Policy Gradient Methods in the Presence of Symmetries and State Abstractions

Prakash Panangaden*[1,2], Sahand Rezaei-Shoshtari*[1,2], Rosie Zhao*[1,3], David Meger[1,2], Doina Precup[1,2,4]

[1]McGill University, [2]Mila, [3]Harvard University, [4]DeepMind

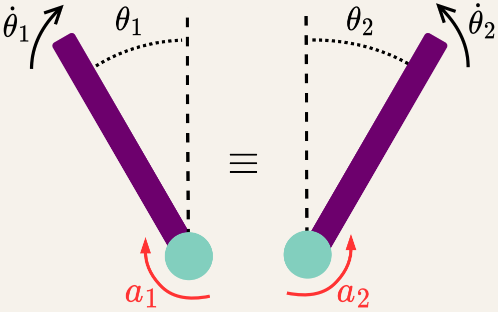
MSR RL Seminar - January 26, 2024

Motivating Abstraction in Reinforcement Learning

- How to capture state abstractions for an arbitrary environment?

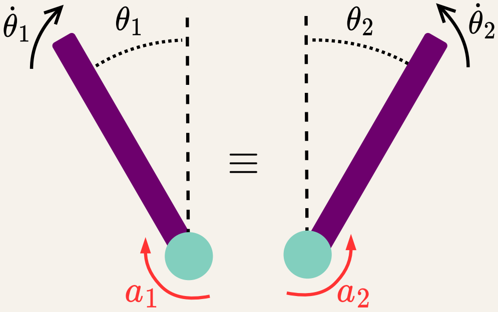
Motivating Abstraction in Reinforcement Learning

- How to capture state abstractions for an arbitrary environment?



Motivating Abstraction in Reinforcement Learning

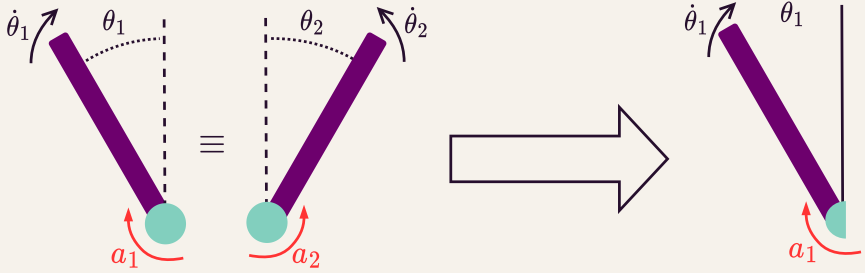
- How to capture state abstractions for an arbitrary environment?



- Equivalence relation on states: $(\theta_1, \dot{\theta}_1), (\theta_2, \dot{\theta}_2)$ are equivalent (**bisimulation relation**)

Motivating Abstraction in Reinforcement Learning

- Alternatively, define a new MDP with "equivalent" dynamics (**MDP homomorphism**)



Abstraction in Reinforcement Learning

- Some notions of abstraction for MDPs:
  - Bisimulation [Blute et al., 1997, Givan et al., 2003] and bisimulation metrics [Desharnais et al., 1999, Ferns et al., 2005, 2011].
  - Sampling-based similarity metrics [Castro et al., 2021].
  - Policy similarity metrics [Agarwal et al., 2020].

Abstraction in Reinforcement Learning

- Some notions of abstraction for MDPs:
  - Bisimulation [Blute et al., 1997, Givan et al., 2003] and bisimulation metrics [Desharnais et al., 1999, Ferns et al., 2005, 2011].
  - Sampling-based similarity metrics [Castro et al., 2021].
  - Policy similarity metrics [Agarwal et al., 2020].
- We focus on **MDP homomorphisms** [Ravindran and Barto, 2001, 2004]:
  - Theoretically defined on *finite* MDPs.
  - In practice, applied to *continuous states* but *discrete actions* [van der Pol et al., 2020a,b, Biza and Platt, 2019].

Key Questions

Key Questions

- How can we **learn** an **approximate** state abstraction without making assumptions about our environment apriori?
- How do we **design algorithms** which leverage a learned abstraction to improve sample efficiency and generalization?

Our Contributions

Our Contributions

1. Defined **continuous MDP homomorphisms** on continuous state and action spaces.

Our Contributions

1. Defined **continuous MDP homomorphisms** on continuous state and action spaces.
2. Proved that **value** and **optimal value** functions are preserved by continuous MDP homomorphisms.

Our Contributions

1. Defined **continuous MDP homomorphisms** on continuous state and action spaces.
2. Proved that **value** and **optimal value** functions are preserved by continuous MDP homomorphisms.
3. Derived the **Homomorphic Policy Gradient (HPG)** theorem.

Our Contributions

1. Defined **continuous MDP homomorphisms** on continuous state and action spaces.
2. Proved that **value** and **optimal value** functions are preserved by continuous MDP homomorphisms.
3. Derived the **Homomorphic Policy Gradient (HPG)** theorem.
4. Developed a deep actor-critic algorithm for learning the optimal policy simultaneously with the MDP homomorphism map in challenging continuous control problems

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.
- The **Bisimulation metric** measures how far apart two **state** pairs are:

$$d_{\text{bisim}}(s_i, s_j) = \max_{a \in \mathcal{A}} c_r \left| R(s_i, a) - R(s_j, a) \right| + c_t K\big(\tau_a(\cdot|s_i), \tau_a(\cdot|s_j)\big)$$

K is the Kantorovich (Wasserstein) metric, measuring the distance between the two transition probabilities.

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.
- The **Bisimulation metric** measures how far apart two **state** pairs are:

$$d_{\text{bisim}}\left(s_i, s_j\right) = \max_{a \in \mathcal{A}} c_r \left| R(s_i, a) - R(s_j, a) \right| + c_t K\left(\tau_a(\cdot|s_i), \tau_a(\cdot|s_j)\right)$$

  K is the Kantorovich (Wasserstein) metric, measuring the distance between the two transition probabilities.
- **Lax bisimulation** relaxes the requirement on action matching. It is precisely the same relation as an MDP homomorphism [Taylor et al., 2008].

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.
- The **Bisimulation metric** measures how far apart two **state** pairs are:

$$d_{\text{bisim}}(s_i, s_j) = \max_{a \in \mathcal{A}} c_r \left| R(s_i, a) - R(s_j, a) \right| + c_t K\left(\tau_a(\cdot|s_i), \tau_a(\cdot|s_j)\right)$$

  K is the Kantorovich (Wasserstein) metric, measuring the distance between the two transition probabilities.
- **Lax bisimulation** relaxes the requirement on action matching. It is precisely the same relation as an MDP homomorphism [Taylor et al., 2008].
- The **Lax bisimulation metric** measures the lax bisimilarity of **state-action** pairs:

$$d_{\text{lax}}\left((s_i, a_i), (s_j, a_j)\right) = c_r \left| R(s_i, a_i) - R(s_j, a_j) \right| + c_t K\left(\tau_{a_i}(\cdot|s_i), \tau_{a_j}(\cdot|s_j)\right)$$

Background: MDP Homomorphisms

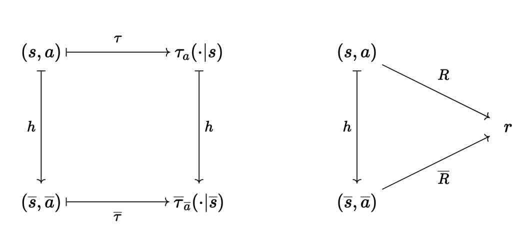
Definition (MDP Homomorphism)

An *MDP homomorphism* $h = (f, g_s) : \mathcal{M} \rightarrow \overline{\mathcal{M}}$ is a surjective map from a finite MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \tau_a, \gamma)$ onto an abstract finite MDP $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{R}, \overline{\tau}_{\overline{a}}, \gamma)$ where $f : \mathcal{S} \rightarrow \overline{\mathcal{S}}$ and $g_s : \mathcal{A} \rightarrow \overline{\mathcal{A}}$ satisfying the following commutative diagrams:

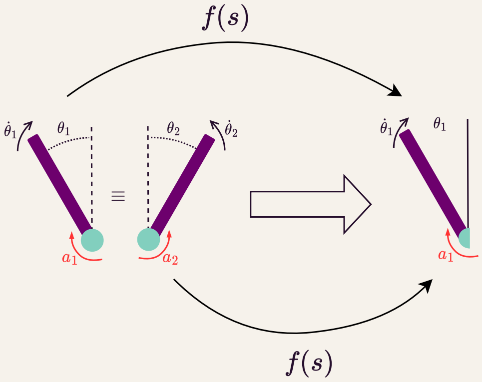
Background: MDP Homomorphisms

Definition (MDP Homomorphism)

An *MDP homomorphism* $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$ is a surjective map from a finite MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \tau_a, \gamma)$ onto an abstract finite MDP $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{R}, \overline{\tau}_{\overline{a}}, \gamma)$ where $f : \mathcal{S} \to \overline{\mathcal{S}}$ and $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ satisfying the following commutative diagrams:



Background: MDP Homomorphisms



$g_s(a) = a$ or $-a$, depending on s.

Background: MDP Homomorphisms

- The **optimal value equivalence** between $\mathcal{M}$ and $\overline{\mathcal{M}}$ [Ravindran and Barto, 2001]:

$$V^*(s) = \overline{V}^*(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^*(s,a) = \overline{Q}^*(f(s), g_s(a)) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

Background: MDP Homomorphisms

- The **optimal value equivalence** between $\mathcal{M}$ and $\overline{\mathcal{M}}$ [Ravindran and Barto, 2001]:

$$V^*(s) = \overline{V}^*(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^*(s,a) = \overline{Q}^*(f(s), g_s(a)) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

- **Policy lifting**: Given a policy $\overline{\pi}$ defined on $\overline{\mathcal{M}}$, we can define a policy $\pi^{\uparrow}$ on $\mathcal{M}$:

$$\pi^{\uparrow}(a|s) = \frac{\overline{\pi}(\overline{a}|f(s))}{|\{a \in g_s^{-1}(\overline{a})\}|}, \qquad \forall s \in \mathcal{S}, a \in g_s^{-1}(\overline{a})$$

$g_s^{-1}(\overline{a})$ is the pre-image of $\overline{a}$ under g_s.

Background: MDP Homomorphisms

- The **optimal value equivalence** between $\mathcal{M}$ and $\overline{\mathcal{M}}$ [Ravindran and Barto, 2001]:

$$V^*(s) = \overline{V}^*(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^*(s,a) = \overline{Q}^*(f(s), g_s(a)) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

- **Policy lifting**: Given a policy $\overline{\pi}$ defined on $\overline{\mathcal{M}}$, we can define a policy $\pi^{\uparrow}$ on $\mathcal{M}$:

$$\pi^{\uparrow}(a|s) = \frac{\overline{\pi}(\overline{a}|f(s))}{|\{a \in g_s^{-1}(\overline{a})\}|}, \qquad \forall s \in \mathcal{S}, a \in g_s^{-1}(\overline{a})$$

  $g_s^{-1}(\overline{a})$ is the pre-image of $\overline{a}$ under g_s.
- We can learn the optimal policy $\overline{\pi}^*$ in the abstract MDP $\overline{\mathcal{M}}$ and **lift** it to obtain the optimal policy in the actual MDP $\mathcal{M}$!

Value Equivalence Property

- **But,** for policy optimization we need to evaluate the policy!

Value Equivalence Property

- **But,** for policy optimization we need to evaluate the policy!

Theorem (Value Equivalence)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, then any two corresponding policies $\pi^{\uparrow} = \mathit{lift}(\overline{\pi})$ have equivalent values:

$$V^{\pi^{\uparrow}}(s) = V^{\overline{\pi}}(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^{\pi^{\uparrow}}(s, a) = Q^{\overline{\pi}}(f(s), g_s(a)) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

Value Equivalence Property

- **But,** for policy optimization we need to evaluate the policy!

Theorem (Value Equivalence)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, then any two corresponding policies $\pi^{\uparrow} = \text{lift}(\overline{\pi})$ have equivalent values:

$$V^{\pi^{\uparrow}}(s) = V^{\overline{\pi}}(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^{\pi^{\uparrow}}(s, a) = Q^{\overline{\pi}}(f(s), g_s(a)) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

- This enables the use of MDP homomorphisms for policy evaluation and policy optimization.

Continuous MDP Homomorphisms

Definition (Continuous MDP)

A *continuous Markov decision process (MDP)* is a 6-tuple:

$$\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \forall a \in \mathcal{A}\ \tau_a : \mathcal{S} \times \Sigma \to [0,1], R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, \gamma)$$

where Σ is a σ-algebra on $\mathcal{S}$.

Continuous MDP Homomorphisms

Definition (Continuous MDP)

A *continuous Markov decision process (MDP)* is a 6-tuple:

$$\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \forall a \in \mathcal{A}\ \tau_a : \mathcal{S} \times \Sigma \to [0,1], R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, \gamma)$$

where Σ is a σ-algebra on $\mathcal{S}$.

Definition (Continuous MDP Homomorphism)

A *continuous MDP homomorphism* is a map $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$ where $f : \mathcal{S} \to \overline{\mathcal{S}}$ and for every s in $\mathcal{S}$, $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ are measurable, surjective maps such that the following hold:

Continuous MDP Homomorphisms

Definition (Continuous MDP)

A *continuous Markov decision process (MDP)* is a 6-tuple:

$$\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \forall a \in \mathcal{A}\ \tau_a : \mathcal{S} \times \Sigma \rightarrow [0,1], R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}, \gamma)$$

where Σ is a σ-algebra on $\mathcal{S}$.

Definition (Continuous MDP Homomorphism)

A *continuous MDP homomorphism* is a map $h = (f, g_s) : \mathcal{M} \rightarrow \overline{\mathcal{M}}$ where $f : \mathcal{S} \rightarrow \overline{\mathcal{S}}$ and for every s in $\mathcal{S}$, $g_s : \mathcal{A} \rightarrow \overline{\mathcal{A}}$ are measurable, surjective maps such that the following hold:

$$\text{Invariance of reward: } \overline{R}(f(s), g_s(a)) = R(s, a) \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

Continuous MDP Homomorphisms

Definition (Continuous MDP)

A *continuous Markov decision process (MDP)* is a 6-tuple:

$$\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \forall a \in \mathcal{A}\ \tau_a : \mathcal{S} \times \Sigma \to [0,1], R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, \gamma)$$

where Σ is a σ-algebra on $\mathcal{S}$.

Definition (Continuous MDP Homomorphism)

A *continuous MDP homomorphism* is a map $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$ where $f : \mathcal{S} \to \overline{\mathcal{S}}$ and for every s in $\mathcal{S}$, $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ are measurable, surjective maps such that the following hold:

$$\text{Invariance of reward: } \overline{R}(f(s), g_s(a)) = R(s,a) \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

$$\text{Equivariance of transitions: } \overline{\tau}_{g_s(a)}(\overline{B}|f(s)) = \tau_a(f^{-1}(\overline{B})|s) \qquad \forall\ s \in \mathcal{S}, a \in \mathcal{A}, \overline{B} \in \overline{\Sigma}$$

Optimal Value Equivalence

Theorem (Optimal Value Equivalence)

If $h = (f, g_s) : \mathcal{M} \rightarrow \overline{\mathcal{M}}$, *then:*

$$V^*(s) = \overline{V}^*(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^*(s,a) = \overline{Q}^*(f(s), g_s(a)) \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

Optimal Value Equivalence

Theorem (Optimal Value Equivalence)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, *then:*

$$V^*(s) = \overline{V}^*(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^*(s,a) = \overline{Q}^*(f(s), g_s(a)) \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

Proof sketch: Induction on the sequence of optimal values. We also use the change of variable formula of the pushforward measure of $\tau_a(\cdot|s)$ with respect to f to change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$.

Policy Lifting with Continuous MDP Homomorphisms

- We need a policy lifting procedure, but this is harder to define in the continuous case.
- For now, let's make the following two simplifying assumptions:

Policy Lifting with Continuous MDP Homomorphisms

- We need a policy lifting procedure, but this is harder to define in the continuous case.
- For now, let's make the following two simplifying assumptions:

Assumption (Deterministic Policies)

We assume the policy is deterministic. The lifting becomes selecting one representative for the preimage $g_s^{-1}\left(\overline{\pi}(f(s))\right)$.

Policy Lifting with Continuous MDP Homomorphisms

- We need a policy lifting procedure, but this is harder to define in the continuous case.
- For now, let's make the following two simplifying assumptions:

Assumption (Deterministic Policies)

We assume the policy is deterministic. The lifting becomes selecting one representative for the preimage $g_s^{-1}\left(\overline{\pi}(f(s))\right)$.

Assumption (Bijective g_s)

We assume g_s is a bijection.

Policy Lifting with Continuous MDP Homomorphisms

- We need a policy lifting procedure, but this is harder to define in the continuous case.
- For now, let's make the following two simplifying assumptions:

Assumption (Deterministic Policies)

We assume the policy is deterministic. The lifting becomes selecting one representative for the preimage $g_s^{-1}\left(\overline{\pi}(f(s))\right)$.

Assumption (Bijective g_s)

We assume g_s is a bijection.

- Therefore, the lifted policy is uniquely defined as:

$$\pi^{\uparrow}(s) = g_s^{-1}\left(\overline{\pi}(f(s))\right)$$

Policy Lifting with Continuous MDP Homomorphisms

- How is policy lifting defined for **general** policies $\bar{\pi} : \bar{\mathcal{S}} \to \mathrm{Dist}(\bar{\mathcal{A}})$?

Policy Lifting with Continuous MDP Homomorphisms

- How is policy lifting defined for **general** policies $\bar{\pi} : \bar{\mathcal{S}} \to \mathrm{Dist}(\bar{\mathcal{A}})$?
- The lifted policy needs to satisfy $\pi^{\uparrow}(g_s^{-1}(\beta)|s) = \overline{\pi}(\beta|f(s))$ for every (Borel set) $\beta \subset \overline{\mathcal{A}}, s \in \mathcal{S}$.

Policy Lifting with Continuous MDP Homomorphisms

- How is policy lifting defined for **general** policies $\bar{\pi} : \bar{\mathcal{S}} \to \mathrm{Dist}(\bar{\mathcal{A}})$?
- The lifted policy needs to satisfy $\pi^{\uparrow}(g_s^{-1}(\beta)|s) = \overline{\pi}(\beta|f(s))$ for every (Borel set) $\beta \subset \overline{\mathcal{A}}, s \in \mathcal{S}$.
- Proved that $\pi^{\uparrow}$ exists, but the proof is non-constructive and the lifting procedure is computationally challenging.

Policy Lifting with Continuous MDP Homomorphisms

- How is policy lifting defined for **general** policies $\bar{\pi} : \bar{\mathcal{S}} \to \mathrm{Dist}(\bar{\mathcal{A}})$?
- The lifted policy needs to satisfy $\pi^{\uparrow}(g_s^{-1}(\beta)|s) = \overline{\pi}(\beta|f(s))$ for every (Borel set) $\beta \subset \overline{\mathcal{A}}, s \in \mathcal{S}$.
- Proved that $\pi^{\uparrow}$ exists, but the proof is non-constructive and the lifting procedure is computationally challenging.
- With the above definition, we obtain the **Value Equivalence** result in the continuous case!

Value Equivalence for General Policies

Theorem (Value Equivalence for General Policies)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$*, then for any policy* $\overline{\pi} : \overline{\mathcal{S}} \to \mathrm{Dist}(\overline{\mathcal{A}})$*, its lifted policy* $\pi^{\uparrow} : \mathcal{S} \to \mathrm{Dist}(\mathcal{A})$ *satisfies*

$$V^{\pi^{\uparrow}}(s) = V^{\overline{\pi}}(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^{\pi^{\uparrow}}(s, a) = Q^{\overline{\pi}}(f(s), g_s(a)) \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}$$

Value Equivalence for General Policies

Theorem (Value Equivalence for General Policies)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$*, then for any policy* $\overline{\pi} : \overline{\mathcal{S}} \to \mathrm{Dist}(\overline{\mathcal{A}})$*, its lifted policy* $\pi^{\uparrow} : \mathcal{S} \to \mathrm{Dist}(\mathcal{A})$ *satisfies*

$$V^{\pi^{\uparrow}}(s) = V^{\overline{\pi}}(f(s)) \quad \forall s \in \mathcal{S}, \qquad Q^{\pi^{\uparrow}}(s, a) = Q^{\overline{\pi}}(f(s), g_s(a)) \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}$$

Proof sketch: Induction on the sequence of value functions. We also use the change of variable formula of the pushforward measure of $\tau_a(\cdot|s)$ with respect to f to change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$, and the pushforward measure of $\pi^{\uparrow}(\cdot|s)$ with respect to g_s to change the integration space from $\mathcal{A}$ to $\overline{\mathcal{A}}$.

Reminder: Deterministic Policy Gradient (DPG)

- Performance measure: $J(\theta) = \mathbb{E}_{\pi}[V^{\pi}(s)]$.

Reminder: Deterministic Policy Gradient (DPG)

- Performance measure: $J(\theta) = \mathbb{E}_\pi[V^\pi(s)]$.
- Deterministic policy gradient (DPG) theorem [Silver et al., 2014]:

$$\nabla_\theta J(\pi_\theta) = \int_{s\in\mathcal{S}} \rho^{\pi_\theta}(s)\nabla_\theta \pi_\theta(s)\nabla_a Q^{\pi_\theta}(s,a)\big|_{a=\pi_\theta(s)} ds$$

where $\rho^{\pi_\theta}(s) = \lim_{t\to\infty} \gamma^t P(s_t = s | s_0, a_{0:t} \sim \pi_\theta)$ is the discounted stationary distribution of states under π_θ.

Reminder: Deterministic Policy Gradient (DPG)

- Performance measure: $J(\theta) = \mathbb{E}_\pi[V^\pi(s)]$.
- Deterministic policy gradient (DPG) theorem [Silver et al., 2014]:

$$\nabla_\theta J(\pi_\theta) = \int_{s \in \mathcal{S}} \rho^{\pi_\theta}(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)\Big|_{a=\pi_\theta(s)} ds$$

  where $\rho^{\pi_\theta}(s) = \lim_{t\to\infty} \gamma^t P(s_t = s | s_0, a_{0:t} \sim \pi_\theta)$ is the discounted stationary distribution of states under π_θ.
- Backbone of DDPG, TD3, DrQ-v2, etc.

Deterministic Homomorphic Policy Gradient (HPG)

- **Goal:** To derive a policy gradient estimator using samples obtained from the abstract MDP $\overline{\mathcal{M}}$.

Deterministic Homomorphic Policy Gradient (HPG)

- **Goal:** To derive a policy gradient estimator using samples obtained from the abstract MDP $\overline{\mathcal{M}}$.

Theorem (Equivalence of Deterministic Policy Gradients)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$*, and* $\pi_\theta^{\uparrow} : \mathcal{S} \to \mathcal{A}$ *is the lifted deterministic policy corresponding to the abstract deterministic policy* $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$*. Then:*

Deterministic Homomorphic Policy Gradient (HPG)

- **Goal:** To derive a policy gradient estimator using samples obtained from the abstract MDP $\overline{\mathcal{M}}$.

Theorem (Equivalence of Deterministic Policy Gradients)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, and $\pi_\theta^\uparrow : \mathcal{S} \to \mathcal{A}$ is the lifted deterministic policy corresponding to the abstract deterministic policy $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$. Then:

$$\nabla_a Q^{\pi_\theta^\uparrow}(s, a)\Big|_{a=\pi_\theta^\uparrow(s)} \nabla_\theta \pi_\theta^\uparrow(s) = \nabla_{\overline{a}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(\overline{s})} \nabla_\theta \overline{\pi}_\theta(\overline{s}).$$

Deterministic Homomorphic Policy Gradient (HPG)

- **Goal:** To derive a policy gradient estimator using samples obtained from the abstract MDP $\overline{\mathcal{M}}$.

Theorem (Equivalence of Deterministic Policy Gradients)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, and $\pi_\theta^{\uparrow} : \mathcal{S} \to \mathcal{A}$ is the lifted deterministic policy corresponding to the abstract deterministic policy $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$. Then:

$$\nabla_a Q^{\pi_\theta^{\uparrow}}(s, a)\Big|_{a=\pi_\theta^{\uparrow}(s)} \nabla_\theta \pi_\theta^{\uparrow}(s) = \nabla_{\overline{a}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(\overline{s})} \nabla_\theta \overline{\pi}_\theta(\overline{s}).$$

Proof sketch: We assume g_s is a bijection and use the chain rule and the inverse function theorem on manifolds.

Deterministic Homomorphic Policy Gradient (HPG)

- Can we just plug the previous result in DPG?

$$\nabla_\theta J(\pi_\theta) = \int_{s \in \mathcal{S}} \rho^{\pi_\theta}(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)\big|_{a=\pi_\theta(s)} ds$$

Deterministic Homomorphic Policy Gradient (HPG)

- Can we just plug the previous result in DPG?

$$\nabla_\theta J(\pi_\theta) = \int_{s \in \mathcal{S}} \rho^{\pi_\theta}(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)\big|_{a=\pi_\theta(s)} ds$$

- **No,** because the integration and stationary state distribution are still on $\mathcal{S}$!

Deterministic Homomorphic Policy Gradient (HPG)

Theorem (Deterministic Homomorphic Policy Gradient Theorem)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$*, and* $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$ *is a deterministic abstract policy defined on* $\overline{\mathcal{M}}$*. Then the gradient of the performance measure* $J(\theta)$*, defined on the actual MDP* $\mathcal{M}$*, w.r.t.* θ *is:*

$$\nabla_\theta J(\theta) = \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \nabla_{\overline{a}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \Big|_{\overline{a} = \overline{\pi}_\theta(\overline{s})} \nabla_\theta \overline{\pi}_\theta(\overline{s}) d\overline{s}.$$

where $\rho^{\overline{\pi}_\theta}(\overline{s})$ *is the discounted state distribution of* $\overline{\mathcal{M}}$ *following* $\overline{\pi}_\theta(\overline{s})$*.*

Deterministic Homomorphic Policy Gradient (HPG)

Theorem (Deterministic Homomorphic Policy Gradient Theorem)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, and $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$ is a deterministic abstract policy defined on $\overline{\mathcal{M}}$. Then the gradient of the performance measure $J(\theta)$, defined on the actual MDP $\mathcal{M}$, w.r.t. θ is:

$$\nabla_\theta J(\theta) = \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \nabla_{\overline{a}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \Big|_{\overline{a} = \overline{\pi}_\theta(\overline{s})} \nabla_\theta \overline{\pi}_\theta(\overline{s}) d\overline{s}.$$

where $\rho^{\overline{\pi}_\theta}(\overline{s})$ is the discounted state distribution of $\overline{\mathcal{M}}$ following $\overline{\pi}_\theta(\overline{s})$.

Proof sketch: We use the previous theorem and the change of variable formula of the push-forward measure of $\tau_a(\cdot|s)$ with respect to f to change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$.

Deterministic Homomorphic Policy Gradient (HPG)

Theorem (Deterministic Homomorphic Policy Gradient Theorem)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, and $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$ is a deterministic abstract policy defined on $\overline{\mathcal{M}}$. Then the gradient of the performance measure $J(\theta)$, defined on the actual MDP $\mathcal{M}$, w.r.t. θ is:

$$\nabla_\theta J(\theta) = \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \nabla_{\overline{a}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \Big|_{\overline{a} = \overline{\pi}_\theta(\overline{s})} \nabla_\theta \overline{\pi}_\theta(\overline{s}) d\overline{s}.$$

where $\rho^{\overline{\pi}_\theta}(\overline{s})$ is the discounted state distribution of $\overline{\mathcal{M}}$ following $\overline{\pi}_\theta(\overline{s})$.

Proof sketch: We use the previous theorem and the change of variable formula of the push-forward measure of $\tau_a(\cdot|s)$ with respect to f to change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$.

- ***We can use the Deterministic HPG of the abstract MDP as an additional gradient estimator for the actual MDP!***

Reminder: Stochastic Policy Gradient (PG)

- Performance measure: $J(\theta) = \mathbb{E}_{\pi}[V^{\pi}(s)]$.

Reminder: Stochastic Policy Gradient (PG)

- Performance measure: $J(\theta) = \mathbb{E}_\pi[V^\pi(s)]$.
- Stochastic policy gradient (PG) theorem [Sutton et al., 1999]:

$$\nabla_\theta J(\pi_\theta) = \int_{s\in\mathcal{S}} \rho^{\pi_\theta}(s) \int_{a\in\mathcal{A}} Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a|s) ds da$$

where $\rho^{\pi_\theta}(s) = \lim_{t\to\infty} \gamma^t P(s_t = s | s_0, a_{0:t} \sim \pi_\theta)$ is the discounted stationary distribution of states under π_θ.

Reminder: Stochastic Policy Gradient (PG)

- Performance measure: $J(\theta) = \mathbb{E}_\pi[V^\pi(s)]$.
- Stochastic policy gradient (PG) theorem [Sutton et al., 1999]:

$$\nabla_\theta J(\pi_\theta) = \int_{s \in \mathcal{S}} \rho^{\pi_\theta}(s) \int_{a \in \mathcal{A}} Q^{\pi_\theta}(s, a) \nabla_\theta \pi_\theta(a|s) ds da$$

where $\rho^{\pi_\theta}(s) = \lim_{t \to \infty} \gamma^t P(s_t = s | s_0, a_{0:t} \sim \pi_\theta)$ is the discounted stationary distribution of states under π_θ.
- Backbone of PPO, TRPO, SAC, etc.

Stochastic Homomorphic Policy Gradient (HPG)

Theorem (Stochastic Homomorphic Policy Gradient Theorem)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$*, and* $\overline{\pi} : \overline{\mathcal{S}} \to \mathrm{Dist}(\overline{\mathcal{A}})$ *is a stochastic abstract policy defined on* $\overline{\mathcal{M}}$*. Then the gradient of the performance measure* $J(\theta)$*, defined on the actual MDP* $\mathcal{M}$*, w.r.t.* θ *is:*

$$\nabla_\theta J(\theta) = \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \int_{\overline{a} \in \overline{\mathcal{A}}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \nabla_\theta \overline{\pi}_\theta(\overline{a} | \overline{s}) d\overline{s} d\overline{a}.$$

where $\rho^{\overline{\pi}_\theta}(\overline{s})$ *is the discounted state distribution of* $\overline{\mathcal{M}}$ *following* $\overline{\pi}_\theta(\overline{a}|\overline{s})$*.*

Stochastic Homomorphic Policy Gradient (HPG)

Theorem (Stochastic Homomorphic Policy Gradient Theorem)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, and $\overline{\pi} : \overline{\mathcal{S}} \to \mathrm{Dist}(\overline{\mathcal{A}})$ is a stochastic abstract policy defined on $\overline{\mathcal{M}}$. Then the gradient of the performance measure $J(\theta)$, defined on the actual MDP $\mathcal{M}$, w.r.t. θ is:

$$\nabla_\theta J(\theta) = \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \int_{\overline{a} \in \overline{\mathcal{A}}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \nabla_\theta \overline{\pi}_\theta(\overline{a}|\overline{s}) d\overline{s} d\overline{a}.$$

where $\rho^{\overline{\pi}_\theta}(\overline{s})$ is the discounted state distribution of $\overline{\mathcal{M}}$ following $\overline{\pi}_\theta(\overline{a}|\overline{s})$.

Proof sketch: We use the definition of the general policy lifting and the change of variable formula of the pushforward measure of $\tau_a(\cdot|s)$ with respect to f to change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$.

Stochastic Homomorphic Policy Gradient (HPG)

Theorem (Stochastic Homomorphic Policy Gradient Theorem)

If $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$, and $\overline{\pi} : \overline{\mathcal{S}} \to \mathrm{Dist}(\overline{\mathcal{A}})$ is a stochastic abstract policy defined on $\overline{\mathcal{M}}$. Then the gradient of the performance measure $J(\theta)$, defined on the actual MDP $\mathcal{M}$, w.r.t. θ is:

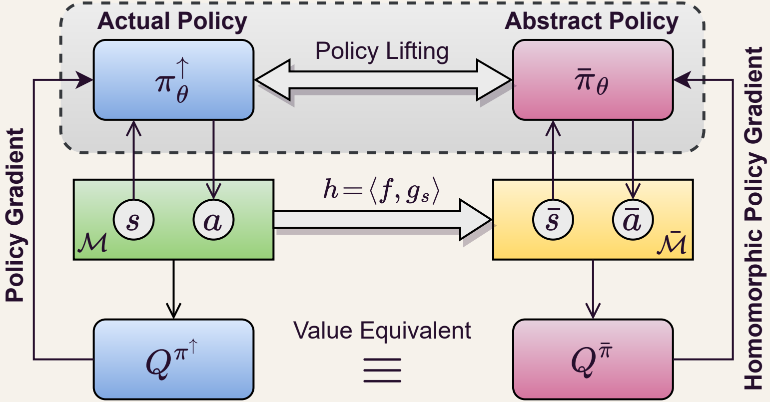
$$\nabla_\theta J(\theta) = \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \int_{\overline{a} \in \overline{\mathcal{A}}} Q^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \nabla_\theta \overline{\pi}_\theta(\overline{a}|\overline{s}) d\overline{s} d\overline{a}.$$

where $\rho^{\overline{\pi}_\theta}(\overline{s})$ is the discounted state distribution of $\overline{\mathcal{M}}$ following $\overline{\pi}_\theta(\overline{a}|\overline{s})$.

Proof sketch: We use the definition of the general policy lifting and the change of variable formula of the pushforward measure of $\tau_a(\cdot|s)$ with respect to f to change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$.

- ***We can use the Stochastic HPG of the abstract MDP as an additional gradient estimator for the actual MDP!***

Homomorphic Actor-Critic Algorithms



Homomorphic Actor-Critic Algorithms

- Deep Homomorphic Policy Gradient (DHPG)

Homomorphic Actor-Critic Algorithms

- Deep Homomorphic Policy Gradient (DHPG)
- The homomorphism map $h = (f, g_s)$, reward function $\overline{R}(\overline{s})$, and stochastic transition dynamics $\overline{\tau}(\cdot|\overline{s}, \overline{a})$ are parameterized by neural networks.

Homomorphic Actor-Critic Algorithms

- Deep Homomorphic Policy Gradient (DHPG)
- The homomorphism map $h = (f, g_s)$, reward function $\overline{R}(\overline{s})$, and stochastic transition dynamics $\overline{\tau}(\cdot|\overline{s}, \overline{a})$ are parameterized by neural networks.
- The actual policy $\pi^{\uparrow}(s)$ is parameterized and the abstract policy is obtained by the inverse of the policy lifting:

$$\overline{a} = g_s(\pi^{\uparrow}(s))$$

Homomorphic Actor-Critic Algorithms

- Deep Homomorphic Policy Gradient (DHPG)
- The homomorphism map $h = (f, g_s)$, reward function $\overline{R}(\overline{s})$, and stochastic transition dynamics $\overline{\tau}(\cdot|\overline{s}, \overline{a})$ are parameterized by neural networks.
- The actual policy $\pi^{\uparrow}(s)$ is parameterized and the abstract policy is obtained by the inverse of the policy lifting:

$$\overline{a} = g_s(\pi^{\uparrow}(s))$$

- Actual critic $Q^{\pi^{\uparrow}}(s, a)$ and abstract critic $\overline{Q}^{\overline{\pi}}(\overline{s}, \overline{a})$ are trained using TD error.

Homomorphic Actor-Critic Algorithms

- Deep Homomorphic Policy Gradient (DHPG)
- The homomorphism map $h = (f, g_s)$, reward function $\overline{R}(\overline{s})$, and stochastic transition dynamics $\overline{\tau}(\cdot|\overline{s}, \overline{a})$ are parameterized by neural networks.
- The actual policy $\pi^{\uparrow}(s)$ is parameterized and the abstract policy is obtained by the inverse of the policy lifting:

$$\overline{a} = g_s(\pi^{\uparrow}(s))$$

- Actual critic $Q^{\pi^{\uparrow}}(s, a)$ and abstract critic $\overline{Q}^{\overline{\pi}}(\overline{s}, \overline{a})$ are trained using TD error.
- Policy is updated by DPG and HPG:

$$\mathcal{L}_{\text{actor}}(\theta) \approx -\mathbb{E}_{s\sim\mathcal{B}}\Big[Q\big(s, \pi_\theta(s)\big) + \overline{Q}\big(f(s), g\big(s, \pi_\theta(s)\big)\big)\Big].$$

Policy Lifting for Stochastic Policies

Using the change of variable formula of the pushforward measure, we can show that the conditional expectations of abstract actions under the two policies are equal:

$$\mathbb{E}_{\pi^{\uparrow}}[g_s(a)|s] = \int_A g_s(a)\pi^{\uparrow}(da|s) = \int_{\bar{A}} \bar{a}\bar{\pi}(d\bar{a}|\bar{s}) = \mathbb{E}_{\bar{\pi}}[\bar{a}|f(s)],$$

Similarly,

$$\mathrm{Var}_{\pi^{\uparrow}}[g_s(a)|s] = \mathrm{Var}_{\bar{\pi}}[\bar{a}|f(s)]$$

Learning Continuous MDP Homomorphisms

- The **lax bisimulation metric** is used to encode lax bisimilar states closer together in the abstract space, similar to the bisimulation loss in DBC [Zhang et al., 2020]:

$$\mathcal{L}_{\text{lax}} = \mathbb{E}_{\mathcal{B}}\Big[\|f(s_i) - f(s_j)\|_1 - \|r_i - r_j\|_1 - \alpha W_2\big(\overline{\tau}(\cdot|f(s_i), g(s_i, a_i)), \overline{\tau}(\cdot|f(s_j), g(s_j, a_j))\big)\Big]$$

where W_2 is the Wasserstein-2 (Kantorovich) metric.

Learning Continuous MDP Homomorphisms

- The **lax bisimulation metric** is used to encode lax bisimilar states closer together in the abstract space, similar to the bisimulation loss in DBC [Zhang et al., 2020]:

$$\mathcal{L}_{\text{lax}} = \mathbb{E}_{\mathcal{B}}\left[\|f(s_i) - f(s_j)\|_1 - \|r_i - r_j\|_1 - \alpha W_2\left(\overline{\tau}(\cdot|f(s_i), g(s_i, a_i)), \overline{\tau}(\cdot|f(s_j), g(s_j, a_j))\right)\right]$$

  where W_2 is the Wasserstein-2 (Kantorovich) metric.

- Invariance of the reward and equivariance of the transition dynamics:

$$\mathcal{L}_{\text{h}} = \mathbb{E}_{(s_i, a_i, s_i', r_i)\sim\mathcal{B}}\left[\left(f(s_i') - \overline{s}_i'\right)^2 + \left(r_i - \overline{R}(f(s_i))\right)^2\right]$$

  where $\overline{s}_i' \sim \overline{\tau}(\cdot|f(s_i), g(s_i, a_i))$.

Learning Continuous MDP Homomorphisms

- The **lax bisimulation metric** is used to encode lax bisimilar states closer together in the abstract space, similar to the bisimulation loss in DBC [Zhang et al., 2020]:

$$\mathcal{L}_{\text{lax}} = \mathbb{E}_{\mathcal{B}}\left[\left\|f(s_i) - f(s_j)\right\|_1 - \|r_i - r_j\|_1 - \alpha W_2\left(\overline{\tau}(\cdot|f(s_i), g(s_i, a_i)), \overline{\tau}(\cdot|f(s_j), g(s_j, a_j))\right)\right]$$

  where W_2 is the Wasserstein-2 (Kantorovich) metric.

- Invariance of the reward and equivariance of the transition dynamics:

$$\mathcal{L}_{\text{h}} = \mathbb{E}_{(s_i, a_i, s_i', r_i)\sim\mathcal{B}}\left[\left(f(s_i') - \overline{s}_i'\right)^2 + \left(r_i - \overline{R}(f(s_i))\right)^2\right]$$

  where $\overline{s}_i' \sim \overline{\tau}(\cdot|f(s_i), g(s_i, a_i))$.

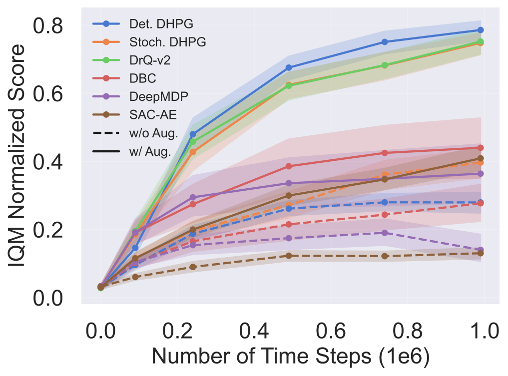
- The final loss for learning continuous MDP homomorphisms is $\mathcal{L}_{\text{lax}} + \mathcal{L}_{\text{h}}$.

Experimental Results

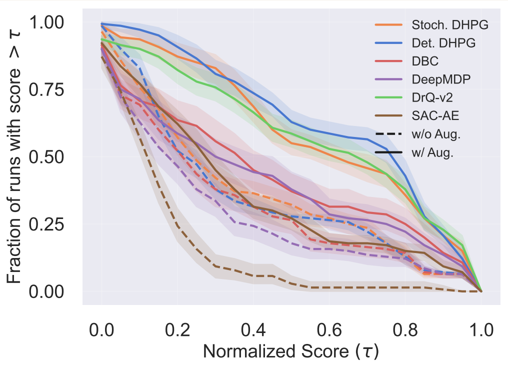
- DeepMind Control Suite, on state and pixel observations.
- We report *interquartile mean (IQM)* and *performance profiles* aggregated on all tasks over *10* random seeds [Agarwal et al., 2021].
- Baselines: DrQ-v2, DBC, DeepMDP, SAC-AE.
- All algorithms have two variations: *with* and *without image augmentation*.

Experimental Results: Performance

- Q: Does HPG improve policy optimization and representation learning?



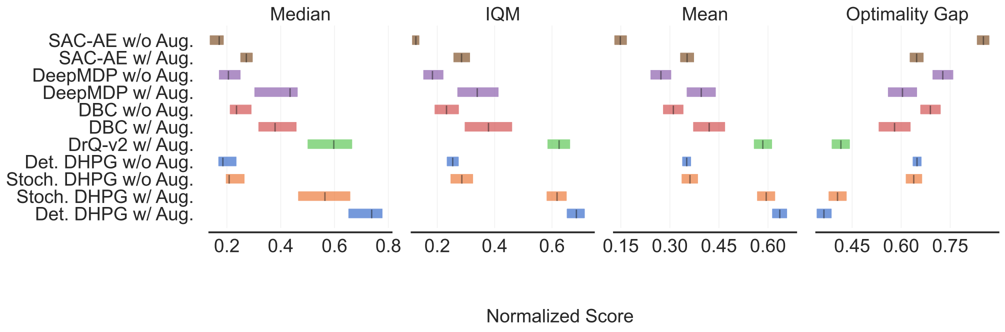
Sample efficiency.



Performance profiles at 500k step mark.

Experimental Results: Performance

- **Q: Does HPG improve policy optimization and representation learning?**



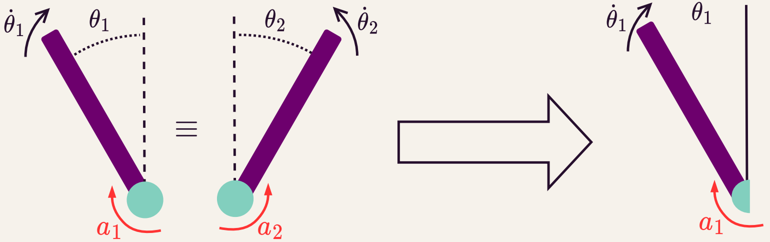
Aggregate metrics at 500k step mark.

Experimental Results: Qualitative Analysis

- **Q: What are the qualitative properties of the learned representations and abstract MDP?**
- Pendulum swingup as simple task with clear symmetries.

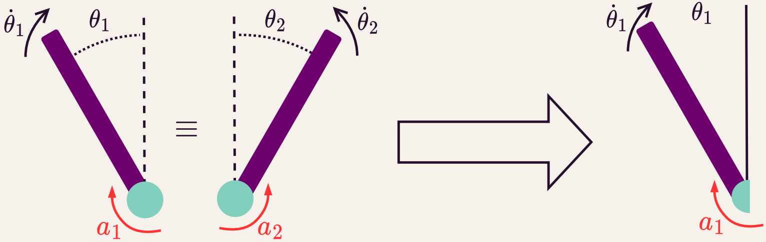
Experimental Results: Qualitative Analysis

- **Q: What are the qualitative properties of the learned representations and abstract MDP?**
- Pendulum swingup as simple task with clear symmetries.
- Two state-action pairs are equivalent under $\theta_1 = -\theta_2$, $\dot{\theta}_1 = -\dot{\theta}_2$, and $a_1 = -a_2$.



Experimental Results: Qualitative Analysis

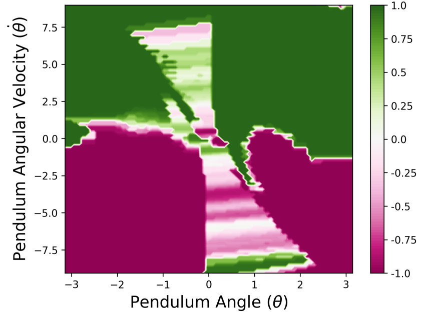
- **Q: What are the qualitative properties of the learned representations and abstract MDP?**
- Pendulum swingup as simple task with clear symmetries.
- Two state-action pairs are equivalent under $\theta_1 = -\theta_2$, $\dot{\theta}_1 = -\dot{\theta}_2$, and $a_1 = -a_2$.



- Therefore, abstract actions are expected to satisfy $g_{s_1}(a_1) = g_{s_2}(a_2)$ for equivalent state-action pairs.

Experimental Results: Qualitative Analysis

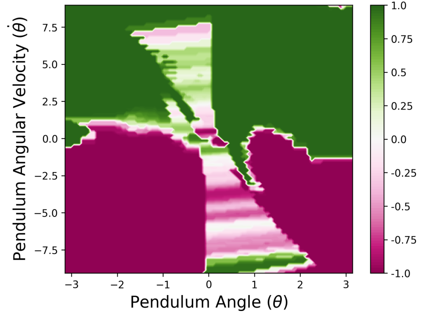
- What are the qualitative properties of the learned representations and abstract MDP?



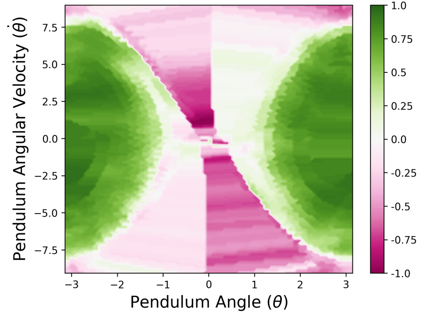
Actual optimal policy $a^* = \pi^{\uparrow *}(s)$

Experimental Results: Qualitative Analysis

- **What are the qualitative properties of the learned representations and abstract MDP?**



Actual optimal policy $a^* = \pi^{\uparrow^*}(s)$



Abstract optimal policy $\overline{a}^* = g_s(a^*) = \overline{\pi}^*(\overline{s})$

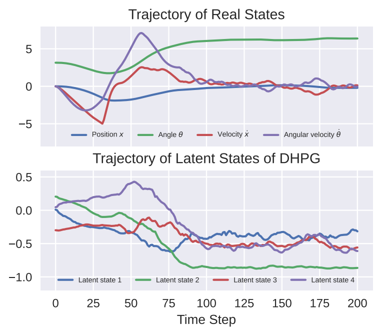
- The abstract optimal policy is symmetric and $g_{s_1}(a_1) = g_{s_2}(a_2)$ for equivalent state-action pairs.

Experimental Results: Recovering the Minimal MDP

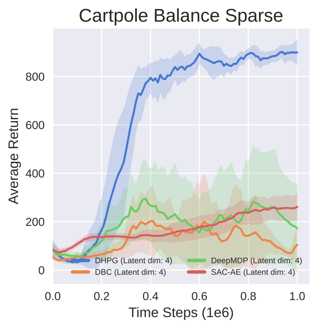
- **Q: Can DHPG learn and recover the minimal MDP image from raw pixel observations?**
- Theoretically, MDP homomorphisms can represent the minimal MDP image.
- We limit the latent space dimensions to the dimension of the real system.

Experimental Results: Recovering the Minimal MDP

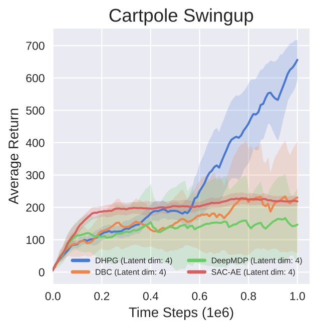
- **Q: Can DHPG learn and recover the minimal MDP image from raw pixel observations?**
- Theoretically, MDP homomorphisms can represent the minimal MDP image.
- We limit the latent space dimensions to the dimension of the real system.



Latent space trajectories.



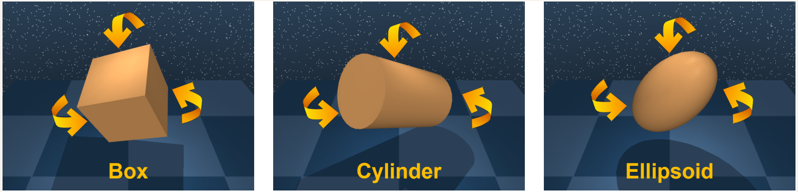
Learning curves.

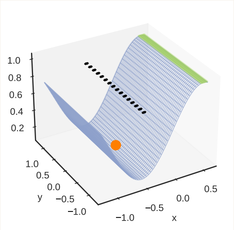


Learning curves.

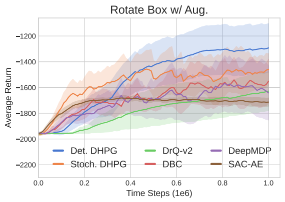
Additional Environments: Continuous Symmetries

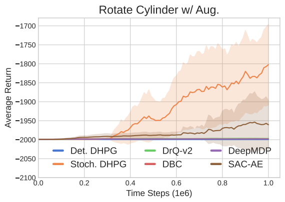
- **Rotate Suite**: reach a goal orientation by rotating the object
- **3D Mountain Car**: translational symmetry along y-axis

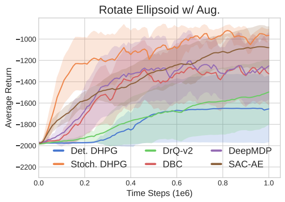




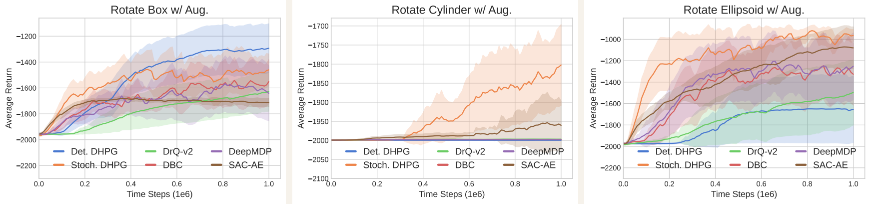
Additional Environments: Rotate Suite

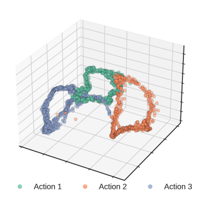




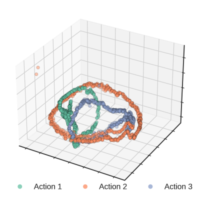


Additional Environments: Rotate Suite

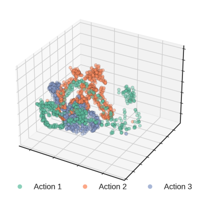




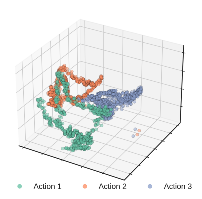
(a) Stoch. DHPG.



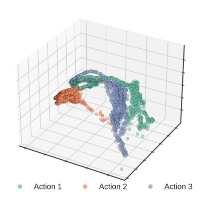
(b) Det. DHPG.



(c) DrQ-v2.

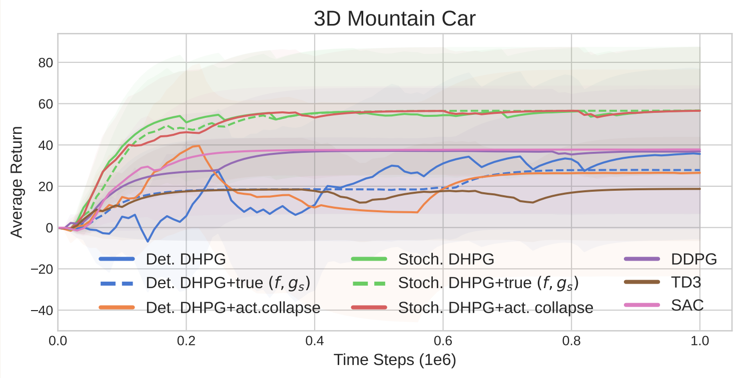


(d) DBC.



(e) DeepMDP.

Additional Environments: 3D Mountain Car



Conclusion

- Defined **continuous MDP homomorphisms** for state-action abstraction in continuous control problems.
- Derived the **homomorphic policy gradient** theorem.
- Demonstrated the potential of MDP homomorphisms in learning **structured representations** that can preserve values and represent the minimal MDP image.

Conclusion

- Defined **continuous MDP homomorphisms** for state-action abstraction in continuous control problems.
- Derived the **homomorphic policy gradient** theorem.
- Demonstrated the potential of MDP homomorphisms in learning **structured representations** that can preserve values and represent the minimal MDP image.
- Looking ahead: new methods for learning state abstractions in more complex domains
- Better theoretical guarantees (convergence rates?)

Thank You!

Extended Journal Paper!

Policy Lifting for Stochastic Policies

Using the change of variable formula of the pushforward measure, we can show that the conditional expectations of abstract actions under the two policies are equal:

$$\mathbb{E}_{\pi^\uparrow}[g_s(a)|s] = \int_A g_s(a)\pi^\uparrow(da|s) = \int_{\bar{A}} \bar{a}\bar{\pi}(d\bar{a}|\bar{s}) = \mathbb{E}_{\bar{\pi}}[\bar{a}|f(s)],$$

Similarly,

$$\text{Var}_{\pi^\uparrow}[g_s(a)|s] = \text{Var}_{\bar{\pi}}[\bar{a}|f(s)]$$

Background: MDP Homomorphisms

Definition (MDP Homomorphism)

An *MDP homomorphism* $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$ is a surjective map from a finite MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \tau_a, \gamma)$ onto an abstract finite MDP $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{R}, \overline{\tau}_{\overline{a}}, \gamma)$ where $f : \mathcal{S} \to \overline{\mathcal{S}}$ and $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ are surjective maps satisfying the following equations:

Background: MDP Homomorphisms

Definition (MDP Homomorphism)

An *MDP homomorphism* $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$ is a surjective map from a finite MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \tau_a, \gamma)$ onto an abstract finite MDP $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{R}, \overline{\tau}_{\overline{a}}, \gamma)$ where $f : \mathcal{S} \to \overline{\mathcal{S}}$ and $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ are surjective maps satisfying the following equations:

$$\text{Invariance of reward: } \overline{R}(f(s), g_s(a)) = R(s, a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

Background: MDP Homomorphisms

Definition (MDP Homomorphism)

An *MDP homomorphism* $h = (f, g_s) : \mathcal{M} \to \overline{\mathcal{M}}$ is a surjective map from a finite MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \tau_a, \gamma)$ onto an abstract finite MDP $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{R}, \overline{\tau}_{\overline{a}}, \gamma)$ where $f : \mathcal{S} \to \overline{\mathcal{S}}$ and $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ are surjective maps satisfying the following equations:

$$\text{Invariance of reward: } \overline{R}(f(s), g_s(a)) = R(s, a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$
$$\text{Equivariance of transitions: } \overline{\tau}_{g_s(a)}(f(s')|f(s)) = \sum_{s'' \in [s']_{B_h|\mathcal{S}}} \tau_a(s''|s) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

- B_h is the partition of $\mathcal{S}$ induced by the equivalence relation h.
- $B_h|\mathcal{S}$ is the projection of B_h onto $\mathcal{S}$.
- $[s']_{B_h|\mathcal{S}}$ denotes the block of $B_h|\mathcal{S}$ to which s' belongs.

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.
- The **Bisimulation metric** measures how far apart two **state** pairs are:

$$d_{\text{bisim}}(s_i, s_j) = \max_{a \in \mathcal{A}} c_r \left| R(s_i, a) - R(s_j, a) \right| + c_t K\left(\tau_a(\cdot|s_i), \tau_a(\cdot|s_j)\right)$$

K is the Kantorovich (Wasserstein) metric, measuring the distance between the two transition probabilities.

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.
- The **Bisimulation metric** measures how far apart two **state** pairs are:

$$d_{\text{bisim}}(s_i, s_j) = \max_{a \in \mathcal{A}} c_r \left| R(s_i, a) - R(s_j, a) \right| + c_t K\left(\tau_a(\cdot|s_i), \tau_a(\cdot|s_j)\right)$$

  K is the Kantorovich (Wasserstein) metric, measuring the distance between the two transition probabilities.
- **Lax bisimulation** relaxes the requirement on action matching. It is precisely the same relation as an MDP homomorphism [Taylor et al., 2008].

Background: Bisimulation and Lax Bisimulation

- **Bisimulation** captures indistinguishability of reward and transitions for **all** $a \in \mathcal{A}$.
- The **Bisimulation metric** measures how far apart two **state** pairs are:

$$d_{\text{bisim}}(s_i, s_j) = \max_{a \in \mathcal{A}} c_r \big| R(s_i, a) - R(s_j, a) \big| + c_t K\big(\tau_a(\cdot|s_i), \tau_a(\cdot|s_j)\big)$$

K is the Kantorovich (Wasserstein) metric, measuring the distance between the two transition probabilities.

- **Lax bisimulation** relaxes the requirement on action matching. It is precisely the same relation as an MDP homomorphism [Taylor et al., 2008].
- The **Lax bisimulation metric** measures the lax bisimilarity of **state-action** pairs:

$$d_{\text{lax}}\big((s_i, a_i), (s_j, a_j)\big) = c_r \big| R(s_i, a_i) - R(s_j, a_j) \big| + c_t K\big(\tau_{a_i}(\cdot|s_i), \tau_{a_j}(\cdot|s_j)\big)$$

Background: Surjection, Injection, and Bijection

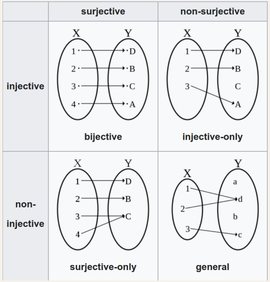


Figure: Image from Wikipedia.

Background: MDP Homomorphisms

An MDP Homomorphism h represented by Commutative Diagrams [Ravindran and Barto, 2001]:

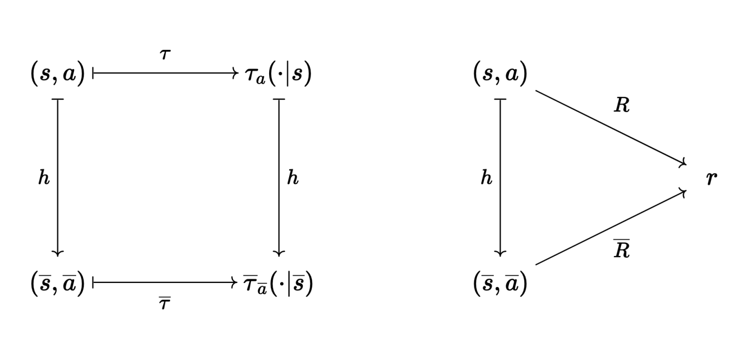


Figure: Image from Ravindran and Barto [2001].

Background: σ-Algebra

Definition (σ-algebra)

Given a set X, a σ-algebra on X is a family Σ of subsets of X such that 1) $X \in \Sigma$, 2) $A \in \Sigma$ implies $A^c \in \Sigma$ (closure under complements), and 3) if $(A_i)_{i\in\mathbb{N}}$ satisfies $A_i \in \Sigma$ for all $i \in \mathbb{N}$, then $\cup_{i\in\mathbb{N}} A_i \in \Sigma$ (closure under countable union). The tuple (X, Σ) is a measurable space.

The σ-algebra of a space specifies the sets in which a measure is defined.

Background: Pushforward Measure and Change of Variables

Definition (Pushforward measure)

Let (X_1, Σ_1) and (X_2, Σ_2) be two measurable spaces, $f : X_1 \to X_2$ a measurable map and $\mu : \Sigma_1 \to [0, \infty]$ a measure on X_1. Then the pushforward measure of μ with respect to f, denoted $f_*(\mu) : \Sigma_2 \to [0, \infty]$ is defined as:

$$(f_*(\mu))(B) = \mu(f^{-1}(B)) \; \forall \, B \in \Sigma_2.$$

Theorem (Change of variables)

A measurable function g on X_2 is integrable with respect to $f_*(\mu)$ if and only if the function $g \circ f$ is integrable with respect to μ, in which case the integrals are equal:

$$\int_{X_2} g d(f_*(\mu)) = \int_{X_1} g \circ f d\mu.$$

Policy Lifting for Stochastic Policies

Using the change of variable formula of the pushforward measure, we can show that the conditional expectations of abstract actions under the two policies are equal:

$$\mathbb{E}_{\pi^\uparrow}[g_s(a)|s] = \int_A g_s(a)\pi^\uparrow(da|s) = \int_{\bar{A}} \bar{a}\bar{\pi}(d\bar{a}|\bar{s}) = \mathbb{E}_{\bar{\pi}}[\bar{a}|f(s)],$$

Similarly,

$$\text{Var}_{\pi^\uparrow}[g_s(a)|s] = \text{Var}_{\bar{\pi}}[\bar{a}|f(s)]$$

References

Richard Blute, Josée Desharnais, Abbas Edalat, and Prakash Panangaden. Bisimulation for labelled markov processes. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 149–158. IEEE, 1997.

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

Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labeled markov systems. In *International Conference on Concurrency Theory*, pages 258–273. Springer, 1999.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for Markov decision processes with infinite state spaces. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 201–208, 2005.

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

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for Markov decision processes. *Advances in Neural Information Processing Systems*, 34, 2021.

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*, 2020.

Balaraman Ravindran and Andrew G Barto. Symmetries and model minimization in markov decision processes, 2001.

Balaraman Ravindran and Andrew G Barto. Approximate homomorphisms: A framework for non-exact minimization in Markov Decision Processes, 2004.

Elise van der Pol, Thomas Kipf, Frans A Oliehoek, and Max Welling. Plannable approximations to mdp homomorphisms: Equivariance under actions. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1431–1439, 2020a.

Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020b.

Ondrej Biza and Robert Platt. Online abstraction with mdp homomorphisms for deep learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.

Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate mdp homomorphisms. *Advances in Neural Information Processing Systems*, 21:1649–1656, 2008.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.