# Provably Efficient Reinforcement Learning with Linear Function Approximation

**Chi Jin** CHIJ@PRINCETON.EDU
*Princeton University*

**Zhuoran Yang** ZY6@PRINCETON.EDU
*Princeton University*

**Zhaoran Wang** ZHAORAN.WANG@NORTHWESTERN.EDU
*Northwestern University*

**Michael I. Jordan** JORDAN@CS.BERKELEY.EDU
*University of California, Berkeley*

**Editors:** Jacob Abernethy and Shivani Agarwal

## Abstract

[1] Modern Reinforcement Learning (RL) is commonly applied to practical problems with an enormous number of states, where *function approximation* must be deployed to approximate either the value function or the policy. The introduction of function approximation raises a fundamental set of challenges involving computational and statistical efficiency, especially given the need to manage the exploration/exploitation tradeoff. As a result, a core RL question remains open: how can we design provably efficient RL algorithms that incorporate function approximation? This question persists even in a basic setting with linear dynamics and linear rewards, for which only linear function approximation is needed.

This paper presents the first provable RL algorithm with both polynomial runtime and polynomial sample complexity in this linear setting, without requiring a "simulator" or additional assumptions. Concretely, we prove that an optimistic modification of Least-Squares Value Iteration (LSVI)—a classical algorithm frequently studied in the linear setting—achieves $\widetilde{\mathcal{O}}(\sqrt{d^3H^3T})$ regret, where $d$ is the ambient dimension of feature space, $H$ is the length of each episode, and $T$ is the total number of steps. Importantly, such regret is independent of the number of states and actions.

**Keywords:** Markov Decision Process, Reinforcement Learning, Value Iteration.

## 1. Introduction

Reinforcement Learning (RL) is a control-theoretic problem in which an agent tries to maximize its expected cumulative reward by interacting with an unknown environment over time (Sutton and Barto, 2011). Modern RL commonly engages practical problems with an enormous number of states, where *function approximation* must be deployed to approximate the *(action-)value function*—the expected cumulative reward starting from a state-action pair—or the *policy*—the mapping from a state to its subsequent action. Function approximation, especially based on deep neural networks, lies at the heart of the recent practical successes of RL in domains such as Atari games

1. Extended abstract. Full version appears as [arXiv:1907.05388, v3].

(Mnih et al., 2013), Go (Silver et al., 2016), robotics (Kober and Peters, 2012), and dialogue systems (Li et al., 2016). Moreover, deep neural networks serve as essential components of generic deep RL algorithms, including Deep Q-Network (DQN) (Mnih et al., 2013), Asynchronous Advantage Actor-Critic (A3C) (Mnih et al., 2016), and Trust Region Policy Optimization (TRPO) (Schulman et al., 2015).

Despite the empirical successes of function approximation in RL, most existing theoretical guarantees apply only to *tabular* RL (see, e.g., Jaksch et al., 2010; Osband et al., 2014; Azar et al., 2017; Jin et al., 2018), in which the states and actions are discrete, and the value function is represented by a table. Due to the curse of dimensionality, only relatively small problems can be tackled by tabular RL. Thus, researchers have turned to function approximation (see, e.g., Sutton, 1988; Bradtke and Barto, 1996; Tsitsiklis and Van Roy, 1997), in theory and in practice. While function approximation greatly expands the potential reach of RL, particularly via deep RL architectures, it raises a number of fundamental theoretical challenges. For example, while the effective state and action spaces can be much larger when function approximation is used, the neighborhoods of most states are not visited even once during a set of learning episodes, which makes it difficult to obtain reliable estimates of value functions (see, e.g., Sutton and Barto, 2011; Szepesvári, 2010; Lattimore and Szepesvári, 2018). To cope with this challenge, relatively simple function classes, including linear function classes, are often used. This introduces, however, a bias, even in the limit of infinite training data, given that the optimal value function and policy may not be linear (see, e.g., Baird, 1995; Boyan and Moore, 1995; Tsitsiklis and Van Roy, 1997). Thus, both in theory and in practice, the design of RL systems must cope with fundamental statistical problems of sparsity and misspecification, all in the context of a dynamical system. Moreover, a core distinguishing feature of RL is that it requires addressing the tradeoff between exploration and exploitation. Addressing this tradeoff algorithmically requires exactly the kinds of statistical estimates that are challenging to obtain in the RL setting due to sparsity, misspecification, and dynamics. Thus the following fundamental question remains open:

**Is it possible to design provably efficient RL algorithms in the function approximation setting?**

By "efficient" we mean efficient in both runtime and sample complexity—the runtime and the sample complexity should not depend on the number of states, but should depend instead on an intrinsic complexity measure of the function class.

Several recent attempts have been made to attack this fundamental problem. However, they either require the access to a "simulator" (Yang and Wang, 2019a) which alleviates the difficulty of exploration, or assume the transition dynamics to be deterministic (Wen and Van Roy, 2013, 2017), to have a low variance (Du et al., 2019), or are parametrizable by a relatively small matrix (Yang and Wang, 2019b), which alleviates the difficulty in estimating the transition dynamics (see Section 1.1 for more details).

Focusing on a linear setting in which the transition dynamics and reward function are assumed to be linear, we present the first algorithm that is provably efficient in both runtime and sample complexity, without requiring additional oracles or stronger assumptions. Concretely, in the general setting of an episodic Markov Decision Process (MDP), we prove that an optimistic version of Least-Squares Value Iteration (LSVI) (Bradtke and Barto, 1996; Osband et al., 2014)—a classical algorithm frequently studied in the linear setting—achieves $\widetilde{\mathcal{O}}(\sqrt{d^3H^3T})$ regret, where $d$ is the ambient dimension of feature space, $H$ is the length of each episode, $T$ is the total number of steps,

and $\widetilde{\mathcal{O}}(\cdot)$ hides only absolute constant and poly-logarithmic factors. Importantly, such regret is independent of $S$ and $A$—the number of states and actions. Our algorithm runs in $\mathcal{O}(d^2AKT)$ time and $\mathcal{O}(d^2H + dAT)$ space, which are again independent of $S$ and thus efficient in practice.

### 1.1. Related Work

**Tabular RL:** Tabular RL is well studied in both model-based (Jaksch et al., 2010; Osband et al., 2014; Azar et al., 2017; Dann et al., 2017) and model-free settings (Strehl et al., 2006; Jin et al., 2018). See also (Koenig and Simmons, 1993; Azar et al., 2011, 2012; Lattimore and Hutter, 2012; Sidford et al., 2018; Wainwright, 2019) for a simplified setting with access to a "simulator" (also called a generative model), which is a strong oracle that allows the algorithm to query arbitrary state-action pairs and return the reward and the next state. The "simulator" significantly alleviates the difficulty of exploration, since a naive exploration strategy which queries all state-action pairs uniformly at random already leads to the most efficient algorithm for finding an optimal policy (Azar et al., 2012).

In the episodic setting with nonstationary dynamics and no "simulators," the best regrets achieved by existing model-based and model-free algorithms are $\widetilde{\mathcal{O}}(\sqrt{H^2SAT})$ (Azar et al., 2017) and $\widetilde{\mathcal{O}}(\sqrt{H^3SAT})$ (Jin et al., 2018), respectively, both of which (nearly) attain the minimax lower bound $\Omega(\sqrt{H^2SAT})$ (Jaksch et al., 2010; Osband and Van Roy, 2016; Jin et al., 2018). Here $S$ and $A$ denote the numbers of states and actions, respectively. Although these algorithms are (nearly) minimax-optimal, they can not cope with large state spaces, as their regret scales linearly in $\sqrt{S}$, where $S$ is often exponentially large in practice (see, e.g., Mnih et al., 2013; Silver et al., 2016; Kober and Peters, 2012; Li et al., 2016). Moreover, the minimax lower bound suggests that, information-theoretically, a large state space cannot be handled efficiently unless further problem-specific structure is exploited. Compared with this line of work, in the current paper we exploit the linear structure of the reward and transition functions and show that the regret of optimistic LSVI scales polynomially in the ambient dimension $d$ rather than the number of states $S$.

**Linear Bandits:** To enable function approximation, another line of related work studies stochastic linear bandits or stochastic linear contextual bandits (see, e.g., Auer, 2002; Dani et al., 2008; Li et al., 2010; Rusmevichientong and Tsitsiklis, 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011), which is a special case of the linear MDP studied in this paper (Assumption **??**) with the episode length $H$ set equal to one. See Bubeck and Cesa-Bianchi (2012); Lattimore and Szepesvári (2018) and the references therein for a detailed survey. The best regrets achieved by existing algorithms are $\widetilde{\mathcal{O}}(d\sqrt{T})$ for linear bandits (Abbasi-Yadkori et al., 2011) and $\widetilde{\mathcal{O}}(\sqrt{dT})$ for linear contextual bandits (Auer, 2002; Chu et al., 2011), both of which scale polynomially in the ambient dimension $d$. We note, however, that while an MDP has state transition, linear bandits do not. This temporal structure captures the fundamental difference in their difficulties of exploration: a naive adaptation of existing linear bandit algorithms to the linear MDP setting yields a regret exponential in $H$—the length of each episode.

**RL with Function Approximation:** In the setting of linear function approximation, there is a long line of classical work on the design of algorithms, but this work does not provide polynomial sample efficiency guarantees (see, e.g., Bradtke and Barto, 1996; Melo and Ribeiro, 2007; Sutton and Barto, 2011; Osband et al., 2014; Azizzadenesheli et al., 2018). Recently, Yang and Wang (2019a) revisited the setting of linear transitions and rewards (Bradtke and Barto, 1996; Melo and

Ribeiro, 2007) (Assumption **??**), and presented a sample-efficient algorithm assuming the access to a "simulator". Similar to the case of tabular setting, the "simulator" greatly alleviates the difficulty of exploration. We also note that their very recent work (Yang and Wang, 2019b), developed independently of the current paper, provides sample efficiency guarantees for exploration in the linear MDP setting. Compared with the current paper, Yang and Wang (2019b) differs in that requires one additional key assumption—that the transition model can be parameterized by a relatively small matrix. This additional assumption reduces the number of free parameters in the transition model from potentially being infinite (for the case with an infinite number of states) to small and finite, and thus mitigates the challenges in estimating the transition model. As a result, their algorithm and main mechanism are based on estimating the unknown matrix, which differs from our approach. Finally, in a broader context, without the assumption of a linear MDP, sample efficiency guarantees have been established for RL under other assumptions, such as that the transition dynamics are fully deterministic (Wen and Van Roy, 2013, 2017), or have low variances (Du et al., 2019). These assumptions can be potentially restrictive in practice, and may not hold even in the tabular setting. In contrast, our results directly cover the standard tabular case with no extra assumptions.

In the setting of general function approximation, Jiang et al. (2017) present a generic algorithm Olive, which enjoys sample efficiency if a complexity measure that they refer to as "Bellman rank" is small. It can be shown that Bellman rank is at most $d$ under Assumption **??**, and thus Olive is sample efficient in our setting. In contrast to our results, Olive is not computationally efficient in general and it does not provide a $\sqrt{T}$ regret bound. Meanwhile, a recent line of work (Zhu and Dunson, 2019; Wang et al., 2019) studies a nonparametric setting with Hölder smooth reward and transition model. The sample complexities provided therein are exponential in dimensionality in the worst case.

## References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Mohammad Gheshlaghi Azar, Remi Munos, M Ghavamzadaeh, and Hilbert J Kappen. Speedy Q-learning. In *Advances in Neural Information Processing Systems*, 2011.

Mohammad Gheshlaghi Azar, Rémi Munos, and Bert Kappen. On the sample complexity of reinforcement learning with a generative model. *arXiv preprint arXiv:1206.6461*, 2012.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272, 2017.

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning*, pages 30–37, 1995.

Justin A Boyan and Andrew W Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems*, pages 369–376, 1995.

Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

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

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5713–5723, 2017.

Simon S Du, Yuping Luo, Ruosong Wang, and Hanrui Zhang. Provably efficient Q-learning with function approximation via distribution shift error checking oracle. *arXiv preprint arXiv:1906.06321*, 2019.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4):1563–1600, 2010.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713. JMLR. org, 2017.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

Jens Kober and Jan Peters. Reinforcement learning in robotics: A survey. In *Reinforcement Learning*, pages 579–610. Springer, 2012.

Sven Koenig and Reid G Simmons. Complexity analysis of real-time reinforcement learning. In *Association for the Advancement of Artificial Intelligence*, pages 99–107, 1993.

Tor Lattimore and Marcus Hutter. PAC bounds for discounted MDPs. In *International Conference on Algorithmic Learning Theory*, pages 320–334, 2012.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, 2018.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pages 661–670, 2010.

Francisco S Melo and M Isabel Ribeiro. Q-learning with linear function approximation. In *International Conference on Computational Learning Theory*, pages 308–322. Springer, 2007.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

Aaron Sidford, Mengdi Wang, Xian Wu, and Yinyu Ye. Variance reduced value iteration and faster algorithms for solving Markov decision processes. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 770–787, 2018.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. PAC model-free reinforcement learning. In *International Conference on Machine Learning*, pages 881–888, 2006.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2011.

Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, 2010.

John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1075–1081, 1997.

Martin J Wainwright. Variance-reduced Q-learning is minimax optimal. *arXiv preprint arXiv:1906.04697*, 2019.

Tianyu Wang, Weicheng Ye, Dawei Geng, and Cynthia Rudin. Towards practical Lipschitz stochastic bandits. *arXiv preprint arXiv:1901.09277*, 2019.

Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems*, pages 3021–3029, 2013.

Zheng Wen and Benjamin Van Roy. Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research*, 42(3):762–782, 2017.

Lin Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004, 2019a.

Lin F Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019b.

Xu Zhu and David B Dunson. Lipschitz bandit optimization with improved efficiency. *arXiv preprint arXiv:1904.11131*, 2019.