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

Pablo Samuel Castro*
Google Research, Brain Team

Tyler Kastner*
McGill University

Prakash Panangaden
McGill University

Mark Rowland
DeepMind

June 16, 2021

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.

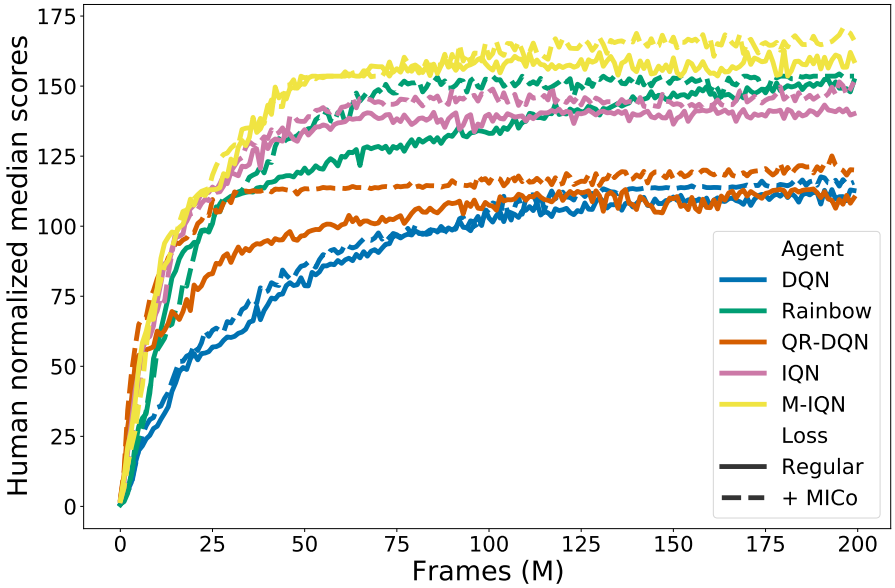

Figure 1: Median human normalized scores across 60 Atari 2600 games, averaged over 5 independent runs.

*Equal contribution. Correspondence to Pablo Samuel Castro: psc@google.com.

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 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 7).

2 Background

Before describing the details of our contributions, we give a brief overview of the required background in reinforcement learning and bisimulation.

2.1 Reinforcement learning

Denoting by $\mathscr{P}(S)$ the set of probability distributions on a set S, we define a Markov decision process $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$ as:

- A finite state space $\mathcal{X}$;
- A finite action space $\mathcal{A}$;
- A transition kernel $P : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathcal{X})$;
- A reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$;
- A discount factor $\gamma \in [0, 1)$.

For notational convenience we introduce the notation $P_x^a \in \mathscr{P}(\mathcal{X})$ for the next-state distribution given state-action pair (x, a), and r_x^a for the corresponding immediate reward.

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:

$$V^\pi(x) := \mathbb{E}_{a\sim\pi(x)} \left[r_x^a + \gamma \mathbb{E}_{x'\sim P_x^a}[V^\pi(x')] \right] .$$

It can be shown that this recurrence uniquely defines V^π through a contraction mapping argument [Bertsekas and Tsitsiklis, 1996].

The control problem is concerned with finding the optimal policy

$$\pi^* = \arg \max_{\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}} V^\pi .$$

It can be shown that while the optimisation problem above appears to have multiple objectives (one for each coordinate of V^π, there is in fact a policy $\pi^* \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$ that simultaneously maximises all coordinates of V^π, and that this policy can be taken to be deterministic; that is, for each $x \in \mathcal{X}$, $\pi(\cdot|x) \in \mathscr{P}(\mathcal{A})$ attributes probability 1 to a single action. In reinforcement learning in particular, we are often interested in finding, or approximating, π^* from direct interaction with the MDP in question via sample trajectories, *without knowledge of P or r* (and sometimes not even $\mathcal{X}$).

2.2 Metrics

A *metric* d on a set X is a function $d : X \times X \to [0, \infty)$ respecting the following axioms for any $x, y, z \in X$:

1. **Identity of indiscernibles:** $d(x, y) = 0 \iff x = y$;
2. **Symmetry:** $d(x, y) = d(y, x)$;
3. **Triangle inequality:** $d(x, y) \leq d(x, z) + d(z, y)$.

A *pseudometric* is similar, but the "identity of indiscernibles" axiom is weakened:

1. $x = y \implies d(x, y) = 0$;
2. $d(x, y) = d(y, x)$;
3. $d(x, y) \leq d(x, z) + d(z, y)$.

Note that the weakened first condition *does* allow one to have $d(x, y) = 0$ when $x \neq y$.

A *(pseudo)metric space* (X, d) is defined as a set X together with a (pseudo)metric d defined on X.

2.3 State similarity and bisimulation metrics

Bisimulation is a fundamental notion of behavioural equivalence introduced by Park and Milner [Milner, 1989] in the early 1980s in the context of nondeterministic transition systems. The probabilistic analogue was introduced by Larsen and Skou [1991]. The notion of an equivalence relation is not suitable to capture the extent to which quantitative systems may resemble each other in behaviour. To provide a quantitative notion, bisimulation metrics were introduced by Desharnais et al. [1999, 2004] in the context of probabilistic transition systems without rewards. In reinforcement learning the reward is an important ingredient, accordingly the *bisimulation metric* for states of MDPs was introduced by Ferns et al. [2004].

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.

Central to the definition of the bisimulation metric is the operator $T_k : \mathcal{M}(\mathcal{X}) \to \mathcal{M}(\mathcal{X})$, defined over $\mathcal{M}(\mathcal{X})$, the space of pseudometrics on $\mathcal{X}$. We now turn to the definition of the operator itself, given by

$$T_k(d)(x, y) = \max_{a \in \mathcal{A}} [|r_x^a - r_y^a| + \gamma W_d(P_x^a, P_y^a)] \,,$$

for each $d \in \mathcal{M}(\mathcal{X})$, and each $x, y \in \mathcal{X}$. It can be verified that the function $T_K(d) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfies the properties of a pseudometric, so under this definition T_K does indeed map $\mathcal{M}(\mathcal{X})$ into itself.

The other central mathematical concept underpinning the operator T_K is the Kantorovich distance W_d[1] using base metric d. W_d is formally a pseudometric over the set of probability distributions $\mathscr{P}(\mathcal{X})$, defined as the solution to an optimisation problem. The problem specifically is formulated as finding an optimal coupling between the two input probability distributions that minimises a notion of transport cost associated with d. Mathematically, for two probability distributions $\mu, \mu' \in \mathscr{P}(\mathcal{X})$, we have

$$W_d(\mu, \mu') = \min_{\substack{(Z, Z') \\ Z \sim \mu, Z' \sim \nu'}} \mathbb{E}[d(Z, Z')] \,.$$

[1] Commonly known as the Wasserstein distance.

Note that the pair of random variables (Z, Z') attaining the minimum in the above expression will in general not be independent. That the minimum is actually attained in the above example in the case of a finite set $\mathcal{X}$ can be seen by expressing the optimisation problem as a linear program. Minima are obtained in much more general settings too; see Villani [2008].

The operator T_K can be analysed in a similar way to standard operators in dynamic programming for reinforcement learning. It can be shown that it is a contraction mapping with respect to the L^∞ metric over $\mathcal{M}(\mathcal{X})$, and that $\mathcal{M}(\mathcal{X})$ is a complete metric space with respect to the same metric [Ferns et al., 2011]. Thus, by Banach's fixed point theorem, T_K has a unique fixed point in $\mathcal{M}(\mathcal{X})$, and repeated application of T_K to any initial pseudometric will converge to this fixed point.

Finally, 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.

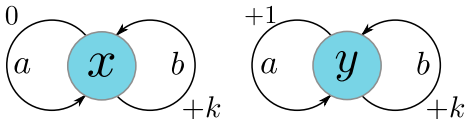

Figure 2: An MDP illustrating that Equation (2) is not generally satisfied. Here, $d^{\sim}(x, y) = (1-\gamma)^{-1}$, but for the policy $\pi(b|x) = 1, \pi(a|y) = 1$, we have $|V^{\pi}(x) - V^{\pi}(y)| = k(1-\gamma)^{-1}$

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., 2020, 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

$$|V^{\pi}(x) - V^{\pi}(y)| \leq d^{\sim}(x, y) \tag{2}$$

for arbitrary policies $\pi \in \Pi$; a simple example is illustrated in Figure 2. More generally, notions of state similarity that the bisimulation metric encodes may not be closely related to behavioural similarity under the 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.

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 a policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, the *MICo update operator* $T_M^\pi : \mathbb{R}^{\mathcal{X}\times\mathcal{X}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ is defined by

$$(T_M^\pi U)(x,y) = |r_x^\pi - r_y^\pi| + \gamma \mathbb{E}_{\substack{x'\sim P_x^\pi \\ y'\sim P_y^\pi}} [U(x',y')] \tag{3}$$

for all functions $U : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, 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 MICo operator T_M^π is a contraction mapping on $\mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ with respect to the L^∞ norm.

Proof. Let $U, U' \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$. Then note that

$$\begin{aligned}
&|(T^\pi U)(x,y) - (T^\pi U')(x,y)| \\
=&\left|\gamma \sum_{x',y'} \pi(a|x)\pi(b|y) P_x^a(x') P_y^b(y')(U-U')(x',y')\right| \\
\leq&\gamma \|U - U'\|_\infty .
\end{aligned}$$

for any $x, y \in \mathcal{X}$, as required. □

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^π_M has a unique fixed point $U^\pi \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$, and repeated application of T^π_M 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^π_M 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:

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

for all $U \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ and $(x, y) \in \mathcal{X} \times \mathcal{X}$, as required. □

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^π_M 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^π_M 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^π_M 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

$$U_{t+1}(x, y) \leftarrow (1 - \epsilon_t(x, y))U_t(x, y) + \epsilon_t(x, y)(|r - \tilde{r}| + \gamma U_t(x', y')) \qquad (4)$$

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 (4). Suppose 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 policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$ 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 to show that if

$$|V^\pi(x) - V^\pi(y)| \leq U(x, y) \text{ for all } x, y \in \mathcal{X}\,, \tag{5}$$

for some $U \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ symmetric in its two arguments, then we also have

$$|V^\pi(x) - V^\pi(y)| \leq (T_M^\pi U)(x, y) \text{ for all } x, y \in \mathcal{X}\,.$$

Since the hypothesis holds for the constant function $U(x, y) = 2R_{\max}/(1-\gamma)$, and T_M^π contracts around U^π, the conclusion then follows. Therefore, suppose Equation (5) holds. Then we have

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

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 a novel notion of distance which we name *diffuse metrics*, which we define below.

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:

1. $d(x, y) \geq 0$ for any $x, y \in \mathcal{X}$,
2. $d(x, y) = d(y, x)$ for any $x, y \in \mathcal{X}$,
3. $d(x, y) \leq d(x, z) + d(y, z)$ $\forall x, y, z \in \mathcal{X}$.

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.

The notion of a distance function having non-zero self distance was first introduced by Matthews [1994] who called it a *partial metric*. We define it below:

Definition 4.10. Given a set $\mathcal{X}$, a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a partial metric if

1. $x = y \iff d(x, x) = d(y, y) = d(x, y)$ for any $x, y \in \mathcal{X}$,
2. $d(x, x) \leq d(y, x)$ for any $x, y \in \mathcal{X}$,
3. $d(x, y) = d(y, x)$ for any $x, y \in \mathcal{X}$,
4. $d(x, y) \leq d(x, z) + d(y, z) - d(z, z)$ $\forall x, y, z \in \mathcal{X}$.

This definition was introduced to recover a proper metric from the distance function: that is, given a partial metric d, one is guaranteed that $\tilde{d}(x, y) = d(x, y) - \frac{1}{2}\left(d(x, x) + d(y, y)\right)$ is a proper metric.

The above definition is still too stringent for the Łukaszyk–Karmowski distance (and hence MICo distance), since it fails axiom 4 (the modified triangle inequality) as shown in the following counterexample.

Example 4.11. The Łukaszyk–Karmowski distance does not satisfy the modified triangle inequality: let $\mathcal{X}$ be $[0,1]$, and ρ be the Euclidean distance $|\cdot|$. Let μ,ν be Dirac measures concentrated at 0 and 1, and let η be $\frac{1}{2}(\delta_0 + \delta_1)$. Then one can calculate that $d_{LK}(\rho)(\mu,\nu) = 1$, while $d_{LK}(\rho)(\mu,\eta) + d_{LK}(\rho)(\nu,\eta) - d_{LK}(\rho)(\eta,\eta) = 1/2$, breaking the inequality.

In terms of the Łukaszyk–Karmowski distance, the MICo distance can be written as the fixed point

$$U^\pi(x,y) = |r^\pi_x - r^\pi_y| + d_{\mathrm{LK}}(U^\pi)(P^\pi_x, P^\pi_y)\,.$$

This characterisation leads to the following result.

Proposition 4.12. 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

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

as required. □

The counterpart of the role played by Dirac distributions in the Łukaszyk–Karmowski distance for the MICo metric is deterministic MDPs. In particular, 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 $x, y \in \mathcal{X}$.

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 (3) 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.

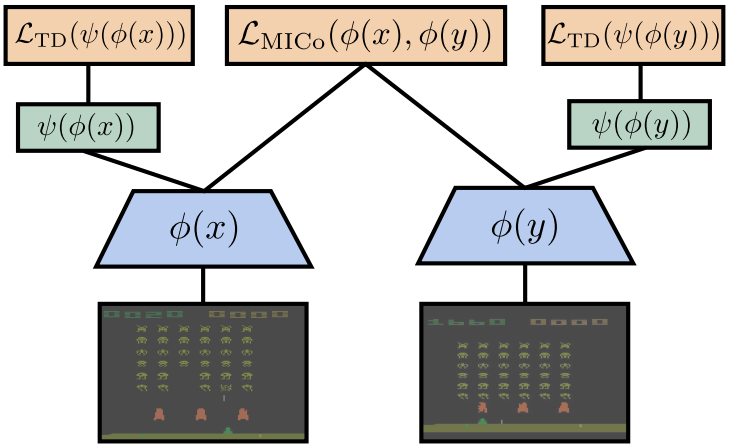

Figure 3: Illustration of network architecture for learning MICo.

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.

Based on Equation (3), 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

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

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 3 (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 3, 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:

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

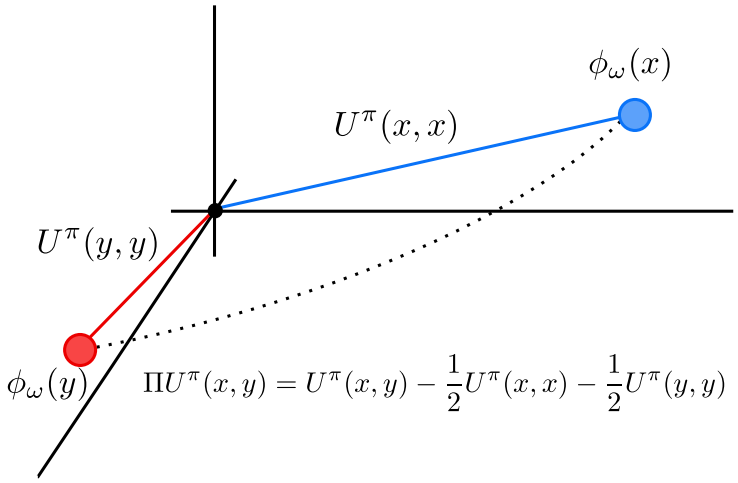

Figure 4: The projection of MICo distances onto representation space.

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 two states x, y, and a policy $\pi \in \Pi$ 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

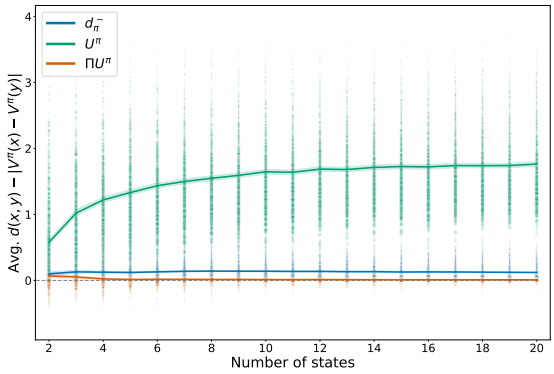

Figure 5: The gap between the difference in values and the various distances for Garnet MDPs with varying numbers of actions (represented by the size of the circles).

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 in an unbiased manner we make use of Garnet MDPs, which are 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 5, our metric *does* on average provide an upper bound 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, despite the reduction Π.

5.2 State features

In order to investigate the usefuleness 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][2]. 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 three domains in Figure 6. Despite the independent couplings, ΠU^π performs on par with π-bisimulation, which optimizes over all transition probability couplings, suggesting that ΠU^π yields good representations.

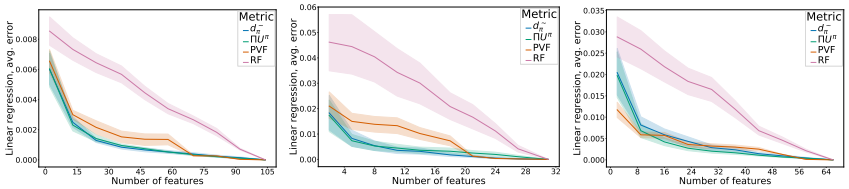

Figure 6: Average error when performing linear regression on varying numbers of features, averaged over 10 runs; shaded areas represent 95% confidence intervals. **Left:** four-rooms GridWorld [Sutton et al., 1999]; **Center:** The mirrored rooms introduced by Castro [2020]; **Right:** The grid task introduced by Dayan [1993].

6 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].

The code necessary to run these experiments is available on GitHub: https://github.com/google-research/google-research/tree/master/mico.

We will first describe the regular network and training setup for these agents so as to facilitate the description of our loss.

6.1 Baseline network and loss description

The networks used by Dopamine for the ALE consist of 3 convolutional layers followed by two fully-connected layers (the output of the networks depends on the agent). We denote the output of the convolutional layers by ϕ_ω with parameters ω, and the remaining fully connected layers by ψ_ξ with parameters ξ. Thus, given an input state x (e.g. a stack of 4 Atari frames), the output of the network is $Q_{\xi,\omega}(x,\cdot) = \psi_\xi(\phi_\omega(x))$. Two copies of this network are maintained: an *online* network and a *target* network; we will denote the parameters of the target network by $\bar{\xi}$ and $\bar{\omega}$. During learning, the parameters of the online network

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

are updated every 4 environment steps, while the target network parameters are synced with the online network parameters every 8000 environment steps. We refer to the loss used by the various agents considered as $\mathcal{L}_{\text{TD}}$; for example, for DQN this would be:

$$\mathcal{L}_{\text{TD}}(\xi, \omega) := \mathbb{E}_{(x,a,r,x')\sim\mathcal{D}}\left[\rho\left(r + \gamma \max_{a'\in\mathcal{A}} Q_{\bar{\xi},\bar{\omega}}(x', a') - Q_{\xi,\omega}(x, a)\right)\right],$$

where $\mathcal{D}$ is a replay buffer with a capacity of 1M transitions, and ρ is the Huber loss.

6.2 MICo loss description

We will be applying the MICo loss to $\phi_\omega(x)$. As described in Section 5, we express the distance between two states as:

$$U_\omega(x, y) = \frac{\|\phi_\omega(x)\|_2 + \|\phi_{\bar{\omega}}(y)\|_2}{2} + \beta\theta(\phi_\omega(x), \phi_{\bar{\omega}}(y)),$$

where $\theta(\phi_\omega(x), \phi_{\bar{\omega}}(y))$ is the angle between vectors $\phi_\omega(x)$ and $\phi_{\bar{\omega}}(y)$ and β is a scalar. Note that we are using the target network for the y representations; this was done for learning stability. We used $\beta = 0.1$ for the results in the main paper, but present some results with different values of β below.

In order to get a numerically stable operation, we implement the angular distance between representations $\phi_\omega(x)$ and $\phi_\omega(y)$ according to the calculations

$$\text{CS}(\phi_\omega(x), \phi_\omega(y)) = \frac{\langle\phi_\omega(x), \phi_\omega(y)\rangle}{\|\phi_\omega(x)\|\|\phi_\omega(y)\|}$$
$$\theta(\phi_\omega(x), \phi_\omega(y)) = \arctan2\left(\sqrt{1 - \text{CS}(\phi_\omega(x), \phi_\omega(y))^2}, \text{CS}(\phi_\omega(x), \phi_\omega(y))\right).$$

Based on Equation (3), our learning target is then (note the target network is used for both representations here):

$$T^U_{\bar{\omega}}(r_x, x', r_y, y') = |r_x - r_y| + \gamma U_{\bar{\omega}}(x', y'),$$

and the loss is

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

We found it important to use the Huber loss 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.

As mentioned in Section 5, we use the same mini-batch sampled for $\mathcal{L}_{\text{TD}}$ for computing $\mathcal{L}_{\text{MICo}}$. Specifically, we follow the method introduced by Castro [2020] for constructing new matrices that allow us to compute the distances between all pairs of sampled states (see code for details on matrix operations).

Our combined loss is then

$$\mathcal{L}_\alpha(\xi, \omega) = (1 - \alpha)\mathcal{L}_{\text{TD}}(\xi, \omega) + \alpha\mathcal{L}_{\text{MICo}}(\omega).$$

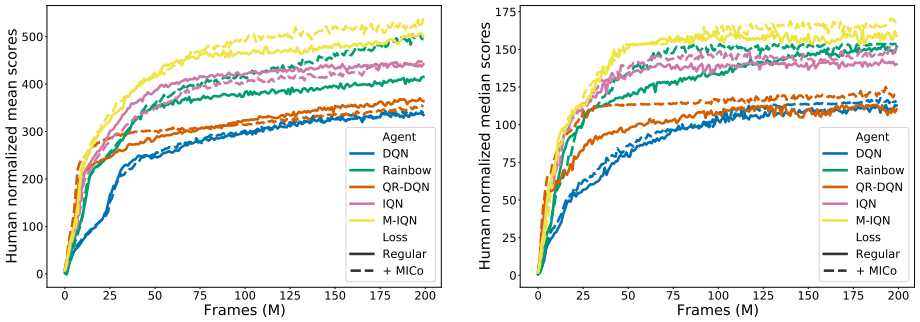

Figure 7: Mean (left) and median (right) human normalized scores across 60 Atari 2600 games, averaged over 5 independent runs.

6.3 Results

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.[3] 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 7 and Figure 8; as can be seen, our loss is able to provide good improvements over the agents they are based on, suggesting that the MICo loss can help learn better representations for control.

The learning curves for all agents and all games are provided in Appendix A.

7 Related Work

Bisimulation originated as a fundamental notion of behavioural equivalence in concurrency theory [Milner, 1989, Larsen and Skou, 1991, van Breugel and Wor-

[3] 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].

rell, 2001a,b], and was later extended from a binary predicate to a quantitative metric notion by Desharnais et al. [1999, 2004]. Bisimulation metrics were first introduced for MDPs by Ferns et al. [2004], and this work has since been steadily 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]. See Li et al. [2006] for a review of 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.

Lastly, 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].

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.

9 Acknowledgements

The authors would like to thank Gheorghe Comanici, Rishabh Agarwal, Nino Vieillard, and Matthieu Geist for their valuable feedback on the paper and experiments. Pablo Samuel Castro would like to thank Roman Novak and Jascha Sohl-Dickstein for their help in getting angular distances to work stably!

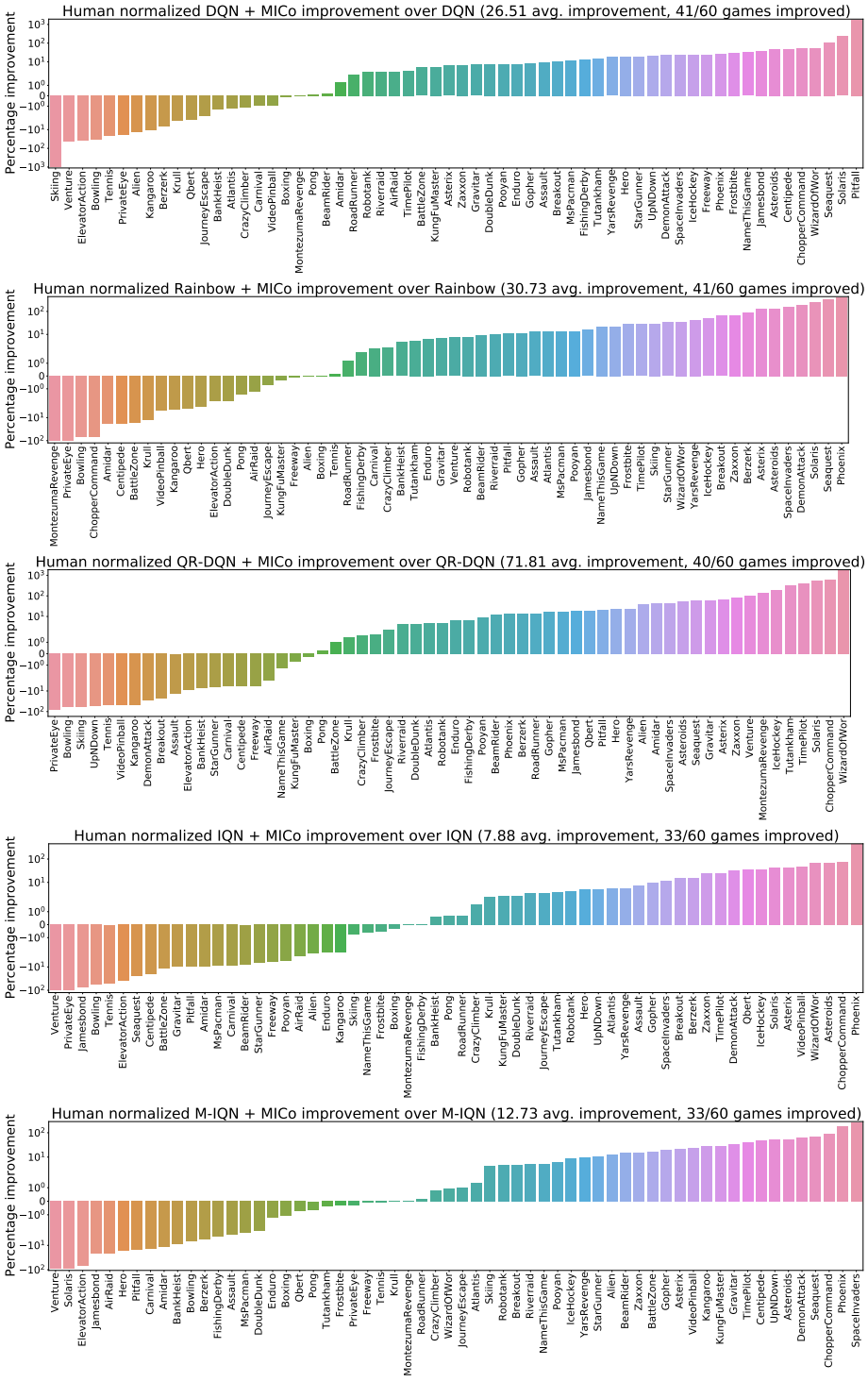

Figure 8: From top to bottom, percentage improvement in returns (averaged over the last 5 iterations) when adding $\mathcal{L}_{\text{MICo}}$ to DQN, Rainbow, QR-DQN, IQN, and M-DQN. The results for are averaged over 5 independent runs.

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.

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.

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)*, 2020.

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.

Kim G Larsen and Arne Skou. Bisimulation through probablistic testing. *Information and Computation*, 94:1–28, 1991.

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.

R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

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.

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.

Franck van Breugel and James Worrell. An algorithm for quantitative verification of probabilistic transition systems. In *International Conference on Concurrency Theory (CONCUR)*, 2001a.

Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic systems. In *International Colloquium on Automata, Languages and Programming*, 2001b.

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.

Appendices

A Complete learning curves

We provide complete results to complement the results presented in the main paper in Figure 9, Figure 10, Figure 11, Figure 12, and Figure 13.

B Hyperparameter sweeps

In Figure 14 we demonstrate the performance of the MICo loss when added to Rainbow over a number of different values of α and β. For each agent, we ran a similar hyperparameter sweep over α and β on the same six games displayed in Figure 14 to determine settings to be used in the full ALE experiments.

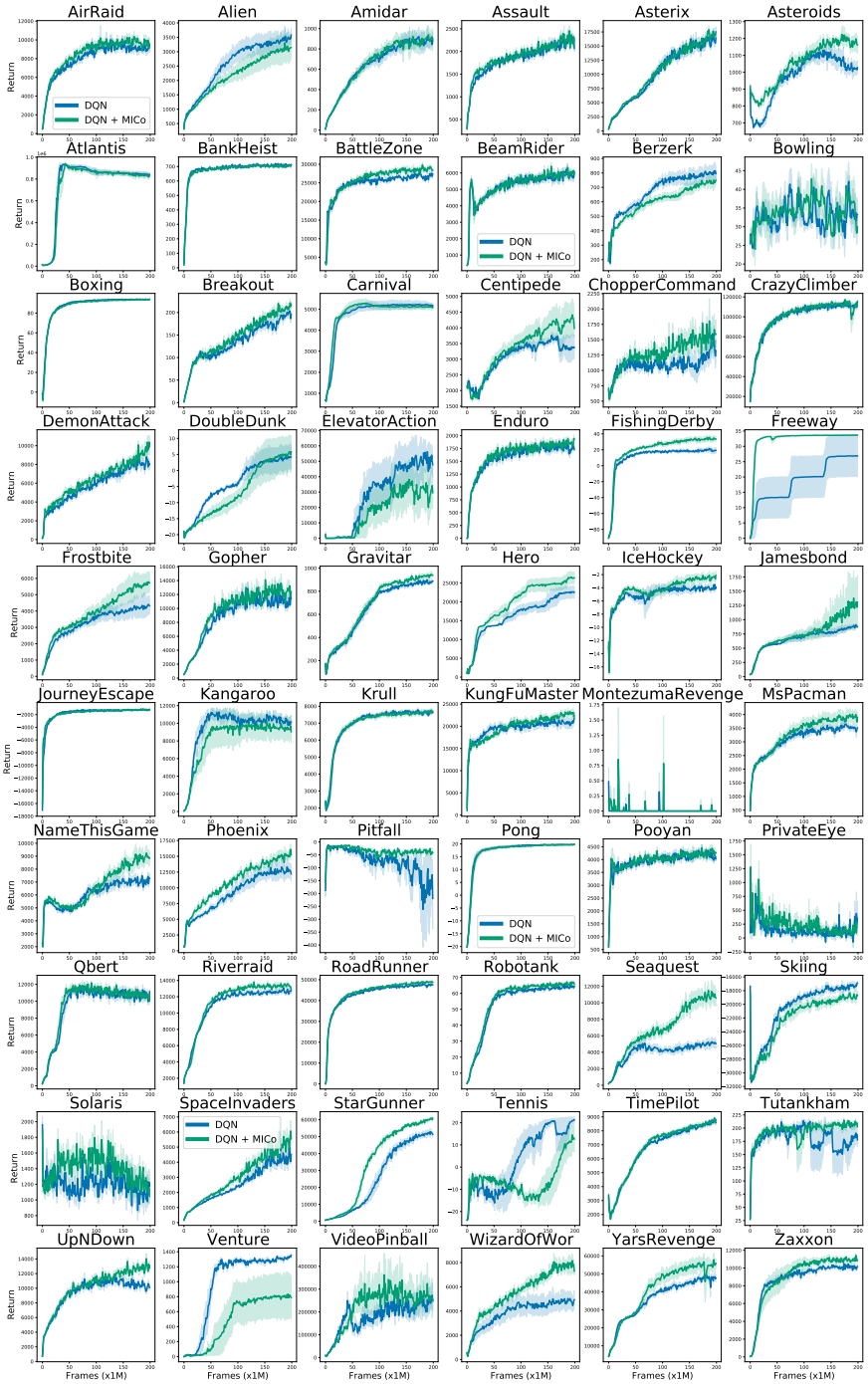

Figure 9: Training curves for DQN agents. The results for all games and agents are over 5 independent runs, and shaded regions report 75% confidence intervals.

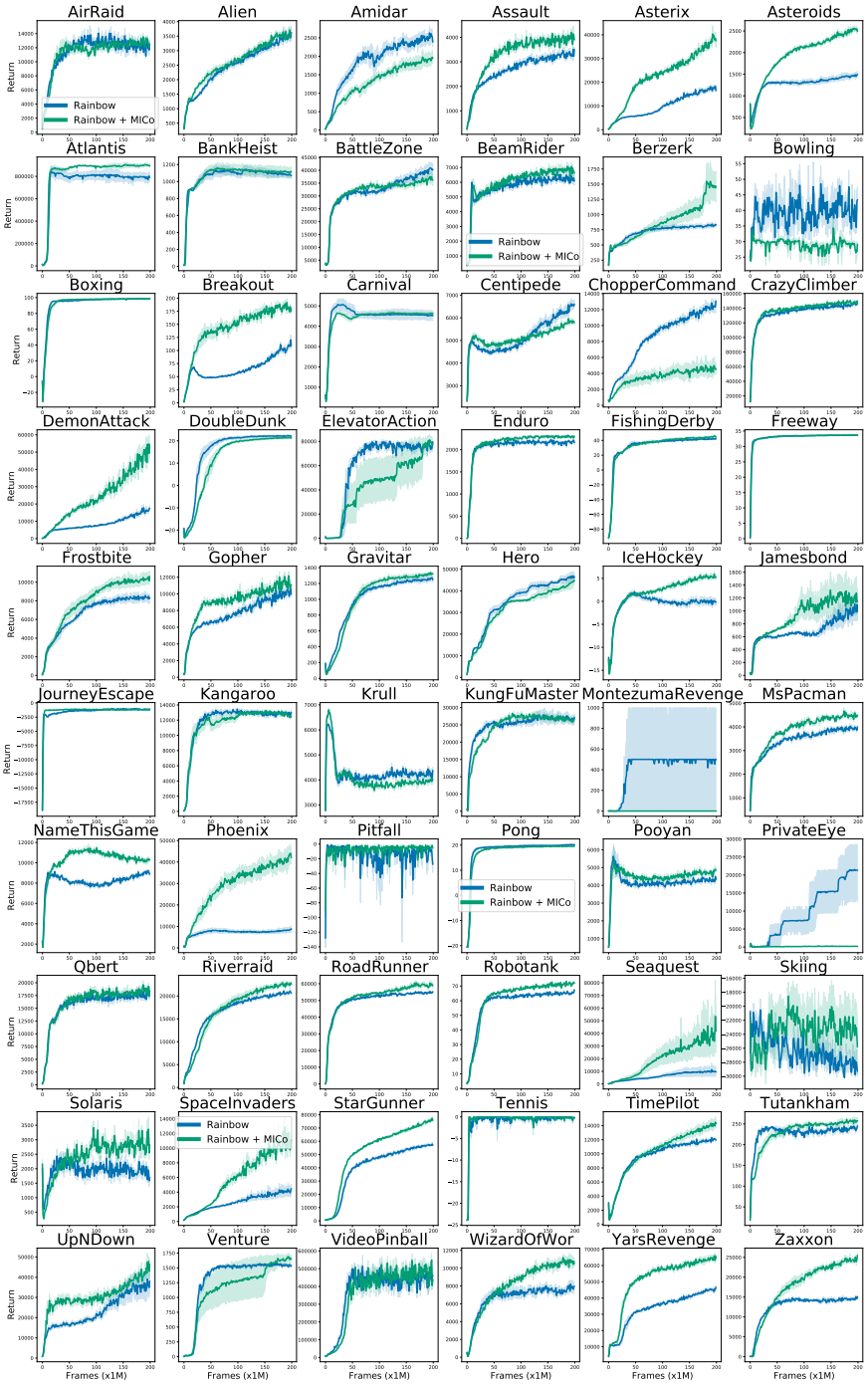

Figure 10: Training curves for Rainbow agents. The results for all games and agents are over 5 independent runs, and shaded regions report 75% confidence intervals.

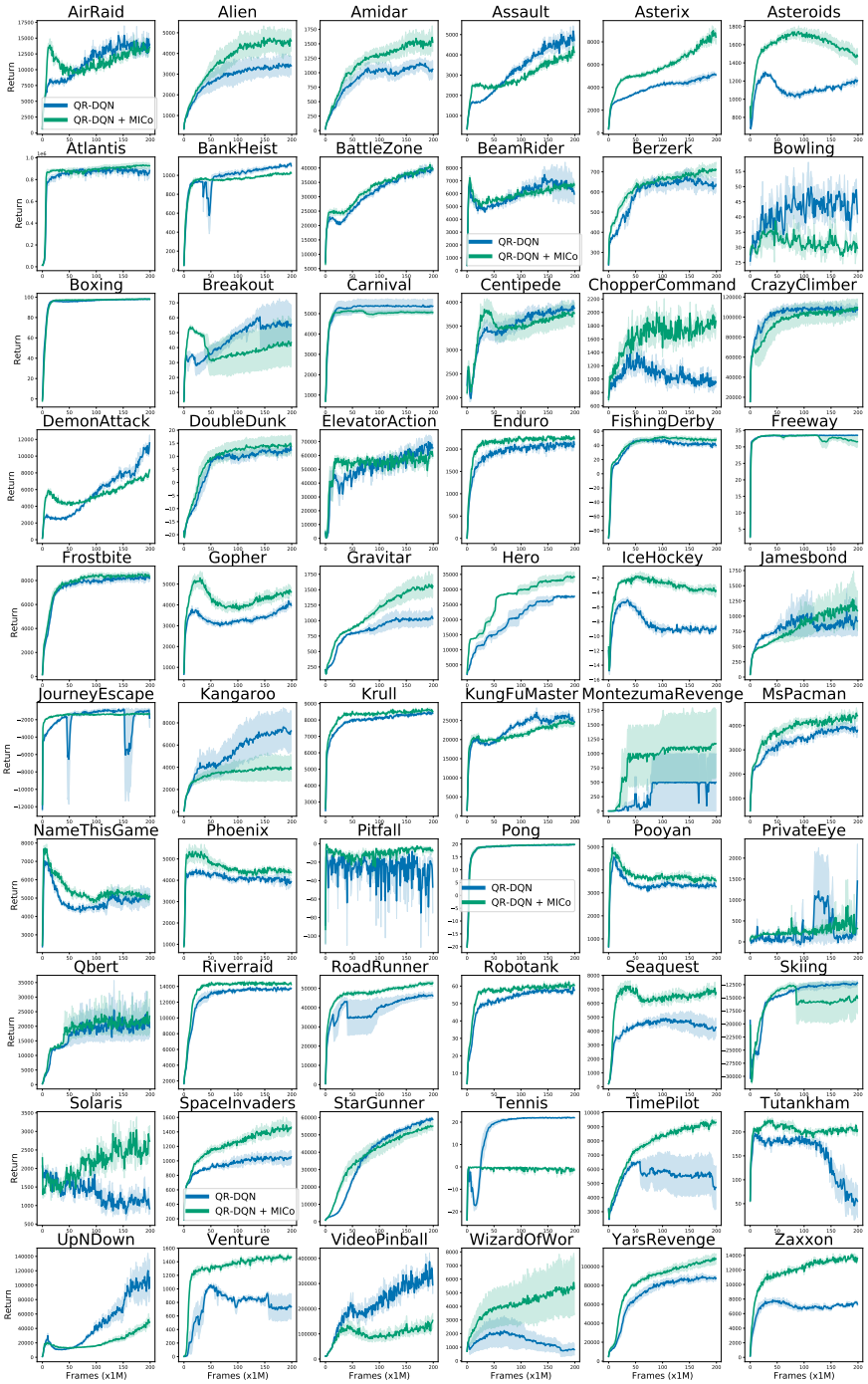

Figure 11: Training curves for QR-DQN agents. The results for all games and agents are over 5 independent runs, and shaded regions report 75% confidence intervals.

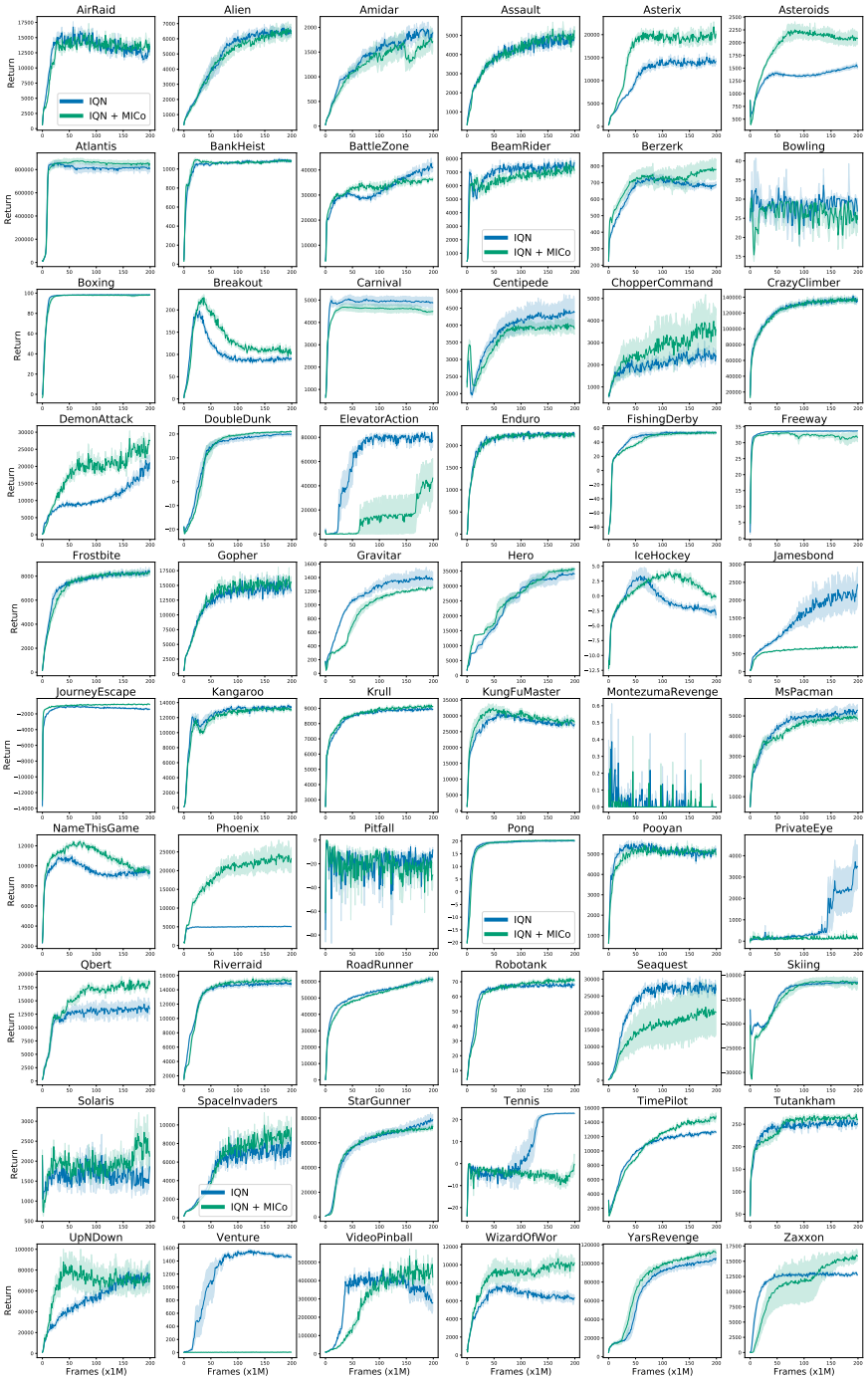

Figure 12: Training curves for IQN agents. The results for all games and agents are over 5 independent runs, and shaded regions report 75% confidence intervals.

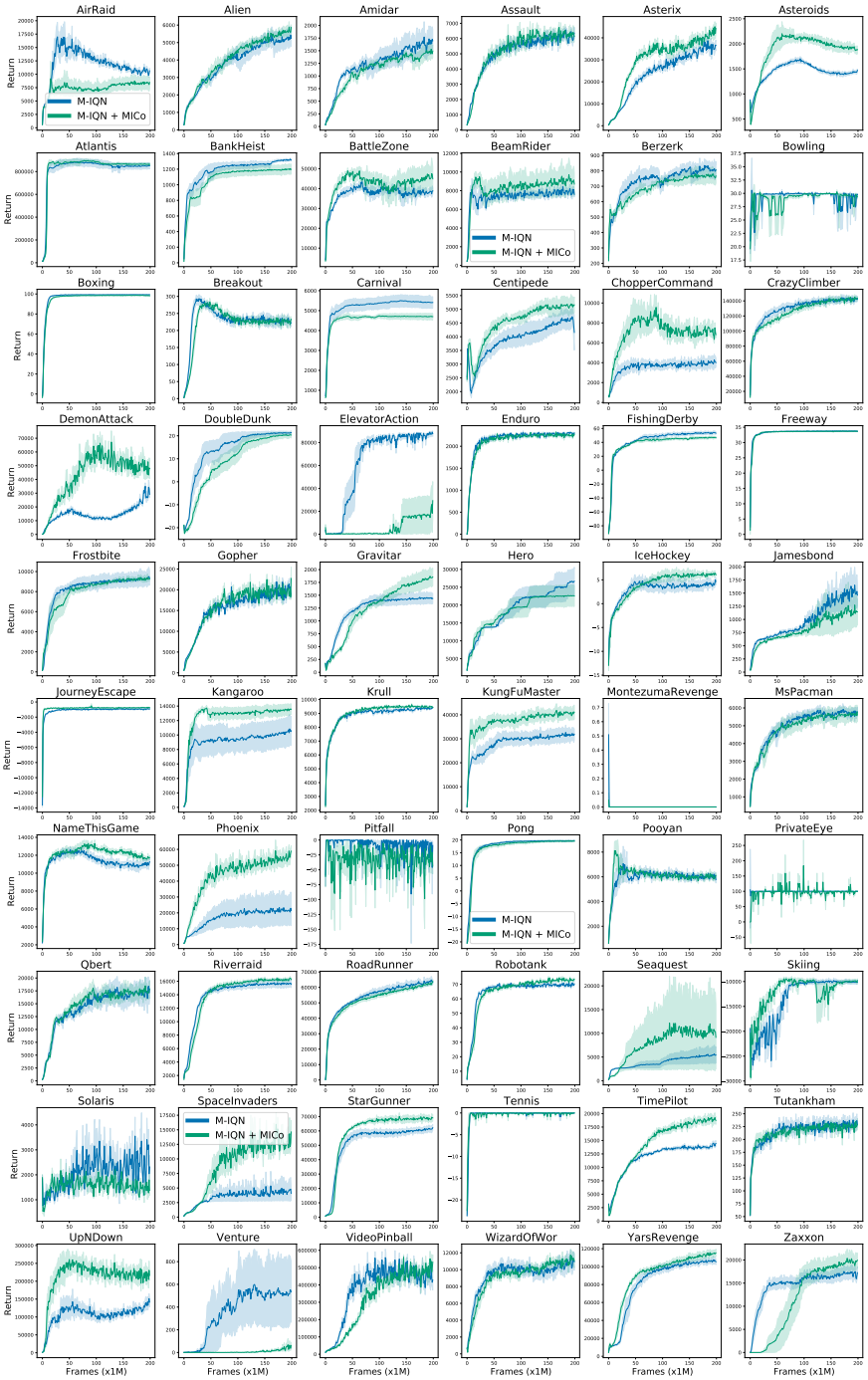

Figure 13: Training curves for M-IQN agents. The results for all games and agents are over 5 independent runs, and shaded regions report 75% confidence intervals.

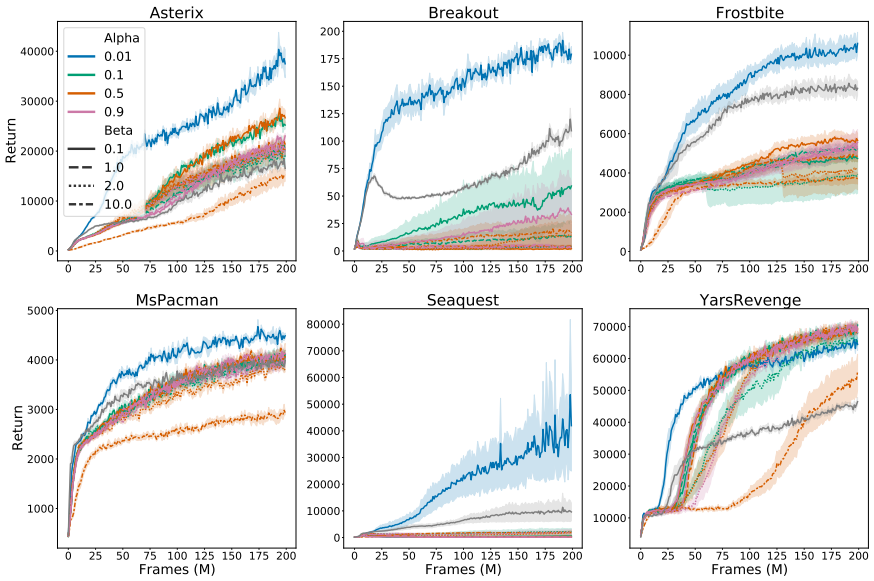

Figure 14: Sweeping over various values of α and β when adding the MICo loss to Rainbow. The grey line represents regular Rainbow.