Sample Efficient Reinforcement Learning In Continuous State Spaces: A Perspective Beyond Linearity

Dhruv Malik [1] **Aldo Pacchiano** [2] **Vishwak Srinivasan** [1] **Yuanzhi Li** [1]

Abstract

Reinforcement learning (RL) is empirically successful in complex nonlinear Markov decision processes (MDPs) with continuous state spaces. By contrast, the majority of theoretical RL literature requires the MDP to satisfy some form of linear structure, in order to guarantee sample efficient RL. Such efforts typically assume the transition dynamics or value function of the MDP are described by linear functions of the state features. To resolve this discrepancy between theory and practice, we introduce the Effective Planning Window (EPW) condition, a structural condition on MDPs that makes *no* linearity assumptions. We demonstrate that the EPW condition permits sample efficient RL, by providing an algorithm which provably solves MDPs satisfying this condition. Our algorithm requires minimal assumptions on the policy class, which can include multi-layer neural networks with nonlinear activation functions. Notably, the EPW condition is directly motivated by popular gaming benchmarks, and we show that many classic Atari games satisfy this condition. We additionally show the necessity of conditions like EPW, by demonstrating that simple MDPs with slight nonlinearities cannot be solved sample efficiently.

1. Introduction

Over the past decade, reinforcement learning (RL) has emerged as the dominant paradigm for sequential decision making. During this time period, video games have served as popular means to benchmark the incremental improvement in state of the art RL. The Arcade Learning Environment (ALE), comprising a suite of classic Atari games, is an archetypical example of such a benchmark (Bellemare et al., 2013). Agents trained by RL can surpass human level performance in such games (Badia et al., 2020).

Motivated by these empirical accomplishments, there has been a major thrust to theoretically characterize the conditions which permit sample efficient RL. In the tabular RL setting, where the number of states is finite and small, sample efficiency bounds scale with cardinality of the state space. However, in practice this cardinality is often large or infinite. For instance, many gaming applications of RL all have continuous state spaces (Berner et al., 2019). These scenarios are handled in the function approximation setting (Du et al., 2019; 2020). Here, each state is associated with a known feature, and one desires a sample efficiency bound that scales with the dimensionality of the features (instead of the cardinality of the state space).

To understand when continuous space RL is sample efficient, theoreticians make certain assumptions on the features or the underlying Markov Decision Process (MDP). A prominent assumption, which has appeared in various forms, is that the problem satisfies some sort of *linear* structure. For instance, in the linear MDP, the transitions and rewards are described by linear functions of the features (Yang & Wang, 2019; Jin et al., 2020; Yang & Wang, 2020; Wang et al., 2021). A weaker, but frequently occurring, form of this assumption is that value function of any policy is nearly linear (Du et al., 2020; Lattimore et al., 2020), or that the optimal value function is linear (Du et al., 2019). Such linear structure is amenable to theoretical analysis.

To obtain a holistic understanding of RL, examining such linear structure is important. But it is unclear whether the aforementioned linearity conditions actually hold in practical scenarios. For instance, it has recently been shown both theoretically and empirically that the optimal value function and optimal policy can be very complex, even in elementary continuous state space MDPs (Dong et al., 2020). Since even powerful linear functions such as the Neural Tangent Kernel are worse in terms of representation power than nonlinear neural networks (Allen-Zhu & Li, 2019; Li et al., 2020), it is unclear whether linear functions can be used to approximate the value function or underlying policy well.

Even in simple RL gaming benchmarks, there is no evi-

[1]Machine Learning Department, Carnegie Mellon University [2]Department of EECS, UC Berkeley. Correspondence to: Dhruv Malik <dhruvm@andrew.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

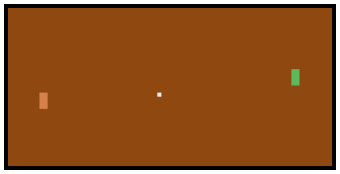

Figure 1. An image of the Atari Pong game. The green paddle must move up and down to hit the ball (the white dot) while playing against the opposing orange paddle.

dence that the aforementioned linearity assumptions hold. Indeed, nonlinear neural networks are predominant means to approximate policies and value functions when solving these games in practice. For instance, consider the Pong game from the ALE benchmark, which is depicted in Figure 1. In this game, the agent must use its paddle to prevent the ball from crossing a boundary, while playing against a pseudorandom opposing paddle. Despite the simplicity of Pong, state of the art methods solve this game using neural networks (Mnih et al., 2013; 2015; Badia et al., 2020), and it is not apparent whether this game is linear in any sense.

This reveals a significant gap between the theory and practice of RL. In theory, one usually employs some sort of linearity assumption to ensure efficient RL. Yet, in practice, RL appears to succeed in domains which do not satisfy such linear structure. In an effort to resolve this discrepancy, we ask the following question:

Which structure is typical of popular RL domains, and does this structure permit sample efficient RL?

This question underlies the analysis of our paper. We make the following contributions:

- We propose the Effective Planning Window (EPW) condition, a structural condition for MDPs which goes beyond linearity. Indeed, this condition is compatible with neural network policies, and MDPs satisfying EPW can have highly nonlinear (stochastic) transitions. Informally, this condition requires the agent to *consistently plan C timesteps ahead*, for a value of C significantly smaller than the horizon length. We show that popular Atari benchmark games satisfy this condition.
- We provide a simple algorithm, which exploits the EPW condition to provably solve MDPs satisfying EPW. We prove the sample efficiency of our algorithm, and show that it requires a number of trajectories that is a lower order polynomial of the horizon length and other relevant problem dependent quantities.
- We argue that one must look beyond linear structure, and further motivate the study and necessity of conditions like EPW, by demonstrating that even slightly nonlinear MDPs cannot be solved sample efficiently.

2. Problem Formulation

2.1. Problem Statement

Notation & Preliminaries. $[n]$ denotes $\{0 \dots n-1\}$ for any integer $n \geq 1$. Recall that an undiscounted, finite horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, H)$ is defined by a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition function $\mathcal{T}$ which maps from state-action pairs to a probability density defined over states, a reward function R which maps from state-action pairs to non-negative real numbers, and a finite planning horizon H. Throughout our paper, we assume that $\mathcal{S} \subseteq \mathbb{R}^d$ and $\mathcal{A}$ is a finite set. We assume a single initial state s_0, and that $\mathcal{S}$ can be partitioned into H different levels. This means that for each $s \in \mathcal{S}$ there exists a unique $h \in [H]$ such that it takes h timesteps to arrive at s from s_0. We say that such a state s lies on level h, and denote $\mathcal{S}_h$ to be the set of states on level h. Taking any action from level $H-1$ exits the game. The notation $\|x\|_2$ denotes the Euclidean norm of x.

A policy maps each state to a corresponding distribution over actions. In practice, one typically uses a policy that is parameterized by parameters belonging to some set $\Theta \subseteq \mathbb{R}^k$. We study such policies, and use $\pi(\theta)$ to denote the policy induced by using parameter $\theta \in \Theta$. When discussing a policy which is not parameterized, we simply use π to denote the policy. We use $\pi_s^a(\theta)$ to denote the probability of taking action a at state s when using the policy $\pi(\theta)$. We use $\pi(\Theta)$ to denote $\{\pi(\theta) \text{ s.t. } \theta \in \Theta\}$, which is the set of feasible policies and defines our policy class. Θ^H denotes the Cartesian product of Θ with itself H times. Given a vector $\overline{\theta} \in \Theta^H$, we let $\pi(\overline{\theta})$ denote the policy which executes $\pi(\overline{\theta}_h)$ at for any state lying on level $h \in [H]$, where $\overline{\theta}_h$ denotes the h^{th} entry of $\overline{\theta}$. The value of a policy $\pi(\theta)$ in MDP $\mathcal{M}$ when initialized at state s is denoted $V^s_{\mathcal{M}}(\pi(\theta))$.

Query Model. We adopt the standard episodic RL setup. Given a desired solution accuracy ϵ and failure probability tolerance δ, we are interested in algorithms which can successfully solve an MDP using a number of trajectories that is at most polynomial in H, $|\mathcal{A}|$, d, k, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

To permit sample efficient RL, prior theoretical work has often assumed that MDP satisfies linear structure. However, it is well documented that RL is empirically successful in highly nonlinear domains. We aim to reduce this gap between theory and practice. We now formally state the problem that we consider throughout our paper.

Our goal is to present nonlinear characteristic conditions which permit sample efficient RL, and argue that these conditions are satisfied in practice by popular RL domains.

2.2. Effective Planning Window Condition

We first state basic conditions that are satisfied by most RL problems encountered in practice. We will later refine these

to obtain our Effective Planning Window (EPW) condition, and then show that EPW enables sample efficient RL.

Let us begin by observing that in practice, the policy class $\pi(\Theta)$ typically satisfies some mild regularity assumptions. We formalize this in the following condition.

Condition 1 (Regular Policy Class). *A policy class $\pi(\Theta)$ is said to be Regular when:*

(a) ***Bounded Domain.*** *There exists $B > 0$ such that each $\theta \in \Theta$ satisfies $\|\theta\|_2 \leq B$.*

(b) ***Lipschitz Continuous Policies.*** *There exists $\phi > 0$ such that for any $\theta, \theta' \in \Theta$ and any $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have $|\pi_s^a(\theta) - \pi_s^a(\theta')| \leq \phi \|\theta - \theta'\|_2$.*

We stress that this is a very mild condition, and places minimal restrictions on $\pi(\Theta)$, which can include a multi-layer neural network with a nonlinear activation function. We now introduce the Generic Game condition.

Condition 2 (Generic Game). *An MDP and Regular policy class pair $(\mathcal{M}, \pi(\Theta))$ form a Generic Game if:*

(a) ***Failure States.*** *There is a set of failure states $\mathcal{F} \subset \mathcal{S}$, and taking any action from a state in $\mathcal{F}$ exits the game.*

(b) ***Complete Policy Class.*** *There exists some $\theta^\star \in \Theta$ such that executing $\pi(\theta^\star)$ from s_0 arrives at some state in $\mathcal{S}_{H-1} \setminus \mathcal{F}$ almost surely. We define $\mathcal{S}^\star$ to be the set of all states $s \in \mathcal{S} \setminus \mathcal{F}$ such that executing $\pi(\theta^\star)$ from s reaches $\mathcal{S}_{H-1} \setminus \mathcal{F}$ almost surely. If a state lies in $\mathcal{S}^\star$ we call it a safe state.*

(c) ***Binary Rewards.*** *For any state $s \in \mathcal{S}_{H-1} \setminus \mathcal{F}$ and any $a \in \mathcal{A}$, $R(s, a) = 1$. For any other state s and any $a \in \mathcal{A}$, $R(s, a) = 0$.*

Note that $\mathcal{F}$ describes states where the agent has lost the game. Also, observe that in Generic Games, an optimal policy is one that arrives at a non-failure state in level $H - 1$ almost surely. Hence $\pi(\theta^\star)$ is indeed an optimal policy.

Let us now describe how popular Atari games can be cast as Generic Games. Recall the famous Pong game depicted in Figure 1. In this (single player) game, an RL agent must learn to move the paddle up and down to hit the ball and prevent it from crossing its boundary. The agent loses the game if the ball crosses its own boundary, and wins the game if it hits the ball past the opposing paddle.

We claim that Pong, together with a neural network policy class, satisfies the Generic Game condition. The first two conditions of Generic Games are easy to verify. Note that the states in Pong are images, so $\mathcal{F}$ includes any state where the ball has crossed the agent's boundary, since this corresponds to the agent failing to complete the game. It is known that Atari can be solved using a neural network policy (Mnih et al., 2015), so a policy class parameterized by neural networks is complete.

To ensure that Pong satisfies the third condition, we need to design an appropriate binary reward function. This is handled by redefining $\mathcal{F}$ to include any state $s \in \mathcal{S}_{H-1}$ where the ball has not crossed the opposing paddle. Then one can simply assign a reward of 1 to any state in $\mathcal{S}_{H-1} \setminus \mathcal{F}$, and 0 to all other states, as required by the Generic Game condition. Hence, playing optimally in this Generic Game framework ensures the ball has moved past the opposing paddle, corresponding to winning the game. This reward function is very close to the one already used by practitioners (Bellemare et al., 2013).

Beyond Pong, other Atari games (and similarly themed video games) can be cast as Generic Games. We defer discussion of this to the main paper, due to space constraints.

Does the Generic Game condition permit sample efficient RL? Unfortunately, there exist Generic Games where the MDP is only slightly nonlinear, but even approximating an optimal policy sample efficiently is impossible, and we defer this to the main paper. So we must further restrict this class of games. We first state a useful definition.

Definition 1 (x-Ancestor). *Given a Generic Game $(\mathcal{M}, \pi(\Theta))$, consider any $h \in [H]$ and any state $s' \in \mathcal{S}_h$. A state $s \in \mathcal{S}$ is an x-ancestor of s', if $s \in \mathcal{S}_{\max\{0, h-x\}}$ and there exists some $\theta \in \Theta$ such that following $\pi(\theta)$ from s will reach s' with nonzero probability.*

We now formally state our Effective Planning Window (EPW) condition, which refines our notion of Generic Games. For the statement of the condition, recall our notion of $\mathcal{S}^\star$, which was defined in the Generic Game condition.

Condition 3 (Effective Planning Window). *A Generic Game $(\mathcal{M}, \pi(\Theta))$ satisfies the Effective Planning Window condition with parameter C if there exists $C \in [H]$ such that the following holds. Consider any $s' \in \mathcal{S} \setminus \mathcal{F}$. If s is a C-ancestor of s', then $s \in \mathcal{S}^\star$.*

Before examining RL benchmark games in the context of this condition, a few comments about the condition itself are in order. The quantity C ensures that any C-ancestor of a non-failure state is a safe state. So if an agent is at timestep t and the game is not over, then at timestep $t - C$ it was in a state from where it could have achieved the highest reward possible in the MDP (if it took the correct sequence of actions). For the purposes of RL, this effectively means that at each timestep, the agent must consistently plan over the next C timesteps instead of the entire horizon length H.

Of course, any Generic Game satisfies the EPW condition for a choice of $C = H - 1$. However, many popular RL benchmark games satisfy the EPW property with a value of C that is much smaller than H. Informally, C is the time required by the agent to successfully *react* to scenarios in

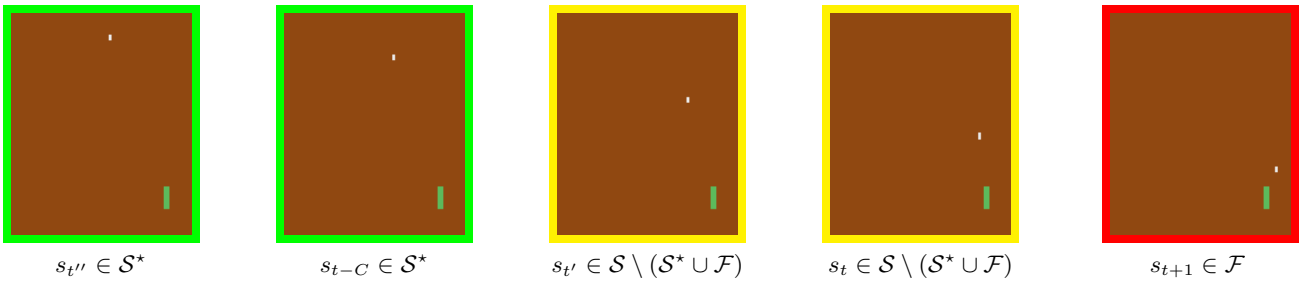

Figure 2. Five states from the Pong game. Here we let $t'' < t - C < t' < t$, and the ball is progressively moving towards the lower right corner. At timesteps t', t, the paddle has not lost the game. However, it does not have enough time to react and reach the ball in time. At timestep $t + 1$ the game is over. At timesteps $t'', t - C$, the paddle has enough time to react and reach the ball.

the game (without losing). Let us understand this in Pong.

In Pong, after the opposing paddle hits the ball, the agent must react to the trajectory of the ball and adjust its position accordingly. If it takes too long to react before it starts adjusting its position, then it will be unable to reach the ball in time. We depict this in Figure 2. More formally, assume that at timestep t the paddle has not lost the game and the ball is moving towards its boundary. At timestep t, the ball may be too close to the boundary, and so the agent will not not have enough time to move its paddle fast enough in order to reach the ball in time. However, at timestep $t - C$ the ball is further away from the boundary, so the agent has enough time to move its paddle appropriately in order to react, reach the ball and hit it back. So at timestep $t - C$ the agent lies in a safe state in $\mathcal{S}^\star$, since it has enough time to adjust its paddle and hit the ball back, and hence play optimally. Notably, if we let C' be the number of timesteps it takes for the ball to traverse from one end of the board to the other, then $C \leq C'$. Hence, when H is large and the agent needs to control the paddle for many rounds, then C is a constant independent of H.

Beyond Pong, other Atari games satisfy the EPW condition, with a constant value of C, and we show this in the main paper. We conclude this section by highlighting two important aspects of the EPW condition.

The Magnitude Of C. We treat C as a constant that is independent of and much smaller than H. This is certainly reasonable given our above discussion. Furthermore, there exist EPW games where $\Omega(|\mathcal{A}|^C)$ sample complexity is necessary to solve the game.

The Challenge Of Solving EPW Games. A deterministic EPW game is straightforward to solve, since an agent can try each of the $|\mathcal{A}|^C$ trajectories to discover which ones do not lead to $\mathcal{F}$. But when transitions are stochastic (as in Atari), this strategy is not possible. Instead, the algorithm must generalize beyond the trajectories it samples, to learn something global about the MDP. Furthermore, stochastic EPW games *cannot* be solved by just splitting the horizon H into H/C distinct planning windows, and then solving these planning problems independently of each other. Instead, the difficulty is that the agent must *consistently* plan C timesteps ahead. Indeed, if a trajectory ends in a failure state after t timesteps, then it is unclear at which of the prior timesteps $\{t - C \dots t - 1\}$ that we took an incorrect action. And we cannot rollback to timestep $t - C$ and rerun the same trajectory to discover when we made a mistake.

3. Main Results

We now formally state Theorem 1, our main result, which shows that EPW games can be solved sample efficiently.

Theorem 1. *There exists an algorithm, which given any $(\mathcal{M}, \pi(\Theta))$ satisfying the EPW condition and poly$(|\mathcal{A}|^C, k, H, \frac{1}{\epsilon}, \log(\frac{\phi B}{\delta}))$ many trajectories, outputs $\overline{\theta}$ satisfying*

$$V_{\mathcal{M}}^{s_0}(\pi(\overline{\theta})) \geq V_{\mathcal{M}}^{s_0}(\pi(\theta^\star)) - \epsilon$$

with probability at least $1 - \delta$.

The algorithm is extremely natural, and deferred to the main paper. Critically, the sample complexity in the above result is independent of the number of states. Also note that we treat C as a constant, as per our discussion while motivating EPW. Nevertheless, as a direct corollary of the work of Du et al. (2020), the exponential dependence on C cannot be improved by a better algorithm or sharper analysis.

We believe that the EPW condition (or other conditions that are similar in spirit) is the *correct* condition for characterizing when sample efficient RL is possible, at least in RL domains like video games. By contrast, the linearity assumptions which prominently appear in prior literature, in addition to lacking clear empirical justification, are quite brittle. To demonstrate this, we show the existence of Generic Games where the optimal value function is a linear combination of two Relu neurons, and the optimal policy is softmax linear, and yet the game cannot be solved sample efficiently. Due to space constraints, this result is deferred to the main paper. This further motivates the importance

of studying conditions such as EPW, instead of assuming that the MDP has linear structure. We must look beyond linearity to obtain a realistic characterization of when sample efficient RL is possible. Our EPW condition, which makes no linearity assumptions, is one example of this.

4. Discussion

In this paper, we studied structural conditions which permit sample efficient RL in continuous state spaces, with a focus on conditions that are typical in popular RL domains such as Atari games. We introduced the EPW condition, which in contrast to prior work, makes no linearity assumptions about the MDP structure. We provided an algorithm which provably solves MDPs satisfying EPW. We analyzed the sample complexity of this algorithm, and showed it requires a number of trajectories that is a lower order polynomial of the horizon length and other relevant problem dependent quantities. We also showed that MDPs which have very slight nonlinearities (but do not satisfy EPW) cannot be solved sample efficiently. Our analysis thus highlights the important need to look beyond linear structure, in order to establish the sample efficiency of RL in popular domains.

A number of open questions remain. First, while our EPW condition is directly motivated by RL gaming domains such as Atari, we emphasize that it is unclear whether EPW is satisfied by other RL application domains such as robotics. A natural direction for future work is to study these domains more closely, and identify structure that permits sample efficient RL in such domains. Second, recall that our algorithm requires access to a particular computational oracle. As discussed, we made this computational abstraction since we placed minimal restrictions on the policy class, so in the worst case obtaining such an oracle could be intractable. Nevertheless, we suspect that when using a neural network policy class with an appropriate architecture, one could approximate this oracle efficiently. It would be interesting to precisely characterize when this is possible. Third, it would be interesting to see whether a variant of our theoretically justified algorithm can be deployed in practice. Using our theoretical insight to design a pragmatic method, with strong empirical performance, is an important direction for future work.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1745016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Allen-Zhu, Z. and Li, Y. What can resnet learn efficiently, going beyond kernels? In *Advances in Neural Information Processing Systems*, 2019.

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. Agent57: Outperforming the Atari human benchmark. In *Proceedings of the International Conference on Machine Learning*, 2020.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Berner, C. et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arxiv:1912.06680*, 2019.

Dong, K., Luo, Y., Yu, T., Finn, C., and Ma, T. On the expressivity of neural networks for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.

Du, S. S., Luo, Y., Wang, R., and Zhang, H. Provably efficient q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, 2019.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of the Conference on Learning Theory*, 2020.

Lattimore, T., Szepesvari, C., and Weisz, G. Learning with good feature representations in bandits and in RL with a generative model. In *Proceedings of the International Conference on Machine Learning*, 2020.

Li, Y., Ma, T., and Zhang, H. R. Learning over-parametrized two-layer neural networks beyond ntk. In *Proceedings of the Conference on Learning Theory*, 2020.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. In *International Conference on Learning Representations*, 2021.

Yang, L. and Wang, M. Sample-optimal parametric q-learning using linearly additive features. In *Proceedings of the International Conference on Machine Learning*, 2019.

Yang, L. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *Proceedings of the International Conference on Machine Learning*, 2020.