SimSR: Simple Distance-based State Representation for Deep Reinforcement Learning

Hongyu Zang [1], **Xin Li** [*] [1], **Mingzhong Wang** [2]

[1] Beijing Institute of Technology
[2] University of the Sunshine Coast
{zanghyu,xinli}@bit.edu.cn, mwang@usc.edu.au

Abstract

This work explores how to learn robust and generalizable state representation from image-based observations with deep reinforcement learning methods. Addressing the computational complexity, stringent assumptions and representation collapse challenges in existing work of bisimulation metric, we devise *Simple State Representation* (SimSR) operator, which achieves equivalent functionality while reducing the complexity by an order in comparison with bisimulation metric. SimSR enables us to design a stochastic-approximation-based method that can practically learn the mapping functions (encoders) from observations to latent representation space. Besides the theoretical analysis, we experimented and compared our work with recent state-of-the-art solutions in visual MuJoCo tasks. The results shows that our model generally achieves better performance and has better robustness and good generalization.

Introduction

Deep reinforcement learning (RL) with image-based observations commonly leverages deep convolutional networks to obtain low-dimensional representations of environments to accomplish sequential decision tasks. Conventionally, the features, which deep convolutional networks embed raw observations into, are modeled as state representations. As these representations are generally learned implicitly as a byproduct of deep RL, they are unlikely to provide sufficient and compact information to summarize the task-relevant states or enhance policy learning. This explains in part why learning policies from physical states, e.g., control inputs like robot's velocity, is generally more sample-efficient than learning policies from image-based environment.

However, since physical states are unobtainable in many real-world domains, recent work in RL has attempted to learn a mapping from image space to low-dimensional representation space that contains the concise state information with the help of auxiliary tasks and self-supervision objectives (Pathak et al. 2017; Laskin, Srinivas, and Abbeel 2020; Lee et al. 2020a; Yarats et al. 2021b; Allen et al. 2021; Fan and Li 2021). Some existing work learns low-dimensional representation by enforcing reconstruction objective to improve the predictiveness and consistency of representations (Jaderberg et al. 2017; Pathak et al. 2017; Yarats et al. 2021b). Other work utilizes contrastive loss for heuristic augmentation pairs to achieve more distinguishable representations of the environment observations (van den Oord, Li, and Vinyals 2018; Anand et al. 2019; Warde-Farley et al. 2019; Laskin, Srinivas, and Abbeel 2020). However, both lines of research are prone to task-agnostic representations as they encode all signals no matter which are relevant to the task or not.

The latest study (Gelada et al. 2019; Castro 2020; Zhang et al. 2021; Castro et al. 2021) presents promising results on learning robust state representation with bisimulation metric (Ferns and Precup 2014; Castro 2020), which implicates the state equivalence/similarity and thus is commonly used for MDP (Markov Decision Process) model reduction/minimization. Though driven by various motivations, the use of bisimulation metric in RL can be thought of as learning suitable measurement in the representation space. These work trains the mapping functions (encoders) to minimize the discrepancy between the representations of similar observation pairs in the latent representation space according to their bisimulation metric. Intuitively, the distance between two encoded observations should correspond to their "behavioral difference" that is jointly determined by the difference of their immediate rewards and the difference of the distributions over their next states. Although effective, the bisimulation metric needs to compute the Wasserstein distance between distributions, which is computationally expensive.

Existing work only partially addressed the complexity issue, resulting in limitations in deploying the solutions. Gelada et al. (2019) require a stringent assumption that the learned representation is Lipschitz, so that the dual form of Wasserstein distance can be used to alleviate the computational complexity. Zhang et al. (2021) assume the state distribution is Gaussian, so that it can use euclidean distance of the Gaussian distribution in the latent space to compute the closed-form Wasserstein distance, thus optimizing the ℓ_1 distance between two representations. However, the inconsistency of ℓ_1 distance and euclidean distance may steer inaccurate approximation. Castro et al. (2021) develop a sample-based metric that does not contain Wasserstein distance. However, this design has the risk to fall in the failure mode of representation collapse since it violates the "zero

[*]Corresponding author.

self-distance" property.

Therefore, we argue that it is crucial to find a measurement in the representation space that is: (1) functionally equivalent to the bisimulation metric, (2) simpler in computational complexity, and (3) insusceptible to representation collapse issue. Intuitively, such a measurement, which plays the same role as bisimulation metric in training encoder, should improve the learning efficiency in continuous and high-dimensional observation space while maintaining the ability of measuring the "long-term behavior" of states and reserving good theoretical properties.

In this paper, we propose *Simple State Representation* (SimSR) to achieve task-relevant latent state representation while satisfying the aforementioned three features. Specifically, we adapt bisimulation metric with the proposed SimSR update operator which is based on the cosine distance to measure the behavioral difference of two states in the representation space. SimSR operator theoretically guarantees that the observations are embedded in a better representation space. Since SimSR operator involves the cosine distance, the mapped latent space can be viewed as a unit sphere, where the "behavioral difference" between representations is characterized by the cosine distance of the encoded observations. Besides, since cosine distance has a guarantee of "zero self-distance", SimSR theoretically avoids the case that states with different values collapse to the same representation in our proposed framework.

To avoid the computation of expectation, SimSR operator applies sample-based computation, which requires the sampling of the next state representations. This can be achieved by either 1) sampling the next observations and encoding them into the latent representation space or 2) encoding the current observations into the latent space and learning the latent transition model explicitly to compute the next state representations. The former approach is more convenient in implementation but it may involve redundant information. In comparison, the latter is able to provide more compact transition dynamics of the entire environment, resulting in better sample efficiency. To provide a comprehensive study, we developed and experimented two types of SimSR update operator for each approach. For the latter operator, we utilized an ensemble version of Gaussian distribution to facilitate the latent transition modelling.

In theory, SimSR operator confines the mapping to a set of functions yielding a low-dimensional state representation space, where the distance between the state representations is consistent to the "behavioral difference" of the states. In practice, neural networks can be utilized to minimize the SimSR loss to find one of such mapping functions. More specifically, we utilized the momentum encoder (He et al. 2020) to stabilize the learning process.

Our contributions can be summarized as follows:

1. We devised SimSR operator and provided theoretical analysis to prove its effectiveness of learning robust representations. It can achieve equivalent functionality while reducing the complexity by an order in comparison with bisimulation metric.
2. We designed the SimSR loss based on stochastic approximation methods to learn state representation. It can be easily adopted to any model-based or model-free RL approaches as an add-on.
3. We empirically experimented SimSR and demonstrated that it can (1) achieve comparable and better performance over existing methods on a set of continuous control benchmarks, (2) learn more robust representation against task-irrelevant information, and (3) potentially generalize to unseen tasks.

Related Work

Representation learning in RL

A good representation of state is crucial to the success of applying RL. There exist previous work (Lange and Riedmiller 2010; Lange, Riedmiller, and Voigtländer 2012; Lee et al. 2020a; Yarats et al. 2021b) devoted to training deep auto-encoders with reconstruction loss to learn low-dimensional representations and improve the policy performance in visual reinforcement learning. Some other RL methods (Shelhamer et al. 2017; Hafner et al. 2019; Lee et al. 2020b) proposed to learn state representation from predictive loss, yet suffering from the additional complexity of balancing various auxiliary losses. More recently, some work (Laskin, Srinivas, and Abbeel 2020; Yarats, Kostrikov, and Fergus 2021; Yarats et al. 2021a; Stooke et al. 2021) has used contrastive losses to learn representations and achieved considerable performance improvement. However, all these methods capture all features in the observation space no matter they are relevant to the downstream tasks or not, which in turn may degrade the policy performance.

Bisimulation

Early work on overcoming the curse of dimensionality (Givan, Dean, and Greig 2003; Li, Walsh, and Littman 2006) defined equivalence relations of the states to aggregate states and reduce the system complexity. However, the equivalence defined on the stochastic processes is impractical since it requires the transition distribution to be exactly the same. Ferns, Panangaden, and Precup (2004); Ferns and Precup (2014) addressed the issue with bisimulation metrics, which measure how similar two states are and can also be used as a distance function to aggregate states more easily. Recently, Gelada et al. (2019) have explored the use of a variety of metrics and showed their l_2 distance representation has the upper bounds as the bisimulation distance, while requiring the stringent assumption of the representations being Lipschitz. In consideration that bisimulation metric may result in improper comparison between states, Castro (2020) developed on-policy bisimulation metric, which unfortunately still suffers the difficulty of computing the intractable Wasserstein distance. As the successors of on-policy bisimulation metric, Zhang et al. (2021) modeled the latent dynamic transition as Gaussian distribution to alleviate the computational complexity of Wasserstein distance, while Castro et al. (2021) introduced the MICo distance to approximate the bisimulation metric. Both of them are restrictive as either having the stringent assumption and the distance inconsistency, or being prone to the failure mode of representation collapse. In comparison, we devised a simple and flexible measurement to characterize the behavioral similarity between states without using the Wasserstein distance, and we proved that our approach can address the aforementioned issues both theoretically and experimentally.

Preliminaries

This section explains the basic notations in modelling RL problems and bisimulation.

Problem setting

This paper focuses on learning from the environment with infinite image-based observations. We consider the underlying RL problems as a **block Markov decision process** (BMDP) (Du et al. 2019), which is represented as a tuple $\langle \mathcal{X}, \mathcal{S}, \mathcal{A}, r, \phi, q, \mathcal{P}, \mathcal{P}_0, \gamma \rangle$ with a finite unobservable state space $\mathcal{S}$, action space $\mathcal{A}$, and possibly infinite but observable space $\mathcal{X}$. With the assumption that each observation $\mathbf{x} \in \mathcal{X}$ uniquely determines its generating state $\mathbf{s} \in \mathcal{S}$, we can obtain the latent state regarding its observation by a projection function $\phi(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{S}$. Therefore, $\mathbf{s}$ and $\phi(\mathbf{x})$ can be used interchangeably. The dynamics of a BMDP is described by the initial latent state distribution $\mathcal{P}_0$ and the state-transition probability function $\mathcal{P}$ which decides the next latent state of the agent $\mathbf{s}' \sim \mathcal{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a})$. The corresponding transition function under the observation space is defined as $\mathbf{x}' \sim \hat{\mathcal{P}}(\mathbf{x}'|\mathbf{x}, \mathbf{a})$, where $\hat{\mathcal{P}}(\mathbf{x}'|\mathbf{x}, \mathbf{a}) = q(\mathbf{x}'|\mathbf{s}')\mathcal{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ and q indicates the probability of an latent state $\mathbf{s}$ presenting itself as an observation $\mathbf{x}$. The agent in a latent state $\mathbf{s} \in \mathcal{S}$ selects an action $\mathbf{a} \in \mathcal{A}$ according to the policy $\pi(\mathbf{a}|\mathbf{s})$.

Following the assumption that the projections from the observation space to the latent state space are deterministic and the environment always provides the reward function depending on the observations, the performance of the state-action pair is quantified by the reward function $r(\mathbf{x}, \mathbf{a})$ given by the environment. Similarly, the latent space reward $r(\mathbf{s}, \mathbf{a})$ can be defined based on the environment reward $r(\mathbf{x}, \mathbf{a})$ with the transition models. γ is a discount factor ($0 < \gamma < 1$) which quantifies how much value we weigh for future rewards. The goal of the agent is to find the optimal policy $\pi(\mathbf{a}|\mathbf{s})$ to maximize the expected reward $\mathbb{E}_{\mathbf{x}_0, \mathbf{a}_0, \ldots} [\sum_{t=0}^{\infty} \gamma^t r(\mathbf{x}_t, \mathbf{a}_t)]$. Besides, we approximate stacked pixel images as the observations in the context.

Bisimulation metric

Bisimulation measures equivalence relations on MDPs with a recursive form: two states are deemed equivalent if they share the equivalent distributions over the next equivalent states and they have the same immediate reward (Larsen and Skou 1989; Givan, Dean, and Greig 2003). However, since bisimulation considers equivalence for all actions, including bad ones, it commonly results in "pessimistic" outcomes. Instead, Castro (2020) developed **π-bisimulation** which removes the requirement of considering each action and only needs to consider the actions induced by a policy π.

Definition 1. (Castro 2020) Given an MDP $\mathcal{M}$, an equivalence relation $E^\pi \subseteq \mathcal{S} \times \mathcal{S}$ is a π-bisimulation relation if whenever $(\mathbf{s}, \mathbf{u}) \in E^\pi$ the following properties hold:

1. $r(\mathbf{s}, \pi) = r(\mathbf{u}, \pi)$
2. $\forall C \in \mathcal{S}_{E^\pi}, P(C|\mathbf{s}, \pi) = P(C|\mathbf{u}, \pi)$

where $\mathcal{S}_{E^\pi}$ is the state space $\mathcal{S}$ partitioned into equivalence classes defined by E^π. Two states $\mathbf{s}, \mathbf{u} \in S$ are **π-bisimilar** if there exists a **π-bisimulation relation** E^π such that $(\mathbf{s}, \mathbf{u}) \in E^\pi$.

However, π-bisimulation is still too stringent to be applied at scale as π-bisimulation relation emphasizes the equivalence is a binary property: either two states are equivalent or not, thus becoming too sensitive to perturbations in the numerical values of the model parameters. The problem becomes even more prominent when deep frameworks are applied. Therefore, we seek a suitable "measurement" of bisimulation to estimate "how bisimilar" two states are.

Thereafter, Castro (2020) proposed a **π-bisimulation metric** to leverage the absolute value between the immediate rewards w.r.t. two states and the 1-Wasserstein distance ($\mathcal{W}_1$) between the transition distributions conditioned on the two states and the policy π to formulate such measurement.

Theorem 1. *(Castro 2020) Define $\mathcal{F}^\pi : \mathcal{M} \rightarrow \mathcal{M}$ by $\mathcal{F}^\pi(d)(\mathbf{s}, \mathbf{u}) = |r^\pi_{\mathbf{s}} - r^\pi_{\mathbf{u}}| + \gamma \mathcal{W}_1(d)\left(\mathcal{P}^\pi_{\mathbf{s}}, \mathcal{P}^\pi_{\mathbf{u}}\right)$, where $r^\pi_{\mathbf{s}} = \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{a}|\mathbf{s}) r(\mathbf{s}, \mathbf{a})$ and $\mathcal{P}^\pi_{\mathbf{s}} = \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{a}|\mathbf{s}) \mathcal{P}(\cdot|\mathbf{s}, \mathbf{a})$. Then $\mathcal{F}^\pi$ has a least fixed point $d^\pi_\sim$, and $d^\pi_\sim$ is a π-bisimulation metric.*

Theorem 1 states it is feasible to simply apply the update operator iteratively $\mathcal{F}^\pi$ on an arbitrary initialized d in a dynamic programming fashion via sampling approaches and function approximators, such as neural networks, to compute $d^\pi_\sim$. Although the Wasserstein distance is a powerful metric to calculate the distance between two probability distributions, it requires to enumerate all states which is impossible in RL tasks of continuous state space. Thus, Castro (2020) only considered the deterministic MDP problems to avoid the computation of Wasserstein distance, and rewrite the operator $\mathcal{F}^\pi$ with a form of $\mathcal{F}^\pi(d)(\mathbf{s}, \mathbf{u}) = |r^\pi_{\mathbf{s}} - r^\pi_{\mathbf{u}}| + \gamma d\left(s', u'\right)$, where s' and u' are the deterministic next states of an agent starting from state $\mathbf{s}$ and $\mathbf{u}$, respectively, under the policy π.

Various extensions have been proposed (Gelada et al. 2019; Zhang et al. 2021). To reduce the computational complexity, they introduced some stringent assumptions, making them impractical for many real-world applications. In comparison, we focus on designing a more flexible and easily applicable distance measurement, which can be used to learn representations without complicated assumption.

Simple State Representation (SimSR) Framework

This section defines Simple State Representation (SimSR) operator. SimSR operators are proven to have fixed points regarding the "distance" between representations in the mapped latent space which properly measures the behavioral difference between states. Consequently, SimSR loss is introduced to construct the mapping function and the state representation from a learning perspective.

SimSR Operator

Our goal is to learn robust latent state representation to benefit RL with boosting efficiency. The intuition is the bisimular states should have "similar" or even the same representations in a low-dimensional feature space. Therefore, the bisimulation metric is applied to guide the learning of the mapping function. As the bisimulation metric needs to be learned iteratively (Theorem 1), we propose to learn the mapping function together with the distance function $\tilde{G}(\mathbf{x},\mathbf{y})$ of the form $G(\phi(\mathbf{x}),\phi(\mathbf{y}))$, where ϕ denotes the mapping function.

The distance between two latent states, which is defined with cosine distance (normalized dot product distance) as the base "metric", is measured as:

$$G(\phi(\mathbf{x}),\phi(\mathbf{y})) = 1-\cos(\phi(\mathbf{x}),\phi(\mathbf{y})) = 1-\frac{\phi(\mathbf{x})^T \cdot \phi(\mathbf{x})}{\|\phi(\mathbf{x})\| \cdot \|\phi(\mathbf{x})\|}. \tag{1}$$

It is worth noting that Theorem 1 assure the base distance always being metric in update iteration. However, cosine distance is not a real metric since it violates triangle inequality. In fact, we favor the base distance that is more suitable for measurement rather than only being metric strictly. We therefore deliberately construct the update operator $\mathscr{F}^\pi$ which is based on cosine distance but still has a fixed point.

Theorem 2. *Let the reward $r^\pi_\mathbf{x} = \sum_{\mathbf{a}\in\mathcal{A}} \pi(\mathbf{a}|\mathbf{s})r(\mathbf{x},\mathbf{a})$, the transition $\hat{\mathcal{P}} = \sum_{\mathbf{a}\in\mathcal{A}} \pi(\mathbf{a}|\mathbf{s})\hat{\mathcal{P}}(\cdot|\mathbf{x},\mathbf{a})$, and $\tilde{G}(\boldsymbol{x},\boldsymbol{y}) = G(\phi(\boldsymbol{x}),\phi(\boldsymbol{y}))$ for all functions $\phi : \mathcal{X} \to \mathcal{S}$ and $G : \mathcal{S}\times\mathcal{S} \to \mathbb{R}$. Given a policy π, Simple State Representation (SimSR) operator $\mathscr{F}^\pi : \mathbb{R}^{\mathcal{S}\times S} \to \mathbb{R}^{\mathcal{S}\times S}$ as:*

$$\begin{aligned}\mathscr{F}^\pi \tilde{G}(\mathbf{x},\mathbf{y}) =& |r^\pi_\mathbf{x} - r^\pi_\mathbf{y}| + \\ & \gamma \mathbb{E}_{\mathbf{x}'\sim\hat{\mathcal{P}}^\pi_\mathbf{x},\mathbf{y}'\sim\hat{\mathcal{P}}^\pi_\mathbf{y}}[\tilde{G}(\mathbf{x}',\mathbf{y}')]\end{aligned} \tag{2}$$

has a fixed point $\tilde{G}^\pi$[1].

SimSR operator $\mathscr{F}^\pi$ can be applied iteratively on an arbitrary initialized ϕ to update the corresponding distance $\tilde{G}$, leading to a set of mapping functions, which can serve as the encoders to embed the observations into the state representations while satisfying the constraints regarding the distance.

In practice, the second term of the right-hand side in Eq. 2 is computed by sample-based approximation to avoid computing W-distance as most existing work. The motivations and benefits of $\mathscr{F}^\pi$ design are outlined below:

Unit length With the cosine distance as the base "metric" in SimSR, all learned state features will be scaled to unit length. Without normalization, the distance may be dominated by the scale of state features, resulting in negative feedback to state representations and disturbance to value approximation and policy learning. Cosine distance effectively restricts the output space to the unit sphere and guarantees that all features contribute equally to the resulting distance. Bojanowski and Joulin (2017); Wang et al. (2017); Wang and Isola (2020) demonstrated that normalized outputs lead to superior representations in their representation learning work. Our experiments also proved that the unified features perform better than those that not unified.

[1]All proofs are provided in Appendix.

Zero self-distance "Representation collapse" in RL means that two states (observations) with different values are collapsed to the same representation, e.g., $\phi(\mathbf{x}) \to 0$ for all $\mathbf{x}$. State representations are prone to representation collapse when they are jointly trained with value functions and policy functions without providing ground states. Let ϕ^π be the mapping function set corresponding to the fixed point $\tilde{G}^\pi$. With the zero self-distance property guaranteed by cosine distance, Theorem 3 assures that collapse issue is negligible for ϕ^π as two observations are encoded with the same representation only if they have same values:

Theorem 3. *$\forall \mathbf{x},\mathbf{y} \in \mathcal{X}$ and for any policy $\pi \in \Pi$,*

$$|V^\pi(\mathbf{x}) - V^\pi(\mathbf{y})| \le \tilde{G}^\pi(\mathbf{x},\mathbf{y}) = G(\phi^\pi(\mathbf{x}),\phi^\pi(\mathbf{y})). \tag{3}$$

Low computational complexity SimSR operator $\mathscr{F}^\pi$ does not compute Wasserstein distance, which drastically simplifies the update procedure. In comparison with π-bisimulation metric, SimSR reduces the computational complexity by an order.

Proposition 1. *The fixed point of π-bisimulation metric can be computed up to a prescribed degree of accuracy δ in $\tilde{O}\left(|\mathcal{S}|^5 \log\delta/\log\gamma\right)$ operations with respect to L^∞ norm, while the fixed point of SimSR distance $\tilde{G}^\pi$ can be computed in $O\left(|\mathcal{S}|^4 \log\delta/\log\gamma\right)$ operations.*

SimSR with latent state dynamics

The target problems of the paper can be modeled as a block MDP with an infinite observation space and a finite latent state space, where the dynamics can be described by either latent state transition function $\mathcal{P}$ or observation transition function $\hat{\mathcal{P}}$. SimSR operator defined in Theorem 2 is computed by sample-based approximation with the observations sampled from $\hat{\mathcal{P}}$ which is provided by the raw environment. Although sampling from an infinite observation space is convenient, this strategy may inject considerable redundant information to the representation learning and the policy learning process. Therefore, we explicitly construct a more compact dynamics model on the limited latent state space. Consequently, we redefine SimSR operator with the latent state dynamics. The learned latent state dynamics are expected to provide agents more diverse knowledge of the goals, leading to better convergence and robustness in the representation learning and policy learning.

Specifically, we first construct the latent state dynamics model $\mathcal{P}(\cdot|\phi(\mathbf{x}),\mathbf{a})$, then encode the observation $\mathbf{x}$ to its latent state $\phi(\mathbf{x})$ and sample the corresponding next latent state from the dynamics model $\mathbf{s}' \sim \mathcal{P}(\cdot|\phi(\mathbf{x}),\mathbf{a})$, and finally leverage cosine distance to acquire SimSR distance. Theorem 4 defines the updated operator and proves its guaranteed convergence.

Theorem 4. *Given a policy π, let SimSR operator $\mathbb{F}^\pi : \mathbb{R}^{\mathcal{S}\times S} \to \mathbb{R}^{\mathcal{S}\times S}$ be*

$$\mathbb{F}^\pi \tilde{G}(\mathbf{x},\mathbf{y}) = |r^\pi_\mathbf{x} - r^\pi_\mathbf{y}| + \gamma \mathbb{E}_{\mathbf{s}'\sim\mathcal{P}^\pi_{\phi(\mathbf{x})},\mathbf{u}'\sim\mathcal{P}^\pi_{\phi(\mathbf{y})}}[G(\mathbf{s}',\mathbf{u}')]. \tag{4}$$

If latent dynamics are specified, $\mathbb{F}^\pi$ has a fixed point $\tilde{\mathbb{G}}^\pi$.

Latent state dynamics modelling

Following (Zhang et al. 2021), we develop an ensemble version of probabilistic dynamics $\{\mathcal{P}_k(\cdot|\phi(\mathbf{x}), \mathbf{a})\}_{k=1}^{K}$. Typically, each probabilistic transition dynamics model can be represented as $\mathcal{P}_{\theta_k}(\cdot|\phi(\mathbf{x}), \mathbf{a}) = \mathcal{N}(\mu_{\theta_k}, \sigma_{\theta_k})$. At the training step, we update the parameters of dynamics models by Gaussian negative log-likelihood loss function:

$$\mathcal{L}_{\mathcal{P}}(\theta_i) = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{\log \sigma^2_{\theta_k}(\phi(\mathbf{x}))}{2} + \frac{(\phi(\mathbf{x}') - \mu_{\theta_k}(\phi(\mathbf{x})))^2}{2\sigma^2_{\theta_k}(\phi(\mathbf{x}))}\right], \tag{5}$$

where $i \in \{1, 2, ..., K\}$. Since the probabilistic models are all randomly initialized, though they share the same gradient, they generally acquire different parameters after training. Therefore, the ensemble model can estimate the uncertainty of the environments in some sense and is more suitable for approximating the MDPs that with a high degree of stochasticity. At the inference step, to apply theorem 4, we randomly sample one of the K probabilistic dynamics to perform the sampling of next latent state $\mathbf{s}'$ and $\mathbf{u}'$.

Representation Learning with SimSR loss

We adopt two convolution layer with a fully connected layer as the encoder module ϕ parameterized by ψ. After encoding the observations, we then ℓ_2-normalize the output of the fully connected layer to scale the features to unit length.

To avoid the computation of the expected distance in Eq. 2 and Eq. 4 regarding the probabilistic models, we, inspired by temporal-difference learning (Sutton and Barto 1998), estimate the target of the distance (the R.H.S of these two equations) by stochastic estimation methods : a sample distance between next latent states is used in place of the real expected distance. However, as we update the estimation of the distance between states based on estimating the distance of successor states, such bootstrapping steps may introduce bias and reduce the representations' consistency. Therefore, we adopt the momentum encoder (He et al. 2020) to stabilize the representation of the states.

Specifically, we draw batch of state pairs and minimise the mean square error (MSE) between both sides of SimSR to guide the learning of the encoder:

$$\begin{aligned}\mathcal{L}_{\phi}(\psi) =&\mathbb{E}_{(\mathbf{x},r(\mathbf{x},\mathbf{a}),\mathbf{a},\mathbf{x}'),(\mathbf{y},r(\mathbf{x},\mathbf{a}),\mathbf{a},\mathbf{y}')\sim\mathcal{D}}\big(\\ &G(\phi_\psi(\mathbf{x}), \phi_\psi(\mathbf{y})) - \hat{G}\big)^2\end{aligned} \tag{6}$$

where $\mathcal{D}$ is the replay buffer and:

$$\hat{G} = \begin{cases} |r^{\pi}_{\mathbf{x}} - r^{\pi}_{\mathbf{y}}| + \gamma G(\phi_{\hat{\psi}}(\mathbf{x}'), \phi_{\hat{\psi}}(\mathbf{y}')), \\ \text{if applying Theorem 2.} \\ |r^{\pi}_{\mathbf{x}} - r^{\pi}_{\mathbf{y}}| + \gamma G(\mathbf{s}', \mathbf{u}') \\ \text{if applying Theorem 4.} \end{cases} \tag{7}$$

In Eq. 7, $\mathbf{s}' \sim \mathcal{P}_i(\cdot|\phi_{\hat{\psi}}(\mathbf{x}), \mathbf{a})$, $\mathbf{u}' \sim \mathcal{P}_i(\cdot|\phi_{\hat{\psi}}(\mathbf{y}), \mathbf{a})$, i is sampled from uniform distribution $U(1, K)$, and the parameter $\hat{\psi}$ of momentum encoder $\phi_{\hat{\psi}}$ is the exponential moving average (EMA) of the parameter ψ of the encoder ϕ_ψ:

$$\hat{\psi} \leftarrow m\hat{\psi} + (1 - m)\psi, \tag{8}$$

where $m \in [0, 1)$ is the momentum coefficient.

Technically, as both ℓ_1 distance (the reward difference) and cosine distance (next state difference) in $\hat{G}$ can be easily extended to matrix operation, we can measure the cross-correlation matrix between the batch-wise state representations and optimize it to accelerate the representation learning. With Theorem 2, the loss function can be rewritten as:

$$\begin{aligned}\mathbb{E}_{B_j}[\mathcal{L}_\phi(\psi)] =&\mathbb{E}_{B_j}\Bigg[(1 - \frac{\phi_\psi(X)^T \cdot \phi_\psi(X)}{\|\phi_\psi(X)\| \cdot \|\phi_\psi(X)\|}) - \\ &(|R^T - R| + \gamma(1 - \frac{\phi_{\hat{\psi}}(X')^T \cdot \phi_{\hat{\psi}}(X')}{\|\phi_{\hat{\psi}}(X')\| \cdot \|\phi_{\hat{\psi}}(X')\|}))\Bigg]^2,\end{aligned} \tag{9}$$

where B_j is a sample batch, X, X', and R are the corresponding batch-wise observations, batch-wise next observations, and batch-wise rewards, respectively. In contrast, DBC (Zhang et al. 2021) develops its loss function at sample-wise level, resulting in lower computational efficiency.

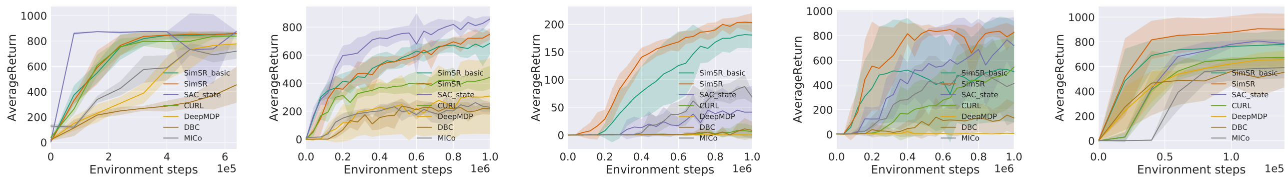
The proposed representation learning approach can be viewed as an auxiliary task with SimSR objective to learn an encoder, on top of which the policy and value network can be easily built. Meanwhile, since the encoder, the policy network, and the value network can be trained simultaneously, our approach can be easily adopted to any model-based or model-free RL methods as an add-on. In practice, we build agents by combining our approach with soft actor critic (SAC) algorithm (Haarnoja et al. 2018) to devise a practical reinforcement learning method. The training process is illustrated by Algorithm 1.

Algorithm 1: SimSR algorithm

for Time $t = 0$ to ∞ **do**
  Encode state $\mathbf{s}_t = \phi(\mathbf{x}_t)$
  Execute action $\mathbf{a}_t \sim \pi(\mathbf{s}_t)$
  Record data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}, r_{t+1}\}$
  Sample batch $B_i \sim \mathcal{D}$
  Train encoder: $\mathbb{E}_{B_i}[\mathcal{L}_\phi(\psi)]$ ▷ Eq. 10
  Train dynamics: $\mathbb{E}_{B_i}[\mathcal{L}_{\mathcal{P}}(\theta_j)]$, where $j \in \{1, ..., K\}$ ▷ Eq. 5
  Train value function: $\mathbb{E}_{B_i}[\mathcal{L}(V)]$ ▷ SAC algorithm
  Train policy function: $\mathbb{E}_{B_i}[\mathcal{L}(\pi)]$ ▷ SAC algorithm
end for

We notice that in some tasks, the critic loss explodes after few iterations and the value function diverges. This happens when agents start learning new behaviors and experiencing new states where the scale of MSE loss may increase exponentially large. Since both representation learning and reinforcement learning use the MSE loss, the combination of them aggravates the volatility, thus causing the explosion. Therefore, we propose to use Huber loss as the surrogate loss function to alleviate the explosion problem:

$$\mathcal{L}_\phi(\psi) = \begin{cases} \mathbb{E}_{(\mathbf{x},r(\mathbf{x},\mathbf{a}),\mathbf{a},\mathbf{x}'),(\mathbf{y},r(\mathbf{x},\mathbf{a}),\mathbf{a},\mathbf{y}')\sim\mathcal{D}}\big[\frac{1}{2}(K)^2\big], \\ \text{for } |K| \leq 1. \\ |K| - \frac{1}{2}, \\ \text{otherwise.} \end{cases} \tag{10}$$



(a) Cartpole Swingup (b) Cheetah Run (c) Hopper Hop (d) Hopper Stand (e) Finger Spin

Figure 1: Performance comparison on 5 DMC tasks over 5 seeds with one standard error shaded in the default setting. For each seed, the average return is computed every 10,000 training steps, averaging over 10 episodes. The horizontal axis indicates the number of environment steps. The vertical axis indicates the average return.

Table 1: Comparison of key features in different algorithms. “Dis. consistency” means that the distance computed in the representation space is consistent with the base “metric” that is used in behavioral difference. “-” means “does not apply”.

	Dis. consist	Unit length	Zero self-dis.	Learn dynamics
SimSR	✓	✓	✓	✓
DBC	✗	✗	✓	✓
MICo	✓	✗	✗	✗
DeepMDP	-	✗	-	✓

where $K = G(\phi_\psi(\mathbf{x}), \phi_\psi(\mathbf{y})) - \hat{G}$.

The most related methods to our work are DBC (Zhang et al. 2021), MICo (Castro et al. 2021)and DeepMDP (Gelada et al. 2019). Table 1 provides the comparison of their key features. Detailed comparison with previous work is provided in Appendix.

Experiments

We experimented to investigate the following questions: (1) In comparison with state-of-the-art algorithms, does SimSR have a better performance in terms of sample efficiency? (2) Can SimSR learn robust state representation? (3) How is the generalization performance of the learned representation?

Accordingly, we first evaluated our method in several standard control tasks from the DeepMind control (DMC) suite (Tassa et al. 2018). We then evaluated the robustness of our method to test if it can handle more realistic and complicated scenarios, where we replaced the background of the environment with natural video as distractors. Finally, we tested the generalization performance of the state representation in unseen tasks.

Effectiveness and robustness

Settings We compared the proposed approaches, SimSR_basic (Theorem 2) and SimSR (Theorem 4), against the selected baselines. As one goal is to close the performance gap between the physical-state based methods and images based methods, we first compared with SAC_state, which is non-visual SAC (Haarnoja et al. 2018) accepting physical features (e.g., position, angle, height) as inputs. We then compared with CURL (Laskin, Srinivas, and Abbeel 2020), which is a recent state-of-the-art image-based RL method that learns latent representation

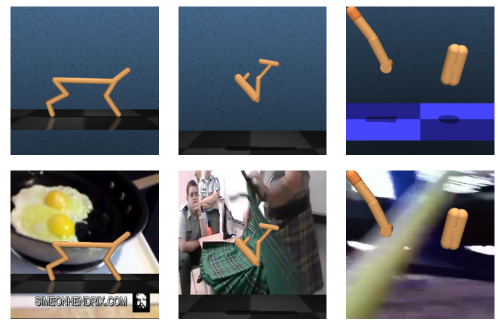
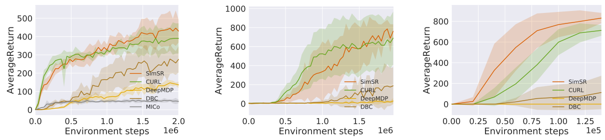


Figure 2: Pixel observations in DMC in the default setting (top row) and in the natural video background settings (bottom row) for cheetah (left column), hopper (middle column), and finger (right column).

via data augmentation and contrastive learning. Thereafter, we compared with the approaches that are most relevant to ours, including DBC (Zhang et al. 2021), MICo (Castro et al. 2021), and DeepMDP (Gelada et al. 2019)[2]. Due to the page limit and the absence of some results of DBC and DeepMDP, we provide more comprehensive performance comparisons on `Ball_In_Cup_Catch`, `Pendulum_Swingup`, and additional environments in Appendix. We keep all hyper-parameters of the algorithm fixed throughout experiments except the action repeat which follows the convention to ensure a fair comparison. The settings of all hyper-parameters and architectures are also provided in Appendix.

Experiments with default setting As shown in the top row of Figure 2, the default setting, which is provided by DMC, has simple backgrounds for the pixel observations. Figure 1 demonstrates that SimSR outperforms all selected state-of-the-art methods by a large margin on 3 of 5 tasks (`Finger_Spin`, `Hopper_Hop`, `Hopper_Stand`) and remains competitive in the rest of 2 tasks. As discussed, MICo may encounter the representation collapse problem where different observations are encoded to the same embedding. Since the policy network and the value network are built on

[2]The results of DBC and DeepMDP are taken from (Zhang et al. 2021) with five seeds. Our code is available at https://github.com/bit1029public/SimSR



(a) Cheetah Run (b) Hopper Stand (c) Finger Spin

Figure 3: Performance comparisons on three DMC tasks over 5 seeds with one standard error shaded in the natural video setting.

top of the encoder, they both may collapse and result in inaccurate learning in MICo.

The results prove that SimSR can generally close the gap between image observation based RL methods and physical state based RL methods regarding the downstream performance without any data augmentation techniques. Moreover, in comparison with the approaches with bisimulation metric, SimSR consistently outperforms MICo, DBC, and DeepMDP on all tasks, showing the effectiveness of the proposed SimSR distance. SimSR performs better than SimSR_basic in most cases, confirming that learning latent transitions can provide agents more diverse knowledge of the goals and achieve better sample efficiency.

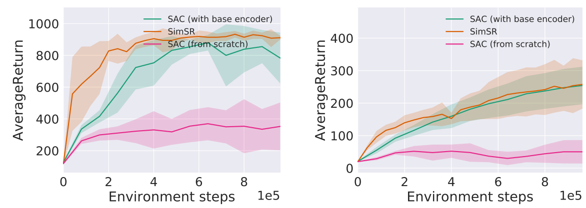
Experiments with natural video setting To investigate the robustness of representation learning against task-irrelevant information, we used natural videos from the Kinetics dataset (Kay et al. 2017), as shown in the bottom row of Figure 2, to substitute the default simple background in the experiments. Figure 3 demonstrates that SimSR is capable of filtering out task-irrelevant information and improving, thus achieving better performance against other bisimulation-based methods.

Since CURL combines contrastive learning with data augmentation, its contrastive loss incorporates the prior knowledge of the environment into the encoder. Therefore, CURL is able to distinguish task irrelevant signals in visual spaces, thus having better performance than other methods without encoding prior knowledge. Still, SimSR outperforms CURL.

Generalization experiments

To evaluate the generalization capability, we tested the learned representation on different tasks, including `Walker_Walk`, `Walker_Stand`, and `Walker_Run`, which share the same observation space but have different reward functions.

We first trained SimSR on `Walker_Walk` until convergence to learn the base encoder. To test its generalization capability, We then trained two SAC agents with image-based inputs on `Walker_Stand` and `Walker_Run`, respectively. One SAC agent utilized the base encoder learned in `Walker_Walk` without further updates during the training process, while the other SAC agent learned its own encoder from scratch. For comparison, we also trained SimSR agents in both tasks to represent the specialized encoders in each task. Figure 4 illustrates the learning curves of all agents in both tasks.



(a) Walker Stand (b) Walker Run

Figure 4: The comparison between SimSR agent and two types of SAC agents on `Walker Stand` and `Walker Run` tasks. The tasks share similar image observations, but have different reward.

The results demonstrate that SimSR agents perform the best in both tasks as they are fine-tuned to their specific environments. The use of learned representations in generalized tasks, as represented by the SAC agent with the learned encoder, achieves very close performance next to the ideal cases of specific models. Moreover, the SAC agent with the learned encoder significantly outperforms the SAC agent that learned from scratch, demonstrating the generalization capability of SimSR. The results empirically prove the potential of SimSR in learning latent representations which can be generalized well over the tasks that share the same observation space but with different reward functions and goals.

Ablation study

We investigated the importance of ℓ_2 normalization, which guarantees the properties of unit length and zero self-distance in the results. Without ℓ_2 normalization, the agent cannot produce acceptable performance. This finding empirically underpins our theoretical analysis: if the distance is not unit length or zero self-distance, the representation may easily collapse. Besides, we also analyzed the effectiveness of the ensemble transition models. The results shows that even with a single probabilistic model, the agent can still achieve remarkable performance, possibly with higher variances. This confirms our hypothesis that the ensemble model can estimate the uncertainty better, thus leading to more stable representations. All results are presented in Appendix.

Conclusion and Future Work

In the paper, we presented SimSR, a distance-based representation learning method, which can be easily adopted to any model-based or model-free RL methods. We theoretically proved that it can achieve equivalent functionality while reducing the complexity by an order in comparison with bisimulation metric. The experiments demonstrated that SimSR is capable of learning robust and generalizable state representation and further enhancing policy learning. The results show that we improved the performance considerably over existing image-based RL methods.

Outside the line of this research, learning state representation with data augmentation can also improve the effectiveness of extracting task-relevant signals. One important future work is to combine two lines of approaches.

Acknowledgments

This work has been partially supported by NSFC under Grant (U19B2020 and 61772074) and National Key R&D Program of China under Grant (2017YFB0803300).

References

Allen, C.; Parikh, N.; Gottesman, O.; and Konidaris, G. 2021. Learning Markov State Abstractions for Deep Reinforcement Learning. *CoRR*, abs/2106.04379.

Anand, A.; Racah, E.; Ozair, S.; Bengio, Y.; Côté, M.; and Hjelm, R. D. 2019. Unsupervised State Representation Learning in Atari. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 8766–8779.

Bojanowski, P.; and Joulin, A. 2017. Unsupervised Learning by Predicting Noise. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 517–526. PMLR.

Castro, P. S. 2020. Scalable Methods for Computing State Similarity in Deterministic Markov Decision Processes. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 10069–10076. AAAI Press.

Castro, P. S.; Kastner, T.; Panangaden, P.; and Rowland, M. 2021. MICo: Learning improved representations via sampling-based state similarity for Markov decision processes. *CoRR*, abs/2106.08229.

Du, S. S.; Krishnamurthy, A.; Jiang, N.; Agarwal, A.; Dudík, M.; and Langford, J. 2019. Provably efficient RL with Rich Observations via Latent State Decoding. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 1665–1674. PMLR.

Fan, J.; and Li, W. 2021. Robust Deep Reinforcement Learning via Multi-View Information Bottleneck. *CoRR*, abs/2102.13268.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for Finite Markov Decision Processes. In Chickering, D. M.; and Halpern, J. Y., eds., *UAI '04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, July 7-11, 2004*, 162–169. AUAI Press.

Ferns, N.; and Precup, D. 2014. Bisimulation Metrics are Optimal Value Functions. In Zhang, N. L.; and Tian, J., eds., *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, 210–219. AUAI Press.

Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; and Bellemare, M. G. 2019. DeepMDP: Learning Continuous Latent Space Models for Representation Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2170–2179. PMLR.

Givan, R.; Dean, T. L.; and Greig, M. 2003. Equivalence notions and model minimization in Markov decision processes. *Artif. Intell.*, 147(1-2): 163–223.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic Algorithms and Applications. *CoRR*, abs/1812.05905.

Hafner, D.; Lillicrap, T. P.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning Latent Dynamics for Planning from Pixels. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2555–2565. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 9726–9735. IEEE.

Jaderberg, M.; Mnih, V.; Czarnecki, W. M.; Schaul, T.; Leibo, J. Z.; Silver, D.; and Kavukcuoglu, K. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *CoRR*, abs/1705.06950.

Lange, S.; and Riedmiller, M. A. 2010. Deep auto-encoder neural networks in reinforcement learning. In *International Joint Conference on Neural Networks, IJCNN 2010, Barcelona, Spain, 18-23 July, 2010*, 1–8. IEEE.

Lange, S.; Riedmiller, M. A.; and Voigtländer, A. 2012. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, June 10-15, 2012*, 1–8. IEEE.

Larsen, K. G.; and Skou, A. 1989. Bisimulation Through Probabilistic Testing. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, 344–352. ACM Press.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 5639–5650. PMLR.

Lee, A. X.; Nagabandi, A.; Abbeel, P.; and Levine, S. 2020a. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Lee, K.; Fischer, I.; Liu, A.; Guo, Y.; Lee, H.; Canny, J.; and Guadarrama, S. 2020b. Predictive Information Accelerates Learning in RL. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, L.; Walsh, T. J.; and Littman, M. L. 2006. Towards a Unified Theory of State Abstraction for MDPs. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2006, Fort Lauderdale, Florida, USA, January 4-6, 2006*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 2778–2787. PMLR.

Shelhamer, E.; Mahmoudieh, P.; Argus, M.; and Darrell, T. 2017. Loss is its own Reward: Self-Supervision for Reinforcement Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Stooke, A.; Lee, K.; Abbeel, P.; and Laskin, M. 2021. Decoupling Representation Learning from Reinforcement Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 9870–9879. PMLR.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press. ISBN 978-0-262-19398-6.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; de Las Casas, D.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; Lillicrap, T. P.; and Riedmiller, M. A. 2018. DeepMind Control Suite. *CoRR*, abs/1801.00690.

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.03748.

Wang, F.; Xiang, X.; Cheng, J.; and Yuille, A. L. 2017. NormFace: L_2 Hypersphere Embedding for Face Verification. In Liu, Q.; Lienhart, R.; Wang, H.; Chen, S. K.; Boll, S.; Chen, Y. P.; Friedland, G.; Li, J.; and Yan, S., eds., *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, 1041–1049. ACM.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 9929–9939. PMLR.

Warde-Farley, D.; de Wiele, T. V.; Kulkarni, T. D.; Ionescu, C.; Hansen, S.; and Mnih, V. 2019. Unsupervised Control Through Non-Parametric Discriminative Rewards. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2021a. Reinforcement Learning with Prototypical Representations. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 11920–11931. PMLR.

Yarats, D.; Kostrikov, I.; and Fergus, R. 2021. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yarats, D.; Zhang, A.; Kostrikov, I.; Amos, B.; Pineau, J.; and Fergus, R. 2021b. Improving Sample Efficiency in Model-Free Reinforcement Learning from Images. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 10674–10681. AAAI Press.

Zhang, A.; McAllister, R. T.; Calandra, R.; Gal, Y.; and Levine, S. 2021. Learning Invariant Representations for Reinforcement Learning without Reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.