Policy-Independent Behavioral Metric-Based Representation for Deep Reinforcement Learning

Weijian Liao[1], **Zongzhang Zhang**[1*], **Yang Yu**[1,2]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China
[2] Peng Cheng Laboratory, Shenzhen, 518055, China
liaowj@lamda.nju.edu.cn, {zzzhang, yuy}@nju.edu.cn

Abstract

Behavioral metrics can calculate the distance between states or state-action pairs from the rewards and transitions difference. By virtue of their capability to filter out task-irrelevant information in theory, using them to shape a state embedding space becomes a new trend of representation learning for deep reinforcement learning (RL), especially when there are explicit distracting factors in observation backgrounds. However, due to the tight coupling between the metric and the RL policy, such metric-based methods may result in less informative embedding spaces which can weaken their aid to the baseline RL algorithm and even consume more samples to learn. We resolve this by proposing a new behavioral metric. It decouples the learning of RL policy and metric owing to its independence on RL policy. We theoretically justify its scalability to continuous state and action spaces and design a practical way to incorporate it into an RL procedure as a representation learning target. We evaluate our approach on DeepMind control tasks with default and distracting backgrounds. By statistically reliable evaluation protocols, our experiments demonstrate our approach is superior to previous metric-based methods in terms of sample efficiency and asymptotic performance in both backgrounds.

1 Introduction

Deep reinforcement learning (RL) has made a series of major breakthroughs in recent years (Mnih et al. 2015; Silver et al. 2018; Vinyals et al. 2019). The quality of the state representation (Lesort et al. 2018) is crucial for the performance of RL algorithms, especially faced with high-dimensional observations, like images. An ideal representation is expected to contain non-redundant information that is sufficient for decision-making.

One way to obtain such representation is to define a *behavioral metric* between states and then use the state similarity quantified by this metric to shape the embedding space via an auxiliary representation learning objective built upon it. Essentially, such metric-based representation learning methods use specific reward sequences as a supervised signal for representation learning, so they are expected to help ignore task-irrelevant information in raw observations.

Bisimulation metrics (Ferns, Panangaden, and Precup 2004, 2011; Ferns and Precup 2014) are representative of such metrics.However, computing them exactly needs the knowledge of ground truth dynamics and the computation complexity prevents their direct use in large-scale problems. The π-bisimulation metric (Castro 2020) and the MICo distance (Castro et al. 2021) overcome the disadvantages by considering state similarity under the dynamics induced by a specific policy. Such modification allows them to be estimated from sampled transitions, leading to tractable approximation. They have successful application to representation learning in large-scale RL tasks (Zhang et al. 2021; Castro et al. 2021). Despite this, they are still problematic due to their dependence on an RL policy. When the RL policy is fixed, e.g., in policy evaluation settings, these two metrics are good enough. However, when the RL policy is learned online, the metrics may provide uninstructive or even erroneous learning signal on shaping the embedding space due to the policy's sub-optimality and consequently only have limited power to help policy learning. In light of this, it is desired to minimize the impact of policy changes on metric estimation as much as possible, while ensuring scalability.

To this end, we propose a new behavioral metric, named the *conservative state-action discrepancy*, to compute the similarity not only between state pairs but also between state-action pairs. We quantify the similarity by evaluating the maximum difference of the reward sequences that two states can achieve upon taking the same action sequence. It is a *policy-independent* behavioral metric from the definition and we give formal reasons on its advantages. We show theoretically that the conservative state-action discrepancy can be exactly derived in the dynamic programming setting and unbiasedly estimated in the sampling setting due to its equivalence to Q-value function in a certain constructed MDP. We also compare the conservative state-action discrepancy with the MICo distance in detail and reveal a new persective on the latter one. The corresponding learning algorithm is originated from Q-learning (Watkins 1989) and computes the relationship between two state-action pairs, so it is called *Q^2-learning*. Taking advantage of the equivalence relationship further, we propose to utilize the deterministic policy gradient algorithm (Silver et al. 2014) to address computation difficulties when faced with continuous action spaces. The approach enables to scale to large tasks as an auxiliary

*Corresponding Author

learning target for representation, as it can be combined with function approximators that satisfy the properties of the conservative state-action discrepancy.

We successfully validate its scalability and effectiveness via experiments on the DeepMind Control Suite (Tassa et al. 2018), which consists of tasks with high-dimensional observations and continuous action spaces. We further replace the default background with frames from natural videos that are unrelated to the control tasks to demonstrate the approach's robustness and ability to ignore task-unrelated information even if faced with distracting observations. Finally, we conduct multiple ablation experiments to illustrate the necessity of our method's design.

2 Background

Q^2-learning combines RL with a behavioral metric. Here we review RL and introduce the concept of behavioral metric.

2.1 Reinforcement Learning

A typical RL setting is formalized with a Markov Decision Process (MDP). An MDP is a tuple $\mathcal{M} = \langle \mathcal{X}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{X} \times \mathcal{A} \to \mathcal{P}(\mathcal{X})$ is the transition function, $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. For notional convenience we use P_x^a and r_x^a to denote the transition distributions and reward functions, respectively. Typically, the reward is bounded, and we denote the range as $[r_{\min}, r_{\max}]$. The return is defined as discounted sum of rewards $R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_{x_i}^{a_i}$. Agent acts according to a policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$. When $\mathcal{P}$ is a Dirac distribution, the policy becomes deterministic. The objective of RL is to find the optimal policy $\pi_\omega : \mathcal{X} \to \mathcal{P}(\mathcal{A})$ to maximize the expected return $J(\omega) = \mathbb{E}_{x_0 \sim \rho, a_t \sim \pi_\omega, x_{t+1} \sim P_{x_t}^{a_t}}[R_0]$, where ρ is the initial state distribution.

$Q^\pi(x, a) = \mathbb{E}_{x_i \sim P, a_i \sim \pi}[R_t | x_0 = x, a_0 = a]$ is the expected return taking action a at state x and following π after, known as the Q-value function. The Bellman (Bellman 1957) evaluation operator defines the connection of the Q-values between consecutive states: $B^\pi(Q)(x, a) \coloneqq r_x^a + \gamma \mathbb{E}_{x' \sim P_x^a, a' \sim \pi}[Q(x', a')]$, and we denote the unique fixed point of B^π as Q^π. The Bellman optimality operator is $B^*(Q)(x, a) \coloneqq r_x^a + \gamma \mathbb{E}_{x' \sim P_x^a} \max_{a'} Q(x', a')$, and we denote the unique fixed point of B^* as Q^*, called the optimal Q-value function.

For continuous control problems, actor-critic methods are widely applied. The policy, known as the actor, is updated via the *deterministic policy gradient theorem* (Silver et al. 2014): $\nabla_\omega J(\omega) = \mathbb{E}_{x \sim p_\pi}[\nabla_a Q^\pi(x, a)|_{a=\pi_\omega(x)} \nabla_\omega \pi_\omega(x)]$.

A specific algorithm is Soft Actor-Critic (SAC) (Haarnoja et al. 2018). It additionally uses the maximum entropy framework to enhance exploration and training stability. It maintains two parameterized functions Q_θ and π_ω to approximate Q-value function and policy, respectively. Here, θ and ω are parameters of the two networks.

The function Q_θ is updated by minimizing the following mean squared Bellman error:

$$\mathcal{L}_\theta = \mathbb{E}_{t \sim \mathcal{B}}[(Q_\theta(x, a) - (r_x^a + \gamma \mathcal{T}))^2], \tag{1}$$

where $t = (x, a, r, x')$ is a tuple sampled from replay buffer $\mathcal{B}$, containing state x, action a, reward r, and next state x'. The Bellman target $\mathcal{T}$ is defined as $\mathbb{E}_{a' \sim \pi_\omega}[Q_{\hat{\theta}}(x', a') - \alpha \log \pi_\omega(a'|x')]$, where $\hat{\theta}$ is the exponentially moving average of θ, the corresponding network $Q_{\hat{\theta}}$ serves as a stable target, and α is the temperature coefficient to control the contribution of the entropy term to the gradient.

The policy π_ω is updated by minimizing the KL-divergence between parameterized policy distribution and a Boltzmann distribution induced by Q-values:

$$\mathcal{L}_\omega = \mathbb{E}_{x \sim \mathcal{B}}[\mathbb{E}_{a \sim \pi_\omega}[\alpha \log \pi_\omega(a|x) - Q_\theta(x, a)]]. \tag{2}$$

2.2 Behavioral Metrics in MDPs

Behavioral metrics is one kind of metrics constructed from environment's information, typically measuring differences in rewards and transitions (Le Lan, Bellemare, and Castro 2021). We continue to use this concept here to collectively refer to a series of metrics describing the similarity of states or state-action pairs in RL from this perspective, including the bisimulation metric, the lax-bisimulation metric (Taylor, Precup, and Panagaden 2008) and so on. We focus on the following two metrics:

Definition 2.1. (Castro 2020; Castro et al. 2021) Given an MDP $\mathcal{M}$ and a policy π, the operator $F^\pi \colon \mathbb{R}^{\mathcal{X} \times \mathcal{X}} \to \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$, with $\mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ be the set of functions from $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$, is

$$F^\pi(D_s)(x, y) \coloneqq |r_x^\pi - r_y^\pi| + \gamma W_1(D_s)(P_x^\pi, P_y^\pi) \tag{3}$$

and the MICo update operator $F_M^\pi \colon \mathbb{R}^{\mathcal{X} \times \mathcal{X}} \to \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ is

$$F_M^\pi(D_s)(x, y) \coloneqq |r_x^\pi - r_y^\pi| + \gamma \mathbb{E}_{\substack{x' \sim P_x^\pi \\ y' \sim P_y^\pi}}[D_s(x', y')] \tag{4}$$

for all functions $D_s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with $r_x^\pi = \sum_{a \in \mathcal{A}} \pi(a|x) r_x^a$ and $P_x^\pi = \sum_{a \in \mathcal{A}} \pi(a|x) P_x^a$ for all $x \in \mathcal{X}$.

The F^π's fixed point is called the *π-bisimulation metric*. It quantifies similarity of states under the dynamics induced by π, including the policy-weighted immediate rewards difference and the Wasserstein-1 distance (Villani 2009), denoted as W_1, of policy-weighted transition distributions.

The F_M^π's fixed point is called the *MICo distance*. It encodes information of state similarity in a self-referential manner. Same as the π-bisimulation metric, the rewards and transitions are related to the policy. The MICo distance only utilizes independent coupling of transition distributions, while all the coupling of them are considered in the π-bisimulation metric due to the existence of the Wasserstein-1 distance. These two fixed points are both policy-dependent, and we use D^π to denote both of them. They can bound differences of value functions under arbitrary policies:

Proposition 2.2. *(Castro 2020; Castro et al. 2021) For any two states $x, y \in \mathcal{X}$ and any policy π, we have $|V^\pi(x) - V^\pi(y)| \leq D^\pi(x, y)$.*

3 Methodology

In this section, we first point out the problems of previous metrics, then introduce the conservative state-action discrepancy and propose a specific architecture to learn it.

3.1 Problems of Policy-Dependent Metrics

To explain the impact of policy-dependent metrics on the state space formally, based on previous work (Kemertas and Aumentado-Armstrong 2021), we derive an upper bound on the diameter of $\mathcal{X}$:

Lemma 3.1. *Policy-dependent metrics of Eqs. 3 and 4 have an upper bound determined by their policy π:* $\text{diam}(\mathcal{X}; D^\pi) = \sup_{x,y\in\mathcal{X}} D^\pi(x,y) \leq \frac{1}{1-\gamma}\sup_{x,y\in\mathcal{X}}|r^\pi_x - r^\pi_y|$.

All Proofs on new theoretical results are in the appendix[1].Consider the extreme case that for any (x, y), the equality $|r^\pi_x - r^\pi_y| = 0$ happens at some training time. For example, at the early training stage the policy could perform badly everywhere so the rewards collected by this policy could be zero everywhere. It results in a degenerate solution that $\text{diam}(\mathcal{X}; D^\pi) = 0$, i.e., the so-called representation collapsing problem.

Additionally, though Proposition 2.2 is always appreciated in the related literature, such property as well means that non-optimal policies' performance have chance to introduce bias to the representation learning when using these metrics to shape the embedding space. The bias is because states may be forced to be in the wrong location in the embedding space. Specifically, two states whose optimal values have a quite small difference, can be mapped far away due to a large difference between their values induced by a sub-optimal policy. Unfortunately, the policy learning itself is based on the learnt representation. If the representation is biased, then it will increase the difficulty of learning the optimal policy, and may reduce learning efficiency or even make the policy converge to local optima. In conclusion, sub-optimal policies arising from the learning process can affect the validity of such metrics.

3.2 Conservative State-Action Discrepancy

To overcome above limitations, we propose a novel behavioral metric.

Definition 3.2. Given an MDP $\mathcal{M}$, we define the *conservative state-action discrepancy* $D^*_{SA} : \mathcal{X}^2 \times \mathcal{A}^2 \to \mathbb{R}$ and the *conservative state discrepancy* $D^*_S : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ in a recursive way as follows:

$$D^*_{SA}([x,a],[y,b]) = |r^a_x - r^b_y| + \gamma \mathbb{E}_{\substack{x'\sim P^a_x \\ y'\sim P^b_y}}[D^*_S(x',y')], \quad (5)$$

$$D^*_S(x,y) = \max_{a\in\mathcal{A}} D^*_{SA}([x,a],[y,a]), \quad (6)$$

for every state pair $(x, y) \in \mathcal{X} \times \mathcal{X}$ and state-action pair $([x, a], [y, b]) \in \mathcal{X}^2 \times \mathcal{A}^2$.

For any state pair or state-action pair, the conservative discrepancy evaluates the maximum difference of the reward sequences between them under the same action sequence. In this case, only states that behaviorally similar enough could have a small distance, so we call it conservative. From another perspective, it introduces pessimism that we always consider the most dissimilar situation between two states. Previous works argue (Castro and Precup 2010; Castro 2020) this pessimism can bring bad impact, however, they only have evidence in the transfer learning setting, as for the online policy learning settings, there is no theoretical or empirical support on whether it will harm performance or not. Thus we insist on the usage of it because it can bring us the following important advantages explicitly.

It has two main differences compared with the previous metrics. One is that previous ones consider the state similarity under the entire policy distribution, while this one only considers under certain action and the other is that it extends behavioral similarity measure to the state-action joint space.

The first difference brings policy-independence. Based on this, the influence of the conservative state discrepancy on the representation space is determined by the structure of the underlying MDP instead of policy's performance.

Lemma 3.3. *The conservative state discrepancy defined in Eq. 6 leads to a fixed upper bound on the diameter of $\mathcal{X}$:* $\text{diam}(\mathcal{X}; D^*_S) \leq \frac{1}{1-\gamma}|r_{\max} - r_{\min}|$.

The second difference allows us to smooth any function defined over the state-action space (e.g., Q-value function).

Proposition 3.4. *For any pair $([x,a],[y,b]) \in \mathcal{X}^2 \times \mathcal{A}^2$, we have $|Q^*(x,a) - Q^*(y,b)| \leq D^*_{SA}([x,a],[y,b])$.*

We get the upper bound of the optimal Q-value function here, which means that using the conservative state-action discrepancy to help learn representation can directly make Q-network generalize better on state-action pairs with similar Q-values.

To compute D^*_{SA} in an iterative way, we define a functional operator F_C: $\mathbb{R}^{\mathcal{X}^2\times\mathcal{A}^2} \to \mathbb{R}^{\mathcal{X}^2\times\mathcal{A}^2}$ as:

$$F_C(D_{SA})([x,a],[y,b]) \coloneqq |r^a_x - r^b_y| + \gamma \mathbb{E}_{\substack{x'\sim P^a_x \\ y'\sim P^b_y}}[\max_{a'} D_{SA}([x',a'],[y',a'])]. \quad (7)$$

for all functions $D_{SA} : \mathcal{X}^2 \times \mathcal{A}^2 \to \mathbb{R}$. Next, we show that the operator F_C's fixed point is the unique solution of Eqs. 5 and 6, and so we can get D^*_{SA} by iteratively applying F_C.

Proposition 3.5. *The operator F_C on $\mathbb{R}^{\mathcal{X}^2\times\mathcal{A}^2}$ is a contraction mapping with respect to the L^∞ norm.*

As F_C is a contraction mapping and $\mathbb{R}^{\mathcal{X}^2\times\mathcal{A}^2}$ is complete under $\|\cdot\|_\infty$, by directly applying Banach's fixed-point theorem we can get the following result:

Corollary 3.6. *F_C has a unique fixed point, i.e., for any initial D_{SA} on $\mathbb{R}^{\mathcal{X}^2\times\mathcal{A}^2}$,* $\lim_{n\to\infty} \underbrace{F_C F_C \cdots F_C}_{n \text{ times}}(D_{SA}) = D^*_{SA}$.

3.3 Links to the MICo Distance

First, we look back on a property of the MICo distance:

Lemma 3.7. *(Castro et al. 2021) The MICo operator F^π_M is the Bellman evaluation operator for the auxiliary MDP $\tilde{\mathcal{M}}_M = \langle\tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{P}, \tilde{r}, \gamma\rangle$, where $\tilde{\mathcal{X}} = \mathcal{X}\times\mathcal{X}, \tilde{\mathcal{A}} = \mathcal{A}\times\mathcal{A}$, $\tilde{P}^{(a,b)}_{(x,y)} = P^a_x P^b_y$, and $\tilde{r}_{(x,y)} = |r^\pi_x - r^\pi_y|$, for any $x, y, x', y' \in \mathcal{X}$ and $a, b \in \mathcal{A}$.*

Similarly, we have the following lemma.

[1] https://www.lamda.nju.edu.cn/liaowj/AAAI23_supp.pdf

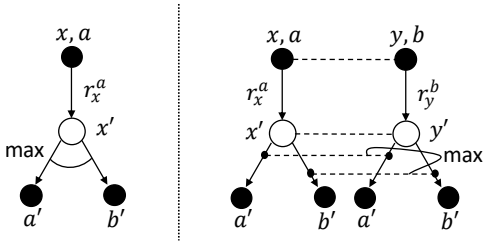

Figure 1: The backup diagrams for Q-learning (left) and Q^2-learning (right). The dashed line means that the connected node is coupled. Black nodes represent (state-)action values and white nodes represent state values.

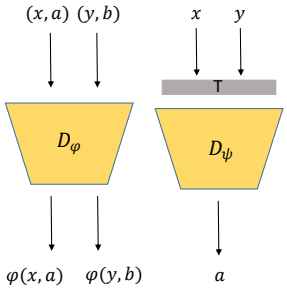

Figure 2: D_φ and D_ψ's architecture. "T" block means the concatenate operation that ensures inputs are invariant to permutations.

Lemma 3.8. *The operator F_C is the Bellman optimality operator for the auxiliary MDP $\tilde{\mathcal{M}}_C = \langle \tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{P}, \tilde{r}, \gamma \rangle$, where $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{P} \rangle$ are the same as the ones in $\tilde{\mathcal{M}}_M$, and $\tilde{r}_{(x,a),(y,b)} = |r_x^a - r_y^b|$.*

With Lemma 3.7, the temporal difference (TD) update rule can be applied to compute the MICo distance through sampling; correspondingly, we can apply Q-learning to compute D_{SA}^* and we name it *Q^2-learning* because the elements of each node in its backup diagram appear in pairs. See Fig. 1 for the difference in their backup diagrams (Sutton and Barto 2018).

In fact, the MICo distance can also have a "state-action pair" version when applied to deterministic policies. To see this, we first introduce:

$$D_{SA}^\pi([x,a],[y,b]) = |r_x^a - r_y^b| + \gamma \mathbb{E}_{\substack{x' \sim P_x^a \\ y' \sim P_y^b}} D_S^\pi(x', y'), \quad (8)$$

$$D_S^\pi(x,y) = \mathbb{E}_{\substack{\pi(a|x) \\ \pi(b|y)}} D_{SA}^\pi([x,a],[y,b]). \quad (9)$$

Here, $\pi(a|x)$ and $\pi(b|y)$ are stochastic policies. It is straightforward to show that these two equations can also define a Bellman evaluation operator for an auxiliary MDP. In fact, this MDP is exactly the same as $\tilde{\mathcal{M}}_C$. Further, if we replace $\mathbb{E}_{\pi(a|x),\pi(b|y)}$ with $\max_{a,b}$, then it immediately transforms into the optimality operator and also becomes a policy-independent behavioral metric, which is very similar to the conservative state-action discrepancy. However, we achieve a more refined state similarity measurement (e.g., tighter upper bound of value functions difference) because $\max_a D_{SA}^*([x,a],[y,a]) \leq \max_{a,b} D_{SA}^\pi([x,a],[y,b])$. When $\pi(a|x)$ and $\pi(b|y)$ are deterministic policies, Eqs. 8 and 9 can be seen as the state-action pair version of MICo.

The following property can guide us in scaling D_{SA}^* to the continuous action space.

Proposition 3.9. *D_{SA}^* and D_S^* can be regarded as the optimal action-value function and optimal state-value function of the auxiliary MDP $\tilde{\mathcal{M}}_C$, respectively.*

3.4 Scale to Continuous Action with Approximation

The "max" operator in D_{SA}^* hinders the application of Q^2-learning to continuous action spaces. By Proposition 3.9, we can use an actor-critic method to overcome the problem, like what DDPG (Lillicrap et al. 2016) or TD3 (Fujimoto, Hoof, and Meger 2018) does. We consider two parameterized functions D_φ and D_ψ with parameters φ and ψ, respectively, as the corresponding critic and actor serve for estimating D_{SA}^*. And we call them as D_{SA}^* critic and D_{SA}^* actor.

However, not as simple as the typical actor and critic functions, the D_{SA}^* critic and actor functions need to satisfy some additional important properties. These properties come from D_{SA}^* itself. We study the characteristics of D_{SA}^* in environments with the deterministic transitions.

Proposition 3.10. *For environments with deterministic transitions, D_{SA}^* is a pseudometric*[2].

Building on this and referring to (Dadashi et al. 2021), we use the Siamese network (Bromley et al. 1993) for the D_{SA}^* critic function since such architecture design preserves the pseudometric property. Concretely, we define the D_{SA}^* critic as $D_\varphi([x,a],[y,b]) = ||\varphi(x,a) - \varphi(y,b)||_1$, where $||\cdot||_1$ is the L_1 norm, and we use parameters φ to denote the network and its parameters simultaneously for simplicity.

As for the D_{SA}^* actor function, we require that $D_\psi(x,y) = D_\psi(y,x)$ for any $x, y \in \mathcal{X}$, i.e., the permutation invariance of input variables holds. The typical way that concatenates two input variables as a single one to be the input violates such invariance. Referring to (Chen, Cheng, and Mallat 2014), we fix it by inputting the model: concatenate$(x+y, |x-y|)$. Thus, its value is irrelevant to the input order and the input can be recovered: $\max(x,y) = (x+y+|x-y|)/2, \min(x,y) = (x+y-|x-y|)/2$, which means it is an injective mapping. The overall architecture is shown in Fig. 2.

And then, similar to Eq. 1, we can learn D_φ by minimizing the following mean-squared Bellman error:

$$\mathcal{L}_\varphi = \mathbb{E}_{\substack{(x,a,r_x^a,x') \sim \mathcal{B} \\ (y,b,r_y^b,y') \sim \mathcal{B}}} [(||\varphi(x,a) - \overline{\varphi}(y,b)||_1 - |r_x^a - r_y^b| - \gamma ||\hat{\varphi}(x', D_\psi(x',y')) - \hat{\varphi}(y', D_\psi(x',y'))||_1)^2]. \quad (10)$$

Here, $\hat{\varphi}$ denotes the target network and its parameters, and it acts like the target network in DQN (Mnih et al.

[2] A pseudometric d is a metric on a set X: $X \times X \to [0, \infty)$ satisfying for any $x, y, z \in X$: (1) $x = y \implies d(x,y) = 0$; (2) $d(x,y) = d(y,x)$; and (3) $d(x,y) \leq d(x,z) + d(z,y)$.

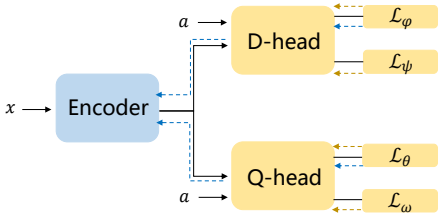

Figure 3: The connection of method's components. The dark solid lines represent the forward computation process. The dashed lines indicate the gradient backpropagation paths of the corresponding loss functions, respectively. The blue ones mean the gradients propagate to the encoder and the yellow ones mean they only propagate to the D-head or Q-head.

2015), which is used to provide a stable learning target. The symbol $\overline{\varphi}$ means that gradients will not pass through φ. We empirically find that using it can make training process more stable. According to the deterministic policy gradient theorem, we can learn D_ψ by maximizing $\mathbb{E}_{x,y}[D_\varphi([x, D_\psi(x, y)], [y, D_\psi(x, y)])]$. Specifically, we minimize this objective for D_ψ:

$$\mathcal{L}_\psi = \mathbb{E}_{\substack{x\sim\mathcal{B}\\ y\sim\mathcal{B}}}[-||\varphi(x, D_\psi(x, y)) - \overline{\varphi}(y, D_\psi(x, y))||_1]. \quad (11)$$

3.5 Integration with Policy Learning

We could apply Q^2-learning to assist any RL algorithm and we choose SAC here. The overall network structure is shown in Fig. 3. The encoder, usually composed of convolutional layers, is used to map the high-dimensional observations to low-dimensional vectors. Its parameters are updated by the gradients coming from both Q-head and D-head. The so-called "head" represents several fully-connected layers. Q-head is used to learn the policy, while D-head is used to learn D^*_{SA}. To make Q-value function more locally smooth, we only allow the gradients from the D^*_{SA} critic to backprop to the encoder. Empirically, this is a good choice.

4 Experiments

In this section, we want to demonstrate the scalability of Q^2-learning and study whether policy-independent metric can truly contribute to online policy learning by ignoring task-irrelevant information, especially faced with complex observations. Thus, we conduct experiments on the DeepMind Control Suite (Tassa et al. 2018), where the state and action spaces of the tasks are both continuous, and the observations are high-dimensional images. Two different background settings are taken into consideration. One is the default setting, called the clean background; and the other setting replaces the default background with natural videos that play a role of distractors to the control task, called the clutter background. Some examples are shown in Fig. 4. The latter setting is adapted from (Zhang et al. 2021), where the difference is that we randomly sample multiple videos from the kinetics dataset (Kay et al. 2017) as background rather than single one in (Zhang et al. 2021), which makes the background more complex and more unpredictable. This

Figure 4: The most left picture is one control task with the clean (default) background and the other three are with the clutter backgrounds, which are frames randomly selected from videos of "driving car" class (Kay et al. 2017).

change is motivated by the realistic situation that there is no static or single background in the real world. We use eight control tasks, from locomotion to swingup, from easy to hard, and from dense reward to sparse reward for relatively sufficient demonstration. They are Walker-Walk, Hopper-Stand, Cheetah-Run, Finger-Spin, Walker-Run, Hopper-Hop, Pendulum-Swingup, and Cartpole-Swinup_sparse, respectively.

We compare our method with MICo (Castro et al. 2021) and DBC (Zhang et al. 2021), based on the MICo distance and the π-bisimulation metric, respectively. Also, the baseline control algorithm, SAC is included. Additionally, we consider an important variant of DBC, DBC-normed (Kemertas and Aumentado-Armstrong 2021), which is an improvement version of DBC. It remediates the potential representation explosion problem of DBC by adding a norm constraint, and except that it uses the Huber loss to train the "Wasserstein part" of the π-bisimulation metric approximately. They further enhance DBC-normed with intrinsic reward and inverse dynamic model to address the sparse reward challenge. However, these two modules will not be involved in our experiments, because we they are orthogonal to discussing what kind of behavioral metrics will contribute to online policy learning. And we could inject such a module into every behavioral metric-based method by careful design. We leave it as a future work. Implementation details are presented in the appendix from the same link.

4.1 Performance Evaluation

We adopt two evaluation protocols from (Agarwal et al. 2021) to display different aspects of performance gains of Q^2-learning, including score distribution, the fraction of runs above a certain threshold, and interquartile mean (IQM), the mean of the middle half of running results. We use stratified bootstrap confidence interval (CI), bootstrap confidence interval (Efron 1992) with stratified sampling, for interval estimation. They are all task-level protocols and can reflect the overall performance across tasks and well-suited for few-run regime due to robustness to outliers and less statistical uncertainty. They are computed based on the normalized score, which is the sum of the undiscounted rewards collected by the agent in an episode divided by 1000 (the most it can collect). Each run provides a normalized score. The computation of them combines all runs across tasks and seeds, with a total of 10 (seeds) $\times$ 8 (tasks) runs in each background. We run 500K environment steps for each

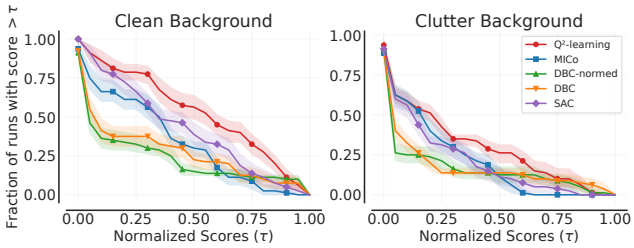

Figure 5: Score distributions using normalized score obtained on the 500K-th step. Shaded regions are 95% stratified bootstrap CIs. 10 seeds for each task are used.

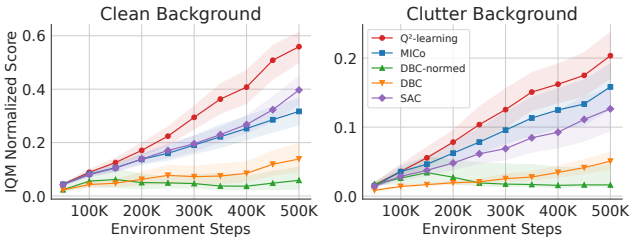

Figure 6: IQM normalized score evolves throughout the training process. The environment step is the actual execution time steps of the simulator (Laskin, Srinivas, and Abbeel 2020). IQM is evaluated every 50K steps. Shaded regions are 95% stratified bootstrap CIs. 10 seeds for each task are used.

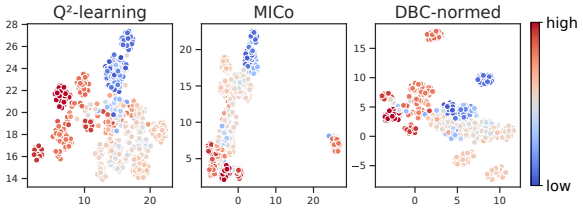

Figure 7: The visualization of representations. We use the UMAP method (McInnes, Healy, and Melville 2018) to project high-dimensional embedding vectors. The color of each point represents its state value's magnitude.

task. Per-task training curves are presented in the appendix from the same link.

In Fig. 5, a method is better in a certain score interval if its curve is above others in that interval. We can easily read methods' advantages in different score intervals. For example, in the clutter background, MICo is slightly better than SAC for $\tau \geq 0.4$ but for $\tau \leq 0.6$, MICo is worse because the fraction of runs above this score is almost zero. In two backgrounds, Q^2-learning's curve is above others in most score intervals. DBC and DBC-normed are highest in the intervals near 1.0 in both backgrounds due to their great performance on Finger-Spin (episodic returns are near 1000). But their curve is lowest in many other intervals, which means a dominant advantage across all tasks is not achieved.

Figure 6 conveys finer quantitative comparisons. The superiority of Q^2-learning is obvious, which has the best sample efficiency and asymptotic performance in both backgrounds, and especially in the clean one, there is a clear gap. While MICo brings no obvious improvement over SAC and it is even little worse than SAC in the clean one. MICo only has limited help to representation learning. As for DBC and DBC-normed, they are both majorly worse than SAC. It is not strange that it performs bad in the clean background because the original paper already shows this. The reason behind its failure in the clutter background is that we use multiple videos as backgrounds which is more complex than the original setting of single video. They need latent dynamic models that are inherently difficult to learn, and more complex backgrounds increase the difficulty of learning.

In some tasks policy-dependent methods can perform very well (e.g., DBC on Finger-Spin), but the overall performance is even worse than the vanilla RL algorithm because poor performance on more other tasks offsets good ones, while our method significantly and consistently improves the baseline. It implies the potential practical problem with policy-dependent metrics. They can result in large performance variability across tasks. We argue it is because that different policy learning and optimization processes on different tasks can affect metric learning. Our metric is more robust and stable at the task level.

4.2 Representation Visualization

To observe the structure of learned representations, we visualize the distribution of observation points of the Hopper-Stand task in the embedding space. We use UMAP (McInnes, Healy, and Melville 2018) to do the dimension reduction. The color associates with the state value, which is computed by another algorithm other than these three for fair comparison. Here, we use SAC that is trained long enough to return a near-optimal policy. And these three methods use the same source of trajectories to do this visualization. The trajectories are sampled from policies trained by these three methods respectively and then mixed up. In Fig. 7, all three methods present a phenomenon that points with same color form clusters, which means that observation points with similar values are mapped together in the embedding space. So it can be seen that Q^2-learning has the same ability as MICo and DBC-normed, i.e., it can build the structural and informative representation for RL to some extent.

According to (McInnes, Healy, and Melville 2018), this method can preserve the structure of the orginal space to some extent, so started from this view of point, compared to MICo and DBC-normed, Q^2-learning seems to achieve a more compact embedding space because it does not have a cluster that is significantly separated from the other points, but MICo and DBC-normed have. Such compactness may imply better smoothness of the value function in the representation space, which helps Q-value function generalize better over the representation space shaped by Q^2-learning (Le Lan, Bellemare, and Castro 2021). It can lead to better sample efficiency and asymptotic performance.

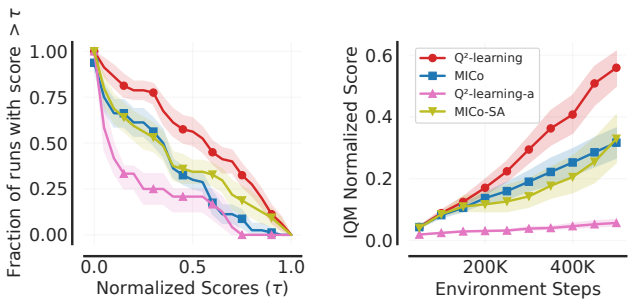

Figure 8: Performance comparison of variants of Q^2-learning in the clean background setting. Q^2-learning-a means the gradient coming from D^*_{SA} actor. Results shown here are computed using scores from all runs (8 tasks with 10 seeds).

4.3 Ablations

We demonstrate the value of each part of our representation learning method via two ablation experiments.

Where do the gradients come from? We use the gradient coming from the D^*_{SA} critic to update the encoder, which means we utilize a conservative state-action discrepancy to shape the embedding space. What if we replace this with a conservative state discrepancy, i.e., use the gradient from the D^*_{SA} actor? As Fig. 8 shows, the corresponding method Q^2-learning-a brings poor effect.

MICo-SA vs. MICo. As stated in Section 3.3, the MICo distance can also be extended to state-action spaces (Eqs. 8 and 9), and applied to representation learning. We call this method MICo-SA. It uses function approximators to represent the metric, like Q^2-learning. The learning procedure is similar to MICo, using the TD update rule. MICo-SA updates the encoder with gradients from that part of the network regarding state-action similarity.

The results are shown in Fig. 8. MICo-SA brings no substantial improvement over MICo, which suggests simply using more parameters to approximate the metric or considering the similarity of state-action pairs *cannot* bring about performance improvement.

Two key characteristics of Q^2-learning are the state-action similarity and policy-independence. From the first ablation, we can see that even with policy-independence, only considering the state similarity, the method fails. From the second ablation, we can see that without policy-independence, MICO-SA and MICo are not much different. So both are indispensable and Q^2-learning combines them effectively.

5 Related Work

Bisimulation equivalence (Givan, Dean, and Greig 2003) is one form of state abstraction (Li, Walsh, and Littman 2006). Its original definition is too strict to put into practice so the subsequent works (Ferns, Panangaden, and Precup 2004, 2011) turn to seek a real-valued metric that can reflect the bisimulation relationship, e.g. bisimulation metric (Ferns, Panangaden, and Precup 2004). The "max" operator also exists in its definition and it is also policy-independent. But it cannot deal with the troublesome of maximization when faced with continuous action spaces. Learning state similarity based on bisimulation online is mainly served for learning an informative representation for high dimensional inputs. Besides that, there are many other similar notions of state similarity proposed for policy transfer (Castro and Precup 2010), generalization (Agarwal et al. 2020), and goal-conditioned RL (Hansen-Estruch et al. 2022).

MDP homomorphism (Ravindran and Barto 2003) is another state abstraction method. The difference is that MDP homomorphism simultaneously considers the behavioral equivalence of state and action. Similarly, follow-up works (Ravindran and Barto 2004; Taylor, Precup, and Panagaden 2008) improve it by relaxing the condition for state-action aggregation to let it be more useful in practice. The corresponding behavioral metric is called the lax-bisimulation metric (Taylor, Precup, and Panagaden 2008). Despite this, it is still hard to be applied in large-scale tasks and in the case that policy is improved online. What our work proposes is also a state-action based behavioral metric, but the difference is that we can easily combine it with policy learning algorithms and scale it to continuous state and action spaces.

There are also lots of works (Ha and Schmidhuber 2018; Watter et al. 2015; Gelada et al. 2019) based on unsupervised learning to aid representation learning in RL. Their core idea is to learn low-dimensional latent states through reconstruction. The reconstruction objective can be the original observation, the next observation, or just the reward. Recently, more and more works (Laskin, Srinivas, and Abbeel 2020; Stooke et al. 2021; Schwarzer et al. 2020; Mazoure et al. 2020) are motivated by self-supervised learning (Henaff 2020; Chen et al. 2020) widely applied in computer vision. These works utilize the state similarity based on temporal vicinity or image augmentation to establish the contrastive loss function. Features learnt in this way can capture the temporally predictable elements contained in the original observation more effectively than unsupervised approaches, but these methods cannot filter out task-irrelevant information in theory.

6 Conclusion and Future Work

We introduce a novel behavioral metric, the conservative state-action discrepancy, which is policy-independent while remains scalability. Theoretical results show that our Q^2-learning is sound. On DeepMind control tasks, empirical results indicate that Q^2-learning consistently improves SAC's performance and is significantly more effective than other metric-based methods in the clean background setting and remains competitive in the clutter background setting. We break past prejudice that the inherent pessimism of policy-independent metrics (Castro and Precup 2010; Castro 2020) is not conducive to their application in RL. As a future topic, we hope to investigate more usages of this scalable policy-independent metric beyond representation learning.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2020AAA0107200), the National Science Foundation of China (61921006, 62076121, 62276126), and the Major Key Project of PCL (PCL2021A12). We thank Xudong Liu for providing computing resource support in the early stage of the experiments, and Chenyang Wu for the insightful discussion.

References

Agarwal, R.; Machado, M. C.; Castro, P. S.; and Bellemare, M. G. 2020. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A. C.; and Bellemare, M. 2021. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Bellman, R. 1957. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5): 679–684.

Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; and Shah, R. 1993. Signature verification using a "Siamese" time delay neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 737–744.

Castro, P. S. 2020. Scalable methods for computing state similarity in deterministic Markov decision processes. In *AAAI Conference on Artificial Intelligence (AAAI)*, 10069–10076.

Castro, P. S.; Kastner, T.; Panangaden, P.; and Rowland, M. 2021. MICo: Improved representations via sampling-based state similarity for Markov decision processes. *Advances in Neural Information Processing Systems (NeurIPS)*.

Castro, P. S.; and Precup, D. 2010. Using bisimulation for policy transfer in MDPs. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1065–1070.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 1597–1607.

Chen, X.; Cheng, X.; and Mallat, S. 2014. Unsupervised deep Haar scattering on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1709–1717.

Dadashi, R.; Rezaeifar, S.; Vieillard, N.; Hussenot, L.; Pietquin, O.; and Geist, M. 2021. Offline reinforcement learning with pseudometric learning. In *International Conference on Machine Learning (ICML)*, 2307–2318.

Efron, B. 1992. Bootstrap methods: Another look at the jackknife. In *Breakthroughs in statistics*, 569–593. Springer.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for finite Markov decision processes. In *Uncertainty in Artificial Intelligence (UAI)*, 162–169.

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

Ferns, N.; and Precup, D. 2014. Bisimulation metrics are optimal value functions. In *Uncertainty in Artificial Intelligence (UAI)*, 210–219.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 1587–1596.

Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; and Bellemare, M. G. 2019. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning (ICML)*, 2170–2179.

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

Ha, D.; and Schmidhuber, J. 2018. World models. *arXiv:1803.10122*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 1861–1870.

Hansen-Estruch, P.; Zhang, A.; Nair, A.; Yin, P.; and Levine, S. 2022. Bisimulation makes analogies in goal-conditioned reinforcement learning. In *International Conference on Machine Learning (ICML)*, 8407–8426.

Henaff, O. 2020. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning (ICML)*, 4182–4192.

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. *arXiv:1705.06950*.

Kemertas, M.; and Aumentado-Armstrong, T. 2021. Towards robust bisimulation metric learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. CURL: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 5639–5650.

Le Lan, C.; Bellemare, M. G.; and Castro, P. S. 2021. Metrics and continuity in reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 8261–8269.

Lesort, T.; Díaz-Rodríguez, N.; Goudou, J.-F.; and Filliat, D. 2018. State representation learning for control: An overview. *Neural Networks*, 108: 379–392.

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*, 531–539.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Mazoure, B.; Tachet des Combes, R.; DOAN, T. L.; Bachman, P.; and Hjelm, R. D. 2020. Deep reinforcement and infomax learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 3686–3698.

McInnes, L.; Healy, J.; and Melville, J. 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Ravindran, B.; and Barto, A. G. 2003. Relativized options: Choosing the right transformation. In *International Conference on Machine Learning (ICML)*, 608–615.

Ravindran, B.; and Barto, A. G. 2004. Approximate homomorphisms: A framework for non-exact minimization in Markov decision processes. In *International Conference on Knowledge Based Computer Systems (KBCS)*.

Schwarzer, M.; Anand, A.; Goel, R.; Hjelm, R. D.; Courville, A.; and Bachman, P. 2020. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations (ICLR)*.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; and Hassabis, D. 2018. A general reinforcement learning algorithm that masters Chess, Shogi, and Go through self-play. *Science*, 362(6419): 1140–1144.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International Conference on Machine Learning (ICML)*, 387–395.

Stooke, A.; Lee, K.; Abbeel, P.; and Laskin, M. 2021. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning (ICML)*, 9870–9879.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction (Second Edition)*. MIT press.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; Casas, D. d. L.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; Lillicrap, T.; and Riedmiller, M. 2018. DeepMind control suite. *arXiv:1801.00690*.

Taylor, J.; Precup, D.; and Panagaden, P. 2008. Bounding performance loss in approximate MDP homomorphisms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1649–1656.

Villani, C. 2009. *Optimal transport: Old and new*. Springer.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; Oh, J.; Horgan, D.; Kroiss, M.; Danihelka, I.; Huang, A.; Sifre, L.; Cai, T.; Agapiou, J. P.; Jaderberg, M.; Vezhnevets, A. S.; Leblond, R.; Pohlen, T.; Dalibard, V.; Budden, D.; Sulsky, Y.; Molloy, J.; Paine, T. L.; Gulcehre, C.; Wang, Z.; Pfaff, T.; Wu, Y.; Ring, R.; Yogatama, D.; Wünsch, D.; McKinney, K.; Smith, O.; Schaul, T.; Lillicrap, T.; Kavukcuoglu, K.; Hassabis, D.; Apps, C.; and Silver, D. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Watkins, C. 1989. Learning from delayed rewards. *PhD thesis, King's College, University of Cambridge*.

Watter, M.; Springenberg, J.; Boedecker, J.; and Riedmiller, M. 2015. Embed to Control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2746–2754.

Zhang, A.; McAllister, R. T.; Calandra, R.; Gal, Y.; and Levine, S. 2021. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*.