Generalization Across Observation Shifts in Reinforcement Learning

Anuj Mahajan *
Amazon
anuj.mahajan.ai@gmail.com

Amy Zhang
University of Texas at Austin
amy.zhang@austin.utexas.edu

Abstract

Learning policies which are robust to changes in the environment are critical for real world deployment of Reinforcement Learning agents. They are also necessary for achieving good generalization across environment shifts. We focus on bisimulation metrics, which provide a powerful means for abstracting task relevant components of the observation and learning a succinct representation space for training the agent using reinforcement learning. In this work, we extend the bisimulation framework to also account for context dependent observation shifts. Specifically, we focus on the simulator based learning setting and use alternate observations to learn a representation space which is invariant to observation shifts using a novel bisimulation based objective. This allows us to deploy the agent to varying observation settings during test time and generalize to unseen scenarios. We further provide novel theoretical bounds for simulator fidelity and performance transfer guarantees for using a learnt policy to unseen shifts. Empirical analysis on the high-dimensional image based control domains demonstrates the efficacy of our method.

1 Introduction

Many practical scenarios in reinforcement learning (RL) applications require the agent to be robust to changes in the observations space between training and deployment. Such changes can occur due to lack of complete information about the deployment environment which often happens as the training environment is usually highly controlled or simulators are used for training the agent, both of these scenarios seldom capture the complexity and noisiness of the real world. Moreover, these changes can also occur due to various practical errors and constraints under which autonomous agents need to be deployed (e.g. variations in sensor position and fitting on automobiles, change in calibration settings of visual input, change in sensor types due to upgrades, calibration changes due to wear and tear etc.).

While existing methods aimed at obtaining better generalization in RL can be partially applied to the above problem, they hardly utilise the rich underlying structure that can enable efficient learning of policies which generalize well across the observation shifts. For instance, methods like domain randomization (Zhao et al., 2020) which work well for supervised perception problems in robotics are insufficient for obtaining good performance on control tasks. Similarly, methods for RL which aim at using unsupervised data for learning control representations: data augmentation (Laskin et al., 2020b; Kostrikov et al., 2020), contrastive learning (Oord et al., 2018), reconstruction (Lange et al., 2012; Hafner et al., 2019) are not well aligned with the objective of maximizing rewards in complex domains. Further, the presence of task irrelevant noise in the environment make it difficult for these methods to generalise across the changes in the observation space. While, state abstraction based methods like bisimulation (Zhang et al., 2021; Gelada et al., 2019) to which our method is closely related, can help ignore irrelevant task features, they do not fully exploit the structure present in observation shift setting towards ensuring better generalization.

*Corresponding author. Part of work done as PhD student at University of Oxford.

Preprint. Under review.

In this work, we propose a novel solution to the aforementioned problem using the concept of conditional bisimulation and application of simulator/specialized setup during train time which help explicitly teach the agent, the similarities across changes in the observation space. Our method leverages the MDP level isomorphism (Ravindran, 2004) in the observation shift setting for obtaining a richer representation loss. Our methods offers two-fold advantage:

- We can learn representations which are robust to shifts in observation space in a sample efficient manner.
- We learn to ignore task irrelavant features as our metric is grounded in rewards.

2 Background

Reinforcement Learning: A Markov Decision Process (MDP) is formally defined as a tuple $\langle S, U, P, r, \gamma, \rho \rangle$. Here S is the state space of the environment and ρ is the initial state distribution. At each time step t, an agent observes the state $s \in S$ and chooses an action $a \in U$ using its policy $\pi : S \to \mathcal{P}(U)$, where $\mathcal{P}(\cdot)$ represents the space of distributions on the argument set. This leads to a state transition governed by the distribution $P(s'|s, a) : S \times U \times S \to [0, 1]$, and the agent receives reward $r(s, a) : S \times U \to [0, R_{max}]$ which can be potentially stochastic. We consider the discounted infinite horizon setting, where the discount factor is given by $\gamma \in [0, 1)$. The state-action trajectory of the agent is represented by $\tau \in T \equiv (S \times U)^*$, we overload the notation to also include rewards as necessary. The value of a policy is defined as: $J^\pi = \mathbb{E}_{\pi,\rho}\left[\sum_{t=0}^{\infty} \gamma^t r_\tau(s_t)\right]$ where the expectation on the RHS is well defined given bounds on rewards and γ. We also define three other useful functions: (1) $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t)|s_0 = s, a_0 = a]$, (2) $V^\pi(s) = \mathbb{E}_{a\sim\pi} Q^\pi(s, a)$, (3) $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$, respectively called the action-value, value and advantage functions. The goal of the MDP problem is to find the optimal policy π^* corresponding to the optimal policy value J^*. It is well known that a deterministic optimal policy always exists for finite MDPs. Further, the optimal value function V^* and optimal action value function Q^* also exhibit important properties like uniqueness and point-wise function dominance over the entire domain (Sutton & Barto, 2011). A standard assumption in the reinforcement learning (RL) setting is that both the rewards and transition kernels are not known to the agent.

Rich observations and Contextual MDPs: An important class of MDP arises when we consider the presence of an underlying parametrized context θ, which governs the rewards and transitions in the MDP framework. This extension is called the Contextual MDP setting (CMDP, Hallak et al. (2015)). Formally, we have $\mathcal{M} \triangleq \langle S, U, P_\theta, r_\theta, \gamma, \rho, \Theta, P_\Theta \rangle$, where Θ defines a space of context parameters, P_Θ is a fixed distribution over the contexts. Thus the transitions $P_\theta : S \times U \times S \times \Theta \to [0, 1]$, and the agent reward $r_\theta : S \times U \times \Theta \to [0, R_{max}]$ are now also functions of the context parameter θ. We now discuss the CMDP setting used in this work: we consider a parametrized context which defines a functional transformation of the underlying MDP state giving rise to context dependent observations. Formally, we have $\mathcal{M} \triangleq \langle S, U, P, r, \gamma, \rho, \Theta, P_\Theta, Z, f \rangle$, where Θ defines a space of context parameters, P_Θ is a fixed distribution over the contexts, Z is the set of observations emitted as $f : S \times \Theta \to Z$. Thus fixing a particular context θ gives us a richly observed MDP (Krishnamurthy et al., 2016; Mahajan, 2023) indexed by θ: $\mathcal{M}_\theta$. We assume that the agent observes θ in our setting. Fig. 1 illustrates the parametrized observation setting. Without loss of generality, we assume $S \subset [0, 1]^n$, $Z \subset [0, 1]^l$, where typically $n << l$. We will use $f(s, \theta), f_\theta(s)$ interchangeably to highlight the corresponding (un)-curried versions of the observation function. We will be focusing on functional forms for observations, but the setting can be extended to scenarios with added independent or correlated noise at each step with suitable assumptions about identifiability (Zhang et al., 2020).

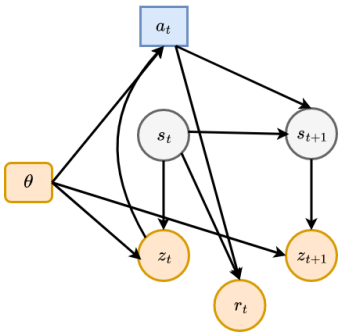


Figure 1: Parametrized observation setting.

Bisimulation: MDP Bisimulation defines a notion of state abstraction which groups states that are behaviorally equivalent (Li et al., 2006). Two states s_i and $s_j$are bisimilar if they both share the same immediate reward and equivalent distributions over the next bisimilar states for all possible actions (Larsen & Skou, 1989; Givan et al., 2003). Formally:

Definition 1 (Bisimulation Relations (Givan et al., 2003))**.** *Given an MDP $\mathcal{M}$, an equivalence relation B between states is a bisimulation relation if, for all states $s_i, s_j \in S$ that are equivalent*

under B (denoted $s_i \equiv_B s_j$) the following conditions hold:

$$r(s_i, a) = r(s_j, a) \quad \forall a \in U,$$
$$P(G|s_i, a) = P(G|s_j, a) \quad \forall a \in U, \quad \forall G \in S_B,$$

where S_B is the partition of S under the relation B (the set of all groups G of states equivalent under B), and $P(G|s, a) = \sum_{s' \in G} P(s'|s, a)$. (See Appendix A.1 for a primer on concepts related to equivalence relations)

Finding the coarsest bisimulation relation is known to be an NP-hard problem (Givan et al., 2003). Further, the exact partitioning induced from a bisimulation relation is generally impractical as it a very strict notion of equivalence and seldom leads to meaningful compression of the original MDP, this is especially true in continuous domains, where infinitesimal changes in the reward function or dynamics can break the bisimulation relation but still imply exploitable aggregation. Thus towards addressing this, Bisimulation Metrics (Ferns et al., 2011; Ferns & Precup, 2014; Castro, 2020; van Breugel & Worrell, 2001) relaxes the concept of exact bisimulation, and instead define a pseudometric space (S, d), where a distance function $d : S \times S \mapsto \mathbb{R}_{\geq 0}$ measures the behavioral similarity between two states. The bisimulation metric is formally defined as a convex combination of the reward difference added to the Wasserstein distance between transition distributions:

Definition 2 (Bisimulation Metric)**.** *From Theorem 2.6 in (Ferns et al., 2011) with $c \in [0, 1)$:*

$$d(s_i, s_j) = \max_{a \in U} (1 - c) \cdot |r^a_{s_i} - r^a_{s_j}| + c \cdot W_1(P^a_{s_i}, P^a_{s_j}; d).$$

W refers to the Wasserstein-p metric between two probability distributions P_i and P_j, defined as $W_p(P_i, P_j; d) = \inf_{\gamma' \in \Gamma(P_i, P_j)} [\int_{S \times S} d(s_i, s_j)^p \, \mathrm{d}\gamma'(s_i, s_j)]^{1/p}$, where $\Gamma(P_i, P_j)$ is the set of all couplings of P_i and P_j. The metric has intuitive interpretations depending on the exact value of p when viewed from the dual perspective, for example $W_1(P_i, P_j; d)$ denotes the cost of transporting mass from distribution P_i to another distribution P_j where the cost is given by the distance metric d (Villani, 2003). This is known as the earth-mover distance. The above definition can also be modified to include scenarios involving stochastic rewards, where a similar metric is chosen between reward distributions. To account for state similarities arising from following a particular policy, the π-bisimulation metric (Castro, 2020) is similarly defined by fixing a policy π and replacing the rewards and transitions used by their policy based expectations:

$$d^\pi(s_i, s_j) = (1 - c) \cdot |r^\pi_{s_i} - r^\pi_{s_j}| + c \cdot W_1(P^\pi_{s_i}, P^\pi_{s_j}; d^\pi). \tag{1}$$

In this work we will consider the max entropy RL framework as it ensures a unique optimal policy π^*_{merl}[‡]. Our goal is to leverage generalization and transfer obtained from informing the agent representation with the bisimulation similarity metric (Eq. (1)) under π^*.

3 Methodology

As previously discussed, it is important that agent policies in RL are robust to observation shifts for deployment in real world scenarios. In this work we wish to learn policies which can generalize well across the support set of the context distribution P_Θ. Our goal specifically would be to learn an effective representation function for the RL task set, $\phi : Z \times \Theta \mapsto Y$ which enables robust learning and deployment of autonomous agents to potentially unseen observation shifts (governed by a change in θ), see Fig. 2. Under suitable notions of invertibility (Assumption 1), the problem of generalizing across parameterized observation shifts (Section 2) lends itself

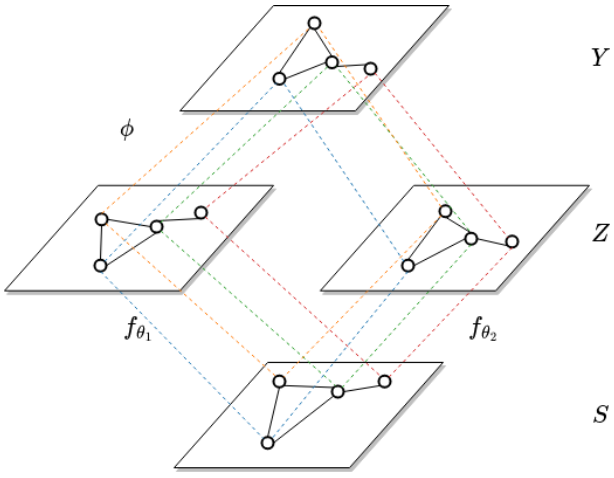


Figure 2: Learning representation invariant to observation shifts. Hollow circles represent states in the space, solid lines depict distances in the corresponding space, dashed lines depict equivalence across spaces tied by the colour.

[‡]We will refer to it as π^* in this work for brevity.

naturally to the notion of MDP isomorphism (Ravindran (2004), Mahajan & Tulabandhula (2017), see Definition 12 in Appendix A.2). This is because given two contexts θ_i, θ_j, there is always a one to one mapping between the observations in $\mathcal{M}_{\theta_i} \iff \mathcal{M}_{\theta_j}$ as directed by the underlying state, this is illustrated in Fig. 2. This inter-context correspondence helps us inform the representation more efficiently. Concretely, we specify the desiderata which the representation function ϕ must follow, as shown in Fig. 3:

- **Base Bisimulation (BB)**: Given a $\theta \in \Theta$, the representation should accurately preserve bisimulation distances between states, thus providing robustness to unimportant noise in observations. Concretely $\forall s_i, s_j \in \mathcal{S}$:

$$d(s_i, s_j) = d_Y(\phi(f_\theta(s_i), \theta), \phi(f_\theta(s_j), \theta)),$$

  where d_Y is a metric on Y (we use $Y = \mathbb{R}^m$ and L1 distance for our experiments).

- **Inter-context consistency (ICC)**: The representation should remain invariant under a fixed state as the context changes. Concretely: $\forall s \in \mathcal{S}$ and $\theta_1, \theta_2 \in \Theta$,

$$d_Y(\phi(f_{\theta_1}(s), \theta_1), \phi(f_{\theta_2}(s), \theta_2)) = 0.$$

- **Cross consistency (CC)**: This requires that the representation distance between two states are consistent across observation shifts:

$$d(s_i, s_j) = d_Y(\phi(f_{\theta_1}(s_i), \theta_1), \phi(f_{\theta_2}(s_j), \theta_2)),$$
$$d(s_i, s_j) = d_Y(\phi(f_{\theta_2}(s_i), \theta_2), \phi(f_{\theta_1}(s_j), \theta_1)).$$

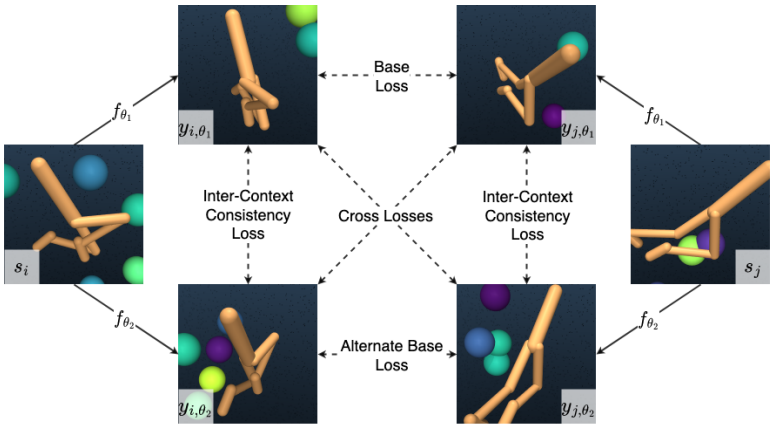


Figure 3: Various bisimulation losses corresponding to above desiderata. s represents underlying state, f_θ the observation function and y the corresponding observation, dashed lines represent bisimulation terms.

Fig. 3 depicts the above representation criteria for 2 different contexts(θ_1, θ_2) on the Mujoco control domain with 3D background objects acting as noise. Towards ensuring the above desiderata, we propose Robust Conditional Bisimulation (RCB) Algorithm 1, a data-efficient approach to learn control policies from unstructured, high-dimensional observations. As evident from Fig. 3, for n parallel simulation calls, our method captures $\binom{2n}{2} \sim O(n^2)$ interactions for representation learning using the above conditional bisimulation terms as opposed to $O(n)$ interactions in existing representation learning methods. For instance data augmentation methods based on contrastive learning (like Oord et al. (2018); Laskin et al. (2020a)) focus only on $(f_{\theta_1}(s), f_{\theta_2}(s))$ pairs whereas as plain bisimulation methods (like Zhang et al. (2021))

Algorithm 1 Robust Conditional Bisimulation (RCB)

for Time $t = 0$ to ∞ **do**
  Observe z_t, θ
  Encode observation $y_t = \phi(z_t, \theta)$
  Execute action $a_t \sim \pi(y_t)$
  Record data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{z_t, a_t, z_{t+1}, r_{t+1}\}$
  Sample batch $\mathcal{B} \sim \mathcal{D}$
  Train policy: $\mathbb{E}_{\mathcal{B}}[J^\pi]$
  Train encoder using pairwise loss: $\mathcal{L}_{rep}(\phi)$ {Eq. (2)}
  Train dynamics: $J(\hat{P},\phi) = (\hat{P}(\phi(z_t, \theta), a_t) - y_{t+1})^2$
end for

focus only on $(f_\theta(s_1), f_\theta(s_2))$ pairs. This order of magnitude increase in utilization of metric information in RCB allows for fast and efficient convergence to an observation invariant representation space.

We combine the above three representation conditions into a sum of squared loss components. For this we sample pairs of experiences i, j from the buffer along with base context θ_1 and an alternate context θ_2 both sampled from P_Θ at episode start. We next compute the embedding of the underlying states under the contexts and finally compute the representation loss term as follows:

$$\begin{aligned}\mathcal{L}_{rep}(\phi) =& \lambda_{base}\big[\big(|\overline{y}_{i,\theta_1} - \overline{y}_{j,\theta_1}|_1 - T_{i,j}\big)^2 + \big(|\overline{y}_{i,\theta_2} - \overline{y}_{j,\theta_2}|_1 - T_{i,j}\big)^2\big] + \\ & \lambda_{icc}\big[|\overline{y}_{i,\theta_1} - \overline{y}_{i,\theta_2}|_1^2 + |\overline{y}_{j,\theta_1} - \overline{y}_{j,\theta_2}|_1^2\big] + \\ & \lambda_{cc}\big[\big(|\overline{y}_{i,\theta_1} - \overline{y}_{j,\theta_2}|_1 - T_{i,j}\big)^2 + \big(|\overline{y}_{i,\theta_2} - \overline{y}_{j,\theta_1}|_1 - T_{i,j}\big)^2\big], \end{aligned} \tag{2}$$

where we use the following notation: $y_{i,\theta} = \phi(f(s_i, \theta), \theta)$ with $\overline{y}_{i,\theta}$ representing embeddings with stopped gradient and the target bisimulation distance $T_{i,j} = |r_i - r_j| + \gamma W_2(\hat{P}(\cdot|y_{i,\theta_1}, a_i), \hat{P}(\cdot|y_{j,\theta_1}, a_j))$. The relative weights for the three loss terms are given by hyperparameters $\lambda_{base}, \lambda_{icc}, \lambda_{cc}$ respectively. We use a setup similar to Zhang et al. (2021) where we use a permuted batch of $\mathcal{B}$ for pairwise representation loss computation in step-8 of Algorithm 1. Similarly we a probabilistic dynamics model $\hat{P}$ which outputs a Gaussian distribution. This allows for a simple to compute closed form W_2 metric which is used to replace the W_1 metric in the original formulation: $W_2(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\mu_j, \Sigma_j))^2 = ||\mu_i - \mu_j||_2^2 + ||\Sigma_i^{1/2} - \Sigma_j^{1/2}||_{\mathcal{F}}^2$, where $||\cdot||_{\mathcal{F}}$ is the Frobenius norm. Fig. 4 depicts the overall representation learning process. Finally, for the policy optimization part in step-7, we can use any max entropy policy gradient method. Access to simulator helps us translate a sampled batch from buffer into any randomly sampled contexts from which we can compute the various losses. However, in general this technique can also be extended to non-simulator settings like data augmentation (Laskin et al., 2020b), this could be specially promising as the latter approaches currently only minimize representation distance between two views of same input and not the bisimulation distance which is more aligned with solving the RL task.

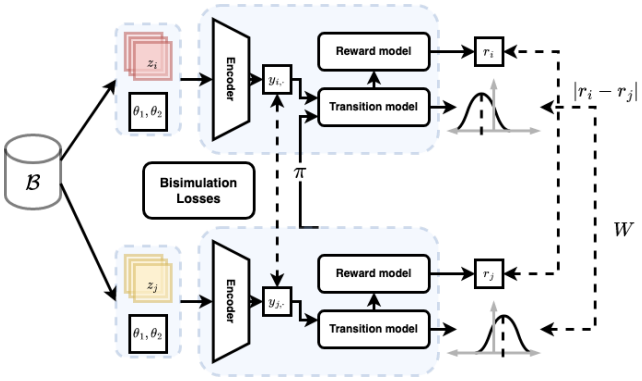


Figure 4: RCB Network architecture.

4 Analysis

We now discuss the important theoretical properties of our approach and study the generalization we can expect from learning representations under the conditional bisimuation framework. Proofs for the results can be found in Appendix A. The first result demonstrates the convergence of the π^*-bisimulation metric Eq. (1) on the joint input space $H \triangleq Z \times \Theta$ (We use the notation $h \triangleq (z, \theta)$ for a tuple in this space). We also overload the notion of policy(π) to implicitly contain ϕ so that it can be viewed as operating on the joint space.

Theorem 1. *Let* $\mathfrak{met}$ *be the space of bounded pseudometrics on* $Z \times \Theta$ *and* π *a policy that is continuously improving. Define* $\mathcal{F} : \mathfrak{met} \mapsto \mathfrak{met}$ *by*

$$\mathcal{F}(d, \pi)(h_i, h_j) = (1 - c)|r_{h_i}^\pi - r_{h_j}^\pi| + cW(d)(P_{h_i}^\pi, P_{h_j}^\pi)$$

Then, $\forall c \in (0, 1)$, $\mathcal{F}$ *has a least fixed point* $\tilde{d}$ *which is a* π^**-bisimulation metric.*

We next discuss an important assumption we need to make towards obtaining generalization results for the observation shifts.

Assumption 1 (Block structure)**.** *We assume that* $f_{\theta_1}(s_1) \cap f_{\theta_2}(s_2) \neq \emptyset \implies s_i = s_j, \forall \theta_1, \theta_2$ *so that the observation map is invertible.*

This means that the observation space Z can be partitioned into disjoint blocks, each containing the support for a particular value of $s \in S$ (Du et al., 2019). This also ensures that inverse observation map $f_\theta^{-1} : Z \to S$ exists. Relaxing Assumption 1 can break any guarantees obtainable on value function similarities arising from state similarity. This is because the same observation can get mapped to entirely different states in the latent MDP each with very different values, making the environment only partially observable. Note however that this requirement is not too restrictive, it is possible to consider added noise scenarios (both independent and correlated e.g. see (Zhang et al., 2020)) which maintain identifiability of the state. Finally, many real-world task observations tend to satisfy this assumption for high dimensional scenarios: e.g. visual projection of non-degenerate objects under different viewing angles.

We next discuss the implications of having learnt an representation ϕ which approximately preserves the π^*-bisimulation metric distances.

Theorem 2 (Aggregation value bound)**.** *Given an MDP $\hat{\mathcal{M}}$ constructed by aggregating tuples h of observation, context in an ϵ-neighborhood of the representation space such that $\delta \triangleq \max_{s,s',\theta_i,\theta_j} ||\phi(f_{\theta_i}(s),\theta_i) - \phi(f_{\theta_j}(s'),\theta_j)| - d_S(s,s')|$, where d_S is a π^*-bisimulation metric on S. Further let $\hat{\phi}$ denote the map from any h to these clusters, the optimal value functions for the two MDPs follow:*

$$|V^*(h) - \hat{V}^*(\hat{\phi}(h))| \leq \frac{2(\epsilon+\delta)}{(1-\gamma)(1-c)} \forall h \in Z \times \Theta$$

Note how the value estimate accuracy from aggregation is fundamentally bottle-necked by the representation learning error δ, this means that even the finest partitions (which use small ϵ) using ϕ will give value approximation only as good as the underlying representation.

We now state the Lipschitz continuity assumptions we use for further analysis. The first Assumption 2 concerns the change in observations z as the context θ changes. Several natural domains like visual projections satisfy this.

Assumption 2. *f is Lipschitz with coefficient L_θ^f with respect to (w.r.t.) θ.*

Next, we assume that the representation map ϕ and the policy π which conditions on the representations y are also Lipschitz w.r.t. the inputs. This can be enforced in practice for example for deep neural networks approximators (Virmaux & Scaman, 2018; Gouk et al., 2021).

Assumption 3. *ϕ is Lipschitz w.r.t. z and θ with coefficients L_z^ϕ, L_θ^ϕ respectively. Similarly, π is Lipschitz with coefficient L_y^π where the distance metric on the policy space is d_{TV} [†], the total variation metric on space of action distributions $\mathcal{P}(U)$.*

We now discuss the amount of generalization which we can expect when a policy assuming context θ_i is run on observation coming from the context θ_j. This can happen for example in scenarios when a shift in observations happens like change in the calibration settings of an autonomous vehicle's sensors. We introduce the notation $\pi_{\theta_i \leftarrow \theta_j}$ to represent the policy obtained from sampling action w.r.t. the restriction π_{θ_i} but using observation inputs from the context θ_j (ie. $\pi(a|\phi(f_{\theta_j}(s),\theta_i))$).

Theorem 3 (Generalization to unseen context)**.** *Under Assumption 2, Assumption 3 we have that for any two contexts θ_i, θ_j:*

$$|J^{\pi_{\theta_i}} - J^{\pi_{\theta_i \leftarrow \theta_j}}| \leq \frac{1}{1-\gamma} E_{\substack{s\sim f_\theta^{-1}\rho^{\pi_{\theta_i}}, \\ a\sim\pi_{\theta_i\leftarrow\theta_j}}} \left[A^{\pi_{\theta_i}}(s,a) + \frac{2\gamma A_{max}}{1-\gamma} L_\theta^f L_z^\phi L_y^\pi d_\Theta(\theta_i,\theta_j) \right]$$

where $A_{max} \triangleq \max_s |E_{a\sim\pi_{\theta_i\leftarrow\theta_j}}[A^{\pi_{\theta_i}}(s,a)]|$ and d_Θ is a metric on the context space.

Theorem 3 gives us the upper bound on the deviation of the expected returns when the agent expects an environment with context θ_i but is actually deployed in an environment with context θ_j.

We next discuss the important performance transfer scenarios when the simulator used for training a policy is not exact. These bounds are useful for situations where it is required to access tolerance of agent performance w.r.t. situations like sim to real deployment. Our first result addresses the setting where the simulator dynamics is not exact w.r.t. the real world, introducing errors ϵ_R, ϵ_P.

[†] note that L_y^π has the effect of squeezing inflations caused by L_θ^f and L_z^ϕ as d_{TV} is a bounded metric

Theorem 4 (Simulator fidelity bound)**.** *For an approximately correct simulator* $(\hat{r}, \hat{P})$ *such that* $\max_{s,a} |\hat{r}(s,a) - r(s,a)| \leq \epsilon_R$ *and* $\max_{s,a} d_{TV}(\hat{P}(s,a), P(s,a)) \leq \epsilon_P$ *we have for any policy* π*:*

$$|J^{\pi} - \hat{J}^{\pi}| \leq \frac{\epsilon_R}{(1-\gamma)} + \frac{\gamma \epsilon_P R_{max}}{(1-\gamma)^2}$$

Next, we consider the case where in addition to the latent transition and reward dynamics, the simulator emission function $\hat{f}$ is also approximate. Let $\epsilon_f \triangleq \max_{s,\theta} d_Y(\phi(\hat{f}_\theta(s)), \phi(f_\theta(s)))$. We are interested in the setting where the policy learns from an approximate simulator $(\hat{r}, \hat{P}, \hat{f})$ but the resultant learnt policy is deployed in the actual world (R, P, f). Note that this is a common practical setting as most simulators, even after knowing the actual underlying state, cannot completely capture the richness in the observations found in the real world. The below result relates the simulator policy performance $(\hat{f})$ to the one obtained by running the simulator policy on real observations (f).

Theorem 5 (Complete simulator fidelity bound)**.** *For an approximately correct simulator* $(\hat{r}, \hat{P}, \hat{f})$ *such that* $\max_{s,a} |\hat{r}(s,a) - r(s,a)| \leq \epsilon_R$*,* $\max_{s,a} d_{TV}(\hat{P}(s,a), P(s,a)) \leq \epsilon_P$ *and* $\epsilon_f \triangleq \max_{s,\theta} d_Y(\phi(\hat{f}_\theta(s)), \phi(f_\theta(s)))$*, we have for any policy* π*:*

$$|J^{\pi_{\hat{f}\leftarrow f}} - \hat{J}^{\pi_{\hat{f}}}| \leq \frac{\epsilon_R}{(1-\gamma)} + \frac{\gamma \epsilon_P R_{max}}{(1-\gamma)^2} + \frac{1}{1-\gamma} E_{\substack{s\sim \hat{f}^{-1}\rho^{\pi_{\hat{f}}}, \\ a\sim\pi_{\hat{f}\leftarrow f}}} \left[A^{\pi_{\hat{f}}}(s,a) + \frac{2\gamma A_{max}}{1-\gamma} L_y^{\pi} \epsilon_f \right].$$

$\pi_{\hat{f}\leftarrow f}$ represents the sampling of actions from $\pi_{\hat{f}}$ but using the observations obtained under the (real world) observation function f. Thus, the above two results (Theorems 4 and 5) are particularly useful for the realistic scenario where we have imprecise simulation dynamics.

5 Experiments

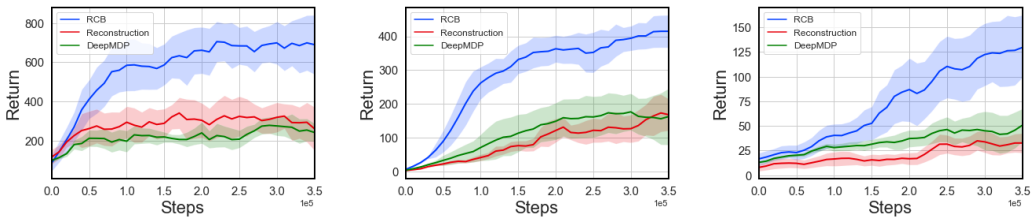
We perform experiments towards understanding whether our method Robust Conditional Bisimulation (RCB) helps learn representations which generalize better to observation shifts. Towards this, we use the DeepMind control suite (DMC, Tassa et al. (2018)) which uses Mujoco (Todorov et al., 2012) as the base simulator. We create new tasks for various agent morphologies where we learn to control the agent using image based input. Further, we also modify the simulator to have 3D spheres randomly bouncing in the environment, which contribute towards noise (we call this Modified-DMC). Note that this noise setting is harder than the simple-distractor setting in Zhang et al. (2021) as the agent has to learn to model 3D noise across different visual perspectives (see Fig. 3). We use two baselines for comparison:

1. DeepMDP (Gelada et al., 2019) which uses reward and forward dynamics predictability for learning a latent representation space.
2. A reconstruction based agent which uses a reward model and an image reconstruction based emission model to inform the representation.

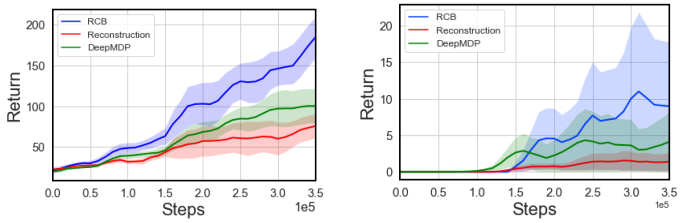
We use SAC (Haarnoja et al., 2018) as the base algorithm for optimizing the MERL objective in Algorithm 1. The architecture for common modules is kept similar across the methods. For fair comparison, we ensure that all the methods get equal access to the simulator experience and augment the representation learning objective for baselines with any extra simulator calls. Additional experimental setup details can be found in Appendix B.

Modified-DMC: For testing the ability to generalize across observation shifts, we use a uniform distribution over the range $P_\Theta = \mathcal{U}(-\pi/4, \pi/4)$ for the camera angle. At the beginning of each episode, we sample a camera angle context from P_Θ, the agents must adapt to changing image perspectives across training and evaluation. For evaluation, we use a fixed set of camera angles: $\{-\pi/4, -\pi/8, 0, \pi/8, \pi/4\}$ over which we compute the agent performance during the evaluation phase and report the average across the angles as the performance metric. Fig. 5 gives the evaluation performance plots for the agents on five different scenarios averaged over five seeds with one standard error shaded (our method RCB in blue, DeepMDP in green, Reconstruction in red). We see that RCB performs significantly better than the baseline agents on all the scenarios. RCB consistently achieves higher performance across the walker tasks (Figures 5(a) to 5(c)). We also note that the performance for Reconstruction worsens as the task becomes more dynamic, we hypothesize this is due to the lack of focus on the core features of the observation which influence the reward and dynamics. We

observe a similar trend on the cheetah domain (Fig. 5(d)) which is slightly easier than walker run. DeepMDP is often unable to perform satisfactorily in the training budget, we posit that this happens as it does not use any inter-context information to inform its representation. Thus, while it may learn close distance embeddings for a fixed context, the embeddings fare poorly across the contexts. RCB alleviates this problem by leveraging both the ICC and cross-consistency objectives in its formulation. We also note that generalization for the hopper domain (Fig. 5(e)) while doing pixel based control is especially hard given the environment stochasticity and the added background noise.



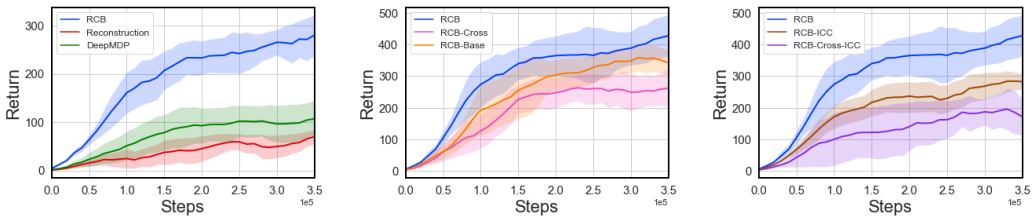
(a) Walker Stand (b) Walker Walk (c) Walker Run



(d) Cheetah Run (e) Hopper Hop

Figure 5: Empirical results on modified-DMC observation generalization tasks for different methods: RCB (our method), DeepMDP, and Reconstruction.

Out-of-distribution Generalization: To test the ability of the algorithms in dealing with unseen observation contexts during test time, we train on Modified-DMC where we use a uniform distribution over the range $P_\Theta = \mathcal{U}(-3\pi/16, 3\pi/16)$ for the camera angle and test on the unseen $\{-\pi/4, \pi/4\}$ angles. Fig. 6(a) gives the performance on the unseen angles for walker walk domain across the algorithms. Once again we note that RCB is able to better generalize to the unseen context due to its learning of a more accurate representation space using the inter-context objectives (**ICC** and **CC** terms in Section 3).



(a) Out-of-distribution (P_Θ) (b) Ablation 1 (c) Ablation 2

Figure 6: Out-of-distribution generalization and ablations

Ablations: To understand the effects of the different bisimulation loss components, we perform ablations removing each component. In Fig. 6(b) we remove the base (RCB-Base) and cross consistency (RCB-Cross) bisimulation terms. We see that removing the cross term has a bigger effect on performance. We believe this is because the cross-bisimulation term has a stronger anchoring effect as it also implicitly accounts for both the base and inter-context terms (see Fig. 3). Next, in Fig. 6(c) we remove the inter-context consistency term (RCB-ICC) and both the inter-context consistency and cross consistency terms (RCB-Cross-ICC). We notice a slight decrease in performance arising from dropping the inter context consistency loss. The RCB-Cross-ICC ablation is similar to DBC (Zhang et al., 2021) as it only contains base bisimulation losses (regular and alternate). We observe a significant decrease in performance in this latter ablation as we drop all the inter-context bisimulation terms helpful in generalization across the contexts. Thus it is important to ensure explicit alignment across the representations for the richly observed MDPs defined by different θ context when desiring good generalization across observation shifts.

6 Related Work

State abstraction in MDPs has been researched from various perspectives including notions which aggregate state based on policy, values, action-values and dynamics (Li et al., 2006). Bisimulation is the strictest form of abstraction based on MDP dynamics (Larsen & Skou, 1989; Givan et al., 2003). Bisimulation metrics were introduced to relax the notion of exact bisimulation for practical applicability (Ferns et al., 2011). Castro (2020) propose method for efficient computation of on policy variant of bisimulation metrics. DBC (Zhang et al., 2021) use bisimulation metrics to learn task relevant features which are robust to noise in the environment. They learn to tie together states distinguishable only by task irrelevant noise using bisimulation for learning a representation. We use the bisimulation framework to learn a representation which can *invert* the change in observation space caused by the varying context, and can be seen as abstracting across the group of isomorphic MDPs indexed by the context. We also provide the first generalization bounds for this setting with important practical applications like sim to real transfer. MDP homomorphism (Ravindran, 2004) is the principled framework of studying structural similarities across MDPs, this naturally extends the idea of state abstraction and opens the the way to leverage abstract similarities on a much broader scope. Taylor et al. (2009) propose lax probabilistic bisimulation using MDP homomorphisms. Mahajan & Tulabandhula (2017) use MDP isomorphism for learning symmetries for sample efficient reinforcement learning. (Mahajan et al., 2022) provide combinatorial generalization bounds for the contextual multi agent setting.

Representation learning for RL on high dimensional inputs has been studied using other methods in addition to bisimulation. Lange et al. (2012) use reconstruction of image inputs using auto-encoders for learning a latent control state. This was later extended to include modelling of the MDP dynamics (Watter et al., 2015; Hafner et al., 2019). Gelada et al. (2019) use a latent dynamics model approach for control and show its relation to bisimulation. Data Augmentation methods like Laskin et al. (2020b) use various image transformations on agent observations for data efficient learning of policies for pixel based control. Laskin et al. (2020a) use random crops on image data to be used under a contrastive based framework for representation learning. Kostrikov et al. (2020) use random image translations for regularising reinforcement learning from images by using multiple shifts to robustly estimate value function loss and targets. Oord et al. (2018); Chen et al. (2020) use self-supervised contrastive approach to learn representations by enforcing similarity constraints between data points.

Robust RL considers rewards maximization under adversarially varying dynamics for the environment. Pinto et al. (2017) use a two agent zero-sum game to model adversarial noise towards learning robust policies. Similarly, Stanton et al. inject noise in the state space and optimise for a minimax problem for robustness, and Tessler et al. (2019) study the robustness problem under action perturbations. Rather, we discuss the setting of adapting to potentially unseen deployment scenarios and provide theoretical guarantees for the policy transfer. (Team et al., 2021) use dynamic curricula for learning robust agent policies. Zhao et al. (2020) compile the various methods used in sim-to-real settings. Domain randomization, particularly used in robotic vision tasks including object localization (Tobin et al., 2017), object detection (Tremblay et al., 2018), pose estimation (Sundermeyer et al., 2018), and semantic segmentation (Yue et al., 2019), varies the training data from simulator across properties like textures, lighting, and camera positions. While domain randomization aims to provide enough simulated variability of the parameters at training time to ensure the model is able to generalize to potentially unseen settings during test, it is often insufficient for getting good results on control tasks due to unutilized information of task structure.

7 Conclusions & Future Work

In this we work we explored how bisimulation can be used to learn representation for RL towards generalization in complex high dimensional environment like visual inputs. We specially focused on learning policies invariant to observation shifts a problem which has several applications in the real world. Further, we analysed the theory of learning under the framework of conditional bisimualtion and proposed novel bounds characterizing state abstraction and generalization in this setting. Of particular importance were the results relating to performance guarantees across observation shifts when learning on a simulator. Finally, we evaluated our method on the modified DM-contol domain and showed its efficacy in comparison to the baseline approach. A current limitation of our theoretical analysis is that it requires invertability (Assumption 1), we aim to relax this in future work using notions of approximate invertability. Another limitation is that the algorithms used in experiments presently take the context vector as input, we aim to replace this with automatic context detection using unsupervised methods in future.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31. PMLR, 2017.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic Markov decision processes. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020.

Simon S. Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. Provably efficient RL with rich observations via latent state decoding. *Computing Research Repository (CoRR)*, abs/1901.09018, 2019. URL http://arxiv.org/abs/1901.09018.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 162–169, 2004. ISBN 0-9749039-0-6. URL http://dl.acm.org/citation.cfm?id=1036843.1036863.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous Markov decision processes. *Society for Industrial and Applied Mathematics*, 40(6):1662–1714, December 2011. ISSN 0097-5397. doi: 10.1137/10080484X. URL https://doi.org/10.1137/10080484X.

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

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *International Conference on Machine Learning (ICML)*, volume 97, pp. 2170–2179, Jun 2019.

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

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv:1801.01290*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, pp. 2555–2565. PMLR, 2019.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes, 2015.

Nan Jiang. Notes on tabular methods. 2018.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pp. 1840–1848, 2016.

Sascha Lange, Martin Riedmiller, and Arne Voigtländer. Autonomous reinforcement learning on raw visual input data in a real world application. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2012. doi: 10.1109/IJCNN.2012.6252823.

K. G. Larsen and A. Skou. Bisimulation through probabilistic testing (preliminary report). In *Symposium on Principles of Programming Languages*, pp. 344–352. Association for Computing Machinery, 1989. ISBN 0897912942. doi: 10.1145/75277.75307. URL https://doi.org/10.1145/75277.75307.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 5639–5650. PMLR, 2020a.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33: 19884–19895, 2020b.

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.

Anuj Mahajan. *Reinforcement learning in large state action spaces*. PhD thesis, University of Oxford, 2023.

Anuj Mahajan and Theja Tulabandhula. Symmetry detection and exploitation for function approximation in deep rl. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 1619–1621. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

Anuj Mahajan, Mikayel Samvelyan, Tarun Gupta, Benjamin Ellis, Mingfei Sun, Tim Rocktäschel, and Shimon Whiteson. Generalization in cooperative multi-agent systems. *arXiv preprint arXiv:2202.00104*, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826. PMLR, 2017.

Balaraman Ravindran. *An algebraic approach to abstraction in reinforcement learning*. University of Massachusetts Amherst, 2004.

Samuel Stanton, Rasool Fakoor, Jonas Mueller, Andrew Gordon Wilson, and Alex Smola. Robust reinforcement learning for shifting dynamics during deployment.

Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2011.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite. *arXiv:1801.00690 [cs]*, 2018. URL http://arxiv.org/abs/1801.00690.

Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate MDP homomorphisms. In *Neural Information Processing (NeurIPS)*, pp. 1649–1656, 2009.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents, 2021.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. *arXiv:1901.09184*, pp. 6215–6224, 2019.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*. IEEE, 2012.

Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic transition systems. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen (eds.), *Automata, Languages and Programming*, pp. 421–432. Springer, 2001. ISBN 978-3-540-48224-6. doi: 10.1007/3-540-48224-5_35.

Cédric Villani. *Topics in optimal transportation*. American Mathematical Society, 01 2003.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Neural Information Processing Systems (NeurIPS)*, pp. 2728–2736, 2015.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11214–11224. PMLR, 2020. URL http://proceedings.mlr.press/v119/zhang20t.html.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning Invariant Representations for Reinforcement Learning without Reconstruction. *arXiv:2006.10742 [cs, stat]*, 2021. URL http://arxiv.org/abs/2006.10742.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

A Additional Definitions and Proofs

A.1 Equivalence relations and classes

We first briefly mention some of the concepts from abstract algebra used in motivating state similarity in MDPs.

Definition 3. *A binary relation $\mathcal{R}$ on a set $\mathcal{S}$ is given by $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$*

Definition 4. *$\mathcal{R}$ is symmetric if $\mathcal{R}(a,b) \implies \mathcal{R}(b,a)$*

Definition 5. *$\mathcal{R}$ is reflexive if $\mathcal{R}(a,a), \forall a \in \mathcal{S}$*

Definition 6. *$\mathcal{R}$ is transitive if $\mathcal{R}(a,b) \wedge \mathcal{R}(b,c) \implies \mathcal{R}(a,c)$*

Definition 7. *$\mathcal{R}$ is equivalence if its reflexive, symmetric and transitive.*

Definition 8. *$\mathcal{P} \triangleq \{\mathcal{C}_i\}$ is a partition of a set $\mathcal{S}$ if $\mathcal{S} = \cup_i \mathcal{C}_i$ and $\mathcal{C}_i \cap \mathcal{C}_j$ is empty if $i \neq j$.*

Definition 9. *If $\mathcal{R}$ is an equivalence relation on $\mathcal{S}$, then $\mathcal{S}$ can be partitioned into equivalence classes with $\mathcal{P}(\mathcal{R}, \mathcal{S}) \triangleq \{\mathcal{C}_i\}$, where $\mathcal{C}_i \subseteq \mathcal{S}, \forall a, b \in \mathcal{C}_i \implies \mathcal{R}(a,b)$ and $\mathcal{C}_i \cap \mathcal{C}_j$ is empty if $i \neq j$.*

Definition 10. *For partitions $\mathcal{P}_1$ and $\mathcal{P}_2$, $\mathcal{P}_1$ is a filtrate of $\mathcal{P}_2$ if $\forall \mathcal{C}_i \in \mathcal{P}_2, \exists \mathcal{D}_j \in \mathcal{P}_1$ s.t. $\mathcal{C}_i = \cup_j \mathcal{D}_j$*

Definition 11. *$\mathcal{P}_c$ is the coarsest partition induced by $\mathcal{R}$ if $\forall$ valid partitions $\mathcal{P}$ under $\mathcal{R}$, $\mathcal{P}$ is a filtrate of $\mathcal{P}_c$*

A.2 Proofs

We first revisit the concept of MDP homomorphisms (Ravindran, 2004) which we will use for establishing important results concerning the conditional bisimulation framework.

Definition 12 (MDP homomorphism (Ravindran, 2004))**.** *Let $\Psi \subset S \times U$ is the set of admissible state-action pairs. MDP homomorphism $\mathcal{H}$ from $M = \langle S, U, \Psi, P, r, \gamma, \rho \rangle$ to $M' = \langle S', U', \Psi', P', r', \gamma, \rho' \rangle$ is defined as a surjection $\mathcal{H} : \Psi \to \Psi'$, which is itself defined by a tuple of surjections $\langle f, \{g_s, s \in S\} \rangle$. In particular, $\mathcal{H}((s,a)) := (f(s), g_s(a))$, with $f : S \to S'$ and $g_s : A_s \to A'_{f(s)}$, which satisfies two requirements: Firstly it preserves the reward function:*

$$r'(f(s), g_s(a)) = r(s,a)$$

and secondly it commutes with transition dynamics of M:

$$P'(f(s), g_s(a), f(s')) = P(s, a, [s']_{B_{\mathcal{H}|S}})$$

Here we use the notation $[\cdot]_{B_{\mathcal{H}|S}}$ to denote the *projection* of equivalence classes B that partition Ψ under the relation $\mathcal{H}((s,a)) = (s', a')$ on to S. Isomorphisms $\chi : \Psi \to \Psi'$ can then be formally defined as homomorphisms between M, M' that completely preserve the system dynamics with the underlying functions f, g_s being bijective.

Theorem 1. *Let $\mathfrak{met}$ be the space of bounded pseudometrics on $Z \times \Theta$ and π a policy that is continuously improving. Define $\mathcal{F} : \mathfrak{met} \mapsto \mathfrak{met}$ by*

$$\mathcal{F}(d, \pi)(h_i, h_j) = (1-c)|r^\pi_{h_i} - r^\pi_{h_j}| + cW(d)(P^\pi_{h_i}, P^\pi_{h_j})$$

Then, $\forall c \in (0,1)$, $\mathcal{F}$ has a least fixed point $\tilde{d}$ which is a π^*-bisimulation metric.

Proof. First, consider the super-MDP over the unified state space $H \triangleq Z \times \Theta$, $\mathcal{M}_{super} \triangleq \langle H, U, P_H, r_H, \gamma, \rho_H \rangle$, where the H subscripted distributions implicitly account for f, P_Θ, ρ. Similarly, let $\mathcal{M}_\theta$ be the MDP obtained by restricting the context to a particular value θ and $\mathcal{M}_{base} \triangleq \langle S, U, P, r, \gamma, \rho \rangle$. We have that under Assumption 1 $\mathcal{M}_\theta$ and $\mathcal{M}_{base}$ are isomorphic and all of $\mathcal{M}_{super}$, $\mathcal{M}_\theta$ and $\mathcal{M}_{base}$ are homomorphic (Ravindran, 2004; Mahajan & Tulabandhula, 2017). Thus we can map the policy dynamics in the super-MDP exactly to the base MDP with states S. We now directly apply metric convergence result of Theorem 1 in (Zhang et al., 2021) on the representation space Y, thus showing that the π bisimulation metric converges after repeated applications of the operator $\mathcal{F}$. □

Theorem 2 (Aggregation value bound)**.** *Given an MDP $\hat{\mathcal{M}}$ constructed by aggregating tuples h of observation, context in an ϵ-neighborhood of the representation space such that $\delta \triangleq \max_{s,s',\theta_i,\theta_j} ||\phi(f_{\theta_i}(s),\theta_i) - \phi(f_{\theta_j}(s'),\theta_j)| - d_S(s,s')|$, where d_S is a π^*-bisimulation metric on S. Further let $\hat{\phi}$ denote the map from any h to these clusters, the optimal value functions for the two MDPs follow:*

$$|V^*(h) - \hat{V}^*(\hat{\phi}(h))| \leq \frac{2(\epsilon+\delta)}{(1-\gamma)(1-c)} \forall h \in Z \times \Theta$$

Proof. We use a proof strategy similar to (Zhang et al., 2021). We have that every θ restriction of $\mathcal{M}_{super}$ is isomorphic to $\mathcal{M}_{base}$ from the above proof. By direct application of Theorem 5.2 in (Ferns et al., 2004) on the MDP $\mathcal{M}_{super}$ for any $h \in Z \times \Theta$:

$$(1-c)|V^*(h) - \hat{V}^*(\hat{\phi}(h))| \leq g(s,\tilde{d}) + \frac{\gamma}{1-\gamma} \max_{s' \in \mathcal{S}} g(s',\tilde{d})$$

where g is the average distance between a state and all other states in its equivalence class under the bisimulation metric $\tilde{d}$. Substituting g with the ϵ-neighborhood ball, and accounting for δ, the error of the representation w.r.t. the metric for each cluster gives us:

$$\begin{aligned}(1-c)|V^*(h) - \hat{V}^*(\hat{\phi}(h))| &\leq 2(\epsilon+\delta) + \frac{\gamma}{1-\gamma} 2(\epsilon+\delta) \\ |V^*(h) - \hat{V}^*(\hat{\phi}(h))| &\leq \frac{1}{1-c}\Big(2(\epsilon+\delta) + \frac{\gamma}{1-\gamma} 2(\epsilon+\delta)\Big) \\ &= \frac{2(\epsilon+\delta)}{(1-\gamma)(1-c)}.\end{aligned}$$

□

Lemma 1. *Let $f : X \to Y$, $g : Y \to Z$ be two functions with Lipschitz constants L_1 and L_2 respectively, then $g(f(\cdot))$ is Lipschitz with $L_1 \cdot L_2$*

Proof. Computing deviations for the various functions and using the definition of Lipschitzness, we have that:

$$\begin{aligned} df &\leq L_1 dx \\ dg &\leq L_2 dy = L_2 df \\ \implies dg &\leq L_2 L_1 dx \end{aligned}$$

Thus $g(f(\cdot))$ is Lipschitz with $L_1 \cdot L_2$ w.r.t. X. □

Theorem 3 (Generalization to unseen context)**.** *Under Assumption 2, Assumption 3 we have that for any two contexts θ_i, θ_j:*

$$|J^{\pi_{\theta_i}} - J^{\pi_{\theta_i \leftarrow \theta_j}}| \leq \frac{1}{1-\gamma} E_{\substack{s \sim f_\theta^{-1} \rho^{\pi_{\theta_i}}, \\ a \sim \pi_{\theta_i \leftarrow \theta_j}}} \Big[A^{\pi_{\theta_i}}(s,a) + \frac{2\gamma A_{max}}{1-\gamma} L_\theta^f L_z^\phi L_y^\pi d_\Theta(\theta_i,\theta_j)\Big]$$

where $A_{max} \triangleq \max_s |E_{a \sim \pi_{\theta_i \leftarrow \theta_j}}[A^{\pi_{\theta_i}}(s,a)]|$ and d_Θ is a metric on the context space.

Proof. Under Assumption 2, Assumption 3, we have that $d_{TV}(\pi_{\theta_i}(s), \pi_{\theta_j}(s)) \leq L_\theta^f L_o^\phi L_y^\pi d_\Theta(\theta_i,\theta_j)$ for all underlying states $s \in S, \theta_i, \theta_j \in \Theta$, by repeated application of Lemma 1. We next apply Corollary 1 from (Achiam et al., 2017) that uses the bound for performance difference as a function of policy TV distance giving us the result. □

Theorem 4 (Simulator fidelity bound)**.** *For an approximately correct simulator $(\hat{r}, \hat{P})$ such that $\max_{s,a} |\hat{r}(s,a) - r(s,a)| \leq \epsilon_R$ and $\max_{s,a} d_{TV}(\hat{P}(s,a), P(s,a)) \leq \epsilon_P$ we have for any policy π:*

$$|J^\pi - \hat{J}^\pi| \leq \frac{\epsilon_R}{(1-\gamma)} + \frac{\gamma \epsilon_P R_{max}}{(1-\gamma)^2}$$

Proof. We proceed similar to (Jiang, 2018) for proving the policy value bound. Let us consider the base MDP $\mathcal{M}_{base}$ as defined above. For any projected policy π here, the value function satisfies $\forall s \in S$:

$$
\begin{aligned}
&|\hat{V}^{\pi}(s) - V^{\pi}(s)| \\
&\leq |(\hat{r}(s,\pi) + \gamma\langle \hat{P}(s,\pi), \hat{V}^{\pi}\rangle) - (r(s,\pi) + \gamma\langle P(s,\pi), V^{\pi}\rangle)| \\
&\leq \epsilon_R + \gamma|\langle \hat{P}(s,\pi), \hat{V}^{\pi}\rangle - \langle P(s,\pi), V^{\pi}\rangle| \\
&\leq \epsilon_R + \gamma|\langle \hat{P}(s,\pi), \hat{V}^{\pi}\rangle - \langle P(s,\pi), \hat{V}^{\pi}\rangle + \langle P(s,\pi), \hat{V}^{\pi}\rangle - \langle P(s,\pi), V^{\pi}\rangle| \\
&\leq \epsilon_R + \gamma[|\langle \hat{P}(s,\pi) - P(s,\pi), \hat{V}^{\pi}\rangle| + |\hat{V}^{\pi} - V^{\pi}|_{\infty}] \\
&\leq \epsilon_R + \gamma[|\langle \hat{P}(s,\pi) - P(s,\pi), \hat{V}^{\pi} - \frac{R_{max}}{2(1-\gamma)}\mathbf{1}\rangle| + |\hat{V}^{\pi} - V^{\pi}|_{\infty}] \\
&\leq \epsilon_R + \gamma[|\hat{P}(s,\pi) - P(s,\pi)|_1 |\hat{V}^{\pi} - \frac{R_{max}}{2(1-\gamma)}\mathbf{1}|_{\infty} + |\hat{V}^{\pi} - V^{\pi}|_{\infty}] \\
&\leq \epsilon_R + \gamma[\frac{\epsilon_P R_{max}}{(1-\gamma)} + |\hat{V}^{\pi} - V^{\pi}|_{\infty}]
\end{aligned}
$$

Here we have viewed the probability transitions and values as vectors. The use of baseline $\frac{R_{max}}{2(1-\gamma)}$ helps tighten the bound by centering the values. We use the definition of TV in last step. As the bound for all $s \in S$ we get after rearranging:

$$|V^{\pi} - \hat{V}^{\pi}|_{\infty} \leq \frac{\epsilon_R}{1-\gamma} + \frac{\gamma\epsilon_P R_{max}}{(1-\gamma)^2}$$

Finally, as J^{π} is a convex combination of V^{π} w.r.t. ρ, we can use the above bound to prove the result. □

Theorem 5 (Complete simulator fidelity bound)**.** *For an approximately correct simulator $(\hat{r}, \hat{P}, \hat{f})$ such that $\max_{s,a} |\hat{r}(s,a) - r(s,a)| \leq \epsilon_R$, $\max_{s,a} d_{TV}(\hat{P}(s,a), P(s,a)) \leq \epsilon_P$ and $\epsilon_f \triangleq \max_{s,\theta} d_Y(\phi(\hat{f}_\theta(s)), \phi(f_\theta(s)))$, we have for any policy π:*

$$|J^{\pi_{\hat{f}\leftarrow f}} - \hat{J}^{\pi_{\hat{f}}}| \leq \frac{\epsilon_R}{(1-\gamma)} + \frac{\gamma\epsilon_P R_{max}}{(1-\gamma)^2} + \frac{1}{1-\gamma} E_{\substack{s\sim\hat{f}^{-1}\rho^{\pi_{\hat{f}}}, \\ a\sim\pi_{\hat{f}\leftarrow f}}} \Big[A^{\pi_{\hat{f}}}(s,a) + \frac{2\gamma A_{max}}{1-\gamma} L_y^{\pi}\epsilon_f\Big]$$

Proof. We consider an intermediate simulator $(\hat{r}, \hat{P}, f)$ which has the same reward and transition as the original simulator $(\hat{R}, \hat{P})$ but uses an exact observation function f (we use ˜ to represent quantities associated with this simulator). We can now decompose the difference bound as:

$$|J^{\pi_{\hat{f}\leftarrow f}} - \hat{J}^{\pi_{\hat{f}}}| \leq |J^{\pi_{\hat{f}\leftarrow f}} - \tilde{J}^{\pi_{\hat{f}\leftarrow f}}| + |\tilde{J}^{\pi_{\hat{f}\leftarrow f}} - \hat{J}^{\pi_{\hat{f}}}|$$

Next we have that the $d_{TV}(\pi_{\hat{f}\leftarrow f}, \pi_{\hat{f}}) \leq L_y^{\pi}\epsilon_f$. Reasoning similarly to Theorem 3 for the right term of RHS which gives an upper bound using the TV difference. Finally also applying Theorem 4 on the left term of RHS we get the theorem's result. □

B Additional experimental details

B.1 Architecture details

We use separate deep networks for actor, critic, transition and reward models. The encoder network for each used 32 filters and a 50 feature dimensions. The actor and critic models each used an MLP trunk of 4 layers and 1024 hidden dimensions on top of the encoder. The reward model used MLP trunk of 2 MLP layers and 512 hidden dimensions on top of the encoder. The transition model type used was a mixture of Gaussians of ensemble size 5. Each component in the transition ensemble uses a 2 MLP layers of 768 hidden dimensions on top of the encoder with the final layer bifurcating for a value for mean and standard deviation per feature dimension. Layer normalization was used for the reward and transition models. Target networks were used for value estimates and were updated every 4 epochs. Relu non-linearity was used for the networks. We exponentially anneal the representation loss with weight $(1.8 - 0.8 * 2^{\frac{\text{steps}}{\text{total steps}}})$. We use identical architectures for the overlapping components

of the baselines (Reconstruction and DeepMDP). The reconstruction agent uses an image decoder with an MLP followed by 2 deconvolution layers with the intermediate layer using 32 filters. Adam optimizer was used for training the parameters of the networks used. Grid search was used for tuning the hyperparameters. Our code is based on implementation by (Zhang et al., 2021) for their work. Each seed takes around 4 days to run on an Nvidia V100 GPU.

B.2 Hyper-parameters used: Conditional bisimulation

Table 1: Hyper-parameters used: Conditional bisimulation

PARAMETER	VALUE
λ_{base}	0.24
λ_{icc}	0.32
λ_{cc}	0.24
Initial steps	1000
Batch size	512
Action repeat	2
Encoder learning rate	10^{-3}
Encoder τ	$5 \cdot 10^{-3}$
Decoder learning rate	10^{-3}
Frames	1000
Actor learning rate	10^{-3}
Critic learning rate	10^{-3}
Critic τ	10^{-2}
α learning rate	10^{-4}
γ	0.99
Total Steps	$3.5 \cdot 10^{5}$
Temperature	0.1