Provable Model-based Nonlinear Bandit and Reinforcement Learning: Shelve Optimism, Embrace Virtual Curvature

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper studies model-based bandit and reinforcement learning (RL) with nonlinear function approximations. We propose to study convergence to approximate local maxima because we show that global convergence is statistically intractable even for one-layer neural net bandit with a deterministic reward. For both nonlinear bandit and RL, the paper presents a model-based algorithm, Virtual Ascent with Online Model Learner (ViOlin), which provably converges to a local maximum with sample complexity that only depends on the sequential Rademacher complexity of the model class. Our results imply novel sample complexity bounds on several concrete settings such as linear bandit with finite or sparse model class, and two-layer neural net bandit. A key algorithmic insight is that optimism may lead to over-exploration even for two-layer neural net model class. On the other hand, for convergence to local maxima, it suffices to maximize the virtual return if the model can also reasonably predict the gradient and Hessian of the real return.

1 Introduction

Recent progresses demonstrate many successful applications of deep reinforcement learning (RL) in robotics [48], games [7, 65], computational biology [55], etc. However, theoretical understanding of deep RL algorithms is limited. Last few years witnessed a plethora of results on linear function approximations in RL [78, 64, 40, 72, 70, 19, 3, 32], but the analysis techniques appear to strongly rely on (approximate) linearity and hard to generalize to neural networks.

The goal of this paper is to theoretically analyze model-based nonlinear bandit and RL with neural net approximation, which achieves amazing sample-efficiency in practice (see e.g., [37, 10, 30, 31, 12]). We focus on the setting where the state and action spaces are continuous.

Past theoretical work on model-based RL studies families of dynamics with restricted complexity measures such as Eluder dimension [59], witness rank [67], the linear dimensionality [75], and others [57, 42, 17]. Implications of these complexity measures have been studied, e.g., finite mixture of dynamics [6] and linear models [63] have bounded Eluder dimensions. However, it turns out that none of the complexity measures apply to the family of MDPs with even barely nonlinear dynamics, e.g., MDPs with dynamics parameterized by all one-layer neural network with a single activation unit (and with bounded weight norms). For example, in Theorem 5.2, we will prove that one-layer neural nets do not have polynomially-bounded Eluder dimension.[1] (See more evidence below.)

The limited progress on neural net approximation is to some extent not surprising. Given a deterministic dynamics with *known* neural net parameters, finding the best parameterized policy still involves

[1] This result is also proved by the concurrent Li et al. [49, Theorem 8] independently.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

optimizing a complex non-concave function, which is in general computationally intractable. More fundamentally, we find that it is also *statistically intractable* for solving the one-hidden-layer neural net bandit problem (which is a strict sub-case of deep RL). In other words, it requires exponential (in the input dimension) samples to find the global maximum (see Theorems 5.1). This also shows that conditions in past work that guarantee global convergence cannot apply to neural nets.

Given these strong impossibility results, we propose to reformulate the problem to finding an approximate local maximum policy with guarantees. This is in the same vein as the recent fruitful paradigm in non-convex optimization where researchers disentangle the problem into showing that all local minima are good and fast convergence to local minima (e.g., see [28, 26, 27, 29, 47]). In RL, local maxima can often be global as well for many cases [4].[2]

Zero-order optimization or policy gradient algorithms can converge to local maxima and become natural potential competitors. They are widely believed to be less sample-efficient than the model-based approach because the latter can leverage the extrapolation power of the parameterized models. Theoretically, our formulation aims to characterize this phenomenon with results showing that the model-based approach's sample complexity mostly depends (polynomially) on the complexity of the model class, whereas policy gradient algorithms' sample complexity polynomially depend on the dimensionality of policy parameters (in RL) or actions (in bandit). Our technical goal is to answer the following question:

> Can we design algorithms that converge to approximate local maxima with sample complexities that depend *only and polynomially* on the complexity measure of the *dynamics/reward* class?

We note that this question is open even if the dynamics hypothesis class is finite, and the complexity measure is the logarithm of its size. The question is also open even for nonlinear bandit problems (where dynamics class is replaced by reward function class), with which we start our research. We consider first nonlinear bandit with *deterministic* reward where the reward function is given by $\eta(\theta, a)$ for action $a \in \mathcal{A}$ under instance $\theta \in \Theta$. We use sequential Rademacher complexity [61, 62] to capture the complexity of the reward function η. Our main result for nonlinear bandit is stated as follows.

Theorem 1.1 (Informal version of Theorem 3.1)**.** *Suppose the sequential Rademacher complexity of a loss function class (defined later) induced by the reward function class $\{\eta(\theta, \cdot) : \theta \in \Theta\}$ is bounded by $\sqrt{R(\Theta)T\mathrm{polylog}(T)}$. Then, there exists an algorithm (ViOlin, Alg. 1) that finds an ϵ-approximate local maximum with $\widetilde{\mathcal{O}}(R(\Theta)\epsilon^{-8})$ samples.*

In contrast to zero-order optimization, which does not use the parameterization of η and has a sample complexity depending on the action dimension, our bound only depends on the complexity of the reward function class. This suggests that our algorithm exploits the extrapolation power of the reward function class. To the best of our knowledge, this is the first action-dimension-free result for both linear and nonlinear bandit problems. More concretely, we instantiate our theorem to the following settings and get new results that leverage the model complexity (more in Section 3.1).

1. Linear bandit with finite parameter space Θ. Because η is concave in action a, our result leads to a sample complexity $\mathcal{O}(\mathrm{poly}(\log|\Theta|, 1/\epsilon))$ for finding an ϵ-approximate optimal action. In this case both zero-order optimization and the SquareCB algorithm in Foster, Rakhlin [24] have sample complexity/regret that depend on the dimension of action space $d_{\mathcal{A}}$.
2. Linear bandit with s-sparse or structured instance parameters. Our algorithm ViOlin achieves an $\mathcal{O}(\mathrm{poly}(s, 1/\epsilon))$ sample complexity when the instance/model parameter is s-sparse and the reward is deterministic. The sample complexity of zero-order optimization depends polynomially on $d_{\mathcal{A}}$. Carpentier, Munos [9] achieve a stronger $\widetilde{\mathcal{O}}(s\sqrt{T})$ regret bound for s-sparse linear bandits with actions set $\mathcal{A} = S^{d-1}$. In contrast, our ViOlin algorithm applies more generally to any structured instance parameter set. Other related results either leverage the rather strong anti-concentration assumption on the action set [71], or have implicit dimension dependency [33, Remark 4.3].
3. Two-layer neural nets bandit. Our algorithm finds an ϵ-approximate local maximum $\widetilde{\mathcal{O}}(\mathrm{poly}(1/\epsilon))$. Zero-order optimization can also find a local maximum but with $\Omega(d_{\mathcal{A}})$ samples.

[2] The all-local-maxima-are-global condition only needs to hold to the ground-truth total expected reward function. This potentially can allow disentangled assumptions on the ground-truth instance and the hypothesis class.

Optimistic algorithms in this case have an exponential sample complexity (see Theorem 5.3). Moreover, when the second layer of the ground-truth network contains all negative weights and the activation is convex and monotone, all local maxima are global because the reward is concave in the input (action) [5].

The results for bandit can be extended to model-based RL with deterministic nonlinear dynamics and deterministic reward.

Theorem 1.2 (Informal version of Theorem 4.4)**.** *Consider RL problems with deterministic dynamics class and stochastic policy class that satisfy some Lipschitz properties. Suppose the sequential Rademacher complexity of the ℓ_2 losses for learning the dynamics is bounded by $\sqrt{R(\Theta)T\mathrm{polylog}(T)}$. Then, ViOlin for RL (Alg. 2) finds an ϵ-approximate local maximum policy with $\widetilde{\mathcal{O}}(R(\Theta)\epsilon^{-8})$ samples.*

To the best of our knowledge, this is the first model-based RL algorithms with provable finite sample complexity guarantees (for local convergence) for general nonlinear dynamics. The work of [54] is the closest prior work which also shows local convergence, but its conditions likely cannot be satisfied by any parameterized models (including linear models). We also present a concrete example of RL problems with nonlinear models satisfying our Lipschitz assumptions in Example 4.3 of Section 4, which may also serve as a testbed for future model-based deep RL analysis. Other prior works on model-based RL do not apply to one-hidden-layer neural nets because they conclude global convergence which is not possible for one-hidden-layer neural nets in the worst case.

Technical novelty: exploring by model-based curvature estimate. The main challenge is that the optimism-in-face-of-uncertainty exploration principle seems to be too aggressive—for a nonlinear model class, even if the ground-truth model is linear, optimistic exploration leads to exponentially large model cumulative prediction errors and exponential sample complexity (see Theorem 5.3). We use an online learning oracle to learn the dynamics, which can dramatically reduce the model prediction errors by hedging the risks. However, now, choosing actions or policies by maximizing virtual return may lack exploration. The main novelty of our algorithm is that we augment the online learning loss with gradient and Hessian estimation errors. The key insight is that, in order to ensure sufficient exploration for converging to local maxima, it suffices for the model to predict the virtual return, its gradient and Hessian reasonably accurately. We refer to the approach as "model-based curvature estimate". Our algorithm alternates between maximizing virtual return (over action or policy) and learning the model parameters by an online learner.

Because the algorithm leverages model extrapolation, the sample complexity of model-based curvature prediction depends on the model complexity instead of action dimension in the zero-optimization approach. Foster, Rakhlin [24] also propose algorithms that do not rely on UCB—their exploration strategy either relies on the finite discrete action space, or leverages the linear structure in the action space and has action-dimension dependency. In contrast, our algorithms' exploration relies more on the learning of the model. Consequently, our sample complexity can be action-dimension-free.

2 Problem Setup and Preliminaries

In this section, we first introduce our problem setup for nonlinear bandit and reinforcement learning, and then the preliminary for online learning and sequential Rademacher complexity.

2.1 Nonlinear Bandit Problem with Deterministic Reward

We consider *deterministic* nonlinear bandits with continuous actions. Let $\theta \in \Theta$ be the parameter that specifies the bandit instance, $a \in \mathbb{R}^{d_{\mathcal{A}}}$ the action, and $\eta(\theta, a) \in [0, 1]$ the reward function. Let $\theta^\star$ denote the unknown ground-truth parameter. Throughout the paper, we work under the *realizability assumption* that $\theta^\star \in \Theta$. A bandit algorithm aims to maximize $\eta(\theta^\star, a)$. Let $a^\star = \mathrm{argmax}_a \, \eta(\theta^\star, a)$ be the optimal action (breaking tie arbitrarily). Let $\|H\|_{\mathrm{sp}}$ be the spectral norm of a matrix H. We assume that the reward function, its gradient and Hessian matrix are Lipschitz, which are standard in the optimization literature (e.g., Johnson, Zhang [41], Ge et al. [26]).

Assumption 2.1. *We assume that for all* $\theta \in \Theta$, $\sup_a \|\nabla_a \eta(\theta, a)\|_2 \leq \zeta_g$ *and* $\sup_a \left\|\nabla_a^2 \eta(\theta, a)\right\|_{\mathrm{sp}} \leq \zeta_h$. *For every* $\theta \in \Theta$ *and* $a_1, a_2 \in \mathbb{R}^{d_{\mathcal{A}}}$, $\left\|\nabla_a^2 \eta(\theta, a_1) - \nabla_a^2 \eta(\theta, a_2)\right\|_{\mathrm{sp}} \leq \zeta_{3\mathrm{rd}} \|a_1 - a_2\|_2$.

As a motivation to consider deterministic rewards, we prove in Theorem C.1 for a special case that no algorithm can find a local maximum in less than $\sqrt{d_{\mathcal{A}}}$ steps. The result implies that an action-dimension-free sample complexity bound is impossible with standard sub-Gaussian noise.[3]

Approximate local maxima. In this paper, we aim to find a local maximum of the real reward function $\eta(\theta^\star, \cdot)$. A point x is an (ϵ_g, ϵ_h)-approximate local maximum of a twice-differentiable function $f(x)$ if $\|\nabla f(x)\|_2 \leq \epsilon_g$, and $\lambda_{max}(\nabla^2 f(x)) \leq \epsilon_h$. As argued in Sec. 1 and proved in Sec. 5, because reaching a global maximum is computational and statistically intractable for nonlinear problems, we only aim to reach a local maximum.

Sample complexity and local regret for converging to local maxima. Let a_t be the action that the algorithm takes at time step t. The sample complexity for converging to approximate local maxima is defined to be the minimal number of steps T such that there exists $t \in [T]$ where a_t is an (ϵ_g, ϵ_h) approximate local maximum with probability at least $1 - \delta$. We also define the "local regret" by comparing with an approximate local maximum. We defer the discussion to Appendix D.5.

2.2 Reinforcement Learning

A finite horizon Markov decision process (MDP) with deterministic dynamics is defined by a tuple $\langle T, r, H, \mu_1 \rangle$. Let $\mathcal{S}$ and $\mathcal{A}$ be the state and action spaces. The dynamics $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ gives next state, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function. H and μ_1 denote the horizon and distribution of initial state respectively. Without loss of generality, we assume that the state space is disjoint for different time steps. That is, there exists disjoint sets $\mathcal{S}_1, \cdots, \mathcal{S}_H$ such that $\mathcal{S} = \cup_{h=1}^H \mathcal{S}_h$, and for any $s_h \in \mathcal{S}, a_h \in a, T(s_h, a_h) \in \mathcal{S}_{h+1}$.

We consider parameterized policy and dynamics. Let $\Pi = \{\pi_\psi : \psi \in \Psi\}$ be the policy class and $\{T_\theta : \theta \in \Theta\}$ the dynamics class. The value function is $V_T^\pi(s_h) \triangleq \mathbb{E}[\sum_{h'=h}^H r(s_{h'}, a_{h'})]$, where $a_h \sim \pi(\cdot \mid s_h), s_{h+1} = T(s_h, a_h)$. Sharing the notation with the bandit setting, let $\eta(\theta, \psi) = \mathbb{E}_{s_1 \sim \mu_1} V_{T_\theta}^{\pi_\psi}(s_1)$ be the expected return of policy π_ψ under dynamics T_θ. Let ρ_T^π be the distribution of state action pairs when running policy π in dynamics T. For simplicity, we do not distinguish ψ, θ from π_ψ, T_θ when the context is clear. For example, we write $V_\theta^\psi = V_{T_\theta}^{\pi_\psi}$.

2.3 Preliminary on Online Learning with Stochastic Input Components

Consider a prediction problem where we aim to learn a function that maps from $\mathcal{X}$ to $\mathcal{Y}$ parameterized by parameters in Θ. Let $\ell((x, y); \theta)$ be a loss function that maps $(\mathcal{X} \times \mathcal{Y}) \times \Theta \to \mathbb{R}_+$. An online learner $\mathcal{R}$ aims to solve the prediction tasks under the presence of an adversarial nature iteratively. At time step t, the following happens.

1. The learner computes a distribution $p_t = \mathcal{R}(\{(x_i, y_i)\}_{i=1}^{t-1})$ over the parameter space Θ.
2. The adversary selects a point $\bar{x}_t \in \bar{\mathcal{X}}$ (which may depend on p_t) and generates a sample ξ_t from some fixed distribution q. Let $x_t \triangleq (\bar{x}_t, \xi_t)$, and the adversary picks a label $y_t \in \mathcal{Y}$.
3. The data point (x_t, y_t) is revealed to the online learner.

The online learner aims to minimize the expected regret in T rounds of interactions, defined as

$$\mathrm{REG}_T^{\mathrm{OL}} \triangleq \mathop{\mathbb{E}}_{\substack{\xi_t \sim q, \theta_t \sim p_t \\ \forall 1 \leq t \leq T}} \left[\sum_{t=1}^T \ell((x_t, y_t); \theta_t) - \inf_{\theta \in \Theta} \sum_{t=1}^T \ell((x_t, y_t); \theta) \right]. \tag{1}$$

The difference of the formulation from the most standard online learning setup is that the ξ_t part of the input is randomized instead of adversarially chosen (and the learner knows the distribution of ξ_t before making the prediction p_t). It was introduced by Rakhlin et al. [60], who considered a more generalized setting where the distribution q in round t can depend on $\{x_1, \cdots, x_{t-1}\}$.

[3] We rely on deterministic reward to estimate the gradient by finite difference. This method can be extended to stochastic rewards with multiple-point feedback [53].

We adopt the notation from Rakhlin et al. [60, 61] to define the (distribution-dependent) sequential Rademacher complexity of the loss function class $\mathcal{L} = \{(x, y) \mapsto \ell((x, y); \theta) : \theta \in \Theta\}$. For any set $\mathcal{Z}$, a $\mathcal{Z}$-valued tree with length T is a set of functions $\{\boldsymbol{z}_i : \{\pm 1\}^{i-1} \to \mathcal{Z}\}_{i=1}^T$. For a sequence of Rademacher random variables $\epsilon = (\epsilon_1, \cdots, \epsilon_T)$ and for every $1 \le t \le T$, we denote $\boldsymbol{z}_t(\epsilon) \triangleq \boldsymbol{z}_t(\epsilon_1, \cdots, \epsilon_{t-1})$. For any $\bar{\mathcal{X}}$-valued tree $\boldsymbol{x}$ and any $\mathcal{Y}$-valued tree $\boldsymbol{y}$, we define the sequential Rademacher complexity as

$$\mathfrak{R}_T(\mathcal{L}; \boldsymbol{x}, \boldsymbol{y}) \triangleq \mathbb{E}_{\xi_1, \cdots, \xi_t} \mathbb{E}_\epsilon \left[\sup_{\ell \in \mathcal{L}} \textstyle\sum_{t=1}^T \epsilon_t \ell((\boldsymbol{x}(\epsilon), \xi_t), \boldsymbol{y}(\epsilon)) \right]. \tag{2}$$

We also define $\mathfrak{R}_T(\mathcal{L}) = \sup_{\boldsymbol{x}, \boldsymbol{y}} \mathfrak{R}_T(\mathcal{L}; \boldsymbol{x}, \boldsymbol{y})$, where the supremum is taken over all $\bar{\mathcal{X}}$-valued and $\mathcal{Y}$-valued trees. Rakhlin et al. [60] proved the existence of an algorithm whose online learning regret satisfies $\textsc{reg}_T^{\mathrm{OL}} \le 2\mathfrak{R}_T(\mathcal{L})$.

3 Model-based Algorithms for Nonlinear Bandit

We first study model-based algorithms for nonlinear continuous bandits problem, which is a simplification of model-based reinforcement learning. We use the notations and setup in Section 2.1.

Abstraction of analysis for model-based algorithms. Typically, a model-based algorithm explicitly maintains an estimated model $\hat{\theta}_t$, and sometimes maintains a distribution, posterior, or confidence region of $\hat{\theta}_t$. We will call $\eta(\theta^\star, a)$ the *real reward* of action a, and $\eta(\hat{\theta}_t, a)$ the *virtual reward*. Most analysis for model-based algorithms (including UCB and ours) can be abstracted as showing the following two properties:

(i) the virtual reward $\eta(\hat{\theta}_t, a_t)$ is sufficiently high.
(ii) the virtual reward $\eta(\hat{\theta}_t, a_t)$ is close to the real reward $\eta(\theta^\star, a_t)$ in the long run.

One can expect that a proper combination of property (i) and (ii) leads to showing the real reward $\eta(\theta^\star, a_t)$ is high in the long run. Before describing our algorithms, we start by inspecting and summarizing the pros and cons of UCB from this viewpoint.

Pros and cons of UCB. The UCB algorithm chooses an action a_t and an estimated model $\hat{\theta}_t$ that maximize the virtual reward $\eta(\hat{\theta}_t, a_t)$ among those models agreeing with the observed data. The pro is that it satisfies property (i) by definition—$\eta(\hat{\theta}_t, a_t)$ is higher than the optimal real reward $\eta(\theta^\star, a^\star)$. The downside is that ensuring (ii) is challenging and often requires strong complexity measure bound such as Eluder dimension (which is not polynomial for even barely nonlinear models, as shown in Theorem 5.2). The difficulty largely stems from our very limited control of $\hat{\theta}_t$ except its consistency with the observed data. To bound the difference between the real and virtual rewards, we essentially require that *any* model that agrees with the past history should extrapolate to any future data accurately (as quantitatively formulated in Eluder dimension). Moreover, the difficulty of satisfying property (ii) is fundamentally caused by the over-exploration of UCB—As shown in the Theorem 5.3, UCB suffers from bad sample complexity with barely nonlinear family of models.

Our key idea: natural exploration via model-based curvature estimate. We deviate from UCB by readjusting the priority of the two desiderata. We prioritize ensuring property (ii) on large model class. We leverage a strong online learning algorithm to predict $\hat{\theta}_t$ with the objective that $\eta(\hat{\theta}_t, a_t)$ matches $\eta(\theta^\star, a_t)$. As a result, the difference between the virtual and real reward depends on the online learnability or the sequential Rademacher complexity of the model class. Sequential Rademacher complexity turns out to be a fundamentally more relaxed complexity measure than Eluder dimension—e.g., two-layer neural networks' sequential Rademacher complexity is polynomial in parameter norm and dimension, but the Eluder dimension is at least exponential in dimension (even with a constant parameter norm). However, an immediate consequence of using online-learned $\hat{\theta}_t$ is that we lose optimism/exploration that ensured property (i).[4]

[4]More concretely, the algorithm can get stuck when (1) a_t is optimal for $\hat{\theta}_t$, (2) $\hat{\theta}_t$ fits actions a_t (and history) accurately, but (3) $\hat{\theta}_t$ does not fit $a^\star$ (because online learner never sees $a^\star$). The passivity of online learning formulation causes this issue—the online learner is only required to predict well for the point that it saw and will see, but not for those points that it never observes.

Algorithm 1 ViOlin: **Vi**rtual Ascent with **On**lin**e** Model Learner (for Bandit)

Set parameter $\kappa_1 = 2\zeta_g$ and $\kappa_2 = 640\sqrt{2}\zeta_h$. Let $\mathcal{H}_0 = \emptyset$; choose $a_0 \in \mathcal{A}$ arbitrarily.
for $t = 1, 2, \cdots$ **do**
  Run $\mathcal{R}$ on $\mathcal{H}_{t-1}$ with loss function ℓ (defined in equation (5)) and obtain $p_t = \mathcal{R}(\mathcal{H}_{t-1})$.
  Let $a_t \leftarrow \operatorname{argmax}_a \mathbb{E}_{\theta_t \sim p_t}[\eta(\theta_t, a)]$.
  Sample $u_t, v_t \sim \mathcal{N}(0, I_{d_\mathcal{A} \times d_\mathcal{A}})$ independently.
  Let $\xi_t = (u_t, v_t)$, $\bar{x}_t = (a_t, a_{t-1})$, and $x_t = (\bar{x}_t, \xi_t)$
  Compute $y_t = [\eta(\theta^\star, a_t), \eta(\theta^\star, a_{t-1}), \langle \nabla_a \eta(\theta^\star, a_{t-1}), u_t\rangle, \langle \nabla_a^2 \eta(\theta^\star, a_{t-1}) u_t, v_t\rangle] \in \mathbb{R}^4$ by applying a finite number of actions in the real environments using equation (3) and (4).
  Update $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{(x_t, y_t)\}$

Our approach realizes property (i) in a sense that the virtual reward will improve iteratively if the real reward is not yet near a local maximum. This is much weaker than what UCB offers (i.e., that the virtual reward is higher than the optimal real reward), but suffices to show the convergence to a local maximum of the real reward function. We achieve this by demanding the estimated model $\hat{\theta}_t$ not only to predict the real reward accurately, but also to predict the gradient $\nabla_a \eta(\theta^\star, a)$ and Hessian $\nabla_a^2 \eta(\theta^\star, a)$ accurately. In other words, we augment the loss function for the online learner so that the estimated model satisfies $\eta(\hat{\theta}_t, a_t) \approx \eta(\theta^\star, a_t)$, $\nabla_a \eta(\hat{\theta}_t, a_t) \approx \nabla_a \eta(\theta^\star, a_t)$, and $\nabla_a^2 \eta(\hat{\theta}_t, a_t) \approx \nabla_a^2 \eta(\theta^\star, a_t)$ in the long run. This implies that when a_t is not at a local maximum of the real reward function $\eta(\theta^\star, \cdot)$, then it's not at a maximum of the virtual reward $\eta(\hat{\theta}_t, \cdot)$, and hence the virtual reward will improve in the next round if we take the greedy action that maximizes it.

Estimating projections of gradients and Hessians. To guide the online learner to predict $\nabla_a \eta(\theta^\star, a_t)$ correctly, we need a supervision for it. However, we only observe the reward $\eta(\theta^\star, a_t)$. Leveraging the deterministic reward property, we use rewards at a and $a + \alpha_1 u$ to estimate the projection of the gradient at a random direction u:

$$\langle \nabla_a \eta(\theta^\star, a), u\rangle = \lim_{\alpha_1 \to 0} \left(\eta(\theta^\star, a + \alpha_1 u) - \eta(\theta^\star, a)\right)/\alpha_1 \tag{3}$$

It turns out that the number of random projections $\langle \nabla_a \eta(\theta^\star, a), u\rangle$ needed for ensuring a large virtual gradient *does not* depend on the dimension, because we only use these projections to estimate the norm of the gradient but not necessarily the exact direction of the gradient (which may require d samples.) Similarly, we can also estimate the projection of Hessian to two random directions $u, v \in d_\mathcal{A}$ by:

$$\left\langle \nabla_a^2 \eta(\theta^\star, a) v, u\right\rangle = \lim_{\alpha_2 \to 0} \left(\langle \nabla_a \eta(\theta^\star, a + \alpha_2 v), u\rangle - \langle \nabla_a \eta(\theta^\star, a), u\rangle\right)/\alpha_2 \tag{4}$$

$$= \lim_{\alpha_2 \to 0} \lim_{\alpha_1 \to 0} \left((\eta(\theta^\star, a + \alpha_1 u + \alpha_2 v) - \eta(\theta^\star, a + \alpha_2 v)) - (\eta(\theta^\star, a + \alpha_1 u) - \eta(\theta^\star, a))\right)/(\alpha_1 \alpha_2)$$

Algorithmically, we can choose infinitesimal α_1 and α_2. Note that α_1 should be at least an order smaller than α_2 because the limitations are taken sequentially.

We create the following prediction task for an online learner: let θ be the parameter, $x = (a, a', u, v)$ be the input, $\hat{y} = [\eta(\theta, a), \eta(\theta, a'), \langle \nabla_a \eta(\theta, a'), u\rangle, \langle \nabla_a^2 \eta(\theta, a') u, v\rangle] \in \mathbb{R}^4$ be the output, and $y = [\eta(\theta^\star, a), \eta(\theta^\star, a'), \langle \nabla_a \eta(\theta^\star, a'), u\rangle, \langle \nabla_a^2 \eta(\theta^\star, a') u, v\rangle] \in \mathbb{R}^4$ be the supervision, and the loss function be

$$\begin{aligned}\ell(((a, a', u, v), y); \theta) \triangleq & ([\hat{y}]_1 - [y]_1)^2 + ([\hat{y}]_2 - [y]_2)^2 + \min\left(\kappa_1^2, ([\hat{y}]_3 - [y]_3)^2\right) \\ & + \min\left(\kappa_2^2, ([\hat{y}]_4 - [y]_4)^2\right)\end{aligned} \tag{5}$$

Here we used $[y]_i$ to denote the i-th coordinate of $y \in \mathbb{R}^4$ to avoid confusing with y_t (the supervision at time t.) Our algorithm is formally stated in Alg. 1 with its sample complexity bound below.

Theorem 3.1. *Suppose the sequential Rademacher complexity of the loss function (defined in Eq. (5)) is bounded by $\sqrt{R(\Theta) T \operatorname{polylog}(T)}$. The sample complexity of Alg. 1 (for finding an $(\epsilon, 6\sqrt{\zeta_{\mathrm{3rd}}\epsilon})$-approximate local maximum) is bounded by $\widetilde{\mathcal{O}}(C_1^4 R(\Theta) \max\left(\zeta_h^4 \epsilon^{-8}, \zeta_{\mathrm{3rd}}^2 \epsilon^{-6}\right))$.*

We present a proof sketch in Appendix D.1 . Proof of Theorem 3.1 is deferred to Appendix D.4. We can also boost the success probability by running Alg. 1 multiple times. In addition, we can prove that our algorithm enjoys a sublinear local regret. The theorem statement and proof is shown in Appendix D.5 and Appendix D.6 respectively.

3.1 Instantiations of Theorem 3.1

In the sequel we sketch some instantiations of our main theorem, whose proofs are deferred to Appendix D.7.

Linear bandit with finite model class. Consider a linear bandit problem with action set $\mathcal{A} = \{a \in \mathbb{R}^d : \|a\|_2 \leq 1\}$ and finite model class $\Theta \subset \{\theta \in \mathbb{R}^d : \|\theta\|_2 = 1\}$. Let $\eta(\theta, a) = \langle \theta, a \rangle$ be the reward. We deal with the constrained action set by using a surrogate loss $\tilde{\eta}(\theta, a) \triangleq \langle \theta, a \rangle - \frac{1}{2}\|a\|_2^2$ and apply Theorem D.3 with reward $\tilde{\eta}$. We claim that the sample complexity (of finding an ϵ-suboptimal action for $\eta(\theta^\star, a)$) is bounded by $\widetilde{\mathcal{O}}(\log|\Theta|\epsilon^{-8})$.[5] Note that the bound is *independent* of the dimension d. By contrast, the SquareCB algorithm [24] depends polynomially on d (see Theorem 7 of [24]). Zero-order optimization approach [20] in this case also gives a $\mathrm{poly}(d)$ regret bound.

Linear bandit with sparse or structured model vectors. We consider the deterministic linear bandit setting where the model class $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq s, \|\theta\|_2 = 1\}$ consists of all s-sparse vectors on the unit sphere. Similarly to finite hypothesis case, we claim that the sample complexity of Alg. 1 is $\widetilde{\mathcal{O}}(\log|\Theta|\epsilon^{-8}\mathrm{polylog}(d))$. The sample complexity of our algorithm only depends on the sparsity level s (up to logarithmic factors), whereas the Eluder dimension of sparse linear hypothesis is still $\Omega(d)$ (see Lemma D.4). Lattimore, Szepesvári [45] show a $\Omega(d)$ sample complexity lower bound for the sparse linear bandit problem with stochastic reward. But here we only consider a deterministic reward and continuous action.

Moreover, we can further extend the result to other linear bandit settings where θ has an additional structure. Suppose $\Theta = \{\theta = \phi(z) : z \in \mathbb{R}^s\}$ for some Lipschitz function ϕ. Then, a simlar approach gives sample complexity bound that only depends on s but not d (up to logarithmic factors). Our results can also be extended to non-linear bandits such as deterministic logistic bandits.

Two-layer neural nets. We consider the reward function given by two-layer neural networks with width m. For matrices $W_1 \in \mathbb{R}^{m \times d}$ and $W_2 \in \mathbb{R}^{1 \times m}$, let $\eta((W_1, W_2), a) = W_2 \sigma(W_1 a) - \frac{1}{2}\|a\|_2^2$ for some nonlinear link function $\sigma : \mathbb{R} \to [0, 1]$ with bounded derivatives up to the third order. Recall that the $(1, \infty)$-norm of $W_1^\top$ is defined by $\max_{i \in [m]} \sum_{j=1}^d |[W_1]_{i,j}|$. That is, the max 1-norm of the rows of W_1. Let the model hypothesis space be $\Theta = \{(W_1, W_2) : \|W_1^\top\|_{1,\infty} \leq 1, \|W_2\|_1 \leq 1\}$ and $\theta \triangleq (W_1, W_2)$. We claim that Alg .1 finds an $(\epsilon, 6\sqrt{\zeta_{3\mathrm{rd}}\epsilon})$-approximate local maximum in $\widetilde{\mathcal{O}}\big(\epsilon^{-8}\mathrm{polylog}(d)\big)$ steps. To the best of our knowledge, this is the first result analyzing nonlinear bandit with neural network parameterization. The result follows from analyzing the sequential Rademacher complexity for η, $\langle \nabla_a \eta, u \rangle$, and $\langle u, \nabla_a^2 \eta \cdot v \rangle$, and finally the resulting loss function ℓ. See Theorem D.6 in Section D.7 for details. We remark here that zero-order optimization in this case has a $\mathrm{poly}(d)$ sample complexity. In addition, if the second layer of the neural network W_2 contains all negative entries, and the activation function σ is monotone and convex, then $\eta((W_1, W_2), a)$ is *concave* in the action. (This is a special case of input convex neural networks [5].) In this case, Alg. 1 finds an ϵ-suboptimal action (see Theorem D.6).

4 Model-based Reinforcement Learning

In this section, we extend the results in Section 3 to model-based reinforcement learning with deterministic dynamics and reward function.

We can always view a model-based reinforcement learning problem with parameterized dynamics and policy as a nonlinear bandit problem in the following way. The policy parameter ψ corresponds to the action a in bandit, and the dynamics parameter θ corresponds to the model parameter θ in bandit. The expected total return $\eta(\theta, \psi) = \mathbb{E}_{s_1 \sim \mu_1} V_{T_\theta}^{\pi_\psi}(s_1)$ is the analogue of reward function in bandit. We intend to make the same regularity assumptions on η as in the bandit case (that is, Assumption 2.1) with a being replaced by ψ. However, when the policy is deterministic, the reward function η has Lipschitz constant with respect to ψ that is exponential in H (even if dynamics and policy are both deterministic with good Lipschitzness). This prohibits efficient optimization over policy parameters. Therefore we focus on *stochastic policies* in this section, for which we expect η and its derivatives to be Lipschitz with respect to ψ.

[5] Recall that in bandit literature, action a is an ϵ-approximate optimal action if $\eta(\theta^\star, a) \geq \eta(\theta^\star, a^\star) - \epsilon$.

Blindly treating RL as a bandit only utilizes the reward but not the state observations. In fact, one major reason why model-based methods are more sample efficient is that it supervises the learning of dynamics by state observations. To reason about the learning about local steps and the dynamics, we make the following additional Lipschitzness of value functions w.r.t to the states and Lipschitzness of policies w.r.t to its parameters, beyond those assumptions for the total reward $\eta(\theta, \psi)$ in Assumption 2.1.

Assumption 4.1. *We assume the following (analogous to Assumption 2.1) on the value function: $\forall \psi \in \Psi, \theta \in \Theta, s, s' \in \mathcal{S}$ we have $|V_\theta^\psi(s) - V_\theta^\psi(s')| \leq L_0 \|s - s'\|_2$; $\|\nabla_\psi V_\theta^\psi(s) - \nabla_\psi V_\theta^\psi(s')\|_2 \leq L_1 \|s - s'\|_2$; $\|\nabla_\psi^2 V_\theta^\psi(s) - \nabla_\psi^2 V_\theta^\psi(s')\|_{\mathrm{sp}} \leq L_2 \|s - s'\|_2$.*

Assumption 4.2. *We assume the following Lipschitzness assumptions on the stochastic policies parameterization π_ψ.*[6] *$\|\mathbb{E}_{a\sim\pi_\psi(\cdot|s)}[(\nabla_\psi \log \pi_\psi(a \mid s))(\nabla_\psi \log \pi_\psi(a \mid s))^\top]\|_{\mathrm{sp}} \leq \chi_g$; $\|\mathbb{E}_{a\sim\pi_\psi(\cdot|s)}[(\nabla_\psi \log \pi_\psi(a \mid s))^{\otimes 4}]\|_{\mathrm{sp}} \leq \chi_f$; $\|\mathbb{E}_{a\sim\pi_\psi(\cdot|s)}[(\nabla_\psi^2 \log \pi_\psi(a \mid s))(\nabla_\psi^2 \log \pi_\psi(a \mid s))^\top]\|_{\mathrm{sp}} \leq \chi_h$.*

Our results depends polynomially on the parameters $L_0, L_1, L_2, \chi_g, \chi_f$ and χ_h. To demonstrate that the Assumption 4.1 and 4.2 contain interesting RL problems with nonlinear models and stochastic policies, we give the following example where these parameters are all on the order of $O(1)$.

Example 4.3. *Let state space $\mathcal{S}$ be the unit ball in $\mathbb{R}^d$ and action space $\mathcal{A}$ be $\mathbb{R}^d$. The (deterministic) dynamics T is given by $T(s, a) = \mathrm{N}_\theta(s + a)$, where N is a nonlinear model parameterized by θ, e.g., a neural network. We assume that θ belongs to a finite hypothesis class Θ that satisfies $\|\mathrm{N}_\theta(s + a)\|_2 \leq 1$ for all $\theta \in \Theta, s \in \mathcal{S}, a \in \mathcal{A}$. Assume that the reward function $r(s, a)$ is L_r-Lipschitz w.r.t ℓ_2-norm, that is, satisfying $|r(s_1, a_1) - r(s_2, a_2)| \leq L_r(\|s_1 - s_2\|_2 + \|a_1 - a_2\|_2)$. We consider a family of stochastic Gaussian policies with the mean being linear in the state: $\pi_\psi(s) = \mathcal{N}(\psi s, \sigma^2 I)$, parameterized by $\psi \in \mathbb{R}^{d\times d}$ with $\|\psi\|_{\mathrm{op}} \leq 1$. We consider $\sigma \in (0, 1)$ as a small constant on the order of 1.*

In this setting, Assumption 2.1, 4.1, and 4.2 hold with all parameters $\zeta_g, \zeta_h, \zeta_{3\mathrm{rd}}, L_0, L_1, L_2, \chi_g, \chi_f$ and χ_h bounded by $\mathrm{poly}(\sigma, 1/\sigma, H, L_r)$.

Bounding the Lipschitz parameters is highly nontrivial and deferred to Appendix F.

We show that the difference of gradient and Hessian of the total reward can be upper-bounded by the difference of dynamics. Let $\tau_t = (s_1, a_1, \cdots, s_H, a_H)$ be a trajectory sampled from policy π_{ψ_t} under the real dynamics $T_{\theta^\star}$. Similarly to [77], when the value function is Lipschitz, we upper bound $\Delta_{t,1} = |\eta(\theta_t, \psi_t) - \eta(\theta^\star, \psi_t)|$ by the one-step model prediction errors. Similarly, we can upper bound the gradient and Hessian errors by model prediction errors. As a result, the loss function simply can be set to

$$\ell((\tau_t, \tau_t'); \theta) = \sum_{(s_h, a_h)\in\tau_t} \|T_\theta(s_h, a_h) - T_{\theta^\star}(s_h, a_h)\|_2^2 + \sum_{(s_h', a_h')\in\tau_t'} \|T_\theta(s_h', a_h') - T_{\theta^\star}(s_h', a_h')\|_2^2 \quad (6)$$

for two trajectories τ, τ' sampled from policy π_{ψ_t} and $\pi_{\psi_{t-1}}$ respectively. Compared to Alg. 1, the loss function is here simpler without relying on finite difference techniques to query gradients projections. Our algorithm for RL is analogous to Alg. 1 by using the loss function in Eq. (6). Our algorithm is presented in Alg. 2 in Appendix E. Main theorem for Alg. 2 is shown below.

Theorem 4.4. *Let $c_1 = HL_0^2(4H^2\chi_h + 4H^4\chi_f + 2H^2\chi_g + 1) + HL_1^2(8H^2\chi_g + 2) + 4HL_2^2$ and $C_1 = 2 + \frac{\zeta_g}{\zeta_h}$. Suppose the sequential Rademacher complexity of the loss function (defined in Eq. (6)) is bounded by $\sqrt{R(\Theta)T\mathrm{polylog}(T)}$. The sample complexity of Alg. 2 (for finding an $(\epsilon, 6\sqrt{\zeta_{3\mathrm{rd}}\epsilon})$-approximate local maximum) is bounded by $\widetilde{\mathcal{O}}(c_1^2 C_1^4 R(\Theta) \max\left(\zeta_h^4\epsilon^{-8}, \zeta_{3\mathrm{rd}}^2\epsilon^{-6}\right))$.*

Instantiation of Theorem 4.4 on Example 4.3. Applying Theorem 4.4 to the Example 4.3 with $\sigma = \Theta(1)$ we get the sample complexity upper bound $\widetilde{\mathcal{O}}\big(\mathrm{poly}(\sigma, 1/\sigma, H, L_r)\log|\Theta|\epsilon^{-8}\big)$.

Comparison with policy gradient. To the best of our knowledge, the best analysis for policy gradient [73] shows convergence to a local maximum with a sample complexity that depends polynomially on $\|\nabla_\psi \log \pi_\psi(a \mid s)\|_2$ [4]. For the instance in in Example 4.3, this translates to a sample

[6] Recall that the injective norm of a k-th order tensor $A \in \mathbb{R}^{d^{\otimes k}}$ is defined as $\|A^{\otimes k}\|_{\mathrm{sp}} = \sup_{u\in S^{d-1}} \langle A, u^{\otimes k}\rangle$.

complexity guarantee on the order of $\sqrt{d}/\sigma$. In contrast, our sample complexity is independent of the dimension d. Instead, our bound depends on the complexity of the model family Θ which could be much smaller than the ambient dimension—this demonstrates that we leverage the model extrapolation.

5 Lower Bounds

We prove several lower bounds to show (a) the hardness of finding global maxima, and (b) the inefficiency of using optimism in nonlinear bandit.

Hardness of global optimality. We show it statistically intractable to find the global optimal policy when the function class is chosen to be the neural networks with ReLU activation.

Theorem 5.1. *When the function class is chosen to be one-layer neural networks with ReLU activation, the minimax sample complexity is* $\Omega(\varepsilon^{-(d-2)})$.

We can also prove that the eluder dimension of the constructed reward function class is exponential.

Theorem 5.2. *The* ε*-eluder dimension of one-layer neural networks is at least* $\Omega(\varepsilon^{-(d-1)})$.

This result is concurrently established by Li et al. [49, Theorem 8]. The proofs of both theorems are deferred to Appendices C.1 and C.2, respectively.

Inefficiency caused by optimism in nonlinear models. The next theorem states that the UCB algorithm that uses optimism-in-face-of-uncertainty principle can overly explore in the action space, even if the ground-truth is simple. Proof of the theorem is deferred to Appendix C.3.

Theorem 5.3. *Consider the case where the ground-truth reward function is linear:* $\langle \theta^\star, a \rangle$ *and the action set is* $a \in S^{d-1}$*. If the hypothesis is chosen to be two-layer neural network with width* d*, UCB algorithm with tightest upper confidence bound suffers exponential sample complexity .*

6 Additional Related Work

There are several provable efficient algorithms without optimism for contextual bandits. Dudik et al. [21], Agarwal et al. [1] instantiate mirror descent for contextual bandits, and the regret depend polynomially on the number of actions. Foster, Rakhlin [24] and Simchi-Levi, Xu [66] exploit the exploration probability inversely depends on the empirical gap. The SquareCB algorithm [24] also extends to infinite actions with linear structure, but the regret depends on the action dimension. Recently, Foster et al. [25] prove an instance-dependent regret bound for contextual bandit.

Deterministic nonlinear bandits can also be formulated as zero-order optimization without noise (see Duchi et al. [20], Liu et al. [53] and references therein), where the reward is assumed to be any 1-Lipschitz function. In contrast, our algorithm exploits the knowledge of the reward function parameterization and achieves an action-dimension-free sample complexity. For stochastic nonlinear bandit, Filippi et al. [23] consider generalized linear model. Valko et al. [69], Zhou et al. [79] focus on rewards in a RKHS or NTK. Yang et al. [76] extends this approach to reinforcement learning.

Another line of research focuses on solving reinforcement learning by optimization on the policy space. Agarwal et al. [4] prove that natural policy gradient can solve tabular MDPs efficiently. Cai et al. [8] incorporate exploration bonus in proximal policy optimization algorithm and achieves polynomial regret in linear MDP setting.

Beyond linear function approximations, there are also extensive studies on various settings that allow efficient algorithms. For example, rich observation MDPs [44, 11, 18, 56], state aggregation [16, 50], Bellman rank [38, 13] and others [17, 52, 58, 78, 42].

7 Conclusion

In this paper, we design new algorithms whose sample complexity toward local maxima are bounded by the sequential Rademacher complexity of particular loss functions. By rearranging the priorities of exploration versus exploitation, our algorithms avoid over-aggressive explorations caused by the optimism in the face of uncertainty principle, and hence apply to nonlinear models and dynamics.

References

[1] *Agarwal Alekh, Hsu Daniel, Kale Satyen, Langford John, Li Lihong, Schapire Robert*. Taming the monster: A fast and simple algorithm for contextual bandits // International Conference on Machine Learning. 2014. 1638–1646.

[2] *Agarwal Alekh, Jiang Nan, Kakade Sham M*. Reinforcement learning: Theory and algorithms // CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep. 2019.

[3] *Agarwal Alekh, Kakade Sham, Krishnamurthy Akshay, Sun Wen*. FLAMBE: Structural complexity and representation learning of low rank MDPs // arXiv preprint arXiv:2006.10814. 2020.

[4] *Agarwal Alekh, Kakade Sham M, Lee Jason D, Mahajan Gaurav*. Optimality and Approximation with Policy Gradient Methods in Markov Decision Processes // Proceedings of Thirty Third Conference on Learning Theory. 125. 09–12 Jul 2020. 64–66. (Proceedings of Machine Learning Research).

[5] *Amos Brandon, Xu Lei, Kolter J Zico*. Input convex neural networks // International Conference on Machine Learning. 2017. 146–155.

[6] *Ayoub Alex, Jia Zeyu, Szepesvari Csaba, Wang Mengdi, Yang Lin F*. Model-Based Reinforcement Learning with Value-Targeted Regression // Proceedings of the 37th International Conference on Machine Learning. 2020.

[7] *Berner Christopher, Brockman Greg, Chan Brooke, Cheung Vicki, Debiak Przemyslaw, Dennison Christy, Farhi David, Fischer Quirin, Hashme Shariq, Hesse Chris, others* . Dota 2 with large scale deep reinforcement learning // arXiv preprint arXiv:1912.06680. 2019.

[8] *Cai Qi, Yang Zhuoran, Jin Chi, Wang Zhaoran*. Provably efficient exploration in policy optimization // International Conference on Machine Learning. 2020. 1283–1294.

[9] *Carpentier Alexandra, Munos Rémi*. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit // Artificial Intelligence and Statistics. 2012. 190–198.

[10] *Clavera Ignasi, Fu Yao, Abbeel Pieter*. Model-Augmented Actor-Critic: Backpropagating through Paths // International Conference on Learning Representations. 2019.

[11] *Dann Christoph, Jiang Nan, Krishnamurthy Akshay, Agarwal Alekh, Langford John, Schapire Robert E.* On Oracle-Efficient PAC-RL with Rich Observations // Advances in Neural Information Processing Systems. 2018.

[12] *Dong Kefan, Luo Yuping, Yu Tianhe, Finn Chelsea, Ma Tengyu*. On the expressivity of neural networks for deep reinforcement learning // International Conference on Machine Learning. 2020. 2627–2637.

[13] *Dong Kefan, Peng Jian, Wang Yining, Zhou Yuan*. Root-n-Regret for Learning in Markov Decision Processes with Function Approximation and Low Bellman Rank // Conference on Learning Theory. 2020. 1554–1557.

[14] *Dong Shi, Ma Tengyu, Van Roy Benjamin*. On the Performance of Thompson Sampling on Logistic Bandits // Conference on Learning Theory. 2019. 1158–1160.

[15] *Dong Shi, Van Roy Benjamin*. An information-theoretic analysis for thompson sampling with many actions // arXiv preprint arXiv:1805.11845. 2018.

[16] *Dong Shi, Van Roy Benjamin, Zhou Zhengyuan*. Provably efficient reinforcement learning with aggregated states // arXiv preprint arXiv:1912.06366. 2019.

[17] *Du Simon S, Kakade Sham M, Lee Jason D, Lovett Shachar, Mahajan Gaurav, Sun Wen, Wang Ruosong*. Bilinear Classes: A Structural Framework for Provable Generalization in RL // arXiv preprint arXiv:2103.10897. 2021.

[18] *Du Simon S, Krishnamurthy Akshay, Jiang Nan, Agarwal Alekh, Dudik Miroslav, Langford John*. Provably efficient RL with Rich Observations via Latent State Decoding // International Conference on Machine Learning. 2019. 1665–1674.

[19] *Du Simon S, Luo Yuping, Wang Ruosong, Zhang Hanrui*. Provably efficient Q-learning with function approximation via distribution shift error checking oracle // Advances in Neural Information Processing Systems. 2019. 8058–8068.

[20] *Duchi John C, Jordan Michael I, Wainwright Martin J, Wibisono Andre*. Optimal rates for zero-order convex optimization: The power of two function evaluations // IEEE Transactions on Information Theory. 2015. 61, 5. 2788–2806.

[21] *Dudik Miroslav, Hsu Daniel, Kale Satyen, Karampatziakis Nikos, Langford John, Reyzin Lev, Zhang Tong*. Efficient optimal learning for contextual bandits // arXiv preprint arXiv:1106.2369. 2011.

[22] *Faury Louis, Abeille Marc, Calauzènes Clément, Fercoq Olivier*. Improved optimistic algorithms for logistic bandits // International Conference on Machine Learning. 2020. 3052–3060.

[23] *Filippi Sarah, Cappé Olivier, Garivier Aurélien, Szepesvári Csaba*. Parametric bandits: the Generalized Linear case // Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 1. 2010. 586–594.

[24] *Foster Dylan, Rakhlin Alexander*. Beyond UCB: Optimal and Efficient Contextual Bandits with Regression Oracles // Proceedings of the 37th International Conference on Machine Learning. 119. 13–18 Jul 2020. 3199–3210. (Proceedings of Machine Learning Research).

[25] *Foster Dylan J, Rakhlin Alexander, Simchi-Levi David, Xu Yunzong*. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective // arXiv preprint arXiv:2010.03104. 2020.

[26] *Ge Rong, Huang Furong, Jin Chi, Yuan Yang*. Escaping from saddle points—online stochastic gradient for tensor decomposition // Conference on Learning Theory. 2015. 797–842.

[27] *Ge Rong, Jin Chi, Zheng Yi*. No Spurious Local Minima in Nonconvex Low Rank Problems: A Unified Geometric Analysis // arXiv preprint arXiv:1704.00708. 2017.

[28] *Ge Rong, Lee Jason D, Ma Tengyu*. Matrix completion has no spurious local minimum // Advances in Neural Information Processing Systems. 2016. 2973–2981.

[29] *Ge Rong, Ma Tengyu*. On the optimization landscape of tensor decompositions // Mathematical Programming. 2020. 1–47.

[30] *Hafner Danijar, Lillicrap Timothy, Ba Jimmy, Norouzi Mohammad*. Dream to Control: Learning Behaviors by Latent Imagination // International Conference on Learning Representations. 2019.

[31] *Hafner Danijar, Lillicrap Timothy, Fischer Ian, Villegas Ruben, Ha David, Lee Honglak, Davidson James*. Learning latent dynamics for planning from pixels // International Conference on Machine Learning. 2019. 2555–2565.

[32] *Hao Botao, Lattimore Tor, Szepesvári Csaba, Wang Mengdi*. Online Sparse Reinforcement Learning // International Conference on Artificial Intelligence and Statistics. 2021. 316–324.

[33] *Hao Botao, Lattimore Tor, Wang Mengdi*. High-Dimensional Sparse Linear Bandits // arXiv preprint arXiv:2011.04020. 2020.

[34] *Hazan Elad, Levy Kfir, Shalev-Shwartz Shai*. Beyond convexity: Stochastic quasi-convex optimization // Advances in Neural Information Processing Systems. 2015. 1594–1602.

[35] *Hazan Elad, Singh Karan, Zhang Cyril*. Efficient regret minimization in non-convex games // International Conference on Machine Learning. 2017. 1433–1441.

[36] *Hsu Daniel, Kakade Sham, Zhang Tong, others* . A tail inequality for quadratic forms of subgaussian random vectors // Electronic Communications in Probability. 2012. 17.

[37] *Janner Michael, Fu Justin, Zhang Marvin, Levine Sergey*. When to trust your model: Model-based policy optimization // arXiv preprint arXiv:1906.08253. 2019.

[38] *Jiang Nan, Krishnamurthy Akshay, Agarwal Alekh, Langford John, Schapire Robert E*. Contextual Decision Processes with Low Bellman Rank are PAC-Learnable // arXiv preprint arXiv:1610.09512. 2016.

[39] *Jin Chi, Allen-Zhu Zeyuan, Bubeck Sebastien, Jordan Michael I*. Is Q-learning provably efficient? // Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018. 4868–4878.

[40] *Jin Chi, Yang Zhuoran, Wang Zhaoran, Jordan Michael I*. Provably efficient reinforcement learning with linear function approximation // Conference on Learning Theory. 2020. 2137–2143.

[41] *Johnson Rie, Zhang Tong*. Accelerating stochastic gradient descent using predictive variance reduction // Advances in Neural Information Processing Systems. 2013. 315–323.

[42] *Kakade Sham, Krishnamurthy Akshay, Lowrey Kendall, Ohnishi Motoya, Sun Wen*. Information theoretic regret bounds for online nonlinear control // arXiv preprint arXiv:2006.12466. 2020.

[43] *Kakade Sham M, Kanade Varun, Shamir Ohad, Kalai Adam*. Efficient learning of generalized linear and single index models with isotonic regression // Advances in Neural Information Processing Systems. 2011. 927–935.

[44] *Krishnamurthy Akshay, Agarwal Alekh, Langford John*. PAC reinforcement learning with rich observations // Advances in Neural Information Processing Systems. 2016. 1840–1848.

[45] *Lattimore Tor, Szepesvári Csaba*. Bandit algorithms. 2020.

[46] *Laurent Beatrice, Massart Pascal*. Adaptive estimation of a quadratic functional by model selection // Annals of Statistics. 2000. 1302–1338.

[47] *Lee Jason D, Simchowitz Max, Jordan Michael I, Recht Benjamin*. Gradient descent only converges to minimizers // Conference on learning theory. 2016. 1246–1257.

[48] *Levine Sergey, Finn Chelsea, Darrell Trevor, Abbeel Pieter*. End-to-end training of deep visuomotor policies // The Journal of Machine Learning Research. 2016. 17, 1. 1334–1373.

[49] *Li Gene, Kamath Pritish, Foster Dylan J., Srebro Nathan*. Eluder Dimension and Generalized Rank. 2021.

[50] *Li Lihong*. A unifying framework for computational reinforcement learning theory. 2009.

[51] *Li Lihong, Lu Yu, Zhou Dengyong*. Provably optimal algorithms for generalized linear contextual bandits // International Conference on Machine Learning. 2017. 2071–2080.

[52] *Littman Michael L, Sutton Richard S, Singh Satinder P*. Predictive representations of state. // NIPS. 14, 1555. 2001. 30.

[53] *Liu Sijia, Chen Pin-Yu, Kailkhura Bhavya, Zhang Gaoyuan, Hero III Alfred O, Varshney Pramod K*. A Primer on Zeroth-Order Optimization in Signal Processing and Machine Learning: Principals, Recent Advances, and Applications // IEEE Signal Processing Magazine. 2020. 37, 5. 43–54.

[54] *Luo Yuping, Xu Huazhe, Li Yuanzhi, Tian Yuandong, Darrell Trevor, Ma Tengyu*. Algorithmic Framework for Model-based Deep Reinforcement Learning with Theoretical Guarantees // International Conference on Learning Representations. 2019.

[55] *Mahmud Mufti, Kaiser Mohammed Shamim, Hussain Amir, Vassanelli Stefano*. Applications of deep learning and reinforcement learning to biological data // IEEE transactions on neural networks and learning systems. 2018. 29, 6. 2063–2079.

[56] *Misra Dipendra, Henaff Mikael, Krishnamurthy Akshay, Langford John*. Kinematic state abstraction and provably efficient rich-observation reinforcement learning // International conference on machine learning. 2020. 6961–6971.

[57] *Modi Aditya, Jiang Nan, Tewari Ambuj, Singh Satinder*. Sample complexity of reinforcement learning using linearly combined model ensembles // International Conference on Artificial Intelligence and Statistics. 2020. 2010–2020.

[58] *Munos Rémi*. Error bounds for approximate value iteration // Proceedings of the National Conference on Artificial Intelligence. 2005. 1006.

[59] *Osband Ian, Roy Benjamin Van*. Model-based reinforcement learning and the eluder dimension // Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1. 2014. 1466–1474.

[60] *Rakhlin Alexander, Sridharan Karthik, Tewari Ambuj*. Online learning: stochastic, constrained, and smoothed adversaries // Proceedings of the 24th International Conference on Neural Information Processing Systems. 2011. 1764–1772.

[61] *Rakhlin Alexander, Sridharan Karthik, Tewari Ambuj*. Online Learning via Sequential Complexities // Journal of Machine Learning Research. 2015. 16, 6. 155–186.

[62] *Rakhlin Alexander, Sridharan Karthik, Tewari Ambuj*. Sequential complexities and uniform martingale laws of large numbers // Probability Theory and Related Fields. 2015. 161, 1-2. 111–153.

[63] *Russo Daniel, Roy Benjamin Van*. Eluder dimension and the sample complexity of optimistic exploration // Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2. 2013. 2256–2264.

[64] *Shariff Roshan, Szepesvári Csaba*. Efficient planning in large MDPs with weak linear function approximation // arXiv preprint arXiv:2007.06184. 2020.

[65] *Silver David, Schrittwieser Julian, Simonyan Karen, Antonoglou Ioannis, Huang Aja, Guez Arthur, Hubert Thomas, Baker Lucas, Lai Matthew, Bolton Adrian, others* . Mastering the game of Go without human knowledge // Nature. 2017. 550, 7676. 354.

[66] *Simchi-Levi David, Xu Yunzong*. Bypassing the Monster: A Faster and Simpler Optimal Algorithm for Contextual Bandits under Realizability // arXiv preprint arXiv:2003.12699. 2020.

[67] *Sun Wen, Jiang Nan, Krishnamurthy Akshay, Agarwal Alekh, Langford John*. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches // Conference on Learning Theory. 2019. 2898–2933.

[68] Reinforcement learning: An introduction. // . 2011.

[69] *Valko Michal, Korda Nathan, Munos Rémi, Flaounas Ilias, Cristianini Nello*. Finite-time analysis of kernelised contextual bandits // Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence. 2013. 654–663.

[70] *Wang Ruosong, Salakhutdinov Ruslan, Yang Lin F*. Provably Efficient Reinforcement Learning with General Value Function Approximation // Advances in Neural Information Processing Systems. 2020.

[71] *Wang Yining, Chen Yi, Fang Ethan X, Wang Zhaoran, Li Runze*. Nearly Dimension-Independent Sparse Linear Bandit over Small Action Spaces via Best Subset Selection // arXiv preprint arXiv:2009.02003. 2020.

[72] *Wang Yining, Wang Ruosong, Du Simon S, Krishnamurthy Akshay*. Optimism in Reinforcement Learning with Generalized Linear Function Approximation // arXiv preprint arXiv:1912.04136. 2019.

[73] *Williams Ronald J*. Simple statistical gradient-following algorithms for connectionist reinforcement learning // Machine learning. 1992. 8, 3-4. 229–256.

[74] *Xu Bing, Wang Naiyan, Chen Tianqi, Li Mu*. Empirical Evaluation of Rectified Activations in Convolutional Network. 2015.

[75] *Yang Lin, Wang Mengdi*. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound // International Conference on Machine Learning. 2020. 10746–10756.

[76] *Yang Zhuoran, Jin Chi, Wang Zhaoran, Wang Mengdi, Jordan Michael I*. Bridging exploration and general function approximation in reinforcement learning: Provably efficient kernel and neural value iterations // arXiv preprint arXiv:2011.04622. 2020.

[77] *Yu Tianhe, Thomas Garrett, Yu Lantao, Ermon Stefano, Zou James, Levine Sergey, Finn Chelsea, Ma Tengyu*. MOPO: Model-based Offline Policy Optimization // arXiv preprint arXiv:2005.13239. 2020.

[78] *Zanette Andrea, Lazaric Alessandro, Kochenderfer Mykel, Brunskill Emma*. Learning Near Optimal Policies with Low Inherent Bellman Error // International Conference on Machine Learning. 2020.

[79] *Zhou Dongruo, Li Lihong, Gu Quanquan*. Neural contextual bandits with UCB-based exploration // International Conference on Machine Learning. 2020. 11492–11502.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] We describe the limitations in Appendix B.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss in Appendix A.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] Our assumptions are stated in Section 2.

(b) Did you include complete proofs of all theoretical results? [Yes] Proofs are given in Appendix C-G.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]