PARTIALLY EQUIVARIANT REINFORCEMENT LEARNING IN SYMMETRY-BREAKING ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Group symmetries provide a powerful inductive bias for reinforcement learning (RL), enabling efficient generalization across symmetric states and actions via group-invariant Markov Decision Processes (MDPs). However, real-world environments almost never realize fully group-invariant MDPs; dynamics, actuation limits, and reward design usually break symmetries, often only locally. Under group-invariant Bellman backups for such cases, local symmetry-breaking introduces errors that propagate across the entire state–action space, resulting in global value estimation errors. To address this, we introduce Partially group-Invariant MDP (PI-MDP), which selectively applies group-invariant or standard Bellman backups depending on where symmetry holds. This framework mitigates error propagation from locally broken symmetries while maintaining the benefits of equivariance, thereby enhancing sample efficiency and generalizability. Building on this framework, we present practical RL algorithms – Partially Equivariant (PE)-DQN for discrete control and PE-SAC for continuous control – that combine the benefits of equivariance with robustness to symmetry-breaking. Experiments across Grid-World, locomotion, and manipulation benchmarks demonstrate that PE-DQN and PE-SAC significantly outperform baselines, highlighting the importance of selective symmetry exploitation for robust and sample-efficient RL.

1 INTRODUCTION

Group symmetries provide a powerful inductive bias in machine learning, enabling models to generalize efficiently. In robotics and continuous control, leveraging equivariance has been shown to improve data efficiency in both behavior cloning (Zeng et al., 2021; Ryu et al., 2023; 2024; Wang et al., 2024; Tie et al., 2024; Huang et al., 2024; Zhao et al., 2025; Seo et al., 2023b;a; 2025a;b), where the data collection is costly, and reinforcement learning (RL) (Van der Pol et al., 2020; Kohler et al., 2024; Wang et al., 2022a;b; Tangri et al., 2024; Nguyen et al., 2023; Finzi et al., 2021a; Park et al., 2024), where exploration can be inefficient. Most existing equivariant RL methods are grounded in the notion of a group-invariant Markov Decision Process (MDP) (Wang et al., 2022b;c), where invariance of the reward and transition functions implies symmetry in the optimal policy.

In practice, however, these symmetry assumptions rarely hold exactly. Real-world environments introduce *symmetry-breaking factors* such as dynamics, actuation limits, or reward shaping. Under the Bellman backups based on the group-invariant MDP, even local violations of symmetry can introduce errors that propagate across the state–action space, leading to degraded value estimates, suboptimal policies, or even training failure. Prior works on approximate equivariance (Finzi et al., 2021a; Park et al., 2024) attempt to mitigate this challenge by relaxing equivariance globally, e.g., by modifying architectures to tolerate violations. While effective to some extent, these methods often lose the sample efficiency benefits of strict equivariance and can become unstable when symmetry-breaking is extensive, since equivariance is still applied indiscriminately across the entire space.

To overcome this limitation, we introduce the framework of the **Partially group-Invariant MDP (PI-MDP)**, which selectively applies the group-invariant structure only in regions where symmetry is preserved (Fig. 1). Our approach builds on the derivation that local symmetry-breaking leads to one-step backup errors that propagate globally. By routing updates to the standard updates under the true MDP, we limit the propagation of one-step backup errors across the space. In particular, we detect symmetry-breaking regions via dynamics model disagreement outliers between an

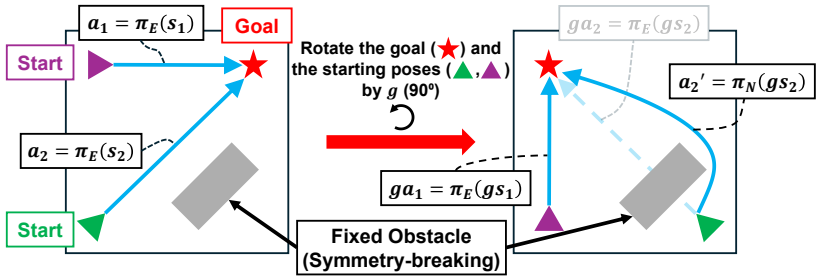


Figure 1: **Overview of partial equivariance in reinforcement learning.** Equivariant networks provide strong inductive bias and sample efficiency in environments with exact symmetry. **Left:** In the symmetric case, the equivariant policy π_E exploits this structure and learns an optimal action a to reach the goal. **Right:** When the agent and goal are rotated by 90° but the obstacle remains fixed, the symmetry is broken. An exactly equivariant policy is forced to output the rotated action ga, which is invalid due to the obstacle in some cases, thereby corrupting training. Our framework introduces a gating function λ_ζ that detects such symmetry-breaking and activates the non-equivariant policy π_N, preserving robustness while retaining the sample efficiency benefits of equivariance in symmetric regions.

equivariant and an unconstrained one-step dynamics model, and apply standard rather than equivariant updates on those outliers while retaining equivariance elsewhere. Building on this framework, we develop practical reinforcement learning algorithms for both discrete and continuous control that retain the benefits of equivariance in symmetric regions while remaining robust to substantial symmetry-breaking. The conceptual overview of our approach is depicted in Fig. 1.

The contributions of our work are summarized as follows: 1) We analyze how local symmetry violations induce global value error via one-step backups, clarifying when selective symmetry is beneficial. 2) We introduce the Partially group-Invariant MDP (PI-MDP) and a practical RL formulation that uses equivariance where symmetry holds and falls back to standard updates where it breaks. 3) Across state-based discrete and continuous control experiments, we show that our method retains the sample efficiency gains of equivariance in symmetric regions and remains robust as symmetry-breaking increases, outperforming strict and approximate-equivariant baselines.

2 Related Work

Group equivariance in continuous control. Recent works have applied group equivariance to imitation learning and classical control (Zeng et al., 2021; Ryu et al., 2023; 2024; Wang et al., 2024; Tie et al., 2024; Huang et al., 2024; Zhao et al., 2025; Seo et al., 2023b;a; 2025a), demonstrating high data efficiency and generalization over baseline models. Parallel efforts have investigated group equivariance in reinforcement learning (RL) (Van der Pol et al., 2020; Kohler et al., 2024; Wang et al., 2022a;b; Tangri et al., 2024; Nguyen et al., 2023), showing improved sample efficiency compared to the conventional RL approaches. However, the effectiveness of equivariant RL remains limited in more general settings, such as robotic control tasks, where inherent symmetry-breaking often arises from factors including occlusions, environmental asymmetries, kinematic singularities, and complex dynamics.

Approximate equivariance. Recent studies have proposed relaxing strict group equivariance to handle symmetry breaking in data (Finzi et al., 2021a; Park et al., 2024; Wang et al., 2022d; Romero & Lohit, 2022; van der Ouderaa et al., 2022; Hofgard et al., 2024). Such approaches introduce approximate equivariance, enabling models to remain robust when exact symmetries do not hold. In reinforcement learning, approximate equivariant architectures have also shown improved robustness and efficiency against symmetry-breaking (Finzi et al., 2021a; Park et al., 2024). For instance, Finzi et al. (2021a) introduced residual pathways to the equivariant linear layers, while Wang et al. (2022d) proposed a relaxed equivariant convolutional layer with expanded kernel parameterizations, which were later adopted in the RL setting by Park et al. (2024). However, these methods primarily focus

on global relaxations of equivariance at the representation level. In contrast, our approach addresses symmetry-breaking by minimizing local equivariance errors during the Bellman backup, thereby preventing their global propagation through value updates.

3 PRELIMINARIES

Reinforcement learning. We consider a Markov decision process (MDP) defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $P(\cdot \mid s, a)$ is the transition kernel, and $\gamma \in (0, 1)$ is the discount factor. The agent learns a policy π to maximize the expected return, $J = \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{\infty} \gamma^t r_t \,\Big|\, s_0 = s, a_0 = a\Big]$. The Bellman operator under a policy π is $(\mathcal{T}^\pi Q)(s, a) = R(s, a) + \gamma\, \mathbb{E}_{s' \sim P(\cdot|s,a)}\Big[\mathbb{E}_{a' \sim \pi(\cdot|s')}[Q(s', a')]\Big]$, while the optimal (hard) Bellman operator is $(\mathcal{T}Q)(s, a) = R(s, a) + \gamma\, \mathbb{E}_{s' \sim P(\cdot|s,a)}\big[\max_{a'} Q(s', a')\big]$.

Group equivariance. A *symmetry* is a transformation that preserves certain properties of a system (Bronstein et al., 2021). The set of all symmetries forms a **group**, which satisfies associativity, identity, inverses, and closure. A **group representation** is a homomorphism $\rho : G \to GL(n)$ that maps each group element $g \in G$ to an invertible $n \times n$ matrix. A function $f : X \to Y$ is **equivariant** if $\rho_Y(g) f(x) = f(\rho_X(g)x), \quad \forall g \in G, x \in X$, where ρ_X and ρ_Y are the group representations acting on X and Y respectively. If instead $f(x) = f(\rho_X(g)x)$, the function is called **group-invariant**. With a slight abuse of notation, we will often write g directly for its action on the relevant space (state, action, or next state).

Group-invariant MDP. A group-invariant MDP (Wang et al., 2022b;c) is an abstract MDP based on MDP homomorphisms (Ravindran & Barto, 2001; 2004), denoted as $\mathcal{M}_G(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. The optimal policy and optimal Q-function of the original MDP are recoverable from the abstract MDP provided the reward and transition kernel are group-invariant:

$$R(s, a) = R(gs, ga), \quad P(s' \mid s, a) = P(gs' \mid gs, ga), \quad \forall g \in G.$$

4 SYMMETRY-BREAKING IN GROUP-INVARIANT MDPS

Most equivariant RL approaches assume the existence of a group-invariant MDP (Sec. 3) (Wang et al., 2022c;b; Mondal et al., 2022; Van der Pol et al., 2020; Tangri et al., 2024). However, many continuous control tasks (e.g., robotics) violate these assumptions in certain regions of the state–action space. We begin by analyzing how such **symmetry-breaking** perturbs Bellman backups and subsequently propagates into the learned value function.

Let $\mathcal{M}_N(\mathcal{S}, \mathcal{A}, R_N, P_N, \gamma)$ denote the standard environment, and let $\mathcal{M}_E(\mathcal{S}, \mathcal{A}, R_E, P_E, \gamma)$ be a group-invariant MDP defined on the same spaces. To construct such a group invariant MDP from $\mathcal{M}_N$, we average the original rewards and dynamics over the symmetry group G:

$$R_E(s, a) = \int_G R_N(s, a)\, d\mu(g) \quad P_E(s'|s, a) = \int_G P_N(gs'|gs, ga)\, d\mu(g),$$

where $d\mu(g)$ is the normalized Haar measure on G (uniform measure for finite groups). This averaging ensures that R_E and P_E satisfy the group-invariance condition, thereby making $\mathcal{M}_E$ the canonical group-invariant approximation of $\mathcal{M}_N$. For $(s, a) \in \mathcal{S} \times \mathcal{A}$, define pointwise discrepancies

$$\begin{aligned} \epsilon_R(s, a) &:= |R_N(s, a) - R_E(s, a)|, \\ \epsilon_P(s, a) &:= \tfrac{1}{2} \int_{\mathcal{S}} \big|P_N(s' \mid s, a) - P_E(s' \mid s, a)\big|\, ds', \end{aligned} \tag{1}$$

where ϵ_R is the absolute reward difference and ϵ_P is the total-variation distance between next-state kernels. Let $\mathcal{T}_i$ denote the Bellman optimality operator in MDP $i \in \{N, E\}$. Assume rewards are bounded as $|R_i(s, a)| \le R_{\max}$ and discount $\gamma \in (0, 1)$. Define $V_{\max} := R_{\max}/(1 - \gamma)$.

We analyze the effect on a single Bellman backup and the induced value function gap. The following result bounds the one-step discrepancy between the optimality operators via local reward and transition mismatches.

Lemma 1 (One-step Bellman error). *For any bounded Q and any $(s, a) \in \mathcal{S} \times \mathcal{A}$,*

$$\left|(\mathcal{T}_N Q)(s,a) - (\mathcal{T}_E Q)(s,a)\right| \leq \epsilon_R(s,a) + 2\gamma \left\|V_Q\right\|_\infty \epsilon_P(s,a).$$

Here $V_Q(s') = \max_{a'} Q(s', a')$ and $\|V_Q\|_\infty \leq \|Q\|_\infty$. If Q is an action–value function, then $\|Q\|_\infty \leq V_{\max}$, hence $\|V_Q\|_\infty \leq V_{\max}$ and we define the pointwise bound

$$\delta(s,a) := \epsilon_R(s,a) + 2\gamma V_{\max} \epsilon_P(s,a).$$

We next show that this local error lifts to a global bound on the optimal action–value functions via contraction.

Proposition 1 (Value-function gap). *Let Q_i^* be the optimal action–value function in MDP i. Then*

$$\|Q_N^* - Q_E^*\|_\infty \leq \frac{1}{1-\gamma} \sup_{s,a} \delta(s,a).$$

The proofs of Lemma 1 and Proposition 1 are provided in Appendix A.1

Intuition. Local symmetry-breaking introduces a one-step Bellman backup error $\delta(s, a)$ which propagates through repeated backups and is amplified by the factor $(1-\gamma)^{-1}$ due to contraction. This results in a global deviation bounded by $\frac{1}{1-\gamma} \sup_{s,a} \delta(s, a)$, which can cause suboptimal policies or unstable training. We visualize this propagation in a Grid-World example, and show that a strictly equivariant policy can fail to learn (Appendix D). Prior works mitigate such errors with global relaxations (Finzi et al., 2021a; Park et al., 2024), whereas our approach employs **local** corrections that are less conservative and effective when symmetry holds only piecewise.

5 Partial Group-Invariance in Markov Decision Processes

In what follows, we present an efficient method for handling local symmetry-breaking. Specifically, we propose a **Partially group-Invariant MDP (PI-MDP)** that interpolates, for each state–action pair, between a group-invariant MDP and the true environment.

5.1 Partially group-Invariant MDP

Definition 1 (PI-MDP). *Let the true MDP be $\mathcal{M}_N = (\mathcal{S}, \mathcal{A}, R_N, P_N, \gamma)$ and the group-invariant MDP be $\mathcal{M}_E = (\mathcal{S}, \mathcal{A}, R_E, P_E, \gamma)$, sharing the same $(\mathcal{S}, \mathcal{A}, \gamma)$. Define a Partially group-Invariant MDP (PI-MDP) $\mathcal{M}_H = (\mathcal{S}, \mathcal{A}, R_H, P_H, \lambda, \gamma)$ with a measurable gating function $\lambda : \mathcal{S} \times \mathcal{A} \to [0, 1]$,*

$$R_H(s,a) := (1 - \lambda(s,a))\, R_E(s,a) + \lambda(s,a)\, R_N(s,a),$$
$$P_H(\cdot \mid s,a) := (1 - \lambda(s,a))\, P_E(\cdot \mid s,a) + \lambda(s,a)\, P_N(\cdot \mid s,a).$$

Since $0 \leq \lambda(s, a) \leq 1$ for all (s, a) and both (R_E, P_E) and (R_N, P_N) are valid, (R_H, P_H) defines a valid MDP.

Remark 1 (Hard gating). *When $\lambda(s, a) \in \{0, 1\}$, the PI-MDP routes pointwise to (R_E, P_E) on symmetric pairs and (R_N, P_N) otherwise. All results below hold for any measurable gating function $\lambda : \mathcal{S} \times \mathcal{A} \to [0, 1]$. In our algorithms, we use hard gating for simplicity and interpretability.*

We first characterize the partially group-invariant optimality operator induced by the gating function.

Theorem 1 (Partially group-invariant optimality operator). *Let $\mathcal{T}_i$ denote the (hard) Bellman optimality operator in MDP $i \in \{E, N, H\}$, $(\mathcal{T}_i Q)(s,a) = R_i(s,a) + \gamma\, \mathbb{E}_{s' \sim P_i(\cdot|s,a)}[\max_{a'} Q(s', a')]$. For any bounded $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and all (s, a),*

$$(\mathcal{T}_H Q)(s,a) = (1 - \lambda(s,a))\, (\mathcal{T}_E Q)(s,a) + \lambda(s,a)\, (\mathcal{T}_N Q)(s,a). \tag{2}$$

If $|R_E|, |R_N| \leq R_{\max}$ and $\gamma \in (0, 1)$, then $\mathcal{T}_H$ is a γ-contraction and admits a unique fixed point Q_H^*.

We next bound the deviation of the fixed point from the true optimum.

Corollary 1 (Proximity bound)**.** *Let Q_N^* be the optimal action–value of the true MDP $\mathcal{M}_N$, and let $V_N(s) = \max_a Q_N^*(s,a)$. Then*

$$\|Q_H^* - Q_N^*\|_\infty \;\le\; \frac{1}{1-\gamma} \Big\| (1-\lambda) \left[\epsilon_R(s,a) + 2\gamma \epsilon_P(s,a) V_{\max} \right] \Big\|_\infty . \tag{3}$$

Moreover, the right-hand side of Eq. (3) is zero whenever, at every (s,a), either $\lambda(s,a) = 1$ (the gating function routes to the true MDP) or the group-invariant MDP coincides with the true MDP at (s,a), that is, $R_E(s,a) = R_N(s,a)$ and $P_E(\cdot \mid s,a) = P_N(\cdot \mid s,a)$. Consequently, symmetric pairs contribute zero via MDP coincidence, and symmetry-breaking pairs contribute zero when λ correctly gates to 1. The proofs of Theorem 1 and Corollary 1 can be found in Appendix A.2.

Intuition. By gating the reward and transition kernels, the PI-MDP is itself a valid MDP. Its optimality operator satisfies the affinity identity in Eq. (2). Since a convex combination of γ-contraction is again a γ-contraction, $\mathcal{T}_H$ admits a unique fixed point Q_H^*. Corollary 1 bounds the deviation from the true optimum: the gap $\|Q_H^* - Q_N^*\|_\infty$ is controlled by the gated mismatch term on the right-hand side of Eq. (3), scaled by $(1-\gamma)^{-1}$. The bound *vanishes* whenever, at every (s,a), either the gating function routes to the true MDP ($\lambda = 1$) or the group-invariant and true MDPs coincide. Thus, when λ correctly localizes symmetry-breaking, Q_H^* closely tracks Q_N^* while reverting to the group-invariant MDP where symmetry holds. We provide the extension of the PI-MDP to the entropy-regularized (soft) setting in Appendix A.3.

6 Partially Equivariant Reinforcement Learning

This section introduces partially equivariant reinforcement learning (Algorithm 1) for the PI-MDP setting (Sec. 5.1). We (i) learn a gating function $\lambda(s,a) \in [0,1]$ that localizes symmetry breaking, and (ii) couple λ to equivariant and unconstrained value/policy heads.

6.1 Learning $\lambda(s,a)$ via disagreement supervision

By Corollary 1, the value gap vanishes when $\lambda(s,a) = 1$ on symmetry-breaking pairs and $\lambda(s,a) = 0$ where the proxy and true MDPs coincide (assuming an oracle binary gate under local symmetry). To approximate this behavior, we train a gating function $\lambda_\omega(s,a) \in [0,1]$ using the *disagreement* between two one-step predictors: an equivariant regressor $\hat{P}_E : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ constrained to respect $\mathcal{M}_E$, and an unconstrained regressor $\hat{P}_N : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ trained freely on data from $\mathcal{M}_N$.

Both predictors are trained on transitions (s,a,s') to minimize mean-squared error (MSE) on the state increment $\Delta s := s' - s$. In regions where the environment is group-invariant, $\hat{P}_E$ is consistent with the group-averaged dynamics P_E, which coincide with the true dynamics P_N. Therefore, the predictor disagreement

$$d(s,a) = \|\Delta \hat{s}_E(s,a) - \Delta \hat{s}_N(s,a)\|_2^2$$

remains small. At symmetry-breaking pairs, however, the equivariance constraint is misspecified: $\hat{P}_E$ is forced toward the group-averaged surrogate P_E, while the unconstrained predictor $\hat{P}_N$ can track P_N. This induces a systematic discrepancy of order $\epsilon_P(s,a)$ in the learned predictions. Consequently, the prediction gap $\|\Delta \hat{s}_E - \Delta \hat{s}_N\|$ is biased upward in precisely those regions where (R_E, R_N) or (P_E, P_N) disagree, providing an indirect detector of symmetry-breaking via the disagreement $d(s,a)$. We assume those symmetry-breaking disagreements as outliers in the online distribution of $d(s,a)$. We label outliers with $y(s,a) \in \{0,1\}$ using an online detector (Appendix B.1) and train λ_ω with binary cross-entropy:

$$\mathcal{L}_\lambda(\omega) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[- y \log \lambda_\omega(s,a) - (1-y) \log(1 - \lambda_\omega(s,a)) \big], \tag{4}$$

where $\mathcal{D}$ is the replay buffer. The gating function is trained concurrently via Eq. (4). During each critic and actor update, we recompute and cache the binary gate $\tilde{\lambda}(s,a) \in \{0,1\}$ from the gating network's output $\lambda_\omega(s,a)$ (see implementation details in Appendix B.1). We treat $\tilde{\lambda}$ as constant within the update (stop-gradient).

Algorithm 1 Partially Equivariant Reinforcement Learning (PERL)

Require: Replay buffer $\mathcal{D}$, critics Q_E, Q_N, policies π_E, π_N
Require: Dynamics predictors $\hat{P}_E, \hat{P}_N$, gating functions $\lambda_\omega, \lambda_\zeta$, targets $\bar{Q}, \bar{\lambda}_\omega$
Initialize all networks
Initialize running statistics (μ, σ) for disagreement
for $t = 1$ to T **do**
  Sample $a_t \sim \pi_\phi(\cdot \mid s_t)$ ▷ gated policy from Eq. (6)
  Store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
  Train dynamics $\hat{P}_E$ and $\hat{P}_N$ to minimize the predictive loss $\mathcal{L}_{\text{dyn}}$ ▷ see App. B.1
  Compute disagreement $d(s,a) = D\big(\hat{P}_E(\cdot \mid s,a), \hat{P}_N(\cdot \mid s,a)\big)$ ▷ see Sec. 6.1, App. B.1
  Update the running statistics (μ, σ) over the disagreement $d(s,a)$
  Update λ_ω with BCE-loss (Eq. (4))
  Update λ_ζ with expectile regression loss (Eq. (7))
  Update the critics with the objective (Eq. (8))
  Update the actor with the objective (Eq. (9)) ▷ SAC only; DQN uses greedy $\arg\max$
  Soft update $\bar{Q}$ and $\bar{\lambda}_\omega$
end for

6.2 Partially Equivariant Reinforcement Learning

We couple the learned gating function to the critic and the actor, thereby implementing the PI-MDP framework under function approximation while training entirely in the true environment $\mathcal{M}_N$.

Gated value mixtures under the true MDP. We parameterize the critic as a gated mixture:

$$Q_\theta(s,a) \;=\; \big(1 - \lambda_\omega(s,a)\big)\, Q_{E,\theta}(s,a) \;+\; \lambda_\omega(s,a)\, Q_{N,\theta}(s,a), \tag{5}$$

where Q_E is an equivariant critic constrained by group symmetries and Q_N is an unconstrained critic with no symmetry bias. The gate $\lambda_\omega : \mathcal{S} \times \mathcal{A} \to [0,1]$ interpolates between the two networks. Conditioned on the binary gating $\tilde{\lambda}$ (cached per minibatch and used with stop-gradient), our TD-based critic (e.g., DQN, SAC) learns under $\mathcal{M}_N$ the best approximation within this mixed hypothesis class. With binary gating, the mixture reduces to a hard switch, activating either Q_E or Q_N depending on whether the state–action lies in a symmetric or symmetry-breaking region.

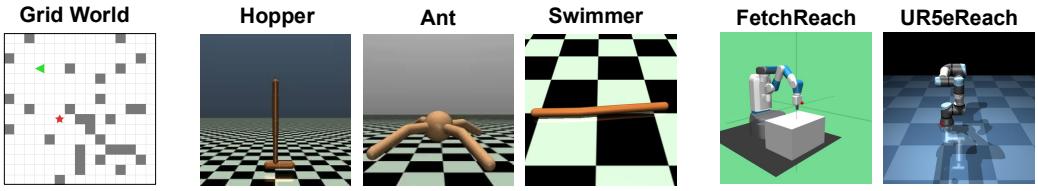
Idealized compatibility (binary oracle gating). If $\lambda(s,a) \in \{0,1\}$ *perfectly* separates symmetric from broken regions and, on symmetric regions, the averaged dynamics coincide $(P_E, R_E) = (P_N, R_N)$, then the partially group-invariant operator $\mathcal{T}_H$ is identical to the true operator $\mathcal{T}_N$. In this idealized case, our TD targets exactly match $(\mathcal{T}_H Q)(s,a)$ and the mixture recovers the interpolating solution in Theorem 1. This motivates the use of λ as a "local oracle" for symmetry-breaking. In practice, we approximate this oracle by the learned gating function λ_ω, producing binary decisions as described above.

Gated policy and actor gating function. For the policy, we employ a state-only gating function $\lambda_\zeta : \mathcal{S} \to [0,1]$ and define a product-of-experts (PoE) blend

$$\pi_\phi(\cdot \mid s) \;\propto\; \pi_{E,\phi}(\cdot \mid s)^{\,1-\lambda_\zeta(s)}\; \pi_{N,\phi}(\cdot \mid s)^{\,\lambda_\zeta(s)}. \tag{6}$$

This form naturally arises from SAC policy improvement: given the critic mixture $Q_\theta = (1 - \lambda_\omega)Q_E + \lambda_\omega Q_N$, the information projection in SAC yields a PoE between the energy models $\exp(Q_E/\alpha)$ and $\exp(Q_N/\alpha)$ (see Appendix A.4 for details). While a fully state–action gate in π would be theoretically appealing, it is intractable in practice because the normalization constant of Eq. (6) would depend on a. We therefore restrict to a state-only gate $\lambda_\zeta(s)$, aligned with the critic gating function via a conservative aggregation loss. This conservativeness is crucial: since symmetry-breaking may occur only for a subset of actions, $\lambda_\zeta(s)$ should activate whenever *any* action at state s is flagged by $\lambda_\omega(s,a)$. This conservative choice does not compromise optimality, as taking the maximum ensures that any critical symmetry-breaking is accounted for while leaving the optimal policy unchanged.

$$\mathcal{L}_\lambda(\zeta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[L_\tau\big(\lambda_\omega(s,a) - \lambda_\zeta(s)\big)\Big], \tag{7}$$



(a) Discrete space (b) Continuous space (locomotion) (c) Continuous space (manipulation)

Figure 2: **Benchmark environments.** We evaluate our method across both discrete and continuous control tasks under symmetry-breaking conditions. Specifically, we use the Grid-World environment for the discrete case, and locomotion and manipulation tasks for the continuous case.

where L_τ is the expectile loss (Kostrikov et al., 2021). Taking $\tau \to 1$ approximates the $\max_a$ operator, ensuring that $\lambda_\zeta(s)$ conservatively reflects the maximum symmetry-breaking signal across actions. During sampling the actions, we obtain a binary gate $\tilde{\lambda}_\zeta(s) \in \{0, 1\}$ by using $\lambda_\zeta(s)$ (details in Appendix B.2). Per sample, Eq. (6) thus collapses to a hard switch between π_E and π_N, retaining interpretability and computational tractability.

Training. We train Q_θ and π_ϕ using standard objectives from deep RL: DQN (Mnih et al., 2013) for value-based methods and SAC (Haarnoja et al., 2018) for actor–critic methods, substituting in our gated parameterizations. In this way, the partially equivariant framework is realized within standard off-the-shelf algorithms, while the gates λ_ω and λ_ζ provide adaptive control over when equivariance is exploited and when it is suppressed.

$$J_Q(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \tfrac{1}{2}\Big(Q_\theta(s,a) - r + \gamma \max_{a'} Q_{\bar{\theta}}(s',a')\Big)^2, \tag{8}$$

where $\bar{\theta}$ denotes target parameters and, $Q_\theta(s,a) = (1-\lambda_\omega(s,a))\, Q_{E,\theta}(s,a) + \lambda_\omega(s,a)\; Q_{N,\theta}(s,a)$.

$$J_\pi(\phi) = \mathbb{E}_{\substack{s\sim\mathcal{D} \\ \epsilon\sim\mathcal{N}(0,I)}}\Big[\alpha \log \pi_\phi(a \mid s) \;-\; \min_{i=1,2} Q_{\theta_i}(s,a)\Big], \qquad a = \tanh\big(g_\phi(s,\epsilon)\big). \tag{9}$$

where $\log \pi_\phi(a \mid s) = (1 - \lambda_\zeta(s))\; \log \pi_{E,\phi}(a \mid s) + \lambda_\zeta(s)\; \log \pi_{N,\phi}(a \mid s)$. Please refer to Algorithm 1 for the pseudocode, and Appendix B for more details.

7 Experiments

Our experiments aim to answer two main questions: (1) How does our method compare in terms of sample efficiency against the conventional RL and strictly equivariant methods? (2) How robust is our method to symmetry-breaking, relative to the state-of-the-art approximate equivariant approach?

7.1 Experimental Setup

We evaluate across three categories of environments: (1) a discrete Grid-World for intuitive analysis, and (2) continuous-control locomotion benchmarks in MuJoCo with state-based observations (Brockman et al., 2016), and (3) robotic manipulation tasks adapted from the Fetch manipulation (Plappert et al., 2018) and a DeepMind Control Suite (DMC) (Tassa et al., 2018)-based UR5e manipulator (Chuang, 2023). We compare our DQN-based (PE-DQN) and SAC-based (PE-SAC) methods against vanilla RL, strictly equivariant method, and one of the state-of-the-art models among approximate equivariant RL approaches, RPP (Finzi et al., 2021a). All experiments use state-based observations and continuous control (except for Grid-World), and we report mean performance with standard error over five random seeds. Fig. 2 provides an overview of environments, with additional details in Appendix C.

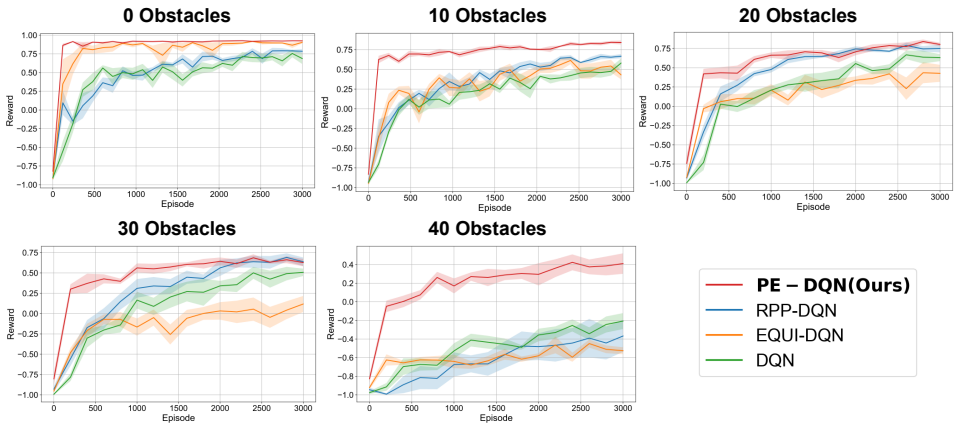


Figure 3: **Performance comparison in the discrete space (Grid-World) environment.** We evaluate the average performance over 3000 episodes with five random seeds. Shaded regions denote standard error. We vary the number of obstacles, which act as symmetry-breaking factors. PE-DQN consistently outperforms the baselines, and the performance gap widens as symmetry-breaking increases, demonstrating both robustness and sample efficiency.

Grid-World. We use a discrete C_4 symmetric Grid-World as a lightweight testbed for analyzing robustness to symmetry-breaking. Symmetry-breaking is introduced by placing obstacles that disrupt transitions implied by rotation symmetry, and we vary the number of obstacles to control the degree of breaking. This environment allows us to clearly examine how PE-DQN adapts as the extent of symmetry-breaking increases.

Locomotion. We evaluate on continuous-control MuJoCo benchmarks using the same symmetry specifications as RPP (Finzi et al., 2021a), which include both exact and approximate symmetries. This setting allows us to test whether PE-SAC can extend the sample-efficiency benefits of equivariance from discrete Grid-World to challenging continuous-control tasks, while remaining robust to symmetry-breaking factors such as external forces or reward perturbations. All baselines are trained with SAC.

Manipulation. We evaluate in manipulation settings, considering two reach tasks with $SO(3)$ symmetry. Fetch Reach serves as a simpler case, where the end-effector is constrained perpendicular to the floor and the goal is specified only by (x, y, z) position. In contrast, UR5e Reach allows free end-effector orientation in addition to position, with a goal specified as an $SE(3)$ pose that includes both position and orientation. The inclusion of orientation control makes the task more representative of real-world manipulators. This progression from Fetch to UR5e enables us to test whether PE-SAC scales from constrained to more realistic manipulation scenarios. Symmetry-breaking naturally arises from collisions, floor contacts, and kinematic singularities. All methods use the same SAC backbone for comparability.

7.2 Analysis

In Fig. 3, we present the reward graphs from the Grid-World experiment across varying obstacle counts. When no symmetry-breaking factors are present, PE-DQN converges to $\lambda \approx 0$, effectively using only the equivariant Q-network. In this strictly symmetric setting, performance matches the exact equivariant approach and surpasses both vanilla DQN and RPP. The lag of RPP arises because its residual non-equivariant pathways cannot be completely suppressed, introducing noise that slows convergence. As obstacle counts increase, vanilla DQN and RPP degrade markedly, while PE-DQN maintains high performance, demonstrating robustness to localized symmetry-breaking and aligning with the theoretical prediction that the value gap is controlled by ϵ_R, ϵ_P under partial invariance.

In Fig. 4, we show results on locomotion and manipulation tasks. PE-SAC attains higher sample efficiency in Hopper and Ant, while matching performance in Swimmer and Fetch Reach. These differences reflect how the gate adapts to symmetry: in Hopper, equivariance speeds early learning

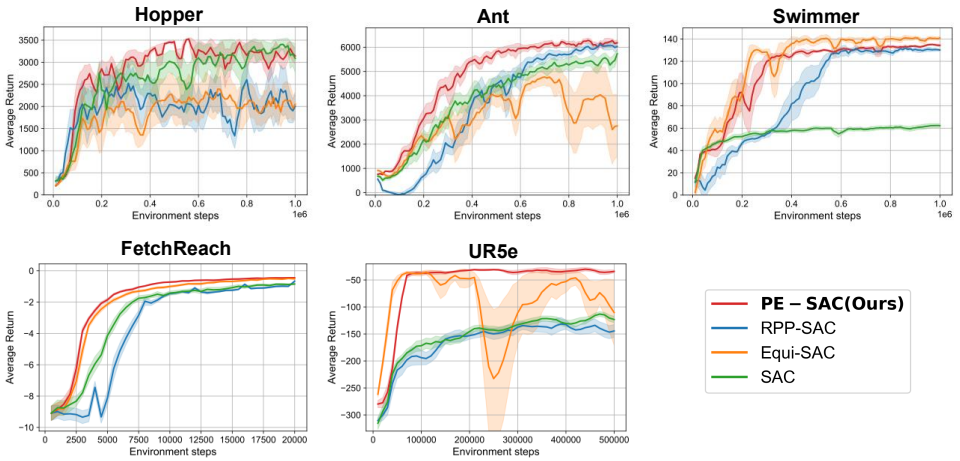


Figure 4: **Performance comparison in the continuous space environments.** Results are averaged over 1M training steps in MuJoCo tasks and 20k steps in the Fetch environment, using five random seeds. Shaded regions denote standard error. For RPP (Finzi et al., 2021a), we re-ran the official code. Discrepancies with the reported numbers arise because RPP reports "max over steps" rather than average performance. PE-SAC consistently outperforms all baselines across these tasks.

but PE-SAC later relies on Q_N as joint limits break symmetry; in Ant, fourfold symmetry aids exploration but leg collisions cause exact-equivariant SAC to fail, where PE-SAC remains stable. Swimmer is nearly perfectly symmetric, so both methods converge similarly, while Fetch Reach is dominated by goal asymmetries, offering little benefit from equivariance.

The UR5e manipulator, with realistic dynamics and unconstrained end-effector orientation, highlights robustness under significant symmetry-breaking. Exact-equivariant SAC initially learned efficiently but soon collapsed as non-symmetric transitions violated equivariance assumptions, destabilizing training. In contrast, PE-SAC shifted to the non-equivariant head in these regions, maintaining stable and sample-efficient performance—a robustness not matched by exact-equivariant or purely non-equivariant baselines.

Overall, these results confirm that selectively mitigating local equivariance errors enables our method to retain the benefits of symmetry exploitation while avoiding its pitfalls, yielding both sample efficiency and robustness across a spectrum of symmetric and symmetry-broken environments.

8 Conclusion

In this work, we introduced the **PI-MDP**, a framework that mitigates global error propagation from local symmetry-breaking. Building on this foundation, we developed **Partially Equivariant RL (PE-RL)** algorithms—PE-DQN for discrete control and PE-SAC for continuous control—that consistently improved sample efficiency and robustness over conventional RL, exact-equivariant methods, and approximate baselines.

The main limitation is additional computation: training requires auxiliary models, increasing wall-clock cost (roughly 2× in Grid-World and up to 5× in MuJoCo). In environments with global symmetry-breaking (e.g., gravity) or reward-only symmetry-breaking, the gate defaults to non-equivariant networks, reducing the method to standard RL. Still, in scenarios with localized symmetry-breaking—common in practice—the method yields clear benefits.

Future work includes extending PE-RL to pixel-based control, advancing the practicality of symmetry-aware reinforcement learning for real-world continuous control.

REFERENCES

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Tony F Chan, Gene H Golub, and Randall J LeVeque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, 1983.

Ian Chuang. Manipulator-mujoco. `https://github.com/ian-chuang/Manipulator-Mujoco`, 2023.

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

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*, pp. 3318–3328. PMLR, 2021b.

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*, pp. 1861–1870. PMLR, 2018.

Elyssa Hofgard, Rui Wang, Robin Walters, and Tess Smidt. Relaxed equivariant graph neural networks. *arXiv preprint arXiv:2407.20471*, 2024.

Haojie Huang, Owen Howell, Dian Wang, Xupeng Zhu, Robin Walters, and Robert Platt. Fourier transporter: Bi-equivariant robotic manipulation in 3d. *arXiv preprint arXiv:2401.12046*, 2024.

Diederik Kingma, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International conference on learning representations (ICLR)*, volume 5. California;, 2015.

Colin Kohler, Anuj Shrivatsav Srikanth, Eshan Arora, and Robert Platt. Symmetric models for visual force policy learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3101–3107. IEEE, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

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*, pp. 15908–15926. PMLR, 2022.

Hai Huu Nguyen, Andrea Baisero, David Klee, Dian Wang, Robert Platt, and Christopher Amato. Equivariant reinforcement learning under partial observability. In *Conference on Robot Learning*, pp. 3309–3320. PMLR, 2023.

Jung Yeon Park, Sujay Bhatt, Sihan Zeng, Lawson LS Wong, Alec Koppel, Sumitra Ganesh, and Robin Walters. Approximate equivariance in reinforcement learning. *arXiv preprint arXiv:2411.04225*, 2024.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

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

Balaraman Ravindran and Andrew G Barto. Approximate homomorphisms: A framework for non-exact minimization in markov decision processes. 2004.

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

Hyunwoo Ryu, Hong-in Lee, Jeong-Hoon Lee, and Jongeun Choi. Equivariant descriptor fields: SE(3)-equivariant energy-based models for end-to-end visual robotic manipulation learning. *International conference on learning representations (ICLR)*, 2023.

Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on SE(3) for visual robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18007–18018, 2024.

Joohwan Seo, Nikhil Potu Surya Prakash, Alexander Rose, Jongeun Choi, and Roberto Horowitz. Geometric impedance control on SE(3) for robotic manipulators. *IFAC-PapersOnLine*, 56(2): 276–283, 2023a.

Joohwan Seo, Nikhil PS Prakash, Xiang Zhang, Changhao Wang, Jongeun Choi, Masayoshi Tomizuka, and Roberto Horowitz. Contact-rich SE(3)-equivariant robot manipulation task learning via geometric impedance control. *IEEE Robotics and Automation Letters*, 9(2):1508–1515, 2023b.

Joohwan Seo, Arvind Kruthiventy, Soomi Lee, Megan Teng, Xiang Zhang, Seoyeon Choi, Jongeun Choi, and Roberto Horowitz. Equicontact: A hierarchical SE(3) vision-to-force equivariant policy for spatially generalizable contact-rich tasks. *arXiv preprint arXiv:2507.10961*, 2025a.

Joohwan Seo, Soochul Yoo, Junwoo Chang, Hyunseok An, Hyunwoo Ryu, Soomi Lee, Arvind Kruthiventy, Jongeun Choi, and Roberto Horowitz. SE(3)-equivariant robot learning and control: A tutorial survey. *International Journal of Control, Automation and Systems*, 23(5):1271–1306, 2025b.

Arsh Tangri, Ondrej Biza, Dian Wang, David Klee, Owen Howell, and Robert Platt. Equivariant offline reinforcement learning. *arXiv preprint arXiv:2406.13961*, 2024.

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:1801.00690*, 2018.

Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. Et-seed: Efficient trajectory-level SE(3) equivariant diffusion policy. *arXiv preprint arXiv:2411.03990*, 2024.

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, Mingxi Jia, Xupeng Zhu, Robin Walters, and Robert Platt. On-robot learning with equivariant models. *arXiv preprint arXiv:2203.04923*, 2022a.

Dian Wang, Robin Walters, and Robert Platt. SO(2)-equivariant reinforcement learning. *arXiv preprint arXiv:2203.04439*, 2022b.

Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant q learning in spatial action spaces. In *Conference on Robot Learning*, pp. 1713–1723. PMLR, 2022c.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

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

Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1911.08251.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pp. 726–747. PMLR, 2021.

Haibo Zhao, Dian Wang, Yizhe Zhu, Xupeng Zhu, Owen Lewis Howell, Linfeng Zhao, Yaoyao Qian, Robin Walters, and Robert Platt. Hierarchical equivariant policy via frame transfer. In *Forty-second International Conference on Machine Learning*, 2025.

A Theoretical Proofs

A.1 Proof of Lemma 1 and Proposition 1

Lemma 1 (One-step Bellman error)**.** *For any bounded Q and any $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\left|(\mathcal{T}_N Q)(s,a) - (\mathcal{T}_E Q)(s,a)\right| \;\le\; \epsilon_R(s,a) \;+\; 2\gamma\,\|V_Q\|_\infty\,\epsilon_P(s,a).$$

Proof. By the triangle inequality,

$$\begin{aligned}
&\left|(\mathcal{T}_N Q)(s,a) - (\mathcal{T}_E Q)(s,a)\right| \\
&= \left|R_N(s,a) - R_E(s,a) + \gamma\big(\mathbb{E}_{s'\sim P_N(\cdot|s,a)}[V_Q(s')] - \mathbb{E}_{s'\sim P_E(\cdot|s,a)}[V_Q(s')]\big)\right| \\
&\le\; \epsilon_R(s,a) + \gamma\left|\mathbb{E}_{P_N}[V_Q] - \mathbb{E}_{P_E}[V_Q]\right|.
\end{aligned}$$

Using the total-variation inequality $\left|\mathbb{E}_P[f] - \mathbb{E}_Q[f]\right| \le 2\|f\|_\infty \,\mathrm{TV}(P,Q)$ with $\mathrm{TV}(P_N,P_E) = \epsilon_P(s,a)$ and $f = V_Q$,

$$\left|\mathbb{E}_{P_N}[V_Q] - \mathbb{E}_{P_E}[V_Q]\right| \le 2\|V_Q\|_\infty\,\epsilon_P(s,a).$$

Combining the bounds leads to the lemma. □

Proposition 1. *(Value-function gap). Let Q_i^* be the optimal action–value function in MDP i. Then,*

$$\|Q_N^* - Q_E^*\|_\infty \le \frac{1}{1-\gamma}\sup_{s,a}\delta(s,a).$$

Proof. Since $Q_N^* = \mathcal{T}_N Q_N^*$ and $Q_E^* = \mathcal{T}_E Q_E^*$, we have

$$\|Q_N^* - Q_E^*\|_\infty = \|\mathcal{T}_N Q_N^* - \mathcal{T}_E Q_E^*\|_\infty \le \|\mathcal{T}_N Q_N^* - \mathcal{T}_N Q_E^*\|_\infty + \|\mathcal{T}_N Q_E^* - \mathcal{T}_E Q_E^*\|_\infty.$$

The Bellman optimality operator is a γ-contraction in the sup norm, so

$$\|\mathcal{T}_N Q_N^* - \mathcal{T}_N Q_E^*\|_\infty \le \gamma\|Q_N^* - Q_E^*\|_\infty.$$

By Lemma 1 applied with $Q = Q_E^*$ and the bounded $\|V_{Q_E^*}\|_\infty \le V_{\max}$, we have

$$\|\mathcal{T}_N Q_E^* - \mathcal{T}_E Q_E^*\|_\infty \le \sup_{s,a}\delta(s,a).$$

Combining the two inequalities gives

$$\|Q_N^* - Q_E^*\|_\infty \le \gamma\|Q_N^* - Q_E^*\|_\infty + \sup_{s,a}\delta(s,a).$$

Rearranging results in

$$\|Q_N^* - Q_E^*\|_\infty \le \frac{1}{1-\gamma}\sup_{s,a}\delta(s,a),$$

which completes the proof. □

A.2 Proof of Theorem 1 and Corollary 1

Theorem 1 (Partially group-invariant optimality operator)**.** *Let $\mathcal{T}_i$ denote the (hard) Bellman optimality operator in MDP $i \in \{E,N,H\}$, $(\mathcal{T}_i Q)(s,a) = R_i(s,a) + \gamma\,\mathbb{E}_{s'\sim P_i(\cdot|s,a)}[\max_{a'} Q(s',a')]$. For any bounded $Q : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ and all (s,a),*

$$(\mathcal{T}_H Q)(s,a) = (1-\lambda(s,a))\,(\mathcal{T}_E Q)(s,a) + \lambda(s,a)\,(\mathcal{T}_N Q)(s,a). \tag{10}$$

If $|R_E|, |R_N| \le R_{\max}$ and $\gamma \in (0,1)$, then $\mathcal{T}_H$ is a γ-contraction and admits a unique fixed point Q_H^*.

Proof. Identity Eq. (10). By Definition 1, for any (s, a),

$$(\mathcal{T}_H Q)(s,a) = (1-\lambda)\Big(R_E(s,a) + \gamma\, \mathbb{E}_{s'\sim P_E(\cdot|s,a)}\big[\max_{a'} Q(s',a')\big]\Big) + \lambda\Big(R_N(s,a) + \gamma\, \mathbb{E}_{s'\sim P_N(\cdot|s,a)}\big[\max_{a'} Q(s',a')\big]\Big),$$

which equals $(1-\lambda)\mathcal{T}_E Q + \lambda \mathcal{T}_N Q$ pointwise.

Contraction. Let Q_1, Q_2 be bounded. Using Eq. (2) and that $\mathcal{T}_E, \mathcal{T}_N$ are γ-contractions,

$$\begin{aligned}
&\big|\mathcal{T}_H Q_1(s,a) - \mathcal{T}_H Q_2(s,a)\big| \\
&= \Big|(1-\lambda(s,a))\big(\mathcal{T}_E Q_1(s,a) - \mathcal{T}_E Q_2(s,a)\big) + \lambda(s,a)\big(\mathcal{T}_N Q_1(s,a) - \mathcal{T}_N Q_2(s,a)\big)\Big| \\
&\le (1-\lambda(s,a))\,\|\mathcal{T}_E Q_1 - \mathcal{T}_E Q_2\|_\infty + \lambda(s,a)\,\|\mathcal{T}_N Q_1 - \mathcal{T}_N Q_2\|_\infty \\
&\le \gamma\,\|Q_1 - Q_2\|_\infty.
\end{aligned}$$

Taking the supremum over (s, a) gives

$$\|\mathcal{T}_H Q_1 - \mathcal{T}_H Q_2\|_\infty \le \gamma\,\|Q_1 - Q_2\|_\infty.$$

Bounded rewards ensure $\mathcal{T}_H$ maps bounded Q into bounded Q. By Banach's fixed point theorem, $\mathcal{T}_H$ has a unique fixed point Q_H^*. □

Corollary 1 (Proximity bound)**.** *Let Q_N^* be the optimal action–value of the true MDP $\mathcal{M}_N$, and let $V_N(s) = \max_a Q_N^*(s,a)$. Then*

$$\|Q_H^* - Q_N^*\|_\infty \;\le\; \frac{1}{1-\gamma}\,\Big\|(1-\lambda)\,\big[\epsilon_R(s,a) + 2\gamma\epsilon_P(s,a)V_{\max}\big]\Big\|_\infty. \tag{11}$$

Proof.

$$\begin{aligned}
\|Q_H^* - Q_N^*\|_\infty &= \|\mathcal{T}_H Q_H^* - \mathcal{T}_N Q_N^*\|_\infty \\
&\le \|\mathcal{T}_H Q_H^* - \mathcal{T}_H Q_N^*\|_\infty + \|\mathcal{T}_H Q_N^* - \mathcal{T}_N Q_N^*\|_\infty \\
&\le \gamma\|Q_H^* - Q_N^*\|_\infty + \|(1-\lambda)\,(\mathcal{T}_E Q_N^* - \mathcal{T}_N Q_N^*)\|_\infty.
\end{aligned}$$

Expanding pointwise,

$$(\mathcal{T}_E Q_N^* - \mathcal{T}_N Q_N^*)(s,a) = (R_E - R_N)(s,a) + \gamma\Big(\mathbb{E}_{s'\sim P_E(\cdot|s,a)}[V_N(s')] - \mathbb{E}_{s'\sim P_N(\cdot|s,a)}[V_N(s')]\Big).$$

By the definition of total variation distance,

$$\big|\mathbb{E}_{P_E}[V_N] - \mathbb{E}_{P_N}[V_N]\big| \;\le\; 2\,\epsilon_P(s,a)\,V_{\max},$$

where $V_{\max} = R_{\max}/(1-\gamma)$, $\epsilon_P(s,a) = \frac{1}{2}\int_{\mathcal{S}} |P_N(s'|s,a) - P_E(s'|s,a)|\, ds'$ (Eq. (1)).

Rearranging gives Eq. (11) □

A.3 Partial Group-Invariance in Soft MDPs

Since the PI-MDP defined in Definition 1 is a valid MDP, the soft policy iteration framework (Haarnoja et al., 2018) applies unchanged. We show the evaluation identity and the standard improvement step for completeness.

Policy evaluation. For a fixed policy π, define the soft state value $V_Q^\pi(S) := \mathbb{E}_{a\sim\pi(\cdot|s)}\big[\,Q(s,a) - \alpha\log\pi(a\mid s)\,\big]$ with temperature $\alpha > 0$. The soft Bellman operator under $\mathcal{M}_H$ is

$$(\mathcal{T}_H^\pi Q)(s,a) = R_H(s,a) + \gamma\,\mathbb{E}_{s'\sim P_H(\cdot|s,a)}\big[V_Q^\pi(s')\big].$$

Writing $\lambda := \lambda(s,a)$ for brevity, R_H and P_H (Definition 1) leads to the pointwise identity

$$\begin{aligned}
(\mathcal{T}_H^\pi Q)(s,a) &= (1-\lambda)\,R_E(s,a) + \lambda\,R_N(s,a) \\
&\quad + \gamma\Big((1-\lambda)\,\mathbb{E}_{s'\sim P_E(\cdot|s,a)}\big[V_Q^\pi(s')\big] + \lambda\,\mathbb{E}_{s'\sim P_N(\cdot|s,a)}\big[V_Q^\pi(s')\big]\Big) \\
&= (1-\lambda)\,(\mathcal{T}_E^\pi Q)(s,a) + \lambda\,(\mathcal{T}_N^\pi Q)(s,a).
\end{aligned}$$

Thus, soft evaluation under $\mathcal{M}_H$ is the same convex combination of the component evaluation as in hard (max) case.

Policy improvement. Treating λ as fixed, the soft policy improvement step follows the SAC formulation:

$$\pi_{k+1}(\cdot \mid s) = \arg\min_{\pi} \; D_{\mathrm{KL}}\left(\pi(\cdot \mid s) \,\middle\|\, \frac{\exp\big(Q^{\pi_k}(s,\cdot)/\alpha\big)}{Z_k(s)}\right), \tag{12}$$

where $Z_k(s)$ is the normalizing constant. Alternating evaluation under $(\mathcal{T}_H^{\pi})$ and the update Eq. (12) is exactly soft policy iteration on $\mathcal{M}_H$. Under the standard assumptions of Haarnoja et al. (2018), this admits a unique soft fixed point and corresponding policy.

A.4 Policy parameterization and tractability for PE-SAC

PoE from SAC policy improvement. For a fixed gating function $\lambda : \mathcal{S} \times \mathcal{A} \to [0,1]$ and $Q_\theta = (1-\lambda)Q_E + \lambda Q_N$, the SAC information projection (for each s)

$$\pi^*(\cdot \mid s) = \arg\min_{\pi} D_{\mathrm{KL}}\left(\pi(\cdot \mid s) \,\middle\|\, \frac{\exp\big(Q_\theta(s,\cdot)/\alpha\big)}{Z_\theta(s)}\right)$$

has a unique solution

$$\begin{aligned}\pi^*(a \mid s) \;&\propto\; \exp\Big(\tfrac{(1-\lambda)Q_E(s,a)+\lambda Q_N(s,a)}{\alpha}\Big) \\ &= \big[\exp\big(Q_E(s,a)/\alpha\big)\big]^{\,1-\lambda(s,a)} \big[\exp\big(Q_N(s,a)/\alpha\big)\big]^{\,\lambda(s,a)}.\end{aligned}$$

If λ is state-only, $\lambda = \lambda(s)$, then the normalizers of $\exp(Q_E/\alpha)$ and $\exp(Q_N/\alpha)$ are constant in a and factor out, leading to the geometric mixture of normalized policies:

$$\pi^*(\cdot \mid s) \;\propto\; \pi_E(\cdot \mid s)^{\,1-\lambda(s)}\, \pi_N(\cdot \mid s)^{\,\lambda(s)}$$

where

$$\pi_E(\cdot \mid s) \propto \exp\big(Q_E(s,\cdot)/\alpha\big), \;\; \pi_N(\cdot \mid s) \propto \exp\big(Q_N(s,\cdot)/\alpha\big).$$

Why an action-dependent gating function breaks reparameterization. Write the energies $f_E := Q_E/\alpha$ and $f_N := Q_N/\alpha$. Define the unnormalized density

$$u_\phi(a \mid s) \;:=\; \exp\big\{(1-\lambda(s,a))\, f_E(s,a) + \lambda(s,a)\, f_N(s,a)\big\}, \qquad Z_\phi(s) := \int_{\mathcal{A}} u_\phi(a \mid s)\, da.$$

When $\lambda = \lambda(s,a)$, the normalizer $Z_\phi(s)$ has no closed form and its gradient with respect to the parameters inside λ, f_E, f_N is intractable. Therefore,

$$\log \pi_\phi(a \mid s) \;=\; (1-\lambda) f_E(s,a) + \lambda f_N(s,a) \;-\; \log Z_\phi(s)$$

cannot be evaluated with a tractable pathwise sampler $a = g_\phi(s,\epsilon)$, so the reparameterized SAC actor objective

$$J(\phi) \;=\; \mathbb{E}_{s,\epsilon}\big[\alpha \log \pi_\phi(a \mid s) - Q_\theta(s,a)\big]$$

is not tractable. This motivates a *state-only* gating function in the actor.

Gaussian policy with squashing (state-only gating). Following SAC, we use an unbounded Gaussian for a pre-squash variable $u \in \mathbb{R}^D$ and apply an elementwise $\tanh$ to obtain bounded actions $a = \tanh(u)$. Let the two pre-squash Gaussian *densities* be

$$p_E(u \mid s) = \mathcal{N}\big(u;\, \mu_E(s), \Sigma_E(s)\big), \qquad p_N(u \mid s) = \mathcal{N}\big(u;\, \mu_N(s), \Sigma_N(s)\big),$$

and let the gating function be state-only, $\lambda = \lambda(s) \in [0,1]$. Define the unnormalized product

$$\tilde{p}_H(u \mid s) \;:=\; p_E(u \mid s)^{\,1-\lambda(s)}\, p_N(u \mid s)^{\,\lambda(s)}.$$

Since the exponents are constants for fixed s, $\tilde{p}_H$ is proportional to a Gaussian. In particular,

$$\begin{aligned} p_H(u \mid s) &= \mathcal{N}\big(u;\, \mu_H(s), \Sigma_H(s)\big), \\ \Sigma_H^{-1}(s) &= (1-\lambda(s))\, \Sigma_E^{-1}(s) + \lambda(s)\, \Sigma_N^{-1}(s), \\ \mu_H(s) &= \Sigma_H(s)\Big((1-\lambda(s))\, \Sigma_E^{-1}(s)\mu_E(s) + \lambda(s)\, \Sigma_N^{-1}(s)\mu_N(s)\Big). \end{aligned} \tag{13}$$

With $a = \tanh(u)$ and the change-of-variables formula (cf. SAC(Haarnoja et al., 2018), Eqs. (20)–(21)),

$$\pi_H(a \mid s) \;=\; p_H(u \mid s)\,\Big|\det\Big(\frac{\partial a}{\partial u}\Big)\Big|^{-1}$$
$$\log \pi_H(a \mid s) \;=\; \log p_H(u \mid s) \;-\; \sum_{i=1}^{D} \log\big(1 - \tanh^2(u_i)\big),$$

where $u = \operatorname{arctanh}(a)$ and the Jacobian $\partial a/\partial u$ is diagonal with entries $1 - \tanh^2(u_i)$. When the gating is binary, $\lambda(s) \in \{0, 1\}$, Eq. (13) reduces to the corresponding expert.

B Implementation Details

B.1 Details for learning $\lambda(s, a)$

We train two one-step dynamics predictors on replay, $\hat{P}_E : (s, a) \mapsto \Delta\hat{s}_E(s, a)$ and $\hat{P}_N : (s, a) \mapsto \Delta\hat{s}_N(s, a)$, intended to approximate the transition dynamics of $\mathcal{M}_E$ and $\mathcal{M}_N$, respectively. Each predictor minimizes mean squared error on the state increment $\Delta s := s' - s$:

$$\mathcal{L}^{(i)}_{\text{dyn}} \;=\; \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\Big[\big\|\Delta\hat{s}_i(s, a) \;-\; \Delta s\big\|_2^2\Big], \qquad i \in \{E, N\}.$$

The dynamics *disagreement* is the squared difference between predicted increments

$$d(s, a) \;=\; \big\|\Delta\hat{s}_E(s, a) \;-\; \Delta\hat{s}_N(s, a)\big\|_2^2.$$

Online thresholding and labels (for supervision). We maintain running statistics (μ_t, σ_t) of $d(s, a)$ via the Welford algorithm (Chan et al., 1983), form a raw threshold $\hat{\tau}_t = \mu_t + k\,\sigma_t$ with $k > 0$ (symmetry breaking assumed sporadic), and exponentially smooth it

$$\tau_t \;\leftarrow\; \beta\,\tau_{t-1} + (1 - \beta)\,\hat{\tau}_t.$$

Binary supervision is then $y(s, a) = \mathbb{1}\{d(s, a) > \tau_t\}$.[1]

Gating function training and stochastic gating. We train the gate network $\lambda_\omega : \mathcal{S} \times \mathcal{A} \to [0, 1]$ with the binary cross-entropy loss (Eq. equation 4) on minibatches from $\mathcal{D}$. During each critic and actor update, we *recompute and cache* the gate probability on the sampled minibatch and use a *stochastic hard gate* obtained by Bernoulli sampling:

$$p(s, a) \;:=\; \lambda_\omega(s, a), \qquad \tilde{\lambda}(s, a) \;\sim\; \text{Bernoulli}\big(p(s, a)\big).$$

We then form $Q_\theta = (1 - \tilde{\lambda})Q_E + \tilde{\lambda}Q_N$ for the critic update and use $\tilde{\lambda}$ for the actor-side alignment (Sec. 6.2). Gradients from Q/π do *not* flow into ω (stop-gradient through both p and $\tilde{\lambda}$).

Target gate to reduce variance. To mitigate non-stationarity and variance from stochastic gating, we maintain an EMA of *probabilities*

$$\bar{p} \;\leftarrow\; \tau_\lambda\,p + (1 - \tau_\lambda)\,\bar{p},$$

and draw the RL gate from $\bar{p}$ instead of p:

$$\tilde{\lambda}(s, a) \;\sim\; \text{Bernoulli}\big(\bar{p}(s, a)\big).$$

Warm-start. To avoid noisy labels before the dynamics predictors stabilize, we use a warm-up period W steps during which the gate loss is disabled (i.e., $y \equiv 0$ and ω is not updated). A small prior routing is used by clamping $\tilde{\lambda}{=}1$ with probability p_{warm} during warm-up.

[1] As a robust alternative one may use running quantiles or median/MAD; we keep $k\sigma$ in main experiments.

B.2 Details for learning $\lambda_\zeta(s)$

We train a state-only actor gate $\lambda_\zeta : \mathcal{S} \to [0,1]$ to conservatively aggregate the action-dependent critic gate via

$$\lambda_\zeta(s) \;\approx\; \max_a \lambda_\omega(s,a).$$

To do so we adopt *expectile regression* with a high expectile level $\tau \to 1$, which approximates the max while remaining stable on in-distribution actions. Concretely, for each state s we draw M candidate actions $\{a_i\}_{i=1}^M$ (from current policies; see sampling details below) and minimize

$$\mathcal{L}_\lambda(\zeta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\frac{1}{M}\sum_{i=1}^{M} L_\tau\Big(\lambda_\omega(s,a_i) - \lambda_\zeta(s;\zeta)\Big)\right], \qquad L_\tau(u) \;=\; |\tau - \mathbb{1}\{u<0\}|\, u^2.$$

This objective encourages $\lambda_\zeta(s)$ to match the upper tail of $\{\lambda_\omega(s,a_i)\}_{i=1}^M$, leading to a conservative state-level gate.

Bernoulli actor gating (training & inference). At both training and inference, we use a binary actor gating sampled from the probability $\lambda_\zeta(s)$:

$$\tilde{\lambda}_\zeta(s) \;\sim\; \text{Bernoulli}\big(\lambda_\zeta(s)\big).$$

For RL updates we cache $\tilde{\lambda}_\zeta$ per minibatch and apply stop-gradient through the sample.

Action sampling for expectiles. We form the candidate set $\{a_i\}_{i=1}^M$ per state by drawing from a mixture of current policies (e.g., π_E, π_N). This increases the chance of including symmetry-breaking actions. We use the same M across tasks (see the hyperparameters Table 8).

Warm-start. To avoid noisy supervision before λ_ω stabilizes, we apply a short warm-up period W steps where $\mathcal{L}_\lambda(\zeta)$ is disabled; We use a small prior bias by clamping $\tilde{\lambda}_\zeta{=}1$ with probability p_{warm} during warm-up.

Gradient isolation. Gradients from the RL losses do *not* flow into λ_ζ; the gate is updated only via the expectile objective above.

B.3 Networks

For the Grid-World experiments, we implemented π_E and $\hat{P}_E$ using equivariant linear layers from `escnn`(Weiler & Cesa, 2019). In all other settings, we used EMLP layers (Finzi et al., 2021b) for π_E and $\hat{P}_E$. The remaining networks, including π_N, $\hat{P}_N$, λ_ω, λ_ζ, and the critics, were implemented as standard MLPs.

B.4 Implementation framework

Our implementation builds on the Residual Pathway Priors (RPP) codebase (Finzi et al., 2021a), which provides flexible infrastructure for combining equivariant and non-equivariant components. We extend this framework with our gated Q-networks, gated policies, and disagreement-based λ supervision, while keeping the training loops and optimization settings consistent with RPP.

C Experimental Details

In this section, we introduce the group symmetries, environment details, and the hyperparameters used in each environment. The implemented group symmetries used in each environment are summarized in Table 1, and the corresponding group representations for each state and action space are summarized in Table 2, 3, 4, 5, 6. We summarize the common hyperparameters for DQN used in Grid-World, including those for PE-DQN in Table 7. For SAC, we use the default hyperparameters (Haarnoja et al., 2018), which are listed in Table 8, including those for PE-SAC. All the experiments were run on NVIDIA RTX 4090 GPUs.

Table 1: Symmetries of environments used in the experiments.

Env	Implemented Symmetries
Grid-World	C_4
Hopper	$\mathbb{Z}_2$
Ant	$\mathbb{Z}_4$
Swimmer	$\mathbb{Z}_2$
Fetch	$SO(3)$
UR5e	$SO(3)$

Grid-World. The symmetry used in Grid-World is the Cyclic group C_4, as summarized in Table 1. It consists of a 15×15 grid, with observations given by the concatenated agent and goal positions $[x_{\text{agent}}, y_{\text{agent}}, x_{\text{goal}}, y_{\text{goal}}]$. The action space is $\{\uparrow, \leftarrow, \downarrow, \rightarrow\}$. Group representations are implemented as two concatenated 2D rotation matrices on the state space and a 4×4 permutation matrix on the action space. Rewards are defined as $+1$ for reaching the goal and -0.01 per step otherwise.

Locomotion. The symmetries used in each environment are summarized in Table 1. The state and action space representations, as well as the hyperparameters, are adopted from RPP (Finzi et al., 2021a). For Swimmer-v2, Finzi et al. (2021a) reports using the approximate symmetry $\mathbb{Z}_2 \times \mathbb{Z}_2$ (left-right, front-back symmetries), but the official code does not provide a correct implementation of this. Therefore, we instead use the exact symmetry $\mathbb{Z}_2$ (left-right symmetries) in our Swimmer experiments.

Manipulation. The symmetries used in each environment are summarized in Table 1. In Fetch Reach, the agent is trained to move the end-effector to a randomly sampled target position in each episode. The corresponding state and action spaces, together with their representations of the exploited symmetries, are provided in Table 5. A dense reward is given at every timestep as the negative Euclidean distance between the current end-effector and the goal position. In UR5e Reach, the agent is trained to reach a randomly sampled $SE(3)$ target pose in each episode. The corresponding state and action spaces with the representations of the exploited symmetries are provided in Table 6. A dense reward is given at every timestep as the negative weighted sum of the Euclidean distance (translational error) and the geodesic distance (rotational error) between the current end-effector and the goal poses. A weight of 0.19098621461 is applied to the geodesic distance term so that a 15° rotational error is treated as equivalent to a 0.05m translational error. We scale action of translation by 0.05 m for both tasks, and rotation by 0.2618 rad (15°) for UR5e Reach task.

Overall. The state and action representations used for the equivariant networks in each environment except Grid-World are shown in Table 2, 3, 4, 5, 6 (last column). In these tables, V denotes an n-dimensional base representation, transformed by permutations for $\mathbb{Z}_n$ and by rotation matrices for $SO(3)$. $\mathbb{R}$ denotes a 1-dimensional scalar representation which is invariant under these group actions. P denotes a 1-dimensional pseudoscalar representation, which is transformed by the sign of the permutation. (e.g., for Swimmer-v2, P flips sign under left-right reflection of the body.) Note that powered representations such as V^n indicate the direct sum of n instances of the representation; this is given here as an example:

$$V^n = \bigoplus_{i=1}^{n} V.$$

Hyperparameters used for locomotion and manipulator (SAC) experiments are shown in Table 8. Those are shared across all tasks, unless specified in the table.

Table 2: Hopper-v2 state and action spaces with their representations

	Name	Description	Dim	Rep
State	Torso z	z-coordinate of the torso	1	$\mathbb{R}$
	Orientation	Torso pitch angle	1	P
	Thigh angle	Thigh joint angle	1	P
	Leg angle	Leg joint angle	1	P
	Foot angle	Foot joint angle	1	P
	Torso velx	Linear velocity of torso (x)	1	P
	Torso velz	Linear velocity of torso (z)	1	$\mathbb{R}$
	Torso angvel	Angular velocity of torso (y)	1	P
	Thigh angvel	Angular velocity of thigh hinge	1	P
	Leg angvel	Angular velocity of leg hinge	1	P
	Foot angvel	Angular velocity of foot hinge	1	P
Action	Thigh	Torque applied on thigh joint	1	P
	Leg	Torque applied on leg joint	1	P
	Foot	Torque applied on foot joint	1	P

Table 3: Ant-v2 state and action spaces with their representations

	Name	Description	Dim	Rep
State	Torso z	z-coordinate of the torso	1	$\mathbb{R}$
	Torso quat	Orientation of the torso (quaternion)	4	$\mathbb{R}^4$
	Hip 1 angle	Angle between torso and front-left link	1	V
	Hip 2 angle	Angle between torso and front-right link	1	
	Hip 3 angle	Angle between torso and back-left link	1	
	Hip 4 angle	Angle between torso and back-right link	1	
	Ankle 1 angle	Angle between two front-left links	1	V
	Ankle 2 angle	Angle between two front-right links	1	
	Ankle 3 angle	Angle between two back-left links	1	
	Ankle 4 angle	Angle between two back-right links	1	
	Torso vel	Linear velocity of torso (x, y, z)	3	$\mathbb{R}^3$
	Torso angvel	Angular velocity of torso (x, y, z)	3	$\mathbb{R}^3$
	Hip 1 angvel	Angular velocity of front-left hip joint	1	V
	Hip 2 angvel	Angular velocity of front-right hip joint	1	
	Hip 3 angvel	Angular velocity of back-left hip joint	1	
	Hip 4 angvel	Angular velocity of back-right hip joint	1	
	Ankle 1 angvel	Angular velocity of front-left ankle joint	1	V
	Ankle 2 angvel	Angular velocity of front-right ankle joint	1	
	Ankle 3 angvel	Angular velocity of back-left ankle joint	1	
	Ankle 4 angvel	Angular velocity of back-right ankle joint	1	
Action	Hip 1	Torque on front-left hip joint	1	V
	Hip 2	Torque on front-right hip joint	1	
	Hip 3	Torque on back-left hip joint	1	
	Hip 4	Torque on back-right hip joint	1	
	Ankle 1	Torque on front-left ankle joint	1	V
	Ankle 2	Torque on front-right ankle joint	1	
	Ankle 3	Torque on back-left ankle joint	1	
	Ankle 4	Torque on back-right ankle joint	1	

Table 4: Swimmer-v2 state and action spaces with their representations

	Name	Description	Dim	Rep
State	Orientation angle	Front tip angle	1	P
	Head joint angle	First rotor angle	1	P
	Tail joint angle	Second rotor angle	1	P
	x, y velocities	Tip velocities along x, y	2	$\mathbb{R}^2$
	Orientation angvel	Front tip angular velocity	1	P
	Head joint angvel	First rotor angular velocity	1	P
	Tail joint angvel	Second rotor angular velocity	1	P
Action	Head joint	Torque on first rotor	1	P
	Tail joint	Torque on second rotor	1	P

Table 5: Fetch Reach state and action spaces with their representations

	Name	Description	Dim	Rep
State	EE pos	End-effector position (x, y, z)	3	V
	EE vel	End-effector velocity (v_x, v_y, v_z)	3	V
	Goal pos	Goal position (x, y, z)	3	V
Action	EE rel trans	Relative translation $(\Delta x, \Delta y, \Delta z)$	3	V
	Gripper cmd	Gripper open/close control	1	$\mathbb{R}$

Table 6: UR5e Reach state and action spaces with their representations

	Name	Description	Dim	Rep
State	EE pos	End-effector position (x, y, z)	3	V
	EE rot6d	End-effector orientation (6D rep.)	6	V^2
	EE velp	End-effector linear velocity (v_x, v_y, v_z)	3	V
	EE velr	End-effector angular velocity $(\omega_x, \omega_y, \omega_z)$	3	V
	Goal pos	Goal position (x, y, z)	3	V
	Goal rot6d	Goal orientation (6D rep.)	6	V^2
Action	EE rel trans	Relative translation $(\Delta x, \Delta y, \Delta z)$	3	$\mathbb{R}^3$
	EE rel rot	Relative rotation (axis–angle) (a_x, a_y, a_z)	3	$\mathbb{R}^3$

Table 7: Hyperparameters used in Grid-World (DQN) experiments.

Hyperparameter	Value
Optimizer	Adam (Kingma et al., 2015)
Learning rate	1×10^{-4}
Hidden size	$[256, 256]$
Batch size	256
Discount factor γ	0.99
Target network update rate τ	0.005
Replay buffer size	1×10^{5}
ε-greedy schedule	$1.0 \rightarrow 0.05$ (15k steps)
λ, $\hat{P}_E$, $\hat{P}_N$ batch size	256
λ, $\hat{P}_E$, $\hat{P}_N$ learning rate	1×10^{-4}
#λ warm-start steps	30,000
λ prior bias	0.5
λ hidden size	$[256, 256]$
λ gradient clipping	1.0
$\hat{P}_E$, $\hat{P}_N$ hidden size	$[256, 256]$
$\hat{P}_E$, $\hat{P}_N$ gradient clipping	1.0
Disagreement coefficient k	1.5
# Threshold update interval steps	100
Threshold EMA β	0.05

Table 8: Hyperparameters used in locomotion and manipulation (SAC) experiments.

Hyperparameter	Value
Optimizer	Adam (Kingma et al., 2015)
Actor learning rate	3×10^{-4}
Critic learning rate	3×10^{-4}
Temperature learning rate	3×10^{-4}
Entropy coefficient	auto-adjust (Haarnoja et al., 2018)
Batch size	256
Discount factor γ	0.99
Target network update rate τ	0.005 (0.004 for RPP Swimmer-v2)
Target entropy	$-0.5 \times \dim(\text{action})$
Hidden size	$[256, 256]$
Gradient clipping	0.5
λ_ω, λ hidden size	$[128, 128]$
$\hat{P}_E$, $\hat{P}_N$ hidden size	$[256, 256]$
λ_ω, λ_ζ, $\hat{P}_E$, $\hat{P}_N$ batch size	256
# $\hat{P}_E$, $\hat{P}_N$ gradient steps	2
λ_ω, λ_ζ learning rate	1×10^{-4}
λ_ω, λ_ζ gradient clipping	0.5
λ_ω, λ_ζ prior bias	0.7685
# Threshold update interval steps	100
Threshold EMA β	0.1
Expectile regression coefficient τ_{exp}	0.8
# Expectile action samples M	4

D Equivariance Error and its Propagation Under Symmetry-Breaking

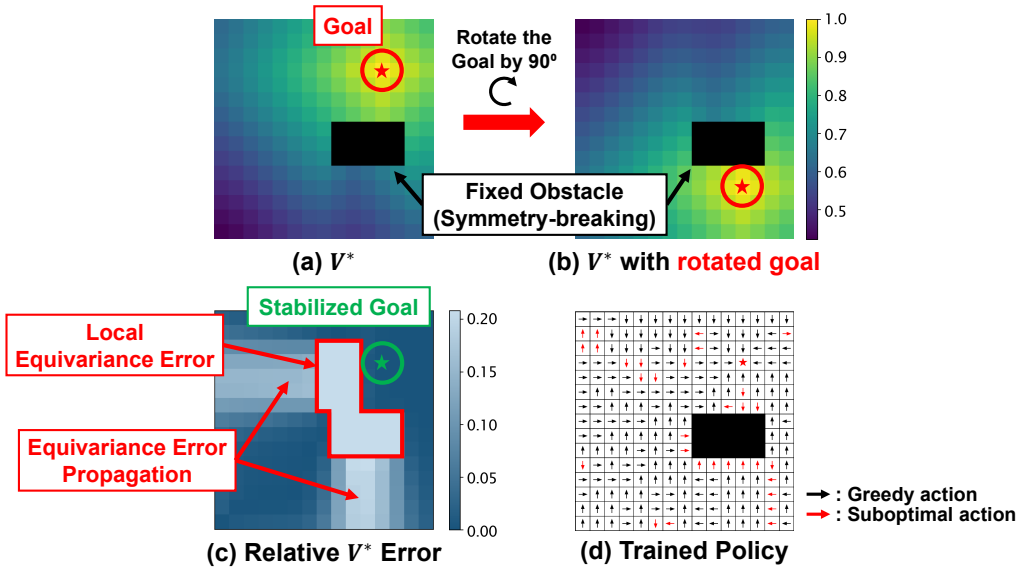


Figure 5: **Equivariance error under symmetry-breaking.** We assess rotational equivariance by comparing the base optimal value function V^* with the value obtained after rotating the goal by $90°$ (red star) while keeping obstacles fixed (black cells), thereby breaking the symmetry. **(a)** Baseline optimal value V^*. **(b)** V^* with the goal rotated by $90°$ while obstacles (black) remain fixed. **(c)** Per-state relative equivariance error $\left(\left|V^*(s) - V^*(gs)\right| / \left|V^*(s)\right|\right)$ with the goal stabilization. The sky-blue cells bordered by a red line coincide with the overlap between the original obstacle and its image under g, creating large local errors. The error then propagates outward, as reflected by the surrounding regions whose shading gradually darkens. This non-local propagation occurs for all $g \in G$ and has broader implications for equivariant RL training. **(d)** Greedy actions from an equivariant DQN. Red arrows denote suboptimal moves, illustrating that the learned policy inherits errors in symmetry-broken regions.

E The Use of Large Language Models

In this paper, we used LLMs solely for text polishing and generating code snippets. Study design, theoretical results, algorithmic contributions, and all experiments/analyses were conceived and implemented by the authors. All code generated with LLM assistance was reviewed and verified by the authors.