\|\mathbf{x}\|_{\Sigma^{-1}}$ stands for $\sqrt{\mathbf{x}^\top \Sigma^{-1} \mathbf{x}}$).

We call the distribution ρ defined in Theorem 8 *quasi-optimal design* for the least square problem. The reason under the term "quasi" in the previous definition comes from a previous result. In fact, in a classical and celebrated paper, [Kiefer and Wolfowitz \(1960\)](#) showed that, among all distribution over \mathcal{X} , we have always $\|\mathbf{x}\|_{\Sigma^{-1}}^2 \geq \tilde{N}$. Moreover, a distribution ρ with at most $\tilde{N}(\tilde{N} + 1)$ support points can be found such that the equality $\|\mathbf{x}\|_{\Sigma^{-1}}^2 = \tilde{N}$ is attained. This distribution takes the name of *optimal design*. Instead, as we need a support much smaller than $\tilde{N}(\tilde{N} + 1)$, we have relied on quasi-optimal design, which ensures $|\text{supp}(\rho)| \leq 4\tilde{N} \log \log(\tilde{N})$, still having $\|\mathbf{x}\|_{\Sigma^{-1}}^2 \leq 2\tilde{N}$.

Appendix C. Proof of Theorem 4

We start with a preliminary result that will be used to bound the difference between \tilde{Q}_h and the original Q_h .

Theorem 9 *Let us assume to be in a ν -Smooth MDP (Assumption 1). Let \tilde{Q}_h be the sequence of perturbed state action value functions, satisfying $\tilde{Q}_h(s, a) = [D_N * \mathcal{T}_h \tilde{Q}_{h+1}](s, a)$. Then,*

$$1) \|\tilde{Q}_h\|_{\mathcal{C}^{\nu,1}} \leq \frac{(\Lambda_d C_{\mathcal{T}})^{H-h+1} - \Lambda_d C_{\mathcal{T}}}{\Lambda_d C_{\mathcal{T}} - 1}, \quad 2) \|\tilde{Q}_h - Q_h^*\|_{\infty} \leq \Psi_h N^{-\nu-1}$$

where the constant Ψ_h is defined as

$$\Psi_h := 2^{\nu+2} K(H-h+1) \frac{(\Lambda_d C_{\mathcal{T}})^{H-h+1} - \Lambda_d C_{\mathcal{T}}}{\Lambda_d C_{\mathcal{T}} - 1}.$$

Proof We perform the proof by backward induction on h , proving the two parts of the thesis at the same time.

Base case: for $h = H + 1$, both \tilde{Q}_h, Q_h^* are identically zero, so both the theses hold.

Inductive case: We start from the first thesis.

$$\begin{aligned} \|D_N * f\|_{\mathcal{C}^{\nu,1}} &= \max_{|\alpha| \leq \nu+1} \|D^\alpha (D_N * f)\|_{L^\infty} \\ &= \max_{|\alpha| \leq \nu+1} \|D_N * D^\alpha f\|_{L^\infty} \\ &\leq \|D_N\|_{L^1} \max_{|\alpha| \leq \nu+1} \|D^\alpha f\|_{L^\infty} = \|D_N\|_{L^1} \|f\|_{\mathcal{C}^{\nu,1}}. \end{aligned} \quad (6)$$

Here, the first statement follows from the definition of norm in $\mathcal{C}^{\nu,1}$, the second by the fact that the derivative of a convolution can be applied to any of the two functions, the third from the inequality $\|f * g\|_{L^\infty} \leq \|f\|_{L^\infty} \|g\|_{L^1}$.

Applying the previous inequality at passage (\star) proves that

$$\begin{aligned}
 \|\tilde{Q}_h\|_{\mathcal{C}^{\nu,1}} &= \|D_N * \mathcal{T}_h \tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} \\
 &\stackrel{(\star)}{\leq} \|D_N\|_1 \|\mathcal{T}_h \tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} \\
 &\leq \|D_N\|_1 C_{\mathcal{T}} (\|\tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} + 1) \\
 &\leq \Lambda_d C_{\mathcal{T}} (\|\tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} + 1) \\
 &\leq \Lambda_d C_{\mathcal{T}} \left(\frac{(\Lambda_d C_{\mathcal{T}})^{H-h} - \Lambda_d C_{\mathcal{T}}}{\Lambda_d C_{\mathcal{T}} - 1} + 1 \right) \\
 &= \Lambda_d C_{\mathcal{T}} \left(\frac{(\Lambda_d C_{\mathcal{T}})^{H-h} - 1}{\Lambda_d C_{\mathcal{T}} - 1} \right) \\
 &= \frac{(\Lambda_d C_{\mathcal{T}})^{H-h+1} - \Lambda_d C_{\mathcal{T}}}{\Lambda_d C_{\mathcal{T}} - 1}.
 \end{aligned}$$

Where we have used induction in the fourth inequality. The proof of this point is completed by bounding $\|D_N\|_{L^1}$ with Proposition 7.

Let us turn to the second point, and note that

$$\begin{aligned}
 \tilde{Q}_h - Q_h^* &= D_N * \mathcal{T}_h \tilde{Q}_{h+1} - \mathcal{T}_h Q_{h+1}^* \\
 &= D_N * \mathcal{T}_h \tilde{Q}_{h+1} - \mathcal{T}_h \tilde{Q}_{h+1} + \mathcal{T}_h \tilde{Q}_{h+1} - \mathcal{T}_h Q_{h+1}^*.
 \end{aligned}$$

Passing to the norms, $\|\mathcal{T}_h \tilde{Q}_{h+1} - \mathcal{T}_h Q_{h+1}^*\|_{\infty} \leq \|\tilde{Q}_{h+1} - Q_{h+1}^*\|_{\infty}$, as the Bellman optimality operator is non-expansive in infinity norm. Coming to the other part we know, by the assumption of smoothness that the Bellman optimality operator is bounded over $\mathcal{C}^{\nu,1}(\mathcal{S} \times \mathcal{A}) \rightarrow \mathcal{C}^{\nu,1}(\mathcal{S} \times \mathcal{A})$, so that $\|\mathcal{T}_h \tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} \leq C_{\mathcal{T}} (\|\tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} + 1)$. Therefore, we have, by theorem 2:

$$\begin{aligned}
 \|D_N * \mathcal{T}_h \tilde{Q}_{h+1} - \mathcal{T}_h \tilde{Q}_{h+1}\|_{\infty} &\leq \frac{2^{\nu+2} \Lambda_d K}{N^{\nu+1}} \|\mathcal{T}_h \tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} \\
 &\leq C_{\mathcal{T}} \left(\frac{2^{\nu+2} \Lambda_d K}{N^{\nu+1}} \|\mathcal{T}_h \tilde{Q}_{h+1}\|_{\mathcal{C}^{\nu,1}} + 1 \right).
 \end{aligned}$$

Putting all pieces together ends the proof. ■

We prove a second theorem which is the main ingredient of the central proof.

Theorem 10 Fix $0 < \varepsilon < 1, \delta > 0$. Apply Algorithm 1 on a ν -Smooth MDP. Then, for a proper choice of $N = \mathcal{O}(\varepsilon^{-\frac{1}{\nu+1}})$, with probability at least $1 - \delta$, we are able to learn a function \hat{Q}_h such that $\|Q_h^* - \hat{Q}_h\|_{L^\infty} < \varepsilon_h$ and $\|\tilde{Q}_h - \hat{Q}_h\|_{L^\infty} < \varepsilon_h$ for

$$\varepsilon_h := \frac{\varepsilon}{(3\Lambda_d)^h},$$

With a number of samples per layer h given by

$$n = \tilde{\mathcal{O}} \left(K(H, d) \varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}} \right),$$

where $K(H, d)$ is a constant that is exponential in both quantities.

Proof We prove the theorem by induction. Case $h = H+1$ is trivial, as both functions $\tilde{Q}_{H+1}, \hat{Q}_{H+1}$ are identically zero.

Part I: Bounding the Bellman error. By design, at time step h , the algorithm performs a linear regression from $\varphi_N(s, a)$ to $D_N * \mathcal{T}_h \hat{Q}_{h+1}(s, a)$. Let us call $z_h = (s_h, a_h)$ an action pair in the dataset \mathcal{D}_h . The training set for the regression takes the form:

$$\{\varphi_N(z_h); Q_h^{\text{TARGET}}[z_h]\}_{z_h \in \mathcal{D}_h} = \{\varphi_N(z_h); \tilde{r}_{h+1}[z_h] + \tilde{V}_{h+1}[z_h]\}_{z_h \in \mathcal{D}_h},$$

where we have split $Q_h^{\text{TARGET}}[z_h]$ in two terms,

$$\tilde{V}_{h+1}[z_h] = \beta_+ V_{h+1}(s_{h+1}^+) - \beta_- V_{h+1}(s_{h+1}^-) \quad \tilde{r}_{h+1}[z_h] = \beta_+ r_h^+ - \beta_- r_h^-,$$

with $V_h(\cdot) = \max_{a \in \mathcal{A}} \varphi_N(\cdot, a)^\top \hat{\boldsymbol{\theta}}_{h+1} = \max_{a \in \mathcal{A}} \hat{Q}_{h+1}(\cdot, a)$, and $s_{h+1}^+ \sim p_h(\cdot | z_h + \eta_+)$, $r_h^+ = R_h(z_h + \eta_+)$ and $s_{h+1}^- \sim p_h(\cdot | z_h + \eta_-)$, $r_h^- = R_h(z_h + \eta_-)$. By definition of the injected noise we have:

$$\begin{aligned} \mathbb{E}[\tilde{r}_{h+1}[z_h] | z_h] &= \mathbb{E}[\beta_+ r_h^+ - \beta_- r_h^- | z_h] \\ &= D_N * (R_{h+1})(z_h), \end{aligned}$$

while for the other part,

$$\begin{aligned} \mathbb{E}[\tilde{V}_{h+1}[z_h] | z_h] &= \mathbb{E}[\beta_+ V_{h+1}(s_{h+1}^+) - \beta_- V_{h+1}(s_{h+1}^-) | z_h] \\ &= \beta_+ \mathbb{E}[V_{h+1}(s_{h+1}^+) | z_h] - \beta_- \mathbb{E}[V_{h+1}(s_{h+1}^-) | z_h] \\ &= \beta_+ \mathbb{E}_{\substack{s_{h+1}^+ \sim p_h(\cdot | z_h + \eta_+) \\ \eta_+ \sim D_N^+}} \left[\max_{a \in \mathcal{A}} \hat{Q}_{h+1}(s_{h+1}^+, a) | z_h \right] \\ &\quad - \beta_- \mathbb{E}_{\substack{s_{h+1}^- \sim p_h(\cdot | z_h + \eta_-) \\ \eta_- \sim D_N^-}} \left[\max_{a \in \mathcal{A}} \hat{Q}_{h+1}(s_{h+1}^-, a) | z_h \right] \\ &= \beta_+ \int_{\mathcal{S} \times \mathcal{A}} D_N^+(y) \int_{\mathcal{S}} \max_{a \in \mathcal{A}} \hat{Q}_{h+1}(x, a) p(x | z_h + y) dx dy \\ &\quad - \beta_- \int_{\mathcal{S} \times \mathcal{A}} D_N^-(y) \int_{\mathcal{S}} \max_{a \in \mathcal{A}} \hat{Q}_{h+1}(x, a) p(x | z_h + y) dx dy \\ &= \int_{\mathcal{S} \times \mathcal{A}} D_N(y) \int_{\mathcal{S}} \max_{a \in \mathcal{A}} \hat{Q}_{h+1}(x, a) p(x | z_h + y) dx dy \\ &= D_N * [\max_{a \in \mathcal{A}} \hat{Q}_{h+1}](z_h). \end{aligned}$$

Where the first passage is by definition of \tilde{V}_h , the third by definition of V_h , the fourth by rewriting the expectation as an integral and the last one by definition of convolution (kernel D_N is symmetric, so adding or subtracting makes no difference). By definition of Bellmann operator, this proves that

$$\mathbb{E}[\tilde{r}_{h+1}[z_h] + \tilde{V}_{h+1}[z_h]|z_h] = D_N * \mathcal{T}\widehat{Q}_{h+1}(z_h). \quad (7)$$

Moreover, note that we have

$$\begin{aligned} \|\tilde{V}_{h+1}(\cdot)\|_{L^\infty} &\leq \|\beta_+ V_{h+1}(\cdot)\|_{L^\infty} + \|\beta_- V_{h+1}(\cdot)\|_{L^\infty} \\ &\leq (\beta_+ + \beta_-) \|V_{h+1}(\cdot)\|_{L^\infty} \\ &\leq \Lambda_d \|\widehat{Q}_{h+1}(\cdot)\|_{L^\infty} \\ &\leq \Lambda_d (\|\widehat{Q}_{h+1}(\cdot) - Q_{h+1}^*(\cdot)\|_{L^\infty} + \|Q_{h+1}^*(\cdot)\|_{L^\infty}) \\ &\leq \Lambda_d (\varepsilon_h + \|Q_{h+1}^*(\cdot)\|_{L^\infty}) \\ &\leq \Lambda_d (\varepsilon_h + H - h) \\ &\leq \Lambda_d (H + 1) \end{aligned}$$

where we have used the definition of \tilde{V}_{h+1} for the first inequality, the fact that $\beta_+ + \beta_- = \Lambda_d$ and the definition of \widehat{V}_{h+1} for the third one, the reverse triangular inequality for the fourth one, the inductive assumption for the fifth one, and for the last one the bound on $\|Q_{h+1}^*(\cdot)\|_{L^\infty}$. Having assumed that the reward at any step is bounded in $[0, 1]$, the whole random variable

$$Y[z_h] := \tilde{r}_{h+1}[z_h] + \tilde{V}_{h+1}[z_h] - \mathbb{E}[\tilde{r}_{h+1}[z_h] + \tilde{V}_{h+1}[z_h]|z_h]$$

is bounded in $[-\sigma, \sigma]$, with $\sigma := 1 + \Lambda_d(H + 1)$, so it is at most σ -subgaussian. Before using these two results to apply standard bound for linear regression, we have to introduce a discretization of the state action space \mathcal{Z} . This is going to be characterized by the following quantities:

- k the number of points in the discretization.
- ε_k the minimum value such that we can make a ε_k -cover of \mathcal{Z} with k points.
- $\mathcal{Z}_{\varepsilon_k}$ the discretization itself.

At this point, the setting allows us to apply Proposition 12:

1. We take \mathcal{X} to be the set \mathcal{X}_φ defined in Algorithm 1, while \mathcal{X}' the subset of \mathcal{X} given by $\mathcal{X}' := \{\varphi_N(s, a) : (s, a) \in \mathcal{Z}_{\varepsilon_k}\}$. By definition, we have $|\mathcal{X}'| = k$.
2. The mean objective $\mathbb{E}[\tilde{r}_{h+1}[z_h] + \tilde{V}_{h+1}[z_h]|z_h]$ corresponds to

$$D_N * \mathcal{T}_h \widehat{Q}_{h+1}(z_h) = \varphi_N(z_h)^\top \boldsymbol{\theta}_h,$$

for some unknown $\boldsymbol{\theta}_h$, as the kernel D_N projects all the functions in the span of $\varphi_N(\cdot)$. Remark that \widehat{Q}_{h+1} is independent of the data used for regression at step h since it is computed with independent data at step $h + 1$ (this is only possible with a generative model), so $\boldsymbol{\theta}_h$ can be treated as a fixed regression target in the backward induction.

3. The noise $Y[z_h]$ is σ -subgaussian.

From Proposition 12, for the estimated $\hat{\boldsymbol{\theta}}_h$, we have

$$\mathbb{P} \left(\sup_{\mathbf{x} \in \mathcal{X}'} |\langle \boldsymbol{\theta}_h, \mathbf{x} \rangle - \langle \hat{\boldsymbol{\theta}}_h, \mathbf{x} \rangle| > \sqrt{\log(2k/\delta)} \sup_{\mathbf{x} \in \mathcal{X}'} \|\mathbf{x}\|_{V_n^{-1}} \sigma \right) \leq \delta, \quad (8)$$

and

$$\mathbb{P} \left(\sup_{\mathbf{x} \in \mathcal{X}} |\langle \boldsymbol{\theta}_h, \mathbf{x} \rangle - \langle \hat{\boldsymbol{\theta}}_h, \mathbf{x} \rangle| > \sqrt{n \log(2n/\delta)} \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_{V_n^{-1}} \sigma \right) \leq \delta. \quad (9)$$

where the matrix V_n corresponds to $V_n = \sum_{z_h \in \mathcal{D}_h} \boldsymbol{\varphi}_N(z_h) \boldsymbol{\varphi}_N(z_h)^\top$.

At this point, thanks to the fact that the dataset \mathcal{D}_h is an optimal design for \mathcal{X} , we can apply Theorem 8, which ensures

$$\sup_{\mathbf{x} \in \mathcal{X}'} \|\mathbf{x}\|_{V_n^{-1}} \leq \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_{V_n^{-1}} \leq \frac{\sqrt{2\tilde{N}}}{\sqrt{n_{\text{tot}}}}.$$

Here, note that $\sqrt{2\tilde{N}}$ comes from the optimal design, and the denominator from the fact that for each point z in the support of ρ we are taking $\lceil n_{\text{tot}} \rho(z) \rceil$ samples in Algorithm 1. Substituting into the previous equations results in

$$\mathbb{P} \left(\sup_{\mathbf{x} \in \mathcal{X}'} |\langle \boldsymbol{\theta}_h, \mathbf{x} \rangle - \langle \hat{\boldsymbol{\theta}}_h, \mathbf{x} \rangle| > \frac{\sqrt{2\tilde{N} \log(2k/\delta)} \sigma}{\sqrt{n_{\text{tot}}}} \right) \leq \delta, \quad (10)$$

and

$$\mathbb{P} \left(\sup_{\mathbf{x} \in \mathcal{X}} |\langle \boldsymbol{\theta}_h, \mathbf{x} \rangle - \langle \hat{\boldsymbol{\theta}}_h, \mathbf{x} \rangle| > \frac{\sqrt{2\tilde{N} n \log(2n/\delta)} \sigma}{\sqrt{n_{\text{tot}}}} \right) \leq \delta. \quad (11)$$

By design of the algorithm, $\hat{Q}_h(\cdot, \cdot) = \boldsymbol{\varphi}_N(\cdot, \cdot)^\top \hat{\boldsymbol{\theta}}_h$ and, by definition, $D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot, \cdot) = \boldsymbol{\varphi}_N(\cdot, \cdot)^\top \boldsymbol{\theta}_h$. Thus, the same equations write,

$$\mathbb{P} \left(\sup_{z \in \mathcal{Z}_{\varepsilon_k}} |\hat{Q}_h(z) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(z)| > \frac{\sqrt{2\tilde{N} \log(2k/\delta)} \sigma}{\sqrt{n_{\text{tot}}}} \right) \leq \delta, \quad (12)$$

and

$$\mathbb{P} \left(\|\hat{Q}_h(\cdot) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} > \frac{\sqrt{2\tilde{N} n \log(2n/\delta)} \sigma}{\sqrt{n_{\text{tot}}}} \right) \leq \delta. \quad (13)$$

Part II: Prove that \hat{Q}_h is Lipschitz continuous with a bounded constant. The two Equations (12) and (13) have very different roles in what follows. For now, we need only to have a universal

upper bound on the infinity norm of the \widehat{Q}_h function, so we forget about Equation (12) and focus on equation (13). Indeed, the latter result allows us to prove that, with probability at least $1 - \delta$,

$$\begin{aligned}
 \|\widehat{Q}_h(\cdot) - \widetilde{Q}_h(\cdot)\|_{L^\infty(\mathcal{Z})} &\leq \|\widehat{Q}_h(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} + \|\widetilde{Q}_h(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} \\
 &= \|\widehat{Q}_h(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} \\
 &\quad + \|D_N * \mathcal{T}_h \widetilde{Q}_{h+1}(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} \\
 &\leq \|\widehat{Q}_h(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} \\
 &\quad + \Lambda_d \|\widetilde{Q}_{h+1}(\cdot) - \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} \\
 &\leq \chi(\widetilde{N}, n, n_{\text{tot}}, \delta) + \Lambda_d \varepsilon \\
 &\leq \chi(\widetilde{N}, n, n_{\text{tot}}, \delta) + \Lambda_d,
 \end{aligned} \tag{14}$$

where the second passage follows by definition of \widetilde{Q}_h , the third from the bound on the L^1 norm of D_N (Equation 3) and the last from Equation (13), having defined $\chi(\widetilde{N}, n, n_{\text{tot}}, \delta) := \frac{\sqrt{2\widetilde{N}n \log(2n/\delta)}\sigma}{\sqrt{n_{\text{tot}}}}$ and by induction. Note that the previous bound is *not* tight. We will need a much better result to prove the theorem; furthermore, we have not used Equation (12) yet. Still, we can use the previous result to prove the smoothness of \widehat{Q}_h . First, notice that

$$\begin{aligned}
 \text{Vol}(\mathcal{Z}) \left\| \widehat{Q}_h(\cdot, \cdot) \right\|_{L^\infty}^2 &\geq \left\| \widehat{Q}_h(\cdot, \cdot) \right\|_{L^2}^2 \\
 &\stackrel{(Par)}{=} \left\| \sum_{i=1}^{\widetilde{N}} [\widehat{\boldsymbol{\theta}}_h]_i [\boldsymbol{\varphi}_N]_i(\cdot, \cdot) \right\|_{L^2}^2 \\
 &= \sum_{i=1}^{\widetilde{N}} [\widehat{\boldsymbol{\theta}}_h]_i^2 \left\| [\boldsymbol{\varphi}_N]_i(\cdot, \cdot) \right\|_{L^2}^2 \\
 &= \sum_{i=1}^{\widetilde{N}} [\widehat{\boldsymbol{\theta}}_h]_i^2 \\
 &= \|\widehat{\boldsymbol{\theta}}_h\|_2^2
 \end{aligned}$$

where the second passage comes from Parseval's theorem, exploiting the fact that the \widetilde{N} components of $\boldsymbol{\varphi}_N$ are all orthogonal in $L^2(\mathcal{S} \times \mathcal{A})$ and normalized to 1. The final inequality is just a standard $L^2 - L^\infty$ norm inequality. In the end this proves $\|\widehat{\boldsymbol{\theta}}_h\|_2 \leq \sqrt{\text{Vol}(\mathcal{Z})} \left\| \widehat{Q}_h(\cdot, \cdot) \right\|_{L^\infty}$. This result, combined with Equation (14), brings to

$$\|\widehat{\boldsymbol{\theta}}_h\|_2^2 \leq \left\| \widehat{Q}_h(\cdot, \cdot) \right\|_{L^2}^2 \leq \text{Vol}(\mathcal{Z}) \left\| \widehat{Q}_h(\cdot, \cdot) \right\|_{L^\infty}^2,$$

as well as the fact that \tilde{Q}_h is bounded by $\frac{(\Lambda_d C_{\mathcal{T}})^{H-h+1} - \Lambda_d C_{\mathcal{T}}}{\Lambda_d C_{\mathcal{T}} - 1}$ (theorem 9), entails that

$$\begin{aligned} \|\hat{\boldsymbol{\theta}}_h\|_2^2 &\leq \text{Vol}(\mathcal{Z}) \|\hat{Q}_h(\cdot, \cdot)\|_{L^\infty}^2 \\ &\leq \text{Vol}(\mathcal{Z}) \left(\|\hat{Q}_h(\cdot) - \tilde{Q}_h(\cdot)\|_{L^\infty} + \|\tilde{Q}_h(\cdot, \cdot)\|_{L^\infty} \right)^2 \\ &\leq \text{Vol}(\mathcal{Z}) \left(\chi(\tilde{N}, n, n_{\text{tot}}, \delta) + \Lambda_d + \frac{(\Lambda_d C_{\mathcal{T}})^{H-h+1} - \Lambda_d C_{\mathcal{T}}}{\Lambda_d C_{\mathcal{T}} - 1} \right)^2 =: \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h). \end{aligned}$$

This equation is fundamental, as it allows us to bound the $\mathcal{C}^{\nu,1}$ -norm of $\hat{Q}_h(\cdot, \cdot)$ everywhere. Indeed,

$$\begin{aligned} \|\hat{Q}_h(\cdot, \cdot)\|_{\mathcal{C}^{\nu,1}} &= \max_{|\alpha| \leq \nu+1} \|D^\alpha \hat{Q}_h(\cdot, \cdot)\|_{L^\infty} \\ &= \max_{|\alpha| \leq \nu+1} \|D^\alpha \boldsymbol{\varphi}_N(\cdot, \cdot)^\top \hat{\boldsymbol{\theta}}_h\|_{L^\infty} \\ &\stackrel{CS}{\leq} \max_{|\alpha| \leq \nu+1} \| \|D^\alpha \boldsymbol{\varphi}_N(\cdot, \cdot)\|_2 \|\hat{\boldsymbol{\theta}}_h\|_2 \|_{L^\infty} \\ &\leq \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h) \max_{|\alpha| \leq \nu+1} \| \|D^\alpha \boldsymbol{\varphi}_N(\cdot, \cdot)^\top \|_2 \|_{L^\infty} \\ &\leq \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h) \sqrt{\tilde{N}} \max_{|\alpha| \leq \nu+1} \| \|D^\alpha \boldsymbol{\varphi}_N(\cdot, \cdot)\|_\infty \|_{L^\infty} \\ &\leq N^{\nu+1} \sqrt{\tilde{N}} \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h). \end{aligned} \tag{15}$$

Where the third passage is by the Cauchy-Schwartz inequality, the fourth one from the bound on $\|\hat{\boldsymbol{\theta}}_h\|_2$ and the last one from the fact that the derivatives of order $|\alpha| \leq \nu + 1$ of the Fourier features of degree N do not exceed $N^{\nu+1}$ (by just computing the derivatives, $\sin(Nx) \rightarrow N \cos(Nx) \rightarrow -N^2 \sin(Nx), \dots$).

Part III: Extend equation (13) to all \mathcal{Z} through the Lipschitzness of \hat{Q}_h .

At this point, we have to fix the previously mentioned cover $\mathcal{Z}_{\varepsilon_k}$ of \mathcal{Z} . It is well-known (Wu, 2017) that to construct an ε_k -cover of $\mathcal{Z} = [-1, 1]^d$ we need $k = 3^d \varepsilon_k^{-d}$ points. In our case, we are interested in a cover that contains a near-maximum point for the function $\hat{Q}_h(\cdot) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)$, therefore the value of ε must be determined based on the Lipschitz constant of the latter function. We have proved in Equation 15 that $\|\hat{Q}_h(\cdot, \cdot)\|_{\mathcal{C}^{\nu,1}} \leq N^{\nu+1} \sqrt{\tilde{N}} \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h)$, so

$$\begin{aligned} \|\hat{Q}_h(\cdot) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{\mathcal{C}^{\nu,1}} &\leq \|\hat{Q}_h(\cdot)\|_{\mathcal{C}^{\nu,1}} + \|D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{\mathcal{C}^{\nu,1}} \\ &\leq \|\hat{Q}_h(\cdot)\|_{\mathcal{C}^{\nu,1}} + \Lambda_d \|\mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{\mathcal{C}^{\nu,1}} \\ &\leq \|\hat{Q}_h(\cdot)\|_{\mathcal{C}^{\nu,1}} + \Lambda_d C_{\mathcal{T}} (\|\hat{Q}_{h+1}(\cdot)\|_{\mathcal{C}^{\nu,1}} + 1) \\ &\leq (1 + \Lambda_d C_{\mathcal{T}}) \sqrt{\tilde{N}} \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h) + \Lambda_d C_{\mathcal{T}} \\ &\leq (1 + 2\Lambda_d C_{\mathcal{T}}) \sqrt{\tilde{N}} \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h). \end{aligned}$$

where the second passage follows by Equation (6) and the third by definition of $C_{\mathcal{T}}$, while the last one follows from Equation (15). As the previous display bounds, by definition, the Lipschitz

constant of the same function, we can just take

$$L_{\text{Bell}} := (1 + 2\Lambda_d C_{\mathcal{T}}) N^{\nu+1} \sqrt{\tilde{N}} \chi_2(\tilde{N}, n, n_{\text{tot}}, \delta, h),$$

and say that the Bellmann error $\widehat{Q}_h(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)$ is Lipschitz continuous with this constant. Therefore, if we want to bound its variation with ε , we need a $\frac{\varepsilon}{L_{\text{Bell}}}$ -cover, which requires

$$k = \left(\frac{3L_{\text{Bell}}}{\varepsilon} \right)^d$$

points. In particular, we want to fix

$$\varepsilon = \frac{\sqrt{2\tilde{N} \log(2k/\delta)} \sigma}{\sqrt{n_{\text{tot}}}},$$

so the number of points required is

$$k = \left(\frac{3L_{\text{Bell}} \sqrt{n_{\text{tot}}}}{\sqrt{2\tilde{N} \log(2k/\delta)} \sigma} \right)^d.$$

With this choice of k , we have, with probability $1 - 2H\delta$ (the coefficient $2H$ is from a union bound, since both Equation (12) and (13) have to hold for every h),

$$\begin{aligned} \|\widehat{Q}_h(\cdot) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(\cdot)\|_{L^\infty(\mathcal{Z})} &\leq \sup_{z \in \mathcal{Z}_{\varepsilon_k}} |\widehat{Q}_h(z) - D_N * \mathcal{T}_h \widehat{Q}_{h+1}(z)| + \varepsilon \\ &\stackrel{(12)}{\leq} \frac{\sqrt{2\tilde{N} \log(2k/\delta)} \sigma}{\sqrt{n_{\text{tot}}}} + \varepsilon \\ &= \frac{\sqrt{8\tilde{N} \log(2k/\delta)} \sigma}{\sqrt{n_{\text{tot}}}}, \end{aligned} \tag{16}$$

where the first passage comes from the definition of the cover, the second from Equation (12), the last one by definition of ε . We have been finally able to bound this pseudo-Bellmann error for all $z \in \mathcal{Z}$. Even if k , the number of points in the cover, depends on $\tilde{N}, n, n_{\text{tot}}$, as its dependence takes the form of $\log(k)$, its presence does not affect the sample complexity significantly.

Part IV: From the Bellmann error to the Q-function error

At this point, it is not difficult to bound by induction the difference between the estimated Q-function \widehat{Q}_h and the target \widetilde{Q}_h . Indeed,

$$\begin{aligned}
 \|\tilde{Q}_h(\cdot) - \hat{Q}_h(\cdot)\|_{L^\infty} &\leq \|\tilde{Q}_h(\cdot) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{L^\infty} + \|\hat{Q}_h(\cdot) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{L^\infty} \\
 &\stackrel{(16)}{\leq} \|D_N * \mathcal{T}_h \tilde{Q}_{h+1}(\cdot) - D_N * \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{L^\infty} + \frac{\sqrt{8\tilde{N} \log(2k/\delta)}\sigma}{\sqrt{n_{\text{tot}}}} \\
 &\leq \Lambda_d \|\mathcal{T}_h \tilde{Q}_{h+1}(\cdot) - \mathcal{T}_h \hat{Q}_{h+1}(\cdot)\|_{L^\infty} + \frac{\sqrt{8\tilde{N} \log(2k/\delta)}\sigma}{\sqrt{n_{\text{tot}}}} \\
 &\leq \Lambda_d \|\tilde{Q}_{h+1}(\cdot) - \hat{Q}_{h+1}(\cdot)\|_{L^\infty} + \frac{\sqrt{8\tilde{N} \log(2k/\delta)}\sigma}{\sqrt{n_{\text{tot}}}} \\
 &\leq \Lambda_d \varepsilon_{h+1} + \frac{\sqrt{8\tilde{N} \log(2k/\delta)}\sigma}{\sqrt{n_{\text{tot}}}} \tag{17}
 \end{aligned}$$

the second inequality being valid by Equation (16), the third one by the usual bound on the L^1 norm of D_N , the fourth one by the non-expansivity of the Bellmann operator and the last one by the inductive hypothesis. We have just proved that, with probability at least $1 - 2\delta$, the estimation error is bounded as

$$\|\tilde{Q}_h(\cdot) - \hat{Q}_h(\cdot)\|_{L^\infty} \leq \Lambda_d \varepsilon_{h+1} + \frac{\sqrt{8\tilde{N} \log(2k/\delta)}\sigma}{\sqrt{n_{\text{tot}}}}. \tag{18}$$

This term can be naturally coupled with the approximation term between \tilde{Q}_h and Q^* , which is bounded, by theorem 9 with

$$\|\tilde{Q}_h - Q_h^*\|_{L^\infty} \leq \Psi_h N^{-\nu-1}.$$

At this point, we set N and n_{tot} in this way (note that k contains terms that are polynomial in N , n_{tot} but, appearing only in the logarithm, this has a negligible effect):

$$N = \left\lceil \left(\frac{\Psi_h}{\Lambda_d \varepsilon_{h+1}} \right)^{\frac{1}{\nu+1}} \right\rceil \quad n_{\text{tot}} = \left\lceil \left(\frac{\sigma \sqrt{8\tilde{N} \log(2k/\delta)}}{\Lambda_d \varepsilon_{h+1}} \right)^2 \right\rceil,$$

so that $\|\tilde{Q}_h - \hat{Q}_h\|_{L^\infty} \leq 2\Lambda_d \varepsilon_{h+1} \leq \varepsilon_h$, and by triangular inequality

$$\|\hat{Q}_h - Q_h^*\|_{L^\infty} \leq \Lambda_d \varepsilon_{h+1} + \|\tilde{Q}_h - \hat{Q}_h\|_{L^\infty} \leq 3\Lambda_d \varepsilon_{h+1} = \varepsilon_h,$$

as needed to complete the inductive step. This has been done with a number of samples that is at most (proposition 3)

$$2(n_{\text{tot}} + 4\tilde{N} \log \log(\tilde{N})) = \tilde{\mathcal{O}}(n_{\text{tot}}) = \tilde{\mathcal{O}} \left(\sigma^2 \Psi_h^{\frac{d}{2(\nu+1)}} \varepsilon_h^{-\frac{d+2(\nu+1)}{\nu+1}} \right),$$

which, in terms of the original ε , corresponds to

$$\tilde{\mathcal{O}}(n_{\text{tot}}) = \tilde{\mathcal{O}} \left(\sigma^2 \Psi_h^{\frac{d}{2(\nu+1)}} (3\Lambda_d)^h \varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}} \right).$$

Note that we have chosen N depending on h , which should not in the definition of our algorithm. Still, for the algorithm we can just take the highest value of N among every step h , and the corresponding n_{tot} . As the dependence on ε is always of $\varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}}$, this does not compromise the result. ■

Now, we can finally proceed to the proof of the main result.

Theorem 4 Fix $\varepsilon > 0, \delta > 0$. Apply Algorithm 1 on a ν -Smooth MDP. Then, for a proper choice of $N = \mathcal{O}(\varepsilon^{-\frac{1}{\nu+1}})$, with probability at least $1 - \delta$, we are able to learn a policy $\hat{\pi}$ that is ε -optimal with a number of samples given by

$$n = \tilde{\mathcal{O}} \left(K(H, d) \varepsilon^{-\frac{d+2(\nu+1)}{\nu+1}} \right).$$

Proof We use theorem 10 with $\varepsilon \leftarrow \varepsilon/2$, which ensures that under the same probability we are able to learn a function \hat{Q}_h such that $\|Q_h^* - \hat{Q}_h\|_{L^\infty} < (1/2)\varepsilon/(3\Lambda_d)^h$. Since the policy outputted by algorithm 1 is just greedy with respect to \hat{Q} , we have

$$\begin{aligned} V_h^*(s) - V_h^{\hat{\pi}}(s) &= Q_h^*(s, \pi^*(s)) - Q_h^{\hat{\pi}}(s, \hat{\pi}(s)) \\ &= \underbrace{Q_h^*(s, \pi^*(s)) - Q_h^*(s, \hat{\pi}(s))}_I + \underbrace{Q_h^*(s, \hat{\pi}(s)) - Q_h^{\hat{\pi}}(s, \hat{\pi}(s))}_{II}. \end{aligned}$$

The first part can be easily bounded in this way:

$$\begin{aligned} I &\leq \|Q_h^*(\cdot, \cdot) - \hat{Q}_h(\cdot, \cdot)\|_{L^\infty} + \hat{Q}_h(s, \pi^*(s)) - \hat{Q}_h(s, \hat{\pi}(s)) \\ &\leq \|Q_h(\cdot, \cdot) - \hat{Q}_h(\cdot, \cdot)\|_{L^\infty}, \end{aligned}$$

as the second term is negative by definition of $\hat{\pi}$. Meanwhile, the second one is bounded by induction:

$$\begin{aligned} II &\leq \mathbb{E}_{s' \sim p_h(\cdot | s, \hat{\pi}(s))} [V_{h+1}^*(s') - V_{h+1}^{\hat{\pi}}(s')] \\ &\leq \|V_{h+1}^*(\cdot) - V_{h+1}^{\hat{\pi}}(\cdot)\|_{L^\infty}. \end{aligned}$$

In this way, we have proved that $\|V_h^*(\cdot) - V_h^{\hat{\pi}}(\cdot)\|_{L^\infty} \leq \|Q_h(\cdot, \cdot) - \hat{Q}_h(\cdot, \cdot)\|_{L^\infty} + \|V_{h+1}^*(\cdot) - V_{h+1}^{\hat{\pi}}(\cdot)\|_{L^\infty}$. This implies that

$$\|V_h^*(\cdot) - V_h^{\hat{\pi}}(\cdot)\|_{L^\infty} \leq \sum_{h'=h}^H \|Q_{h'}^*(\cdot, \cdot) - \hat{Q}_{h'}(\cdot, \cdot)\|_{L^\infty} \leq \frac{1}{2} \sum_{h'=h}^H \varepsilon / (3\Lambda_d)^{h'}. \quad (19)$$

In particular, for $h = 1$ we get $\|V_1^*(\cdot) - V_1^{\hat{\pi}}(\cdot)\|_{L^\infty} \leq \frac{1}{2} \frac{\varepsilon}{1-3\Lambda_d} \leq \varepsilon$. This corresponds to the definition of ε -optimal policy. ■

Appendix D. Proofs from section 4.2

In this section we prove that every sample complexity bound for algorithms in the Lipschitz MDP setting must grow exponentially with the time horizon H . Lipschitz MDPs assume that the transition function of the model, as well as the reward function, are Lipschitz continuous with constants L_p and L_r , respectively. Mathematically, the condition on the reward function writes as, for every $h \in [H]$, $s, s' \in \mathcal{S}$, $a, a' \in \mathcal{A}$:

$$|R_h(s, a) - R_h(s', a')| \leq L_r(\|s - s'\|_2 + \|a - a'\|_2).$$

While for the transition function, which maps a state-action pair into a probability distribution, we bound its difference in the Wasserstein metric $\mathcal{W}(\cdot, \cdot)$ (Rachelson and Lagoudakis, 2010):

$$\mathcal{W}(p_h(\cdot|s, a), p_h(\cdot|s', a')) \leq L_p(\|s - s'\|_2 + \|a - a'\|_2),$$

The Wasserstein metric is a notion of distance for probability measures on metric spaces. For two measures μ, ζ on a metric space Ω , it is defined as $\mathcal{W}(\mu, \zeta) := \sup_{f \in \mathcal{C}^{0,1}; \|f\|_{\mathcal{C}^{0,1}}=1} \int_{\Omega} f(\omega) d(\mu - \zeta)(\omega)$.

The proof strategy is the following: we start from an instance of a Lipschitz bandit problem with a Lipschitz constant that is exponential in H , and show that this can be reduced to a standard Lipschitz MDP (where all Lipschitz constants are independent on H). Since it has been shown that the sample complexity in a Lipschitz bandit problem is proportional to the Lipschitz constant, this shows that the regret of the Lipschitz MDP is also exponential in H .

Theorem 11 *The minimum number of samples necessary to learn an ε -optimal policy, with probability at least $1 - \delta$, in a (L_p, L_r) -Lipschitz MDP satisfies $n = \Omega(L_p^{d(H-2)} \varepsilon^{-d-2})$.*

Proof Let $f : [-1, 1]^d \rightarrow [-1, 1]$ be an $2L_p^{H-2}$ -Lipschitz function and η a noise bounded in $[-1, 1]$.

Define $\tilde{f} := (L_p^{-H+2}/2)f$, $\tilde{\eta} := (L_p^{-H+2}/2)\eta$, so that $\tilde{f} : [-1, 1]^d \rightarrow [-L_p^{-H+2}/2, L_p^{-H+2}/2]$ is a $1/2$ -Lipschitz function and $\tilde{\eta}$ a noise bounded in $[-L_p^{-H+2}/2, L_p^{-H+2}/2]$. Define the following MDP:

- The state and action space coincide: $\mathcal{S} = \mathcal{A} = [-1, 1]^{d/2}$. In this way, $\mathcal{S} \times \mathcal{A} = [-1, 1]^d$
- The starting state is $[0, \dots, 0]$ almost surely.
- The transition function is defined in the following way:
 - For $h = 1$, $p_1(s'|s, a) = \delta(s' = a)$, so that the first action becomes the second state.
 - For $h = 2$, $p_2(s'|s, a) = \delta(s'^{(1)} = \tilde{f}(s, a) + \tilde{\eta}) \prod_{i=2}^{d/2} \delta_0(s'^{(i)})$, meaning that the next state has the first coordinate equal to $\tilde{f}(s, a)$ plus the noise η , and all the other ones set to zero. Note that this is coherent with the definition of f , which goes $[-1, 1]^{d/2} = \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$.
 - For $h = 3, \dots, H$ we have $p_h(s'|s, a) = \delta(s' = L_p s)$, so that the next state is the previous one times a constant (note that, by the bounds on $\tilde{f}, \tilde{\eta}$, the state never hits the boundary).

- The reward function r_h is zero for the first $H - 1$ time steps, and $r_H(s, a) = s^{(1)}$, the first component of the state.

By definition, it is easy to check that the MDP is Lipschitz with $L_p = L_p$ and $L_r = 1$.

In this very peculiar MDP, where only the first two actions a_1 and a_2 matter, the return can be expressed as a function of them. Precisely, since the reward is only given at the last time step, we have

$$\text{Return}(a_1, a_2) = L_p^{H-2}(\tilde{f}(a_1, a_2) + \tilde{\eta}) = \frac{1}{2}(f(a_1, a_2) + \eta).$$

In this way, we have shown that the return for this Lipschitz MDP corresponds exactly to the feedback in the d -dimensional Lipschitz bandit problem with reward function $f/2$ (which is L_p^{H-2} -LC) and noise $\eta/2$.

This shows that any Lipschitz bandit problem with Lipschitz constant L_p^{H-2} can be reduced to a Lipschitz MDP with constants bounded independently of H . Therefore, the sample complexity on the latter problem is at most as high as the one of the former one. As the sample complexity of the latter problem is well-known to be of order $\Omega(L^d \varepsilon^{-2-d}) = \Omega(L_p^{(H-2)d} \varepsilon^{-2-d})$, the proof is complete. \blacksquare

Appendix E. Auxiliary Results

In the main proof, we need this bound for regression, which is a variant of the classical results for ordinary least squares with fixed design.

Proposition 12 *Assume to have a dataset of the form $\{\mathbf{x}_t, y_t\}_{t=1}^n$, where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \mathbb{R}$, with $y_t = \mathbf{x}_t^\top \boldsymbol{\theta} + \xi_t$, for some unknown $\boldsymbol{\theta} \in \mathbb{R}^{\tilde{N}}$, and a random noise $\{\xi_t\}_{t=1}^n$ with independent σ -subgaussian components and independent from $\{\mathbf{x}_t\}_{t=1}^n$. Suppose that \mathbf{x}_t for $t = 1, \dots, n$ span the whole space $\mathbb{R}^{\tilde{N}}$ and define the LS estimator*

$$\hat{\boldsymbol{\theta}} := \left(\sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^\top \right)^{-1} \sum_{t=1}^n \mathbf{x}_t y_t.$$

Then, for every $\delta > 0$, fixing an arbitrary subset $\mathcal{X}' \subset \mathcal{X}$ with $|\mathcal{X}'| = k$ we have

- $\mathbb{P} \left(\sup_{\mathbf{x} \in \mathcal{X}'} |\langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \hat{\boldsymbol{\theta}}, \mathbf{x} \rangle| > \sqrt{\log(2k/\delta)} \sup_{\mathbf{x} \in \mathcal{X}'} \|\mathbf{x}\|_{V_n^{-1}} \sigma \right) \leq \delta.$
- $\mathbb{P}(\sup_{\mathbf{x} \in \mathcal{X}} |\langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \hat{\boldsymbol{\theta}}, \mathbf{x} \rangle| > \sqrt{n \log(2n/\delta)} \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_{V_n^{-1}} \sigma) \leq \delta.$

Proof By simplicity, let us call

$$V_n = \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^\top.$$

Let $\mathbf{x} \in \mathcal{X}$. We have

$$\begin{aligned}
\langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \widehat{\boldsymbol{\theta}}, \mathbf{x} \rangle &= \langle \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}, \mathbf{x} \rangle \\
&= \left\langle V_n^{-1} V_n \boldsymbol{\theta} - V_n^{-1} \sum_{t=1}^n \mathbf{x}_t y_t, \mathbf{x} \right\rangle \\
&= \left\langle V_n^{-1} \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^\top \boldsymbol{\theta} - V_n^{-1} \sum_{t=1}^n \mathbf{x}_t (\mathbf{x}_t^\top \boldsymbol{\theta} + \xi_t), \mathbf{x} \right\rangle \\
&= \left\langle -V_n^{-1} \sum_{t=1}^n \mathbf{x}_t \xi_t, \mathbf{x} \right\rangle \\
&= - \sum_{t=1}^n \langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle \xi_t.
\end{aligned}$$

Being $\{\xi_t\}_t$ a sequence of σ -subgaussian independent random variables, the sum is also subgaussian conditionally on the data points with a constant given by

$$\sigma_n(\mathbf{x}) = \sigma \sqrt{\sum_{t=1}^n \langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle^2}.$$

We can bound the last quantity in the following way

$$\begin{aligned}
\frac{\sigma_n(\mathbf{x})^2}{\sigma^2} &= \sum_{t=1}^n \langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle^2 \\
&= \text{Tr} \left(\sum_{t=1}^n \langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle^2 \right) \\
&= \sum_{t=1}^n \text{Tr} \left(\langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle^2 \right) \\
&= \sum_{t=1}^n \text{Tr} \left(\mathbf{x}^\top V_n^{-1} \mathbf{x}_t \mathbf{x}_t^\top V_n^{-1} \mathbf{x} \right) \\
&= \text{Tr} \left(\mathbf{x}^\top V_n^{-1} \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^\top V_n^{-1} \mathbf{x} \right) \\
&= \text{Tr} \left(\mathbf{x}^\top V_n^{-1} \mathbf{x} \right) = \|\mathbf{x}\|_{V_n^{-1}}^2.
\end{aligned}$$

This entails that $\sigma_n(\mathbf{x}) = \|\mathbf{x}\|_{V_n^{-1}} \sigma$. Now, fixing any $\delta > 0$ and applying Hoeffding's inequality, we have

$$\mathbb{P}(\langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \widehat{\boldsymbol{\theta}}, \mathbf{x} \rangle > \sqrt{\log(1/\delta)} \|\mathbf{x}\|_{V_n^{-1}} \sigma) \leq \delta.$$

At this point, by a union bound over \mathcal{X}' , we have the first part of the statement:

$$\mathbb{P}(\sup_{\mathbf{x} \in \mathcal{X}'} \langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \widehat{\boldsymbol{\theta}}, \mathbf{x} \rangle > \sqrt{\log(k/\delta)} \sup_{\mathbf{x} \in \mathcal{X}'} \|\mathbf{x}\|_{V_n^{-1}} \sigma) \leq \delta.$$

For the second statement, we have to proceed in a different way. Indeed, by the same argument as before, we have

$$\begin{aligned}
 \sup_{\mathbf{x} \in \mathcal{X}} \langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \widehat{\boldsymbol{\theta}}, \mathbf{x} \rangle &\leq - \sum_{t=1}^n \langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle \xi_t \\
 &\stackrel{CS}{\leq} \sqrt{\sum_{t=1}^n \langle V_n^{-1} \mathbf{x}_t, \mathbf{x} \rangle^2} \sqrt{\sum_{t=1}^n \xi_t^2} \\
 &\leq \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_{V_n^{-1}} \sqrt{\sum_{t=1}^n \xi_t^2}.
 \end{aligned}$$

Here, the second inequality is due to the Cauchy-Schwartz inequality, and the third one to the same inequality that we have used to prove the first statement. At this point note that

$$\sqrt{\sum_{t=1}^n \xi_t^2} \leq \sqrt{n} \max_t |\xi_t|,$$

which, being the maximum of a sequence of σ -subgaussian random variables, is bounded by $\sqrt{n} \sqrt{\log(2n/\delta)} \sigma$ with probability $1 - \delta$. This entails that

$$\mathbb{P}(\sup_{\mathbf{x} \in \mathcal{X}} |\langle \boldsymbol{\theta}, \mathbf{x} \rangle - \langle \widehat{\boldsymbol{\theta}}, \mathbf{x} \rangle| > \sqrt{n \log(2n/\delta)} \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_{V_n^{-1}} \sigma) \leq \delta,$$

completing the second part of the proof. ■

Appendix F. Computational complexity

The computational complexity of our algorithm can be divided into the one for the preliminary operations and the one for learning from the samples (after line 10). Algorithmically, line 1 is performed by fixing ε' and constructing an ε' -grid of the state-action space, which amounts to $k = (2/\varepsilon')^d$ points. As the matrix norm is continuous, this introduces an error that can be made arbitrarily small with ε' . Once this is done, the optimal design (line 2) can be computed in just $k\tilde{N}^2$ steps (see [Lattimore et al. \(2020\)](#) just before theorem 4.4). The learning part requires to solve a linear regression problem, which is well known to take $n\tilde{N}^2 + \tilde{N}^3$ steps, where n is the number of samples. Therefore, the total computational complexity is roughly of order $k\tilde{N}^2 + n\tilde{N}^2 + \tilde{N}^3$. Even if k is exponential in d , note that this term is dominated by the number of samples n , which is exponential in dH . Note that n (i.e., the sample complexity) must be exponential in dH for any algorithm, as the lower bound (theorem 6) shows. Therefore, our computational complexity grows as $n\tilde{N}^2$. In contrast, note that other algorithms for other very general RL settings often require access to an optimization oracle [Jin et al. \(2020\)](#); [Du et al. \(2021\)](#). Even in the much more restrictive case of low-rank MDPs, the state of the art was computationally efficient [Uehara et al. \(2021\)](#) until very recently, when an efficient counterpart [Mhammedi et al. \(2024\)](#) was proved to work.