Approximate Equivariance in Reinforcement Learning

Jung Yeon Park[*1] **Sujay Bhatt**[2] **Sihan Zeng**[2] **Lawson L.S. Wong**[1]

Alec Koppel[2] **Sumitra Ganesh**[2] **Robin Walters**[1]

[1]Northeastern University [2]J.P. Morgan AI Research

Abstract

Equivariant neural networks have shown great success in reinforcement learning, improving sample efficiency and generalization when there is symmetry in the task. However, in many problems, only approximate symmetry is present, which makes imposing exact symmetry inappropriate. Recently, approximately equivariant networks have been proposed for supervised classification and modeling physical systems. In this work, we develop approximately equivariant algorithms in reinforcement learning (RL). We define approximately equivariant MDPs and theoretically characterize the effect of approximate equivariance on the optimal Q function. We propose novel RL architectures using relaxed group convolutions and experiment on several continuous control domains and stock trading with real financial data. Our results demonstrate that approximate equivariance matches prior work when exact symmetries are present, and outperforms them when domains exhibit approximate symmetry. As an added byproduct of these techniques, we observe increased robustness to noise at test time.

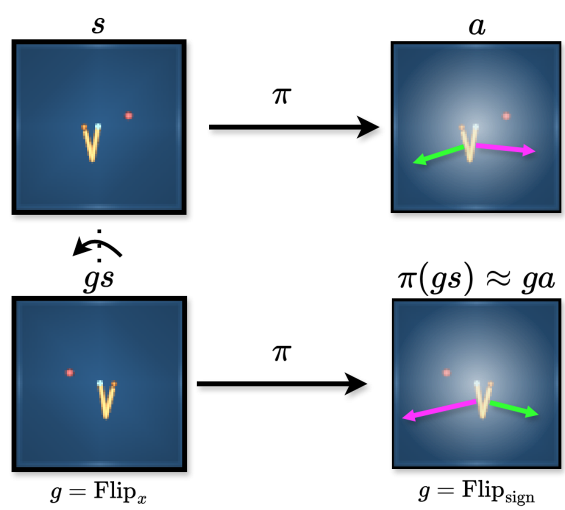

Figure 1: An approximately equivariant policy π on a Reacher domain, where the goal is to determine the torques (green, magenta) to apply on each joint for the fingertip to reach the target (red). Due to wear, the first joint is more responsive to positive torques. When the state is flipped, the policy also flips the actions but can learn to adjust for symmetry breaking factors.

1 Introduction

Symmetry is a powerful inductive bias that can be used to improve generalization and data efficiency in deep learning. One way to leverage symmetry is through equivariant neural networks, which are model classes constrained to respect the symmetry of a known ground truth. Equivariant neural networks have successfully been applied to image classification (Cohen and Welling, 2016; Worrall et al., 2017), particle physics (Bogatskiy et al., 2020), molecular biology (Satorras et al., 2021; Thomas et al., 2018), and robotic manipulation (Wang et al., 2022b). Empirical studies have demonstrated that equivariant networks require much less data than their standard neural network counterparts (Winkels and Cohen, 2018; Wang et al., 2022b) and can generalize better to unseen data (Wang et al., 2020; Fuchs et al., 2020).

However, equivariant neural networks crucially assume that the data is perfectly symmetric in both the inputs and outputs, which may not be true in real-world data such as fluid dynamics (Wang et al., 2022c) or financial data (Black, 1986). By relaxing the strict equivariance constraints, approximately equivariant networks

[*]Work done during internship at J.P. Morgan AI Research.

can outperform exactly equivariant and unconstrained networks in the presence of asymmetry. While various approaches to achieve approximate equivariance have been proposed (Wang et al., 2022c; van der Ouderaa et al., 2022; McNeela, 2023; Kim et al., 2023), they focused on vision-based tasks or dynamics modeling.

One area where symmetry has been especially useful is in reinforcement learning (RL), where equivariant networks greatly improve sample efficiency (Wang et al., 2022b; Zhu et al., 2022), a key challenge in RL. However, most works consider exact symmetry and use exact equivariant networks, which cannot address symmetry breaking in the reward or transition functions or noise in the observations. In this work, we employ relaxed group convolutional neural networks to RL (Wang et al., 2022c), which are flexible enough to adapt to approximate equivariance but also have improved efficiency and robustness.

In this paper, we theoretically and empirically investigate approximately equivariant reinforcement learning. Our key contributions are to:

- formalize the notion of approximately equivariant MDPs and prove that the (optimal) value function in such MDPs exhibits approximate equivariance, motivating the use of approximately equivariant RL,
- introduce a novel approximately equivariant RL architecture using relaxed group convolutions,
- demonstrate improved sample efficiency and robustness to noise for our approximately equivariant RL compared to other baselines with or without symmetry biases,
- successfully apply approximate equivariant RL to real-world financial data.

2 Related Work

Equivariant Reinforcement Learning Early works explored equivalence classes in reinforcement learning from the lens of abstractions by defining MDP homomorphisms (Ravindran and Barto, 2002; Zinkevich and Balch, 2001). More recently, several approaches have combined function approximation with RL with equivariant neural networks (Van der Pol et al., 2020; Wang et al., 2022b; Mondal et al., 2020) with significantly improved sample efficiency. However, all of these works considered perfectly symmetric domains where the policy is constrained to be exactly equivariant. This paper considers domains with symmetry breaking factors where exactly equivariant networks can be suboptimal.

Approximate Equivariant Architectures There has been recent interest in exploring approximate equivariance and approximately equivariant neural networks (Finzi et al., 2021; Wang et al., 2022c; Romero and Lohit, 2022; van der Ouderaa et al., 2022; McNeela, 2023; Petrache and Trivedi, 2024; Samudre et al., 2024). Wang et al. (2022c, 2024b) express the exactly equivariant group convolution kernel as a linear combination of kernels to achieve relaxed equivariance and discover symmetry breaking factors. van der Ouderaa et al. (2022) define a nonstationary kernel and a tunable frequency parameter to control the amount of approximate equivariance. McNeela (2023) propose using a neural network to approximate the exponential map from the Lie algebra to the group. Petrache and Trivedi (2024) give theoretical bounds on when approximate equivariance can improve generalization. However, none of these works studied approximate equivariance in RL, the main focus of this work.

Closest to our setting is Residual Pathway Priors (Finzi et al., 2021), which considered soft equivariance constraints in model-free RL. They construct a neural network layer as the sum of an exactly equivariant layer and a non-equivariant layer, deriving intuition from residual connections. We take a different approach in this work and use relaxed group convolutions Wang et al. (2022c), which are flexible enough to learn different outputs for each transformation.

Learning with Latent Symmetry Other works also apply equivariant neural networks to domains with latent symmetry. These are cases where the full state has exact symmetry but only partial observations with an unknown group action are available to the model. Park et al. (2022) learn the out-of-plane rotations from 2D images using a symmetric embedding network while others have learned 3D rotational features from images using manifold latent variables (Falorsi et al., 2018) or disentanglement (Quessard et al., 2020). Wang et al. (2022a) find that equivariant models where the group acts directly on observation space perform well in RL even with camera skew or occlusions. They define extrinsic equivariance (transformed samples are outside the data distribution) and show that it can benefit in some scenarios but can also be harmful in certain cases (Wang et al., 2024a). Unlike these works where the observation is partial and does not contain full information about the state, we assume that the domains are fully observable and consider various symmetry breaking factors.

3 Background

In this section, we provide some background on symmetry groups and equivariant functions. As building blocks of exactly and approximately equivariant networks, we also describe exact and relaxed group convolutions, respectively.

A symmetry group G is a set equipped with a binary operation that satisfies associativity, existence of an identity, and existence of inverses. A group can act on vector space X via a group representation ρ_X which homomorphically assigns each element $g \in G$ an invertible matrix $\rho_X(g) \in \mathrm{GL}(X)$. For example, for a finite group G, the regular representation acts on $\mathbb{R}^{|G|}$ by permuting basis elements $\{e_g : g \in G\}$ as $\rho_{\mathrm{reg}}(h)e_g = e_{hg}$. A function $f : X \to Y,\ x \mapsto y$ is G-equivariant if $f(\rho_X(g)(x)) = \rho_Y(g)f(x)$. That is, transformations of the input x by g correspond to transformations of the output by the same group element. We can enforce this constraint in equivariant neural networks to learn only over the space of equivariant functions by replacing linear layers with group convolutional layers. One benefit of enforcing equivariance constraints is lower sample complexity as the network searches over a reduced function class.

3.1 Group Convolution

One method to construct equivariant network layers is by group convolutions (Cohen and Welling, 2016), which we briefly describe here. Group convolutions map between features which are signals over the group $f : G \to \mathbb{R}$. For inputs not natively of this form, a lift operation must first be performed. Let $\psi_\theta : G \to \mathbb{R}$ be the convolutional kernel parameterized by θ. A G-equivariant group convolutional layer is defined as

$$(f \star \psi_\theta)(g) = \sum_{h \in G} \psi_\theta(g^{-1}h)f(h). \tag{1}$$

Equivariance follows from the fact that the kernel depends only on the product $g^{-1}h$ and not the specific elements (g, h). For example, if we consider equivariance across translations, we obtain the standard convolution where $h, g \in \mathbb{Z}^2$ and $g^{-1}h = h - g$. Another possible approach to construct equivariant network layers is with G-steerable convolutions (Cohen and Welling, 2017), which can generalize to continuous groups.

3.2 Relaxed Group Convolution

A key component of our method is the relaxed version of the group convolution (Wang et al., 2022c). The kernel ψ is replaced with several kernels $\{\psi_l\}_{l=1}^{L}$ and the output is composed as a linear combination. The relaxed group convolution is defined as

$$(f \tilde{\star} \psi_\theta)(g) = \sum_{h \in G} f(h) \sum_{l=1}^{L} w^l(h)\psi_\theta^l(g^{-1}h), \tag{2}$$

where w^l are the relaxed weights and each ψ_θ^l are constrained to be exactly equivariant. Note that as $w^l(h)$ depends on the specific element h, this breaks the strict equivariance of the group convolution. Wang et al. (2022c) also introduce relaxed versions of steerable convolutions.

3.3 Approximate Equivariance

There have been several different definitions of approximate, relaxed, or partial equivariance. In this paper, we use the definition given by Petrache and Trivedi (2024) and first give some background to build up to its definition. Let G be a group and $f : X \to Y, x \mapsto y$ be the task function.

Definition 1 (Equivariance Error)**.** *For $g \in G$ and $x \in X$, the equivariance error $ee(f, g, x)$ is defined as*

$$ee(f, g, x) = \|f(g(x)) - g(f(x))\|, \tag{3}$$

Equivariance error measures exactly how far a function is from perfect equivariance with respect to G for a particular x. For an exactly G-equivariant function, $ee(f, g, x) = 0$ for all $g \in G$ and $x \in X$.

Definition 2. *(ε-stabilizer) The ε-stabilizer of f and G is defined as*

$$Stab_\varepsilon(f, G) = \{g \in G \mid ee(f, g, x) \leq \varepsilon\}. \tag{4}$$

The ε-stabilizer gives the set of group elements for which the equivariance error is under some threshold.

Definition 3 (Approximate G-Equivariance)**.** *Given a function $f : \mathcal{X} \to \mathcal{Y}$ and a group G, f is approximately G-equivariant if $Stab_\varepsilon(f, G) = G$.*

We adopt the definition of approximate equivariance where f has bounded equivariance error for all $g \in G$, in contrast to *partial equivariance*, where $\mathrm{Stab}_\varepsilon(f, G) < G$.

4 Method: Approximately Equivariant Reinforcement Learning

We first theoretically characterize the problem by defining approximately equivariant Markov Decision Processes (MDP). We then prove that environments with approximate symmetry admit approximately invariant Q functions. This motivates our method of using approximately equivariant neural networks to learn the policy and Q function.

4.1 Approximately Equivariant MDP

Let a Markov Decision Process (MDP) be represented by a tuple $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, instantaneous reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a transition function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ and discount factor $\gamma \in (0, 1)$. The goal of solving an MDP is to find a policy $\pi \in \Pi$, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ (denoted as $\pi(a|s)$) that maximizes the expected return from any state $V_t^\pi(s) = \mathbb{E}^\pi\Big[\sum_{k=t}^{\infty} \gamma^{k-t} R_k \Big| s_t = s\Big]$. The expected return from a state s after taking action a under a policy π is given by the Q function,

$$Q_t^\pi(s, a) := \mathbb{E}^\pi\Big[\sum_{k=t}^{\infty} \gamma^{k-t} R_k \Big| s_t = s, a_t = a\Big].$$

Let $V_t(s_t) := \sup_\pi V_t^\pi(s_t)$ and

$$Q_t(s_t, a_t) = \mathbb{E}\Big[R_t + \gamma V_{t+1}(s_{t+1}) \Big| s_t, a_t\Big]$$

denote the policy independent expected return. Let G be a group acting on $\mathcal{S}$ and $\mathcal{A}$. Denote the action of an element $g \in G$ on s and a by gs and ga, respectively. We now extend the definition of Equivariant MDPs (Van der Pol et al., 2020) to cases where the symmetry is approximate.

Definition 4. *An MDP is $(G, \epsilon_R, \epsilon_P)$-invariant if*

$$|R(gs, ga) - R(s, a)| \leq \epsilon_R, \forall g \in G$$
$$d_{\mathscr{F}}\Big(P(gs' \mid gs, ga), P(s' \mid s, a)\Big) \leq \epsilon_P, \forall g \in G,$$

where $d_{\mathscr{F}}(\mu, \nu) := \sup_{f \in \mathscr{F}} \Big| \int_{\mathcal{S}} f d\mu - \int_{\mathcal{S}} f d\nu \Big|$ is an integral probability metric (IPM) between two distributions $\mu, \nu \in \Delta(\mathcal{X})$.

Some well known examples of IPM include (Sriperumbudur et al., 2009): total variation distance ($\mathscr{F} = \{f : \|f\|_\infty \leq 1\}$) and Kantorovich metric ($\mathscr{F} = \{f : \|f\|_{\text{Lip}} \leq 1\}$). A useful property of IPMs is, given a function class $\mathscr{F}$ and a function f (Müller, 1997)

$$\Big| \int_{\mathcal{S}} f d\mu - \int_{\mathcal{S}} f d\nu \Big| \leq \rho_{\mathscr{F}}(f) \cdot d_{\mathscr{F}}(\mu, \nu),$$

where the Minkowski functional w.r.t $\mathscr{F}$ is

$$\rho_{\mathscr{F}}(f) = \inf\{\rho \in \mathbb{R}_{\geq 0} : \rho^{-1} f \in \mathscr{F}\}.$$

For the total variation distance $\rho_{\mathscr{F}}(f) := \frac{1}{2}(\max f - \min f)$ and for Kantorovich metric $\rho_{\mathscr{F}}(f) := \|f\|_{\text{Lip}}$.

For any uniformly bounded function V, consider the Bellman equation

$$V(gs) = \max_{a \in \mathcal{A}} \Big\{ r(gs, ga) + \gamma \int_{\mathcal{S}} V(gs') P(gs'|gs, ga) \Big\}. \tag{5}$$

Let V^* denote the fixed point[1] of (5) and Q^* denote the corresponding action-value function.

[1] Fixed point exists by Banach fixed point theorem.

Theorem 1. *Let the rewards $R \in [R_{\min}, R_{\max}]$ be bounded and let $g \in G$ be an onto mapping. For any state s_t and action a_t at arbitrary time t, we have*

$$|Q_t(s_t, a_t) - Q^*(gs_t, ga_t)| \leq \alpha,$$
$$|V_t(s_t) - V^*(gs_t)| \leq \alpha,$$

where $\alpha = \frac{\epsilon_R + \gamma \rho_{\mathscr{F}}(V^*) \epsilon_P}{1 - \gamma}$.

The proof is provided in Appendix A. When the Kantorovich metric is used for uncertainty characterization, $\rho_{\mathscr{F}}(V^*) = \|V^*\|_{\text{Lip}}$, where $\|\cdot\|_{\text{Lip}}$ is the Lipschitz norm of the value function (Gelada et al., 2019). For total variation distance, $\rho_{\mathscr{F}}(V^*) = |R_{\max} - R_{\min}|$. Theorem 1 provides a gap-quantification between the Q-function on the original and symmetry transformed domain in terms of the mismatch in the reward and transition invariance. This implies that when the invariance mismatch is small – i.e., when the domain has only minor symmetry violations – the Q-function is approximately group-invariant. This motivates us to use approximately equivariant neural networks for the policy and critic in domains with inexact symmetry.

4.2 Approximately Equivariant Actor-Critic

We propose approximately equivariant versions of two commonly used actor-critic algorithms, DrQv2 (Yarats et al., 2021) and SAC (Haarnoja et al., 2018). In doing so, we generalize exactly equivariant versions of SAC (Wang et al., 2022b) and DrQv2 (Wang et al., 2022a) from previous works by replacing strictly equivariant layers with relaxed equivariant layers.

Encoder, Policy, and Critic We replace each group convolution with relaxed group convolutions (see Appendix B for a version using relaxed steerable convolutions) for the encoder, policy, and critics. Practically, each relaxed group convolution layer contains L exactly equivariant kernels ψ_l and the output is a linear combination of the outputs of these convolutions and relaxed weights $w^l(g)$. The $w^l(g)$ also transform as the regular representation of G.

The encoder E and the policy π are approximately equivariant. The latent state z output by E is defined to transform as the regular representation of G. The action representation is domain-specific. The critics are approximately invariant and output scalars $q_{(s,a)}$ that are fixed by G, i.e. transform via the trivial representation. For more details, please see Section 5 and Appendix B.

Illustrative Example We illustrate how to apply our proposed approximately equivariant actor-critic architecture on the `Reacher` domain; see Figure 2. The

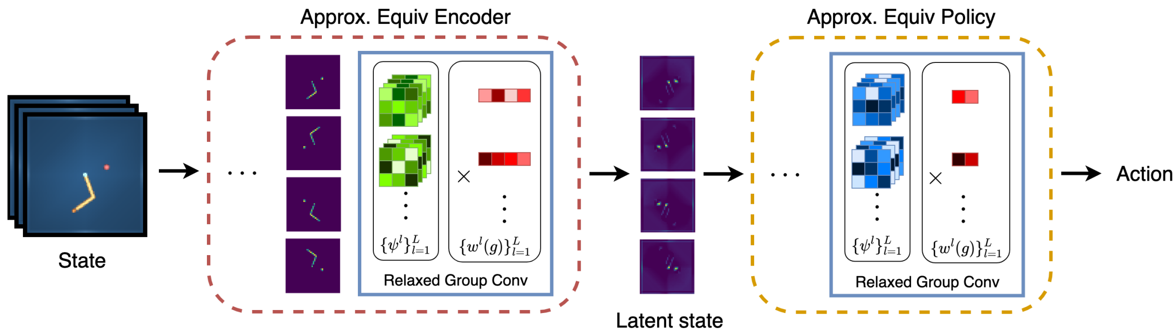

Figure 2: Illustration of the approximately D_2-equivariant encoder and policy (critic is not shown for space). The D_2 group consists of vertical reflections and π rotations. Both the encoder and policy consist of relaxed group convolution layers.

state is a stack of consecutive images $s \in \mathbb{R}^{C\times H\times W}$ and the action $a \in \mathbb{R}^2$ corresponds to torques for the first and second arms. As in the example in Figure 1, the first joint is more responsive to positive torques. For this domain, we implement approximate equivariance with respect to the group D_2 of vertical reflections and π rotations. The group D_2 transforms the input states by image transformations. Latent representations are images $z\colon \mathbb{R}^2 \to \mathbb{R}^C$ where $g \in D_2$ acts on the pixel axes by image transformation and on the channel axis by permutations corresponding to the regular representation of D_2, i.e. $(gz)(x,y) = \rho_{\text{reg}}(g)z(g^{-1}\cdot(x,y))$. For the output, the torques a_1 and a_2 are scalars that change sign under reflection but are invariant under rotations.

5 Experiments

We experiment on how approximately equivariant RL compares to methods with exact equivariance and no equivariance in domains with both exact symmetry and various symmetry breaking factors, and to elucidate when approximate equivariance should be preferred. We consider standard continuous control domains and stock trading with real-world data.

5.1 Continuous Control

We first experiment on four continuous control domains in DeepMind Control Suite (Tassa et al., 2018). Similar to Wang et al. (2022a), we consider a subset of the domains which have symmetry: `Acrobot`, `Cartpole`, `BallInCup` have reflectional symmetry described by the group D_1 and `Reacher` has D_2 symmetry. For all domains, the observations are a stack of 3 consecutive RGB images.

We modify the domains to carefully control the type and degree of symmetry breaking present. We first remove fixed background features such as random stars in the sky and checkered floors (see Figure 4). These features break symmetry to some extent since they do not transform with the underlying state, but give a form of mild symmetry breaking termed extrinsic equivariance which has an inconsistent impact on equivariant models (Wang et al., 2022a). We then introduce several different symmetry breaking factors for each domain: 1) `repeat_action` - the action is repeated twice in a certain region of the domain, 2) `gravity` - gravity is modified from $[0, 0, -9.81]$ to $[a, -a, -9.81]$ where $a \neq 0$, 3) `reflect_action` - the action direction is flipped in certain regions of the domain. `repeat_action` and `reflect_action` test local symmetry breaking factors, while `gravity` tests a global symmetry breaking factor. See Appendix C.1 for more details.

Models For the continuous control tasks, we implement an approximately equivariant(`ApproxEquiv`) version of a SOTA image-based RL algorithm DrQv2 (Yarats et al., 2021). We compare with exactly equivariant (`ExactEquiv`) and non equivariant (`NonEquiv`) versions of the same architecture. We largely use the hyperparameters from Yarats et al. (2021), but reduce the latent dimension for more tractable computation for all methods. We also compare against a approximately equivariant model, Residual Pathway Priors (RPP) (Finzi et al., 2021), and extend it to the DrQv2 architecture. As mentioned in the original paper, we find that RPP is somewhat sensitive to the speed τ of the critic moving average, and had to reduce its value for `Acrobot` and `BallInCup` for stability.

Results Figure 3 show the total episode reward over training. As expected, we confirm that `NonEquiv` has much lower sample efficiency than the models with a symmetry bias. In the `repeat_action` and `reflect_action` variants of Acrobot, `ApproxEquiv` significantly outperforms `ExactEquiv` and `RPP`. It does slightly worse than `ExactEquiv` on the `Reacher` domain but beats `RPP`, suggesting that our domain mod-

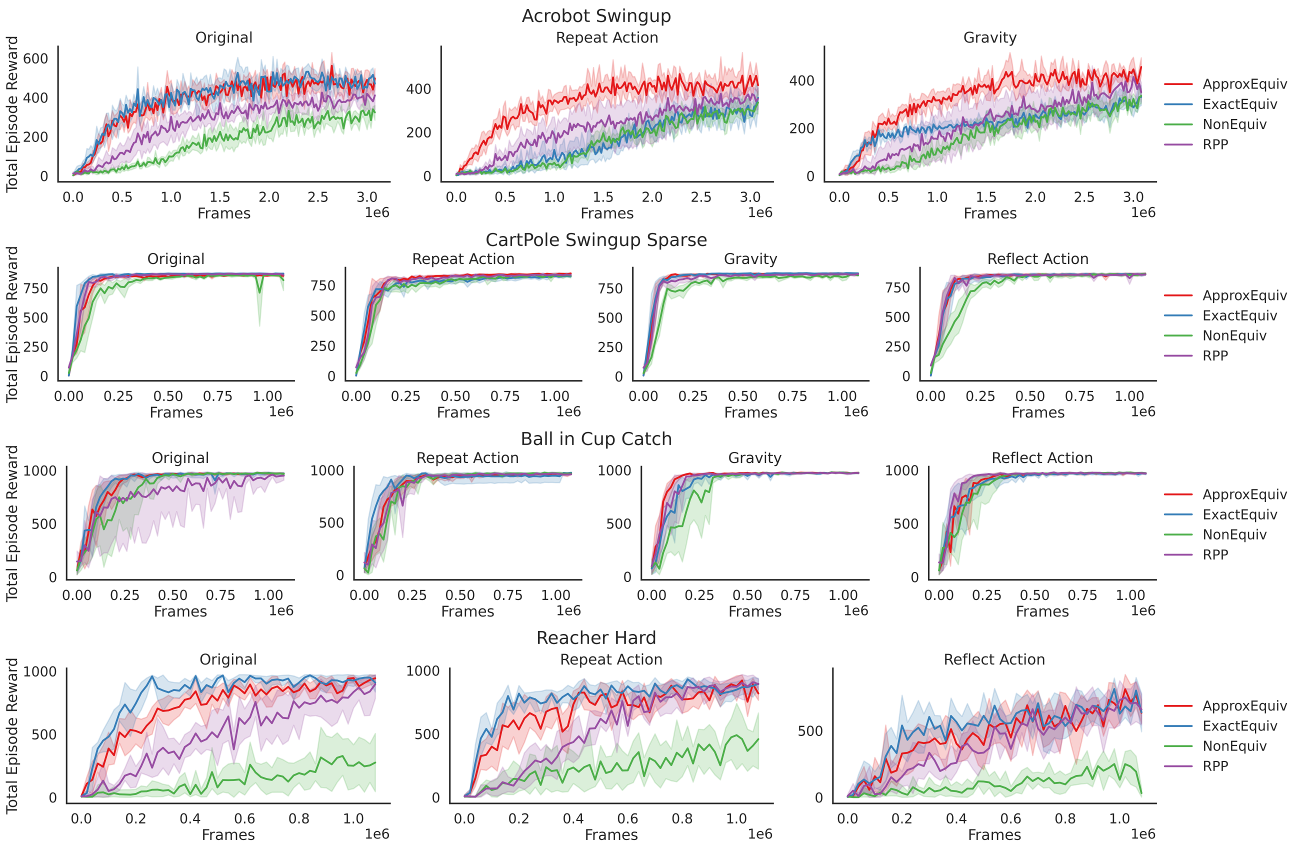

Figure 3: Total episode reward on selected domains in the DeepMind Control Suite, shaded regions indicate 95% confidence intervals (CI). Our approximately equivariant agent performs similarly to an exactly equivariant agent in the original domains due to exact symmetry and outperforms it on some modified domains with inexact symmetry as it can adjust for symmetry breaking. Both methods outperform a non-equivariant agent.

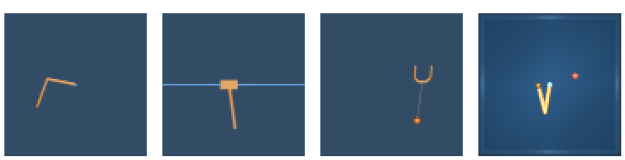

Figure 4: Selected domains in DeepMind Control Suite. The domains were modified to remove extrinsic symmetry and to include several types of symmetry breaking factors such as repeating or reflecting actions in certain states, or by artificially modifying gravity.

ifications were not "severe" enough to achieve incorrect equivariance. It is also possible that `ExactEquiv` can infer the symmetry breaking factors from the 3 frames of input, making the task a case of extrinsic equivariance where an equivariant model can succeed Wang et al. (2022a). In `CartPole` and `BallInCup`, all methods perform similarly and learn an optimal policy quickly. In domains with exact symmetry (`original`), our method `ApproxEquiv` performs similarly to `ExactEquiv`, showing there is no cost in performance by giving the model the ability to adapt to symmetry breaking in cases where it is not needed.

This result supports Proposition 3.1 from Wang et al. (2024b), which proves that relaxed group convolutions initialized to be exactly equivariant stay exactly equivariant when trained with exact data symmetry.

We visualize the relaxed weights of the first layers of the encoder and policy over all runs in Figure 5. If these weights are equal, the model is equivariant; the more they differ the more the model has relaxed the symmetry constraint. For `Acrobot` and `CartPole`, the weights differ more for the modified domains than the original symmetric domain, especially for the encoder, while the policy weights vary more for the modified domains of `BallInCup`. This indicates the relaxed equivariant models have adapted to the symmetry breaking in the domains.

To quantitatively evaluate the models, we select the best-performing policy from all runs and measure the total reward over 50 episodes. The results echo the training curves in Figure 3, where `ApproxEquiv` performs well, particularly in the domains with symmetry breaking factors (see Table 1). To test whether approximately equivariant models are robust to noisy

Table 1: Total episode reward on 50 rollouts for the best policy in the original and noisy domains. Gray values indicate 95% CI. `ApproxEquiv` learns a better policy than baselines on the modified domains and is more robust to noisy inputs.

		No Noise			Noisy		
		ApproxEquiv	ExactEquiv	NonEquiv	ApproxEquiv	ExactEquiv	NonEquiv
Acrobot	Original	389±11	**522**±21	309±22	344±14	**402**±22	190±14
	Gravity	**471**±17	382±15	358±23	**369**±15	218±10	202±12
Cartpole	Original	876±0.2	**881**±0.1	**881**±0.1	778±22	**855**±0.6	572.5±25
	Repeat Action	**859**±0.4	749±13	855±0.6	**624**±6.0	523±21	192±5.0
Ball in Cup	Original	961±0.0	958±0.0	**970**±0.0	**882**±7.7	783±24	0±0.0
	Gravity	**969**±0.0	966±0.0	959±0.0	**888**±13.2	0±0.0	1.8±1.8
Reacher	Original	903±33	**950**±15	519±68	**778**±41	745±44	247±52
	Reflect Action	**757**±55	707±59	243±58	**659**±42	217±41	82±29

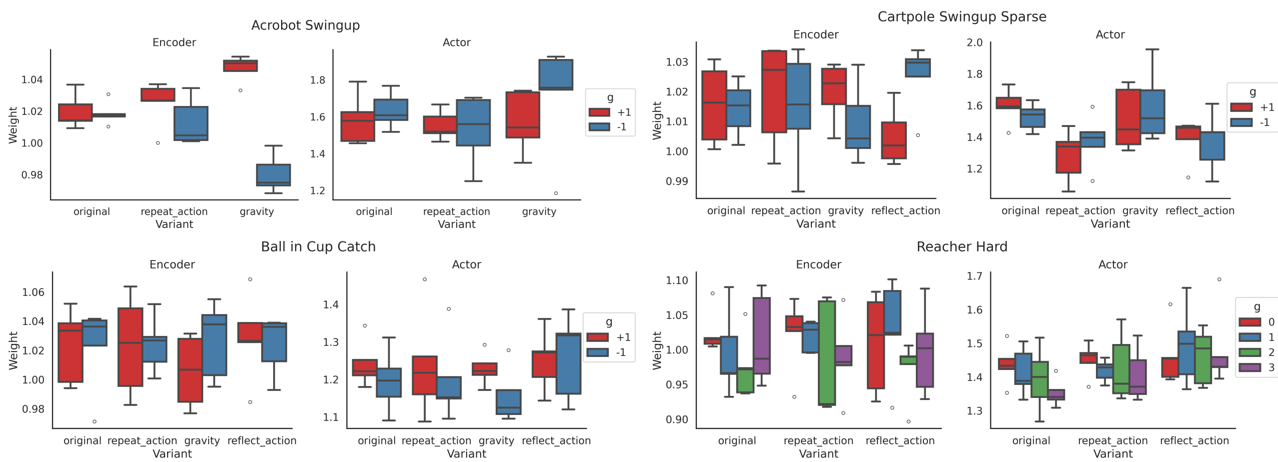

Figure 5: Visualization of relaxed weights for the first layer of the encoder and policy over all runs. Similar weights for each g indicate perfect equivariance while differing values indicate symmetry breaking. The modified variants of most domains exhibit larger differences and/or increased variance in the relaxed weights compared to the `original` variant.

observations, we also consider variants of the domains where Gaussian noise are added to the input images ($\sigma = 0.02$ for `Acrobot` and `Reacher`, $\sigma = 0.06$ for `CartPole` and `BallInCup`). Interestingly, we find that our approach is more robust to noisy inputs than `ExactEquiv` or `NonEquiv`, especially on the `BallInCup` and `Reacher` domains.

5.2 Stock Trading

We also consider a stock trading task using real world price data, formulated as as an MDP (Liu et al., 2018). Given a fixed amount of initial cash, the objective is to learn the optimal number of stocks to buy and sell (once daily) to maximize the portfolio value. The state consists of the current cash balance, the stock prices, the number of shares in the current portfolio, and other technical indicators of each stock. The actions are the number of stocks to buy and sell for each stock. The reward is the scaled difference in portfolio values between consecutive timesteps. We assume that the market dynamics are not affected by our trading. There is a small 0.1% transaction cost for every trade. We use real financial data scraped from Yahoo Finance (yfi, 1997) and consider the stocks in the Dow Jones index from 2001-01-01 to 2024-07-01 (see Appendix C.2 for sample data). Unlike Liu et al. (2018) who used only the current timestep, we use a sliding window approach and use the previous 9 timesteps for the state. See Appendix C.2 for a more detailed description.

Models For this domain, we use SAC (Haarnoja et al., 2018) as our RL algorithm and consider equivariance to both the translation group and scale-translation group across the time dimension. As our actions do not affect stock prices, which in turn is directly correlated with the reward, we learn an approximately invariant policy and invariant critic for both symmetry groups. As before we compare ap-

Table 2: Test results on the stock trading dataset. Gray values indicate 95% CI over 5 runs. The approximately equivariant agents for both scale-translation (ST) and translation (T) outperform the exactly equivariant and non equivariant methods.

		Final Portfolio Value ($mm)	Annualized Return (%)	Sharpe Ratio
ApproxEquiv	ST	1.489±0.16	12.0±3.4	0.63±0.1
	T	1.428±0.04	10.6±3.8	0.60±0.1
ExactEquiv	ST	1.411±0.15	10.3±3.4	0.62±0.2
	T	1.307±0.18	7.8±4.3	0.50±0.3
NonEquiv		1.378±0.05	9.6±1.3	0.62±0.1
Uniform		1.412	10.4	0.71
ˆDJI		1.293	7.7	0.53

proximately equivariant, strictly equivariant, and unconstrained models. We evaluate each method on the final portfolio value (equivalent to the total episode reward), annualized return, and the Sharpe ratio (Sharpe, 1994), which is a standard financial metric that measures an asset's risk-adjusted performance. We also include as baselines a uniform holding strategy `Uniform`, where we initially buy equal values of each stock and hold, and the Dow Jones index `^DJI`.

Results Table 2 lists the average test results of the learned policies on the stock trading domain. The `ApproxEquiv` model for both translation (T) and scale-translation (ST) outperform all baselines, with annualized returns of 10.6% and 12.0% respectively. The `Exact ST-Equiv` model outperforms `NonEquiv`, while the `Exact T-Equiv` model does worse. These observations suggest that temporal scale and translation symmetries can be good biases in analyzing financial data and that translation symmetry may be more approximate than scale. We also visualize 10 episode rollouts of the best-performing policies in Figure 6, with the portfolio values on the left and transaction costs on the right. The Approx ST-Equiv method achieves the highest portfolio value for most timesteps and incurs lower transaction costs than the exactly equivariant policies. We note that overall the annualized returns are fairly low, as the test dataset from 2021-01-01 to 2024-07-01 includes both the COVID-19 pandemic and 2022 stock market decline.

We visualize the relaxed weights of the first layer of the encoder across translation (left) and scale (right) in Figure 7. For translation, our model places higher weights on the very last timestep. This matches our intuition as the most recent stock prices and portfolio holdings would be most informative in determining the optimal action. For scale, we find that the relaxed weights do not differ greatly, but there is increased variance with increasing scale.

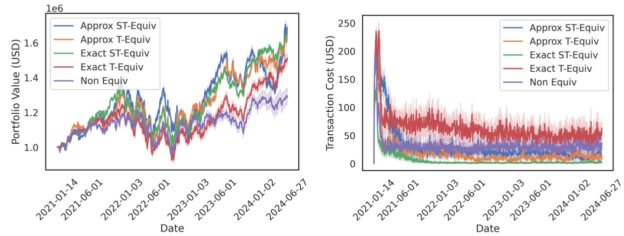

Figure 6: 10 episode rollouts from the best performing policy for each method. Approx ST-Equiv often achieves the highest portfolio value for each time step and incurs minimal transaction costs.

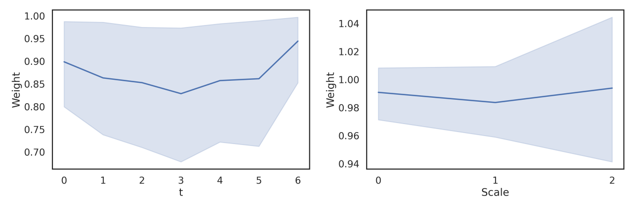

Figure 7: Visualization of relaxed weights for translation (left) and scale (right) over all runs. The relaxed weights for translation differ for each timestep and are similar for scale.

6 Discussion

In this work, we propose a novel approximately equivariant architecture using relaxed group convolutions for model-free reinforcement learning. We first define an approximately equivariant MDP and provide a bound on the optimal Q function. We then apply our architecture to two commonly used model-free algorithms, DrQv2 and SAC, and experiment on continuous control domains and a stock trading problem with real-world data. Our results demonstrate that the approximately equivariant model performs similarly to an exactly equivariant model in domains with perfect symmetry but outperforms it in most domains with symmetry breaking factors. This suggests that our method can act as a much more flexible alternative that can boost sample efficiency in a wider variety of settings and is also more robust to perturbations.

Limitations and Future Work While we did consider real-world data in the stock trading domain, our continuous control domains used simplified observations and synthetic symmetry breaking. Furthermore, exactly equivariant networks perform better in some modified domains than others (`Reacher` vs. `Acrobot`). Another limitation is that, as with all equivariant networks, the symmetry group and how it acts on the state and action spaces need to be known in advance. An interesting future direction could be to quantify exactly what types of symmetry breaking factors could lead to higher performance for approximately equivariant RL, possibly by measuring equivariance error. Other future work includes proving bounds on the optimal policy $\pi(s)$ and $\pi(gs)$ or applying approximately equivariant RL in robotic manipulation, where kinematic constraints or obstacles can break symmetry.

Acknowledgments

This project was supported in part by NSF grants #2134178, 2107256, 2314182.

Disclaimer

This paper was prepared for informational purposes ["in part" if the work is collaborative with external partners] by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan") and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

"Yahoo! Finance". https://finance.yahoo.com/, 1997. Accessed: 2024-07-31.

Fischer Black. Noise. *The journal of finance*, 41(3): 528–543, 1986.

Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pages 992–1002. PMLR, 2020.

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

Taco S Cohen and Max Welling. Steerable cnns. In *International Conference on Learning Representations*, 2017.

Luca Falorsi, Pim De Haan, Tim R Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018.

Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. *Advances in Neural Information Processing Systems*, 34:30037–30049, 2021.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

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.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Hyunsu Kim, Hyungi Lee, Hongseok Yang, and Juho Lee. Regularizing towards soft equivariance under mixed symmetries. In *International Conference on Machine Learning*, pages 16712–16727. PMLR, 2023.

David M Knigge, David W Romero, and Erik J Bekkers. Exploiting redundancy: Separable group convolutional networks on lie groups. In *International Conference on Machine Learning*, pages 11359–11386. PMLR, 2022.

Xiao-Yang Liu, Zhuoran Xiong, Shan Zhong, Hongyang Yang, and Anwar Walid. Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522*, 2018.

Daniel McNeela. Almost equivariance via lie algebra convolutions. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023.

Arnab Kumar Mondal, Pratheeksha Nair, and Kaleem Siddiqi. Group equivariant deep reinforcement learning. *arXiv preprint arXiv:2007.03437*, 2020.

Alfred Müller. How does the value function of a markov decision process depend on the transition probabilities? *Mathematics of Operations Research*, 22(4):872–885, 1997.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan-Willem Van De Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models. In *International Conference on Machine Learning*, pages 17372–17389. PMLR, 2022.

Mircea Petrache and Shubhendu Trivedi. Approximation-generalization trade-offs under (approximate) group equivariance. *Advances in Neural Information Processing Systems*, 36, 2024.

Robin Quessard, Thomas Barrett, and William Clements. Learning disentangled representations and group structure of dynamical environments. *Advances in Neural Information Processing Systems*, 33:19727–19737, 2020.

Balaraman Ravindran and Andrew G Barto. Model minimization in hierarchical reinforcement learning. In *Abstraction, Reformulation, and Approximation: 5th International Symposium, SARA 2002 Kananaskis, Alberta, Canada August 2–4, 2002 Proceedings 5*, pages 196–211. Springer, 2002.

David W Romero and Suhas Lohit. Learning partial equivariances from data. *Advances in Neural Information Processing Systems*, 35:36466–36478, 2022.

Ashwin Samudre, Mircea Petrache, Brian D Nord, and Shubhendu Trivedi. Symmetry-based structured matrices for efficient approximately equivariant networks. *arXiv preprint arXiv:2409.11772*, 2024.

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.

William F Sharpe. The sharpe ratio. *Journal of portfolio management*, 21(1):49–58, 1994.

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1802.00690*, 2018.

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.

Tycho van der Ouderaa, David W Romero, and Mark van der Wilk. Relaxing equivariance constraints with non-stationary continuous filters. *Advances in Neural Information Processing Systems*, 35:33818–33830, 2022.

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*, 33:4199–4210, 2020.

Dian Wang, Jung Yeon Park, Neel Sortur, Lawson LS Wong, Robin Walters, and Robert Platt. The surprising effectiveness of equivariant models in domains with latent symmetry. In *The Eleventh International Conference on Learning Representations*, 2022a.

Dian Wang, Robin Walters, and Robert Platt. SO(2)-equivariant reinforcement learning. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=7F9cOhdvfk_.

Dian Wang, Xupeng Zhu, Jung Yeon Park, Mingxi Jia, Guanang Su, Robert Platt, and Robin Walters. A general theory of correct, incorrect, and extrinsic equivariance. *Advances in Neural Information Processing Systems*, 36, 2024a.

Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*, 2020.

Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pages 23078–23091. PMLR, 2022c.

Rui Wang, Elyssa Hofgard, Han Gao, Robin Walters, and Tess Smidt. Discovering symmetry breaking in physical systems with relaxed group convolution. In *Forty-first International Conference on Machine Learning*, 2024b.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.

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.

Marysia Winkels and Taco S Cohen. 3d g-cnns for pulmonary nodule detection. *arXiv preprint arXiv:1804.04656*, 2018.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037, 2017.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

Xupeng Zhu, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, and Robert Platt. Sample efficient grasp learning using equivariant models. In *Robotics: Science and Systems*, 2022.

Martin Zinkevich and Tucker R Balch. Symmetry in markov decision processes and its implications for single agent and multiagent learning. In *Proceedings of the Eighteenth International Conference on Machine Learning*, page 632, 2001.

APPENDIX

A PROOF OF THEOREM 1

For a given stochastic policy $\pi = (\pi_1, \pi_2 \cdots, \pi_{T-1})$, let

$$\mathcal{V}_t^\pi(s) = \mathbb{E}^\pi\Big[\sum_{k=t}^{T-1}\gamma^{k-t}R_k\Big|s_t = s\Big],$$
$$\mathcal{Q}_t^\pi(s,a) = \mathbb{E}^\pi\Big[R_t + \gamma\mathcal{V}_{t+1}^\pi(s_{t+1})\Big|s_t = a, a_t = a\Big],$$

be the finite-horizon counterparts of the expected return and action-value. Recursively define the policy independent counterparts as follows:

$$\mathcal{V}_T(s_T) = 0, \quad \mathcal{V}_T(gs_T) = 0,$$
$$\mathcal{Q}_t(s_t, a_t) = R(s_t, a_t) + \gamma\int_{\mathcal{S}}\mathcal{V}_{t+1}(s_{t+1})P(s_{t+1}|s_t, g_t),$$
$$\mathcal{Q}_t(gs_t, ga_t) = R(gs_t, ga_t) + \gamma\int_{\mathcal{S}}\mathcal{V}_{t+1}(gs_{t+1})P(gs_{t+1}|gs_t, ga_t),$$
$$\mathcal{V}_t(s_t) = \max_{a_t\in\mathcal{A}}\mathcal{Q}_t(s_t, a_t), \quad \mathcal{V}_t(gs_t) = \max_{a_t\in\mathcal{A}}\mathcal{Q}_t(gs_t, ga_t).$$

Theorem 1. *Let the rewards $R \in [R_{\min}, R_{\max}]$ be bounded and let $g \in G$ be an onto mapping. For any state s_t and action a_t at arbitrary time t, we have*

$$|Q_t(s_t, a_t) - Q^*(gs_t, ga_t)| \leq \alpha,$$
$$|V_t(s_t) - V^*(gs_t)| \leq \alpha,$$

where $\alpha = \frac{\epsilon_R + \gamma\rho_{\mathscr{F}}(V^*)\epsilon_P}{1-\gamma}$.

The proof is established by first deriving the deviation for an (arbitrary) finite-horizon discounted problem and then using this to derive the bounds for the infinite horizon case. All intermediate results are collected as propositions.

Proposition 1. *For a $(G, \epsilon_R, \epsilon_P)$-invariant MDP, the following holds at any t,*

$$|\mathcal{Q}_t(s_t, a_t) - \mathcal{Q}_t(gs_t, ga_t)| \leq \alpha_t, \quad \textit{and} \quad |\mathcal{V}_t(s_t) - \mathcal{V}_t(gs_t)| \leq \alpha_t,$$

where α_t is given by the following recursion: $\alpha_{T+1} = 0$ and

$$\alpha_t = \epsilon_R + \gamma\Big\{\rho_{\mathscr{F}}(\mathcal{V}_{t+1})\epsilon_P + \alpha_{t+1}\Big\}.$$

Proof. We will prove the results using induction. First, note that the result is true for T by definition. Suppose the result is true for $t+1$, and consider the differential at time t,

$$\begin{aligned}
|\mathcal{Q}_t(s_t, a_t) - \mathcal{Q}_t(gs_t, ga_t)| \leq\; & |R(s_t, a_t) - R(gs_t, ga_t)| \\
& + \gamma\Big|\int_{\mathcal{S}}\mathcal{V}_{t+1}(s_{t+1})P(s_{t+1}|s_t, a_t) - \int_{\mathcal{S}}\mathcal{V}_{t+1}(gs_{t+1})P(gs_{t+1}|gs_t, ga_t)\Big| \\
& \leq \epsilon_R + \gamma\Big|\int_{\mathcal{S}}\mathcal{V}_{t+1}(s_{t+1})P(s_{t+1}|s_t, a_t) - \int_{\mathcal{S}}\mathcal{V}_{t+1}(gs_{t+1})P(s_{t+1}|s_t, a_t)\Big| \\
& + \gamma\Big|\int_{\mathcal{S}}\mathcal{V}_{t+1}(gs_{t+1})P(s_{t+1}|s_t, a_t) - \int_{\mathcal{S}}\mathcal{V}_{t+1}(gs_{t+1})P(gs_{t+1}|gs_t, ga_t)\Big| \\
& \leq \epsilon_R + \gamma\rho_{\mathscr{F}}(\mathcal{V}_{t+1})\epsilon_P + \gamma\int_{\mathcal{S}}\Big|\mathcal{V}_{t+1}(s_{t+1}) - \mathcal{V}_{t+1}(gs_{t+1})\Big|P(s_{t+1}|s_t, a_t).
\end{aligned}$$

The last inequality follows by using the decomposition using Minkowski's functional. Further, note that

$$\Big|\mathcal{V}_{t+1}(s_{t+1}) - \mathcal{V}_{t+1}(gs_{t+1})\Big| \le \max_{a_{t+1}\in\mathcal{A}} |\mathcal{Q}_{t+1}(s_{t+1}, a_{t+1}) - \mathcal{Q}_{t+1}(gs_{t+1}, ga_{t+1})| \le \alpha_{t+1},$$

by induction assumption and the fact that when g is onto

$$\max_{a'\in g\mathcal{A}} \mathcal{Q}_{t+1}(gs_t, a') = \max_{a\in\mathcal{A}} \mathcal{Q}_{t+1}(gs_t, ga).$$

The result follows. □

Proposition 2. *Let the rewards $R \in [R_{\min}, R_{\max}]$. For an arbitrary, but finite, horizon T*

$$\mathcal{Q}_t(s_t, a_t) + \frac{\gamma^{T-t}}{1-\gamma} R_{min} \le Q_t(s_t, a_t) \le \mathcal{Q}_t(s_t, a_t) + \frac{\gamma^{T-t}}{1-\gamma} R_{max}$$

Proof. We have by definition,

$$\begin{aligned}
Q_t(s_t, a_t) &= \mathbb{E}\Big[\sum_{k=t}^{\infty} \gamma^{k-t} R_k \Big| s_t = s, a_t = a\Big] \\
&= \mathbb{E}\Bigg[R_t + \gamma \mathbb{E}\Big[\sum_{k=t+1}^{\infty} \gamma^{k-(t+1)} R_k \Big| s_{t+1}\Big] \Bigg| s_t = s, a_t = a\Bigg] \\
&\le \mathbb{E}\Bigg[R_t + \gamma \mathbb{E}\Big[\mathcal{V}_{t+1}(s_{t+1}) + \frac{\gamma^{T-(t+1)} R_{\max}}{1-\gamma}\Big] \Big| s_{t+1}\Big] \Bigg| s_t = s, a_t = a\Bigg] \\
&= \mathcal{Q}_t(s_t, a_t) + \frac{\gamma^{T-t}}{1-\gamma} R_{\max}.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
Q_t(s_t, a_t) &= \mathbb{E}\Big[\sum_{k=t}^{\infty} \gamma^{k-t} R_k \Big| s_t = s, a_t = a\Big] \\
&\ge \mathbb{E}\Bigg[\sum_{k=t}^{T-1} \gamma^{k-t} R_k + \sum_{k=T}^{\infty} \gamma^{k-t} R_{\min} \Big| s_t, a_t\Bigg] \\
&= \mathbb{E}\Bigg[R_t + \gamma \sum_{k=t+1}^{T-1} \gamma^{k-(t+1)} R_k \Big| s_t, a_t\Bigg] + \frac{\gamma^{T-t}}{1-\gamma} R_{\min} \\
&= \mathcal{Q}_t(s_t, a_t) + \frac{\gamma^{T-t}}{1-\gamma} R_{\min}.
\end{aligned}$$

□

We now prove Theorem 1.

Proof of Theorem 1:

Defining $\mathcal{B}$ to be the Bellman operator, we know that (5) can be written as $V = \mathcal{B}V$. Consider a sequence of value functions $\mathcal{V}^{(n)}$ on the symmetry transformed domain as follows: $\mathcal{V}^{(0)}(gs) = 0$ and $\mathcal{V}^{(n+1)} = \mathcal{B}\mathcal{V}^{(n)}$. For an arbitrary T, we have using Proposition 1 for any $t \in \{1, \cdots, T\}$,

$$|\mathcal{V}_t(s_t) - \mathcal{V}_t^{(T-t)}(gs_t)| \le \alpha_t,$$

where

$$\alpha_t = \epsilon_R + \sum_{\tau=t+1}^{T-1} \gamma^{\tau-t} [\rho_{\mathscr{F}}(\mathcal{V}^{(T-\tau)})\epsilon_P + \epsilon_R].$$

From Proposition 2, we have, noting that $\mathcal{V}(s) = \max_a \mathcal{Q}(s,a)$, that

$$\mathcal{V}_t^{(T-t)}(gs_t) - \alpha_t + \frac{\gamma^{T-t}}{1-\gamma} R_{\min} \leq V_t(s_t) \leq \mathcal{V}_t^{(T-t)}(gs_t) + \alpha_t + \frac{\gamma^{T-t}}{1-\gamma} R_{\max}$$

By Banach fixed point theorem, we know that $\lim_{T\to\infty} \mathcal{V}_t^{(T-t)} = V^*$. By continuity of $\rho_{\mathscr{F}}(\cdot)$, we have that $\lim_{T\to\infty} \rho_{\mathscr{F}}(\mathcal{V}^{(T-\tau)}) = \rho_{\mathscr{F}}(V^*)$ whence $\lim_{T\to\infty} \alpha_t = \alpha := \frac{\epsilon_R + \gamma \rho_{\mathscr{F}}(V^*)\epsilon_P}{1-\gamma}$. Therefore, taking the limit, we have

$$V^*(gs_t) - \alpha \leq V_t(s_t) \leq V^*(gs_t) + \alpha.$$

Similar argument establishes the result for Q using the onto function g. The result holds. □

B BACKGROUND AND METHOD

B.1 Equivariance with Group Convolutions

Group convolutions (Cohen and Welling, 2016) generalize standard convolutions, which are translation-equivariant, to be equivariant to a group G. Group convolutions act on signals over the group $f : G \to \mathbb{R}$. As many data samples are not natively of this form (e.g. an image), the input must first be lifted onto a function in G. For example, let $f_0 : \mathbb{Z}^2 \to \mathbb{R}$ be the input signal, a grayscale image, and $H = D_2$ be the group. The lifting convolution lifts f_0 from $\mathbb{Z}^2$ to $G = D_2 \ltimes \mathbb{Z}^2$ by

$$(f_0 \star \psi)(x,h) = \sum_{y\in\mathbb{Z}^2} f_0(y)\psi(h^{-1}(y-x)), \tag{6}$$

where $h \in H$. Practically, the lift operation creates $|H|$, the order of group H, images by acting on x by h^{-1}. Typically the lift operation is the first layer of the network, followed by subsequent group convolutions, nonlinearities, or other equivariant layers. We use relaxed versions of the lift and group convolutions as described in Wang et al. (2022c) and the main paper.

B.2 Steerable Convolutions

As an alternative to group convolutions, one can use steerable convolutions (Weiler et al., 2018) that use weight tying to generalize to continuous groups and are more parameter-efficient. Let $H < O(2)$ be the subgroup which acts on $\mathbb{R}^2$ by matrix multiplication on the input and output channel spaces $\mathbb{R}^c$ and $\mathbb{R}^d$ by ρ_{in} and ρ_{out}, respectively. Then $G = H \ltimes \mathbb{R}^2$. Given input signal $f : \mathbb{R}^2 \to \mathbb{R}^c$, then standard convolution over $\mathbb{R}^2$ with kernel $\psi : \mathbb{R}^2 \to \mathbb{R}^{d\times c}$ is G-equivariant if ψ satisfies

$$\psi(hx) = \rho_{\text{out}}(g)\psi(x)\rho_{\text{in}}(h^{-1}), \tag{7}$$

for all $h \in H$. Intuitively, this kernel constraint ensures that the output features transform by ρ_{out} when the input features are transformed by ρ_{in}. Kernels that satisfy this constraint have been solved for many common subgroups of $E(2)$, see Weiler and Cesa (2019) for more details.

Using the example of grayscale images as in Section B.1, let the input feature be $f : \mathbb{Z}^2 \to \mathbb{R}$ and $\{\psi_k\}_{k=1}^K$ be a precomputed, nontrainable equivariant kernel basis of K kernels that satisfy Eq. (7). Assume that both the number of input and output channels is 1 and let $w \in \mathbb{R}^K$ be the trainable coefficients of the kernels. Then a G-steerable convolution is defined as

$$(f \star \psi)(x) = \sum_{y\in\mathbb{Z}^2} \sum_{k=1}^K (w_k \psi_k(y)) f(x+y), \tag{8}$$

where $x \in \mathbb{Z}^2$ is the spatial position and w_k is the weight associated with kernel ψ_k.

Relaxed Steerable Convolution As described in Wang et al. (2022c), one can also use relaxed versions of steerable convolutions by letting the trainable weights w also depend on y. A relaxed G-steerable convolution is defined as

$$(f \tilde{\star} \psi)(x) = \sum_{y\in\mathbb{Z}^2} \sum_{k=1}^K (w_k(y)\psi_k(y)) f(x+y). \tag{9}$$

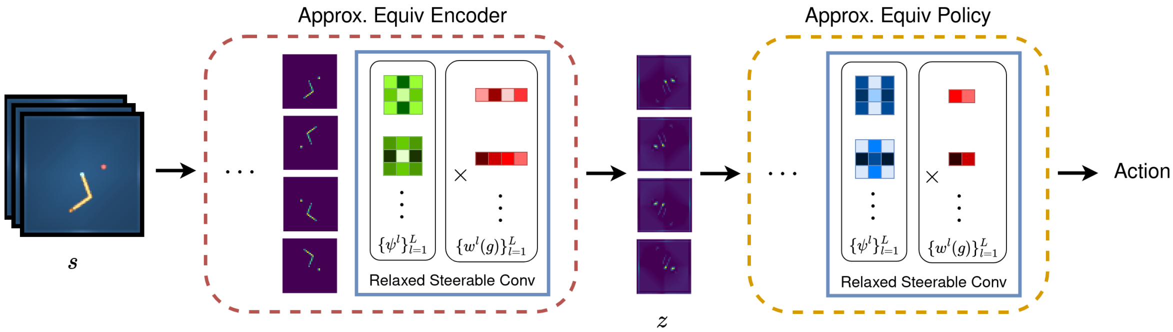

Figure 8: Illustration of an approximately D_2-equivariant encoder and policy using relaxed steerable convolution layers. The critic is not shown and is approximately invariant.

Allowing the trainable weights w_k to also depend on the absolute spatial position y breaks the equivariance constraint in Eq. (7).

By replacing relaxed group convolutions with relaxed steerable convolutions, we can also design a variant of our proposed approximately equivariant RL architecture (Figure 8).

C EXPERIMENTS

C.1 Continuous Control

Acrobot We use the `swingup` task. The domain consists of two joints where the goal is to apply torque to the inner joint so that both joints are near vertical. We use D_1 as the symmetry group, i.e. vertical reflection, and the action $a \in \mathbb{R}$ transforms via the sign representation ρ_{sign}, where $\rho_{\text{sign}}(\text{flip})(a) = -a$. For variants, we consider 1) `repeat_action` - the action is repeated when the inner joint is in the fourth quadrant and 2) `gravity` - gravity $\vec{g} = [0, 0, -9.81]$ is modified to $[-2, 2, -9.81]$.

CartPole We consider the `swingup` task. The domain consists of a pole swinging on a cart and the goal is to move the cart left or right ($a \in \mathbb{R}$) to make the pole upright. The symmetry group and action representation are the same as in `Acrobot`, D_1 and ρ_{sign}. For variants, we consider 1) `repeat_action` - the action is repeated when the pole is in the first quadrant, 2) `gravity` - gravity is modified to $[0.2, -0.2, -9.81]$, and 3) `reflect_action` - the pole angle is in $[0, \frac{\pi}{4}]$. Gravity is modified less than in `Acrobot` as too high values forced the cart out of frame.

Cup Catch The domain consists of a ball attached to the bottom of the cup and the goal is to move the cup to catch the ball inside the cup. The action $(x, z) \in \mathbb{R}^2$ is the cup's spatial position. The symmetry group is D_1 and the action representation is $\rho_{\text{sign}} \oplus \rho_0$, where the x position transforms via the sign representation and the z transforms via the trivial representation ρ_0. For variants, we consider 1) `repeat_action` - the ball x position greater than 0.0 and z position is greater than 0.3, 2) `gravity` - gravity is modified to $[-2, 2, -9.81]$, and 3) `reflect_action` - same as `repeat_action`.

Reacher We consider the `hard` task. The domain consists of two joints and the goal is to apply torques to make the end effector reach the target. The action $a \in \mathbb{R}^2$. The symmetry group is D_2, i.e. vertical reflections and π rotations, and the action transforms via the quotient representation $2\rho_{\text{quot}}$, where the torques for both joints are invariant to rotations and flip signs for vertical reflections. For variants, we consider 1) `repeat_action` - the inner joint angle is in $[0, \frac{\pi}{2}]$ and 2) `reflect_action` - the inner joint angle is in $[\frac{\pi}{2}, \pi]$.

C.1.1 Training Details

For all DeepMind Control Suite (DMC) domains, we fix the episode length to 1000 and use RGB image of size 85×85. We considered four domains of varying difficulty, of which `Acrobot` is the hardest. In the original DrQv2 implementation (Yarats et al., 2021), the encoder reduces the spatial dimensions to 35×35, which is then flattened to be input to the policy and critic. We follow Wang et al. (2022a) and further reduce the spatial dimensions to 7×7 for faster training for all models. We reduce the replay buffer size from 1,000,000 to 500,000 to slightly reduce the memory footprint. All other hyperparameters are kept the same as in Yarats et al. (2021).

For the exactly equivariant and approximately equivariant models, we reduce the number of channels by $\sqrt{|G|}$ where $|G|$ is the order of the group to preserve roughly the same number of parameters as the non-equivariant model. We use $L = 1$ filters for the approximately equivariant model in all experiments.

RPP contains both the non-equivariant layers and exactly equivariant layers and thus has roughly twice as many parameters as `ExactEquiv`. For the critic moving average speed τ, we use the default $\tau = 0.01$ for `CartPole` and `Reacher` and $\tau = 0.009$ for `Acrobot` and `Ball in Cup`.

The plots in Figure 3 show the mean reward of 10 episodes, evaluated every 20,000 environment steps. For the results in Table 1, we use $\sigma = 0.02$ for `Acrobot` and `Reacher` and $\sigma = 0.06$ for `CartPole` and `Ball in Cup`.

The continuous control experiments were run on single GPUs of different types. `Acrobot` was run on an Nvidia RTX 4090 and all other experiments were run on an Nvidia RTX 2080 Ti. We note that the wall clock time for training both exactly and approximately equivariant models is longer than that for a non equivariant model, even though they are generally more sample efficient. This is because equivariant neural networks often incur more overhead in implementation - for group convolutions, the kernel must be transformed and the outputs must be stacked and for steerable convolutions, the basis must be projected onto matrices at every forward pass.

C.2 Stock Trading

We formulate the stock trading problem as an MDP as described in Liu et al. (2018). The state consists of the cash balance c_t, the stock prices p_t^n, the number of shares in the current portfolio h_t^n, and other technical indicators i_t^n for time t stock $n \in \{1, \ldots, N\}$. The actions x_t^n are the number of stocks to buy and sell for each stock n and are bounded to $[-M, M]$ where M was set to 100. The reward r_t is the scaled difference in portfolio values between consecutive timesteps and we assume that the market dynamics are not affected by our trading. There is a small transaction cost $\epsilon^n = 0.001$ for every trade. Initially, the portfolio contains 0 shares and the cash balance is 1,000,000. This can be formulated as a constrained program as follows

$$
\begin{aligned}
\max \quad & \sum_t r_t \\
\text{s.t.} \quad & -M \le a_t^n \le M, && \forall n, t \\
& a_t^n \ge -h_t^n, && \forall n, t \\
& a_t^n \le \lfloor c_t / (p_t^n (2 + \epsilon^n)) \rfloor && \forall n, t \\
& c_t \ge 0 && \forall t \\
& c_{t+1} = c_t - \sum_n a_t^n p_t^n (1 + \epsilon^n) && \forall t \\
& h_{t+1}^n = h_t^n + a_t^n && \forall n, t \\
& r_{t+1} = (c_{t+1} - c_t) + \sum_n (p_{t+1}^n h_{t+1}^n - p_t^n h_t^n) && \forall t \\
& c_0 = 1{,}000{,}000 \\
& h_0^n = 0 && \forall n \\
& h_t^n \in \mathbb{Z}^+, a_t^n \in \mathbb{Z}, c_t \in \mathbb{R}^+.
\end{aligned}
$$

The financial data was pulled from Yahoo Finance (yfi, 1997) for the time period 2001-01-01 to 2024-07-01 (see Figure 9 for sample stock prices). We split the train, validation, and test data into time periods 2001-01-01-2019-01-01, 2019-01-01 - 2021-01-01, and 2021-01-01-2024-07-01, respectively. As historical stock prices and portfolio can be important for determining the action, we use the previous $H = 9$ timesteps for the state, unlike Liu et al. (2018).

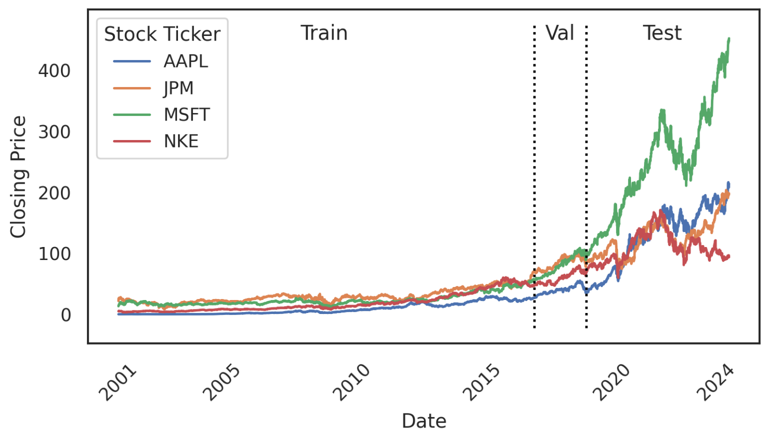

Figure 9: Sample stock trading data. We use a sliding window of the stock prices, current portfolio, cash balance, and other indicators as the state. The dataset is split into train/val/test as shown.

C.2.1 Training Details

or all models, we use 4 layers for the shared encoder, 1 layer for the actor, and 2 layers for the critic. The non equivariant model uses linear layers after flattening the input, while the exactly equivariant and approximately equivariant models use group convolutions and relaxed group convolutions with a kernel size of 5, respectively. We consider both temporal translations and temporal scale-translations. For scale-translation, we use separable group convolutions (Knigge et al., 2022) and use 3 scale factors $0.8, 0.98, 1.2$. We control the number of channels so that the total number of parameters is roughly equal to the non equivariant model. We use $L = 1$ filters for the approximately equivariant model in all experiments.

The stock trading experiments were run on a single Nvidia RTX 2080 Ti. All other hyperparameters are given in Table 3.

Table 3: Hyperparameters used for stock trading experiments

Hyperparameter	ApproxEquiv	ExactEquiv	NonEquiv
Batch size		64	
Learning rate		1e-4	
α		0.05	
τ		0.005	
Discount factor		0.99	
Hidden dim/channels	64	64	128
Encoder output dim/channels		256	