Policy Gradient Methods in the Presence of Symmetries and State Abstractions

Prakash Panangaden[*] PRAKASH@CS.MCGILL.CA
School of Computer Science, McGill University
and Mila – Quebec AI Institute
Montreal, QC, Canada

Sahand Rezaei-Shoshtari[*] SREZAEI@CIM.MCGILL.CA
School of Computer Science, McGill University
and Mila – Quebec AI Institute
Montreal, QC, Canada

Rosie Zhao[*] ROSIEZHAO@G.HARVARD.EDU
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA

David Meger DMEGER@CIM.MCGILL.CA
School of Computer Science, McGill University
and Mila – Quebec AI Institute
Montreal, QC, Canada

Doina Precup DPRECUP@CS.MCGILL.CA
School of Computer Science, McGill University
and Mila – Quebec AI Institute
and DeepMind
Montreal, QC, Canada

Abstract

Reinforcement learning (RL) on high-dimensional and complex problems relies on abstraction for improved efficiency and generalization. In this paper, we study abstraction in the continuous-control setting, and extend the definition of Markov decision process (MDP) homomorphisms to the setting of continuous state and action spaces. We derive a policy gradient theorem on the abstract MDP for both stochastic and deterministic policies. Our policy gradient results allow for leveraging approximate symmetries of the environment for policy optimization. Based on these theorems, we propose a family of actor-critic algorithms that are able to learn the policy and the MDP homomorphism map simultaneously, using the lax bisimulation metric. Finally, we introduce a series of environments with continuous symmetries to further demonstrate the ability of our algorithm for action abstraction in the presence of such symmetries. We demonstrate the effectiveness of our method on our environments, as well as on challenging visual control tasks from the DeepMind Control Suite. Our method's ability to utilize MDP homomorphisms for representation learning leads to improved performance, and the visualizations of the latent space clearly demonstrate the structure of the learned abstraction.

Keywords: reinforcement learning, policy optimization, abstraction, symmetry, representation learning

∗. Equal contributions; alphabetically ordered.

1. Introduction

Reinforcement learning on high-dimensional observations relies on representation learning and abstraction for learning a simpler problem that can be solved efficiently (Li et al., 2006; Abel et al., 2016). A major obstacle, however, is the coupling between states, actions, and rewards, particularly in complex continuous control problems. Strategies have been developed to find ways to reduce the state space by capturing *behavioral equivalence* between individual states. One formalization of this for MDPs is *bisimulation* (Givan et al., 2003), which was originally introduced for labelled transition systems in the early 1980's (Milner, 1989b). Bisimulation defines an equivalence relation over the state space, which allows one to quotient the state space by considering the equivalence classes under this relation. Bisimulation and their associated bisimulation metrics (Ferns et al., 2004)— which are used to approximate this equivalence relation — have previously been used for abstraction and model minimization.

Alternatively, one could use the quotiented state space to define a new environment with transition dynamics and rewards that preserve the structure of the original state space, and define a function between the original and new MDP. Thus, closely related to bisimulation are *MDP homomorphisms* (Ravindran, 2004; Ravindran and Barto, 2001, 2004), which capture behavioral equivalence via maps between MDPs that have certain preservation properties. Similar to bisimulation, one can use MDP homomorphisms to exploit (approximate) symmetries of an MDP for joint state-action abstraction.

MDP homomorphisms, developed in the context of discrete state and action spaces, are structure-preserving maps between MDPs that preserve value functions. Typically, they are used to map an MDP to an abstract MDP in a way such that no relevant information is lost. Ravindran and Barto (2001) show that policies can be pulled back, or *lifted*, from the abstract MDP to the original one while preserving optimality. Pulling back a policy in this way is a tricky construction and explicitly uses the finiteness of the state and action spaces. From the practical perspective, recent works have shown that using MDP homomorphisms are effective in guiding the learning in discrete problems (van der Pol et al., 2020a,b; Biza and Platt, 2019). Figure 1 shows schematics and key properties of MDP homomorphisms, which we formally define in Section 3.

Our first contribution is that we extend MDP homomorphisms to the continuous setting. This is crucial if we are to use these ideas for control of dynamical systems in physical spaces, as in robotics. The mathematics involved is significantly deeper than in the finite case and in some cases the finite case provides no guidance on how to proceed. We show that the value functions and the optimal value function are preserved for both stochastic and deterministic policies, as in the finite case. Lifting the policy from the abstract space to the original is one crucial example where we have to do something completely different from Ravindran and Barto (2001), where we appeal to using classical tools in functional analysis.

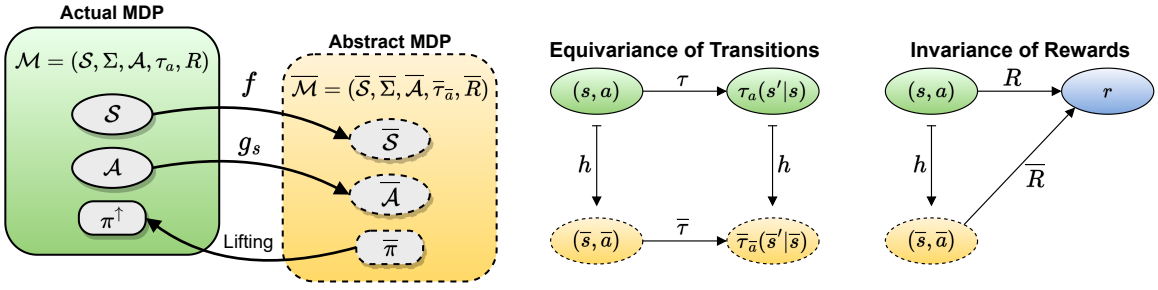
The next significant contribution is that we derive a version of the policy gradient theorem (Sutton et al., 2000; Silver et al., 2014) that tightly integrates the MDP homomorphism in the policy optimization process. In other words, one can use the policy gradient obtained from the abstract MDP, referred to as the *homomorphic policy gradient* (HPG), to optimize for the performance measure defined on the original MDP. We rigorously prove this result for both deterministic and stochastic policies, and show that HPG can act as an additional

gradient estimator capable of utilizing approximate symmetries for improved sample efficiency. As policy gradient methods remain a key family of RL algorithms, particularly for continuous control problems (Kiran et al., 2021; Arulkumaran et al., 2017), our homomorphic policy gradient derivation can have significant outcome for policy gradient algorithms in the presence of state abstraction.

Our third contribution is that we propose a deep actor-critic algorithm, referred to as the *deep homomorphic policy gradient* (DHPG) algorithm based on our novel theoretical results. DHPG is able to simultaneously optimize the policy, learn the homomorphism map, and exploit the abstraction of MDP homomorphisms for policy optimization. We empirically show that state-action abstractions learned through MDP homomorphisms provide a natural inductive bias for representation learning on challenging visual control problems, resulting in performance and sample efficiency improvements over strong baselines.

Finally, we show how to collapse an MDP when there is a group of symmetries which is also continuous. Thus, for example if a system is spherically symmetric the system is invariant under the action of the rotation group $SO(3)$ and this is certainly not a finite group. Discrete symmetries can and do occur in continuous systems but in general one will be dealing with continuous symmetries. Additionally, to demonstrate the ability of DHPG in learning continuous symmetries, we have developed a series of environments with continuous symmetries. In summary, our contributions can be listed as:

1. Defining continuous MDP homomorphisms on continuous state and action spaces, and proving the existence of the lifted policy in the general case of stochastic policies.
2. Proving the value and optimal value equivalence of MDP homomorphisms for the general case of stochastic policies.
3. Deriving the homomorphic policy gradient theorem for both stochastic and deterministic policies.
4. Developing a family of deep actor-critic algorithms that are able to learn the optimal policy simultaneously with the MDP homomorphisms map. Our algorithm works for both stochastic and deterministic policies through the use of a novel and computationally efficient policy lifting procedure.



(a) Components of an MDP homomorphism. (b) Commutative diagrams for MDP homomorphisms.

Figure 1: Overview of an MDP homomorphism $h = (f, g_s)$. **(a)** Components of an MDP homomorphism map, and the relation between the actual and abstract MDPs. **(b)** Commutative diagrams for MDP homomorphisms demonstrating the equivariance of transitions and the invariance of rewards. Diagram is adapted from Ravindran and Barto (2001).

5. Developing a series of novel RL environments with continuous symmetries that allow for benchmarking the ability of agents in learning and leveraging continuous environmental symmetries.

Notably, compared to the prior work of Rezaei-Shoshtari et al. (2022), our theoretical and empirical contributions are not limited to deterministic policies and bijective action encoders. Instead, we prove the value equivalence property and the homomorphic policy gradient theorem for the general case of stochastic policies and surjective action encoders, and propose a computationally efficient way for lifting a general stochastic policy. Empirically, we show that stochastic DHPG is superior to deterministic DHPG in environments with continuous symmetries as it is capable of a more powerful action abstraction. Our code for DHPG and the novel environments with continuous symmetries are publicly available[1].

The paper is structured as follows: in Sections 2 and 3 we provide an overview of related work and introduce relevant background, including finite MDP homomorphisms, bisimulation, and policy gradient methods. In Section 4 we formally introduce continuous MDPs and continuous MDP homomorphisms and prove key equivalence properties. In Section 5 we prove the stochastic and deterministic homomorphic policy gradient theorems and subsequently introduce the DHPG algorithm in Section 6. Finally, we provide experimental results of DHPG on continuous control tasks in Section 7.

2. Related Work

State Abstraction. Abstraction can be defined as a process that maps the original representation to an abstract representation that is more compact and easier to work with (Li et al., 2006). Probabilistic bisimulation, which we will refer to as just "bisimulation"(Larsen and Skou, 1991b) is one notion of behavioral equivalence between systems. It was extended to continuous state spaces by Blute et al. (1997b) and Desharnais et al. (2002) and extended to MDPs by Givan et al. (2003). Bisimulation metrics (Desharnais et al., 1999; Ferns et al., 2005b, 2006, 2011) define a pseudometric to quantify the degree of behavioural similarity. Recently, Zhang et al. (2020) defined a loss function for learning representations via bisimilarity of latent states, and Kemertas and Aumentado-Armstrong (2021) have further improved its robustness. Castro (2020) has proposed a method to approximate the bisimulation metric for deterministic MDPs with continuous states but discrete actions. van der Pol et al. (2020a) have defined a contrastive loss based on MDP homomorphisms for learning an abstract MDP for planning, but their method is only applicable to finite MDPs. Another approach is to directly embed the MDP homomorphic relation in the network architecture (van der Pol et al., 2020b, 2021). Other recently proposed metrics (Le Lan et al., 2021) seek to learn representations that preserve values (Grimm et al., 2020, 2021) or policies (Agarwal et al., 2020), or via a sampling-based similarity metric (Castro et al., 2021). Recently, Kemertas and Jepson (2022) have incorporated the bisimulation relation within the approximate policy iteration. Finally, state abstractions can in principle help improve transferring of policies (Abel et al., 2019; Castro and Precup, 2010; Soni and Singh, 2006; Sorg and Singh, 2009; Rajendran and Huber, 2009), or learning temporally extended actions (Castro and Precup, 2011; Wolfe and Barto, 2006a,b; Sutton et al., 1999).

1. https://github.com/sahandrez/homomorphic_policy_gradient

Action Abstraction. Action representations are often studied in the context of large discrete action spaces (Sallans and Hinton, 2004) as a form of a look-up embedding that is known *a-priori* (Dulac-Arnold et al., 2015), factored representations (Sharma et al., 2017), or policy decomposition (Chandak et al., 2019). Action representations can also be learned from expert demonstrations (Tennenholtz and Mannor, 2019). More related to our work is dynamics-aware embeddings (Whitney et al., 2019) where a combined state-action embedding for continuous control is learned. In contrast, we use the notion of homomorphisms to learn the state-dependent action representations, while preserving values. Lastly, action representations can be combined with temporal abstraction (Sutton et al., 1999) for discovering temporally extended actions (Ravindran and Barto, 2003; Abel et al., 2020; Castro and Precup, 2010, 2011).

State Representation Learning. Extant methods for learning the underlying state space from raw observations often use latent models (Gelada et al., 2019; Hafner et al., 2019a,b; Ha and Schmidhuber, 2018; Biza et al., 2021), auxiliary prediction tasks (Jaderberg et al., 2016; Liu et al., 2019; Lyle et al., 2021), physics-inspired inductive biases (Jonschkowski and Brock, 2015; Cranmer et al., 2020; Greydanus et al., 2019), unsupervised learning (Hjelm et al., 2018; Liu and Abbeel, 2021), or self-supervised learning (Anand et al., 2019; Sinha et al., 2021; Hansen et al., 2020; Hansen and Wang, 2021; Fan et al., 2021). From another point of view, representation learning can be effectively decoupled from the RL problem (Eslami et al., 2018; Stooke et al., 2021). Symmetries of the environment can also be used for representation learning (Mondal et al., 2022; Mahajan and Tulabandhula, 2017; Park et al., 2022; Wang et al., 2021; Higgins et al., 2018, 2021; Quessard et al., 2020; Caselles-Dupré et al., 2019). In fact, MDP homomorphisms are specializations of such approaches for RL. A key distinguishing factor of MDP homomorphisms is their ability to take actions into account for representation learning in the same premises as Thomas et al. (2017). Recently, simple image augmentation methods have shown significant improvements in RL performance (Yarats et al., 2020; Lee et al., 2019). Since these approaches are in general orthogonal to state abstractions, they can be combined together.

Equivariant Representation Learning. Using equivariance to leverage symmetries in data has been a fruitful line of machine learning research, where enforcing equivariance properties in the model architecture has led to state-of-the-art performance across several data modalities and applications. These domains include segmentation and classification tasks in computer vision (Cohen and Welling, 2016), medical imaging (Winkels and Cohen, 2019; Veeling et al., 2018), 3D model classification (Thomas et al., 2018; Chen et al., 2021), quantum chemistry (Qiao et al., 2021; Satorras et al., 2021; Batzner et al., 2022), and protein structure classification (Eismann et al., 2021; Ganea et al., 2021; Jumper et al., 2021). Since the utility of translation equivariance was demonstrated for traditional CNNs (LeCun et al., 1989, 1995), in recent years these convolutional layers have been generalized to be equivariant to discrete groups— such as finite rotations, translations, and reflections (Cohen and Welling, 2016)— and continuous groups— such as the rotation group SO(3) and the Euclidean and special Euclidean groups E(3) and SE(3) (Cohen et al., 2018; Kondor et al., 2018; Weiler et al., 2018; Cohen et al., 2019). The equivariance constraints imposed on these architectures are very rigid, and previous work has shown that true equivariance is difficult to achieve (Azulay and Weiss, 2018; Engstrom et al., 2017). Further, the groups

for these equivariant networks are typically fixed and known apriori; however, there has been work which constructs the appropriate equivariant network for arbitrary matrix Lie groups (Finzi et al., 2021) and presents algorithms to automatically discover symmetries pertaining to Lie groups (Dehmamy et al., 2021).

3. Background

3.1 Markov Decision Processes

Reinforcement learning is based on an agent interacting with its environment and acquiring rewards as it does so. It seeks to maximize the expected reward and learns to do this through its interaction with the environment. Markov decision processes are the basic model formalizing the interaction between an agent and its environment.

Definition 1 (MDP) *A* Markov decision process *(MDP) is a tuple* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \tau_a, R, \gamma)$ *where* $\mathcal{S}$ *is a set of states,* $\mathcal{A}$ *is a set of actions, for each* $a \in \mathcal{A}$ *we have* $\tau_a : \mathcal{S} \to \Delta(\mathcal{S})$ *where* $\Delta(\mathcal{S})$ *denotes the set of probability distributions over* $\mathcal{S}$*,* $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the reward function, and* $\gamma \in [0, 1)$ *is the discount factor.*

Initially, we assume $\mathcal{S}$ and $\mathcal{A}$ to be finite; in Section 4, we will define MDPs on more general state and action spaces. From a state $s \in \mathcal{S}$, an agent acting according to policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ selects actions $a \sim \pi(\cdot|s)$ and transitions to $s' \sim \tau_a(\cdot|s)$, yielding reward $r = R(s, a)$. The objective is to maximize the expected return by learning an optimal policy:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)].$$

Here, note that we assume $\gamma < 1$ to ensure convergence of the return (although $\gamma = 1$ is permitted for episodic tasks).

The *value function* $V^\pi(s)$ gives the expected return starting from state s and following policy π. The *action-value function* $Q^\pi(s, a)$ gives the expected return starting from state s, taking action a and thereafter following π. The value function is the fixed point of the Bellman operator $T^\pi : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ defined as:

$$T^\pi V(s) := \mathbb{E}_{\substack{a\sim\pi(\cdot|s) \\ s'\sim\tau_a(\cdot|s)}} [r + \gamma V(s')].$$

Similarly the optimal value function V^* is the fixed point of the Bellman optimality operator $T^* : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$:

$$T^* V(s) := \max_a \left[\mathbb{E}_{s'\sim\tau_a(\cdot|s)}[r + \gamma V(s')]\right].$$

Analogous Bellman equations are defined for Q^π and Q^* (Sutton and Barto, 2018).

3.2 Policy Gradient Theorems

Reinforcement learning algorithms can be broadly divided into *value-based* and *policy gradient* (PG) methods. While value-based methods select actions via a greedy maximization

step based on the learned action-values, policy gradient methods directly optimize a parameterized policy π_θ based on the gradient of the performance measure $J(\theta)$, defined as:

$$J(\theta) = \mathbb{E}_\pi[V^\pi(s)], \tag{1}$$

where the expectation is taken with respect to the policy, transitions, and the initial state distribution of the actual MDP. Unlike value-based methods, policy gradient algorithms inherit the strong, albeit local, convergence guarantees of the gradient descent and are naturally extendable to continuous actions. The fundamental theorem underlying policy gradient methods is the *policy gradient theorem* (Sutton et al., 2000):

Theorem 2 (Sutton et al. (2000)) *Let $\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A})$ be a stochastic policy defined on $\mathcal{M}$. Then the gradient of the performance measure $J(\theta)$ w.r.t. θ is:*

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

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.

In Theorem 2, $\rho^{\pi_\theta}(s)$ is assumed to exist and to be independent of the initial state distribution (ergodicity assumption). The significance of the policy gradient theorem is that the effect of policy changes on the state distribution does not appear in its expression, allowing for a sample-based estimate of the gradient (Williams, 1992). The deterministic policy gradient (DPG) is derived for deterministic policies by Silver et al. (2014) as:

Theorem 3 (Silver et al. (2014)) *Let $\pi_\theta : \mathcal{S} \to \mathcal{A}$ be a deterministic policy defined on $\mathcal{M}$. Then the gradient of the performance measure $J(\theta)$ w.r.t. θ is:*

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

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.

Since DPG does not need to integrate over the action space, it is often more sample-efficient than the stochastic policy gradient (Silver et al., 2014). However, noise needs to be manually injected during exploration as the deterministic policy does not have any inherent means of exploration. Finally, it is worth noting that due to the differentiation of the value function with respect to a, DPG is only applicable to continuous actions.

3.3 Bisimulation and Bisimulation Metrics

Bisimulation is a fundamental equivalence relation on the state space which captures the idea of behavioural similarity. It was introduced in the late 1970's and early 1980's by Milner (1980, 1989a) and Park (1981) in a non-probabilistic context and then extended to probabilistic systems by Larsen and Skou (1991a). The extension to continuous state spaces was done by Blute et al. (1997a) and Desharnais et al. (2002). These models did not involve rewards but it is a minor modification to add rewards as was done by Givan et al. (2003). The bisimulation relation on an MDP is formally defined as:

Definition 4 (Bisimulation) *A* bisimulation relation *on an MDP* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \tau_a, R, \gamma)$ *is an equivalence relation* B *on* $\mathcal{S}$ *such that if* sBt *holds then for any action* a *and any equivalence class of* C *of* B *we have:*

- $R(s,a) = R(t,a)$ *and*
- $\tau_a(C|s) = \tau_a(C|t)$.

If there exists such a relation between two states s *and* t *we say that* s *and* t *are bisimilar and write* $s \sim t$.

It is possible to define bisimulation as the greatest fixed point of a suitable operator on the complete lattice of equivalence relations on $\mathcal{S}$ (Milner, 1989b). Bisimulation is not robust to small perturbations in the system parameters. In a quantitative setting like MDPs we need to use metrics which give a quantitative notion to similarity.

In order to define a "metric" which can be viewed as a quantitative version of bisimulation, it is natural to think of a pseudometric with the property that its kernel is the bisimulation equivalence relation. This is defined through a fixed-point construction. We equip $\mathfrak{M}$, the space of 1-bounded pseudometrics on $\mathcal{S}$, with the following metric:

$$\Delta(m_1, m_2) := \sup_{x,y \in \mathcal{S}} |m_1(x,y) - m_2(x,y)|.$$

Here, m_1, m_2 are elements of $\mathfrak{M}$, *i.e.* 1-bounded pseudometrics. We then define an operator called $T_K : \mathfrak{M} \to \mathfrak{M}$ as follows:

$$T_K(m)(x,y) = \max_{a \in \mathcal{A}} [|R(x,a) - R(y,a)| + \gamma W_1(m)(\tau_a(x), \tau_a(y))].$$

Here $\tau_a(x)$ represents the probability distribution over the state space when the system executes an a-transition starting from x and similarly for $\tau_a(y)$. The metric W_1 on probability distributions is the well-known Kantorovich metric[2] which depends on m. One can readily show that the space $\mathfrak{M}$ equipped with Δ is a complete metric space and that the function or operator T_K is contractive with respect to the metric Δ. Thus, by the Banach fixed-point theorem, it has a unique fixed point. This is the bismulation metric[3].

3.4 Finite MDP Homomorphisms

Closely related to the concept of behavioural equivalence of states in MDPs are model minimization methods, which identify reductions in the original MDP to obtain an equivalent, smaller MDP. This gave rise to the notion of MDP homomorphism, originally proposed by Ravindran and Barto (2001). We will present the definitions and various results about MDP homomorphisms assuming the state and action spaces are finite.

Definition 5 (MDP Homomorphism) *An* MDP homomorphism h *between MDPs* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \tau_a, R, \gamma)$ *and* $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$ *is a tuple of surjective maps* $h = (f, g_s)$ *where* $f : \mathcal{S} \to \overline{\mathcal{S}}$ *and* $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ *for each* $s \in \mathcal{S}$ *such that:*

1. $R(s,a) = \overline{R}(f(s), g_s(a))$ *for every* $s \in \mathcal{S}, a \in \mathcal{A}$;

2. More often called the "Wasserstein" metric for reasons that have no historical validity.
3. In Ferns et al. (2005a) a different fixed-point theorem based on lattice theory was used.

2. *For every* $s, s' \in \mathcal{S}$ *and* $a \in \mathcal{A}$,

$$\overline{\tau}_{g_s(a)}(f(s')|f(s)) = \sum_{s'' \in [s']_{B_h|\mathcal{S}}} \tau_a(s''|s), \tag{2}$$

where B_h *is the partition of* $\mathcal{S} \times \mathcal{A}$ *induced by the equivalence relation of* h, $B_h \mid \mathcal{S}$ *is the projection of* B_h *onto* $\mathcal{S}$, *and* $[s']_{B_h|\mathcal{S}}$ *is the partition of* $B_h \mid \mathcal{S}$ *containing* s'.

In other words, the probability of transitioning between $f(s)$ and $f(s')$ in the image MDP $\overline{\mathcal{M}}$ under action $g_s(a)$ equals the probability of transitioning from s to the subset $[s']_{B_h|\mathcal{S}}$ in the original MDP $\mathcal{M}$ under action a. Figure 1b shows the commutative diagram of MDP homomorphisms. A key property of MDP homomorphisms is the *optimal value equivalence*, showing the optimal value function is preserved under this mapping.

Theorem 6 (Ravindran and Barto (2001)) *Let* h *be an MDP homomorphism from* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \tau_a, R, \gamma)$ *to* $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$. *Then for any* $(s, a) \in \mathcal{S} \times \mathcal{A}$, *we have:*

$$Q^*(s, a) = \overline{Q}^*(f(s), g_s(a)).$$

The optimal policies of an MDP and its image under an MDP homomorphism are also closely related. Given a policy on the image MDP, we can define a new, lifted policy on the original MDP that has the "equivalent behaviour".

Definition 7 (Lifted Policy) *Let* h *be an MDP homomorphism from* $M = (S, A, \tau_a, R, \gamma)$ *to* $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$, *and let* $\overline{\pi} : \overline{\mathcal{S}} \to \Delta(\overline{\mathcal{A}})$ *be a policy on* $\overline{\mathcal{M}}$. *Then* $\overline{\pi}$ *lifted to* $\mathcal{M}$ *is a policy* $\pi^{\uparrow} : \mathcal{S} \to \Delta(\mathcal{A})$ *such that for any* $(s, a) \in \mathcal{S} \times \mathcal{A}$, *we have:*

$$\pi^{\uparrow}(a|s) = \frac{\overline{\pi}(g_s(a)|f(s))}{|g_s^{-1}(\{g_s(a)\})|}.$$

Note that for these results to hold, it suffices for the lifted policy to satisfy:

$$\sum_{a \in g_s^{-1}(\{g_s(a)\})} \pi^{\uparrow}(a|s) = \overline{\pi}(g_s(a)|f(s)) \quad \forall\ s \in \mathcal{S} \tag{3}$$

but in order to make the lifted policy unique, Ravindran and Barto (2001) choose to uniformly spread the probability of taking $g_s(a)$ from $f(s)$ across all actions a' satisfying $g_s(a) = g_s(a')$. We have the following result that the lifted policy of the optimal policy of $\overline{\mathcal{M}}$ is an optimal policy for $\mathcal{M}$:

Theorem 8 (Ravindran and Barto (2001)) *Let* h *be an MDP homomorphism from* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \tau_a, R, \gamma)$ *to* $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$, *and let* $\overline{\pi}^* : \overline{\mathcal{S}} \to \Delta(\overline{\mathcal{A}})$ *be an optimal policy on* $\overline{\mathcal{M}}$. *Then the lifted policy* $\pi^* : \mathcal{S} \to \Delta(\mathcal{A})$ *is an optimal policy for* $\mathcal{M}$.

Furthermore, Rezaei-Shoshtari et al. (2022) show that given this definition of a lifted policy, we have a *value equivalence* result, showing that all value functions—not just the optimal one— are preserved under the MDP homomorphism mapping.

Theorem 9 (Rezaei-Shoshtari et al. (2022)) *Let* h *be an MDP homomorphism from* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \tau_a, R, \gamma)$ *to* $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$. *Then for any* $(s, a) \in \mathcal{S} \times \mathcal{A}$, *abstract policy* $\overline{\pi} : \overline{\mathcal{S}} \to \Delta(\overline{\mathcal{A}})$, *and its lifted policy* $\pi^{\uparrow} : \mathcal{S} \to \Delta(\mathcal{A})$, *we have:*

$$Q^{\pi^{\uparrow}}(s, a) = \overline{Q}^{\overline{\pi}}(f(s), g_s(a)).$$

4. Continuous MDP Homomorphisms

Our introduction of MDP homomorphisms in the previous section was strictly applicable where the state and action spaces were finite. In this section, we will formalize MDP homomorphisms for general continuous domains. First, we define continuous MDPs and state our underlying assumptions, which require care regarding measurability and differentiability of spaces.

Definition 10 (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 $\mathcal{S}$, the state space is assumed to be an appropriate topological space, Σ is a σ-algebra on $\mathcal{S}$[4]*, $\mathcal{A}$, the space of* actions*, is a locally compact metric space, usually taken to be a subset of $\mathbb{R}^n$, τ_a is the transition probability kernel for each possible action a, for each fixed s, $\tau_a(\cdot|s)$ is a probability distribution on Σ while R is the reward function, and γ is the discount factor. Furthermore, for all $s \in \mathcal{S}$ and $B \in \Sigma$ the map $a \mapsto \tau_a(B|s)$ is smooth.*

The last assumption is required for differentiability with respect to actions a, which is needed in Section 5 for deriving the homomorphic policy gradient theorem.

Probability theory on continuous spaces works well when the underlying space is assumed to be Polish (see Appendix B for definitions) but we do not need the properties of Polish spaces for our results. The assumption on the action space is needed for the proof that policies can be lifted; it is possible that this could be proved with different assumptions but locally compact metric spaces are general enough to cover any example we have seen.

Next we will define continuous MDP homomorphisms and establish results for both optimal value equivalence and value equivalence.

Definition 11 (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:*

$$\textit{Invariance of reward: } \overline{R}(f(s), g_s(a)) = R(s,a) \qquad \forall s \in \mathcal{S}, a \in \mathcal{A} \tag{4}$$

$$\textit{Equivariance of transitions: } \overline{\tau}_{g_s(a)}(\overline{B}|f(s)) = \tau_a(f^{-1}(\overline{B})|s) \qquad \forall\ s \in \mathcal{S}, a \in \mathcal{A}, \overline{B} \in \overline{\Sigma} \tag{5}$$

Note that if g_s is the identity map, the second condition reduces to $\overline{\tau}_a(\overline{B}|f(s)) = \tau_a(f^{-1}(\overline{B})|s)$ which is simply the condition for preservation of transition probabilities as used in bisimulation (Desharnais et al., 2002).

The condition on the rewards translates directly from the finite case. The equivariance of transitions is defined using the σ-algebra defined on the image MDP; it states that the measure $\overline{\tau}_{g_s(a)}(\cdot|f(s))$ is the pushforward measure of $\tau_a(\cdot|s)$ under the state mapping f. In the results to follow, we will see the reason we require this condition.

4. Usually the Borel algebra.

4.1 Optimal Value Equivalence

In this continuous setting, we will show that optimal value equivalence still holds. The proof is similar to Theorem 6, however, we utilize the change of variables formula (see Theorem 25 in Appendix B) to change the domain of integration in the continuous Bellman equation instead of re-indexing the summation.

Theorem 12 (Optimal Value Equivalence) *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau_a, R, \gamma)$. Then for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have:*

$$Q^*(s, a) = \overline{Q}^*(f(s), g_s(a)),$$

where $Q^*, \overline{Q}^*$ are the optimal action-value functions for $\mathcal{M}$ and $\overline{\mathcal{M}}$, respectively.

Proof We will first prove the following claim:

Claim 13 *For $m \geq 1$, define the sequence $Q_m : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as:*

$$Q_m(s, a) = R(s, a) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \sup_{a' \in \mathcal{A}} Q_{m-1}(s', a')$$

and $Q_0(s, a) = R(s, a)$. Define the sequence $\overline{Q}_m : \overline{\mathcal{S}} \times \overline{\mathcal{A}} \to \mathbb{R}$ analogously. Then for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we claim:

$$Q_m(s, a) = \overline{Q}_m(f(s), g_s(a)).$$

We will prove this claim by induction on m. The base case $m = 0$ follows from the reward invariance property of continuous MDP homomorphisms:

$$Q_0(s, a) = R(s, a) = \overline{R}(f(s), g_s(a)) = \overline{Q}_0(f(s), g_s(a)).$$

For the inductive case, note that:

$$\begin{aligned}
Q_m(s, a) &= R(s, a) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \sup_{a' \in \mathcal{A}} Q_{m-1}(s', a') && (6)\\
&= \overline{R}(f(s), g_s(a)) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \sup_{a' \in \mathcal{A}} \overline{Q}_{m-1}(f(s'), g_{s'}(a')) && (7)\\
&= \overline{R}(f(s), g_s(a)) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \sup_{a' \in \overline{\mathcal{A}}} \overline{Q}_{m-1}(f(s'), a') && (8)\\
&= \overline{R}(f(s), g_s(a)) + \gamma \int_{s' \in \overline{\mathcal{S}}} \overline{\tau}_{g_s(a)}(ds'|f(s)) \sup_{a' \in \overline{\mathcal{A}}} \overline{Q}_{m-1}(s', a') && (9)\\
&= \overline{Q_{m-1}}(f(s), g_s(a)), && (10)
\end{aligned}$$

where Equation 7 follows from the inductive hypothesis, Equation 8 follows from g_s being surjective, and Equation 9 follows from the change of variables formula; indeed, from Definition 11 we have the pushforward measure of $\tau_a(\cdot|s)$ with respect to f equals $\overline{\tau}_{g_s(a)}(\cdot|f(s))$ and here we are integrating a function from $\mathcal{S} \to \mathbb{R}$ defined as $s' \mapsto \sup_{a' \in \mathcal{A}} Q_{m-1}(s', a')$. This concludes the induction proof. Since $\lim_{m \to \infty} Q_m(s, a) = Q^*(s, a)$, it follows that $Q^*(s, a) = \overline{Q^*}(f(s), g_s(a))$. ■

4.2 Lifting Policies and Value Equivalence

Recall that in the finite setting, we had an exact equation defining lifted policies via an MDP homomorphism. In the continuous case, finding a lifted policy that exists in general and that also gives a value equivalence result is not trivial. We will use the following condition to define a lifted policy for continuous MDP homomorphisms.

Definition 14 (Policy Lifting) *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau_a, R, \gamma)$. Then for any policy $\overline{\pi} : \overline{\mathcal{S}} \to \Delta(\overline{\mathcal{A}})$ defined on $\overline{\mathcal{M}}$, a policy $\pi^{\uparrow} : S \to \Delta(\mathcal{A})$ on $\mathcal{M}$ is a* lifted *policy of $\overline{\pi}$ if:*

$$\pi^{\uparrow}(g_s^{-1}(\beta)|s) = \overline{\pi}(\beta|f(s)) \tag{11}$$

for every Borel set $\beta \subseteq \overline{\mathcal{A}}$ and $s \in S$. In other words, $\overline{\pi}(f(s), \cdot)$ is the pushforward measure of $\pi^{\uparrow}(s, \cdot)$ for all $s \in S$ with respect to g_s.

Note that Definition 14 does not define a measure, since we need to specify a value assigned to $\pi^{\uparrow}(s, B)$ for *all* Borel sets B in $\mathcal{A}$, not just those arising as inverse images $g_s^{-1}(\beta)$. However, naively defining:

$$\pi^{\uparrow}(B|s) = \overline{\pi}(g_s(B)|f(s))$$

poses immediate issues because g_s does not map Borel sets to Borel sets and $B \nsubseteq g_s^{-1}(g_s(B))$ in general. In other words, we could only use this definition if g_s is bijective and maps measurable sets to measurable sets. However, as shown in the next result, such a measure satisfying the condition in Definition 14 indeed exists in general, assuming $\mathcal{A}$ and $\overline{\mathcal{A}}$ are locally compact metric spaces. The proof uses results in functional analysis, specifically the Hahn-Banach and Riesz Representation theorem. Notably, the bijection assumption of g_s is one of the limitations of the prior work of Rezaei-Shoshtari et al. (2022), which is removed in our paper.

Proposition 15 *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau_a, R, \gamma)$, where $\mathcal{A}$ and $\overline{\mathcal{A}}$ are locally compact metric spaces. Then for any policy $\overline{\pi} : \overline{\mathcal{S}} \to \Delta(\overline{\mathcal{A}})$ defined on $\overline{\mathcal{M}}$, there exists a lifted policy $\pi^{\uparrow} : S \to \Delta(\mathcal{A})$ in the sense of Definition 14.*

Proof Define the functional $p : C_0(\mathcal{A}) \to \mathbb{R}$ as:

$$p(\psi) = \max_{a \in \mathcal{A}} \psi(a).$$

Clearly $p(\varphi + \psi) \leq p(\varphi) + p(\psi)$ and $p(\alpha\psi) = \alpha p(\psi)$ for every $\psi, \varphi \in C_0(\mathcal{A})$ and $0 < \alpha < \infty$. Indeed, p is a semi-norm. Note that since g_s is surjective, we can define the subspace $U := \{\eta \circ g_s : \eta \in C_0(\overline{\mathcal{A}})\} \subseteq C_0(\mathcal{A})$. Let ρ be the linear functional on U defined as

$$\rho(\eta \circ g_s) = \int_{a' \in \overline{\mathcal{A}}} \eta(a') \overline{\pi}(da'|f(s)).$$

We have:

$$\rho(\eta \circ g_s) \leq \overline{\pi}(f(s), \overline{\mathcal{A}}) \max_{a' \in \overline{\mathcal{A}}} \eta(a') = \max_{a \in \mathcal{A}} (\eta \circ g_s)(a) = p(\eta \circ g_s),$$

since $\overline{\pi}(\cdot|f(s))$ is a probability measure and g_s is surjective. By the Hahn-Banach theorem, we can extend ρ to a linear functional $\hat{\rho}$ on $C_0(\mathcal{A})$ where $\hat{\rho}(\psi) \leq p(\psi)$ for every $\psi \in C_0(\mathcal{A})$. It follows that if $\psi \leq 0$ then $\hat{\rho}(\psi) \leq 0$, whence if $\psi \geq 0$ then $\hat{\rho}(\psi) = -\hat{\rho}(-\psi) \geq 0$. Since this implies that $\hat{\rho}$ is a positive linear functional and $\mathcal{A}$ is a locally compact metric space, by the Riesz Representation theorem there is a unique Radon measure μ on $\mathcal{A}$ such that:

$$\hat{\rho}(\psi) = \int_{a \in \mathcal{A}} \psi(a) d\mu(a).$$

It follows that for every $\eta \in C_0(\overline{\mathcal{A}})$:

$$\int_{a \in \overline{\mathcal{A}}} \eta(a') \overline{\pi}(da'|f(s)) = \rho(\eta \circ g_s) = \int_{a \in \mathcal{A}} (\eta \circ g_s)(a) d\mu(a) = \int_{a' \in \overline{\mathcal{A}}} \eta(a') dg_{s*}\mu(da'),$$

where the first equality is by definition of ρ, the second equality follows from μ extending ρ, and the last equality following by the change of variables formula. Thus $\overline{\pi}(\cdot|f(s))$ is the pushforward measure of μ with respect to g_s. Setting $\pi^{\uparrow}(\cdot|s) = \mu$ gives the result. ■

Recall that finding a lifted policy reduces to the following question: given a surjective measurable function $g_s : \mathcal{A} \to \bar{\mathcal{A}}$ and a probability measure $\bar{\pi}$ on $\bar{\mathcal{A}}$, does there exist a measure $\pi^{\uparrow}$ on $\mathcal{A}$ such that the resulting pushforward measure $g_{s*}\pi^{\uparrow} = \bar{\pi}$? This is a result that holds more generally for analytic subsets of Polish spaces, the original result proven in Varadarajan (1963) (see Lemma 2.2).

Now that we have proven a lifted policy exists for continuous setting, we proceed to prove a value equivalence result for continuous MDP homomorphisms. The proof is very similar to optimal value equivalence, and in fact only requires one more application of change of variables with respect to the lifted policy.

Theorem 16 (Value Equivalence) *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}_{\overline{a}}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau_a, R, \gamma)$, and let $\pi^{\uparrow}$ be a lifted policy corresponding to $\overline{\pi}$. Then for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have:*

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

where $Q^{\pi^{\uparrow}}(s, a)$ and $\overline{Q}^{\overline{\pi}}(f(s), g_s(a))$ are the action-value functions for policies $\pi^{\uparrow}$ and $\overline{\pi}$ respectively.

Proof Similarly as in Theorem 12, we define the sequence $Q_m^{\pi^{\uparrow}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as:

$$Q_m^{\pi^{\uparrow}}(s, a) = R(s, a) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \int_{a' \in \mathcal{A}} \pi^{\uparrow}(da'|s') Q_{m-1}^{\pi^{\uparrow}}(s', a')$$

for $m \geq 1$ and $Q_0^{\pi^{\uparrow}}(s, a) = 0$. Analogously define $\overline{Q}_{m-1}^{\overline{\pi}} : \overline{\mathcal{S}} \times \overline{\mathcal{A}} \to \mathbb{R}$. For the inductive case, we can perform change of variables twice to change the domain of integration from S to $\overline{\mathcal{S}}$

and $\mathcal{A}$ to $\overline{\mathcal{A}}$ respectively:

$$Q_m^{\pi^\uparrow}(s,a) = R(s,a) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \int_{a' \in \mathcal{A}} \pi^\uparrow(da'|s') Q_{m-1}^{\pi^\uparrow}(s',a') \tag{12}$$

$$= \overline{R}(f(s), g_s(a)) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \int_{a' \in \mathcal{A}} \pi^\uparrow(da'|s') \overline{Q}_{m-1}^{\overline{\pi}}(f(s'), g_s(a')) \tag{13}$$

$$= \overline{R}(f(s), g_s(a)) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \int_{\overline{a} \in \overline{\mathcal{A}}} \overline{\pi}(d\overline{a}|f(s')) \overline{Q}_{m-1}^{\overline{\pi}}(f(s'), \overline{a}) \tag{14}$$

$$= \overline{R}(f(s), g_s(a)) + \gamma \int_{\overline{s} \in \overline{\mathcal{S}}} \overline{\tau}_{g_s(a)}(d\overline{s}|f(s)) \int_{\overline{a} \in \overline{\mathcal{A}}} \overline{\pi}(d\overline{a}|\overline{s}) \overline{Q}_{m-1}^{\overline{\pi}}(\overline{s}, \overline{a}) \tag{15}$$

$$= \overline{Q}_{m-1}^{\overline{\pi}}(f(s), g_s(a)). \tag{16}$$

In a similar manner to Theorem 12, we conclude that $Q^{\pi^\uparrow}(s,a) = \overline{Q}^{\overline{\pi}}(f(s), g_s(a))$. ■

Theorem 16 posits that the value function of any policy on the reduced MDP equals the value function of its corresponding lifted policy on the original MDP. Since this holds true for any optimal policy, it follows from Theorem 12 that a lifted optimal policy is optimal for the original MDP. Thus, we have recovered all desirable properties for continuous MDP homomorphisms from the finite case.

5. Homomorphic Policy Gradient

In order to directly integrate the notion of MDP homomorphisms into policy gradients and incorporate their state-action abstraction as an inductive bias for policy optimization, we derive the *homomorphic policy gradient* (HPG) theorem. Notably, our results are derived for continuous states and actions and hold for both stochastic and deterministic policies; this is in contrast to the prior work of Rezaei-Shoshtari et al. (2022) in which the derivation of the homomorphic policy gradient theorem is limited to deterministic policies.

In this section, we assume the policy is parameterized by differentiable functions (e.g., neural networks) and the MDP homomorphic image can be obtained through a parameterized homomorphism map. Importantly, learning such parameterized MDP homomorphism map is detailed in Section 6. Finally, following the prior works on policy gradient methods (Sutton et al., 2000; Silver et al., 2014), we define the performance measure on the actual MDP as described in equation (1) .

Since the derivation of the policy gradient theorem for stochastic and deterministic policies are substantially different and require distinct steps and assumptions, in the remainder of this section, we derive the homomorphic policy gradient theorem for stochastic and deterministic policies independently from one another.

5.1 Stochastic HPG Theorem

The stochastic HPG theorem can be derived with the underlying assumptions of continuous MDP homomorphisms, as in Definition 11, and the regularity conditions described in Appendix A. Notably, the only requirement on the MDP homomorphism map is that $f : \mathcal{S} \to \overline{\mathcal{S}}$ and $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ are measurable, surjective maps adhering to the invariance of reward and equivariance of transitions in Definition 11. This is in contrast to the deterministic HPG theorem which poses further restrictions on g_s.

Theorem 17 (Stochastic Homomorphic Policy Gradient) *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau, R, \gamma)$, and let $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \Delta(\overline{\mathcal{A}})$ be a stochastic policy defined on $\overline{\mathcal{M}}$. Then the gradient of the performance measure $J(\theta)$ w.r.t. θ is:*

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

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

Proof The proof follows along the same lines of the stochastic policy gradient theorem (Sutton et al., 2000). First, we derive a recursive expression for $\nabla_\theta V^{\pi^\uparrow_\theta}(s)$ as:

$$\begin{aligned}
\nabla_\theta V^{\pi^\uparrow_\theta}(s) &= \nabla_\theta \int_{a \in \mathcal{A}} \pi^\uparrow_\theta(da|s) Q^{\pi^\uparrow_\theta}(s, a) \\
&= \int_{a \in \mathcal{A}} \Big[\nabla_\theta \pi^\uparrow_\theta(da|s) Q^{\pi^\uparrow_\theta}(s, a) + \pi^\uparrow_\theta(da|s) \nabla_\theta Q^{\pi^\uparrow_\theta}(s, a) \Big] \\
&= \int_{a \in \mathcal{A}} \Big[\nabla_\theta \pi^\uparrow_\theta(da|s) Q^{\pi^\uparrow_\theta}(s, a) + \pi^\uparrow_\theta(da|s) \nabla_\theta \Big(R(s, a) + \gamma \int_{s' \in \mathcal{S}} \tau_a(ds'|s) V^{\pi^\uparrow_\theta}(s') \Big) \Big] \\
&= \int_{a \in \mathcal{A}} \nabla_\theta \pi^\uparrow_\theta(da|s) Q^{\pi^\uparrow_\theta}(s, a) + \gamma \int_{a \in \mathcal{A}} \pi^\uparrow_\theta(da|s) \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \nabla_\theta V^{\pi^\uparrow_\theta}(s') \\
&= \int_{a \in \mathcal{A}} \nabla_\theta \pi^\uparrow_\theta(da|s) \overline{Q}^{\overline{\pi}_\theta}(f(s), g_s(a)) + \gamma \int_{a \in \mathcal{A}} \pi^\uparrow_\theta(da|s) \int_{s' \in \mathcal{S}} \tau_a(ds'|s) \nabla_\theta V^{\overline{\pi}_\theta}(f(s')) \qquad (17) \\
&= \int_{a \in \mathcal{A}} \nabla_\theta \pi^\uparrow_\theta(da|s) \overline{Q}^{\overline{\pi}_\theta}(f(s), g_s(a)) + \gamma \int_{a \in \mathcal{A}} \pi^\uparrow_\theta(da|s) \int_{\overline{s} \in \overline{\mathcal{S}}} \overline{\tau}_{g_s(a)}(d\overline{s}|f(s)) \nabla_\theta V^{\overline{\pi}_\theta}(\overline{s}) \qquad (18) \\
&= \int_{\overline{a} \in \overline{\mathcal{A}}} \nabla_\theta \overline{\pi}_\theta(d\overline{a}|f(s)) \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a}) + \gamma \int_{\overline{a} \in \overline{\mathcal{A}}} \overline{\pi}_\theta(d\overline{a}|f(s)) \int_{\overline{s} \in \overline{\mathcal{S}}} \overline{\tau}_{\overline{a}}(d\overline{s}|f(s)) \nabla_\theta V^{\overline{\pi}_\theta}(\overline{s}).
\end{aligned}$$

Here we apply value equivalence in equation (17), a change of variables from $\mathcal{S}$ to $\overline{\mathcal{S}}$ over $\tau_a(\cdot|s)$ and $\overline{\tau}_{g_s(a)}(\cdot|f(s))$ respectively in equation (18), and a change of variables from $\mathcal{A}$ to $\overline{\mathcal{A}}$ over the lifted policy and the policy over the abstract MDP respectively. Note that here, some care may be necessary to rigorously verify the interchanging of the gradient over θ and the integral over $\mathcal{A}$; however, this is a necessary condition to prove any type of policy gradient result on continuous domains, not specifically to the stochastic HPG theorem.

As in the proof of the stochastic policy gradient theorem, we can continue to roll out the definition of $\nabla_\theta V^{\overline{\pi}_\theta}(\overline{s})$ in the space of the *abstract MDP* $\overline{\mathcal{M}}$. Denoting $\overline{\pi}^k(\cdot|f(s))$ to be the probability distribution over $\overline{\mathcal{S}}$ taking k steps following $\overline{\pi}$ from state $f(s)$, we have:

$$\nabla_\theta V^{\pi^\uparrow_\theta}(s) = \sum_{k=0}^{\infty} \gamma^k \int_{\overline{s} \in \overline{\mathcal{S}}} \overline{\pi}^k_\theta(d\overline{s}|f(s)) \int_{\overline{a} \in \overline{\mathcal{A}}} \nabla_\theta \overline{\pi}_\theta(d\overline{a}|f(s)) \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a}).$$

Finally, we conclude that:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta V^{\pi^\uparrow_\theta}(s) \\
&= \int_{\overline{s} \in \overline{\mathcal{S}}} \sum_{k=0}^{\infty} \gamma^k \overline{\pi}^k_\theta(d\overline{s}|f(s)) \int_{\overline{a} \in \overline{\mathcal{A}}} \nabla_\theta \overline{\pi}_\theta(d\overline{a}|f(s)) \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a}) \\
&= \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(d\overline{s}) \int_{\overline{a} \in \overline{\mathcal{A}}} \overline{Q}^{\overline{\pi}_\theta}(\overline{s}, \overline{a}) \nabla_\theta \overline{\pi}_\theta(d\overline{a}|\overline{s}),
\end{aligned}$$

as desired, where $\rho^{\overline{\pi}_\theta}(\overline{s})$ is the discounted stationary distribution induced by the policy $\overline{\pi}_\theta$. ■

5.2 Deterministic HPG Theorem

In contrast to stochastic HPG where the homomorphism map can be any measurable surjective map, the deterministic case requires the action encoder $g_s : \mathcal{A} \to \overline{\mathcal{A}}$ to be a *local diffeomorphism* (see Appendix B for definitions). The important implication of this requirement is that the action encoder g_s needs to be locally bijective, hence the abstract action space must have the same dimensionality as the actual action space. First, we show the *equivalence of policy gradients*:

Theorem 18 (Equivalence of Deterministic Policy Gradients) *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau, R, \gamma)$, and let $\pi^\uparrow_\theta : \mathcal{S} \to \mathcal{A}$ be the lifted deterministic policy corresponding to the abstract deterministic policy $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$. Then for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have:*

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

Proof Assuming the conditions described in Appendix A, we first take the derivative of the deterministic policy lifting relation w.r.t. the policy parameters θ using the chain rule:

$$\begin{aligned}
(g_s \circ \pi^\uparrow_\theta)(s) &= (\overline{\pi}_\theta \circ f)(s) \\
d(g_s \circ \pi^\uparrow_\theta)_\theta(s) &= d(\overline{\pi}_\theta \circ f)_\theta(s) \\
d(g_s)_{\pi^\uparrow_\theta(s)} \circ d(\pi^\uparrow_\theta)_\theta(s) &= d(\overline{\pi}_\theta \circ f)_\theta(s) \\
\underbrace{\nabla_a g_s(a)\big|_{a=\pi^\uparrow_\theta(s)}}_{P} \nabla_\theta \pi^\uparrow_\theta(s) &= \nabla_\theta \overline{\pi}_\theta(f(s)),
\end{aligned} \tag{19}$$

where $\circ$ is the composition operator and the dimensions of the matrices are $P \in \mathbb{R}^{|\overline{\mathcal{A}}| \times |A|}$, $\nabla_\theta \pi^\uparrow_\theta(s) \in \mathbb{R}^{|A| \times |\theta|}$, and $\nabla_\theta \overline{\pi}_\theta(\overline{s}) \in \mathbb{R}^{|\overline{\mathcal{A}}| \times |\theta|}$. Second, we take the derivative of the value equivalence theorem w.r.t. the actions a using the chain rule:

$$\begin{aligned}
Q^{\pi^\uparrow_\theta}(s,a) &= \overline{Q}^{\overline{\pi}_\theta}(f(s), g_s(a)) \\
dQ^{\pi^\uparrow_\theta}(s,a)_a &= d\overline{Q}^{\overline{\pi}_\theta}(f(s), g_s(a))_a \\
\nabla_a Q^{\pi^\uparrow_\theta}(s,a)\big|_{a=\pi^\uparrow_\theta(s)} &= \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s),\overline{a})\big|_{\overline{a}=\overline{\pi}_\theta(f(s))} \underbrace{\nabla_a g_s(a)\Big|_{a=g_s^{-1}(\overline{\pi}_\theta(f(s)))}}_{P},
\end{aligned} \tag{20}$$

where the dimensions of the matrices are $\nabla_a Q^{\pi^\uparrow_\theta}(s,a) \in \mathbb{R}^{|A|}$, $\nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(\overline{s},\overline{a}) \in \mathbb{R}^{|\overline{\mathcal{A}}|}$, and similarly as before $P \in \mathbb{R}^{|\overline{\mathcal{A}}| \times |A|}$. As we assumed the g_s to be a local diffeomorphism, the inverse function theorem states that the matrix P is invertible, thus we right-multiply both sides of equation (20) by P^{-1} and left-multiply the resulting equation by equation (19) to obtain

the desired result:

$$\nabla_a Q^{\pi^{\uparrow}_\theta}(s,a)\big|_{a=\pi^{\uparrow}_\theta(s)} P^{-1} P \nabla_\theta \pi^{\uparrow}_\theta(s) = \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s),\overline{a})\big|_{\overline{a}=\overline{\pi}(f(s))} \nabla_\theta \overline{\pi}_\theta(f(s))$$
$$\nabla_a Q^{\pi^{\uparrow}_\theta}(s,a)\big|_{a=\pi^{\uparrow}_\theta(s)} \nabla_\theta \pi^{\uparrow}_\theta(s) = \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s),\overline{a})\big|_{\overline{a}=\overline{\pi}_\theta(f(s))} \nabla_\theta \overline{\pi}_\theta(f(s)).$$

■

Theorem 18 highlights that the gradient of the abstract MDP is equivalent to that of the original, despite the underlying spaces being abstracted. This implies that performing HPG on the abstract MDP is equivalent to performing DPG on the actual MDP, allowing us to use them synergistically to update the same parameters θ, as shown in Figure 2.

While one can naively use Theorem 18 to substitute gradients of the standard DPG, theoretically this does not produce any useful results as the expectation remains estimated with respect to the stationary state distribution of the actual MDP $\mathcal{M}$ under $\pi^{\uparrow}_\theta(s)$. However, using properties of continuous MDP homomorphisms, we can change the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$, and consequently estimate the policy gradient with respect to the stationary distribution of the abstract MDP $\overline{\mathcal{M}}$ under $\overline{\pi}_\theta(\overline{s})$:

Theorem 19 (Deterministic Homomorphic Policy Gradient) *Let $\overline{\mathcal{M}} = (\overline{\mathcal{S}}, \overline{\Sigma}, \overline{\mathcal{A}}, \overline{\tau}, \overline{R}, \overline{\gamma})$ be the image of a continuous MDP homomorphism $h = (f, g_s)$ from $\mathcal{M} = (\mathcal{S}, \Sigma, \mathcal{A}, \tau, R, \gamma)$, and let $\overline{\pi}_\theta : \overline{\mathcal{S}} \to \overline{\mathcal{A}}$ be 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:*

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

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

Proof The proof follows along the same lines of the deterministic policy gradient theorem (Silver et al., 2014), but with additional steps for changing the integration space from $\mathcal{S}$ to $\overline{\mathcal{S}}$. First, we derive a recursive expression for $\nabla_\theta V^{\pi^{\uparrow}_\theta}(s)$ as:

$$\begin{aligned}
\nabla_\theta V^{\pi^{\uparrow}_\theta}(s) &= \nabla_\theta Q^{\pi^{\uparrow}_\theta}\Big(s, \pi^{\uparrow}_\theta(s)\Big) \\
&= \nabla_\theta \Big[R(s, \pi^{\uparrow}_\theta(s)) + \gamma \int_{s'\in\mathcal{S}} \tau_{\pi^{\uparrow}_\theta(s)}(ds'|s) V^{\pi^{\uparrow}_\theta}(s') \Big] \\
&= \nabla_\theta \pi^{\uparrow}_\theta(s) \nabla_a R(s,a)\Big|_{a=\pi^{\uparrow}_\theta(s)} \\
&\quad + \gamma \int_{s'\in\mathcal{S}} \Big[\tau_{\pi^{\uparrow}_\theta(s)}(ds'|s) \nabla_\theta V^{\pi^{\uparrow}_\theta}(s') + \nabla_\theta \pi^{\uparrow}_\theta(s) \nabla_a \tau_a(ds'|s)\Big|_{a=\pi^{\uparrow}_\theta(s)} V^{\pi^{\uparrow}_\theta}(s') \Big] \quad (21) \\
&= \nabla_\theta \pi^{\uparrow}_\theta(s) \nabla_a \Big[R\big(s,a\big) + \gamma \int_{s'\in\mathcal{S}} \tau_a(ds'|s) V^{\pi^{\uparrow}_\theta}(s') \Big]\Big|_{a=\pi^{\uparrow}_\theta(s)} + \gamma \int_{s'\in\mathcal{S}} \tau_{\pi^{\uparrow}_\theta(s)}(ds'|s) \nabla_\theta V^{\pi^{\uparrow}_\theta}(s') \\
&= \nabla_\theta \pi^{\uparrow}_\theta(s) \nabla_a Q^{\pi^{\uparrow}_\theta}(s,a)\Big|_{a=\pi^{\uparrow}_\theta(s)} + \gamma \int_{s'\in\mathcal{S}} \tau_{\pi^{\uparrow}_\theta(s)}(ds'|s) \nabla_\theta V^{\overline{\pi}_\theta}(f(s')) \quad (22) \\
&= \nabla_\theta \pi^{\uparrow}_\theta(s) \nabla_a Q^{\pi^{\uparrow}_\theta}(s,a)\Big|_{a=\pi^{\uparrow}_\theta(s)} + \gamma \int_{\overline{s}'\in\overline{\mathcal{S}}} \overline{\tau}_{g_s(\pi^{\uparrow}_\theta(s))}(d\overline{s}'|f(s)) \nabla_\theta V^{\overline{\pi}_\theta}(f(s')) \quad (23)
\end{aligned}$$

$$= \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} + \gamma \int_{\overline{s}' \in \overline{\mathcal{S}}} \overline{\tau}_{\overline{\pi}_\theta(\overline{s})}(d\overline{s}'|\overline{s}) \nabla_\theta V^{\overline{\pi}_\theta}(\overline{s}') \tag{24}$$

$$= \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} + \gamma \int_{\overline{s}' \in \overline{\mathcal{S}}} p(\overline{s} \to \overline{s}', 1, \overline{\pi}_\theta) \nabla_\theta V^{\overline{\pi}_\theta}(\overline{s}') d\overline{s}'.$$

Where $p(\overline{s} \to \overline{s}', t, \overline{\pi}_\theta)$ is the probability of going from $\overline{s}$ to $\overline{s}'$ under the policy $\overline{\pi}_\theta(\overline{s})$ in t time steps. In equation (21) we were able to apply the Leibniz integral rule to exchange the order of derivative and integration because of the regularity conditions on the continuity of the functions. In equation (22) we used the value equivalence property, and in equation (23) we used the change of variables formula based on the pushforward measure of $\tau_a(\cdot|s)$ with respect to f. Finally, in equation (24) we used the equivalence of policy gradients from Theorem 18. By recursively rolling out the formula above, we obtain:

$$\begin{aligned}
\nabla_\theta V^{\pi^\uparrow_\theta}(s) &= \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} \\
&\quad + \gamma \int_{\overline{s}' \in \overline{\mathcal{S}}} p(\overline{s} \to \overline{s}', 1, \overline{\pi}_\theta) \nabla_\theta \overline{\pi}_\theta(f(s')) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s'), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s'))} d\overline{s}' \\
&\quad + \gamma^2 \int_{\overline{s}' \in \overline{\mathcal{S}}} p(\overline{s} \to \overline{s}', 1, \overline{\pi}_\theta) \int_{\overline{s}'' \in \overline{\mathcal{S}}} p(\overline{s}' \to \overline{s}'', 1, \overline{\pi}_\theta) \nabla_\theta V^{\pi^\uparrow_\theta}(f(s'')) d\overline{s}'' d\overline{s}' \\
&= \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} \\
&\quad + \gamma \int_{\overline{s}' \in \overline{\mathcal{S}}} p(\overline{s} \to \overline{s}', 1, \overline{\pi}_\theta) \nabla_\theta \overline{\pi}_\theta(f(s')) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s'), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s'))} d\overline{s}' \\
&\quad + \gamma^2 \int_{\overline{s}'' \in \overline{\mathcal{S}}} p(\overline{s} \to \overline{s}'', 2, \overline{\pi}_\theta) \nabla_\theta V^{\overline{\pi}_\theta}(f(s'')) d\overline{s}'' \qquad (25) \\
&\vdots \\
&= \int_{\overline{s}' \in \overline{\mathcal{S}}} \sum_{t=0}^{\infty} \gamma^t p(\overline{s} \to \overline{s}', t, \overline{\pi}_\theta) \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} d\overline{s}'. \qquad (26)
\end{aligned}$$

Where in equation (25) we exchanged the order of integration using the Fubini's theorem that requires the boundedness of $\|\nabla_\theta V^{\overline{\pi}_\theta}(s)\|$ as described in the regularity conditions. Finally, we take the expectation of $\nabla_\theta V^{\pi^\uparrow_\theta}(s)$ over the initial state distribution:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \int_{s \in \mathcal{S}} p_1(s) V^{\pi^\uparrow_\theta}(s) ds \\
&= \int_{s \in \mathcal{S}} p_1(s) \nabla_\theta V^{\pi^\uparrow_\theta}(s) ds \\
&= \int_{s \in \mathcal{S}} p_1(s) \int_{\overline{s}' \in \overline{\mathcal{S}}} \sum_{t=0}^{\infty} \gamma^t p(\overline{s} \to \overline{s}', t, \overline{\pi}_\theta) \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} d\overline{s}' ds \\
&= \int_{\overline{s} \in \overline{\mathcal{S}}} \overline{p}_1(\overline{s}) \int_{\overline{s}' \in \overline{\mathcal{S}}} \sum_{t=0}^{\infty} \gamma^t p(\overline{s} \to \overline{s}', t, \overline{\pi}_\theta) \nabla_\theta \overline{\pi}_\theta(f(s)) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(f(s), \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(f(s))} d\overline{s}' d\overline{s} \qquad (27) \\
&= \int_{\overline{s} \in \overline{\mathcal{S}}} \rho^{\overline{\pi}_\theta}(\overline{s}) \nabla_\theta \overline{\pi}_\theta(\overline{s}) \nabla_{\overline{a}} \overline{Q}^{\overline{\pi}_\theta}(\overline{s}, \overline{a})\Big|_{\overline{a}=\overline{\pi}_\theta(\overline{s})} d\overline{s}. \qquad (28)
\end{aligned}$$

Where $\rho^{\overline{\pi}_\theta}(\overline{s})$ is the discounted stationary distribution induced by the policy $\overline{\pi}_\theta$. In equation (27) we used the change of variable formula. Similar to the steps before, we have used

the Leibniz integral rule to exchange the order of integration and derivative, used Fubini's theorem to exchange the order of integration. ■

5.3 Comparing the Stochastic and Deterministic HPG Theorems

The significance of the homomorphic policy gradients (Theorems 17 and 19), which form the basis of our proposed homomorphic actor-critic algorithms, is twofold. First, we can get another estimate for the policy gradient based on the approximate MDP homomorphic image in addition to the standard policy gradient estimator. Although the two policy gradient estimates are not statistically independent from one another as they are tied through the homomorphism map, HPG will potentially have less variance at the expense of some bias due to the approximation of the MDP homomorphism.

Second, since the minimal image of an MDP is the MDP homomorphic image (Ravindran and Barto, 2001), the abstract critic $\overline{Q}^{\overline{\pi}_\theta}$ is trained on a simplified problem. In other words, each abstract state-action pair $(\overline{s}, \overline{a})$ used to train $\overline{Q}^{\overline{\pi}_\theta}$ represents all (s, a) pairs that are equivalent under the MDP homomorphism relation, thus improving sample efficiency. However, the amount of complexity reduction is dependent on the approximate symmetries of the environment, as also supported by our empirical results.

Figure 2 shows the schematics of the homomorphic policy gradient theorem and its tangential use alongside the standard PG theorem. To conclude this section, we provide a conceptual comparison between the stochastic and deterministic HPG variants, following up with an empirical comparison in Section 7.

Dimensionality Reduction in the Action Space. A key aspect of MDP homomorphisms is the notion of "*collapse*": the state map f and state-dependent action maps g_s are specifically surjective. For instance, continuous symmetries of a physical system with respect to an action corresponds to an invariance of a quantity, and effectively allows for reduction in the dimensionality of the action space (Noether, 1971; Bluman and Kumei, 2013). In the context of RL agents, the ability to identify and leverage continuous symmetries of the environment results in the dimensionality reduction of the action space which in turn significantly simplifies the learning problem. However, such action reductions do not meet the conditions of the deterministic HPG theorem, as it requires the action map g_s to be a *local diffeomorphism*. Thus, the underlying assumptions do not account for strict collapses. In contrast, the stochastic HPG does not impose any additional structure on g_s, which consequently allows for effective dimensionality reduction of the action space, without risking the optimality of the policy.

Maximum Entropy RL. Having the result for stochastic policies give theoretical guarantees when integrating MDP homomorphisms in a wider variety of algorithms. For instance, the *maximum entropy RL* framework generalizes the expected return formulation to encompass the entropy of the policy, resulting in improvements in robustness and exploration (Ziebart et al., 2008; Ziebart, 2010; Haarnoja et al., 2018). Of course, such methods are only applicable to stochastic policies. Thus, in contrast to the deterministic HPG, stochastic HPG is capable of benefiting from the addition of the policy's entropy. Importantly, the entropy of the pushforward measure is at most the entropy of the original

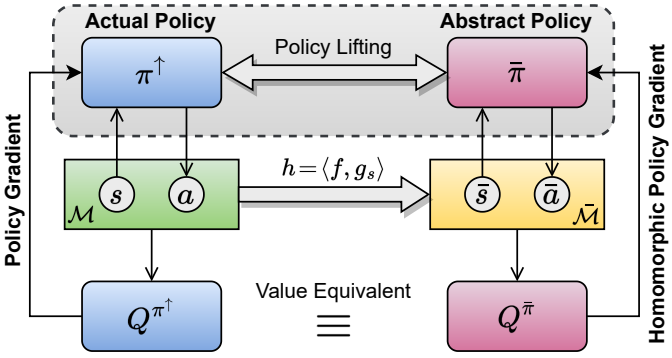


Figure 2: Schematics of HPG. The actual MDP $\mathcal{M}$ is used to train $Q^{\pi^{\uparrow}}$ and update $\pi^{\uparrow}$ with the standard PG theorem, while the abstract MDP $\overline{\mathcal{M}}$ is used to train $\overline{Q}^{\overline{\pi}}$ and update $\overline{\pi}$ with the homomorphic PG theorem. $\overline{\mathcal{M}}$ is the MDP homomorphic image of $\mathcal{M}$ obtained by learning the homomorphism map $h=(f, g_s)$. The policies $\pi^{\uparrow}$ and $\overline{\pi}$ are coupled together through the lifting procedure.

measure (this is a consequence of the conditional Jensen's inequality— see Smorodinsky (2006)); hence seeking maximal entropy policies in the abstract MDP correspond to high-entropy policies in the original space as well.

Sample Efficiency. The sample efficiency of deterministic and stochastic PG methods varies significantly depending on the choice of the algorithm, network architecture, exploration strategy, and implementation (Henderson et al., 2018); nevertheless, it is generally observed that deterministic PG methods are more sample efficient (Lillicrap et al., 2015; Fujimoto et al., 2018; Barth-Maron et al., 2018). One hypothesis is that since deterministic PG integrates only over the state space, in contrast to stochastic PG which integrates over both state and action spaces, the policy gradient estimation is more sample efficient, particularly in high-dimensional action spaces (Silver et al., 2014). The same reasoning is applicable to HPG variants. We have carried out a thorough empirical study on a variety of environments in Section 7 to further study the characteristics of the two HPG variants.

6. Homomorphic Actor-Critic Algorithms

In this section, we outline a practical deep reinforcement learning algorithm based on the stochastic and deterministic HPG theorems, referred to as the *Deep Homomorphic Policy Gradient* (DHPG) algorithms. While the overall structure of the algorithm and learning the MDP homomorphisms map are similar in both cases of stochastic and deterministic policies, the policy lifting procedure requires additional intricate steps in the stochastic case. Algorithm 1 describes the pseudo-code of DHPG algorithms.

Denoting pixel observations as o_t, the underlying states as s_t, and the abstract states as $\bar{s}_t$, the main components of the DHPG algorithm are: the MDP homomorphism map $h_{\phi,\eta}=(f_\phi(s_t), g_\eta(s_t, a_t))$, pixel encoder $E_\mu(o_t)$, actual critic $Q_\psi(s_t, a_t)$ and policy $\pi^{\uparrow}_\theta(a_t|s_t)$, abstract critic $\overline{Q}_{\overline{\psi}}(\bar{s}_t, \bar{a}_t)$ and policy $\overline{\pi}_{\overline{\theta}}(\bar{a}_t|\bar{s}_t)$, reward predictor $\overline{R}_\rho(\bar{s}_t)$, and probabilistic transition dynamics $\overline{\tau}_\nu(\bar{s}_{t+1}|\bar{s}_t, \bar{a}_t)$ which outputs a Gaussian distribution. Finally, we lever-

age target critic networks $Q_{\psi'}$ and $\overline{Q}_{\overline{\psi}'}$ for a more stable training and use a vanilla replay buffer (Mnih et al., 2013; Lillicrap et al., 2015).

Policy Lifting Procedure. In general, the lifted policy needs to satisfy the relation $\pi^{\uparrow}(g_s^{-1}(\beta)|s) = \overline{\pi}(\beta|f(s))$ for every Borel set $\beta \subseteq \overline{\mathcal{A}}$ and $s \in \mathcal{S}$. As discussed in Section 4.2, Proposition 15 proves the existence of the lifted policy $\pi^{\uparrow}$ from an abstract policy $\overline{\pi}$, however, it does not provide an explicit method for construction of the lifted policy.

If the abstract policy is deterministic, the lifted policy can be simply obtained by choosing one representative for the preimage $g_s^{-1}\big(\overline{\pi}(f(s))\big)$. If we select g_s to be a bijection, as was assumed in Section 5.2, the lifted policy can be uniquely defined as $\pi^{\uparrow}(s) = g_s^{-1}\big(\overline{\pi}(f(s))\big)$. This allows for parameterizing the two policies using the same network. In practice, we parameterize the actual policy $\pi_\theta^{\uparrow}$ and obtain the abstract policy as $\overline{\pi}_\theta(f(s)) = g_s(\pi_\theta^{\uparrow}(s))$.

The solution is not as straightforward for stochastic abstract policies; while Bayesian approaches for constructing solutions to stochastic inverse problems exist (Butler et al., 2018), we choose a sampling-based method to derive a loss function as an approximation of the policy lifting procedure (Kaipio and Somersalo, 2006). 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:

$$\mathbb{E}_{\pi^{\uparrow}}[g_s(a)|s] = \int_{a\in\mathcal{A}} g_s(a)\pi^{\uparrow}(da|s) = \int_{\overline{a}\in\overline{\mathcal{A}}} \overline{a}\;\overline{\pi}(d\overline{a}|\overline{s}) = \mathbb{E}_{\overline{\pi}}[\overline{a}|f(s)].$$

A similar result holds for all finite moments; in particular, the conditional variance of abstract actions under the two policies are equal— that is:

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

Therefore, we can derive a policy lifting loss as a measure of the consistency of the two policies with respect to the MDP homomorphism map and the lifting procedure. Assuming the policies $\pi_\theta^{\uparrow}$ and $\overline{\pi}_{\overline{\theta}}$ are parameterized by independent neural networks, the loss function is obtained by matching the conditional expectation and standard deviation (SD) of abstract actions conditioned on observations sampled from the replay buffer:

$$\mathcal{L}_{\text{lift.}}(\theta,\overline{\theta}) = \big(\mathbb{E}_{\pi_\theta^{\uparrow}}[g_s(a)|s] - \mathbb{E}_{\overline{\pi}_{\overline{\theta}}}[\overline{a}|f(s)]\big)^2 + \big(\mathrm{SD}_{\pi_\theta^{\uparrow}}[g_s(a)|s] - \mathrm{SD}_{\overline{\pi}_{\overline{\theta}}}[\overline{a}|f(s)]\big)^2. \tag{29}$$

As discussed, the policy lifting loss in Equation (29) is not required for deterministic DHPG.

Training the Critic. Actual and abstract critics are trained using n-step TD error for a faster reward propagation (Barth-Maron et al., 2018). The loss function for each critic is therefore defined as the expectation of the n-step Bellman error estimated over transitions samples from the replay buffer $\mathcal{B}$:

$$\mathcal{L}_{\text{actual critic}}(\psi) = \mathbb{E}_{(s,a,s',r)\sim\mathcal{B}}\Big[\big(R_t^{(n)} + \gamma^n Q_{\psi'}(s_{t+n},a_{t+n}) - Q_\psi(s_t,a_t)\big)^2\Big] \tag{30}$$

$$\mathcal{L}_{\text{abstract critic}}(\overline{\psi},\phi,\eta) = \mathbb{E}_{(s,a,s',r)\sim\mathcal{B}}\Big[\big(R_t^{(n)} + \gamma^n \overline{Q}_{\overline{\psi}'}(\overline{s}_{t+n},\overline{a}_{t+n}) - \overline{Q}_{\overline{\psi}}(\overline{s}_t,\overline{a}_t)\big)^2\Big], \tag{31}$$

where $\overline{s}_t = f_\phi(s_t)$ and $\overline{a}_t = g_\eta(s_t,a_t)$ are computed using the learned MDP homomorphism, ψ' and $\overline{\psi}'$ denote parameters of target networks, and $R_t^{(n)} = \sum_{i=0}^{n-1}\gamma^i r_{t+i}$ is the n-step return.

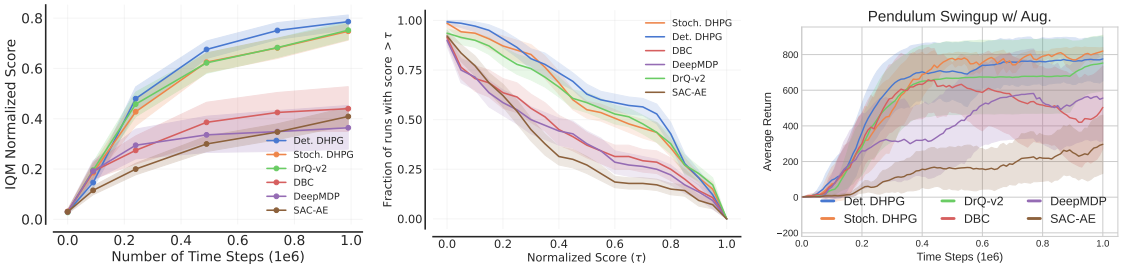
Training the Actor. For stochastic policies, we train the actual policy using the standard PG theorem (Sutton et al., 2000), and train the abstract policy via the stochastic HPG (Theorem 17). While we can employ any stochastic actor-critic algorithm for training the actual policy, we use SAC (Haarnoja et al., 2018) to further demonstrate the applicability of our method to the maximum entropy RL framework. Notably, as discussed in Section 5.3, the entropy regularizer term needs to be accounted for only during the actual policy update.

In the case of deterministic policies, the actual policy is trained using the deterministic PG (Silver et al., 2014) and the abstract policy is updated using the deterministic HPG (Theorem 19). Notably, since in this case the actual and abstract policies share the same set of parameters ($\theta = \overline{\theta}$), both policy updates are applied to the same set of policy parameters.

Learning the continuous MDP Homomorphism Map. We learn the continuous MDP homomorphism map using the lax bisimulation metric (Taylor et al., 2008), similarly to the prior work on continuous MDP homomorphisms (Rezaei-Shoshtari et al., 2022). We use the lax bisimulation metric (Taylor et al., 2008), Equation (32), to propose a loss

Algorithm 1 Deep Homomorphic Policy Gradient (DHPG)

Initialize target networks $\psi' \leftarrow \psi$, $\overline{\psi'} \leftarrow \overline{\psi}$.
for $t = 0$ to T **do**
  Encode observation: $s_t = E_\mu(o_t)$
  Select and execute action:
  if stochastic policy **then**
    $a_t \sim \pi^{\uparrow}_{\theta}(\cdot|s_t)$
  else
    $a_t = \pi^{\uparrow}_{\theta}(s_t) + \epsilon$, where $\epsilon \sim N(0, \sigma)$
  end if
  Store transition in the replay buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup (s_t, a_t, s_{t+1}, r_t)$
  Sample a mini-batch: $B_i \sim \mathcal{B}$
  Permute the mini-batch: $B_j = \text{permute}(B_i)$
  Train $h_{\phi,\eta}, E_\mu, \overline{\tau}_\nu, \overline{R}_\rho$: $\mathcal{L}_{\text{lax}} + \mathcal{L}_{\text{h}}$ ▷ Eqns. 33-34
  Train critics $Q_\psi, \overline{Q}_{\overline{\psi}}$: $\mathcal{L}_{\text{actual}} + \mathcal{L}_{\text{abstract}}$ ▷ Eqns. 30-31
  if stochastic policy **then**
    # Lifted and abstract policies are respectively parameterized by θ and $\overline{\theta}$
    Train $\pi^{\uparrow}_{\theta}$ using PG + MaxEnt ▷ SAC (Haarnoja et al., 2018)
    Train $\overline{\pi}_{\overline{\theta}}$ using stochastic HPG ▷ Theorem 17
    Update policies $\pi^{\uparrow}_{\theta}$ and $\overline{\pi}_{\overline{\theta}}$ with the policy lifting loss: $\mathcal{L}_{\text{lift.}}$ ▷ Eqn. 29
  else
    # Lifted and abstract policies share the same parameters θ
    Train $\pi^{\uparrow}_{\theta}$ using DPG ▷ TD3 (Fujimoto et al., 2018)
    Train $\overline{\pi}_{\theta}$ using deterministic HPG ▷ Theorem 19
  end if
  Update target networks: $\psi' \leftarrow \alpha\psi + (1-\alpha)\psi'$, $\overline{\psi'} \leftarrow \alpha\overline{\psi} + (1-\alpha)\overline{\psi'}$
end for



(a) Sample efficiency. (b) Performance profiles. (c) Learning curves.

Figure 3: Results of DM Control tasks with pixel observations obtained on 10 seeds. RLiable metrics are aggregated over 14 tasks. All methods are **with** image augmentation. **(a)** RLiable IQM scores as a function of number of steps for comparing sample efficiency, **(b)** RLiable performance profiles at 500k steps, **(c)** learning curves on the pendulum swingup task. Full results are in Appendix C.1. Shaded regions represent 95% confidence intervals.

function that encodes lax bisimilar states closer together in the abstract space. The lax bisimulation metric is applicable to continuous actions and as a (pseudo-)metric, it can naturally represent approximate MDP homomorphisms. The lax bisimulation metric between two state-action pairs (s_i, a_i) and (s_j, a_j) is defined as:

$$d_{\text{lax}}\big((s_i,a_i),(s_j,a_j)\big) = c_r|R(s_i,a_i) - R(s_j,a_j)| + c_t W_1\big(\tau(\cdot|s_i,a_i),\tau(\cdot|s_j,a_j)\big), \tag{32}$$

where the first term measures the distance between the reward terms and W_1 is the Kantorovich metric measuring the distance between transition probabilities. Following the same intuition as prior works on using bisimulations for representation learning (Zhang et al., 2020), we define our proposed lax bisimulation loss over pairs of transition tuples sampled from the replay buffer. We permute samples to compute their pairwise distance in the abstract space and their pairwise lax bisimilarity distance. Consequently, we minimize the distance between these two terms:

$$\mathcal{L}_{\text{lax}}(\phi,\eta) = \mathbb{E}_{\mathcal{B}}\Big[\|f_\phi(s_i) - f_\phi(s_j)\|_1 - \|r_i - r_j\|_1 - \alpha W_2\big(\overline{\tau}_\nu(\cdot|f_\phi(s_i), g_\eta(s_i,a_i)), \overline{\tau}_\nu(\cdot|f_\phi(s_j), g_\eta(s_j,a_j))\big)\Big]^2 \tag{33}$$

Here, the expectation is taken over two samples $(s_i, a_i, s_i', r_i), (s_j, a_j, s_j', r_j) \sim \mathcal{B}$ from the replay buffer.

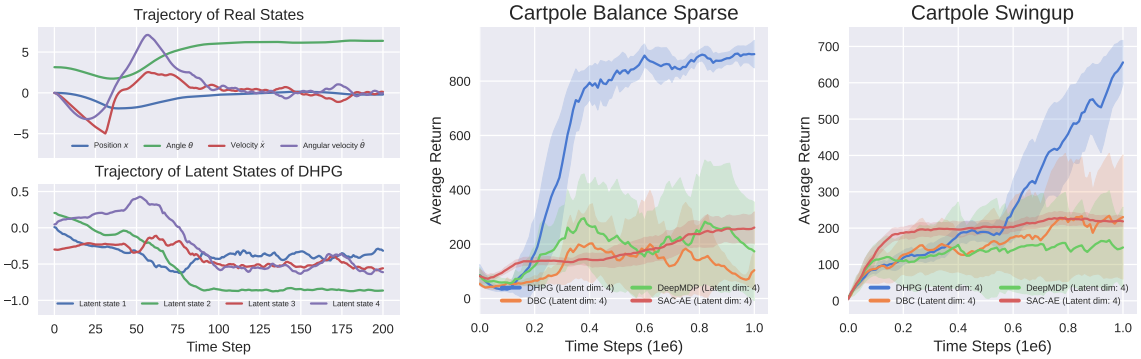
Similarly to Zhang et al. (2020), we replaced the Kantorovich (W_1) metric with the W_2 metric as there is an explicit formula for it for Gaussian distributions. Finally, we apply the conditions of a continuous MDP homomorphism map from Definition 11 via the loss function of:

$$\mathcal{L}_{\text{h}}(\phi,\eta,\nu,\rho) = \mathbb{E}_{(s_i,a_i,s_i',r_i)\sim\mathcal{B}}\Big[\big(f_\phi(s_i') - \overline{s}_i'\big)^2 + \big(r_i - \overline{R}_\rho(f_\phi(s_i))\big)^2\Big], \tag{34}$$

where $\overline{s}_i' \sim \overline{\tau}_\nu(\cdot|f_\phi(s_i), g_\eta(s_i,a_i))$. The final loss function is obtained as $\mathcal{L}_{\text{lax}}(\phi,\eta) + \mathcal{L}_{\text{h}}(\phi,\eta,\rho,\nu)$.

7. Experiments

In our experiments, we aim to answer the following key questions:



(a) State trajectories over time. (b) Learning curves. (c) Learning curves.

Figure 4: Effectiveness of DHPG in recovering the minimal MDP from pixels. All methods are limited to a 4-dimensional latent space which is equal to the dimensions of the real state space of cartpole. **(a)** Trajectories of real states obtained from Mujoco and trajectories of latent states of DHPG. **(b, c)** Learning curves averaged on 10 seeds.

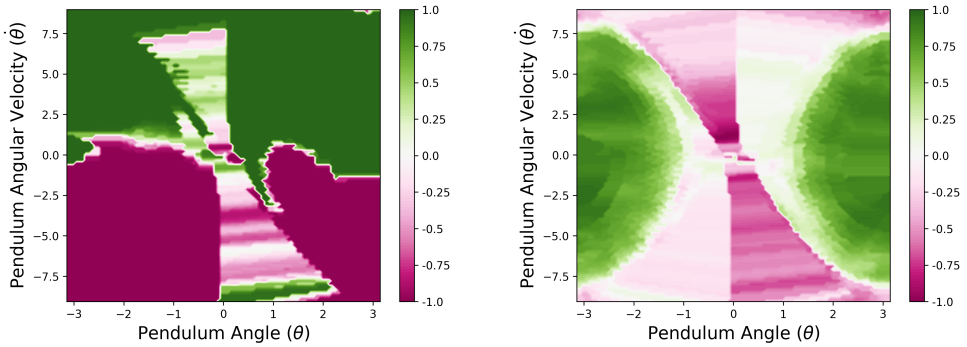
1. Does the homomorphic policy gradient improve policy optimization?
2. What are the qualitative properties of the learned representations and the abstract MDP?
3. Can DHPG learn and recover the minimal MDP image from raw pixel observations?

We evaluate DHPG on continuous control tasks from DM Control on pixel observations. Importantly, to reliably evaluate our algorithm against the baselines and to correctly capture the distribution of results, we follow the best practices proposed by Agarwal et al. (2021) and report the interquartile mean (IQM) and performance profiles aggregated on all tasks over 10 random seeds. While our baseline results are obtained using the official code, when possible[5], some of the results may differ from the originally reported ones due to the difference in the seed numbers and our goal to present a faithful representation of the true performance distribution (Agarwal et al., 2021).

7.1 DeepMind Control Suite

We compare the effectiveness of DHPG on pixel observations against DBC (Zhang et al., 2020), DeepMDP (Gelada et al., 2019), SAC-AE (Yarats et al., 2021b), and state-of-the-art performing DrQ-v2 (Yarats et al., 2021a). All methods use n-step returns, share the same hyperparameters in Appendix D.1 and all hyperparameters are adapted from DrQ-v2 *without any further tuning*. We acknowledge that since DrQ-v2 is based upon DDPG, the hyperparameters we used may be more advantageous to det erministic DHPG in comparison with stochastic DHPG. Importantly, for a fair comparison with DrQ-v2 which uses image augmentation, we present two variations of DHPG and other baselines, *with and without image augmentation*.

5. We use the official implementations of DrQv2, DBC, and SAC-AE, while we re-implement DeepMDP due to the unavailability of the official code. See Appendix D.2 for full details.



(a) Actual optimal policy.

(b) Abstract optimal policy.

Figure 5: Contours of actual and abstract optimal actions over the state space of the pendulum-swingup task. Colors represent action values, and states are $s = (\theta, \dot{\theta})$. **(a)** Actual optimal policy; contours of optimal actions $a^* = \pi^{\uparrow^*}(s)$. **(b)** Abstract optimal policy; contours of abstract optimal actions $\overline{a}^* = g_s(a^*) = \overline{\pi}^*(\overline{s})$. The relation $g_{s_1}(a_1) = g_{s_2}(a_2)$ holds for equivalent state-action pairs, and the abstract optimal policy is symmetric.

DHPG outperforms or matches other algorithms on pixel observations, demonstrating its effectiveness in representation learning. Results with image augmentation are presented in Figure 3 and *full results are in Appendix C.1.* Aggregated over 14 tasks, deterministic DHPG outperforms state-of-the-art DrQ-v2 and stochastic DHPG is as performant. Although these yield slight performance gains overall, the comparison in performance between DHPG and DrQ-v2 is highly task-dependent (see Figure 12 in Appendix C.1). For complex tasks such as Walker Run or Cheetah Run, DHPG obtains equal or slightly worse performance compared to DrQ-v2; however, on domains with clear symmetries—and hence easily learnable MDP homomorphism maps— DHPG outperforms DrQ-v2. In particular, DHPG without image augmentation outperforms DrQ-v2 on domains such as Cartpole and Pendulum, demonstrating its capability of representation learning.

Deterministic DHPG and stochastic DHPG have approximately similar sample efficiency, with deterministic DHPG being slightly better. As discussed in Section 5.3, deterministic policy gradient in theory is more sample efficient than the stochastic policy gradient, since it does not need to integrate over the action space (Silver et al., 2014). Additionally, due to the complications of lifting a stochastic policy, stochastic DHPG has more components which can negatively impact the learning performance. As a result, deterministic DHPG is slightly more sample efficient than stochastic DHPG.

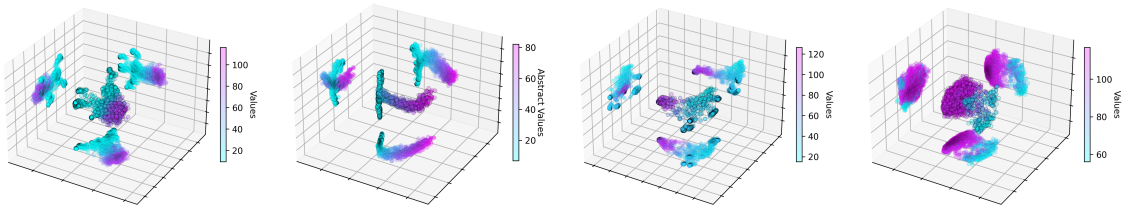
DHPG can learn and recover a low-dimensional MDP image. A key strength of MDP homomorphisms is their ability to represent the minimal MDP image (Ravindran and Barto, 2001), which is particularly important when learning from pixel observations. To demonstrate this ability, we have limited the latent space dimensions to the dimension of the real system and compared DHPG (without image augmentation) with baselines in Figure 4. While other methods are not able to learn the tasks, DHPG can successfully learn the policy and the minimal low-dimensional latent space. Surprisingly, trajectories of the latent states resemble that of the real states as shown in Figure 4a.

The learned mapping $h{=}(f, g_s)$ demonstrates properties of an MDP homomorphism. We use the pendulum swingup task to visualize its learned MDP homomorphism, as its symmetries are perfectly intelligible. Two state-action pairs $(s_1 = (\theta_1, \dot{\theta}_1), a_1)$ and $(s_2 = (\theta_2, \dot{\theta}_2), a_2)$ are equivalent if $a_1 = -a_2$, $\theta_1 = -\theta_2$, and $\dot{\theta}_1 = -\dot{\theta}_2$. Therefore, the learned action representations are expected to reflect this by setting $g_{s_1}(a_1) = g_{s_2}(a_2)$. Figure 5a shows contours of optimal actions over $\mathcal{S}$, while Figure 5b shows action representations $\overline{a} = g_s(a)$ of optimal actions over $\mathcal{S}$. Clearly, abstract actions adhere to the aforementioned relation for equivalent state-action pairs, indicating $g_s(a)$ is in fact representing the action encoder of an MDP homomorphism mapping.

The abstract MDP demonstrates properties of an MDP homomorphic image. To qualitatively demonstrate the significance of learning joint state-action representations, Figure 6 shows visualizations of latent states for quadruped-walk, a task with symmetries around movements of its four legs. Interestingly, while the latent space of DHPG (Figure 6a) shows distinct states for each leg, abstract state encoder f_ϕ has mapped corresponding legs (e.g., left forward leg and right backward leg) to the same abstract latent state (Figure 6b) as they are some homomorphic image of one another. Clearly, DBC and DrQ-v2 are not able to achieve this.

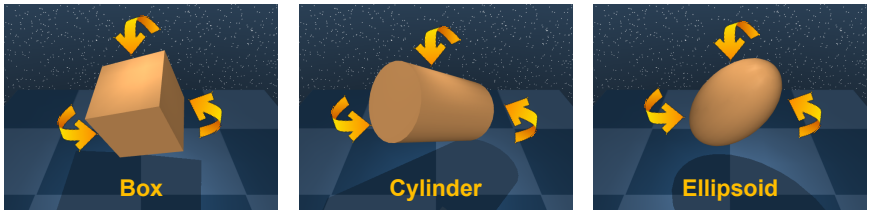
The learned representations and the MDP homomorphism map transfer to new tasks within the same domain. Importantly, one consideration with representation learning methods relying on rewards is the transferability of the learned representations to a new reward setting within the same domain. To ensure that our method does not hinder such transfer, we have carried out experiments in which the actor, critics, and the learned MDP homomorphism map are transferred to another task from the same domain. Results, given in Appendix C.3 show that our method has not compromised transfer abilities.

Additional Experiments. We study the value equivalence property as a measure for the quality of the learned MDP homomorphisms in Appendix C.2, and we present ablation studies on DHPG variants, and the impact of n-step return on our method in Appendices



(a) DHPG latent states. (b) DHPG abstract states. (c) DBC latent states. (d) DrQ-v2 latent states.

Figure 6: PCA projection of learned representations for quadruped-walk with pixel observations. **(a)** Latent states $s = E_\mu(o)$, **(b)** abstract latent states $\overline{s} = f_\phi(E_\mu(o))$ for DHPG, **(c)** latent states $s = E_\mu(o)$ for DBC, and **(d)** DrQ-v2. Color of each point denotes its value learned by $Q(s, a)$ or $\overline{Q}(\overline{s}, \overline{a})$. Points are also projected onto each main plane. The homomorphism map of DHPG has mapped the latent states of corresponding legs (e.g., left forward leg and right backward leg) **(a)** on to the same abstract latent states **(b)**, indicating a clear structure in $\overline{\mathcal{S}}$.



(a) Rotate Suite. (b) 3D mountain car.

Figure 7: Novel environments with symmetries. **(a)** Rotate Suite is a series of visual control tasks with the goal of rotating a 3D object along its axes to achieve a goal orientation. Symmetries of the environment are declared by symmetries of the object. **(b)** 3D mountain car has a continuous translational symmetry along the y-axis, shown as a dotted black line. The orange point represents the car and the green line represents the goal position.

C.4 and C.5, respectively. Finally, we compare the computation time of our method against the baselines in Appendix C.6.

7.2 Environments with Continuous Symmetries

As discussed in Section 5.3, the key difference between the deterministic and stochastic HPG theorems is that the former requires a bijective action encoder, whereas the latter does not impose any structure on it. The implication of such requirement is that deterministic DHPG is not capable of abstracting actions beyond relabling them. While relabling actions is sufficient for discrete symmetries, environments with continuous symmetries can in principle have their action dimensions reduced. To showcase the superiority of stochastic DHPG in action abstraction, we carry out experiments on a suite of novel environments with continuous symmetries. These environments are publicly available[6].

- **Rotate Suite** is a series of visual control tasks developed based on the DeepMind Control Suite. The goal in each environment is to rotate a 3D object along its axes to achieve a goal orientation. Symmetries of the environment are declared by symmetries of the object. Thus, the box rotation has discrete symmetries, while cylinder and ellipsoid rotation have continuous rotational symmetries. Figure 7a shows examples of these interactive environments.
- **3D Mountain Car** is an extension of the 2D mountain car problem (Moore, 1990) in which the mountain curve is extended along the y-axis, creating a mountain surface. The agent has a 2D action on the mountain surface, in contrast to the 1D action space of the 2D mountain car. As a result of this extension, the problem has a continuous translational symmetry along the y-axis and the action along side this axis is redundant. Figure 7b shows the surface of the 3D mountain car.

Stochastic DHPG outperforms deterministic DHPG in the environments with continuous symmetries. Results are presented in Figures 8 and 9. Notably, stochastic DHPG outperforms other baselines, as well as deterministic DHPG on environments with

6. `https://github.com/sahandrez/rotate_suite`

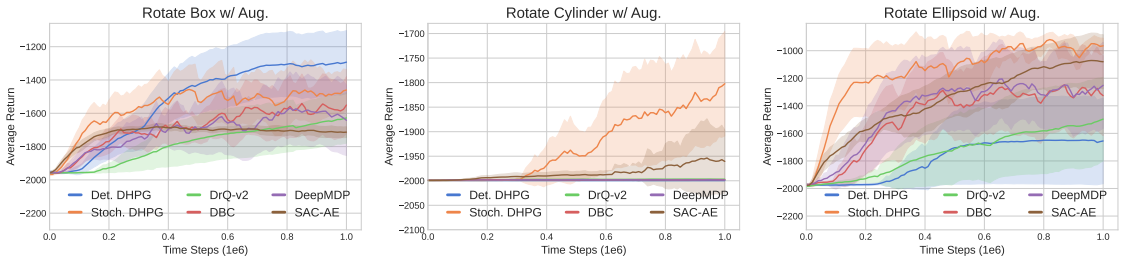


Figure 8: Learning curves for Rotate Suite environments obtained on 10 seeds. **(a)** Box rotation. **(b)** Cylinder rotation. **(c)** Ellipsoid rotation. Shaded regions represent 95% confidence intervals. Stochastic DHPG outperforms deterministic DHPG on environments with continuous symmetries (cylinder and ellipsoid rotation).

continuous symmetries. This is due to the fact that in theory, stochastic DHPG does not impose any structures on the action encoder and is therefore able to achieve higher levels of action abstraction, compared to deterministic DHPG.

DHPG is able to learn a structured latent space that adheres to the properties of a 3D rotation. Visualizations of latent state trajectories in the cylinder rotation task are presented in Figure 10. Each trajectory, color-coded by the action dimension, is collected by applying a constant rotation around one of the main axes (pitch, roll, and yaw). Interestingly, the latent trajectories of DHPG are fully disentangled and resemble 3D rotations in the latent space. However, none of the other baselines exhibits such structure in their latent representation.

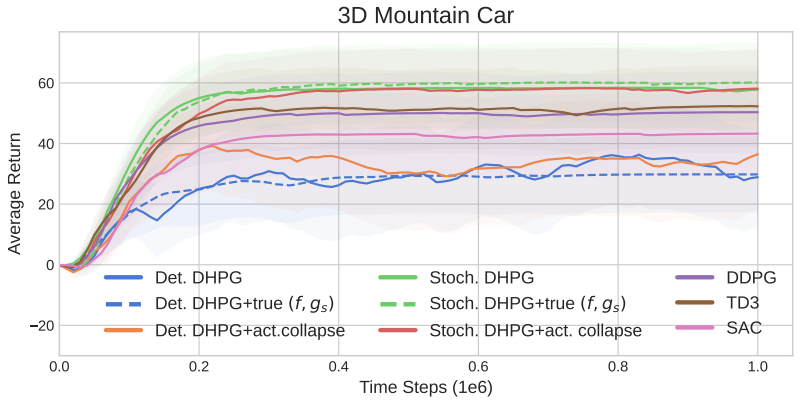
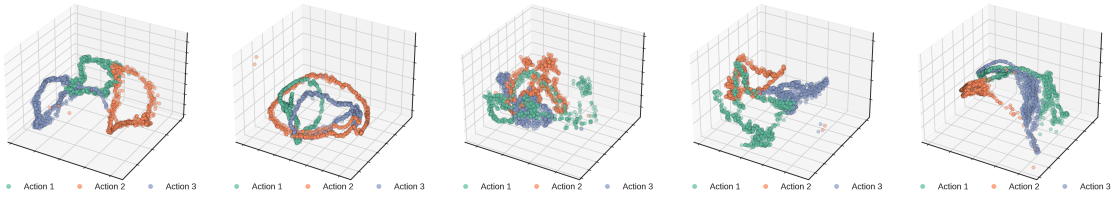


Figure 9: Learning curves for the 3D mountain car environment obtained on 50 seeds. Shaded regions represent 95% confidence intervals. Stochastic DHPG outperforms deterministic DHPG due to the continuous translational symmetry of the environment. DHPG+true (f, g_s) uses the ground truth homomorphism map, and DHPG+act. collapse removes has 1D abstract action space, as opposed to the original 2D action space.



(a) Stoch. DHPG. (b) Det. DHPG. (c) DrQ-v2. (d) DBC. (e) DeepMDP.

Figure 10: Visualization of latent state trajectories in the cylinder rotation task, color-coded by the action dimension. Action dimensions respectively correspond to *pitch*, *roll*, and *yaw* and each trajectory is collected by applying a constant rotation around a specific axis. Latent states are projected into a 3D space using PCA.

8. Conclusion

In this paper, we developed the novel theory of continuous MDP homomorphisms using measure theory, and we rigorously proved their value and optimal value equivalence properties. We derived the homomorphic policy gradient for both stochastic and deterministic policies, in order to directly use a joint state-action abstraction for policy optimization. Importantly, we rigorously proved that applying our homomorphic policy gradient on the abstract MDP is equivalent to applying the standard policy gradient on the actual MDP. Based on our novel theoretical results, we developed a family of deep actor-critic algorithms, with either stochastic or deterministic policies, that can simultaneously learn the policy and the MDP homomorphism map using the lax bisimulation metric. Our algorithm, referred to as Deep Homomorphic Policy Gradient (DHPG) improves upon strong baselines in challenging visual control problems. The visualization of the latent space demonstrates the strong potential of MDP homomorphisms in learning structured representations that can preserve value functions. Finally, we introduced a series of environments with continuous symmetries to further demonstrate the ability of our algorithm for action abstraction in the presence of continuous symmetries.

We believe that our work will open-up future possibilities for the application of MDP homomorphisms in challenging continuous control problems by enabling other RL algorithms to benefit from the abstraction power of MDP homomorphisms and the homomorphic policy gradient theorems.

Acknowledgments and Disclosure of Funding

SRS is supported by an NSERC CGS-D scholarship. RZ was supported by an NSERC CGS-M scholarship at the time this work was completed. PP is supported by a research grant from NSERC. The computing resources for this research were provided by Calcul Quebec and the Digital Research Alliance of Canada.

References

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *International Conference on Machine Learning*, pages 2915–2923. PMLR, 2016.

David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L Littman, and Lawson LS Wong. State abstraction as compression in apprenticeship learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3134–3142, 2019.

David Abel, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael Littman. Value preserving state-action abstractions. In *International Conference on Artificial Intelligence and Statistics*, pages 1639–1650. PMLR, 2020.

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.

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.

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in Atari. *Advances in Neural Information Processing Systems*, 32:8769–8782, 2019.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6): 26–38, 2017.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva Tb, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.

O Biza, R Platt, JW van de Meent, and L Wong. Learning discrete state abstractions with deep variational inference. *Advances in Approximate Bayesian Inference*, 2021.

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.

George W Bluman and Sukeyuki Kumei. *Symmetries and differential equations*, volume 81. Springer Science & Business Media, 2013.

R. Blute, J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. In *Proceedings of the Twelfth IEEE Symposium On Logic In Computer Science, Warsaw, Poland.*, 1997a.

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, 1997b.

Vladimir Igorevich Bogachev and Maria Aparecida Soares Ruas. *Measure theory*, volume 1. Springer, 2007.

T Butler, J Jakeman, and Tim Wildey. Combining push-forward measures and bayes' rule to construct consistent solutions to stochastic inverse problems. *SIAM Journal on Scientific Computing*, 40(2):A984–A1011, 2018.

Hugo Caselles-Dupré, Michael Garcia Ortiz, and David Filliat. Symmetry-based disentangled representation learning requires interaction with environments. *Advances in Neural Information Processing Systems*, 32:4606–4615, 2019.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10069–10076, 2020.

Pablo Samuel Castro and Doina Precup. Using bisimulation for policy transfer in MDPs. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Pablo Samuel Castro and Doina Precup. Automatic construction of temporally extended actions for MDPs using bisimulation metrics. In *European Workshop on Reinforcement Learning*, pages 140–152. Springer, 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.

Yash Chandak, Georgios Theocharous, James Kostas, Scott Jordan, and Philip Thomas. Learning action representations for reinforcement learning. In *International Conference on Machine Learning*, pages 941–950. PMLR, 2019.

Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pages 1321–1330. PMLR, 2019.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Nima Dehmamy, Robin Walters, Yanchen Liu, Dashun Wang, and Rose Yu. Automatic symmetry discovery with lie algebra convolutional network. *Advances in Neural Information Processing Systems*, 34:2503–2515, 2021.

J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labeled Markov systems. In *Proceedings of CONCUR99*, number 1664 in Lecture Notes in Computer Science. Springer-Verlag, 1999.

J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labeled Markov processes. *Information and Computation*, 179(2):163–193, Dec 2002.

Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

Stephan Eismann, Raphael JL Townshend, Nathaniel Thomas, Milind Jagota, Bowen Jing, and Ron O Dror. Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. *Proteins: Structure, Function, and Bioinformatics*, 89(5): 493–501, 2021.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. 2017.

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Animashree Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. In *International Conference on Machine Learning*, pages 3088–3099. PMLR, 2021.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision precesses. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 162–169, July 2004.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for Markov decision processes with infinite state spaces. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 201–208, July 2005a.

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, 2005b.

Norm Ferns, Pablo Samuel Castro, Doina Precup, and Prakash Panangaden. Methods for computing state similarity in Markov decision processes. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 174–181, 2006.

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

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, pages 3318–3328. PMLR, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi S Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. In *International Conference on Learning Representations*, 2021.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

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

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:5541–5552, 2020.

Christopher Grimm, André Barreto, Greg Farquhar, David Silver, and Satinder Singh. Proper value equivalence. *Advances in Neural Information Processing Systems*, 34:7773–7786, 2021.

David Ha and Jürgen Schmidhuber. World models. *arXiv e-prints*, pages arXiv–1803, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019a.

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*, pages 2555–2565. PMLR, 2019b.

Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021.

Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations*, 2020.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Irina Higgins, Peter Wirnsberger, Andrew Jaegle, and Aleksandar Botev. Symetric: Measuring the quality of learnt hamiltonian dynamics inferred from vision. *Advances in Neural Information Processing Systems*, 34, 2021.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Rico Jonschkowski and Oliver Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, 2015.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.

Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Mete Kemertas and Allan Douglas Jepson. Approximate policy iteration with bisimulation metrics. *Transactions on Machine Learning Research*, 2022.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31, 2018.

Serge Lang. *Differential and Riemannian manifolds*, volume 160. Springer Science & Business Media, 2012.

K. G. Larsen and A. Skou. Bisimulation through probablistic testing. *Information and Computation*, 94:1–28, 1991a.

Kim G Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and computation*, 94(1):1–28, 1991b.

Charline Le Lan, Marc G Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8261–8269, 2021.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for MDPs. *ISAIM*, 4:5, 2006.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR, 2021.

Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 1–9. PMLR, 2021.

Anuj Mahajan and Theja Tulabandhula. Symmetry learning for function approximation in reinforcement learning. *arXiv preprint arXiv:1706.02999*, 2017.

R. Milner. *A Calculus for Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

Robin Milner. *Communication and Concurrency*. Prentice-Hall, 1989a.

Robin Milner. *Communication and concurrency*, volume 84. Prentice hall Englewood Cliffs, 1989b.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Arnab Kumar Mondal, Vineet Jain, Kaleem Siddiqi, and Siamak Ravanbakhsh. Eqr: Equivariant representations for data-efficient reinforcement learning. In *International Conference on Machine Learning*, pages 15908–15926. PMLR, 2022.

Andrew William Moore. Efficient memory-based learning for robot control. Technical report, University of Cambridge, Computer Laboratory, 1990.

Emmy Noether. Invariant variation problems. *Transport theory and statistical physics*, 1 (3):186–207, 1971. English translation of the original German paper published in 1918.

David Park. Title unknown. Slides for Bad Honnef Workshop on Semantics of Concurrency, 1981.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan Willem van de Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models. *arXiv preprint arXiv:2204.11371*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Zhuoran Qiao, Anders S Christensen, Matthew Welborn, Frederick R Manby, Anima Anandkumar, and Thomas F Miller III. Unite: Unitary n-body tensor equivariant network with applications to quantum chemistry. *arXiv preprint arXiv:2105.14655*, 2021.

Robin Quessard, Thomas D Barrett, and William R Clements. Learning group structure and disentangled representations of dynamical environments. *arXiv preprint arXiv:2002.06991*, 2020.

Srividhya Rajendran and Manfred Huber. Learning to generalize and reuse skills using approximate partial policy homomorphisms. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 2239–2244. IEEE, 2009.

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

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

Balaraman Ravindran and Andrew G Barto. Relativized options: Choosing the right transformation. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 608–615, 2003.

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

Sahand Rezaei-Shoshtari, Rosie Zhao, Prakash Panangaden, David Meger, and Doina Precup. Continuous mdp homomorphisms and homomorphic policy gradient. In *Advances in Neural Information Processing Systems*, 2022.

Brian Sallans and Geoffrey E Hinton. Reinforcement learning with factored states and actions. *The Journal of Machine Learning Research*, 5:1063–1088, 2004.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Sahil Sharma, Aravind Suresh, Rahul Ramesh, and Balaraman Ravindran. Learning to factor policies and action-value functions: Factored action space representations for deep reinforcement learning. *arXiv preprint arXiv:1705.07269*, 2017.

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.

Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *5th Annual Conference on Robot Learning*, 2021.

Meir Smorodinsky. *Ergodic theory entropy*, volume 214. Springer, 2006.

Vishal Soni and Satinder Singh. Using homomorphisms to transfer options across continuous reinforcement learning domains. In *Proceedings of the 21st national conference on Artificial intelligence-Volume 1*, pages 494–499, 2006.

Jonathan Sorg and Satinder Singh. Transfer via soft homomorphisms. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 741–748, 2009.

Michael Spivak. *Calculus on manifolds: a modern approach to classical theorems of advanced calculus*. CRC press, 2018.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112 (1-2):181–211, 1999.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

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

Guy Tennenholtz and Shie Mannor. The natural language of actions. In *International Conference on Machine Learning*, pages 6196–6205. PMLR, 2019.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Valentin Thomas, Jules Pondard, Emmanuel Bengio, Marc Sarfati, Philippe Beaudoin, Marie-Jean Meurs, Joelle Pineau, Doina Precup, and Yoshua Bengio. Independently controllable factors. *arXiv preprint arXiv:1708.01289*, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

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.

Elise van der Pol, Herke van Hoof, Frans A Oliehoek, and Max Welling. Multi-agent mdp homomorphic networks. In *International Conference on Learning Representations*, 2021.

Veeravalli S Varadarajan. Groups of automorphisms of borel spaces. *Transactions of the American Mathematical Society*, 109(2):191–220, 1963.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018.

Dian Wang, Robin Walters, and Robert Platt. So(2)-equivariant reinforcement learning. In *International Conference on Learning Representations*, 2021.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

William Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. In *International Conference on Learning Representations*, 2019.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Marysia Winkels and Taco S Cohen. Pulmonary nodule detection in ct scans with equivariant cnns. *Medical image analysis*, 55:15–26, 2019.

Alicia P Wolfe and Andrew G Barto. Decision tree methods for finding reusable mdp homomorphisms. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 21, page 530. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006a.

Alicia Peregrin Wolfe and Andrew G Barto. Defining object types and options using mdp homomorphisms. In *Proceedings of the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning*. Citeseer, 2006b.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2020.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2021a.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021b.

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.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3*, pages 1433–1438, 2008.

A. Assumptions and Conditions

The derivation of our homomorphic policy gradient theorem is for continuous state and action spaces. Therefore, we have assumed the following regularity conditions on the actual MDP $\mathcal{M}$ and its MDP homomorphic image $\overline{\mathcal{M}}$ under the MDP homomorphism map h. The conditions are largely based on the regularity conditions of the deterministic policy gradient theorem (Silver et al., 2014):

Regularity conditions 1: $\tau_a(s'|s)$, $\nabla_a \tau_a(s'|s)$, $\overline{\tau}_{\overline{a}}(\overline{s}'|\overline{s})$, $\nabla_{\overline{a}} \overline{\tau}_{\overline{a}}(\overline{s}'|\overline{s})$, $R(s,a), \nabla_a R(s,a)$, $\overline{R}(\overline{s},\overline{a}), \nabla_{\overline{a}} \overline{R}(\overline{s},\overline{a})$, $\pi_\theta^\uparrow(s), \nabla_\theta \pi_\theta^\uparrow(s), \overline{\pi}_\theta(\overline{s})$, $\nabla_\theta \overline{\pi}_\theta(\overline{s})$, $p_1(s)$, and $\overline{p}_1(\overline{s})$ are continuous with respect to all parameters and variables $s, \overline{s}, a, \overline{a}, s'$, and $\overline{s}'$.

Regularity conditions 2: There exists a b and L such that $\sup_s p_1(s) < b$, $\sup_{\overline{s}} \overline{p}_1(\overline{s}) < b$, $\sup_{a,s,s'} \tau_a(s'|s) < b$, $\sup_{\overline{a},\overline{s},\overline{s}'} \overline{\tau}_{\overline{a}}(\overline{s}'|\overline{s}) < b$, $\sup_{a,s} R(s,a) < b$, $\sup_{\overline{a},\overline{s}} \overline{R}(\overline{s},\overline{a}) < b$, $\sup_{a,s,s'} \|\nabla_a \tau_a(s'|s)\| < L$, $\sup_{\overline{a},\overline{s},\overline{s}'} \|\nabla_{\overline{a}} \overline{\tau}_{\overline{a}}(\overline{s}'|\overline{s})\| < L$, $\sup_{s,a} \|\nabla_a R(s,a)\| < L$, $\sup_{\overline{s},\overline{a}} \|\nabla_{\overline{a}} \overline{R}(\overline{s},\overline{a})\| < L$.

Regularity conditions 3: The action mapping $g_s(a)$ is a local diffeomorphism (Definition 30). Hence it is continuous with respect to a and locally bijective with respect to a. Additionally, $\nabla_a g_s(a)$ is continuous with respect to the parameter a, and there exists a L such that $\sup_{s,a} \|\nabla_a g_s(a)\| < L$.

B. Mathematical tools

Various mathematical concepts from measure theory and differential geometry are briefly presented in this section. We only explicitly introduce concepts which are directly mentioned or relevant to the proofs presented in Sections 4 and 5; for a more comprehensive overview, we direct the reader to textbooks such as Bogachev and Ruas (2007); Lang (2012); Spivak (2018).

B.1 Metric spaces and topology

A set equipped with a metric is called a metric space and is usually written as a pair, typically (X, d). Given a metric one can define standard notions from basic analysis: convergent sequence, limit of a sequence, Cauchy sequence and continuous function. If every Cauchy sequence converges we say the metric space is *complete*. Limits of convergent sequences are unique in proper metric spaces but not in pseudometric spaces. A function $f : (X, d) \to (Y, d')$ between metric spaces is said to be *nonexpansive* if:

$$\forall x, x' \in X, d'(f(x), f(x')) \leq d(x, x').$$

A function is said to be *contractive* if there is some number $c \in (0, 1)$ such that:

$$\forall x, x' \in X, d'(f(x), f(x')) < c \cdot d(x, x').$$

The fundamental theorem about metric spaces, called the *Banach fixed-point theorem*, states the following:

Theorem 20 *If f is a contractive function from a complete metric space X to itself then there is a* unique *fixed point for f, i.e. a unique point $x_0 \in X$ such that $f(x_0) = x_0$.*

We assume that the readers are familiar with basic concepts of topology: open and closed sets, base and subbase, convergence of sequences and continuity of functions.

A topological space is *completely metrizable* if it can be equipped with a metric which generates its topology and the resulting metric space is complete. A topological space is *separable* if it contains a countable, dense subset— that is, every nonempty subset in the topological space contains at least one element of this subset. A *Polish space* is a topological space that is separable and completely metrizable. Polish spaces have "desirable properties" and are used primarily in areas of descriptive set theory and measure theory.

Another fundamental concept of topological spaces is *compactness*. An *open cover* of a topological space X is a family of open subsets whose union includes all of X. A *subcover* of a cover is a subcollection of the open sets in the cover that also covers X. A topological space is said to be *compact* if every open cover has a finite subcover. In metric spaces, the compact sets are exactly the closed and bounded sets. A space is said to be *locally compact* if every point is contained in an open set that is contained in a compact set.

B.2 Measure theory

Measure theory attempts to generalize notions of length, area and volume or mass to more complicated subsets than the simple ones that one first encounters in geometry. In common situations, like the real line, it is not possible to define a measure on *all* subsets in such a way that one's normal intuitions of length survive. In the real line there is no measure that is defined on all subsets and which assigns to all intervals its length. One needs, therefore, to choose well behaved families of subsets on which one can define measures.

Definition 21 (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.*

A set equipped with a σ-algebra is called a *measurable* space. The σ-algebra of a space specifies the sets for which a measure can be defined; in probability theory—and in our use case—a σ-algebra represents a collection of events which can be assigned probabilities.

Given any family of subsets there is a *smallest* σ-algebra that includes the given family: this is called the σ-algebra generated by the family. In metric spaces the σ-algebra generated by the open sets is called the *Borel* σ-algebra.

Given a σ-algebra one can define a measure.

Definition 22 *A (probability)* measure μ *on a σ-algebra Σ defined on X is a function $\mu : \Sigma \to [0, \infty]$ ($\mu : \Sigma \to [0, 1]$) satisfying:*

- $\mu(\varnothing) = 0$
- *If $\{A_i\}_{i\in\mathbb{N}}$ is a countable family of pairwise disjoint subsets in Σ, then $\mu(\cup_i A_i) = \sum_{i\in\mathbb{N}} \mu(A_i)$.*

For a probability measure we require $\mu(X) = 1$.

The functions that play a key role are called measurable functions.

Definition 23 *A* measurable *function or map between two measurable spaces* $(X\Sigma)$ *and* (Y, Λ) *is a function* $f : X \to Y$ *such that for any* $B \in \Lambda$, $f^{-1}(B) \in \Sigma$.

We can define measures on the image of a measurable function based on a measure in the preimage. This yields the definition of the pushforward measure and the change of variables formula which is crucial for our proofs in switching the domain of integration across a measurable function.

Definition 24 (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 25 (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:*

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

B.3 Manifolds

Differential manifolds formalize doing differential calculus on curved surfaces. Unlike vector spaces, we cannot define addition of points in an arbitrary topological space; we need additional structure. Hence the strategy is to define "patches" of the topological space that "look like" patches of a vector space and then glue them together. This motivates the definition of a differential or smooth manifold. The word "smooth" is a synonym for *infinitely differentiable* or C^∞.

Definition 26 *An* n*-dimensional* ***smooth (or differential) manifold*** *is a topological space* M[7] *equipped with a family of pairs, called* ***charts****,* $\{(U_i, \phi_i) | i \in \mathcal{A}\}$ *where:*

- *Each* U_i *is an open subset of* M,
- *each* $\phi_i : U_i \to \mathbf{R}^n$ *is a* homeomorphism *between* U_i *and the image* $V_i := \phi_i(U_i)$,
- *the* $\{U_i\}$ *form an open cover of* M.

In addition, the following compatibility condition *must be satisfied:*
if $U_i \cap U_j \neq \varnothing$ *then the map*

$$\phi_j \circ \phi_i^{-1}\Big|_{\phi_i(U_i \cap U_j)} : \phi_i(U_i \cap U_j) \to \phi_j(U_i \cap U_j)$$

is infinitely differentiable, written as C^∞. *A collection of compatible charts is called an* ***atlas****.*

7. Assumed to be paracompact and Hausdorff.

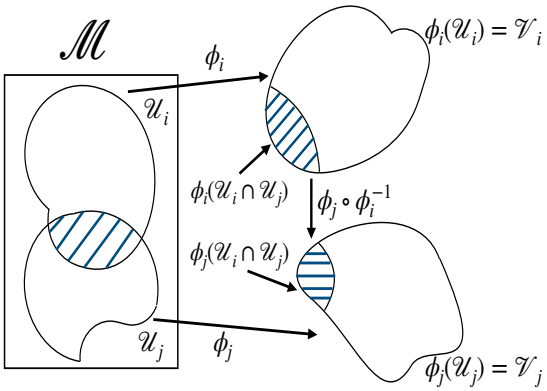


Figure 11: The compatibility condition on charts.

Note that the very last condition refers to a map between an open set in $\mathbf{R}^n$ and another open set in $\mathbf{R}^n$; hence its meaning is clear since $\mathbf{R}^n$ is a vector space. The picture in Fig. 11 illustrates the meaning of this condition. Now everything can be defined in terms of charts.

Definition 27 *A* smooth function $f : M \to \mathbf{R}$ *is a function such that for any chart* (U, ϕ) *the map* $f \circ \phi^{-1} : \mathbf{R}^n \to \mathbf{R}$[8] *is smooth.*

A smooth map between manifolds M and M' can be defined similarly.

Definition 28 *A* smooth map $f : M \to M'$ *is a function such that for any chart* (U, ϕ) *of* M *and* (U', ϕ') *of* M' *the map* $\phi' \circ \psi \circ \phi^{-1} : \mathbf{R}^n \to \mathbf{R}^{n'}$ *is smooth.*

Smooth maps and smooth functions are automatically continuous.

Differential manifolds come with a notion of isomorphism called a *diffeomorphism.*

Definition 29 *A smooth map between two manifolds is called a* diffeomorphism *if it is a bijection and the inverse map is also smooth.*

Definition 30 (Local diffeomorphism) *Let* M *and* N *be differentiable manifolds. A function* $f : M \to N$ *is a* local diffeomorphism*, if for each point* $x \in M$ *there exists an open set* U *containing* x *such that* $f(U)$ *is open in* N *and* $f|_U : U \to f(U)$ *is a diffeomorphism.*

Once the structure of a smooth manifold is in place one can define the notion of derivative operator. The tangent to a curve can also be defined in terms of differentiation. A tangent vector t at a point x should be thought of as a directional derivative. Standard results from multivariable calculus can be invoked to show that the set of tangent vectors at a point form an n-dimensional vector space. One writes T_x for this vector space, which is called *the tangent space* at x.

The cleanest way to axiomatize the concept of tangent vector at a point x is as follows. Let $\mathcal{F}$ be the set of smooth real-valued functions defined on M.

8. Actually this is only defined on $\phi(U)$ not all of $\mathbf{R}^n$ but it would clutter the notation too much to constantly put in the correct restrictions.

Definition 31 *Given a point x of M we define a* ***tangent vector at*** *x to be a map $t : \mathcal{F} \to \mathbf{R}$ such that, $\forall a, b \in \mathbf{R}$ and $\forall f, g \in \mathcal{F}$:*

1. $t(af + bg) = at(f) + bt(g)$,
2. $t(fg) = f(x)t(g) + g(x)t(f)$.

It follows immediately that if f is a constant function then $t(f) = 0$. Note how the second condition makes specific reference to the point x.

A smooth map $\psi : M \to N$ induces a map between tangent spaces at the corresponding points. The *differential* of the map ψ at x is the linear map $\mathrm{d}\psi : T_x M \to T_{\psi(x)} N$ defined as follows. Let g be a smooth function on a neighbourhood of $\psi(x)$ and let t be a tangent vector at x. We have to define a tangent vector at $\psi(x)$ so it should be able to act on g. We define

$$\mathrm{d}\psi(t)(g) \coloneqq t(g \circ \psi).$$

The following theorems are fundamental and used in the proofs of the policy gradient theorems.

Theorem 32 (Inverse function theorem for manifolds) *If $f : M \to N$ is a smooth map whose differential $df_x : T_x M \to T_{f(x)} N$ is an isomorphism at a point $x \in M$. Then f is a local diffeomorphism at x.*

Theorem 33 (Chain rule for manifolds) *If $f : M \to N$ and $g : N \to O$ are smooth maps of manifolds, then:*

$$d(g \circ f)_x = dg_{f(x)} \circ df_x.$$

C. Full Results

As discussed in Section 7, we evaluate DHPG on continuous control tasks from DM Control on pixel observations, as well as custom designed environments. Importantly, to reliably evaluate our algorithm against the baselines and to correctly capture the distribution of results, we follow the best practices proposed by Agarwal et al. (2021) and report the interquartile mean (IQM) and performance profiles aggregated on all tasks over 10 random seeds. While our baseline results are obtained using the official code, when possible, some of the results may differ from the originally reported ones due to the difference in the seed numbers and our goal to present a faithful representation of the true performance distribution (Agarwal et al., 2021).

We use the official implementations of DrQv2, DBC, and SAC-AE, while we re-implement DeepMDP due to the unavailability of the official code; See Appendix D.2 for more details on the baselines.

C.1 DeepMind Control Suite

Figures 12-13 show full results obtained on 16 DeepMind Control Suite tasks with pixel observations to supplement the results of Section 7.1. Domains that require excessive exploration and large number of time steps (e.g., acrobat, swimmer, and humanoid) and domains with visually small targets (e.g., reacher hard and finger turn hard) are not included in this benchmark.

Figures 14-15 and 16-17 respectively show performance profiles and aggregate metrics (Agarwal et al., 2021) on 14 tasks; hopper hop and walker run are removed from RLiable evaluation as none of the algorithms have acquired reasonable performance within 1 million time-steps.

In Figures 12, 14, and 16 all methods are *with* image augmentation, while in Figures 13, 15, and 17 all methods are *without* image augmentation.

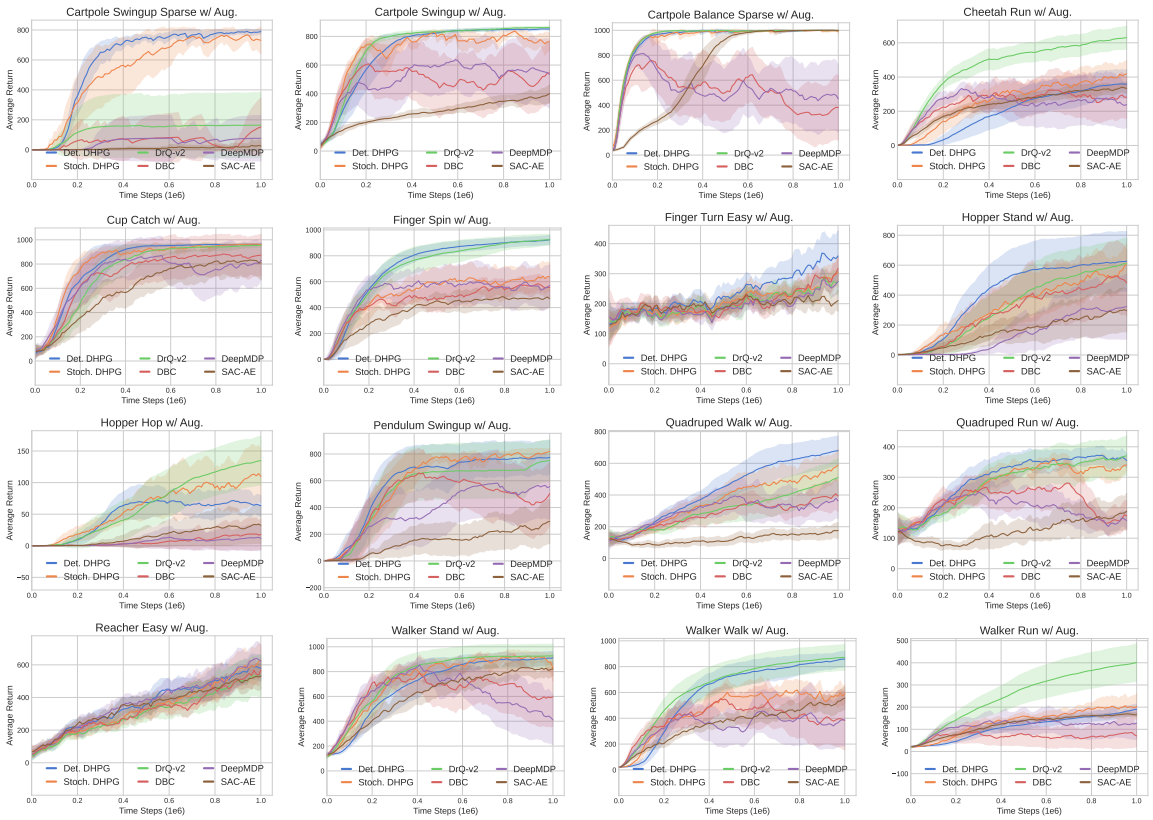


Figure 12: Learning curves for 16 DM control tasks with pixel observations. All methods are **with** image augmentation. Mean performance is obtained over 10 seeds and shaded regions represent 95% confidence intervals. Plots are smoothed uniformly for visual clarity.

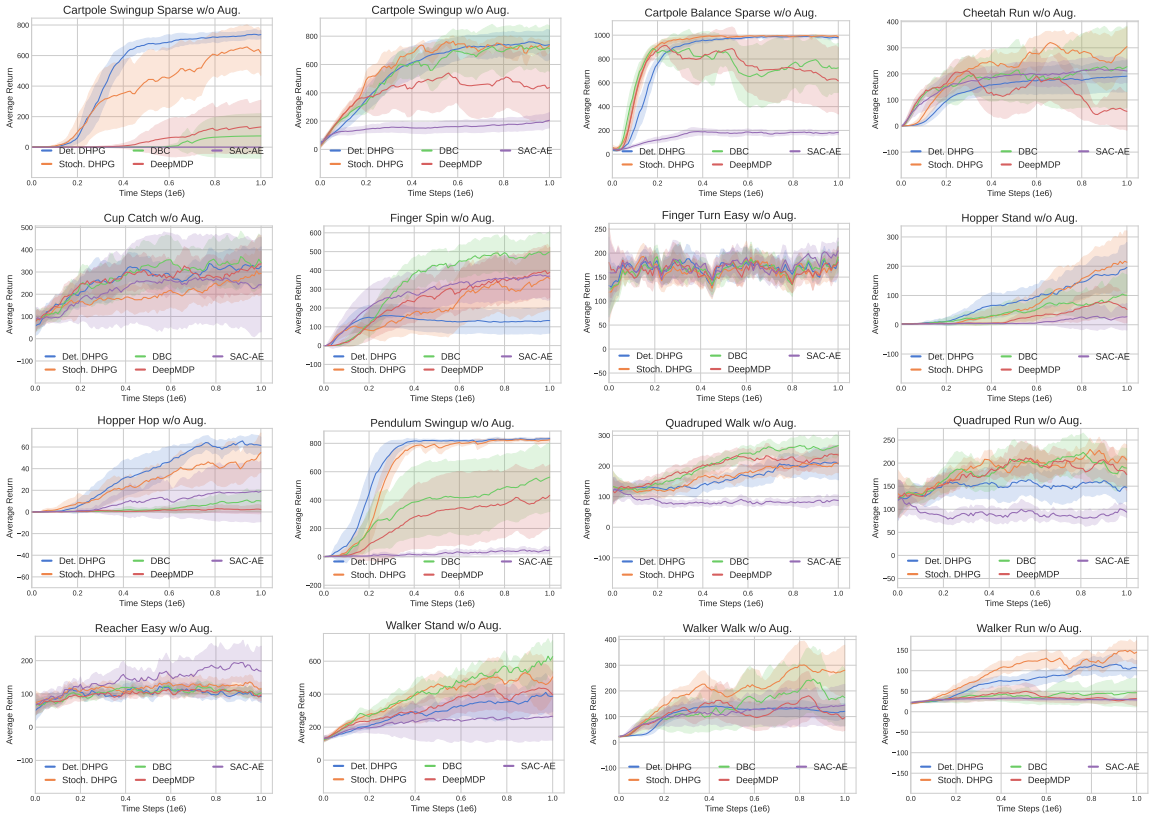
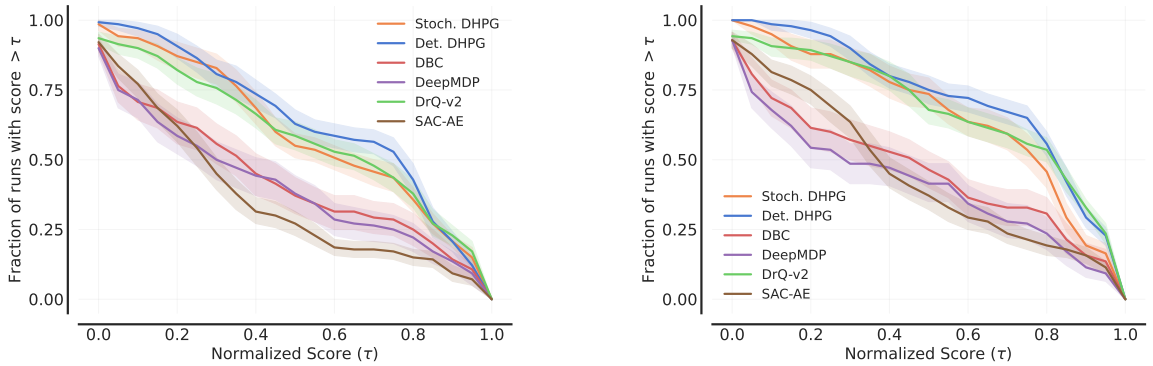


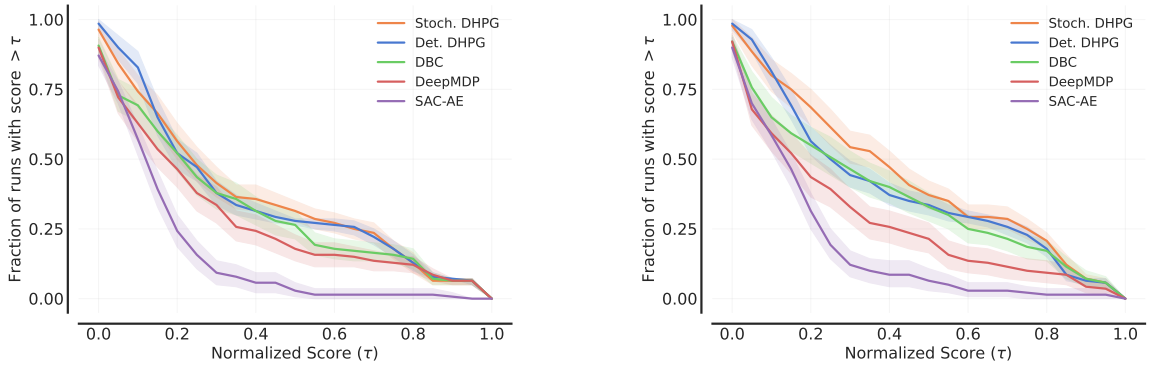
Figure 13: Learning curves for 16 DM control tasks with pixel observations. All methods are **without** image augmentation. Mean performance is obtained over 10 seeds and shaded regions represent 95% confidence intervals. Plots are smoothed uniformly for visual clarity.



(a) 500k step benchmark.

(b) 1m step benchmark.

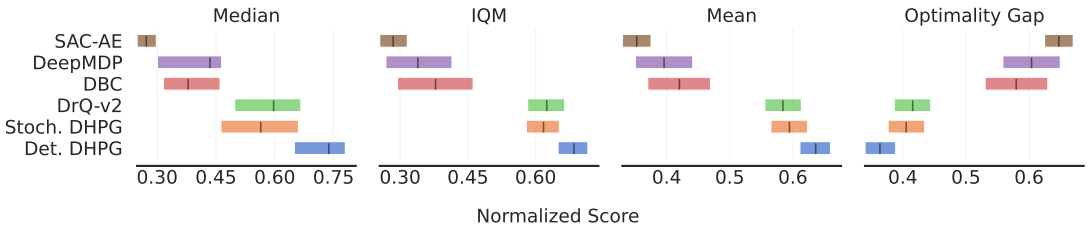
Figure 14: Performance profiles for pixel observations based on 14 tasks over 10 seeds, at 500k steps **(a)**, and at 1m steps **(b)**. All methods are **with** image augmentation. Shaded regions represent 95% confidence intervals.



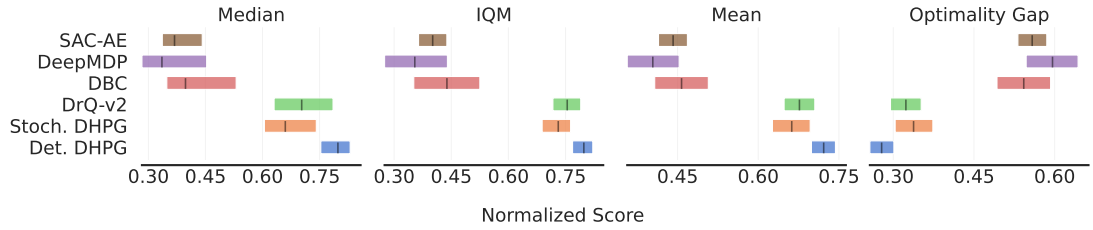
(a) 500k step benchmark.

(b) 1m step benchmark.

Figure 15: Performance profiles for pixel observations based on 14 tasks over 10 seeds, at 500k steps **(a)**, and at 1m steps **(b)**. All methods are **without** image augmentation. Shaded regions represent 95% confidence intervals.

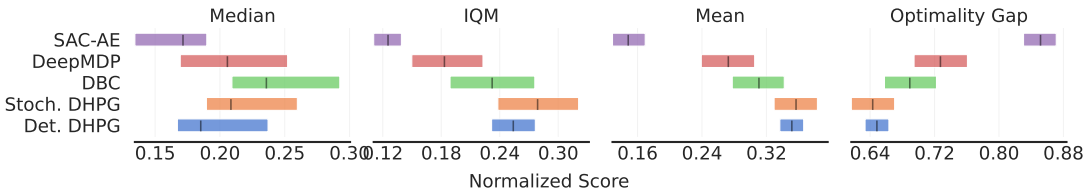


(a) 500k step benchmark.

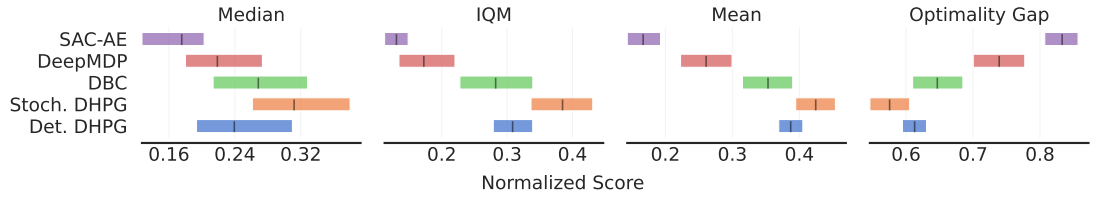


(b) 1m step benchmark.

Figure 16: Aggregate metrics for pixel observations with 95% confidence intervals based on 14 tasks over 10 seeds, at 500k steps **(a)**, and at 1m steps **(b)**. All methods are **with** image augmentation.



(a) 500k step benchmark.

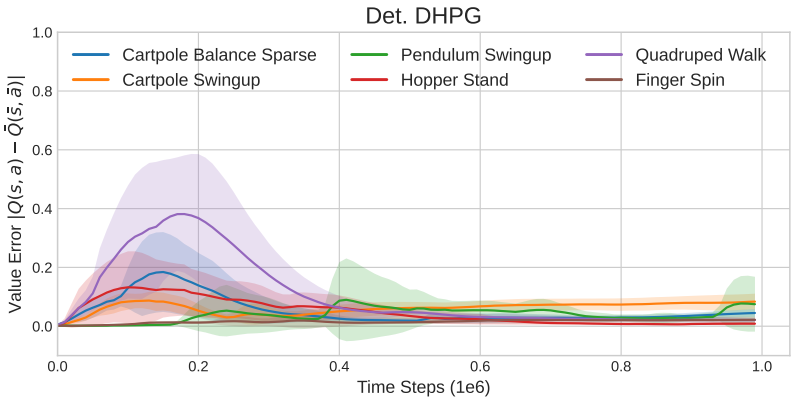


(b) 1m step benchmark.

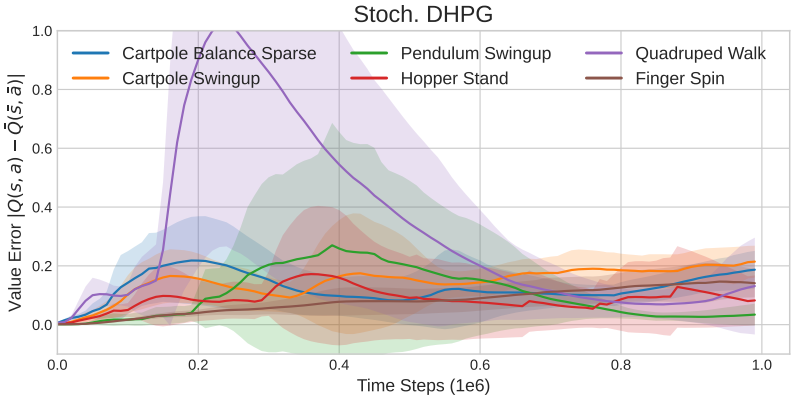
Figure 17: Aggregate metrics for pixel observations with 95% confidence intervals based on 14 tasks over 10 seeds, at 500k steps **(a)**, and at 1m steps **(b)**. All methods are **without** image augmentation.

C.2 Value Equivalence Property in Practice

We can use the value equivalence between the critics of the actual and abstract MDPs as a measure for the quality of learned MDP homomorphisms, since the two critics are not directly trained to minimize this distance, instead they have equivalent values through the learned MDP homomorphism map. Figure 18 shows the normalized mean absolute error of $|Q(s,a)-\overline{Q}(\overline{s},\overline{a})|$ during training, indicating the property is holding in practice. Expectedly, for lower-dimensional tasks with easily learnable homomorphism maps (e.g., cartpole) the error is reduced earlier than more complicated tasks (e.g., quadruped). But importantly, in all cases the error decreases over time and is at a reasonable range towards the end of the training, meaning the continuous MDP homomorphisms is adhering to conditions of Definition 11.



(a) Value equivalence for deterministic DHPG.

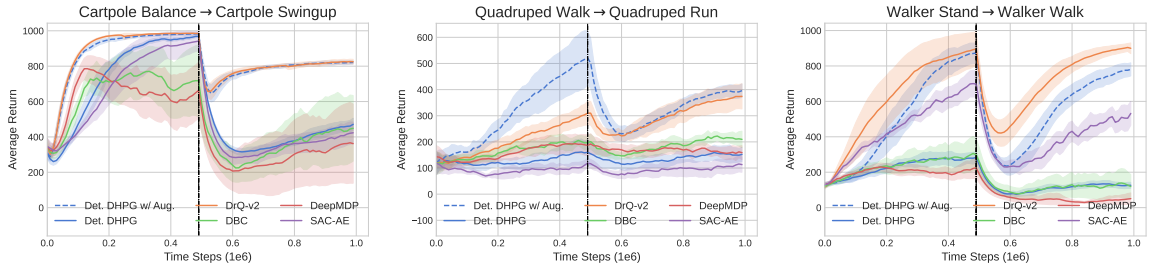


(b) Value equivalence for stochastic DHPG.

Figure 18: Normalized mean absolute error $|Q(s,a) - \overline{Q}(\overline{s},\overline{a})|$ as a measure for the value equivalence property during training of different tasks from pixel observations. The error is measured on samples from the replay buffer and is normalized by the range of the value function. The error is averaged over 10 seeds and shaded regions represent 95% confidence intervals.

C.3 Transfer Learning Experiments

As discussed in Section 7, the purpose of transfer experiments is to ensure that using MDP homomorphisms does not compromise transfer abilities. We select the deterministic DHPG for these experiments. Figure 19 shows learning curves for a series of transfer scenarios in which the critic, actor, and representations are transferred to a new task within the same domain. DHPG matches the same transfer abilities of other methods.



(a) Cartpole. (b) Quadruped. (c) Walker.

Figure 19: Learning curves for transfer experiments with pixel observations. At 500k time step mark, all components are transferred to a new task on the same domain. Mean performance is obtained over 10 seeds and shaded regions represent 95% confidence intervals. Plots are smoothed uniformly for visual clarity.

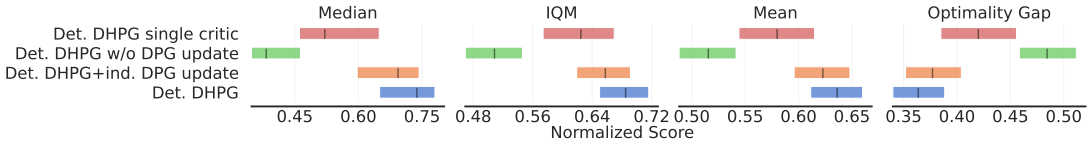
C.4 Ablation Study on the Combination of HPG with the Standard Policy Gradient

We carry out an ablation study on the combination of HPG with the standard policy gradient for actor updates. Since the deterministic DHPG algorithm is generally simpler as the lifted policy can be analytically obtained, we evaluate the performance of four variants of the deterministic DHPG (all variants use image augmentation):

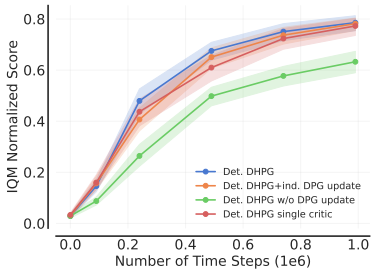
1. **DHPG:** Gradients of HPG and DPG are added together and a single actor update is done based on the sum of gradients. This is the deterministic DHPG algorithm that is used throughout the paper.
2. **DHPG with independent DPG update:** Gradients of HPG and DPG are independently used to update the actor.
3. **DHPG without DPG update:** Only HPG is used to update the actor.
4. **DHPG with single critic:** A single critic network is trained for learning values of both the actual and abstract MDP. Consequently, HPG and DPG are used to update the actor.

Figure 21 shows learning curves obtained on 16 DeepMind Control Suite tasks with pixel observations, and Figure 20 shows RLiable (Agarwal et al., 2021) evaluation metrics. In general, summing the gradients of HPG and DPG (variant 1) results in lower variance of gradient estimates compared to independent HPG and DPG updates (variant 2). Interestingly, the variant of DHPG without DPG (variant 3) performs reasonably well or even outperforms other variants in simple tasks where learning MDP homomorphisms is easy (e.g., cartpole and pendulum), indicating the effectiveness of our method in using **only**

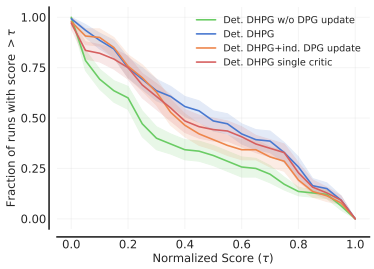
the abstract MDP to update the policy of the actual MDP. However, in the case of more complicated tasks (e.g., walker), DPG is required to additionally use the actual MDP for policy optimization. Finally, using a single critic for both the actual and abstract MDPs (variant 4) can improve sample efficiency in symmetrical MDPs, but may result in performance drops in non-symmetrical MDPs due to the large error bound between the two MDPs, $\|Q^{\pi^\uparrow}(s,a) - Q^{\overline{\pi}}(\overline{s},\overline{a})\|$ (Taylor et al., 2008).



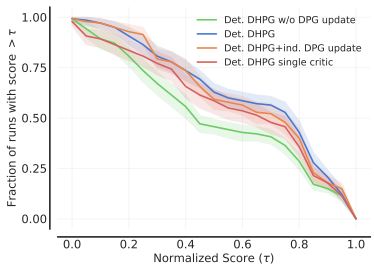
(a) Aggregate metrics at 500k.



(b) Sample efficiency.



(c) Performance profiles at 250k.



(d) Performance profiles at 500k.

Figure 20: Ablation study on the combination of HPG and DPG. RLiable evaluation metrics for pixel observations averaged on 14 tasks over 10 seeds. Aggregate metrics at 500k steps **(a)**, IQM scores as a function of number of steps for comparing sample efficiency **(b)**, performance profiles at 250k steps **(c)**, performance profiles at 500k steps **(d)**. Shaded regions represent 95% confidence intervals.

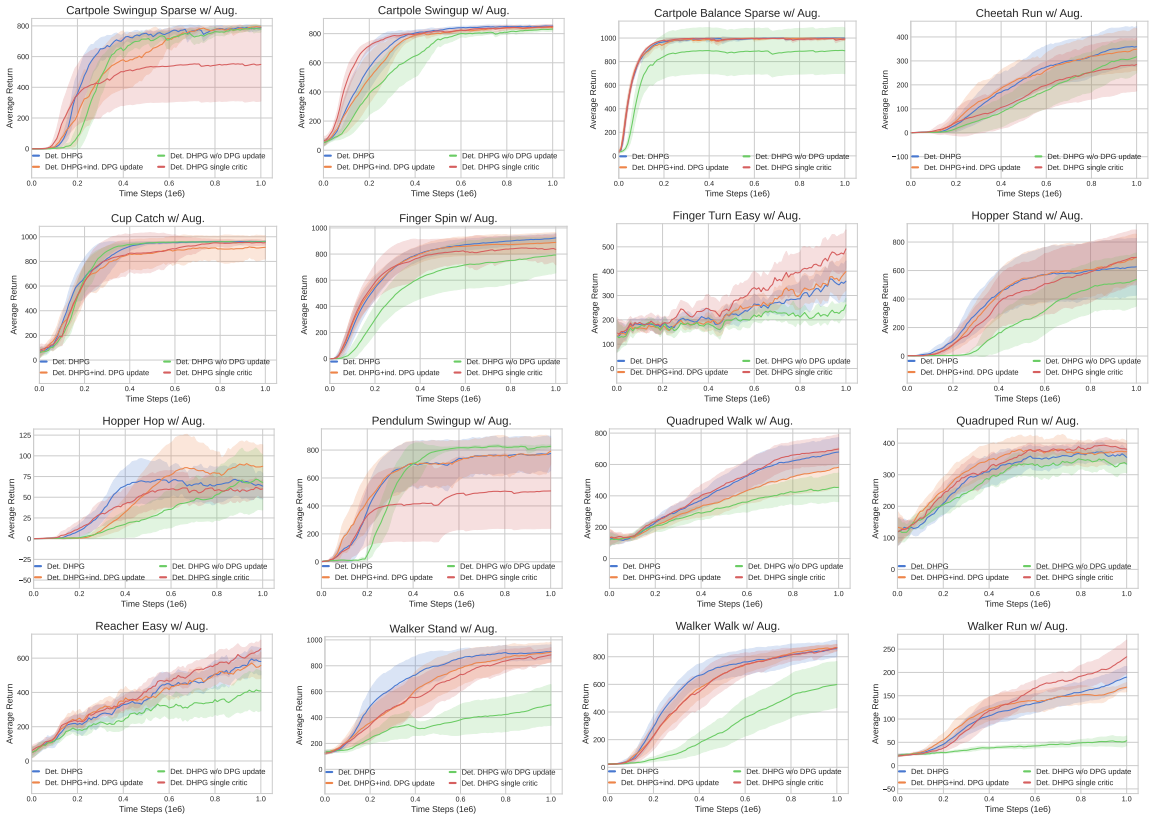
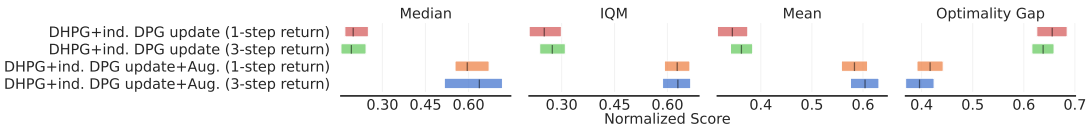


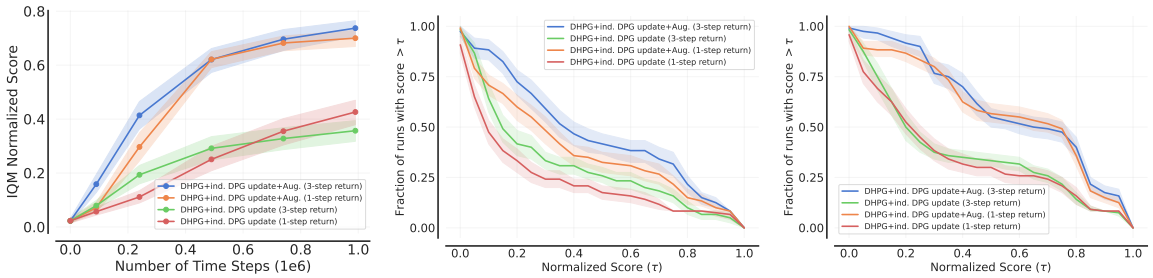
Figure 21: Ablation study on the combination of HPG and DPG. Learning curves for 16 DM control tasks with pixel observations. Mean performance is obtained over 10 seeds and shaded regions represent 95% confidence intervals. Plots are smoothed uniformly for visual clarity.

C.5 Ablation Study on n-step Return

We carry out an ablation study on the choice of n-step return for DHPG. Similarly to the ablation study in Appendix C.4, we select the deterministic DHPG since it is generally simpler and its lifted policy can be analytically obtained. Figure 22 shows RLiable (Agarwal et al., 2021) evaluation metrics for DHPG with 1-step and 3-step returns for pixel observations. We show the impact of n-step return on DHPG with and without image augmentation. Overall, n-step return appears to improve the early stages of training. In the case of DHPG without image augmentation, the final performance of 1-step return is better than 3-step return, perhaps indicating that using n-step return can render learning MDP homomorphisms more difficult.



(a) Aggregate metrics at 500k steps.



(b) Sample efficiency. (c) Performance profiles at 250k. (d) Performance profiles at 500k.

Figure 22: Ablation study on n-step return. RLiable evaluation metrics for pixel observations averaged on 12 tasks over 10 seeds. Aggregate metrics at 1m steps **(a)**, IQM scores as a function of number of steps for comparing sample efficiency **(b)**, performance profiles at 250k steps **(c)**, and performance profiles at 500k steps **(d)**. Shaded regions represent 95% confidence intervals.

C.6 Comparison of Computation Time

Figure 23 compares the computation cost of our method against the baselines. The horizontal axis represents wall clock time in hours. Since our method does not require image reconstruction, it is more computationally efficient than SAC-AE and DeepMDP. However, the bisimulation computation, the HPG update, and the policy lifting loss (in the case of stochastic DHPG) increase the computation costs of our method in comparison to DrQ-v2.

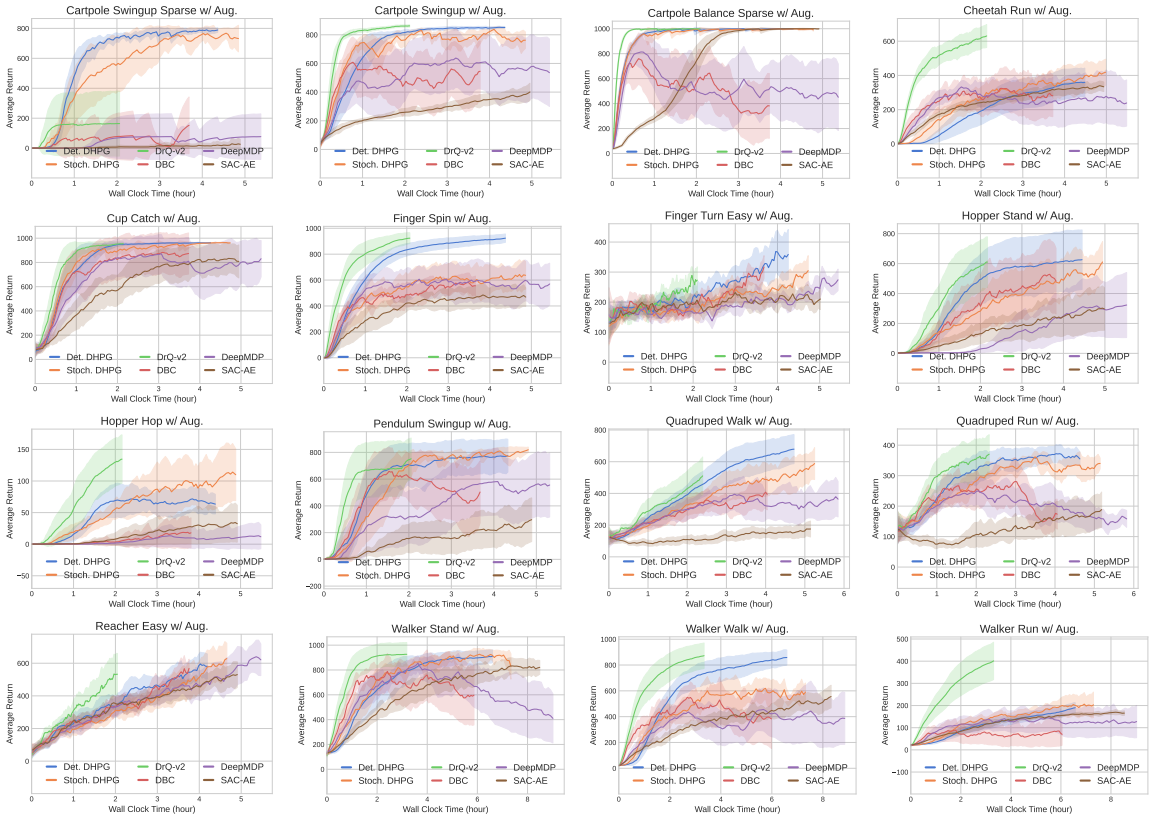


Figure 23: Learning curves for 16 DM control tasks with pixel observations. The horizontal axis is the **wall clock time** in hours. All methods are **with** image augmentation. Mean performance is obtained over 10 seeds and shaded regions represent 95% confidence intervals. Plots are smoothed uniformly for visual clarity.

D. Implementation Details

D.1 Hyperparameters

Algorithm 1 in Section 6 presents the pseudo-code of stochastic and deterministic DHPG algorithms for pixel observations.

In the image augmentation version of DHPG, as well as all the baselines, we use image augmentation of DrQ (Yarats et al., 2020) that simply applies random shifts to pixel observations. First, 84×84 images are padded by 4 pixels (by repeating boundary pixels), and then a random 84×84 crop is selected, rendering the original image shifted by ± 4 pixels. Similarly to Yarats et al. (2021a), we also apply bilinear interpolation on top of the shifted image by replacing each pixel value with the average of four nearest pixel values. Our code is publicly available at `https://github.com/sahandrez/homomorphic_policy_gradient`.

We implemented our method in PyTorch (Paszke et al., 2019) and results were obtained using Python v3.8.10, PyTorch v1.10.0, CUDA 11.4, and Mujoco 2.1.1 (Todorov et al., 2012) on A100 GPUs on a cloud computing service. Table 1 present the hyperparameters used in our experiments. The hyperparameters are all adapted from DrQ-v2 (Yarats et al., 2021a) *without any further hyperparameter tuning*. We have kept the same set of hyperparameters across all algorithms and tasks, except for the walker domain which similarly to DrQ-v2 (Yarats et al., 2021a), we used n-step return of $n = 1$ and mini-batch size of 512.

The core RL components (actor and critic networks), as well as the components of DHPG (state and action encoders, transition and reward models) are all MLP networks with the ReLU activation function and one hidden layer with dimension of 256. The image encoder is based on the architecture of DrQ-v2 which is itself based on SAC-AE (Yarats et al.,

Table 1: Hyperparameters used in our experiments.

Hyperparameter	**Setting**
Learning rate	1e−4
Optimizer	Adam
n-step return	3
Mini-batch size	256
Actor update frequency d	2
Target networks update frequency	2
Target networks soft-update τ	0.01
Target policy smoothing stddev. clip c	0.3
Feature dim.	50
Action repeat	2
Frame stack	3
Hidden dim.	256
Replay buffer capacity	10^6
Discount γ	0.99
Seed frames	4000
Exploration steps	2000
Exploration stddev. schedule	$\text{linear}(1.0, 0.1, 1\text{e}6)$

2021b) and consists of four convolutional layers of $32 \times 3 \times 3$ with ReLU as their activation functions, followed by a one-layer fully-connected neural network with layer normalization (Ba et al., 2016) and tanh activation function. The stride of the convolutional layers are 1, except for the first layer which has stride 2. The image decoder of the baseline models with image reconstruction is based on SAC-AE (Yarats et al., 2021b) and has a single-layer fully connected neural network followed by four transpose convolutional layers of $32 \times 32 \times 3$ with ReLU activation function. The stride of the transpose convolutional layers are 1, except for the last layer which has stride 2.

D.2 Baseline Implementations

All of the baselines are submitted in the supplemental material. We use the official implementations of DBC and SAC-AE. DeepMDP does not have a publicly available code, and we use the implementation available in the official DBC code-base. As discussed in Section 7, we have run two versions of the baselines, with and without image augmentation. The image augmented variants, use the same image augmentation method of DrQ-v2 described in Appendix D.1.