MICo: Improved representations via sampling-based state similarity for Markov decision processes

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present a new behavioural distance over the state space of a Markov decision process, and demonstrate the use of this distance as an effective means of shaping the learnt representations of deep reinforcement learning agents. While existing notions of state similarity are typically difficult to learn at scale due to high computational cost and lack of sample-based algorithms, our newly-proposed distance addresses both of these issues. In addition to providing detailed theoretical analysis, we provide empirical evidence that learning this distance alongside the value function yields structured and informative representations, including strong results on the Arcade Learning Environment benchmark.

1 Introduction

The success of reinforcement learning (RL) algorithms in large-scale, complex tasks depends on forming useful representations of the environment with which the algorithms interact. Feature selection and feature learning has long been an important subdomain of RL, and with the advent of deep reinforcement learning there has been much recent interest in understanding and improving the representations learnt by RL agents.

Much of the work in representation learning has taken place from the perspective of *auxiliary tasks* [Jaderberg et al., 2017, Bellemare et al., 2017, Fedus et al., 2019]; in addition to the primary reinforcement learning task, the agent may attempt to predict and control additional aspects of the environment. Auxiliary tasks shape the agent's representation of the environment *implicitly*, typically via gradient descent on the additional learning objectives. As such, while auxiliary tasks continue to play an important role in improving the performance of deep RL algorithms, our understanding of the effects of auxiliary tasks on representations in RL is still in its infancy.

In contrast to the implicit representation shaping of auxiliary tasks, a separate line of work on *behavioural metrics*, such as bisimulation metrics [Desharnais et al., 1999, 2004, Ferns et al., 2004, 2006], aims to capture structure in the environment by learning a metric measuring behavioral similarity between states. Recent works have successfully used behavioural metrics to shape the representations of deep RL agents [Gelada et al., 2019, Zhang et al., 2021, Agarwal et al., 2021]. However, in practice behavioural metrics are difficult to estimate from both statistical and computational perspectives, and these works either rely on specific assumptions about transition dynamics to make the estimation tractable, and as such can only be applied to limited classes of environments, or are applied to more general classes of environments not covered by theoretical guarantees.

The principal objective of this work is to develop new measures of behavioral similarity that avoid the statistical and computational difficulties described above, and simultaneously capture richer information about the environment. We introduce the *MICo (***M***atching under* **I***ndependent* **Co***uplings) distance*, and develop the theory around its computation and estimation, making comparisons with

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

existing metrics on the basis of computational and statistical efficiency. We demonstrate the usefulness of the representations that MICo yields, both through empirical evaluations in small problems (where we can compute them exactly) as well as in the Arcade Learning Environment [Bellemare et al., 2013, Machado et al., 2018], in which the performance of a wide variety of existing value-based deep RL agents is improved by directly shaping representations via the MICo distance (see Figure 4).

2 Background

Before describing the details of our contributions, we give a brief overview of the required background in reinforcement learning and bisimulation. We provide more extensive background in Appendix A.

Reinforcement learning. We consider a Markov decision process $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$ defined by a finite state space $\mathcal{X}$, finite action space $\mathcal{A}$, transition kernel $P : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathcal{X})$, reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. For notational convenience we will write P_x^a and r_x^a for transitions and rewards, respectively. Policies are mappings from states to distributions over actions: $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$ and induce a *value function* $V^\pi : \mathcal{X} \to \mathbb{R}$ defined via the recurrence: $V^\pi(x) := \mathbb{E}_{a\sim\pi(x)} \left[r_x^a + \gamma \mathbb{E}_{x'\sim P_x^a}[V^\pi(x')]\right]$. In RL we are concerned with finding the optimal policy $\pi^* = \arg\max_{\pi\in\mathscr{P}(\mathcal{A})^{\mathcal{X}}} V^\pi$ from interaction with sample trajectories with an MDP, *without knowledge of P or r* (and sometimes not even $\mathcal{X}$), and the optimal value function V^* induced by π^*.

State similarity and bisimulation metrics. Various notions of similarity between states in MDPs have been considered in the RL literature, with applications in policy transfer, state aggregation, and representation learning. The *bisimulation metric* [Ferns et al., 2004] is of particular relevance for this paper, and defines state similarity in an MDP by declaring two states $x, y \in \mathcal{X}$ to be close if their immediate rewards are similar, and the transition dynamics at each state leads to next states which are also judged to be similar. This self-referential notion is mathematically formalised by defining the bisimulation metric $d^\sim$ as the unique fixed-point of the operator $T_K : \mathcal{M}(X) \to \mathcal{M}(X)$, where $\mathcal{M}(X) = \{d \in [0, \infty)^{\mathcal{X}\times\mathcal{X}} : d \text{ symmetric and satisfies the triangle inequality}\}$ is the set of pseudometrics on $\mathcal{X}$, given by $T_K(d)(x, y) = \max_{a\in\mathcal{A}}[|r_x^a - r_y^a| + \gamma W_d(P_x^a, P_y^a)]$. Here, W_d is the Kantorovich distance (also known as the Wasserstein distance) over the set of distributions $\mathscr{P}(\mathcal{X})$ with base distance d, defined by $W_d(\mu, \nu) = \inf_{X\sim\mu,Y\sim\nu} \mathbb{E}[d(X, Y)]$, for all $\mu, \nu \in \mathscr{P}(\mathcal{X})$, where the infimum is taken over all couplings of (X, Y) with the prescribed marginals [Villani, 2008].

The mapping T_K is a γ-contraction on $\mathcal{M}(X)$ under the L^∞ norm [Ferns et al., 2011], and thus by standard contraction mapping arguments analogous to those used to study value iteration, it has a unique fixed point, the bisimulation metric $d^\sim$. Ferns et al. [2004] show that this metric bounds differences in the optimal value function, hence its importance in RL:

$$|V^*(x) - V^*(y)| \leq d^\sim(x, y) \quad \forall x, y \in \mathcal{X}\,. \tag{1}$$

Representation learning in RL. In large-scale environments, it is infeasible to express value functions directly as vectors in $\mathbb{R}^{\mathcal{X}\times\mathcal{A}}$. Instead, RL agents must approximate value functions in a more concise manner, by forming a *representation* of the environment, that is, a feature embedding $\phi : \mathcal{X} \to \mathbb{R}^M$, and predicting state-action values linearly from these features. *Representation learning* is the problem of finding a useful representation ϕ. Increasingly, deep RL agents are equipped with additional losses to aid representation learning. A common approach is to require the agent to make additional predictions (so-called *auxilliary tasks*) with its representation, typically with the aid of extra network parameters, with the intuition that an agent is more likely to learn useful features if it is required to solve many related tasks. We refer to such methods as *implicit* representation shaping, since improved representations are a side-effect of learning to solve auxiliary tasks.

Since bisimulation metrics capture additional information about the MDP in addition to that summarised in value functions, bisimulation metrics are a natural candidate for auxiliary tasks in deep reinforcement learning. Gelada et al. [2019], Agarwal et al. [2021], and Zhang et al. [2021] introduce auxiliary tasks based on bisimulation metrics, but require additional assumptions on the underlying MDP in order for the metric to be learnt correctly (Lipschitz continuity, deterministic, and Gaussian transitions, respectively). The success of these approaches provides motivation in this paper to introduce a notion of state similarity applicable to arbitrary MDPs, without further restriction. Further, we learn this state similarity *explicitly*: that is, without the aid of any additional network parameters.

3 Advantages and limitations of the bisimulation metric

The bisimulation metric $d^{\sim}$ is a strong notion of distance on the state space of an MDP; it is useful in policy transfer through its bound on optimal value functions [Castro and Precup, 2010] and because it is so stringent, it gives good guarantees for state aggregations [Ferns et al., 2004, Li et al., 2006]. However, it has been difficult to use at scale and compute online, for a variety of reasons that we summarize below.

(i) **Computational complexity.** The metric can be computed via fixed-point iteration since the operator T_K is a contraction mapping. The map T_K contracts at rate γ with respect to the L^∞ norm on $\mathcal{M}$, and therefore obtaining an ε-approximation of $d^{\sim}$ under this norm requires $O(\log(1/\varepsilon)/\log(1/\gamma))$ applications of T_K to an initial pseudometric d_0. The cost of each application of T_K is dominated by the computation of $|\mathcal{X}|^2|\mathcal{A}|$ W_d distances for distributions over $\mathcal{X}$, each costing $\tilde{O}(|\mathcal{X}|^{2.5})$ in theory [Lee and Sidford, 2014], and $\tilde{O}(|\mathcal{X}|^3)$ in practice [Pele and Werman, 2009, Guo et al., 2020a, Peyré and Cuturi, 2019]. Thus, the overall practical cost is $\tilde{O}(|\mathcal{X}|^5|\mathcal{A}|\log(\varepsilon)/\log(\gamma))$.

(ii) **Bias under sampled transitions.** Computing T_K requires access to the transition probability distributions P_x^a for each $(x, a) \in \mathcal{X} \times \mathcal{A}$ which, as mentioned in Section 2, are typically not available; instead, stochastic approximations to the operator of interest are employed. Whilst there has been work in studying online, sample-based approximate computation of the bisimulation metric [Ferns et al., 2006, Comanici et al., 2012], these methods are generally biased, in contrast to sample-based estimation of standard RL operators.

(iii) **Lack of connection to non-optimal policies.** One of the principal behavioural characterisations of the bisimulation metric $d^{\sim}$ is the upper bound shown in Equation (1). However, in general we do not have $|V^\pi(x) - V^\pi(y)| \leq d^{\sim}(x, y)$ for arbitrary policies $\pi \in \Pi$; a simple example is illustrated in Figure 1. More generally, notions of state similarity that the bisimulation metric encodes may not be closely related to behavioural similarity under an arbitrary policy π. Thus, learning about $d^{\sim}$ may not in itself be useful for large-scale reinforcement learning agents.

Property (i) expresses the intrinsic computational difficulty of computing this metric. Property (ii) illustrates the problems associated with attempting to move from operator-based computation to online, sampled-based computation of the metric (for example, when the environment dynamics are unknown). Finally, property (iii) shows that even if the metric is computable exactly, the information it yields about the MDP may not be practically useful. Although π-bisimulation (introduced by Castro [2020] and extended by Zhang et al. [2021]) addresses property (iii), their practical algorithms are limited to MDPs with deterministic transitions [Castro, 2020] or MDPs with Gaussian transition kernels [Zhang et al., 2021]. Taken together, these three properties motivate the search for a metric without these shortcomings, which can be used in combination with deep reinforcement learning.

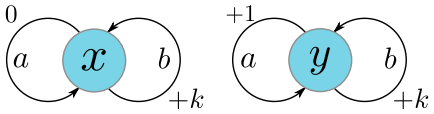

Figure 1: MDP illustrating that the upper bound for any π is not generally satisfied. Here, $d^{\sim}(x, y) = (1-\gamma)^{-1}$, but for $\pi(b|x) = 1, \pi(a|y) = 1$, we have $|V^\pi(x) - V^\pi(y)| = k(1-\gamma)^{-1}$.

4 The MICo distance

We now present a new notion of distance for state similarity, which we refer to as *MICo* (**M**atching under **I**ndependent **Co**uplings), designed to overcome the drawbacks described above.

Motivated by the drawbacks described in Section 3, we make several modifications to the operator T_K introduced above: (i) in order to deal with the prohibitive cost of computing the Kantorovich distance, which optimizes over all coupling of the distributions P_x^a and P_y^a, we use the independent coupling; (ii) to deal with lack of connection to non-optimal policies, we consider an on-policy variant of the metric, pertaining to a chosen policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$. This leads us to the following definition.

Definition 4.1. Given $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, the *MICo update operator* $T_M^\pi : \mathbb{R}^{\mathcal{X}\times\mathcal{X}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ is:

$$(T_M^\pi U)(x, y) = |r_x^\pi - r_y^\pi| + \gamma \mathbb{E}_{\substack{x' \sim P_x^\pi \\ y' \sim P_y^\pi}} [U(x', y')] \tag{2}$$

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

As with the bisimulation operator, this can be thought of as encoding desired properties of a notion of similarity between states in a self-referential manner; the similarity of two states $x, y \in \mathcal{X}$ should be determined by the similarity of the rewards and the similarity of the states they lead to.

Proposition 4.2. The operator T_M^π is a contraction mapping on $\mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ with respect to the L^∞ norm.

Proof. See Appendix B. □

The following corollary now follows immediately from Banach's fixed-point theorem and the completeness of $\mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ under the L^∞ norm.

Corollary 4.3. The MICo operator T_M^π has a unique fixed point $U^\pi \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$, and repeated application of T_M^π to any initial function $U \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ converges to U^π.

Having defined a new operator, and shown that it has a corresponding fixed-point, there are two questions to address: Does this new notion of distance overcome the drawbacks of the bisimulation metric described above; and what does this new object tell us about the underlying MDP?

4.1 Addressing the drawbacks of the bisimulation metric

We introduced the MICo distance as a means of overcoming some of the shortcomings associated with the bisimulation metric, described in Section 3. In this section, we provide a series of results that show that the newly-defined notion of distance addressess each of these shortcomings. The proofs of these results rely on the following lemma, connecting the MICo operator to a lifted MDP. This result is crucial for much of the analysis that follows, so we describe the proof in full detail.

Lemma 4.4 (Lifted MDP). The MICo operator T_M^π is the Bellman evaluation operator for an auxiliary MDP.

Proof. Given the MDP specified by the tuple $(\mathcal{X}, \mathcal{A}, P, R)$, we construct an auxiliary MDP $(\widetilde{\mathcal{X}}, \widetilde{\mathcal{A}}, \widetilde{P}, \widetilde{R})$, by taking the state space to be $\widetilde{\mathcal{X}} = \mathcal{X}^2$, the action space to be $\widetilde{\mathcal{A}} = \mathcal{A}^2$, the transition dynamics to be given by $\widetilde{P}^{(a,b)}_{(u,v)}((x,y)) = P^a_u(x)P^b_v(y)$ for all $(x,y),(u,v) \in \mathcal{X}^2$, $a, b \in \mathcal{A}$, and the action-independent rewards to be $\widetilde{R}_{(x,y)} = |r^\pi_x - r^\pi_y|$ for all $x, y \in \mathcal{X}$. The Bellman evaluation operator $\widetilde{T}^{\tilde{\pi}}$ for this auxiliary MDP at discount rate γ under the policy $\tilde{\pi}(a,b|x,y) = \pi(a|x)\pi(b|y)$ is given by (for all $U \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ and $(x,y) \in \mathcal{X}\times\mathcal{X}$):

$$
\begin{aligned}
(\widetilde{T}^{\tilde{\pi}}U)(x,y) &= \widetilde{R}_{(x,y)} + \gamma \sum_{(x',y')\in\mathcal{X}^2} \widetilde{P}^{(a,b)}_{(x,y)}((x',y'))\tilde{\pi}(a,b|x,y)U(x',y') \\
&= |r^\pi_x - r^\pi_y| + \gamma \sum_{(x',y')\in\mathcal{X}^2} P^\pi_x(x')P^\pi_y(y')U(x',y') = (T^\pi_M U)(x,y)\,. \qquad \square
\end{aligned}
$$

Remark 4.5. Ferns and Precup [2014] noted that the bisimulation metric can be interpreted as the optimal value function in a related MDP, and that the functional T_K of T_K can be interpreted as a Bellman optimality operator. However, their proof was non-constructive, the related MDP being characterised via the solution of an optimal transport problem. In contrast, the connection described above is constructive, and will be useful in understanding many of the theoretical properties of MICo. Ferns and Precup [2014] also note that the W_d distance in the definition of T_K can be upper-bounded by taking a restricted class of couplings of the transition distributions. The MICo metric can be viewed as restricting the coupling class precisely to the singleton containing the independent coupling.

With Lemma 4.4 established, we can now address each of the points (i), (ii), and (iii) from Section 3.

(i) Computational complexity. The key result regarding the computational complexity of computing the MICo distance is as follows.

Proposition 4.6 (MICo computational complexity). The computational complexity of computing an ε-approximation in L^∞ to the MICo metric is $O(|\mathcal{X}|^4 \log(\varepsilon)/\log(\gamma))$.

Proof. Since, by Proposition 4.2, the operator T_M^π is a γ-contraction under L^∞, we require $\mathcal{O}(\log(1/\varepsilon)/\log(1/\gamma))$ applications of the operator to obtain an ε-approximation in L^∞. Each iteration of value iteration updates $|\mathcal{X}|^2$ table entries, and the cost of each update is $\mathcal{O}(|\mathcal{X}|^2)$, leading to an overall cost of $O(|\mathcal{X}|^4 \log(\varepsilon)/\log(\gamma))$. □

In contrast to the bisimulation metric, this represents a computational saving of $O(|\mathcal{X}|)$, which arises from the lack of a need to solve optimal transport problems over the state space in computing the MICo distance. There is a further saving of $\mathcal{O}(|\mathcal{A}|)$ that arises since MICo focuses on an individual policy π, and so does not require the max over actions in the bisimulation operator definition.

(ii) Online approximation. Due to the interpretation of the MICo operator T_M^π as the Bellman evaluation operator in an auxiliary MDP, established in Lemma 4.4, algorithms and associated proofs of correctness for computing the MICo distance online can be straightforwardly derived from standard online algorithms for policy evaluation. We describe a straightforward approach, based on the TD(0) algorithm, and also note that the wide range of online policy evaluation methods incorporating off-policy corrections and multi-step returns, as well as techniques for applying such methods at scale, may also be used.

Given a current estimate U_t of the fixed point of T_M^π and a pair of observations (x, a, r, x'), $(y, b, \tilde{r}, y')$ generated under π, we can define a new estimate U_{t+1} via

$$U_{t+1}(x, y) \leftarrow (1 - \epsilon_t(x, y))U_t(x, y) + \epsilon_t(x, y)(|r - \tilde{r}| + \gamma U_t(x', y')) \tag{3}$$

and $U_{t+1}(\tilde{x}, \tilde{y}) = U_t(\tilde{x}, \tilde{y})$ for all other state-pairs $(\tilde{x}, \tilde{y}) \neq (x, y)$, for some sequence of stepsizes $\{\epsilon_t(x, y) \mid t \geq 0, (x, y) \in \mathcal{X}^2\}$. Sufficient conditions for convergence of this algorithm can be deduced straightforwardly from corresponding conditions for TD(0). We state one such result below. An important caveat is that the correctness of this particular algorithm depends on rewards depending only on state; one can switch to state-action metrics if this hypothesis is not satisfied.

Proposition 4.7. Suppose rewards depend only on state, and consider the sequence of estimates $(U_t)_{t\geq 0}$, with U_0 initialised arbitrarily, and U_{t+1} updated from U_t via a pair of transitions (x_t, a_t, r_t, x_t'), $(y_t, b_t, \tilde{r}_t, y_t')$ as in Equation (3). If all state-pairs tuples are updated infinitely often, and stepsizes for these updates satisfy the Robbins-Monro conditions. Then $U_t \to U^\pi$ almost surely.

Proof. Under the assumptions of the proposition, the update described is exactly a TD(0) update in the lifted MDP described in Lemma 4.4. We can therefore appeal to Proposition 4.5 of Bertsekas and Tsitsiklis [1996] to obtain the result. □

Thus, in contrast to the Kantorovich metric, convergence to the exact MICo metric is possible with an online algorithm that uses sampled transitions.

(iii) Relationship to underlying policy. In contrast to the bisimulation metric, we have the following on-policy guarantee for the MICo metric.

Proposition 4.8. For any $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$ and states $x, y \in \mathcal{X}$, we have $|V^\pi(x) - V^\pi(y)| \leq U^\pi(x, y)$.

Proof. We apply a coinductive argument [Kozen, 2007] to show that if $|V^\pi(x) - V^\pi(y)| \leq U(x, y)$ for all $x, y \in \mathcal{X}$, for some $U \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ symmetric in its two arguments, then we also have $|V^\pi(x) - V^\pi(y)| \leq (T_M^\pi U)(x, y)$ for all $x, y \in \mathcal{X}$. Since the hypothesis holds for the constant function $U(x, y) = 2\max_{z,a} |r(z, a)|/(1 - \gamma)$, and T_M^π contracts around U^π, the conclusion then follows. Therefore, suppose the hypothesis holds. Then we have

$$\begin{aligned}
V^\pi(x) - V^\pi(y) &= r_x^\pi - r_y^\pi + \gamma \sum_{x'\in\mathcal{X}} P_x^\pi(x')V(x') - \gamma \sum_{y'\in\mathcal{X}} P_y^\pi(y')V(y') \\
&\leq |r_x^\pi - r_y^\pi| + \gamma \sum_{x',y'\in\mathcal{X}} P_x^\pi(x')P_y^\pi(y')(V^\pi(x') - V^\pi(y')) \\
&\leq |r_x^\pi - r_y^\pi| + \gamma \sum_{x',y'\in\mathcal{X}} P_x^\pi(x')P_y^\pi(y')U(x', y') = (T_M^\pi U)(x, y)\,.
\end{aligned}$$

By symmetry, $V^\pi(y) - V^\pi(x) \leq (T_M^\pi U)(x, y)$, as required. □

4.2 Diffuse metrics

To characterize the nature of the fixed point U^π, we introduce the notion of a *diffuse metric*.

Definition 4.9. Given a set $\mathcal{X}$, a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *diffuse metric* if the following axioms hold: (i) $d(x, y) \geq 0$ for any $x, y \in \mathcal{X}$; (ii) $d(x, y) = d(y, x)$ for any $x, y \in \mathcal{X}$; (iii) $d(x, y) \leq d(x, z) + d(y, z)\ \forall x, y, z \in \mathcal{X}$.

These differ from the standard metric axioms in the first point: we no longer require that a point has zero self-distance, and two distinct points may have zero distance. Notions of this kind are increasingly common in machine learning as researchers develop more computationally tractable versions of distances, as with entropy-regularised optimal transport distances [Cuturi, 2013], which also do not satisfy the axiom of zero self-distance.

An example of a diffuse metric is the Łukaszyk–Karmowski distance [Łukaszyk, 2004], which is used in the MICo metric as the operator between the next-state distributions. Given a diffuse metric space $(\mathcal{X}, \rho)$, the Łukaszyk–Karmowski distance d^{ρ}_{LK} is a diffuse metric on probability measures on $\mathcal{X}$ given by $d^{\rho}_{\text{LK}}(\mu, \nu) = \mathbb{E}_{x\sim\mu, y\sim\nu}[\rho(x, y)]$. This example demonstrates the origin of the name *diffuse* metrics: the non-zero self distances arises from a point being spread across a probability distribution. In terms of the Łukaszyk–Karmowski distance, the MICo distance can be written as the fixed point $U^{\pi}(x, y) = |r^{\pi}_x - r^{\pi}_y| + d_{\text{LK}}(U^{\pi})(P^{\pi}_x, P^{\pi}_y)$. This characterisation leads to the following result.

Proposition 4.10. The MICo distance is a diffuse metric.

Proof. Non-negativity and symmetry of U^{π} are clear, so it remains to check the triangle inequality. To do this, we define a sequence of iterates $(U_k)_{k\geq 0}$ in $\mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ by $U_0(x, y) = 0$ for all $x, y \in \mathcal{X}$, and $U_{k+1} = T^{\pi}_M U_k$ for each $k \geq 0$. Recall that by Corollary 4.3 that $U_k \to U^{\pi}$. We will show that each U_k satisfies the triangle inequality by induction. By taking limits on either side of the inequality, we will then recover that U^{π} itself satisfies the triangle inequality. The base case of the inductive argument is clear from the choice of U_0. For the inductive step, assume that for some $k \geq 0$, $U_k(x, y) \leq U_k(x, z) + U_k(z, y)$ for all $x, y, z \in \mathcal{X}$. Now for any $x, y, z \in \mathcal{X}$, we have

$$\begin{aligned} U_{k+1}(x, y) &= |r^{\pi}_x - r^{\pi}_y| + \gamma\mathbb{E}_{X'\sim P^{\pi}_x, Y'\sim P^{\pi}_y}[U_k(X', Y')] \\ &\leq |r^{\pi}_x - r^{\pi}_z| + |r^{\pi}_z - r^{\pi}_y| + \gamma\mathbb{E}_{X'\sim P^{\pi}_x, Y'\sim P^{\pi}_y, Z'\sim P^{\pi}_z}[U_k(X', Z') + U_k(Z', Y')] \\ &= U_{k+1}(x, z) + U_{k+1}(z, y)\,. \end{aligned}$$

□

It is interesting to note that a state $x \in \mathcal{X}$ has zero self-distance iff the Markov chain induced by π initialised at x is deterministic, and the magnitude of a state's self-distance is indicative of the amount of "dispersion" in the distribution. Hence, in general, we have $U^{\pi}(x, x) > 0$, and $U^{\pi}(x, x) \neq U^{\pi}(y, y)$ for distinct states $x, y \in \mathcal{X}$. See the appendix for further discussion of diffuse metrics and related constructions.

5 The MICo loss

The impetus of our work is the development of principled mechanisms for directly shaping the representations used by RL agents so as to improve their learning. In this section we present a novel loss based on the MICo update operator T^{π}_M given in Equation (2) that can be incorporated into any value-based agent. Given the fact that MICo is a diffuse metric that can admit non-zero self-distances, special care needs to be taken in how these distances are learnt; indeed, traditional mechanisms for measuring distances between representations (e.g. Euclidean and cosine distances) are geometrically-based and enforce zero self-distances.

We assume a value-based agent learning an estimate $Q_{\xi,\omega}$ defined by the composition of two function approximators ψ and ϕ with parameters ξ and ω, respectively: $Q_{\xi,\omega}(x, \cdot) = \psi_{\xi}(\phi_{\omega}(x))$. We will refer to $\phi_{\omega}(x)$ as the *representation* of state x and aim to make distances between representations match the MICo distance; we refer to ψ_{ξ} as the *value approximator*. We define the parameterized representation distance, U_{ω}, as an approximant to U^{π}:

$$U^{\pi}(x, y) \approx U_{\omega}(x, y) := \frac{\|\phi_{\omega}(x)\|_2 + \|\phi_{\omega}(y)\|_2}{2} + \beta\theta(\phi_{\omega}(x), \phi_{\omega}(y))$$

where $\theta(\phi_{\omega}(x), \phi_{\omega}(y))$ is the angle between vectors $\phi_{\omega}(x)$ and $\phi_{\omega}(y)$ and β is a scalar (in our results we use $\beta = 0.1$ but present results with other values of β in the appendix).

Based on Equation (2), our learning target is then $T^{U}_{\bar{\omega}}(r_x, x', r_y, y') = |r_x - r_y| + \gamma U_{\bar{\omega}}(x', y')$, where $\bar{\omega}$ is a separate copy of the network parameters that are synchronised with ω at infrequent intervals. This is a common practice that was introduced by Mnih et al. [2015] (and in fact, we use the same update schedule they propose). The loss for this learning target is

$$\mathcal{L}_{\text{MICo}}(\omega) = \mathbb{E}_{\langle x, r_x, x'\rangle, \langle y, r_y, y'\rangle}\left[\left(T^{U}_{\bar{\omega}}(r_x, x', r_y, y') - U_{\omega}(x, y)\right)^2\right]$$

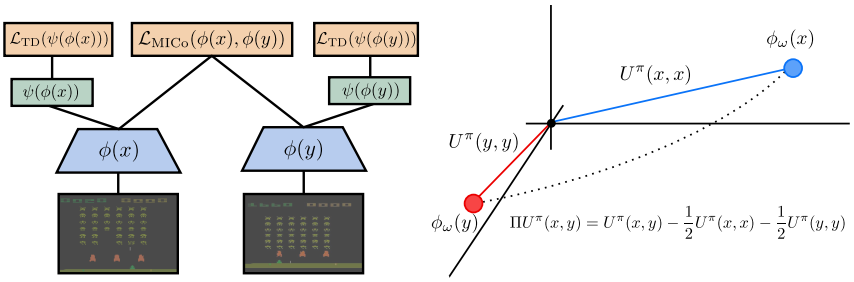

Figure 2: **Left:** Illustration of network architecture for learning MICo; **Right:** The projection of MICo distances onto representation space.

where $\langle x, r_x, x' \rangle$ and $\langle y, r_y, y' \rangle$ are pairs of transitions sampled from the agent's replay buffer. We can combine $\mathcal{L}_{\text{MICo}}$ with the temporal-difference loss $\mathcal{L}_{\text{TD}}$ of any value-based agent as $(1-\alpha)\mathcal{L}_{\text{TD}} + \alpha\mathcal{L}_{\text{MICo}}$, where $\alpha \in (0, 1)$. Each sampled mini-batch is used for both MICo and TD losses. Figure 2 (left) illustrates the network architecture used for learning.

Although the loss $\mathcal{L}_{\text{MICo}}$ is designed to learn the MICo diffuse metric U^π, the values of the metric itself are parametrised through U_ω defined above, which is constituted by several distinct terms. This appears to leave a question as to how the representations $\phi_\omega(x)$ and $\phi_\omega(y)$, as Euclidean vectors, are related to one another when the MICo loss is minimised. Careful inspection of the form of $U_\omega(x, y)$ shows that the (scaled) angular distance between $\phi_\omega(x)$ and $\phi_\omega(y)$ can be recovered from U_ω by subtracting the learnt approximations to the self-distances $U^\pi(x, x)$ and $U^\pi(y, y)$ (see Figure 2, right). We therefore define the reduced MICo distance ΠU^π, which encodes the distances enforced between the representation vectors $\phi_\omega(x)$ and $\phi_\omega(y)$, by:

$$\beta\theta(\phi_\omega(x), \phi_\omega(y)) \approx \Pi U^\pi(x, y) = U^\pi(x, y) - \frac{1}{2}U^\pi(x, x) - \frac{1}{2}U^\pi(y, y)\,.$$

In the following section we investigate the following two questions: **(1)** How informative of V^π is ΠU^π?; and **(2)** How useful are the features encountered by ΠU^π for policy evaluation? We conduct these investigations on tabular environments where we can compute the metrics exactly, which helps clarify the behaviour of our loss when combined with deep networks in Section 6.

5.1 Value bound gaps

Although Proposition 4.8 states that we have $|V^\pi(x) - V^\pi(y)| \leq U^\pi(x, y)$, we do not, in general, have the same upper bound for $\Pi U^\pi(x, y)$ as demonstrated by the following result.

Lemma 5.1. There exists an MDP with $x, y \in \mathcal{X}$, and $\pi \in \Pi$ where $|V^\pi(x) - V^\pi(y)| > \Pi U^\pi(x, y)$.

Proof. Consider a single-action MDP with two states (x and y) where y is absorbing, x transitions with equal probability to x and y, and a reward of 1 is received only upon taking an action from state x. There is only one policy for this MDP which yields the value function $V(x) \approx 1.8$ and $V(y) = 0$. The MICo distance gives $U(x, x) \approx 1.06$, $U(x, y) \approx 1.82$, and $U(y, y) = 0$, while the reduced MICo distance yields $\Pi U(x, x) = \Pi U(y, y) = 0$, and $\Pi U(x, y) \approx 1.29 < |V(x) - V(y)| = 1.8$. □

Despite this negative result, it is worth evaluating how often *in practice* this inequality is violated and by how much, as this directly impacts the utility of this distance for learning representations.

To do so, we make use of *Garnet MDPs*, a class of randomly generated MDPs [Archibald et al., 1995, Piot et al., 2014]. Given a specified number of states $n_{\mathcal{X}}$ and the number of actions $n_{\mathcal{A}}$, Garnet$(n_{\mathcal{X}}, n_{\mathcal{A}})$ is generated as follows: **1.** The branching factor $b_{x,a}$ of each transition P_x^a is sampled uniformly from $[1 : n_{\mathcal{X}}]$. **2.** $b_{x,a}$ states are picked uniformly randomly from $\mathcal{X}$ and assigned a random value in $[0, 1]$; these values are then normalized to produce a proper distribution P_x^a. **3.** Each r_x^a is sampled uniformly in $[0, 1]$.

For each Garnet$(n_{\mathcal{X}}, n_{\mathcal{A}})$ we sample 100 stochastic policies $\{\pi_i\}$ and compute the average gap: $\frac{1}{100|\mathcal{X}|^2}\sum_i \sum_{x,y} d(x, y) - |V^{\pi_i}(x) - V^{\pi_i}(y)|$, where d stands for any of the considered metrics. Note we are measuring the *signed* difference, as we are interested in the frequency with which the upper bound is violated. As seen in Figure 3 (left), our metric *does* on average provide an upper bound

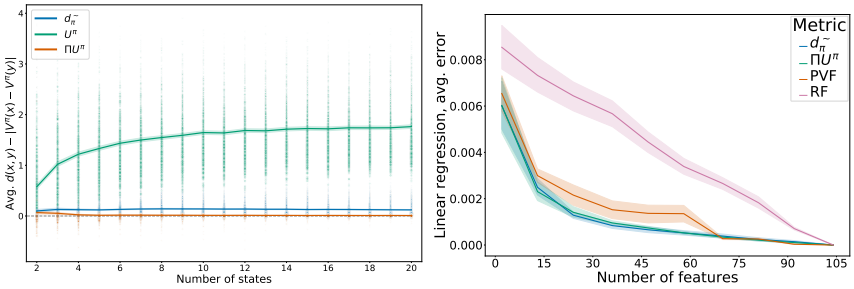

Figure 3: **Left:** The gap between the difference in values and the various distances for Garnet MDPs with varying numbers of actions (represented by circle sizes); **Right:** Average error when performing linear regression on varying numbers of features in the four-rooms GridWorld, averaged over 10 runs; shaded areas represent 95% confidence intervals.

on the difference in values that is also tighter bound than those provided by U^π and π-bisimulation. This suggests that the resulting representations remain informative of value similarities.

5.2 State features

In order to investigate the usefulness of the representations produced by ΠU^π, we construct state features directly by using the computed distances to project the states into a lower-dimensional space with the UMAP dimensionality reduction algorithm [McInnes et al., 2018][1]. We then apply linear regression of the true value function V^π against the features to compute $\hat{V}^\pi$ and measure the average error across the state space. As baselines we compare against random features (RF), Proto Value Functions (PVF) [Mahadevan and Maggioni, 2007], and the features produced by π-bisimulation [Castro, 2020]. We present our results on the well-known four-rooms GridWorld [Sutton et al., 1999] in Figure 3 (right) and provide results on more environments in the appendix. Despite the independent couplings, ΠU^π performs on par with π-bisimulation, which optimizes over all couplings.

6 Large-scale empirical evaluation

Having developed a greater understanding of the properties inherent to the representations produced by the MICo loss, we evaluate it on the Arcade Learning Environment [Bellemare et al., 2013]. We added the MICo loss to all the JAX agents provided in the Dopamine library [Castro et al., 2018]: DQN [Mnih et al., 2015], Rainbow [Hessel et al., 2018], QR-DQN [Dabney et al., 2018b], and IQN [Dabney et al., 2018a], using mean squared error loss to minimize $\mathcal{L}_{\text{TD}}$ for DQN (as suggested by Obando-Ceron and Castro [2021]). Given the state-of-the-art results demonstrated by the Munchausen-IQN (M-IQN) agent [Vieillard et al., 2020], we also evaluated incorporating our loss into M-IQN.[2] For all experiments we used the hyperparameter settings provided with Dopamine. We found that a value of $\alpha = 0.5$ worked well with quantile-based agents (QR-DQN, IQN, and M-IQN), while a value of $\alpha = 0.01$ worked well with DQN and Rainbow. We hypothesise that the difference in scale of the quantile, categorical, and non-distributional loss functions concerned leads to these distinct values of α performing well. We found it important to use the Huber loss [Huber, 1964] to minimize $\mathcal{L}_{\text{MICo}}$ as this emphasizes greater accuracy for smaller distances as oppoosed to larger distances. We experimented using the MSE loss but found that larger distances tended to overwhelm the optimization process, thereby degrading performance.

We evaluated on all 60 Atari 2600 games over 5 seeds and report the results in Figure 4; the fact that the MICo loss provides consistent improvements over a wide range of baseline agents of varying complexity suggests that the MICo loss can help learn better representations for control. Additional details are provided in the appendix.

[1] Note that since UMAP expects a metric, it is ill-defined with the diffuse metric U^π.

[2] Given that the authors of M-IQN had implemented their agent in TensorFlow (whereas our agents are in JAX), we have reimplemented M-IQN in JAX and run 5 independent runs (in contrast to the 3 run by Vieillard et al. [2020].

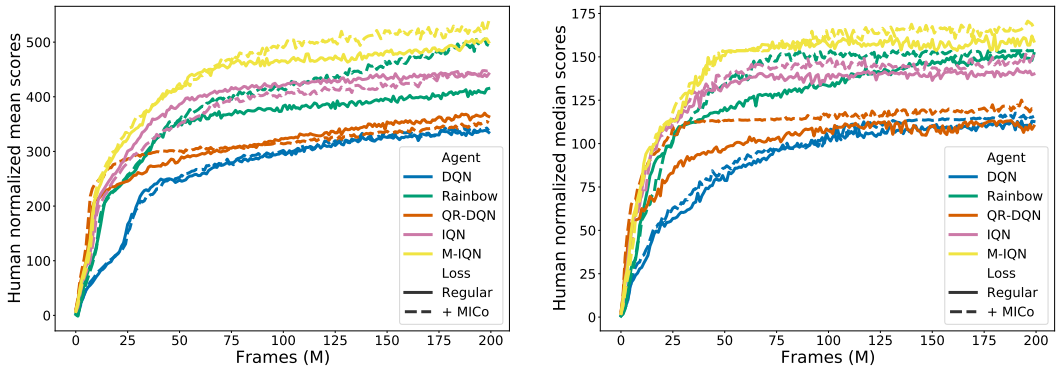

Figure 4: Mean (left) and median (right) human normalized scores across 60 Atari 2600 games, averaged over 5 independent runs.

7 Related Work

Bisimulation metrics were introduced for MDPs by Ferns et al. [2004], and have been extended in a number of directions [Ferns et al., 2005, 2006, Taylor, 2008, Taylor et al., 2009, Ferns et al., 2011, Comanici et al., 2012, Bacci et al., 2013a,b, Abate, 2013, Ferns and Precup, 2014, Castro, 2020], with applications including policy transfer [Castro and Precup, 2010, Santara et al., 2019], representation learning [Ruan et al., 2015, Comanici et al., 2015], and state aggregation [Li et al., 2006].

A range of other notions of state similarity in MDPs have also been considered, such as action sequence equivalence [Givan et al., 2003], temporally extended metrics [Amortila et al., 2019], MDP homomorphisms [Ravindran and Barto, 2003], utile distinction [McCallum, 1996], and policy irrelevance [Jong and Stone, 2005]. Li et al. [2006] review different notions of similarity applied to state aggregation. Recently, Le Lan et al. [2021] performed an exhaustive analysis of the continuity properties, relative to functions of interest in RL, of a number of existing metrics in the literature.

The notion of zero self-distance, central to the diffuse metrics defined in this paper, is increasingly encountered in machine learning applications involving approximation of losses. Of particular note is entropy-regularised optimal transport [Cuturi, 2013] and related quantities [Genevay et al., 2018, Fatras et al., 2020, Chizat et al., 2020, Fatras et al., 2021].

More broadly, many approaches to representation learning in deep RL have been considered, such as those based on auxiliary tasks (see e.g. [Sutton et al., 2011, Jaderberg et al., 2017, Bellemare et al., 2017, François-Lavet et al., 2019, Gelada et al., 2019, Guo et al., 2020b]), and other approaches such as successor features [Dayan, 1993, Barreto et al., 2017].

8 Conclusion

In this paper, we have introduced the MICo distance, a notion of state similarity that can be learnt at scale and from samples. We have studied the theoretical properties of MICo, and proposed a new loss to make the non-zero self-distances of this diffuse metric compatible with function approximation, combining it with a variety of deep RL agents to obtain strong performance on the Arcade Learning Environment. In contrast to auxiliary losses that *implicitly* shape an agent's representation, MICo directly modifies the features learnt by a deep RL agent; our results indicate that this helps improve performance. To the best of our knowledge, this is the first time *directly* shaping the representation of RL agents has been successfully applied at scale. We believe this represents an interesting new approach to representation learning in RL; continuing to develop theory, algorithms and implementations for direct representation shaping in deep RL is an important and promising direction for future work.

Broader impact statement

This work lies in the realm of "foundational RL" in that it contributes to the fundamental understanding and development of reinforcement learning algorithms and theory. As such, despite us agreeing in the importance of this discussion, our work is quite far removed from ethical issues and potential societal consequences.

References

Alessandro Abate. Approximation metrics based on probabilistic bisimulations for general state-space Markov processes: A survey. *Electr. Notes Theor. Comput. Sci.*, 297:3–25, 2013.

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Philip Amortila, Marc G Bellemare, Prakash Panangaden, and Doina Precup. Temporally extended metrics for Markov decision processes. In *AAAI Workshop on Safe AI*, 2019.

T. W. Archibald, K. I. M. McKinnon, and L. C. Thomas. On the generation of Markov decision processes. *The Journal of the Operational Research Society*, 46(3):354–361, 1995.

Giorgio Bacci, Giovanni Bacci, Kim G Larsen, and Radu Mardare. Computing behavioral distances, compositionally. In *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2013a.

Giorgio Bacci, Giovanni Bacci, Kim G Larsen, and Radu Mardare. On-the-fly exact computation of bisimilarity distances. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2013b.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado Van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic Markov Decision Processes. In *AAAI Conference on Artificial Intelligence*, 2020.

Pablo Samuel Castro and Doina Precup. Using bisimulation for policy transfer in MDPs. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010.

Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A Research Framework for Deep Reinforcement Learning. 2018.

Lenaic Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. Faster wasserstein distance estimation with the sinkhorn divergence. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gheorghe Comanici, Prakash Panangaden, and Doina Precup. On-the-fly algorithms for bisimulation metrics. In *International Conference on Quantitative Evaluation of Systems (QEST)*, 2012.

Gheorghe Comanici, Doina Precup, and Prakash Panangaden. Basis refinement strategies for linear value function approximation in MDPs. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018a.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence*, 2018b.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Comput.*, 5(4):613–624, July 1993.

Josée Desharnais, Vineet Gupta, Radhakrishnan Jagadeesan, and Prakash Panangaden. Metrics for labeled Markov systems. In *International Conference on Concurrency Theory (CONCUR)*, 1999.

Josée Desharnais, Vineet Gupta, Radhakrishnan Jagadeesan, and Prakash Panangaden. A metric for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, June 2004.

Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch Wasserstein: asymptotic and gradient properties. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv*, 2021.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv*, 2019.

Norm Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.

Norm Ferns, Pablo Samuel Castro, Doina Precup, and Prakash Panangaden. Methods for computing state similarity in Markov decision processes. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous Markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for Markov decision processes with infinite state spaces. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined reinforcement learning via abstract representations. In *AAAI Conference on Artificial Intelligence*, 2019.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning (ICML)*, 2019.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

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

Wenshuo Guo, Nhat Ho, and Michael I. Jordan. Fast algorithms for computational optimal transport and Wasserstein barycenter. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020a.

Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2020b.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining Improvements in Deep Reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2018.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations (ICLR)*, 2017.

Nicholas K Jong and Peter Stone. State abstraction discovery from irrelevant state variables. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

Dexter Kozen. Coinductive proof principles for stochastic processes. *Logical methods in computer science*, 3, November 2007.

Charline Le Lan, Marc G. Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2021.

Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2014.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for MDPs. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2006.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the Arcade Learning Environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research*, 8:2169–2231, December 2007.

Steve Matthews. Partial metric topology. *Annals of the New York Academy of Sciences*, 728(1): 183–197, 1994.

Andrew Kachites McCallum. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, The University of Rochester, 1996.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. UMAP: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Johan S Obando-Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *International Conference on Machine Learning (ICML)*, 2021.

Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Bilal Piot, Matthieu Geist, and Olivier Pietquin. Difference of convex functions programming for reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*. 2014.

Balaraman Ravindran and Andrew G. Barto. SMDP homomorphisms: An algebraic approach to abstraction in semi-Markov decision processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

Sherry Shanshan Ruan, Gheorghe Comanici, Prakash Panangaden, and Doina Precup. Representation discovery for MDPs using bisimulation metrics. In *AAAI Conference on Artificial Intelligence*, 2015.

Anirban Santara, Rishabh Madan, Balaraman Ravindran, and Pabitra Mitra. ExTra: Transfer-guided exploration. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2019.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

R.S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

Jonathan Taylor. Lax probabilistic bisimulation. Master's thesis, McGill University, 2008.

Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate MDP homomorphisms. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Munchausen reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Cédric Villani. *Optimal Transport*. Springer-Verlag Berlin Heidelberg, 2008.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

Szymon Łukaszyk. A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, 33:299–304, 03 2004.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See page 9.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code is included as part of the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Full details provided in the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code provided in the supplemental material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]