DeepMDP: Learning Continuous Latent Space Models for Representation Learning

Carles Gelada [1] **Saurabh Kumar** [1] **Jacob Buckman** [2] **Ofir Nachum** [1] **Marc G. Bellemare** [1]

Abstract

Many reinforcement learning (RL) tasks provide the agent with high-dimensional observations that can be simplified into low-dimensional continuous states. To formalize this process, we introduce the concept of a *DeepMDP*, a parameterized latent space model that is trained via the minimization of two tractable latent space losses: prediction of rewards and prediction of the distribution over next latent states. We show that the optimization of these objectives guarantees (1) the quality of the embedding function as a representation of the state space and (2) the quality of the DeepMDP as a model of the environment. Our theoretical findings are substantiated by the experimental result that a trained DeepMDP recovers the latent structure underlying high-dimensional observations on a synthetic environment. Finally, we show that learning a DeepMDP as an auxiliary task in the Atari 2600 domain leads to large performance improvements over model-free RL.

1. Introduction

In reinforcement learning (RL), it is typical to model the environment as a Markov Decision Process (MDP). However, for many practical tasks, the state representations of these MDPs include a large amount of redundant information and task-irrelevant noise. For example, image observations from the Arcade Learning Environment (Bellemare et al., 2013a) consist of 33,600-dimensional pixel arrays, yet it is intuitively clear that there exist lower-dimensional approximate representations for all games. Consider PONG; observing only the positions and velocities of the three objects in the frame is enough to play. Converting each frame into such a simplified state before learning a policy facilitates the learning process by reducing the redundant and irrelevant information presented to the agent. Representation learning techniques for reinforcement learning seek to improve the learning efficiency of existing RL algorithms by doing exactly this: learning a mapping from states to simplified states.

Bisimulation metrics (Ferns et al., 2004; 2011) define two states to be behaviourally similar if they (1) produce the close immediate reward and (2) they transition to states which themselves are behaviourally similar. Bisimulation metrics have been used reduce the dimensionality of the state space by aggregating states (a form of representation learning) but have not received much attention due to their high computational cost. Furthermore, state aggregation techniques, whether based on bisimulation or other methods (Abel et al., 2017; Li et al., 2006; Singh et al., 1995; Givan et al., 2003; Jiang et al., 2015; Ruan et al., 2015), suffer from poor compatibility with function approximation methods. Instead, to support stochastic gradient descent-based training procedures we explore the use of continuous latent representations. Specifically, for any MDP, we propose utilizing the latent space of its corresponding *DeepMDP*.

A DeepMDP is a latent space model of an MDP which has been trained to minimize two tractable losses: predicting the rewards and predicting the distribution of next latent states. DeepMDPs can be viewed as a formalization of recent works which use neural networks to learn latent space models of the environment (Ha & Schmidhuber, 2018; Oh et al., 2017; Hafner et al., 2018). The state of a DeepMDP can be interpreted as a representation of the original MDP's state, and doing so reveals a profound theoretical connection to bisimulation. We show that minimization of the DeepMDP losses guarantees that two non-bisimilar states can never be collapsed into a single representation. Additionally, this guarantees that value functions in the DeepMDP are good approximations of value functions in the original task MDP. These results serve not only provide a theoretically-grounded approach to representation learning but also represent a promising first step towards principled latent-space model-based RL algorithms.

In a synthetic environment, we show that a DeepMDP learns to recover the low-dimensional latent structure underlying

[1]Google Brain [2]Center for Language and Speech Processing, Johns Hopkins University. Correspondence to: Carles Gelada <cgel@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

high-dimensional observations. We then demonstrate that learning a DeepMDP as an auxiliary task to model-free RL in the Atari 2600 environment (Bellemare et al., 2013b) leads to significant improvement in performance when compared to a baseline model-free method.

2. Background

Define a Markov Decision Process (MDP) in standard fashion: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ (Puterman, 1994). For simplicity of notation we will assume that $\mathcal{S}$ and $\mathcal{A}$ are discrete spaces unless otherwise stated. A policy π defines a distribution over actions conditioned on the state, $\pi(a|s)$. Denote by Π the set of all stationary policies. The value function of a policy $\pi \in \Pi$ at a state s is the expected sum of future discounted rewards by running the policy from that state. $V^\pi : \mathcal{S} \to \mathbb{R}$ is defined as:

$$V^\pi(s) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) | s_0 = s \right].$$

The action value function is similarly defined:

$$Q^\pi(s, a) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) | s_0 = s, a_0 = a \right]$$

We denote π^* as the optimal policy in $\mathcal{M}$; i.e., the policy which maximizes expected future reward. We denote the optimal state and action value functions with respect to π^* as V^*, Q^*. We denote the stationary distribution of a policy π in $\mathcal{M}$ by d_π; i.e.,

$$d_\pi(s) = \sum_{\dot{s} \in \mathcal{S}, \dot{a} \in \mathcal{A}} \mathcal{P}(s|\dot{s}, \dot{a}) \pi(\dot{a}|\dot{s}) d_\pi(\dot{s}) \, d\dot{a}$$

The state-action stationary distribution is given by $\xi_\pi(s, a) = d_\pi(s)\pi(a|s)$. Although only non-terminating MDPs have stationary distributions, a state distribution for terminating MDPs with similar properties exists (Gelada & Bellemare, 2019).

2.1. Wasserstein Distance

For two distribution P and Q defined on a metric space $\langle \chi, d \rangle$, the optimal transport problem (Villani, 2008) studies how to transform the probability mass of P into Q with the minimum cost, where the cost of a particle from point x to y comes from a metric $d(x, y)$. The Wasserstein-1 metric between P and Q, denoted by $W(P, Q)$, is the minimal possible cost of such a transport.

Definition 1. *Let d be any metric. The Wasserstein-1 metric W between distributions P and Q is defined as*

$$W_d(P, Q) = \inf_{\lambda \in \Gamma(P,Q)} \int d(x, y) \lambda(x, y) \, dx \, dy.$$

where $\Gamma(P, Q)$ denotes the set of all couplings of P and Q.

When it's clear what the underlying metric is, we will simply write W. The Wasserstein has an equivalent dual form (Mueller, 1997):

$$W_d(P, Q) = \sup_{f \in \mathbb{F}_d} | \mathop{\mathbb{E}}_{x \sim P} f(x) - \mathop{\mathbb{E}}_{y \sim Q} f(y)|, \tag{1}$$

where $\mathbb{F}_d$ is the set of absolutely continuous 1-Lipschitz functions:

$$\mathbb{F}_d = \{f : |f(x) - f(y)| \leqslant d(x, y)\}.$$

The Wasserstein metric can be extended to pseudometrics. A pseudometric d satisfies all the properties of a metric except *identity of indiscernibles*, $d(x, y) = 0 \Leftrightarrow x = y$. The kernel of a pseudometric is the equivalence relation defined for all states where the pseudometric is 0. Note how the triangle inequality of the pseudometric ensures the kernel is a valid equivalence satisfying the associative property. Intuitively, using a pseudometric for the Wasserstein can be interpreted letting points that are different be equivalent under the pseudometric and thus, requiring no transportation.

Central to the results in this work is the connection between the Wasserstein metric and Lipschitz smoothness. The following property, trivially derived from the dual form of the Wasserstein distance, will be used throughout. For any K-Lipschitz function f,

$$| \mathop{\mathbb{E}}_{x \sim P} f(x) - \mathop{\mathbb{E}}_{y \sim Q} f(y)| \leqslant K \cdot W(P, Q) \tag{2}$$

2.2. Lipschitz MDPs

Several works have studied Lipschitz smoothness constraints on the transition and reward functions (Hinderer, 2005; Asadi et al., 2018), to provide conditions for value functions to be Lipschitz. Closely following their formulation, we define Lipschitz MDPs as follows:

Definition 2. *Let $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ be an MDP with a continuous, metric state space $(\mathcal{S}, d_\mathcal{S})$, where $d_\mathcal{S} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$, and a discrete action space $\mathcal{A}$. We say $\mathcal{M}$ is (K_R, K_P)-Lipschitz if, for all $s_1, s_2 \in \mathcal{S}$ and $a \in \mathcal{A}$:*

$$|\mathcal{R}(s_1, a) - \mathcal{R}(s_2, a)| \leqslant K_\mathcal{R} d_\mathcal{S}(s_1, s_2)$$
$$W(\mathcal{P}(\cdot|, s_1, a), \mathcal{P}(\cdot|s_2, a)) \leqslant K_\mathcal{P} d_\mathcal{S}(s_1, s_2)$$

From here onwards, we will we restrict our attention to the set of Lipschitz MDPs for which the constant $K_\mathcal{P}$ is sufficiently small, as stated in the following assumption.

Assumption 1. *The Lipschitz constant $K_\mathcal{P}$ of the transition function $\mathcal{P}$ is strictly smaller than $\frac{1}{\gamma}$.*

From a practical standpoint, Assumption 1 is relatively strong, but simplifies our analysis by ensuring that close states cannot have future trajectories that are "divergent."

An MDP might not exhibit divergent behaviour even when $K_{\mathcal{P}} \geqslant \frac{1}{\gamma}$. In particular, when episodes terminate after a finite amount of time, Assumption 1 becomes unnecessary. We leave as future work how to improve on this assumption.

The main use of Lipschitz MDPs will be to study the Lipschitz properties of value functions.[1]

Definition 3. *A policy $\pi \in \Pi$ is K_V-Lipschitz-valued if for all $s_1, s_2 \in \mathcal{S}$ and $a, a \in \mathcal{A}$:*

$$|V^\pi(s_1) - V^\pi(s_2)| \leqslant K_V d_{\mathcal{S}}(s_1, s_2)$$
$$|Q^\pi(s_1, a) - Q^\pi(s_2, a)| \leqslant K_V d_{\mathcal{S}}(s_1, s_2)$$

Lipschitz MDPs allow us to provide a simple condition for a policy to have a Lipschitz value function.

Lemma 1. *Let $\mathcal{M}$ be (K_R, K_P)-Lipschitz and let π be any policy with the property that $\forall s_1, s_2 \in \mathcal{S}$,*

$$|V^\pi(s_1) - V^\pi(s_2)| \leqslant \max_{a \in \mathcal{A}} |Q^\pi(s_1, a) - Q^\pi(s_2, a)|$$

then π is $\frac{K_{\mathcal{R}}}{1-\gamma K_{\mathcal{P}}}$-Lipschitz-valued.

Proof. See the Appendix for all proofs. □

This defines a rich set of Lipschitz-Valued policies. Notably, the optimal policy π^* satisfies the condition of Lemma 1.

Corollary 1. *Let $\mathcal{M}$ be (K_R, K_P)-Lipschitz, then π^* is $\frac{K_{\mathcal{R}}}{1-\gamma K_{\mathcal{P}}}$-Lipschitz-Valued.*

2.3. Latent Space Models

For some MDP $\mathcal{M}$, let $\overline{\mathcal{M}} = \langle \bar{\mathcal{S}}, \mathcal{A}, \bar{\mathcal{R}}, \bar{\mathcal{P}}, \gamma \rangle$ be an MDP where $\bar{\mathcal{S}} \subset \mathbb{R}^D$ for finite D and the action space $\mathcal{A}$ is shared between $\mathcal{M}$ and $\overline{\mathcal{M}}$. Furthermore, let $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ be an embedding function which connects the state spaces of these two MDPs. We refer to $(\overline{\mathcal{M}}, \phi)$ as a *latent space model* of $\mathcal{M}$.

Since $\overline{\mathcal{M}}$ is, by definition, an MDP, value functions can be defined in the standard way. We use $\bar{V}^{\bar{\pi}}, \bar{Q}^{\bar{\pi}}$ to denote the value functions of a policy $\bar{\pi} \in \bar{\Pi}$, where $\bar{\Pi}$ is the set of policies defined on the state space $\bar{\mathcal{S}}$. We use $\bar{\pi}^*$ to denote the optimal policy in $\overline{\mathcal{M}}$. The corresponding optimal state and action value functions are then $\bar{V}^*, \bar{Q}^*$. For ease of notation, when $s \in \mathcal{S}$, we use $\bar{\pi}(\cdot|s) := \bar{\pi}(\cdot|\phi(s))$ to denote first using ϕ to map s to the state space $\mathcal{S}$ of $\overline{\mathcal{M}}$ and subsequently using $\bar{\pi}$ to generate the probability distribution over actions.

Although similar definitions of latent space models have been previously studied (Francois-Lavet et al., 2018; Zhang et al., 2018; Ha & Schmidhuber, 2018; Oh et al., 2017; Hafner et al., 2018; Kaiser et al., 2019; Silver et al., 2017), the parametrizations and training objectives used to learn such models have varied widely. For example Ha & Schmidhuber (2018); Hafner et al. (2018); Kaiser et al. (2019) use pixel prediction losses to learn the latent representation while (Oh et al., 2017) chooses instead to optimize the model to predict next latent states with the same value function as the sampled next states.

In this work, we study the minimization of latent space losses defined with respect to rewards and transitions in the latent space:

$$L_{\bar{\mathcal{R}}}(s, a) = |\mathcal{R}(s, a) - \bar{\mathcal{R}}(\phi(s), a)| \tag{3}$$
$$L_{\bar{\mathcal{P}}}(s, a) = W\left(\phi\mathcal{P}(\cdot|s, a), \bar{\mathcal{P}}(\cdot|\phi(s), a)\right) \tag{4}$$

where we use the shorthand notation $\phi\mathcal{P}(\cdot|s, a)$ to denote the probability distribution over $\bar{\mathcal{S}}$ of first sampling $s' \sim \mathcal{P}(\cdot|s, a)$ and then embedding $\bar{s}' = \phi(s')$. Francois-Lavet et al. (2018) and Chung et al. (2019) have studied similar latent losses, but to the best of our knowledge ours is the first theoretical analysis of latent space models as auxiliary losses.

We use the term **DeepMDP** to refer to a parameterized latent space model which minimizes the latent losses $L_{\bar{\mathcal{R}}}$ and $L_{\bar{\mathcal{P}}}$ (sometimes referred to as the DeepMDP losses). In Section 3, we derive theoretical guarantees of DeepMDPs when minimizing the DeepMDP losses over the whole state space (which we term global optimization). However, our principal objective is to learn DeepMDPs parameterized by deep networks, which requires losses in the form of expectations. We show in Section 4 that similar theoretical guarantees can be obtained in this setting.

3. Global DeepMDP Bounds

We refer to the following losses as the *global* DeepMDP losses, to emphasize their dependence on the whole state and action space:[2]

$$L^\infty_{\bar{\mathcal{R}}} = \sup_{s,a} L_{\bar{\mathcal{R}}}(s, a) \tag{5}$$
$$L^\infty_{\bar{\mathcal{P}}} = \sup_{s,a} L_{\bar{\mathcal{P}}}(s, a) \tag{6}$$

3.1. Value Difference Bound

We start by bounding the difference of the value functions $Q^{\bar{\pi}}$ and $\bar{Q}^{\bar{\pi}}$ for any policy $\bar{\pi} \in \bar{\Pi}$. Note that $Q^{\bar{\pi}}(s, a)$ is computed using $\mathcal{P}$ and $\mathcal{R}$ on $\mathcal{S}$ while $\bar{Q}^{\bar{\pi}}(\phi(s), a)$ is computed using $\bar{\mathcal{P}}$ and $\bar{\mathcal{R}}$ on $\bar{\mathcal{S}}$.

Lemma 2. *Let $\mathcal{M}$ and $\overline{\mathcal{M}}$ be an MDP and DeepMDP respectively, with an embedding function ϕ and global loss*

[1] Another benefit of MDP smoothness is improved learning dynamics. Pirotta et al. (2015) suggest that the smaller the Lipschitz constant of an MDP, the faster it is to converge to a near-optimal policy.

[2] The ∞ notation is a reference to the ℓ_∞ norm

functions $L^{\infty}_{\bar{\mathcal{R}}}$ and $L^{\infty}_{\bar{\mathcal{P}}}$. For any $K_{\bar{V}}$-Lipschitz-valued policy $\bar{\pi} \in \bar{\Pi}$ the value difference can be bounded by

$$\left|Q^{\bar{\pi}}(s,a) - \bar{Q}^{\bar{\pi}}(\phi(s),a)\right| \leqslant \frac{L^{\infty}_{\bar{\mathcal{R}}} + \gamma K_{\bar{V}} L^{\infty}_{\bar{\mathcal{P}}}}{1-\gamma},$$

The previous result holds for all policies $\bar{\Pi} \subseteq \Pi$, a subset of all possible policies Π. The reader might ask whether this is an interesting set of policies to consider; in Section 5, we characterize this set of policies via a connection with bisimulation.

A similar bound can be found in Asadi et al. (2018), who study non-latent transition models with the use of an exact reward function. We also note that our results are arguably simpler, since we do not require the treatment of MDP transitions in terms of distributions over a set of deterministic components.

3.2. Representation Quality Bound

When a representation is used to predict value functions of policies $\bar{\pi} \in \bar{\Pi}$, a clear failure case is when two states with different value functions are collapsed to the same representation. The following result demonstrates that when the global DeepMDP losses $L^{\infty}_{\bar{\mathcal{R}}} = 0$ and $L^{\infty}_{\bar{\mathcal{P}}} = 0$, this failure case can never occur for the embedding function ϕ.

Theorem 1. *Let $\mathcal{M}$ and $\bar{\mathcal{M}}$ be an MDP and DeepMDP respectively, with an embedding function ϕ and global loss functions $L^{\infty}_{\bar{\mathcal{R}}}$ and $L^{\infty}_{\bar{\mathcal{P}}}$. For any $K_{\bar{V}}$-Lipschitz-valued policy $\bar{\pi} \in \bar{\Pi}$ the representation ϕ guarantees that for any $s_1, s_2 \in \mathcal{S}$ and $a \in \mathcal{A}$,*

$$\left|Q^{\bar{\pi}}(s_1,a) - Q^{\bar{\pi}}(s_2,a)\right| \leqslant K_{\bar{V}} \left\|\phi(s_1) - \phi(s_2)\right\|_2 + 2\frac{\left(L^{\infty}_{\bar{\mathcal{R}}} + \gamma K_{\bar{V}} L^{\infty}_{\bar{\mathcal{P}}}\right)}{1-\gamma}$$

This result justifies using the embedding function ϕ as a representation to predict values.

3.3. Suboptimality Bound

Although we do not make use of any form of model-based RL in this work, we bound the performance loss of running the optimal policy of $\bar{\mathcal{M}}$ in $\mathcal{M}$, compared to the optimal policy π^*.

Theorem 2. *Let $\mathcal{M}$ and $\bar{\mathcal{M}}$ be an MDP and a (K_R, K_P)-Lipschitz DeepMDP respectively, with an embedding function ϕ and global loss functions $L^{\infty}_{\bar{\mathcal{R}}}$ and $L^{\infty}_{\bar{\mathcal{P}}}$. For all $s \in \mathcal{S}$, the suboptimality of the optimal policy $\bar{\pi}^*$ of $\bar{\mathcal{M}}$ evaluated on $\mathcal{M}$ can be bounded by,*

$$V^*(s) - V^{\bar{\pi}^*}(s) \leqslant 2\frac{L^{\infty}_{\bar{\mathcal{R}}} + \gamma K_{\bar{V}} L^{\infty}_{\bar{\mathcal{P}}}}{1-\gamma}$$

Where $K_{\bar{V}} = \frac{K_{\bar{\mathcal{R}}}}{1-\gamma K_{\bar{\mathcal{P}}}}$ is an upper bound to the Lipschitz constant of the value function $\bar{V}^{\bar{\pi}^*}$, as shown by Corollary 1.

4. Local DeepMDP Bounds

In large-scale tasks, data from many regions of the state space is often unavailable,[3] making it infeasible to measure – let alone optimize – global losses. Further, when the capacity of a model is limited, or when sample efficiency is a concern, it might not even be desirable to precisely learn a model of the whole state space. Interestingly, we can still provide similar guarantees based on the DeepMDP losses, as measured under an *expectation* over a state-action distribution, denoted here as ξ. We refer to these as the losses *local* to ξ. Taking $L^{\xi}_{\bar{\mathcal{R}}}$, $L^{\xi}_{\bar{\mathcal{P}}}$ to be the reward and transition losses under ξ, respectively, we have the following local DeepMDP losses:

$$L^{\xi}_{\bar{\mathcal{R}}} = \mathop{\mathbb{E}}_{s,a\sim\xi} |\mathcal{R}(s,a) - \bar{\mathcal{R}}(\phi(s),a)|, \tag{7}$$

$$L^{\xi}_{\bar{\mathcal{P}}} = \mathop{\mathbb{E}}_{s,a\sim\xi} \left[W\left(\phi\mathcal{P}(\cdot|s,a), \bar{\mathcal{P}}(\cdot|\phi(s),a)\right)\right]. \tag{8}$$

Losses of this form are compatible with the stochastic gradient decent methods used by neural networks. Thus, study of the local losses allows us to bridge the gap between theory and practice.

4.1. Value Difference Bound

We provide a value function bound for the local case, analogous to Lemma 2.

Lemma 3. *Let $\mathcal{M}$ and $\bar{\mathcal{M}}$ be an MDP and DeepMDP respectively, with an embedding function ϕ. For any $K_{\bar{V}}$-Lipschitz-valued policy $\bar{\pi} \in \bar{\Pi}$, the expected value function difference can be bounded using the local loss functions $L^{\xi_{\bar{\pi}}}_{\bar{\mathcal{R}}}$ and $L^{\xi_{\bar{\pi}}}_{\bar{\mathcal{P}}}$ measured under $\xi_{\bar{\pi}}$, the stationary state action distribution of $\bar{\pi}$.*

$$\mathop{\mathbb{E}}_{s,a\sim\xi_{\bar{\pi}}} \left|Q^{\bar{\pi}}(s,a) - \bar{Q}^{\bar{\pi}}(\phi(s),a)\right| \leqslant \frac{\left(L^{\xi_{\bar{\pi}}}_{\bar{\mathcal{R}}} + \gamma K_{\bar{V}} L^{\xi_{\bar{\pi}}}_{\bar{\mathcal{P}}}\right)}{1-\gamma},$$

The provided bound guarantees that for any policy $\bar{\pi} \in \bar{\Pi}$ which visits state-action pairs (s,a) where $L_{\bar{\mathcal{R}}}(s,a)$ and $L_{\bar{\mathcal{P}}}(s,a)$ are small, the DeepMDP will provide accurate value functions for any states likely to be seen under the policy. [4]

[3] Challenging exploration environments like Montezuma's Revenge are a prime example.

[4] The value functions might be inaccurate in states that the policy $\bar{\pi}$ rarely visits.

4.2. Representation Quality Bound

We can also extend the local value difference bound to provide a local bound on how well the representation ϕ can be used to predict the value function of a policy $\bar{\pi} \in \bar{\Pi}$, analogous to Theorem 1.

Theorem 3. *Let $\mathcal{M}$ and $\overline{\mathcal{M}}$ be an MDP and DeepMDP respectively, with an embedding function ϕ. Let $\bar{\pi} \in \bar{\Pi}$ be any $K_{\bar{V}}$-Lipschitz-valued policy with stationary distribution $d_{\bar{\pi}}(s)$ and let $L_{\bar{\mathcal{R}}}^{\xi_{\bar{\pi}}}$ and $L_{\bar{\mathcal{P}}}^{\xi_{\bar{\pi}}}$ be the local loss functions measured under $\xi_{\bar{\pi}}$, the stationary state action distribution of $\bar{\pi}$. For any two states $s_1, s_2 \in \mathcal{S}$, the local representation similarity can be bounded by*

$$|V^{\bar{\pi}}(s_1) - V^{\bar{\pi}}(s_2)| \leqslant K_{\bar{V}} \left\|\phi(s_1) - \phi(s_2)\right\|_2 + \frac{L_{\bar{\mathcal{R}}}^{\xi_{\bar{\pi}}} + \gamma K_{\bar{V}} L_{\bar{\mathcal{P}}}^{\xi_{\bar{\pi}}}}{1-\gamma} \left(\frac{1}{d_{\bar{\pi}}(s_1)} + \frac{1}{d_{\bar{\pi}}(s_2)}\right)$$

Thus, the representation quality argument given in 3.2 holds for any two states s_1 and s_2 which are visited often by a policy $\bar{\pi}$.

5. Connection to Bisimulation

As we will now see, the representation learned by optimizing the DeepMDP losses is closely connected to bisimulation. Givan et al. (2003) first studied bisimulation relations in the context of RL as a formalization of behavioural equivalence between states. They proposed grouping equivalent states to reduce the dimensionality of the MDP.

Definition 4 (Givan et al. (2003))**.** *Given an MDP $\mathcal{M}$, an equivalence relation B between states is a bisimulation relation if for all states $s_1, s_2 \in \mathcal{S}$ equivalent under B (i.e. $s_1 B s_2$), the following conditions hold,*

$$R(s_1, a) = R(s_2, a)$$
$$\mathcal{P}(G|s_1, a) = \mathcal{P}(G|s_2, a), \forall G \in \mathcal{S}/B$$

Where $\mathcal{S}/B$ denotes the partition of $\mathcal{S}$ under the relation B, the set of all sets of equivalent states, and where $\mathcal{P}(G|s, a) = \sum_{s' \in G} \mathcal{P}(s'|s, a)$.

Essentially, two states are bisimilar if (1) they have the same immediate reward for all actions and (2) both of their distributions over next-states contain states which themselves are bisimilar. Of particular interest is the maximal bisimulation relation $\sim$, which defines the partition $\mathcal{S}/\sim$ with the fewest elements.

A drawback of bisimulation relations is their *all-or-nothing* nature. Two states that are nearly identical, but differ slightly in their reward or transition functions, are treated as though they were just as unrelated as two states with nothing in common. Ferns et al. (2004) introduced the usage of bisimulation metrics, which are pseudometrics that to quantify the behavioural similarity of two discrete states, and proposed the aggregation of states that are ϵ away under the bisimulation metric. We present the extension of bisimulation metrics to continuous state spaces as proposed in Ferns et al. (2011).

Definition 5. *Given an MDP $\mathcal{M}$, a bisimulation metric $\widetilde{d}$ satisfies the fixed point:*

$$\widetilde{d}(s_1, s_2) = \max_a (1-\gamma) \left|\mathcal{R}(s_1, a) - \mathcal{R}(s_2, a)\right| + \gamma W_{\widetilde{d}}(\mathcal{P}(\cdot|s_1, a), \mathcal{P}(\cdot|s_2, a))$$

This recurrent formulation of bisimulation metrics has a single fixed point $\widetilde{d}$, which has as its kernel the maximal bisimulation relation $\sim$ (i.e. $\widetilde{d}(s_1, s_2) = 0 \iff s_1 \sim s_2$).

We show a connection between bisimulation metrics and the representation ϕ learned by the global DeepMDP losses (see Lemma 5 in Appendix). The main application of this result will be to characterize the set of policies $\bar{\Pi}$. With that aim, we first define the concept of Lipschitz-bisimilar policies:

Definition 6. *We denote by $\widetilde{\Pi}_K$ the set of K-**Lipschitz-bisimilar policies**, s.t. for all $s_1, s_2 \in \mathcal{S}, a \in \mathcal{A}$,*

$$\{\pi : \pi \in \Pi, |\pi(a|s_1) - \pi(a|s_2)| \leqslant K\widetilde{d}(s_1, s_2)\}.$$

Lipschitz-bisimilar policies act differently only on states that are different under the bisimulation metric: $\widetilde{\Pi}$ excludes any policies which act differently on states that are fundamentally equivalent. We guarantee that the set of deep policies is sufficiently expressive by showing that minimizing the global DeepMDP losses ensures that for any $\widetilde{\pi} \in \widetilde{\Pi}_K$, there is a deep policy $\bar{\Pi}$ which is close. The following result thus characterizes the set of policies $\bar{\Pi}$:

Theorem 4. *Let $\mathcal{M}$ be an MDP and $\overline{\mathcal{M}}$ be a ($K_{\bar{\mathcal{R}}}$, $K_{\bar{\mathcal{P}}}$)-Lipschitz DeepMDP, with an embedding function ϕ, and global loss functions $L_{\bar{\mathcal{R}}}^{\infty}$ and $L_{\bar{\mathcal{P}}}^{\infty}$. Denote by $\bar{\Pi}_K$ the set of K-**Lipschitz deep policies** $\{\bar{\pi} : \bar{\pi} \in \bar{\Pi}, |\bar{\pi}(a|s_1) - \bar{\pi}(a|s_2)| \leqslant K \left\|\phi(s_1) - \phi(s_2)\right\|_2, \forall s_1, s_2 \in \mathcal{S}, a \in \mathcal{A}\}$. Finally define the constant $C = \frac{(1-\gamma)K_{\bar{\mathcal{R}}}}{1-\gamma K_{\bar{\mathcal{P}}}}$. Then for any $\widetilde{\pi} \in \widetilde{\Pi}_K$ there exists a $\bar{\pi} \in \bar{\Pi}_{CK}$ which is close to $\widetilde{\pi}$ in the sense that, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,*

$$|\widetilde{\pi}(a|s) - \bar{\pi}(a|s)| \leqslant L_{\bar{\mathcal{R}}}^{\infty} + \gamma L_{\bar{\mathcal{P}}}^{\infty} \frac{K_{\bar{\mathcal{R}}}}{1-\gamma K_{\bar{\mathcal{P}}}}$$

We speculate that similar results should be possible based on *local* DeepMDP losses, but they would require a generalization of bisimulation metrics to the local setting.

Although bisimulation metrics have been used for state aggregation (Givan et al., 2003; Ferns et al., 2004; Ruan et al., 2015), feature discovery (Comanici & Precup, 2011) and

transfer learning between MDPs (Castro & Precup, 2010), they have not been scaled up to modern deep reinforcement learning techniques. In that sense, our method is of independent interest as a practical representation learning scheme for deep reinforcement learning that provides the desirable properties of bisimulation metrics via learning objectives that are tractable to compute.

6. Related Work

We have shown that learning a DeepMDP via the minimization of latent space losses leads to representations that:

- Allow for the good approximation of a large set of *interesting* policies.
- Allow for the good approximation of the *value function* of these policies.

Thus, DeepMDPs are a mathematically sound approach to representation learning that is compatible with neural nets and simple to implement.

A similar connection between the quality of representations and model based objectives in the linear setting was made by Parr et al. (2008). There exists an extensive body of literature on exploiting the transition function structure for representation learning (Mahadevan & Maggioni, 2007; Barreto et al., 2017), but these works do not rely on the reward function. Recently, Bellemare et al. (2019) approached the representation learning problem from the perspective that a good representation is one that allows the prediction via a linear map of any possible value function in the value function polytope (Dadashi et al., 2019). Other auxiliary tasks, without the same level of theoretical justification, been shown to improve the performance of RL agents (Jaderberg et al., 2016; van den Oord et al., 2018; Mirowski et al., 2017). Lyle et al. (2019) argued that the performance benefits of Distributional RL (Bellemare et al., 2017a) can also be explained as a form of auxiliary task.

7. Empirical Evaluation

Our results depend on minimizing losses in expectation, which is the main requirement for deep networks to be applicable. Still, two main obstacles arise when turning these theoretical results into practical algorithms:

(1) Minimization of the Wasserstein Arjovsky et al. (2017) first proposed the use of the Wasserstein distance for Generative Adversarial Networks (GANs) via its dual formulation (see Equation 1). Their approach consists of training a network, constrained to be 1-Lipschitz, to attain the supremum of the dual. Once this supremum is attained, the Wasserstein can be minimized by differentiating through the network. Quantile regression has been proposed as an alternative solution to the minimization of the Wasserstein (Dabney et al., 2018b), (Dabney et al., 2018a), and has shown to perfom well for Distributional RL. The reader might note that issues with the stochastic minimization of the Wasserstein distance have been found by Bellemare et al. (2017b) and Bikowski et al. (2018). In our experiments, we circumvent these issues by assuming that both $\mathcal{P}$ and $\bar{\mathcal{P}}$ are deterministic. This reduces the Wasserstein distance $W\left(\phi\mathcal{P}(\cdot|s,a), \bar{\mathcal{P}}(\cdot|\phi(s),a)\right)$ to $\left\|\phi(\mathcal{P}(s,a)) - \bar{\mathcal{P}}(\phi(s),a)\right\|_2$, where $\mathcal{P}(s,a)$ and $\bar{\mathcal{P}}(\bar{s},a)$ denote the deterministic transition functions.

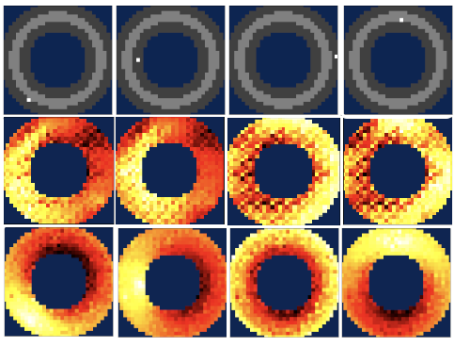
(2) Control the Lipschitz constants $K_{\bar{\mathcal{R}}}$ and $K_{\bar{\mathcal{P}}}$. We also turn to the field of Wasserstein GANs for approaches to constrain deep networks to be Lipschitz. Originally, Arjovsky et al. (2017) used a projection step to constraint the discriminator function to be 1-Lipschitz. Gulrajani et al. (2017a) proposed using a gradient penalty, and sowed improved learning dynamics. Lipschitz continuity has also been proposed as a regularization method by Gouk et al. (2018), who provided an approach to compute an upper bound to the Lipschitz constant of neural nets. In our experiments, we follow Gulrajani et al. (2017a) and utilize the gradient penalty.

7.1. DonutWorld Experiments

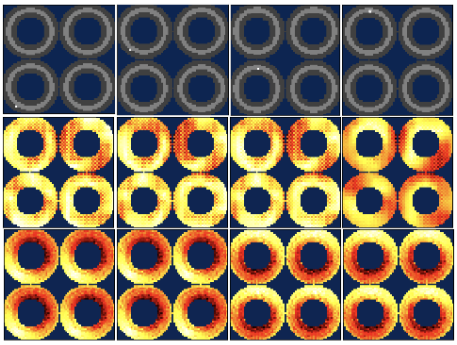
In order to evaluate whether we can learn effective representations, we study the representations learned by DeepMDPs in a simple synthetic environment we call *DonutWorld*. DonutWorld consists of an agent rewarded for running clockwise around a fixed track. Staying in the center of the track results in faster movement. Observations are given in terms of 32x32 greyscale pixel arrays, but there is a simple 2D latent state space (the x-y coordinates of the agent). We investigate whether the x-y coordinates are correctly recovered when learning a two-dimensional representation.

This task epitomizes the low-dimensional dynamics, high-dimensional observations structure typical of Atari 2600 games, while being sufficiently simple to experiment with. We implement the DeepMDP training procedure using Tensorflow and compare it to a simple autoencoder baseline. See Appendix B for a full environment specification, experimental setup, and additional experiments. Code for replicating all experiments is included in the supplementary material.

In order to investigate whether the learned representations learned correspond well to reality, we plot a heatmap of closeness of representation for various states. Figure 1(a) shows that the DeepMDP representations effectively recover the underlying state of the agent, i.e. its 2D position, from the high-dimensional pixel observations. In contrast, the autoencoder representations are less meaningful, even when the autoencoder solves the task near-perfectly.



(a) One-track DonutWorld.



(b) Four-track DonutWorld.

Figure 1. Given a state in our DonutWorld environment (first row), we plot a heatmap of the distance between that latent state and each other latent state, for both autoencoder representations (second row) and DeepMDP representations (third row). More-similar latent states are represented by lighter colors.

In Figure 1(b), we modify the environment: rather than a single track, the environment now has four identical tracks. The agent starts in one uniformly at random, and cannot move between tracks. The DeepMDP hidden state correctly merges all states with indistinguishable value functions, learning a deep state representation which is almost completely invariant to which track the agent is in.

7.2. Atari 2600 Experiments

In this section, we demonstrate practical benefits of approximately learning a DeepMDP in the Arcade Learning Environment (Bellemare et al., 2013a). Our results on *representation-similarity* indicate that learning a DeepMDP is a principled method for learning a high-quality representation. Therefore, we minimize DeepMDP losses as an auxiliary task alongside model-free reinforcement learning, learning a single representation which is shared between both tasks. Our implementations of the proposed algorithms are based on Dopamine (Castro et al., 2018).

We adopt the Distributional Q-learning approach to model-free RL; specifically, we use as a baseline the C51 agent (Bellemare et al., 2017a), which estimates probability masses on a discrete support and minimizes the KL divergence between the estimated distribution and a target distribution. C51 encodes the input frames using a convolutional neural network $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, outputting a dense vector representation $\bar{s} = \phi(s)$. The C51 Q-function is a feed-forward neural network which maps $\bar{s}$ to an estimate of the reward distribution's logits.

To incorporate learning a DeepMDP as an auxiliary learning objective, we define a deep reward function and deep transition function. These are each implemented as a feed-forward neural network, which uses $\bar{s}$ to estimate the immediate reward and the next-state representation, respectively. The overall objective function is a simple linear combination of the standard C51 loss and the Wasserstein distance-based approximations to the local DeepMDP loss given by Equations 7 and 8. For experimental details, see Appendix C.

By optimizing ϕ to jointly minimize both C51 and DeepMDP losses, we hope to learn meaningful $\bar{s}$ that form the basis for learning good value functions. In the following subsections, we aim to answer the following questions: (1) How does the learning of a DeepMDP affect the overall performance of C51 on Atari 2600 games? (2) How do the DeepMDP objectives compare with similar representation-learning approaches?

7.3. DeepMDPs as an Auxiliary Task

We show that when using the best performing DeepMDP architecture described in Appendix $C.2$, we obtain nearly consistent performance improvements over C51 on the suite of 60 Atari 2600 games (see Figure 2).

7.4. Comparison to Alternative Objectives

We empirically compare the effect of the DeepMDP auxilliary objectives on the performance of a C51 agent to a variety of alternatives. In the experiments in this section, we replace the deep transition loss suggested by the DeepMDP bounds with each of the following:

(1) *Observation Reconstruction*: We train a state decoder to reconstruct observations $s \in \mathcal{S}$ from $\bar{s}$. This framework is similar to (Ha & Schmidhuber, 2018), who learn a latent space representation of the environment with an autoencoder, and use it to train an RL agent.

(2) *Next Observation Prediction*: We train a transition model to predict next observations $s' \sim \mathcal{P}(\cdot|s, a)$ from the current state representation $\bar{s}$. This framework is similar to model-based RL algorithms which predict future observations (Xu et al., 2018).

(3) *Next Logits Prediction*: We train a transition model to predict next-state representations such that the Q-function correctly predicts the logits of (s', a'), where a' is the ac-

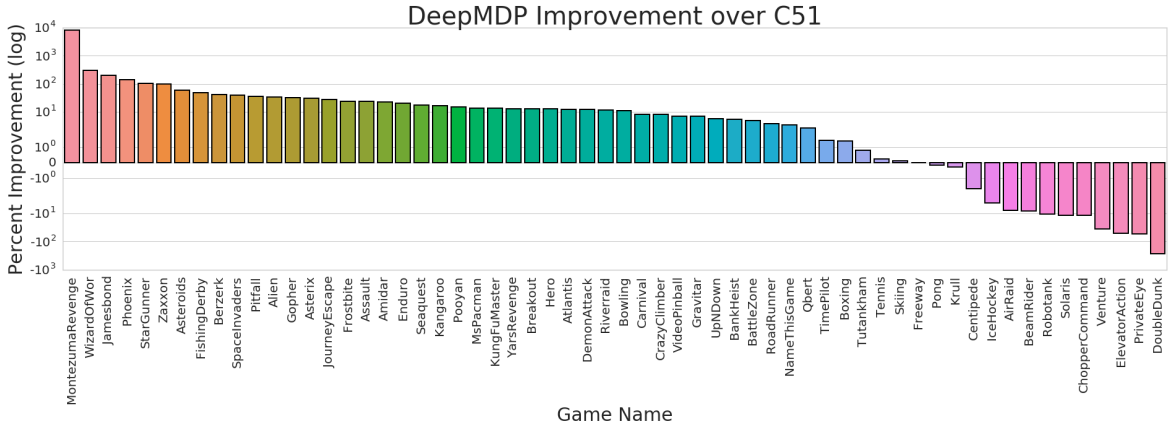


Figure 2. We compare the DeepMDP agent versus the C51 agent on the 60 games from the ALE (3 seeds each). For each game, the percentage performance improvement of DeepMDP over C51 is recorded.

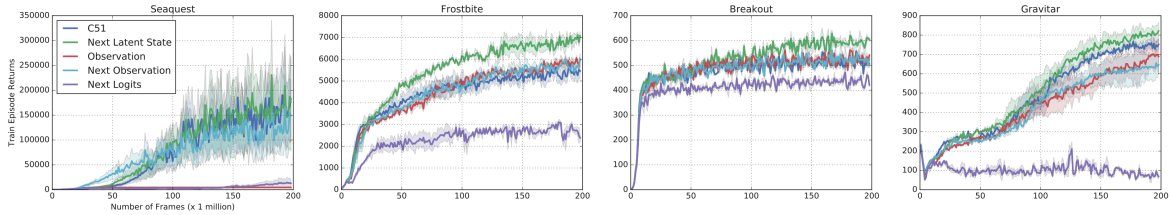


Figure 3. Using various auxiliary tasks in the Arcade Learning Environment. We compare predicting the next state's representation (Next Latent State, recommended by theoretical bounds on DeepMDPs) with reconstructing the current observation (Observation), predicting the next observation (Next Observation), and predicting the next C51 logits (Next Logits). Training curves for a baseline C51 agent are also shown.

tion associated with the max Q-value of s'. This can be understood as a distributional analogue of the Value Prediction Network, VPN, (Oh et al., 2017). Note that this auxiliary loss is used to update only the parameters of the representation encoder and the transition model, not the Q-function.

Our experiments demonstrate that the deep transition loss suggested by the DeepMDP bounds (i.e. predicting the next state's representation) outperforms all three ablations (see Figure 3). Accurately modeling Atari 2600 frames, whether through observation reconstruction or next observation prediction, forces the representation to encode irrelevant information with respect to the underlying task. VPN-style losses have been shown to be helpful when using the learned predictive model for planning (Oh et al., 2017); however, we find that with a distributional RL agent, using this as an auxiliary task tends to hurt performance.

8. Conclusions

We introduce the concept of a DeepMDP: a parameterized latent space model trained via the minimization of tractable latent space losses. Theoretical analysis of DeepMDPs reveals several insights. A novel connection to bisimulation metrics guarantees that our analysis applies to a large set of interesting policies. Further, the representation allows the value functions of these policies can be predicted. Together, these findings suggest a novel approach to representation learning. Our results are corroborated by strong performance on large-scale Atari 2600 experiments, demonstrating that model-based DeepMDP auxiliary losses can be useful auxiliary tasks in model-free RL. Using the transition and reward models of the DeepMDP for model-based RL (e.g. planning, exploration) is a promising future research direction. Additionally, extending DeepMDPs to accommodate different action spaces or time scales from the original MDPs could be a promising path towards learning hierarchical models of the environment.

9. Acknowledgements

The authors would like to thank Philip Amortila and Robert Dadashi for invaluable feedback on the theoretical results; Pablo Samuel Castro, Doina Precup, Nicolas Le Roux, Sasha Vezhnevets, Simon Osindero, Arthur Gretton, Adrien Ali Taiga, Fabian Pedregosa and Shane Gu for useful discussions and feedback.

References

Abel, D., Hershkowitz, D. E., and Littman, M. L. Near optimal behavior via approximate state abstraction. *arXiv preprint arXiv:1701.04113*, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *ICML*, 2017.

Asadi, K., Misra, D., and Littman, M. L. Lipschitz continuity in model-based reinforcement learning. *arXiv preprint arXiv:1804.07193*, 2018.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., Silver, D., and van Hasselt, H. P. Successor features for transfer in reinforcement learning. In *NIPS*, 2017.

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, June 2013a.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. H. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.*, 47:253–279, 2013b.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2017a.

Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., and Munos, R. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017b.

Bellemare, M. G., Dabney, W., Dadashi, R., Taiga, A. A., Castro, P. S., Roux, N. L., Schuurmans, D., Lattimore, T., and Lyle, C. A geometric perspective on optimal representations for reinforcement learning. *CoRR*, abs/1901.11530, 2019.

Bikowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

Castro, P. and Precup, D. Using bisimulation for policy transfer in mdps. *Proceeedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010)*, 2010.

Castro, P. S., Moitra, S., Gelada, C., Kumar, S., and Bellemare, M. G. Dopamine: A research framework for deep reinforcement learning. *arXiv*, 2018.

Chung, W., Nath, S., Joseph, A. G., and White, M. Two-timescale networks for nonlinear value function approximation. In *International Conference on Learning Representations*, 2019.

Comanici, G. and Precup, D. Basis function discovery using spectral clustering and bisimulation metrics. In *AAMAS*, 2011.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. In *ICML*, 2018a.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. In *AAAI*, 2018b.

Dadashi, R., Taiga, A. A., Roux, N. L., Schuurmans, D., and Bellemare, M. G. The value function polytope in reinforcement learning. *CoRR*, abs/1901.11524, 2019.

Ferns, N., Panangaden, P., and Precup, D. Metrics for finite markov decision processes. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI'04:162–169, 2004.

Ferns, N., Panangaden, P., and Precup, D. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Francois-Lavet, V., Bengio, Y., Precup, D., and Pineau, J. Combined reinforcement learning via abstract representations. *arXiv preprint arXiv:1809.04506*, 2018.

Gelada, C. and Bellemare, M. G. Off-policy deep reinforcement learning by bootstrapping the covariate shift. *CoRR*, abs/1901.09455, 2019.

Givan, R., Dean, T., and Greig, M. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. *CoRR*, abs/1804.04368, 2018.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *NIPS*, 2017a.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017b.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pp. 2455–2467, 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Hinderer, K. Lipschitz continuity of value functions in markovian decision processes. *Math. Meth. of OR*, 62: 3–22, 2005.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Jiang, N., Kulesza, A., and Singh, S. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 179–188, 2015.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Sepassi, R., Tucker, G., and Michalewski, H. Model-based reinforcement learning for atari. *CoRR*, abs/1903.00374, 2019.

Li, L., Walsh, T. J., and Littman, M. L. Towards a unified theory of state abstraction for mdps. In *ISAIM*, 2006.

Lyle, C., Castro, P. S., and Bellemare, M. G. A comparative analysis of expected and distributional reinforcement learning. *CoRR*, abs/1901.11084, 2019.

Mahadevan, S. and Maggioni, M. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.

Mirowski, P. W., Pascanu, R., Viola, F., Soyer, H., Ballard, A. J., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., and Hadsell, R. Learning to navigate in complex environments. *CoRR*, abs/1611.03673, 2017.

Mueller, A. Integral probability metrics and their generating classes of functions. 1997.

Oh, J., Singh, S., and Lee, H. Value prediction network. In *Advances in Neural Information Processing Systems*, pp. 6118–6128, 2017.

Parr, R., Li, L., Taylor, G., Painter-Wakefield, C., and Littman, M. L. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *ICML*, 2008.

Pirotta, M., Restelli, M., and Bascetta, L. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2-3):255–283, 2015.

Puterman, M. L. Markov decision processes: Discrete stochastic dynamic programming. 1994.

Ruan, S. S., Comanici, G., Panangaden, P., and Precup, D. Representation discovery for mdps using bisimulation metrics. In *AAAI*, 2015.

Silver, D., van Hasselt, H. P., Hessel, M., Schaul, T., Guez, A., Harley, T., Dulac-Arnold, G., Reichert, D. P., Rabinowitz, N. C., Barreto, A., and Degris, T. The predictron: End-to-end learning and planning. In *ICML*, 2017.

Singh, S. P., Jaakkola, T., and Jordan, M. I. Reinforcement learning with soft state aggregation. In *Advances in neural information processing systems*, pp. 361–368, 1995.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Villani, C. Optimal transport: Old and new. 2008.

Xu, H., Li, Y., Tian, Y., Darrell, T., and Ma, T. Algorithmic framework for model-based reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M. J., and Levine, S. Solar: Deep structured latent representations for model-based reinforcement learning. *arXiv preprint arXiv:1808.09105*, 2018.