FLAMBE: Structural Complexity and Representation Learning of Low Rank MDPs

Alekh Agarwal
Microsoft Research, Redmond
alekha@microsoft.com

Sham Kakade
Microsoft Research, NYC
sham@cs.washington.edu

Akshay Krishnamurthy
Microsoft Research, NYC
akshaykr@microsoft.com

Wen Sun
Cornell University
ws455@cornell.edu

Abstract

In order to deal with the curse of dimensionality in reinforcement learning (RL), it is common practice to make parametric assumptions where values or policies are functions of some low dimensional feature space. This work focuses on the representation learning question: how can we *learn* such features? Under the assumption that the underlying (unknown) dynamics correspond to a low rank transition matrix, we show how the representation learning question is related to a particular *non-linear* matrix decomposition problem. Structurally, we make precise connections between these low rank MDPs and latent variable models, showing how they significantly generalize prior formulations, such as block MDPs, for representation learning in RL. Algorithmically, we develop FLAMBE, which engages in exploration and representation learning for provably efficient RL in low rank transition models. On a technical level, our analysis eliminates reachability assumptions that appear in prior results on the simpler block MDP model and may be of independent interest.

1 Introduction

The ability to learn effective transformations of complex data sources, sometimes called representation learning, is an essential primitive in modern machine learning. Advances in this area have led to remarkable achievements in language modeling, vision, and serve as a partial explanation for the success of deep learning more broadly (Bengio et al., 2013). In Reinforcement Learning (RL), several works have shown empirically that learning succinct representations of perceptual inputs can accelerate the search for decision-making policies (Bellemare et al., 2016; Pathak et al., 2017; Tang et al., 2017; Oord et al., 2018; Srinivas et al., 2020). However, representation learning for RL is far more subtle than it is for supervised learning (Du et al., 2019a; Van Roy and Dong, 2019; Lattimore and Szepesvari, 2020), and the theoretical foundations of representation learning for RL are nascent.

The first question that arises in this context is: what is a good representation? Intuitively, a good representation should help us achieve greater sample efficiency on downstream tasks. For supervised learning, several theoretical works adopt the perspective that a good representation should permit simple models to achieve high accuracy on tasks of interest (Baxter, 2000; Maurer et al., 2016; Arora et al., 2019; Tosh et al., 2020). Lifting this perspective to reinforcement learning, it is natural to expect that we can express value functions and policies as simple functions of our representation. This may allow us to leverage recent work on sample efficient RL with parametric function approximation.

The second question is: how do we *learn* such a representation when it is not provided in advance? This question is particularly challenging because representation learning is intimately tied to explo-

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Algorithm	Setting	Sample Complexity	Computation
PCID (Du et al., 2019b)	block MDP	$d^4H^2K^4\left(\frac{1}{\eta^4\gamma^2}+\frac{1}{\varepsilon^2}\right)$	Oracle efficient
HOMER (Misra et al., 2020)	block MDP	$d^8H^4K^4\left(\frac{1}{\eta^3}+\frac{1}{\varepsilon^2}\right)$	Oracle efficient
OLIVE (Jiang et al., 2017)	low Bellman rank	$\frac{d^2H^3K}{\varepsilon^2}$	Inefficient
Sun et al. (2019)	low Witness rank	$\frac{d^2H^3K}{\varepsilon^2}$	Inefficient
FLAMBE (this paper)	low rank MDP	$\frac{d^7K^9H^{22}}{\varepsilon^{10}}$	Oracle efficient

Table 1: Comparison of methods for representation learning in RL. Settings from least to most general are: block MDP, low rank MDP, low Bellman rank, low Witness rank. In all cases d is the embedding dimension, H is the horizon, K is the number of actions, η and γ parameterize reachability and margin assumptions, and ε is the accuracy. Dependence on function classes and logarithmic factors are suppressed. Oracle and realizability assumptions vary. Block MDP algorithms discover a *one-hot* representation to discrete latent states. Bellman/Witness rank approaches can take a class Φ of embedding functions and search over simple policies or value functions composed with Φ (see Section 4 and Appendix A.3 for details).

ration. We cannot learn a good representation without a comprehensive dataset of experience from the environment, but a good representation may be critical for efficient exploration.

This work considers these questions in the context of low rank MDPs (Jiang et al., 2017) (also known as factorizing MDPs (Rendle et al., 2010), factored linear MDPs (Yao et al., 2014), and linear MDPs (Jin et al., 2020b; Yang and Wang, 2020)), which we argue provide a natural framework for studying representation learning in RL. Concretely, these models assume there exists low dimensional embedding functions $\phi(x, a), \mu(x')$ such that the transition operator T satisfies $T(x' \mid x, a) = \langle \phi(x, a), \mu(x') \rangle$, where $T(x' \mid x, a)$ specifies the probability of the next state x' given the previous state x and action a. Low rank MDPs address the first issue above (on what constitutes a good representation) in that if the features ϕ are known to the learner, then sample efficient learning is possible (Jin et al., 2020b; Yang and Wang, 2020).

Our contributions. We address the question of learning the representation ϕ in a low rank MDP. To this end our contributions are both structural and algorithmic.

1. **Expressiveness of low rank MDPs.** We first provide a re-formulation of the low rank dynamics in terms of an equally expressive, but more interpretable latent variable model. We provide several structural results for low rank MDPs, relating it to other models studied in prior work on representation learning for RL. In particular, we show that low rank MDPs are significantly more expressive than the block MDP model (Du et al., 2019b; Misra et al., 2020).
2. **Feature learning.** We develop a new algorithm, called FLAMBE for "Feature learning and model based exploration", that learns a representation for low rank MDPs. We prove that under realizability assumptions, FLAMBE learns a *uniformly accurate* model of the environment as well as a feature map that enables the use of linear methods for RL, in a statistically and computationally efficient manner. These guarantees enable downstream reward maximization, for *any* reward function, with no additional data collection.

Our results and techniques provide new insights on representation learning for RL and also significantly increase the scope for provably efficient RL with rich observations (see Table 1).

2 Low Rank MDPs

We consider an episodic Markov decision process $\mathcal{M}$ with episode length $H \in \mathbb{N}$, state space $\mathcal{X}$, and a finite action space $\mathcal{A} = \{1, \ldots, K\}$. In each episode, a trajectory $\tau = (x_0, a_0, x_1, a_1, \ldots, x_{H-1}, a_{H-1}, x_H)$ is generated, where (a) x_0 is a starting state, and (b) $x_{h+1} \sim T_h(\cdot \mid x_h, a_h)$, and (c) all actions $a_{0:H-1}$ are chosen by the agent. We assume the starting state is fixed and that there is only one available action at time 0.[1] The operators $T_h : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ denote the (non-stationary) transition dynamics for each time step.

[1] This easily accommodates the standard formulation with a non-degenerate initial distribution by defining $T_0(\cdot \mid x_0, a_0)$ to be the initial distribution. This setup is notationally more convenient, since we do not need special notation for the starting distribution.

As is standard in the literature, a policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ is a (randomized) mapping from states to actions. We use the notation $\mathbb{E}[\cdot \mid \pi, \mathcal{M}]$ to denote expectations over states and actions observed when executing policy π in MDP $\mathcal{M}$. We abuse notation slightly and use $[H]$ to denote $\{0, \ldots, H-1\}$.

Definition 1. *An operator* $T : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ *admits a* low rank decomposition *with dimension* $d \in \mathbb{N}$ *if there exist two embedding functions* $\phi^\star : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ *and* $\mu^\star : \mathcal{X} \to \mathbb{R}^d$ *such that*

$$\forall x, x' \in \mathcal{X}, a \in \mathcal{A} : T(x' \mid x, a) = \langle \phi^\star(x,a), \mu^\star(x') \rangle .$$

For normalization,[2] *we assume that* $\|\phi^\star(x,a)\|_2 \leq 1$ *for all* x, a *and for any function* $g : \mathcal{X} \to [0,1]$, $\left\| \int \mu^\star(x) g(x) dx \right\|_2 \leq \sqrt{d}$*. An MDP* $\mathcal{M}$ *is a* low rank MDP *if for each* $h \in [H]$*,* T_h *admits a low rank decomposition with dimension* d*. We use* $\phi^\star_h, \mu^\star_h$ *to denote the embeddings for* T_h*.*

Throughout we assume that $\mathcal{M}$ is a low rank MDP with dimension d. Note that the normalization condition on $\mu^\star$ ensures that the Bellman backup operator is well-behaved.

Function approximation for representation learning. We consider state spaces $\mathcal{X}$ that are arbitrarily large, so that some form of function approximation is necessary to generalize across states. For representation learning, it is natural to grant the agent access to two function classes $\Phi \subset \{\mathcal{X} \times \mathcal{A} \to \mathbb{R}^d\}$ and $\Upsilon \subset \{\mathcal{X} \to \mathbb{R}^d\}$ of candidate embeddings, which we can use to identify the true embeddings $(\phi^\star, \mu^\star)$. To facilitate this model selection task, we posit a *realizability* assumption.

Assumption 1 (Realizability)**.** *We assume that for each* $h \in [H]$*:* $\phi^\star_h \in \Phi$ *and* $\mu^\star_h \in \Upsilon$*.*

We desire sample complexity bounds that scale logarithmically with the cardinality of the classes Φ and Υ, which we assume to be finite. Extensions that permit infinite classes with bounded statistical complexity (e.g., VC-classes) are not difficult.

In Appendix A, we show that the low rank assumption alone, without Assumption 1, is not sufficient for obtaining performance guarantees that are independent of the size of the state space. Hence, additional modeling assumptions are required, and we encode these in Φ, Υ.

Learning goal. We focus on the problem of reward-free exploration (Hazan et al., 2019; Jin et al., 2020a), where the agent interacts with the environment with no reward signal. When considering model-based algorithms, a natural reward-free goal is *system identification*: given function classes Φ, Υ, the algorithm should learn a model $\widehat{\mathcal{M}} := (\widehat{\phi}_{0:H-1}, \widehat{\mu}_{0:H-1})$ that uniformly approximates the environment $\mathcal{M}$. We formalize this with the following performance criteria:

$$\forall \pi, h \in [H] : \mathbb{E}\left[\left\| \left\langle \widehat{\phi}_h(x_h, a_h), \widehat{\mu}_h(\cdot) \right\rangle - T_h(\cdot \mid x_h, a_h) \right\|_{\mathrm{TV}} \mid \pi, \mathcal{M} \right] \leq \varepsilon. \tag{1}$$

Here, we ask that our model accurately approximates the one-step dynamics from the state-action distribution induced by following *any* policy π for h steps in the real environment.

System identification also implies a quantitative guarantee on the learned representation $\widehat{\phi}_{0:H-1}$: we can approximate the Bellman backup of any value function on any data-distribution.

Lemma 1. *If* $\widehat{\mathcal{M}} = (\widehat{\phi}_{0:H-1}, \widehat{\mu}_{0:H-1})$ *satisfies* (1)*, then*

$$\forall h \in [H], V : \mathcal{X} \to [0,1], \exists \theta : \max_\pi \mathbb{E}\left[\left| \langle \theta, \widehat{\phi}_h(x_h, a_h) \rangle - \mathbb{E}\left[V(x_{h+1}) \mid x_h, a_h\right] \right| \mid \pi, \mathcal{M} \right] \leq \varepsilon.$$

Thus, linear function approximation using our learned features suffices to fit the Q function associated with any policy and explicitly given reward.[3] The guarantee also enables dynamic programming techniques for policy optimization. In other words, (1) verifies that we have found a good representation, in a quantitative sense, and enables tractable reward maximization for any known reward function.

3 Related work

Low rank models are prevalent in dynamics and controls (Thon and Jaeger, 2015; Littman and Sutton, 2002; Singh et al., 2004). The low rank MDP in particular has been studied in several works in

[2] See the proof of Lemma B.1 in Jin et al. (2020b) for this form of the normalization assumption.

[3] Formally, we append the immediate reward to the learned features $\widehat{\phi}_{0:H-1}$.

the context of planning (Barreto et al., 2011; Barreto and Fragoso, 2011), estimation (Duan et al., 2020), and in the generative model setting (Yang and Wang, 2019). Regarding nomenclature, to our knowledge the name *low rank MDP* appears first in Jiang et al. (2017), although Rendle et al. (2010) refer to it as *factorizing MDP*, Yao et al. (2014) call it a factored linear MDP, and Barreto et al. (2011) refer to a similar model as *stochastic factorization*. More recently, it has been called the *linear MDP* by Jin et al. (2020b). We use *low rank MDP* because it highlights the key structural property of the dynamics, and because we study the setting where the embeddings are unknown, which necessitates non-linear function approximation.

Turning to reinforcement learning with function approximation and exploration, a large body of effort focuses on (essentially) linear methods (Yang and Wang, 2020; Jin et al., 2020b; Cai et al., 2020; Modi et al., 2020; Du et al., 2019c; Wang et al., 2019; Agarwal et al., 2020). Closest to our work are the results of Jin et al. (2020b) and Yang and Wang (2020), who consider low rank MDPs with known feature maps $\phi^\star_{0:H-1}$ (Yang and Wang (2020) also assumes that $\mu^\star_{0:H-1}$ is known up to a linear map). These results motivate our representation learning formulation, but, on their own, these algorithms cannot leverage the inductive biases provided by neural networks to scale to rich state spaces.

There are methods for more general, non-linear, function approximation, but these works either (a) require strong environment assumptions such as determinism (Wen and Van Roy, 2013; Du et al., 2020), (b) require strong function class assumptions such as bounded Eluder dimension (Russo and Van Roy, 2013; Osband and Van Roy, 2014), (c) have sample complexity scaling linearly with the function class size (Lattimore et al., 2013; Ortner et al., 2014) or (d) are computationally intractable (Jiang et al., 2017; Sun et al., 2019; Dong et al., 2020). Note that Ortner et al. (2014); Jiang et al. (2015) consider a form of representation learning, abstraction selection, but the former scales linearly with the number of candidate abstractions, while the latter does not address exploration.

Bellman/Witness rank. We briefly expand on this final category of computationally inefficient methods. For model-free reinforcement learning, Jiang et al. (2017) give an algebraic condition, in terms of a notion called the Bellman rank, on the environment and a given function approximation class, under which sample efficient reinforcement learning is always possible. Sun et al. (2019) extend the definition to model-based approaches, with the notion of Witness rank. As we will see in the next section, the low rank MDP with a function class derived from Φ (and Υ) admits low Bellman (resp., Witness) rank, and so these results imply that our setting is statistically tractable.

Block MDPs. Finally, we turn to theoretical works on representation learning for RL. Du et al. (2019b) introduce the *block MDP* model, in which there is a finite latent state space $\mathcal{S}$ that governs the transition dynamics, and each "observation" $x \in \mathcal{X}$ is associated with a latent state $s \in \mathcal{S}$, so the state is *decodable*. The natural representation learning goal is to recover the latent states, and Du et al. (2019b); Misra et al. (2020) show that this can be done, in concert with exploration, in a statistically and computationally efficient manner. Since the block MDP can be easily expressed as a low rank MDP, our results can be specialized to this setting, where they yield comparable guarantees. On the other hand, we will see that the low rank MDP is significantly more expressive, and so our results greatly expand the scope for provably efficient representation learning and reinforcement learning.

4 Expressiveness of low rank MDPs

Before turning to our algorithmic development, we discuss connections between low-rank MDPs and models studied in prior work. This discussion is facilitated by formalizing a connection between MDP transition operators and latent variable graphical models.

Definition 2. *The* latent variable representation *of a transition operator $T : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ is a latent space $\mathcal{Z}$ along with functions $\psi : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{Z})$ and $\nu : \mathcal{Z} \to \Delta(\mathcal{X})$, such that $T(\cdot \mid x, a) = \int \nu(\cdot \mid z)\psi(z \mid x, a)dz$. The* latent variable dimension *of T, denoted d_{LV} is the cardinality of smallest latent space $\mathcal{Z}$ for which T admits a latent variable representation.*

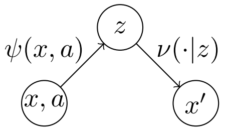

Figure 1: The latent variable interpretation.

See Figure 1. In this representation, (1) each (x, a) pair induces a "posterior" distribution $\psi(x, a) \in \Delta(\mathcal{Z})$ over z, (2) we sample $z \sim \psi(x, a)$, and (3) then sample $x' \sim \nu(\cdot \mid z)$, where ν specifies the "emission" distributions. As notation, we typically write $\nu(x) \in \mathbb{R}^{\mathcal{Z}}$ with coordinates $\nu(x)[z] = \nu(x \mid z)$ and we call ψ, ν the *simplex features*, following the example described by Jin et al. (2020b). When considering H-step MDPs, this representation allows us to augment the trajectory τ with the

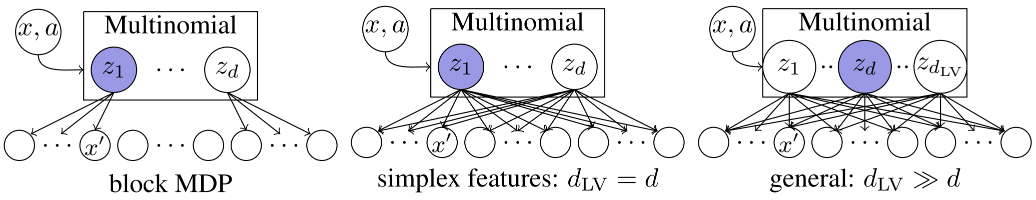

Figure 2: The latent variable interpretation of low rank MDPs, where (x, a) induces a distribution over latent variable z. Left: in block MDPs, latent variables induce a partition over the next state x'. Center: simplex features have embedding dimension equal to the number of latent variables. Right: low rank MDPs can have exponentially more latent variables than the dimension, $d_{\mathrm{LV}} \gg d$.

latent variables $\tau = (x_0, a_0, z_1, x_1, \ldots z_{H-1}, x_{H-1}, a_{H-1}, x_H)$. Here note that z_h is the latent variable that generates x_h.

Note that all transition operators admit a trivial latent variable representation, as we may always take $\psi(x, a) = T(\cdot \mid x, a)$. However, when T is endowed with additional structure, the latent variable representations are more interesting. For example, this viewpoint already certifies a factorization $T(x' \mid x, a) = \langle \psi(x, a), \nu(x') \rangle$ with embedding dimension $|\mathcal{Z}|$, and so d_{LV} (if it is finite) is an upper bound on the rank of the transition operator. On the other hand, compared with Definition 1, this factorization additionally requires that $\psi(x, a)$ and $\nu(\cdot \mid z)$ are probability distributions. Since the factorization is non-negative, d_{LV} is the *non-negative rank* of the transition operator.

The latent variable representation enables a natural comparison of the expressiveness of various models, and, as we will see in the next section, yields insights that facilitate algorithm design. We now examine models that have been introduced in prior works and their properties relative to Definition 1.

Block MDPs. A block MDP (Du et al., 2019b; Misra et al., 2020) is clearly a latent variable model with $\mathcal{Z}$ corresponding to the latent state space $\mathcal{S}$ and the additional restriction that two latent variables z and z' have disjoint supports in their respective emissions $\nu(\cdot \mid z)$ and $\nu(\cdot \mid z')$ (see the left panel of Figure 2). Therefore, a block MDP is a low rank MDP with rank $d \leq |\mathcal{S}|$, but the next result shows that a low rank MDP is significantly more expressive.

Proposition 1. *For any $d \geq 2$ and any $M \in \mathbb{N}$ there exists an environment on $|\mathcal{X}| = M$ states, that can be expressed as a low rank MDP with embedding dimension d, but for which any block MDP representation must have M latent states.*

In fact, the MDP that we construct for the proof, admits a latent variable representation with $|\mathcal{Z}| = d$, but does not admit a non-trivial block MDP representation. This separation exploits the decodability restriction of block MDPs, which is indeed quite limiting in terms of expressiveness.

Simplex features. Given the latent variable representation and the fact that it certifies a rank of at most d_{LV}, it is natural to ask if this representation is canonical for all low rank MDPs. In other words, for any transition operator with rank d, can we express it as a latent variable model with $|\mathcal{Z}| = d$, or equivalently with simplex features of dimension d?

As discussed above, this model is indeed more expressive than the block MDP. However, the next result answers the above question in the negative. The latent variable representation is *exponentially* weaker than the general low rank representation in the following sense:

Proposition 2. *For any even $n \in \mathbb{N}$, there exists an MDP that can be cast as a low rank MDP with embedding dimension $O(n^2)$, but which has $d_{\mathrm{LV}} \geq 2^{\Omega(n)}$.*

See the center and right panels of Figure 2. The result is proved by recalling that the latent variable dimension determines the non-negative rank of T, which can be much larger than its rank (Rothvoß, 2017; Yannakakis, 1991). It showcases how low rank MDPs are quite different from latent variable models of comparable dimension and demonstrates how embedding functions with negative values can provide significant expressiveness.

Bellman and Witness rank. As our last concrete connection, we remark here that the low rank MDP with a function class derived from Φ (and Υ) admits low Bellman (resp., Witness) rank.

Proposition 3 (Informal)**.** *The low rank MDP model always has Bellman rank at most d. Additionally, given Φ and assuming $\phi^\star_{0:H-1} \in \Phi$, we can construct a function classes $(\mathcal{G}, \Pi)$, so that* OLIVE *when run with $(\mathcal{G}, \Pi)$ has sample complexity $\tilde{O}\left(\mathrm{poly}(d, H, K, \log|\Phi|, \epsilon^{-1})\right)$.*

See Proposition 5 for a more precise statement. An analogous result holds for the Witness rank notion of Sun et al. (2019) (see Proposition 6 in the appendix). Unfortunately both OLIVE, and the algorithm of Sun et al. (2019) are not computationally tractable, as they involve enumeration of the employed function class. We turn to the development of computationally tractable algorithms in the next section.

5 Main results

We now turn to the design of algorithms for representation learning and exploration in low rank MDPs. As a computational abstraction, we consider the following optimization and sampling oracles.

Definition 3 (Computational oracles)**.** *Define the following oracles for the classes Φ, Υ:*

1. *The* maximum likelihood oracle, MLE, *takes a dataset D of (x, a, x') triples, and returns*

$$\textsc{Mle}(D) := \mathrm{argmax}_{\phi\in\Phi,\mu\in\Upsilon} \textstyle\sum_{(x,a,x')\in D} \log(\langle \phi(x,a), \mu(x')\rangle).$$

2. *The* sampling oracle, SAMP, *is a subroutine which, for any $(\phi, \mu) \in \Phi \times \Upsilon$ and any (x, a), returns a sample $x' \sim \langle \phi(x,a), \mu(\cdot)\rangle$. Multiple calls to the procedure result in independent samples.*

We assume access to both oracles as a means towards practical algorithms that avoid explicitly enumerating over all functions in Φ and Υ. Related assumptions are quite common in the literature (Misra et al., 2020; Du et al., 2019b; Agarwal et al., 2014), and in practice, both oracles can be reasonably approximated whenever optimizing over Φ, Υ is feasible (e.g., neural networks). Regarding MLE, other optimization oracles are possible, and in the appendix (Remark 19) we sketch how our proof can accommodate a generative adversarial oracle as a replacement (Goodfellow et al., 2014; Arora et al., 2017). While the sampling oracle is less standard, one might implement SAMP via optimization methods like the Langevin dynamics (Welling and Teh, 2011) or through reparametrization techniques such as the Gumbel-softmax trick (Jang et al., 2017; Figurnov et al., 2018).[4]

5.1 Algorithm description

The algorithm is called FLAMBE, for “Feature Learning And Model-Based Exploration.” Pseudocode is displayed in Algorithm 1. Before turning to the description, we clarify our use of h-step policies and policy mixtures. Throughout the algorithm we compute policies by optimizing reward functions that are only defined at a single stage $\tilde{h} \in [H]$. This yields an $\tilde{h}$*-step policy*, and when executing such policy, we always terminate the episode by taking actions $\{a_h\}_{h>\tilde{h}}$ uniformly at random. We also use *policy mixtures* or distributions over policies. When executing a policy mixture, we sample a policy from the distribution at the beginning of the episode and use that policy for the entire episode.

FLAMBE is iterative in nature, where in iteration j, we use an exploration policy ρ_{j-1} to collect a dataset of transitions (Line 6), and then we pass all the transitions collected so far to the MLE oracle (Line 7). The MLE oracle returns embedding functions $(\widehat{\phi}_h, \widehat{\mu}_h)$ for each h, which define transition operators $\widehat{T}_h$ for the learned model $\widehat{\mathcal{M}}$. Then FLAMBE calls a planning sub-routine to compute the exploration policy ρ_j (Lines 10- 11) that we use in the next iteration. After $J_{\max}$ iterations, we output the current model $\widehat{\mathcal{M}}$.

For the planning step, intuitively we seek an exploratory policy ρ that induces good coverage over the state space when executed in the model. We do this by solving one planning problem per time step h in Algorithm 2 using a technique inspired by elliptical potential arguments from linear bandits (Dani et al., 2008). Using the $\tilde{h}$-step model $\widehat{T}_{0:\tilde{h}-1}$, we iteratively maximize certain quadratic forms of our learned features $\widehat{\phi}_{\tilde{h}-1}$ to find new directions not covered by the previously discovered policies, and we update the exploratory policy mixture to include the maximizer. The planning algorithm terminates when no policy can achieve large quadratic form, which implies that we have found all reachable directions in $\widehat{\phi}_{h-1}$. This yields a policy mixture ρ_h^{pre} with component policies that are

[4] We do not explicitly consider approximate oracles, but additive approximations can be accommodated in our proof. In particular, if SAMP returns a sample from a distribution that is $\varepsilon_{\mathrm{samp}}$ close in total variation to the target distribution in $\mathrm{poly}(1/\varepsilon_{\mathrm{samp}})$ time, then we retain computational efficiency.

Algorithm 1 FLAMBE: Feature Learning And Model-Based Exploration

```
Input: Environment $\mathcal{M}$, function classes $\Phi, \Upsilon$, subroutines MLE and SAMP, parameters $\beta, n$.
For each $h$ set $\rho_h^{\text{pre}}$ to be the random policy, which takes all actions uniformly at random.
Set $\rho_0 = \{\rho_h^{\text{pre}}\}_{h=0}^{H-1}$ and $D_h = \emptyset$ for each $h \in \{0, \ldots, H-1\}$.
for $j = 1, \ldots, J_{\max}$ do
    for $h = 0, \ldots, H-1$ do
        Obtain $D_h \leftarrow D_h \cup$ EXECUTE_IN_REAL_WORLD$(\rho_{j-1}, h, n)$.
        Solve maximum likelihood problem: $(\widehat{\phi}_h, \widehat{\mu}_h) \leftarrow$ MLE$(D_h)$.
        Set $\widehat{T}_h(x_{h+1} \mid x_h, a_h) = \left\langle \widehat{\phi}_h(x_h, a_h), \widehat{\mu}_h(x_{h+1}) \right\rangle$.
    end for
    For each $h$, call planner (Algorithm 2) with $h$ step model $\widehat{T}_{0:h-1}$ and $\beta$ to obtain $\rho_h^{\text{pre}}$.
    Set $\rho_j = \{\rho_h^{\text{pre}}\}_{h=0}^{H-1}$ to be the set of exploration policies.
end for
function EXECUTE_IN_REAL_WORLD$(\rho = \{\rho_t^{\text{pre}}\}_{t=0}^{H-1}, h, n)$
    Initialize $D = \emptyset$.
    for $i = 1, \ldots, n$ do
        Pick $t \in \{0, \ldots, H-1\}$ uniformly at random.
        Starting in $x_0$, execute $h-1$ actions using $\rho_t^{\text{pre}}$ to get $x_h$. Take $a_h$ uniformly to get $x_{h+1}$.
        Augment $D \leftarrow D \cup \{(x_h, a_h, x_{h+1})\}$.
    end for
    return $D$.
end function
```

Algorithm 2 Elliptical planner

```
Input: MDP $\widetilde{\mathcal{M}} = (\phi_{0:\tilde{h}}, \mu_{0:\tilde{h}})$, subroutine SAMP, parameter $\beta > 0$. Initialize $\Sigma_0 = I_{d \times d}$.
for $t = 1, 2, \ldots,$ do
    Use SAMP to compute (see text for details)
        $$\pi_t = \operatorname*{argmax}_{\pi} \mathbb{E}\left[\phi_{\tilde{h}}(x_{\tilde{h}}, a_{\tilde{h}})^\top \Sigma_{t-1}^{-1} \phi_{\tilde{h}}(x_{\tilde{h}}, a_{\tilde{h}}) \mid \pi, \widetilde{\mathcal{M}}\right].  \qquad (2)$$
    If the objective is at most $\beta$, halt and output $\rho = \text{unif}(\{\pi_\tau\}_{\tau<t})$.
    Compute $\Sigma_{\pi_t} = \mathbb{E}\left[\phi_{\tilde{h}}(x_{\tilde{h}}, a_{\tilde{h}})\phi_{\tilde{h}}(x_{\tilde{h}}, a_{\tilde{h}})^\top \mid \pi, \widetilde{\mathcal{M}}\right]$. Update $\Sigma_t \leftarrow \Sigma_{t-1} + \Sigma_{\pi_t}$.
end for
```

linear in the learned features $\widehat{\phi}_{0:h-1}$. The challenge in our analysis is to relate this coverage in the model to that in the true environment as we discuss in the next section.

Algorithm 2 is a model-based planner, so it requires no interaction with the environment. The main computational step is the optimization problem (2), which is a policy optimization problem in a known low rank MDP from which we can sample efficiently. We solve this problem by running the reinforcement learning algorithm of Jin et al. (2020b) (See Lemma 14 in the appendix) with the reward function corresponding to the objective in (2) and using SAMP to simulate the algorithm's interaction with the environment. Note that we are optimizing over *all policies*, which is possible because the Bellman backups in a low rank MDP are linear functions of the features (c.f., Lemma 1). The sampling oracle can also be used to approximate all expectations, and, with sufficient accuracy, this has no bearing on the final results. Our proofs do account for the sampling errors.

5.2 Theoretical Results

We now state the main guarantee.

Theorem 2. *Fix $\delta \in (0, 1)$. If $\mathcal{M}$ is a low rank MDP with dimension d and horizon H and Assumption 1 holds, then* FLAMBE *with subroutine Algorithm 2 and appropriate settings*[5] *of β, $J_{\max}$, and n, computes a model $\widehat{\mathcal{M}}$ such that (1) holds with probability at least $1 - \delta$. The total number of*

[5] The precise settings for β, $J_{\max}$, and n are given in the appendix.

trajectories collected is

$$\tilde{O}\left(\frac{H^{22}K^9d^7\log(|\Phi||\Upsilon|/\delta)}{\varepsilon^{10}}\right).$$

The algorithm runs in polynomial time with polynomially many calls to MLE *and* SAMP *(Definition 3).*

Thus, FLAMBE provably learns low rank MDP models in a statistically and computationally efficient manner, under Assumption 1. While the result is comparable to prior work in the dependencies on d, H, K and ε, we instead highlight the more conceptual advances over prior work.

- The key advancement over the block MDP algorithms (Du et al., 2019b; Misra et al., 2020) is that FLAMBE applies to a significantly richer class of models with comparable function approximation assumptions.[6] A secondary, but important, improvement is that FLAMBE *does not require any reachability assumptions*, unlike these previous results. We remark that Feng et al. (2020) avoid reachability restrictions in block MDPs, but their function approximation/oracle assumptions are much stronger than ours.[7]
- Over Jin et al. (2020b); Yang and Wang (2020), the key advancement is that we address the representation learning setting where the embeddings $\phi^\star_{0:H-1}$ are not known a priori. On the other hand, our bound scales with the number of actions K. We believe that additional structural assumptions on Φ are required to avoid the dependence on K in the representation learning setting.
- Over Jiang et al. (2017); Sun et al. (2019), the key advancement is computational efficiency. However, the low rank MDP is less general than what is covered by their theory, and our sample complexity is worse in the polynomial factors.

As remarked earlier, the logarithmic dependence on the sizes of Φ, Υ can be relaxed to alternative notions of capacity for continuous classes.

We also state a sharper bound for a version of FLAMBE that operates directly on the simplex factorization. The main difference is that we use a conceptually simpler planner (See Algorithm 3 in the appendix) and the sample complexity bound scales with d_{LV}.

Theorem 3. *Fix $\delta \in (0,1)$. If $\mathcal{M}$ admits a simplex factorization with embedding dimension d_{LV}, Assumption 1 holds, and all $\phi \in \Phi$ satisfy $\phi(x,a) \in \Delta([d_{\mathrm{LV}}])$, then* FLAMBE *with Algorithm 3 as the subroutine and appropriate setting*[8] *of $J_{\max}$ and n computes a model $\widehat{\mathcal{M}}$ such that (1) holds with probability at least $1-\delta$. The total number of trajectories collected is*

$$\tilde{O}\left(\frac{H^{11}K^5d_{\mathrm{LV}}^5\log(|\Phi||\Upsilon|/\delta)}{\varepsilon^{3}}\right).$$

The algorithm runs in polynomial time with polynomially many calls to MLE *and* SAMP *(Definition 3).*

This bound scales much more favorably with H, K and ε, but incurs a polynomial dependence on the latent variable dimension d_{LV}, instead of the embedding dimension d. For many problems, including block MDPs, we expect that $d_{\mathrm{LV}} \approx d$, in which case using this version of FLAMBE may be preferable. However, Theorem 3 requires that we encode simplex constraints into our function class Φ, for example using the softmax. When d_{LV} is small, this may be a practically useful design choice.

Challenges in the analysis. We highlight three main challenges in the analysis. The first challenge arises in the analysis of the model learning step, where we want to show that our model $\widehat{T}_h$, learned by maximum likelihood estimation, accurately approximates the true dynamics T_h on the data distributions induced by the exploratory policies $\rho_{0:j-1}$. This requires a generalization argument, but both the empirical MLE objective and population version — the KL divergence — are unbounded, so we cannot use standard uniform convergence techniques. Instead, we employ (and slightly adapt)

[6] The realizability assumption for Block MDPs implies realizability of $\phi^\star_{0:H-1}$ and the support of the components of $\mu^\star_{0:H-1}$ (recall that for Block MDPs these components have disjoint support). This is slightly weaker than Assumption 1, but stronger than just assuming realizability of $\phi^\star_{0:H-1}$.

[7] Unlike ours, their oracle assumption is not a purely computational abstraction, but rather it implicitly places statistical restrictions on the emission distributions.

[8] This version does not require the parameter β.

results from the statistics literature (Van de Geer, 2000; Zhang, 2006) to show that MLE yields convergence in the Hellinger and total variation distances. While these arguments are well-known in the statistics community, we highlight them here because we believe they may be broadly useful in the context of model-based RL.

The second challenge is to transfer the model error guarantee from the exploratory distributions to distributions induced by other policies. Intuitively, error transfer should be possible if the exploratory policies cover the state space, but we must determine how we measure and track coverage. Leveraging the low rank dynamics, we show that if a policy π induces a distribution over true features $\phi^\star_{h-1}$ that is in the span of the directions visited by the exploratory policies, then we can transfer the MLE error guarantee for $\widehat{T}_h$ to π's distribution. This suggests measuring coverage for time h in terms of the second moment matrix of the true features at the previous time induced by the exploratory policies ρ, that is $\Sigma_{h-1} = \mathbb{E}_\rho \phi^\star_{h-1}\phi^{\star,\top}_{h-1}$. Using these matrices, we prove a sharp *simulation lemma* that bounds the model error observed by a policy π in terms of the model error on the exploratory distribution and the probability that π visits features for which $\phi^{\star,\top}_{h-1}\Sigma^{-1}_{h-1}\phi^\star_{h-1}$ is large.

The simulation lemma suggests a planning strategy and an "explore-or-terminate" argument: the next exploratory policy should maximize $\phi^{\star,\top}_{h-1}\Sigma^{-1}_{h-1}\phi^\star_{h-1}$, in which case either we visit some new direction and make progress, or we certify (1), since no policy can make this quantity large. However, we cannot maximize this objective directly, since we do not know $\phi^\star_{h-1}$ or Σ_{h-1}! Moreover, we cannot use $\widehat{\phi}_{h-1}$ to approximate $\phi^\star_{h-1}$ in the objective, since even if the model is accurate, the features may not be. Instead we plan to find a policy that induces a well-conditioned covariance matrix in terms of $\widehat{\phi}_{h-2}$, the learned features at time $h-2$. By composing this policy with a random action and applying our simulation lemma, we can show that this policy either explores at some previous time or approximately maximizes $\phi^{\star,\top}_{h-1}\Sigma^{-1}_{h-1}\phi^\star_{h-1}$, which allows us to apply the explore-or-terminate argument. Note that this argument uses two random actions: we take a_{h-1} at random to approximately maximize $\phi^{\star,\top}_{h-1}\Sigma^{-1}_{h-1}\phi^\star_{h-1}$, even though we optimize with $\widehat{\phi}_{h-2}$, and then we take a_h at random for the MLE step. Combining this reasoning with an elliptical potential argument, we can bound the number of iterations in which exploration can happen, which leads to the final result.

6 Discussion

This paper studies representation learning and exploration for low rank MDPs. We provide an intuitive interpretation of these models in terms of a latent variable representation, and we prove a number of structural results certifying that low rank MDPs are significantly more expressive than models studied in prior work. We also develop FLAMBE, a computationally and statistically efficient model-based algorithm for system identification in low rank MDPs. Policy optimization follows as a corollary.

Our results raise a number of promising directions for future work. On the theoretical side, in the reward-sensitive setting, can we avoid learning the entire model, perhaps by utilizing "value-aware" methods (Farahmand et al., 2017; Ayoub et al., 2020)? Can we remove realizability conditions on $\mu^\star_{0:H-1}$ and develop provably efficient model-free algorithms for representation learning in the low rank MDP? On the empirical side, can we leverage the algorithmic insights of FLAMBE to develop new practically effective representation learning algorithms for complex reinforcement learning tasks? Finally, this work considers representation learning for a set of related RL problems which all share the same underlying dynamics, but differ in their reward functions. A natural future direction is to consider learning representations that enable more general task variations. We look forward to answering these questions in future work.

Broader impact

This paper is theoretical in nature, and so we expect the ethical and societal consequences of our specific results to be minimal. More broadly, we do expect that reinforcement learning will have significant impact on society. There is much potential for benefits to humanity in the often-referenced application domains of precision medicine, personalized education, and elsewhere. There is also much potential for harms, both malicious and unintentional. To this end, we hope that research into the foundations of reinforcement learning can help enable these applications and mitigate harms through the development of algorithms that are efficient, robust, and safe.

Acknowledgments and Disclosure of Funding

We thank Ruosong Wang for insightful discussions about removing reachability restrictions. Funding in direct support of this work was provided by Microsoft. SK gratefully acknowledges funding from the ONR award N00014-18-1-2247 and NSF Awards CCF-1703574 and CCF-1740551. AK gratefully acknowledges funding from NSF Award IIS-1763618. This work was completed while WS was at Microsoft Research. Additional relevant institutions: University of Washington, University of Massachusetts Amherst, Cornell University.

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, 2014.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. PC-PG: Policy cover directed exploration for provable policy gradient learning. In *Advances in Neural Information Processing Systems*, 2020.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *International Conference on Machine Learning*, 2017.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, 2019.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin F Yang. Model-based reinforcement learning with value-targeted regression. *arXiv:2006.01107*, 2020.

Andre Barreto, Doina Precup, and Joelle Pineau. Reinforcement learning using kernel-based stochastic factorization. In *Advances in Neural Information Processing Systems*, 2011.

André MS Barreto and Marcelo D Fragoso. Computing the stationary distribution of a finite markov chain through stochastic factorization. *SIAM Journal on Matrix Analysis and Applications*, 2011.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 2000.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, 2016.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, 2020.

Joel E Cohen and Uriel G Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matices. *Linear Algebra and its Applications*, 1993.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, 2008.

Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2012.

Kefan Dong, Jian Peng, Yining Wang, and Yuan Zhou. $\sqrt{n}$-regret for learning in Markov decision processes with function approximation and low Bellman rank. In *Conference on Learning Theory*, 2020.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2019a.

Simon S Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019b.

Simon S Du, Yuping Luo, Ruosong Wang, and Hanrui Zhang. Provably efficient Q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, 2019c.

Simon S Du, Jason D Lee, Gaurav Mahajan, and Ruosong Wang. Agnostic Q-learning with function approximation in deterministic systems: Tight bounds on approximation error and sample complexity. In *Advances in Neural Information Processing Systems*, 2020.

Yaqi Duan, Mengdi Wang, Zaiwen Wen, and Yaxiang Yuan. Adaptive low-nonnegative-rank approximation for state aggregation of Markov chains. *SIAM Journal on Matrix Analysis and Applications*, 2020.

Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards–B*, 1965.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, 2017.

Fei Feng, Ruosong Wang, Wotao Yin, Simon S Du, and Lin F Yang. Provably efficient exploration for rl with unsupervised learning. In *Advances in Neural Information Processing Systems*, 2020.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, 2018.

Samuel Fiorini, Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. Combinatorial bounds on nonnegative rank and extended formulations. *Discrete Mathematics*, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

Elad Hazan, Sham M Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, 2015.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, 2017.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, 2020a.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020b.

Tor Lattimore and Csaba Szepesvari. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, 2020.

Tor Lattimore, Marcus Hutter, Peter Sunehag, et al. The sample-complexity of general reinforcement learning. In *International Conference on Machine Learning*, 2013.

Michael L Littman and Richard S Sutton. Predictive representations of state. In *Advances in Neural Information Processing Systems*, 2002.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 2016.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International Conference on Machine Learning*, 2020.

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *Conference on Artificial Intelligence and Statistics*, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.

Ronald Ortner, Odalric-Ambrym Maillard, and Daniil Ryabko. Selecting near-optimal approximate state representations in reinforcement learning. In *International Conference on Algorithmic Learning Theory*. Springer, 2014.

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, 2014.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized Markov chains for next-basket recommendation. In *International Conference on World Wide Web*, 2010.

Thomas Rothvoß. The matching polytope has exponential extension complexity. *Journal of the ACM*, 2017.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, 2013.

Satinder Singh, Michael R James, and Matthew R Rudary. Predictive state representations: a new theory for modeling dynamical systems. In *Conference on Uncertainty in Artificial Intelligence*, 2004.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, 2019.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Michael Thon and Herbert Jaeger. Links between multiplicity automata, observable operator models and predictive state representations: a unified learning framework. *The Journal of Machine Learning Research*, 2015.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *arXiv:2003.02234*, 2020.

Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

Sara Van de Geer. *Empirical Processes in M-estimation*. Cambridge University Press, 2000.

Benjamin Van Roy and Shi Dong. Comments on the Du-Kakade-Wang-Yang lower bounds. *arXiv:1911.07910*, 2019.

Yining Wang, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. *arXiv:1912.04136*, 2019.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.

Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems*, 2013.

Lin Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, 2019.

Lin F Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, 2020.

Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 1991.

Hengshuai Yao, Csaba Szepesvári, Bernardo Avila Pires, and Xinhua Zhang. Pseudo-mdps and factored linear action models. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2014.

Tong Zhang. From ϵ-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 2006.