CAN EUCLIDEAN SYMMETRY HELP IN REINFORCEMENT LEARNING AND PLANNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

In robotic tasks, changes of reference frames do not affect the underlying physics of the problem. Isometric transformations, including translations, rotations, and reflections, collectively form the Euclidean group. In this work, we study reinforcement learning and planning tasks that have Euclidean group symmetry. We show that MDPs with continuous symmetries have linear approximations that satisfy steerable kernel constraints, which are widely studied in equivariant machine learning. Guided by our theory, we propose an equivariant model-based RL algorithm algorithm, which is based on sampling-based MPPI for continuous action spaces. We test our proposed equivariant TD-MPC algorithm on a set of standard RL benchmark tasks. Our work shows that equivariant methods can give a great boost in performance on control tasks with continuous symmetry.

1 INTRODUCTION

Robot decision-making tasks often involve the movement of robots in two or three-dimensional Euclidean space. Different reference frames can be used to model the robot and environment, but the *underlying physics* of the system must be *independent* of the choice of reference frame (Einstein, 1905). The set of all such reference frame transformations is called the Euclidean group $\mathrm{E}(d)$. In this work, we show that utilizing the Euclidean frame symmetry inherent in many robotic planning and control problems allows for the design of more efficient learning algorithms. The use of symmetry in decision-making has been studied in model-free and model-based reinforcement learning (RL), planning, optimal control, and other related fields (Ravindran & Barto, 2004; Zinkevich & Balch, 2001; van der Pol et al., 2020a; Mondal et al., 2020; Wang et al., 2021; Zhao et al., 2022b). Despite this, there is no unified theory of how symmetry can be utilized to develop better RL or planning algorithms for robotics applications.

In many problems in robotics, we are interested in the Markov Decision Process (MDP) that describes a robot moving in 2D or 3D space. Motivated by the study of *geometric graphs* and geometric deep learning (Bronstein et al., 2021), we define *Geometric MDPs* as the class of MDPs that correspond to the decision process of a robot moving in Euclidean space. The question that we aim to answer is: *Can Euclidean symmetry guarantee benefits in (model-based) RL algorithms?* To answer it, we aim to first formally describe what "benefits" mean and how symmetry enables them, then show a model-based RL algorithm that is developed with the guidance of the theory.

To begin, we present a theoretical framework that studies the linearized dynamics of geometric MDPs and shows that the matrices that appear in linearized dynamics are G-steerable kernels (Cohen & Welling, 2016d). Using recent results on parameterizations of steerable kernels (Lang & Weiler, 2020b), we show that the steerable kernel solution significantly reduces the number of parameters needed to specify the linearized dynamics. We can use it to predict parameter reduction for tasks with geometric structure. The reduction is infinite for continuous tasks with continuous symmetry, such as moving 2D particle.

Inspired by the theoretical results which show that equivariant versions of linearized model-based approaches contain a smaller number of parameters than general models, we propose an equivariant sampling-based model-based RL algorithm for Geometric MDPs. It is based on Model Predictive Path Integral (MPPI); we propose a strategy that enforces symmetry on the sampling process: if the input state is rotated, the output action should be rotated accordingly, as demonstrated in Figure 1.

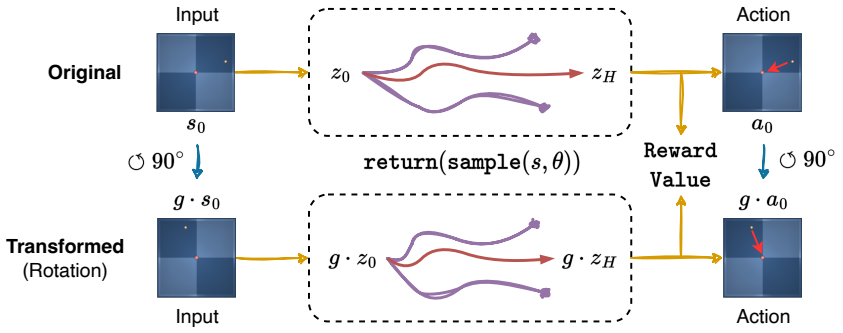

Figure 1: Illustration of equivariance in the proposed sampling-based planning algorithm $a_0 = \texttt{plan}(s_0)$: if the input state is rotated, the output action should be rotated accordingly. This requires the learned functions are G-equivariant or G-invariant networks and a special sampling strategy, introduced in our method.

Our method extends the prior work from (1) planning on 2D grids with value-based planning (Zhao et al., 2022b) and (2) model-free equivariant RL (van der Pol et al., 2020b; Wang et al., 2021) to continuous state and action spaces. We take inspiration from geometric deep learning (Bronstein et al., 2021) and consider the features in neural networks to transform under Euclidean symmetry. Our algorithm is constructed to be equivariant with respect to changes of the reference frame, which is usually known beforehand. We evaluate the proposed algorithm on DeepMind Control suite and MetaWorld continuous-control tasks and show its sample efficiency against non-equivariant methods, which demonstrates the benefits of equivariance in model-based RL with sampling-based planning (MPPI) and the value of our theory.

Our contributions can be summarized as follows: *(i)* We define a class of MDPs that correspond to the movement of a physical agent in two or three dimensional Euclidean space ("Geometric MDPs"). *(ii)* By analyzing the linearization of Geometric MDPs, our theory shows a reduction in the number of free parameters in the ground-truth linearized dynamics and optimal control policy. *(iii)* Motivated by our theory, we propose a sampling-based model-based RL algorithm that leverages Euclidean symmetry for Geometric MDPs. *(iv)* Our empirical results demonstrate the effectiveness of our method in solving MDPs on control tasks with continuous symmetries.

2 Problem Statement: Symmetry and Choice of Reference Frame

To theoretically study how symmetry benefits in solving MDPs, we describe the source of symmetry and define a class of MDPs that has symmetry constraints and can be linearized.

2.1 Geometric Structure in MDP

The set of all isometric changes of reference frame form the Euclidean symmetry group $\mathrm{E}(d)$ (Bronstein et al., 2021; Weiler & Cesa, 2021; Lang & Weiler, 2020b). Any subgroup of $\mathrm{E}(d)$ can be expressed in semi-direct product form as $\left(\mathbb{R}^d, +\right) \rtimes G$, where G is the stabilizer group of origin and the action on a vector x includes a translation part t and rotation/reflection part g, i.e., $x \mapsto (tg) \cdot x := gx + t$ (Lang & Weiler, 2020b).

To transform an MDP (to a different reference frame), we require the MDP to have the group G-action on both the state and action space (Zhao et al., 2022b; Wang et al., 2021; van der Pol et al., 2020b). This definition unifies different types of prior work and allows the state and actions spaces to be a any spaces equipped with a G-action (van der Pol et al., 2020b; Wang et al., 2021; Zhao et al., 2022b; Teng et al., 2023). The compact group $G \leq \mathrm{GL}(d)$ can be any group, including the group of proper 3D transformations $\mathrm{SO}(3)$ or finite subgroups like the icosahedral group or cyclic groups (Brandstetter et al., 2021). We define a class of MDPs with geometric structure, extending a previously studied discrete case (Zhao et al., 2022b).

Definition 1 (Geometric MDP) *A Geometric MDP (GMDP) $\mathcal{M}$ is an MDP with a (compact) symmetry group $G \leq \mathrm{GL}(d)$ that acts on the state and action space. It is written as a tuple*

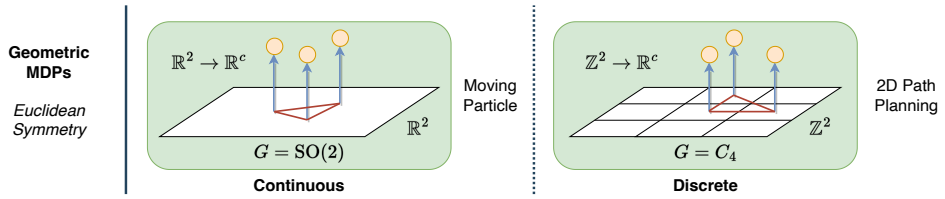

Figure 2: Illustration of MDPs with underlying geometric structure, e.g., a 2D particle moving or a path planing problem, which have 2D rotation groups G that have G-action on the MDP state and action space.

$\langle \mathcal{S}, \mathcal{A}, P, R, \gamma, G, \rho_{\mathcal{S}}, \rho_{\mathcal{A}} \rangle$. *The state and action spaces $\mathcal{S}, \mathcal{A}$ have (continuous) group actions that transform them, defined by $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$.*

The *symmetry properties* in MDPs are specified by equivariance and invariance of the transition and reward functions respectively (Zinkevich & Balch, 2001; Ravindran & Barto, 2004; van der Pol et al., 2020a; Wang et al., 2021; Zhao et al., 2022b; 2023a):

$$\forall g \in G, \forall s, a, s', \qquad P(s' \mid s, a) = P(g \cdot s' \mid g \cdot s, g \cdot a) \tag{1}$$

$$\forall g \in G, \forall s, a, \qquad R(s, a) = R(g \cdot s, g \cdot a) \tag{2}$$

where g acts on the state and action spaces by group representations $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$ respectively. For example, the *standard representation* $\rho_{\text{std}}(g)$ of $\mathrm{SO}(2)$ assigns each rotation $g \in \mathrm{SO}(2)$ a 2D rotation matrix $R_{2\times 2}$, while the *trivial representation* $\rho_{\text{tri}}(g)$ assigns identity $\mathbf{1}_{1\times 1}$ to all g.

Continuous G-action. Compared to prior work, we additionally require *continuous* group action $\cdot_G : G \times X \to X$ and find it gives promising theoretical results, which is optional for implementation. If the G-actions on $\mathcal{S}$ and $\mathcal{A}$ are continuous[1], there is an interesting geometric interpretation based on fiber bundle theory (Husemöller, 2013). The linearized dynamics of a system are much more constrained[2] (i.e., have fewer free parameters) when the system has continuous G-action. See Appendix D for more detail.

Examples. We list some examples in the Table 1 to demonstrate that the definition covers previous work (a homogeneous space, a group, or any other space as long as equipped with a G-action (van der Pol et al., 2020b; Wang et al., 2021; Zhao et al., 2022b; Teng et al., 2023)) and what symmetry could bring. We list the symmetry group G, the state and action spaces $\mathcal{S}$, $\mathcal{A}$, and provide how symmetry G reduces the space via quotient $\mathcal{S}/G$ and how "large" a group G can relate different states in that MDP via orbit Gx. We provide detailed explanation in Appendix B.4.

2.2 Related Topics

We discuss geometric graphs and the use of symmetry in reinforcement learning. For further discussion, please see Appendix B.

Geometric graphs. Our definition of GMDP is closely related to the concept of *geometric graphs* (Bronstein et al., 2021; Brandstetter et al., 2021), which model MDPs as state-action connectivity graphs. Previous works studied algorithmic alignments and dynamic programming on geometric graphs (Xu et al., 2019; Dudzik & Veličković, 2022). We propose extending this concept to include additional geometric structures by embedding the MDP into a geometric space such as $\mathbb{R}^2$ or $\mathbb{R}^3$. In discussion of GMDP, we focus on 2D and 3D Euclidean symmetry (Lang & Weiler, 2020b; Brandstetter et al., 2021; Weiler & Cesa, 2021), with the corresponding symmetry groups of $\mathrm{E}(2)$ and $\mathrm{E}(3)$, respectively. The relation between equivariant message passing and dynamic programming / value iteration on geometric MDPs is discussed in Section 3.

Symmetry in MDPs. Symmetry in decision-making tasks has been explored in previous works on MDPs and control, with research on symmetry in MDPs with no function approximation (Ravindran & Barto, 2004; Ravindran & Barto; Zinkevich & Balch, 2001) and symmetry in model-free (deep)

[1]If the group G is additionally a *compact* Lie group, there exists a map $p : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{B}$ that projects the state-action space $\mathcal{S} \times \mathcal{A}$ to a lower dimensional base space $\mathcal{B}$ (Cohen et al., 2020b). The existence and smoothness of the projection p can be established using principal bundle theory.

[2]Continuous symmetries correspond to conservation laws, while discrete (non-differentiable) symmetries do not have corresponding conservation laws (Zee, 2016).

Table 1: Examples of geometric MDPs. G denotes the MDP symmetry group. $\mathcal{S}$ denotes the MDP state space. $\mathcal{A}$ denotes the MDP action space. We can quantitatively measure the savings obtained by exploiting equivariance. "Images" refers to panoramic egocentric images $\mathbb{Z}^2 \to \mathbb{R}^{H\times W\times 3}$. $\circ$ denotes group element composition. We list the quotient space $\mathcal{S}/G$ to give intuition on savings. The $Gx = \{g \cdot x \mid g \in G\}$ column shows the G-orbit space of $\mathcal{S}$ ($\cong$ denotes isomorphic equivalence).

ID	G	$\mathcal{S}$	$\mathcal{A}$	$\mathcal{S}/G$	Gx	Task
1	C_4	$\mathbb{Z}^2$	C_4	$\mathbb{Z}^2/C_4$	C_4	2D Path Planning (Tamar et al., 2016)
2	C_4	Images	C_4	$\mathbb{Z}^2/C_4$	C_4	2D Visual Navigation (Zhao et al., 2022b)
3	$\mathrm{SO}(2)$	$\mathbb{R}^2$	$\mathbb{R}^2$	$\mathbb{R}^+$	S^1	2D Continuous Navigation
4	$\mathrm{SO}(3)$	$\mathbb{R}^3 \times \mathbb{R}^3$	$\mathbb{R}^3$	$\mathbb{R}^+ \times \mathbb{R}^3$	S^2	3D Free particle (with velocity)
5	$\mathrm{SO}(3)$	$\mathbb{R}^3 \rtimes \mathrm{SO}(3)$	$\mathbb{R}^3 \times \mathbb{R}^3$	$\mathbb{R}^+ \times \mathbb{R}^3$	S^2	Moving 3D Rigid Body
6	$\mathrm{SO}(2)$	$\mathrm{SO}(2)$	$\mathbb{R}^2$	$\{e\}$	S^1	Free Particle on $\mathrm{SO}(2) \cong S^1$ manifold
7	$\mathrm{SO}(3)$	$\mathrm{SO}(3)$	$\mathbb{R}^3$	$\{e\}$	S^2	Free Particle on $\mathrm{SO}(3)$ (Teng et al., 2023)
8	$\mathrm{SO}(2)$	$\mathrm{SE}(2)$	$\mathrm{SE}(2)$	$\mathbb{R}^2$	S^1	Top-down grasping (Zhu et al., 2022)
9	$\mathrm{SO}(2)$	$(S^1)^2 \times (\mathbb{R}^2)^2$	$\mathbb{R}^2$	$S^1 \times (\mathbb{R}^2)^2$	S^1	Two-arm manipulation (Tassa et al., 2018)

RL using equivariant policy networks (van der Pol et al., 2020a; Mondal et al., 2020; Wang et al., 2021). Additionally, the use of symmetry in value-based planning on a 2D grid is analyzed by Zhao et al. (2022b). We extend this line of work by focusing on MDPs with continuous state and action spaces and sampling-based planning/control algorithms.

3 Theory: Why is Symmetry Useful in Geometric MDPs?

The goal of this section is to provide theoretical guidance on assessing the potential benefits of symmetry in a Geometric MDP for a Reinforcement Learning (RL) algorithm, particularly when planning using learned dynamic models.

3.1 Properties of Geometric MDPs

In RL, the optimal policy mapping is G-equivariant (Ravindran & Barto, 2004). To incorporate symmetry constraints, a strategy is to constrain the entire policy mappingto be equivariant: $a_t = \texttt{policy}(s_t)$ (van der Pol et al., 2020b; Wang et al., 2021; Zhao et al., 2022b; 2023a), as shown in Figure 1. Many model-based RL algorithms rely on iteratively applying Bellman operations (Sutton & Barto, 2018). Thus, we first show that symmetry G in a Geometric MDP (GMDP) results in G-equivariant Bellman operator, which indicates that we can constrain the iterative process in model-based RL algorithms to be G-equivariant to exploit symmetry. Additionally, for GMDPs[3], a specific instance of DP-based algorithm, value iteration, can be connected with geometric graph neural network (Bronstein et al., 2021). These properties do not require linearization and do not require continuous group actions.

Theorem 1 *The Bellman operator of a GMDP is equivariant under Euclidean group* $\mathrm{E}(d)$.

Theorem 2 *For a GMDP, value iteration resembles* $\mathrm{E}(d)$*-equivariant geometric message passing.*

We provide proofs and derivation in Appendix D. This is an extension to the theorems in (Zhao et al., 2022b) on 2D discrete groups, where they showed that value iteration is equivariant under discrete subgroups of the Euclidean group: discrete translations, rotations, and reflections. We generalize this result to groups of the form of $\left(\mathbb{R}^d, +\right) \rtimes G$, where G is continuous[4].

3.2 Linearizing Geometric MDPs: G-steerable Kernel Constraints

[3]For non-geometric graphs, Dudzik & Veličković (2022) show the equivalence between dynamic programming on a (general non-geometric) MDP and a message-passing GNN.

[4]For the translation part, one may use relative/normalized positions or induced representations (Cohen et al., 2020b; Lang & Weiler, 2020b).

The dynamics function in GMDPs is generally nonlinear. In this subsection[5], we derive the iterative linearization of dynamics of GMDPs to get G-equivariant linear maps. We focus on the linearization for two reasons: *(1)* if infinitesimal group actions on state-action space exists, the symmetry of the nonlinear GMDP is *equivalent* to G-steerable constraints of the linear dynamics, *(2)* the linearized dynamics is connected to LQR and is easier to analyze, such as the dimensions of the (linear) dynamics function, policy function, and more.

Iterative Linearization. We assume the dynamics is deterministic $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and *iteratively linearize* f at each step. It is naturally connected to *time-varying* iterative Linear Quadratic Regulator (iLQR). We highlight the linearization procedure of $f(\boldsymbol{s}_t, \boldsymbol{a}_t)$, where matrices A and B depend *arbitrarily* on time step t. Later, we assume that it only depends on state and action $(\boldsymbol{s}_t, \boldsymbol{a}_t)$.

$$\text{Original:} \quad \boldsymbol{s}_{t+1} = f(\boldsymbol{s}_t, \boldsymbol{a}_t) \quad \to \quad \text{Linearized at step } t\text{:} \quad \boldsymbol{s}_{t+1} = A_t \cdot \boldsymbol{s}_t + B_t \cdot \boldsymbol{a}_t \tag{3}$$

Theorem 3 *If a Geometric MDP has an infinitesimal G-action on the state-action space $\mathcal{S} \times \mathcal{A}$, the linearized dynamics is also G-equivariant: the matrix-valued functions $A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}$ and $B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{A}}$ satisfy G-steerable kernel constraints.*

Under infinitesimal symmetry transformation $g \approx 1_G \in G$, the state and action spaces transform as $\boldsymbol{s} \mapsto \rho_\mathcal{S}(g) \cdot \boldsymbol{s}, \boldsymbol{a} \mapsto \rho_\mathcal{A}(g) \cdot \boldsymbol{a}$ where $\rho_\mathcal{S}$ and $\rho_\mathcal{A}$ are representations of the group G. Additionally, the dynamics must satisfy,

$$\rho_\mathcal{S}(g) \cdot f(\boldsymbol{s}, \boldsymbol{a}) = f(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}) \tag{4}$$

Let us consider the linearized problem at point $p = (\boldsymbol{s}_0, \boldsymbol{a}_0)$. Assuming that the state and control do not change too drastically over a short period of time and that the time-varying A and B only depend on the linearization point p but not other factors, we can approximate the true dynamics as

$$\boldsymbol{s}_{t+1} = A(p) \cdot \boldsymbol{s}_t + B(p) \cdot \boldsymbol{a}_t, \quad A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}, \quad B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{A}}. \tag{5}$$

Now, linearizing f and using the symmetry constraint, the matrix-valued functions $A(p)$ and $B(p)$ must satisfy the constraints

$$\forall g \in G, \quad A(g \cdot p) = \rho_\mathcal{S}(g) A(p) \rho_\mathcal{S}(g^{-1}), \quad B(g \cdot p) = \rho_\mathcal{S}(g) B(p) \rho_\mathcal{A}(g^{-1}) \tag{6}$$

so that A is a G-steerable kernel with input and output representation $\rho_\mathcal{S}$, and B has input type $\rho_\mathcal{S}$ and output type $\rho_\mathcal{A}$. The kernel constraints relate $A(p)$ and $B(p)$ at different points. We use the Figure 3 to demonstrat the idea. On each orbit (left), the constraints can be solved exactly: the matrices on same orbits (same colors) are related and have explicit parameterization given in Lang & Weiler (2020a). Thus, these matrices can be spanned on a basis (denoted by K) and live in a smaller "base" space $\mathcal{B} = X/G$ with a certain form $A_\downarrow : \mathcal{B} \to \mathbb{R}^{2\times 2}$.

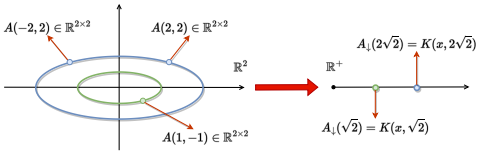

Figure 3: A schematic showing how a matrix-valued kernel $A : X \to \mathbb{R}^{2\times2}$ is constrained by the SO(2)-steerable kernel constraints on a set of orbits $A(g \cdot p) = \rho_{\text{out}}(g) A(p) \rho_{\text{in}}(g^{-1})$. This example is further explained in Appendix D.

Benefits for control. Symmetry further enables better control. We can further show that the policy and value function can be parameterized with fewer parameters based on the discrete algebraic Riccati equation (DARE) for time-varying LQR problem.

Theorem 4 *The LQR feedback matrix in $a_t^\star = -K(p)s_t$ and value matrix in $V = s_t^\top P(p) s_t$ are G-steerable kernels, or matrix-valued functions: $K : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{A} \times d_\mathcal{S}}$ and $P : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}$.*

What tasks are suitable for Euclidean equivariance? We find tasks that have dominated *global* Euclidean symmetry (change of reference frame) and less local symmetry can have relatively better parameter reduction, shown in Table 1. Tasks with more kinematic constraints make them harder to exploit Euclidean equivariance. For examples, for kinematic chains, it inherently has local coordinates, which makes it harder to use Euclidean symmetry w.r.t. the global reference frame. To better exploit symmetry there, it needs to explicitly consider constraints, such as the case in position-based (Tsai, 2017) or particle-based dynamics (Han et al., 2022).

[5]For simplicity, the theory omits an encoder from the state to latent space $\texttt{enc} : \mathcal{S} \to \mathcal{Z}$ in TD-MPC and MuZero style algorithms. In implementation, we follow TD-MPC that uses a learned encoder, which, when paired with an equivariant downstream network, helps learn symmetric representations. (Park et al., 2022).

Interpretation and Examples. The theory means that the linearized dynamics must satisfy a more restrictive set of conditions to be G-equivariant compared to the full, non-linear dynamics. It gives us a theoretical estimation of free parameters for *each* task, as well as improvement of sample efficiency when using (Jedra & Proutiere, 2021). We discuss further examples and computation of reduced dimensions in Appendix D. A toy example that illustrates how symmetry reduces the number of free parameters is moving a 3D particle (3D `PointMass`, Example 4 in Table 1): the matrix $A(p)$ (for $p \in \mathcal{S} \times \mathcal{A}$) will have dimension 6×6 but can be decomposed to 2×2 blocks of 3×3 sub-matrices, each only with 3 free parameters. Thus, for each orbit p, $A(p)$ and $B(p)$ only have $4 \times 3 = 12$ free parameters. Additionally, on a given orbit, $A(p)$ and $B(p)$ have explicit forms as shown in Figure 3. In summary, G-equivariance constrains the number of parameters in the dynamics and policy functions, enabling more sample-efficient learning.

4 Symmetry in Sampling-based Model-based RL Algorithms

In this section, after confirming the effectiveness of symmetry in planning, we develop an equivariant model-based RL algorithm for continuous action spaces to exploit continuous symmetry. To plan in continuous spaces, we require *sampling-based* methods such as MPPI (Williams et al., 2015; 2017b), extending them to preserve equivariance. We build on prior work (Zhao et al., 2022b) that used value-based planning on a discrete state space $\mathbb{Z}^2$ and discrete group D_4, extending this work to the continuous case. The idea is to ensure that the algorithm $a_t = \texttt{plan}(s_t)$ produces the same actions up to transformations, i.e., it is G-equivariant: $g \cdot a_t \equiv g \cdot \texttt{plan}(s_t) = \texttt{plan}(g \cdot s_t)$, as shown in Figure 1. The principle is applicable for MDPs with other symmetry groups.

4.1 Components

We use TD-MPC (Hansen et al., 2022) as the backbone of our implementation and introduce their procedure and demonstrate how to incorporate symmetry into sampling-based planning algorithms.

- *Planning with learned models.* We use the MPPI (Model Predictive Path Integral) control method (Williams et al., 2015; 2016; 2017a;b), as adopted in TD-MPC (Hansen et al., 2022). We sample N trajectories with horizon H using the learned dynamics model, with actions from a learned policy, and estimate the expectation of total return.
- *Training models.* The learnable components in equivariant TD-MPC include: an `encoder` that processes input observation, `dynamics` and `reward` networks that simulate the MDP, and `value` and `policy` networks that guide the planning process.
- *Loss.* The only requirement is that loss is G-invariant. The loss terms in TD-MPC include value-prediction MSE loss and dynamics/reward-consistency MSE loss, which all satisfy invariance.

4.2 Integrating Symmetry

Zhao et al. (2022b) consider how the Bellman operator transforms under symmetry transformation. For sampling-based methods, one needs to consider how the sampling procedure changes under symmetry transformation. Specifically, under a symmetry transformation, differently sampled trajectories must transform equivariantly. This is shown in Figure 1. The equivariance of the transition model in sampling-based approches to machine learning has also been studied in (Park et al., 2022). There are several components that need G-equivariance, and we discuss them step-by-step and illustrate them in Figure 1.

1. **`dynamics` and `reward` model.** In the definition of symmetry in Geometric MDPs (and symmetric MDPs (Ravindran & Barto, 2004; van der Pol et al., 2020b; Zhao et al., 2022b)) in Equation 1, the transition and reward functions are G-equivariant and G-invariant respectively. Therefore, in implementation, the transition network is deterministic and uses a G-equivariant MLP, and the reward network is constrained to be G-invariant. Additionally, in implementation, planning is typically performed in latent space, using a latent dynamics model $\bar{f}(\boldsymbol{z}, \boldsymbol{a}) = \boldsymbol{z}'$. To do this, we require a G-equivariant encoder $h : \mathcal{S} \to \mathcal{Z}$, satisfying $\rho_{\mathcal{Z}}(g) \cdot h(\boldsymbol{s}) = h(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s})$. We omit the encoder in our description below for notational simplicity.

2. **`value` and `policy` model.** The optimal value function produces a scalar for each state and is G-invariant, while the optimal policy function is G-equivariant (Ravindran & Barto, 2004). If we use G-equivariant transition and G-invariant reward networks in updating our value function $\mathcal{T}[V_\theta] = \sum_{\boldsymbol{a}} R_\theta(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'} P_\theta(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) V_\theta(\boldsymbol{s}')$, the learned value network V_θ will also satisfy the symmetry constraint. Similarly, we can extract an optimal policy from the value network, which is also G-equivariant (van der Pol et al., 2020b; Wang et al., 2021; Zhao et al., 2022b).
3. **`MPC` procedure.** We consider equivariance in the MPC procedure in two parts: `sample` trajectories from the MDP using learned models, and compute their `returns`, $\texttt{return}(\texttt{sample}(s, \theta))$. We discuss the invariance and equivariance of it in the next subsection.

We list the equivariance or invariance conditions that each network needs to satisfy. Alternatively, for scalar functions, we can also say they transform under *trivial* representation ρ_0 and are thus invariant. All modules are implemented via G-steerable equivariant MLPs: $\rho_{\text{out}}(g) \cdot y = \rho_{\text{out}}(g) \cdot \texttt{MLP}(x) = \texttt{MLP}(\rho_{\text{in}}(g) \cdot x)$.

$$f_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S} : \qquad \rho_{\mathcal{S}}(g) \cdot f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = f_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{7}$$

$$R_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \qquad R_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{8}$$

$$Q_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \qquad Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{9}$$

$$\pi_\theta : \mathcal{S} \to \mathcal{A} : \qquad \rho_{\mathcal{A}}(g) \cdot \pi_\theta(\cdot \mid \boldsymbol{s}_t) = \pi_\theta(\cdot \mid \rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t) \tag{10}$$

4.3 Equivariance of `MPC`

We analyze how to constrain the underlying MPC planner to be equivariant. We use MPPI (Model Predictive Path Integral) (Williams et al., 2015; 2017a), which has been used in TD-MPC for action selection. An MPPI procedure samples multiple H-horizon trajectories $\{\tau_i\}$ from the current state $\boldsymbol{s}_t$ using the learned models. We use `sample` to refer to the procedure: $\tau_i \equiv \texttt{sample}(\boldsymbol{s}_t; f_\theta, R_\theta, Q_\theta, \pi_\theta) = (\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{s}_{t+H})$. Another procedure `return` computes the accumulated return, evaluating the value of a trajectory for top-k trajectories:

$$\texttt{return}(\tau) = \mathbb{E}_\tau \left[\gamma^H Q_\theta\left(\boldsymbol{s}_H, \boldsymbol{a}_H\right) + \sum_{t=0}^{H-1} \gamma^t R_\theta\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) \right] = \mathbb{E}_\tau \left[U(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H-1}) \right] \tag{11}$$

A trajectory is transformed element-wise by a transformation g: $g \cdot \tau_i = (g \cdot \boldsymbol{s}_t, g \cdot \boldsymbol{a}_t, g \cdot \boldsymbol{s}_{t+1}, g \cdot \boldsymbol{a}_{t+1}, \ldots, g \cdot \boldsymbol{s}_{t+H})$. However, since μ and σ in action sampling are *not state-dependent*, the MPPI `sample` does not exactly preserve equivariance: rotating the input does not *deterministically* guarantee a rotated output.

We propose a strategy to fix it. We consider the simplified case with a single time step, so the sampling draws N actions from a random Gaussian distribution $\mathcal{N}(\mu, \sigma^2 I)$, denoted as $\mathbb{A} = \{\boldsymbol{a}_i\}_{i=1}^N$. The return is simply $Q(s, a)$. Assuming we only select the best trajectory ($K = 1$), we require the following procedure to be equivariant: $a_0 = \arg\max_a Q(s_0, a)$. In other words, if we rotate state $g \cdot s_0$, the selected action is also rotated $g \cdot a_0$. Thus, a simple strategy is to augment the action sampling with G: $G\mathbb{A} = \{g \cdot a_i \mid g \in G\}_{i=1}^N$. We indicate that sampling strategy as G-`sample`.

Proposition 5 *The `return` procedure is G-invariant, and the G-augmented G-`sample` procedure that augment $\mathbb{A}$ using transformation in G is G-equivariant when $K = 1$.*

We further explain in Appendix E. In summary, for sampling and computing return, they satisfy the following conditions, indicating that the procedure $\texttt{return}(G\texttt{-sample}(s, \theta))$ is invariant, i.e., not changed under group transformation for any g. We use $\texttt{return}(\tau_i)$ to indicate the return of a specific trajectory τ_i and $g \cdot \tau_i$ to denote group action on it.

$$G\texttt{-sample} : \boldsymbol{s}_t, \theta \mapsto \tau_i : \qquad g \cdot \tau_i \sim G\texttt{-sample}(g \cdot \boldsymbol{s}_t; f_\theta, R_\theta, Q_\theta, \pi_\theta) \tag{12}$$

$$\texttt{return} : \tau_i \mapsto \mathbb{R} : \qquad \texttt{return}(\tau_i) = \texttt{return}(g \cdot \tau_i) \tag{13}$$

5 Evaluation: Sampling-based Planning

In this section, we present the setup and results for our proposed sampling-based planning algorithm: equivariant version of TD-MPC. The additional details and results are available in Appendix F.

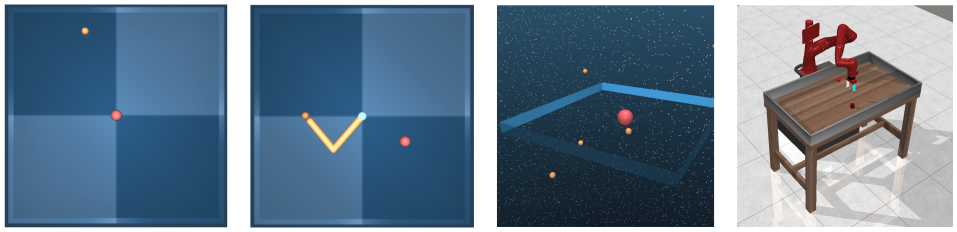

Figure 4: Tasks used in experiments: (1) `PointMass` in 2D, (2) `Reacher`, (3) Customized 3D version of `PointMass` with multiple particles to control, and (4) MetaWorld task to reach an object with gripper.

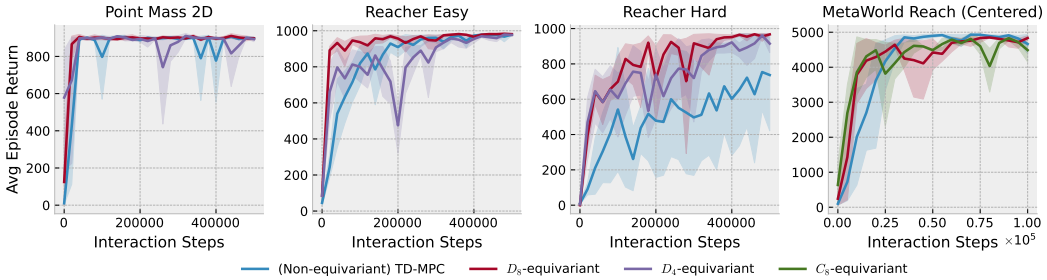

Figure 5: Results on 2D `PointMass`, `Reacher`, and MetaWorld `Reach` task.

Tasks. We verify the algorithm on a few selected and customized tasks using DeepMind Control suite (DMC) (Tassa et al., 2018), visualized in Figure 4. One task is 2D particle moving in $\mathbb{R}^2$, `PointMass`. We customize tasks based on it: (1) 3D particle moving in $\mathbb{R}^3$ (disabled gravity), and (2) 3D N-point moving that has several particles to control simutaneously. The goal is to move particle(s) to a target position. We also experiment with tasks on a two-link arm, `Reacher` (easy and hard), where the goal is to move the end-effector to a random position in a plane. `Reacher Easy` and `Hard` are top-down where the goal is to reach a random 2D position. If we rotate the MDP, the angle between the first and second links is not affected, i.e., it is G-invariant. The first joint and the target position are transformed under rotation, so we set it to ρ_1 standard representation (2D rotation matrices). The complete state and action representations are given in Table 1. The system has $\mathrm{O}(2)$ rotation and also reflection symmetry, hence we use D_8 and D_4 groups.

We also use MetaWorld tabletop manipulation (Yu et al., 2019). The action space is 3D gripper movement $(\Delta x, \Delta y, \Delta z)$ and 1D openness. The state space (1) gripper position, (2) 3D position plus 4D quaternion of at most 2 relevant objects, (3) 3D randomized goal position, depending on tasks. If we consider tasks with gravity, the MDP itself should exhibit $\mathrm{SO}(2)$ symmetry about the gravity axis. We make the origin at workspace center and the gripper initialized at the origin, so the task respects rotation equivariance around the origin.

Experimental setup. We compare against the non-equivariant version of TD-MPC (Hansen et al., 2022). Here, we by default make all components equivariant as described in the algorithm section. In Sec F.3, we include ablation studies for disabling or enabling each equivariant component. The training procedure follows TD-MPC (Hansen et al., 2022). We use the state as input and for equivariant TD-MPC, we divide the orignal hidden dimension by $\sqrt{N}$, where N is the group order, to keep the number of parameters roughly equal between the equivariant and non-equivariant versions. We mostly follow the original hyperparameters except for `seed_steps`. We use 5 random seeds for each method.

Algorithm setup: equivariance. We use discretized subgroups in implementing G-equivariant MLPs with the `escnn` package (Weiler & Cesa, 2021), which are more stable and easier to implement than continuous equivariance. For the 2D case, we use $\mathrm{O}(2)$ subgroups: dihedral groups D_4 and D_8 (4 or 8 rotation components), or rotation group C_8 (45° rotations). For the 3D case, we use the Icosahedral and Octahedral groups, which are finite subgroups of $\mathrm{SO}(3)$ with orders 60 and 24 respectively.

Results. In Figures 5 and 6, we show the reward curves and demonstrate that our equivariant methods can reach near-optimal performance 2x or 3x faster in terms of training interaction steps for

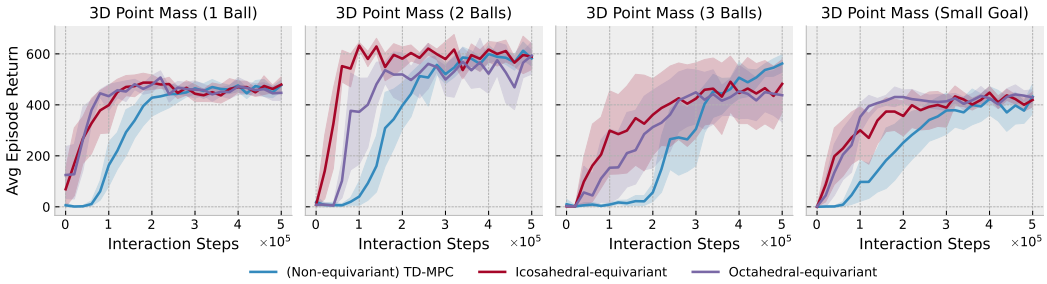

Figure 6: Results on a set of customized 3D N-ball `PointMass` tasks, with $N = 1, 2, 3$, and customized a 3D `PointMass` with smaller target.

several tasks. Recall the examples of Geometric MDPs, our theory not only motivates the algorithm design, but also gives an estimation of what tasks can benefit from (continuous) symmetry. The results justify the theoretical estimation on improvement of sample efficiency: less free parameters result in better regret bound and faster learning, as shown for LQR (Jedra & Proutiere, 2021).

In terms of the actual results, the default `PointMass` 2D version seems easy to solve, while the D_8-equivariant version learns slightly faster. For `Reacher`, as shown in Figure 5, D_8 outperforms the non-equivariant TD-MPC by noticeable margins, especially on the `Hard` domain. D_4 is slightly worse than D_8 but still better than the baseline. The rightmost subfigure shows a MetaWorld task `Reach`, which is to reach a button on a desk using the parallel gripper. We add $\mathrm{SO}(2)$ equivariance to the algorithm about the gravity axis and evaluate the C_8 and D_8-equivariant versions, which both give more efficient learning.

On `Reacher` tasks, we also compare against a *planning-free* baseline by removing MPPI planning with the learned model and only keep policy learning, shown in Fig 11, which is effectively similar to the DDPG algorithm (Lillicrap et al., 2016).

We design a set of harder 3D versions of `PointMass` and use $\mathrm{SO}(3)$ subgroups to implement 3D equivariant versions of TD-MPC, using Icosahedral- and Octahedral-equivariant MLPs. Figure 6 shows $N = 1, 2, 3$ balls in 3D `PointMass`, and the rightmost figure shows 1-ball 3D version with smaller target (0.02 compared to 0.03 in N-ball version). We find the Icosahedral (order 60) equivariant TD-MPC always learns faster and uses fewer samples to achieve best rewards, compared to the non-equivariant version. The Octahedral (order 24) equivariant version also performs similarly. The best absolute rewards in the 1-ball case is interestingly lower than 2- and 3-ball casees, which may be caused by higher possible return due to the presence of 2 or 3 balls that can reach the goal.

With higher-order 2D discrete subgroups, the performance plateaus but computational costs increase, so we use up to D_8. We also find TD-MPC is especially sensitive to a hyperparameter `seed_steps` that controls the number of warmup trajectories. In contrast, our equivariant version is robust to it and sometimes learn better with less warmup. In the shown curves, we do not use warmup across non-equivariant and equivariant ones and present additional results in Appendix F.

6 Conclusion and Discussion

In conclusion, we underscore the value of Euclidean symmetry in model-based RL algorithms. We define a subclass of MDPs, Geometric MDPs, prevalent in robotics, which exhibit additional structure. The linearized approximation of these MDPs adheres to steerable kernel constraints, substantially reducing parameter space. Drawing from this, we developed a model-based RL algorithm, utilizing Euclidean symmetry, that outperforms standard techniques in common RL benchmarks. This is the first method considering the importance of equivariance in sampling-based RL methods. It contributes to a deeper understanding of symmetry in RL algorithms and offers insights for future research. However, our theory and experience also show that while Euclidean symmetry can bring significant savings in parameters, it does not always offer practical benefits for some tasks with local coordinates. For instance, locomotion tasks do not greatly benefit from it. Also, our approach assumes the symmetry group is known, typically determined by the robot workspace dimension, usually 3D.

REFERENCES

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction Networks for Learning about Objects, Relations and Physics. *arXiv:1612.00222 [cs]*, December 2016. URL http://arxiv.org/abs/1612.00222. arXiv: 1612.00222.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261 [cs, stat]*, October 2018. URL http://arxiv.org/abs/1806.01261. arXiv: 1806.01261.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J. Bekkers, and Max Welling. Geometric and Physical Quantities Improve E(3) Equivariant Message Passing. *arXiv:2110.02905 [cs, stat]*, December 2021. URL http://arxiv.org/abs/2110.02905. arXiv: 2110.02905.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J. Bekkers, and Max Welling. Geometric and Physical Quantities Improve E(3) Equivariant Message Passing. *arXiv:2110.02905 [cs, stat]*, March 2022. URL http://arxiv.org/abs/2110.02905. arXiv: 2110.02905.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv:2104.13478 [cs, stat]*, April 2021. URL http://arxiv.org/abs/2104.13478. arXiv: 2104.13478.

Mitchell R. Cohen, Khairi Abdulrahim, and James Richard Forbes. Finite-horizon LQR control of quadrotors on $SE_2(3)$. *IEEE Robotics and Automation Letters*, 5(4):5748–5755, oct 2020a. doi: 10.1109/lra.2020.3010214. URL https://doi.org/10.1109%2Flra.2020.3010214.

Taco Cohen, Mario Geiger, and Maurice Weiler. A General Theory of Equivariant CNNs on Homogeneous Spaces. *arXiv:1811.02017 [cs, stat]*, January 2020b. URL http://arxiv.org/abs/1811.02017. arXiv: 1811.02017.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. 2016a. doi: 10.48550/ARXIV.1602.07576. URL https://arxiv.org/abs/1602.07576.

Taco S. Cohen and Max Welling. Steerable cnns. 2016b. doi: 10.48550/ARXIV.1612.08498. URL https://arxiv.org/abs/1612.08498.

Taco S. Cohen and Max Welling. Group Equivariant Convolutional Networks. *arXiv:1602.07576 [cs, stat]*, June 2016c. URL http://arxiv.org/abs/1602.07576. arXiv: 1602.07576.

Taco S. Cohen and Max Welling. Steerable CNNs. November 2016d. URL https://openreview.net/forum?id=rJQKYt5ll.

Carmel Domshlak, Michael Katz, and Alexander Shleyfman. Enhanced Symmetry Breaking in Cost-Optimal Planning as Forward Search. pp. 5. doi: 10/gq5m5r.

Andrew Dudzik and Petar Veličković. Graph Neural Networks are Dynamic Programmers. *arXiv:2203.15544 [cs, math, stat]*, March 2022. URL http://arxiv.org/abs/2203.15544. arXiv: 2203.15544.

A. Einstein. Zur elektrodynamik bewegter körper. *Annalen der Physik*, 322(10):891–921, 1905. doi: https://doi.org/10.1002/andp.19053221004. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19053221004.

Bryn Elesedy and Sheheryar Zaidi. Provably Strict Generalisation Benefit for Equivariant Models. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2959–2969. PMLR, July 2021. URL https://proceedings.mlr.press/v139/elesedy21a.html. ISSN: 2640-3498.

N. Ferns, P. Panangaden, and Doina Precup. Metrics for Finite Markov Decision Processes. In *AAAI*, 2004.

Daniel Fiser, Alvaro Torralba, and Alexander Shleyfman. Operator Mutexes and Symmetries for Simplifying Planning Tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):7586–7593, July 2019. ISSN 2374-3468. doi: 10/ghkkbq. URL https://ojs.aaai.org/index.php/AAAI/article/view/4751. Number: 01.

Maria Fox and Derek Long. The Detection and Exploitation of Symmetry in Planning Problems. In *In IJCAI*, pp. 956–961. Morgan Kaufmann, 1999.

Maria Fox and Derek Long. Extending the exploitation of symmetries in planning. In *In Proceedings of AIPS'02*, pp. 83–91, 2002.

Maani Ghaffari, Ray Zhang, Minghan Zhu, Chien Erh Lin, Tzu-Yuan Lin, Sangli Teng, Tingjun Li, Tianyi Liu, and Jingwei Song. Progress in symmetry preserving robot perception and control through geometry and learning. *Frontiers in Robotics and AI*, 9, 2022. ISSN 2296-9144. doi: 10.3389/frobt.2022.969380. URL https://www.frontiersin.org/articles/10.3389/frobt.2022.969380.

Matthew Hampsey, Pieter van Goor, Tarek Hamel, and Robert Mahony. Exploiting different symmetries for trajectory tracking control with application to quadrotors, 2022a.

Matthew Hampsey, Pieter van Goor, and Robert Mahony. Tracking control on homogeneous spaces: the equivariant regulator (eqr), 2022b.

Jiaqi Han, Wenbing Huang, Hengbo Ma, Jiachen Li, Joshua B. Tenenbaum, and Chuang Gan. Learning Physical Dynamics with Subequivariant Graph Neural Networks, October 2022. URL http://arxiv.org/abs/2210.06876. arXiv:2210.06876 [cs].

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal Difference Learning for Model Predictive Control. Technical Report arXiv:2203.04955, arXiv, March 2022. URL http://arxiv.org/abs/2203.04955. arXiv:2203.04955 [cs] type: article.

D. Husemöller. *Fibre Bundles*. Graduate Texts in Mathematics. Springer New York, 2013. ISBN 9781475722611. URL https://books.google.com/books?id=pCXlBwAAQBAJ.

Yassir Jedra and Alexandre Proutiere. Minimal Expected Regret in Linear Quadratic Control, September 2021. URL http://arxiv.org/abs/2109.14429. arXiv:2109.14429 [cs, eess, math, stat].

HENRIK KARSTOFT. Homogeneous connections and moduli spaces. *Mathematica Scandinavica*, 70(2):227–246, 1992. ISSN 00255521, 19031807. URL http://www.jstor.org/stable/24492007.

Risi Kondor and Shubhendu Trivedi. On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups. *arXiv:1802.03690 [cs, stat]*, November 2018. URL http://arxiv.org/abs/1802.03690. arXiv: 1802.03690.

Leon Lang and Maurice Weiler. A wigner-eckart theorem for group equivariant convolution kernels, 2020a. URL https://arxiv.org/abs/2010.10952.

Leon Lang and Maurice Weiler. A Wigner-Eckart Theorem for Group Equivariant Convolution Kernels. September 2020b. URL https://openreview.net/forum?id=ajOrOhQOsYx.

Lisa Lee, Emilio Parisotto, Devendra Singh Chaplot, Eric Xing, and Ruslan Salakhutdinov. Gated Path Planning Networks. *arXiv:1806.06408 [cs, stat]*, June 2018. URL http://arxiv.org/abs/1806.06408. arXiv: 1806.06408.

Lihong Li, Thomas J. Walsh, and M. Littman. Towards a Unified Theory of State Abstraction for MDPs. In *AI&M*, 2006.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning Particle Dynamics for Manipulating Rigid Bodies, Deformable Objects, and Fluids. April 2019. URL https://openreview.net/forum?id=rJgbSn09Ym.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. DDPG - Deep Deterministic Policy Gradient. pp. 14, 2016. ZSCC: NoCitationData[s0].

Guozheng Lu, Wei Xu, and Fu Zhang. On-Manifold Model Predictive Control for Trajectory Tracking on Robotic Systems. *IEEE Transactions on Industrial Electronics*, 70(9):9192–9202, September 2023. ISSN 1557-9948. doi: 10.1109/TIE.2022.3212397. Conference Name: IEEE Transactions on Industrial Electronics.

J.W. Milnor and J.D. Stasheff. *Characteristic Classes*. Annals of mathematics studies. Princeton University Press, 1974. ISBN 9780691081229. URL https://books.google.com/books?id=5zQ9AFk1i4EC.

Arnab Kumar Mondal, Pratheeksha Nair, and Kaleem Siddiqi. Group Equivariant Deep Reinforcement Learning. *arXiv:2007.03437 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2007.03437. arXiv: 2007.03437.

Shravan Matthur Narayanamurthy and Balaraman Ravindran. On the hardness of finding symmetries in Markov decision processes. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 688–695, Helsinki, Finland, 2008. ACM Press. ISBN 978-1-60558-205-4. doi: 10/bkswc2. URL http://portal.acm.org/citation.cfm?doid=1390156.1390243.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan Willem van de Meent, and Robin Walters. Learning Symmetric Embeddings for Equivariant World Models. *arXiv:2204.11371 [cs]*, April 2022. URL http://arxiv.org/abs/2204.11371. arXiv: 2204.11371.

Nir Pochter, Aviv Zohar, and Jeffrey S. Rosenschein. Exploiting Problem Symmetries in State-Based Planners. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, August 2011. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3732.

Balaraman Ravindran and Andrew G. Barto. Symmetries and Model Minimization in Markov Deision Proesses.

Balaraman Ravindran and Andrew G Barto. *An algebraic approach to abstraction in reinforcement learning*. PhD thesis, University of Massachusetts at Amherst, 2004.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. *arXiv:1806.01242 [cs, stat]*, June 2018. URL http://arxiv.org/abs/1806.01242. arXiv: 1806.01242.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) Equivariant Graph Neural Networks. *arXiv:2102.09844 [cs, stat]*, February 2021. URL http://arxiv.org/abs/2102.09844. arXiv: 2102.09844.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *arXiv:1911.08265 [cs, stat]*, November 2019. URL http://arxiv.org/abs/1911.08265. arXiv: 1911.08265.

J. P. Serre. Groupes finis, 2005. URL https://arxiv.org/abs/math/0503154.

Alexander Shleyfman, Michael Katz, Malte Helmert, Silvan Sievers, and Martin Wehrle. Heuristics and Symmetries in Classical Planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), March 2015. ISSN 2374-3468. doi: 10/gq5m5s. URL https://ojs.aaai.org/index.php/AAAI/article/view/9649. Number: 1.

Silvan Sievers. Structural Symmetries of the Lifted Representation of Classical Planning Tasks. pp. 8.

Silvan Sievers, Martin Wehrle, Malte Helmert, and Michael Katz. An Empirical Case Study on Symmetry Handling in Cost-Optimal Planning as Heuristic Search. In Steffen Hölldobler, Rafael Peñaloza, and Sebastian Rudolph (eds.), *KI 2015: Advances in Artificial Intelligence*, volume 9324, pp. 166–180. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24488-4 978-3-319-24489-1. doi: 10.1007/978-3-319-24489-1_13. URL http://link.springer.com/10.1007/978-3-319-24489-1_13. Series Title: Lecture Notes in Computer Science.

Silvan Sievers, Gabriele Röger, Martin Wehrle, and Michael Katz. Theoretical Foundations for Structural Symmetries of Lifted PDDL Tasks. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29:446–454, 2019. ISSN 2334-0843. doi: 10/gq5m5t. URL https://ojs.aaai.org/index.php/ICAPS/article/view/3509.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. ISBN 978-0-262-03924-6.

Aviv Tamar, YI WU, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value Iteration Networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c21002f464c5fc5bee3b98ced83963b8-Abstract.html.

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

Russ Tedrake. *Underactuated Robotics*. 2023. URL https://underactuated.csail.mit.edu.

Sangli Teng, William Clark, Anthony Bloch, Ram Vasudevan, and Maani Ghaffari. Lie algebraic cost function design for control on lie groups, 2022.

Sangli Teng, Dianhao Chen, William Clark, and Maani Ghaffari. An Error-State Model Predictive Control on Connected Matrix Lie Groups for Legged Robot Control, January 2023. URL http://arxiv.org/abs/2203.08728. arXiv:2203.08728 [cs, eess].

Tsung-Chih Tsai. Position Based Dynamics. In Newton Lee (ed.), *Encyclopedia of Computer Graphics and Games*, pp. 1–5. Springer International Publishing, Cham, 2017. ISBN 978-3-319-08234-9. doi: 10.1007/978-3-319-08234-9_92-1. URL https://doi.org/10.1007/978-3-319-08234-9_92-1.

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, 2020a.

Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling. MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning. *arXiv:2006.16908 [cs, stat]*, June 2020b. URL http://arxiv.org/abs/2006.16908. arXiv: 2006.16908.

Dian Wang, Robin Walters, and Robert Platt. $\mathrm{SO}(2)$-Equivariant Reinforcement Learning. September 2021. URL https://openreview.net/forum?id=7F9cOhdvfk_.

Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning Efficient Multi-agent Communication: An Information Bottleneck Approach. *arXiv:1911.06992 [cs]*, June 2020. URL http://arxiv.org/abs/1911.06992. arXiv: 1911.06992.

Maurice Weiler and Gabriele Cesa. General $E(2)$-Equivariant Steerable CNNs. *arXiv:1911.08251 [cs, eess]*, April 2021. URL http://arxiv.org/abs/1911.08251. arXiv: 1911.08251.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data, 2018a.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data, 2018b.

Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model Predictive Path Integral Control using Covariance Variable Importance Sampling. *arXiv:1509.01149 [cs]*, October 2015. URL http://arxiv.org/abs/1509.01149. arXiv: 1509.01149.

Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433–1440, May 2016. doi: 10/gf9knc.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model Predictive Path Integral Control: From Theory to Parallel Computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, February 2017a. ISSN 0731-5090, 1533-3884. doi: 10/f9vx74. URL https://arc.aiaa.org/doi/10.2514/1.G001921.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, Singapore, May 2017b. IEEE. ISBN 978-1-5090-4633-1. doi: 10/ggdv8n. URL https://ieeexplore.ieee.org/document/7989202/.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What Can Neural Networks Reason About? May 2019. URL https://arxiv.org/abs/1905.13211v4.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *arXiv:1910.10897 [cs, stat]*, October 2019. URL http://arxiv.org/abs/1910.10897. arXiv: 1910.10897.

A. Zee. *Group Theory in a Nutshell for Physicists*. In a Nutshell. Princeton University Press, 2016. ISBN 9780691162690. URL https://books.google.com/books?id=FWkujgEACAAJ.

Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson L. S. Wong. Toward Compositional Generalization in Object-Oriented World Modeling. In *ICML 2022*, April 2022a. URL http://arxiv.org/abs/2204.13661. arXiv: 2204.13661.

Linfeng Zhao, Xupeng Zhu, Lingzhi Kong, Robin Walters, and Lawson L. S. Wong. Integrating Symmetry into Differentiable Planning. In *ICLR 2023*. ICLR, June 2022b. doi: 10.48550/arXiv.2206.03674. URL http://arxiv.org/abs/2206.03674. arXiv:2206.03674 [cs] type: article.

Linfeng Zhao, Hongyu Li, Taskin Padir, Huaizu Jiang, and Lawson L. S. Wong. E(2)-equivariant graph planning for navigation, 2023a.

Linfeng Zhao, Huazhe Xu, and Lawson L. S. Wong. Scaling up and Stabilizing Differentiable Planning with Implicit Differentiation. In *ICLR 2023*, February 2023b. URL https://openreview.net/forum?id=PYbe4MoHf32.

Xupeng Zhu, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, and Robert Platt. Sample Efficient Grasp Learning Using Equivariant Models. *arXiv:2202.09468 [cs]*, February 2022. URL http://arxiv.org/abs/2202.09468. arXiv: 2202.09468.

Martin Zinkevich and Tucker Balch. Symmetry in Markov decision processes and its implications for single agent and multi agent learning. In *In Proceedings of the 18th International Conference on Machine Learning*, pp. 632–640. Morgan Kaufmann, 2001.

CONTENTS

A OUTLINE

The appendix is organized as follows: (1) additional discussion, including related work and theoretical background, (2) theory, derivation, and proofs, (3) implementation details and further empirical results, and (4) additional mathematical background.

B ADDITIONAL DISCUSSION

B.1 DISCUSSION: SYMMETRY IN DECISION-MAKING

In this work, we study the Euclidean symmetry $\mathrm{E}(d)$ from geometric transformations between *reference frames*. This is a specific set of symmetries that an MDP can have – isometric transformations

of Euclidean space $\mathbb{R}^d$, such as the distance is preserved. This can be viewed as a special case under the framework of MDP homomorphism, where symmetries relate two different MDPs via MDP *homomorphism* (or more strictly, *isomorphism*). We refer the readers to (Ravindran & Barto, 2004) for more details. We also discuss symmetry in other related fields.

Classic planning algorithms and model checking have leveraged the use of symmetry properties, (Fox & Long, 1999; 2002; Pochter et al., 2011; Domshlak et al.; Shleyfman et al., 2015; Sievers et al., 2015; Sievers; Sievers et al., 2019; Fiser et al., 2019) as evident from previous research. In particular, Zinkevich & Balch (2001) demonstrate that the value function of an MDP is invariant when symmetry is present. However, the utilization of symmetries in these algorithms presents a fundamental problem since they involve constructing equivalence classes for symmetric states, which is difficult to maintain and incompatible with differentiable pipelines for representation learning. Narayanamurthy & Ravindran (2008) prove that maintaining symmetries in trajectory rollout and forward search is intractable (NP-hard). To address the issue, recent research has focused on state abstraction methods such as the coarsest state abstraction that aggregates symmetric states into equivalence classes studied in MDP homomorphisms and bisimulation (Ravindran & Barto, 2004; Ferns et al., 2004; Li et al., 2006). However, the challenge lies in that these methods typically require perfect MDP knowledge and do not scale well due to the complexity of constructing and maintaining abstraction mappings (van der Pol et al., 2020a). To deal with the difficulties of symmetry in forward search, recent studies have integrated symmetry into reinforcement learning based on MDP homomorphisms (Ravindran & Barto, 2004), including van der Pol et al. (2020a) that integrate symmetry through an equivariant policy network. Furthermore, Mondal et al. (2020) previously applied a similar idea without using MDP homomorphisms. Park et al. (2022) learn equivariant transition models, but do not consider planning, and Zhao et al. (2022a) focuses on permutation symmetry in object-oriented transition models. Recent research by (Zhao et al., 2022b; 2023b) on 2D discrete symmetry on 2D grids has used a value-based planning approach.

There are some benefits of explicitly considering symmetry in continuous control. The possibility of hitting orbits is negligible, so there is no need for orbit-search on symmetric states in forward search in continuous control. Additionally, the planning algorithm implicitly plans in a smaller continuous MDP $\mathcal{M}/G$ (Ravindran & Barto, 2004). Furthermore, from equivariant network literature (Elesedy & Zaidi, 2021), the generalization gap for learned equivariant policy and value networks are smaller, which allows them to generalize better.

B.2 Additional Related Work

Geometric deep learning and equivariant networks. Geometric deep learning is a field that examines how to maintain geometric properties, such as symmetry and curvature, in data analysis (Bronstein et al., 2021). To preserve symmetries in data, researchers have developed equivariant neural networks. For instance, Cohen & Welling (2016c) introduced G-CNNs, followed by Steerable CNNs (Cohen & Welling, 2016d), which generalize scalar feature fields to vector fields and the induced representations. Moreover, Kondor & Trivedi (2018); Cohen et al. (2020b) have studied the theory on equivariant maps and convolutions for scalar fields through trivial representations and vector fields through induced representations, respectively. Furthermore, Weiler & Cesa (2021) propose $E(2)$-CNN, a method for solving kernel constraints for $E(2)$ and its subgroups, by decomposing into irreducible representations. Researchers have also explored how to use steerable features to maintain symmetries in deep learning models. For example, Brandstetter et al. (2022) developed steerable message-passing GNNs that use equivariant steerable features, while Satorras et al. (2021) use only invariant scalar features to build $E(n)$-equivariant graph networks. The idea of steerable features is further developed in Brandstetter et al. (2022), who propose steerable message passing graph networks for 3D space.

Learned dynamics model for interactive system. A framework for learning interaction dynamics between objects in a scene was proposed by Battaglia et al. (2016). This approach is based on relational inductive biases that consider the relationships among objects. Battaglia et al. (2018) later expanded on this framework by introducing relational networks that learn the dynamics between objects within a graph-based representation. Similarly, Sanchez-Gonzalez et al. (2018) developed a graph neural network model for physics simulation to learn dynamics in a graph-based representation. Furthermore, Li et al. (2019) introduced a particle-based dynamics network that focuses on

Table 2: (Copied from main text) Examples of geometric MDPs. G denotes the MDP symmetry group. $\mathcal{S}$ denotes the MDP state space. $\mathcal{A}$ denotes the MDP action space. We can quantitatively measure the saving of equivariance. "Images" refers to panoramic egocentric images $\mathbb{Z}^2 \to \mathbb{R}^{H\times W\times 3}$. $\circ$ denotes group element composition. We list the quotient space $\mathcal{S}/G$ to give intuition on saving. $Gx = \{g \cdot x \mid g \in G\}$ column shows the G-orbit space of $\mathcal{S}$ ($\cong$ denotes isomorphic to).

ID	G	$\mathcal{S}$	$\mathcal{A}$	$\mathcal{S}/G$	Gx	Task
1	C_4	$\mathbb{Z}^2$	C_4	$\mathbb{Z}^2/C_4$	C_4	2D Path Planning (Tamar et al., 2016)
2	C_4	Images	C_4	$\mathbb{Z}^2/C_4$	C_4	2D Visual Navigation (Zhao et al., 2022b)
3	$\mathrm{SO}(2)$	$\mathbb{R}^2$	$\mathbb{R}^2$	$\mathbb{R}^+$	S^1	2D Continuous Navigation
4	$\mathrm{SO}(3)$	$\mathbb{R}^3 \times \mathbb{R}^3$	$\mathbb{R}^3$	$\mathbb{R}^+ \times \mathbb{R}^3$	S^2	3D Free particle (with velocity)
5	$\mathrm{SO}(3)$	$\mathbb{R}^3 \rtimes \mathrm{SO}(3)$	$\mathbb{R}^3 \times \mathbb{R}^3$	$\mathbb{R}^+ \times \mathbb{R}^3$	S^2	Moving 3D Rigid Body
6	$\mathrm{SO}(2)$	$\mathrm{SO}(2)$	$\mathbb{R}^2$	$\{e\}$	S^1	Free Particle on $\mathrm{SO}(2) \cong S^1$ manifold
7	$\mathrm{SO}(3)$	$\mathrm{SO}(3)$	$\mathbb{R}^3$	$\{e\}$	S^2	Free Particle on $\mathrm{SO}(3)$ (Teng et al., 2023)
8	$\mathrm{SO}(2)$	$\mathrm{SE}(2)$	$\mathrm{SE}(2)$	$\mathbb{R}^2$	S^1	Top-down grasping (Zhu et al., 2022)
9	$\mathrm{SO}(2)$	$(S^1)^2 \times (\mathbb{R}^2)^2$	$\mathbb{R}^2$	$S^1 \times (\mathbb{R}^2)^2$	S^1	Two-arm manipulation (Tassa et al., 2018)

the physical interactions between particles in a simulation. This approach enables the generation of realistic animations and predictions of future states.

B.3 Limitations and Future Work

Although Euclidean symmetry group is infinite and seems huge, it does not guarantee significant performance gain in all cases. Our theory helps us understand when such Euclidean symmetry may not be very beneficial The key issue is that when a robot has kinematic constraints, Euclidean symmetry does not change those features, which means that equivariant constraints cannot share parameters and reduce dimensions. We empirically show this on using local vs. global reference frame in the additional experiment in Sec F. For further work, one possibility is to explicit consider constraints while keep using global positions.

B.4 Illustration and Examples of Geometric MDPs

In Figure 2, we present visual examples of Geometric MDPs and non-geometric MDPs in both discrete and continuous cases. Geometric MDP examples include moving a point robot in a 2D continuous space ($\mathbb{R}^2$, Example 3 in Table 2) or a discrete space ($\mathbb{Z}^2$, Example 1 (Tamar et al., 2016)), which is the abstraction of 2D discrete or continuous navigation. Table 2 includes more relevant examples. We use visual navigation over a 2D grid ($\mathbb{Z}^2 \rtimes C_4$, Example 2 (Lee et al., 2018; Zhao et al., 2022b)) as another example of a Geometric MDP. In this example, each position in $\mathbb{Z}^2$ and orientation in C_4 has an image in $\mathbb{R}^{H\times W\times 3}$, which is a *feature map* $\mathbb{Z}^2 \rtimes C_4 \to \mathbb{R}^{H\times W\times 3}$. The agent only navigates on the 2D grid $\mathbb{Z}^2$ (potentially with an orientation of C_4), but not the raw pixel space. Example (4) extends to the continuous 3D space and also include linear velocity $\mathbb{R}^3$. Alternatively, we can consider (5) moving a rigid body with $\mathrm{SO}(3)$ rotation. In (6) and (7), we consider moving free particle positions on $\mathrm{SO}(2), \mathrm{SO}(3)$, which are examples of optimal control *on manifold* in (Lu et al., 2023; Teng et al., 2023). Here, $G = \mathrm{SO}(3)$ acts on $\mathcal{S} = \mathrm{SO}(3)$ by group composition. (8) top-down grasping needs to predict $\mathrm{SE}(2)$ action on grasping an object on plane with $\mathrm{SE}(2)$ pose. It additionally has translation symmetry, so the state space is technically $\mathrm{SE}(2)/\mathrm{SE}(2) = \{e\}$. (9) is the `Reacher` task that we studies later, which controls a two-joint arm. It is easy to see that because two links are connected, kinematic constraints come in and equivariance does not save so much. Additionally, Example (3) is later implemented as `PointMass`, which is (8) top-down grasping without $\mathrm{SO}(2)$ rotation.

B.5 Continuous Group Actions for Geometric MDPs

If the G-actions on $\mathcal{S}$ and $\mathcal{A}$ are continuous, there is an interesting geometric interpretation based on fiber bundle theory (Husemöller, 2013). This requirement is not mandatory for implementation but allows for more rigorous theoretical results. Continuous symmetries correspond to conservation laws, while discrete (non-differentiable) symmetries do not have corresponding conservation laws

(Zee, 2016). The linearized dynamics of a system are much more constrained (i.e., have fewer free parameters) when the system has continuous G-actions. If the group G is a *compact* Lie group and has continuous G-actions, there exists a map $p : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{B}$ that projects the state-action space $\mathcal{S} \times \mathcal{A}$ to a lower dimensional base space $\mathcal{B}$ (Cohen et al., 2020b). The existence and smoothness of the projection p can be established using principal bundle theory. See Appendix D for more detail.

C Mathematical Background

C.1 Background for Representation Theory and G-steerable Kernels

We establish some notation and review some elements of group theory and representation theory. For a comprehensive review of group theory and representation theory, please see (Serre, 2005). The identity element of any group G will be denoted as e. We will always work over the field $\mathbb{R}$ unless otherwise specified.

C.2 Group Definition

A group is a non-empty set equipped with an associative binary operation $\cdot : G \times G \to G$ where $\cdot$ satisfies

$$\text{Existence of identity: } \exists e \in G, \text{ s.t. } \forall g \in G, \quad e \cdot g = g \cdot e = g$$
$$\text{Existence of inverse: } \forall g \in G, \exists g^{-1} \in G \text{ s.t. } g \cdot g^{-1} = g^{-1} \cdot g = e$$

For a complete reference on group theory, please see Zee (2016).

C.2.1 Group Representations

A group is an abstract object. Oftentimes, when working with groups, we are most interested in group *representations*. Let V be a vector space over $\mathbb{C}$. A *representation* (ρ, V) of G is a map $\rho : G \to \text{Hom}[V, V]$ such that

$$\forall g, g' \in G, \quad \forall v \in V, \quad \rho(g \cdot g')v = \rho(g) \cdot \rho(g')v$$

Concisely, a group representation is a embedding of a group into a set of matrices. The matrix embedding must obey the multiplication rule of the group. Over $\mathbb{R}$ and $\mathbb{C}$ all representations break down into irreducible representations Serre (2005). We will denote the set of irreducible representations of a group G and $\hat{G}$.

C.2.2 Group Actions

Let Ω be a set. A group action Φ of G on Ω is a map $\Phi : G \times \Omega \to \Omega$ which satisfies

$$\text{Identity: } \forall \omega \in \Omega, \quad \Phi(e, \omega) = \omega$$
$$\text{Compositionality: } \forall g_1, g_2 \in G, \quad \forall \omega \in \Omega, \quad \Phi(g_1 g_2, \omega) = \Phi(g_1, \Phi(g_2, \omega))$$

We will often suppress the Φ function and write $\Phi(g, \omega) = g \cdot \omega$.

$$\begin{array}{ccc} \Omega & \xrightarrow{\Psi} & \Omega' \\ \downarrow{\scriptstyle \Phi(g,\cdot)} & & \downarrow{\scriptstyle \Phi'(g,\cdot)} \\ \Omega & \xrightarrow{\Psi} & \Omega' \end{array}$$

Figure 7: Commutative Diagram For G-equivariant function: Let $\Phi(g, \cdot) : G \times \Omega \to \Omega$ denote the action of G on Ω. Let $\Phi'(g, \cdot) : G \times \Omega' \to \Omega'$ denote the action of G on Ω' The map $\Psi : \Omega \to \Omega'$ is G-equivariant if and only if the following diagram is commutative for all $g \in G$.

Let G have group action Φ on Ω and group action Φ' on Ω'. A mapping $\Psi : \Omega \to \Omega'$ is said to be G-equivariant if and only if

$$\forall g \in G, \forall \omega \in \Omega, \quad \Psi(\Phi(g, \omega)) = \Phi'(g, \Psi(\omega)) \tag{14}$$

Diagrammatically, Ψ is G-equivariant if and only if the diagram C.2.2 is commutative.

G-Intertwiners Let (ρ, V) and (σ, W) be two G-representations. The set of all G-equivariant linear maps between (ρ, V) and (σ, W) will be denoted as

$$\mathrm{Hom}_G[(\rho, V), (\sigma, W)] = \{\Phi \mid \Phi : V \to W, \ \forall g \in G, \ \Phi(\rho(g)v) = \sigma(g)\Phi(v)\}$$

Hom_G is a vector space over $\mathbb{C}$. When the linear maps are restricted to be real, Hom_G forms a vector space over $\mathbb{R}$. A linear map $\Phi \in \mathrm{Hom}_G[(\rho, V), (\sigma, W)]$ is said to *intertwine* the representations (ρ, V) and (σ, W). An intertwiner Φ is a map that makes the diagram C.2.2 commutative.

$$\begin{array}{ccc} (\rho, V) & \xrightarrow{\Phi} & (\sigma, W) \\ \downarrow{\scriptstyle \rho(g)} & & \downarrow{\scriptstyle \sigma(g)} \\ (\rho, V) & \xrightarrow{\Phi} & (\sigma, W) \end{array}$$

Figure 8: Commutative Diagram For G-intertwiner. The map $\Psi :\in \mathrm{Hom}_G[(\rho, V), (\sigma, W)]$ if and only if the following diagram is commutative for all $g \in G$.

Computing a basis for the vector space $\mathrm{Hom}_G[(\rho, V), (\sigma, W)]$ is an important procedure in the theory of steerable kernels (Cohen & Welling, 2016a).

C.2.3 Clebsch-Gordon Coefficients

Let G be a compact group. Let $\hat{G}$ be the irreducible representations of G Let (ρ, V) and (σ, W) be irreducible representations of G. The tensor product representation will not in general be irreducible and

$$(\rho, V) \otimes (\sigma, W) = \bigoplus_{\tau \in \hat{G}} c^{\tau}_{\sigma,\rho}(\tau, V_\tau)$$

where $c^{\tau}_{\sigma,\rho}$ are the Clebsch-Gordon multiplicities which count the number of copies of the irreducible (τ, V_τ) in the tensor product representation $(\rho, V) \otimes (\sigma, W)$. Clebsch-Gordon Coefficients $C^{\tau}_{\rho_1\rho_2}$ are the coefficients of the representation (τ, V_τ) in the tensor product basis. Specifically, let

$$|\tau i_\tau\rangle = \sum_{j_1=1}^{d_1} \sum_{j_2=1}^{d_2} \underbrace{\langle \rho_1 j_1, \rho_2 j_2 | \tau i_\tau \rangle}_{(C^{\tau}_{\rho_1\rho_2})_{i_\tau, j_1 j_2}} |\rho_1 j_1, \rho_2 j_2\rangle$$

Clebsch-Gordon coefficients are an integral part of the general solution to steerable kernel constraint Lang & Weiler (2020a).

C.2.4 Characterization of Steerable Kernels on Homogeneous Spaces

We briefly summarize the results of (Lang & Weiler, 2020a). Let X be a homogeneous space of a compact group G. Let $(\sigma, V_\sigma) \in \hat{G}$ and $(\rho, V_\rho) \in \hat{G}$ be two G-irreducibles. Consider the kernel constraint

$$K(g \cdot x) = \sigma(g) K(x) \rho(g^{-1})$$

where $K : X \to \mathrm{Hom}[V_\rho, V_\sigma]$. Then, there exists a set of generalized spherical harmonics $Y^i_{\rho,k} : X \to \mathbb{R}$ where $\rho \in \hat{G}$ is a G-irreducible and the index $i \in \{1, 2, ..., d_\rho\}$ and $k \in \{1, 2, ..., m_\rho\}$ where $m_\rho \leq d_\rho$ is called the muplicity which satisfy the relation

$$\forall g \in G, \quad Y^i_{\rho,k}(g^{-1} \cdot x) = \sum_{i'=1}^{d_\rho} \rho_{ii'}(g) Y^{i'}_{\rho,k}(x)$$

The set of $Y^i_{\rho,k}$ form a basis for all square integrable functions on X.

Let us define

$$K^{\tau ks}_{\sigma\rho}(x) = \sum_{i_\tau=1}^{d_\tau} \sum_{j_\sigma=1}^{d_\sigma} \sum_{i_\rho=1}^{d_\rho} |\sigma j_\sigma\rangle \underbrace{\langle s, \sigma j_\sigma | \tau i_\tau, \rho i_\rho \rangle}_{\text{Clebsch-Gordon}} \underbrace{Y^{i_\rho}_{\rho,k}(x)}_{\text{harmonics}} \langle \rho i_\rho |$$

Then, using the main result of (Lang & Weiler, 2020a), the matrices $K_{\sigma\rho}^{\tau ks}(x)$ form a basis for the space of G-steerable kernels with input representation ρ and output representation σ. Then, the kernel K can be written in the form

$$K_{\sigma\rho}(x) = \sum_{\tau\in\hat{G}} \sum_{k=1}^{m_\rho} \sum_{s=1}^{m_\sigma} c^{\tau ks} K_{\sigma\rho}^{\tau ks}(x)$$

where $c_{\tau ks} \in \text{Hom}_G[(\sigma, V_\sigma), (\sigma, V_\sigma)]$ is a (σ, V_σ)-endomorphism. The total number of free parameters in $K_{\sigma\rho}$ is

$$\dim K_{\sigma\rho} = m_\rho m_\sigma \sum_{\tau\in\hat{G}} C_{\tau\rho}^{\sigma} \times \dim \text{Hom}_G[(\sigma, V_\sigma), (\sigma, V_\sigma)] \leq 4 m_\rho m_\sigma \sum_{\tau\in\hat{G}} c_{\tau\rho}^{\sigma}$$

which depends on both multiplicity m_τ of the homogeneous space X and the Clebsch-Gordon Coefficients $c_{\tau\rho}^{\sigma}$ of the group G.

D Theory and Proofs

The section is organized as follows. We first give the proofs to Theorem 1 and 2. Then, we discuss how we linearize dynamics of a Geometric MDP and G-steerable kernels in detail. The goal of the theory is to show that, in linearized case, Euclidean symmetry can provably *reduce* number of free parameters and the dimensions of the solution space. Under RL setup with *unknown* dynamics (and cost) function, the Euclidean equivariance constraints then potentially bring significant benefit because of *less parameters*.

D.1 Theorem 1 and 2: Equivariance in Geometric MDPs

Theorem 1 *The Bellman operator of a GMDP is equivariant under Euclidean group* $\mathrm{E}(d)$.

Proof. The Bellman (optimality) operator is defined as

$$\mathcal{T}[V](\boldsymbol{s}) := \max_{\boldsymbol{a}} R(\boldsymbol{s}, \boldsymbol{a}) + \int d\boldsymbol{s}' P(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}) V(\boldsymbol{s}'), \tag{15}$$

where the input and output of the Bellman operator are both value function $V : \mathcal{S} \to \mathbb{R}$. The theorem directly generalizes to Q-value function.

Under group transformation g, a feature map (field) $f : X \to \mathbb{R}^{c_{\text{out}}}$ is transformed as:

$$\left[L_g f\right](x) = \left[f \circ g^{-1}\right](x) = \rho_{\text{out}}(g) \cdot f\left(g^{-1} x\right), \tag{16}$$

where ρ_{out} is the G-representation associated with output $\mathbb{R}^{c_{\text{out}}}$. For the *scalar* value map, ρ_{out} is identity, or trivial representation.

For any group element $g \in \mathrm{E}(d) = \mathbb{R}^d \rtimes \mathrm{O}(d)$, we transform the Bellman (optimality) operator step-by-step and show that it is equivariant under $\mathrm{E}(d)$:

$$L_g\left[\mathcal{T}[V]\right](\boldsymbol{s}) \overset{(1)}{=} \mathcal{T}[V](g^{-1}\boldsymbol{s}) \tag{17}$$

$$\overset{(2)}{=} \max_{\boldsymbol{a}} R(g^{-1}\boldsymbol{s}, \boldsymbol{a}) + \int d\boldsymbol{s}' \cdot P(\boldsymbol{s}' \mid g^{-1}\boldsymbol{s}, \boldsymbol{a}) V(\boldsymbol{s}') \tag{18}$$

$$\overset{(3)}{=} \max_{\bar{\boldsymbol{a}}} R(g^{-1}\boldsymbol{s}, g^{-1}\bar{\boldsymbol{a}}) + \int d(g^{-1}\bar{\boldsymbol{s}}) \cdot P(g^{-1}\bar{\boldsymbol{s}} \mid g^{-1}\boldsymbol{s}, g^{-1}\boldsymbol{a}) V(g^{-1}\bar{\boldsymbol{s}}) \tag{19}$$

$$\overset{(4)}{=} \max_{\bar{\boldsymbol{a}}} R(\boldsymbol{s}, \bar{\boldsymbol{a}}) + \int d(g^{-1}\bar{\boldsymbol{s}}) \cdot P(\bar{\boldsymbol{s}} \mid \boldsymbol{s}, \boldsymbol{a}) V(g^{-1}\bar{\boldsymbol{s}}) \tag{20}$$

$$\overset{(5)}{=} \max_{\bar{\boldsymbol{a}}} R(\boldsymbol{s}, \bar{\boldsymbol{a}}) + \int d\bar{\boldsymbol{s}} \cdot P(\bar{\boldsymbol{s}} \mid \boldsymbol{s}, \boldsymbol{a}) V(g^{-1}\bar{\boldsymbol{s}}) \tag{21}$$

$$\overset{(6)}{=} \mathcal{T}[L_g[V]](\boldsymbol{s}) \tag{22}$$

For each step:

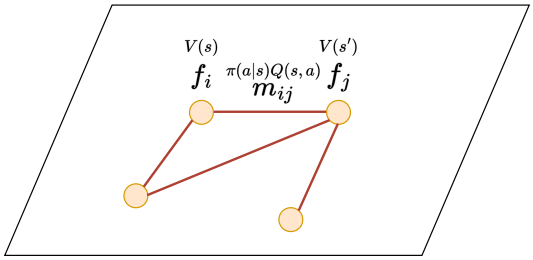

Figure 9: Demonstrate the idea of implementing value iteration with geometric message passing.

- (1) By definition of the (left) group action on the feature map $V : \mathcal{S} \to \mathbb{R}$, such that $g \cdot V(\boldsymbol{s}) = \rho_0(g)V(g^{-1}\boldsymbol{s}) = V(g^{-1}\boldsymbol{s})$. Because V is a scalar feature map, the output transforms under trivial representation $\rho_0(g) = \mathrm{Id}$.
- (2) Substitute in the definition of Bellman operator.
- (3) Substitute $\boldsymbol{a} = g^{-1}(g\boldsymbol{a}) = g^{-1}\bar{\boldsymbol{a}}$. Also, substitute $g^{-1}\bar{\boldsymbol{s}} = \boldsymbol{s}'$.
- (4) Use the symmetry properties of Geometric MDP: $P(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}) = P(g \cdot \boldsymbol{s} \mid g \cdot \boldsymbol{s}, g \cdot \boldsymbol{a})$ and $R(\boldsymbol{s}, \boldsymbol{a}) = R(g \cdot \boldsymbol{s}, g \cdot \boldsymbol{a})$.
- (5) Because $g \in \mathrm{E}(d)$ is isometric transformations (translations $\mathbb{R}^d$, rotations and reflections $\mathrm{O}(d)$) and the state space carries group action, the measure ds is a G-invariant measure $d(gs) = ds$. Thus, $d\bar{\boldsymbol{s}} = d(g^{-1}\bar{\boldsymbol{s}})$.
- (6) By the definition of the group action on V.

The proof requires the MDP to be a Geometric MDP with Euclidean symmetry and the state space carries a group action of Euclidean group. Therefore, the Bellman operator of a Geometric MDP is $\mathrm{E}(d)$-equivariant. Additionally, we can also parameterize the dynamics and reward functions with neural networks, and the learned Bellman operator is also equivariant.

The proof is analogous to the case in (Zhao et al., 2022b), where the symmetry group is $p4m = \mathbb{Z}^2 \rtimes D_4$, which is a discretized subgroup of $\mathrm{E}(2)$. A similar statement can also be found in symmetric MDP (Zinkevich & Balch, 2001), MDP homomorphism induced from symmetry group (Ravindran & Barto, 2004), and later work on symmetry in deep RL (van der Pol et al., 2020b; Wang et al., 2021).

Theorem 2 *For a GMDP, value iteration is an* $\mathrm{E}(d)$*-equivariant geometric message passing.*

Proof. We prove by constructing value iteration with For a more rigorous account on the relationship between dynamic programming (DP) and message passing on *non-geometric* MDPs, see (Dudzik & Veličković, 2022).

Notice that they satisfy the following equivariance conditions:

$$P_\theta : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+ : \quad P_\theta(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t) = P_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_{t+1} \mid \rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{23}$$

$$R_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad R_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{24}$$

$$Q_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{25}$$

$$V_\theta : \mathcal{S} \to \mathbb{R} : \quad V_\theta(\boldsymbol{s}_t) = V_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t) \tag{26}$$

$$\tag{27}$$

We construct geometric message passing such that it uses *scalar* messages and features and resembles value iteration. The idea is visualized in Fig 9.

Then, we can use geometric message passing network to construct value iteration, which is to iteratively apply Bellman operators. We adopt the definition of geometric message passing based on

(Brandstetter et al., 2021) as follows.

$$\tilde{\mathbf{m}}_{ij} = \phi_m \left(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \tilde{\mathbf{a}}_{ij} \right) \tag{28}$$

$$\tilde{\mathbf{f}}'_i = \phi_f \left(\tilde{\mathbf{f}}_i, \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{m}}_{ij}, \tilde{\mathbf{a}}_i \right). \tag{29}$$

The tilde means they are steerable under G transformations.

We want to construct value iteration:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \tag{30}$$

$$V'(s) = \sum_{a} \pi(a|s) Q(s,a) \tag{31}$$

To construct a *geometric graph*, we let vertices $\mathcal{V}$ be states $\boldsymbol{s}$ and edges $\mathcal{E}$ be state-action transition $(\boldsymbol{s}, \boldsymbol{a})$ labelled by $\boldsymbol{a}$. For the geometric features on the graph, there are node features and edge features. Node features include maps/functions on the state space: $\mathcal{S} \to \mathbb{R}^D$, and edge features include functions on the state-action space $\mathcal{S} \times \mathcal{A} \to \mathbb{R}^D$.

For example, state value function $V : \mathcal{S} \to \mathbb{R}$ is (scalar) node feature, and Q-value function $Q_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and reward function $R_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are edge features. The message $\tilde{\mathbf{m}}_{ij}$ is thus a scalar for every edge: $\tilde{\mathbf{m}}_{ij} = \pi(a|s)Q(s,a)$, and $\tilde{\mathbf{f}}'_i$ is updated value function $\tilde{\mathbf{f}}'_i = V'(s)$. It is possible to extend value iteration to vector form as in Symmetric Value Iteration Network and Theorem 5.2 in (Zhao et al., 2022b), while we leave it for future work.

D.2 Linear-Quadratic Control: Linearizing Geometric MDPs

Linear-Quadratic Regulator (LQR) is one of the most frequently used methods in optimal control Tedrake (2023). LQR is a computationally efficient method for solving problems with linear dynamics and quadratic costs. LQR has various noise robustness and optimally guarantees. Even if the dynamical system is nonlinear, linear-quadratic control methods have been used after iteratively linearizing the dynamics and quadratizing the cost.

Many of the problems where LQR is applied have symmetries. Recently has the control community began to study how symmetry can be used to increase the performance of classical control algorithms Teng et al. (2023; 2022); Ghaffari et al. (2022). Hampsey et al. (2022b;a); Cohen et al. (2020a) specifically consider LQR on homogeneous spaces but do not establish the connection to steerable kernels.

We show how Euclidean symmetry inherently simplifies the linearized problem. We assume the dynamics and cost (reward) are **unknown** and need to be *learned*, thus equivariance constraints come in and *reduce the number of free parameters*.

D.3 Theorem 3: Equivariance of Linearized Dynamics

We show the derivation of steerable kernel constraint in this subsection and further discuss the characteristics of steerable kernels in the next subsection.

Theorem 3: *If a Geometric MDP has an infinitesimal G-action on the state-action space $\mathcal{S} \times \mathcal{A}$, the linearized dynamics is also G-equivariant: the matrix-value functions $A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}$ and $B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{A}}$ satisfy G-steerable kernel constraints.*

Oftentimes, the problem of interest has continuous symmetry G that acts on the space $\mathcal{S} \times \mathcal{A}$. We will assume that the Lie group action $G \times (\mathcal{S} \times \mathcal{A}) \to \mathcal{S} \times \mathcal{A}$ is continuous.

Under infinitesimal symmetry transformation $g \approx 1_G \in G$, let us suppose that the state and action space transform as

$$\boldsymbol{s} \to \rho_\mathcal{S}(g) \cdot \boldsymbol{s}, \quad \boldsymbol{a} \to \rho_\mathcal{A}(g) \cdot \boldsymbol{a}$$

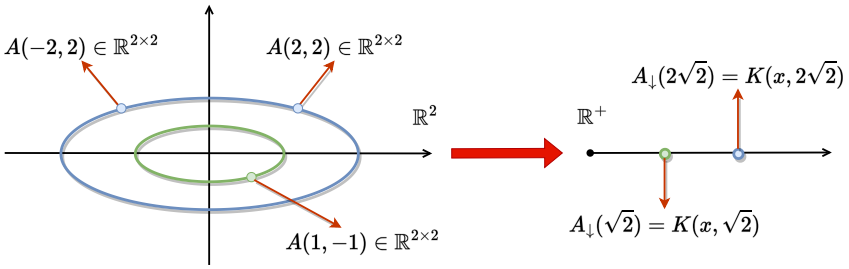

Figure 10: *Adopted from main text.* A schematic showing how a matrix-valued kernel $A : X \to \mathbb{R}^{2\times 2}$ is constrained by the $\mathrm{SO}(2)$-steerable kernel constraints on a set of orbits $A(g \cdot p) = \rho_{\text{out}}(g)A(p)\rho_{\text{in}}(g^{-1})$. On each orbit (left), the constraints can be solved exactly: the matrices on same orbits (same colors) are related and have explicit parameterization given in Lang & Weiler (2020a). Thus, these matrices can be spanned on a basis (denoted by K) and live in a smaller "base" space $\mathcal{B} = X/G$ with a certain form $A_\downarrow : \mathcal{B} \to \mathbb{R}^{2\times 2}$.

where $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$ are representations of the group G. Let A and B be the lineariziations of the dynamics f at the point $p = (\boldsymbol{s}, \boldsymbol{a})$, with

$$A(p) = \frac{\partial f}{\partial s}|_p, \quad B(p) = \frac{\partial f}{\partial a}|_p$$

Under infinitesimal symmetry transformation $g \approx 1_G \in G$, the dynamics must satisfy,

$$\rho_{\mathcal{S}}(g) \cdot f(\boldsymbol{s}, \boldsymbol{a}) = f(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}) \tag{32}$$

Because we assume the state and action space has *continuous* group action, we can apply Taylor expansion for continuous group actions and only keep the first order term. Let $g = 1_G + \delta g + \mathcal{O}(\delta g^2)$ be the expansion around the identity element, then

$$\rho_{\mathcal{S}}(g) = \rho_{\mathcal{S}}(1_G) + \rho_{\mathcal{S}}(\delta g) + \mathcal{O}(\delta g^2) = \mathbf{1}_{d_{\mathcal{S}}} + \rho_{\mathcal{S}}(\delta g) + \mathcal{O}(\delta g^2)$$
$$\rho_{\mathcal{A}}(g) = \rho_{\mathcal{A}}(1_G) + \rho_{\mathcal{A}}(\delta g) + \mathcal{O}(\delta g^2) = \mathbf{1}_{d_{\mathcal{A}}} + \rho_{\mathcal{A}}(\delta g) + \mathcal{O}(\delta g^2)$$

inserting into 32, and collecting terms of order $\mathcal{O}(\delta g)$, we have that,

$$\rho_{\mathcal{S}}(\delta g)\frac{\partial f}{\partial s}|_{\delta p} = \frac{\partial f}{\partial s}|_{\delta g \cdot p}\rho_{\mathcal{S}}(\delta g), \quad \rho_{\mathcal{S}}(\delta g)\frac{\partial f}{\partial a}|_p = \frac{\partial f}{\partial a}|_{\delta g \cdot p}\rho_{\mathcal{S}}(\delta g)$$

In general, we solve the non-linear problem by linearizing and iterating. Let us consider the linearized problem at point $p = (\boldsymbol{s}_0, \boldsymbol{a}_0)$. Assuming that the state and control do not change too drastically over a short period of time, we can approximate the true dynamics as

$$\boldsymbol{s}_{t+1} = A(p) \cdot \boldsymbol{s}_t + B(p) \cdot \boldsymbol{a}_t$$

where $A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_{\mathcal{S}} \times d_{\mathcal{S}}}$ and $B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_{\mathcal{S}} \times d_{\mathcal{A}}}$.

Now, linearizing the dynamics f and using the symmetry constraint. The matrix valued functions $A(p)$ and $B(p)$ must satisfy the constraints

$$\forall g \in G, \quad A(g \cdot p) = \rho_{\mathcal{S}}(g)A(p)\rho_{\mathcal{S}}(g^{-1}), \quad B(g \cdot p) = \rho_{\mathcal{S}}(g)B(p)\rho_{\mathcal{A}}(g^{-1}) \tag{33}$$

so that A is a G-steerable kernel with input representation $\rho_{\mathcal{S}}$ and output representation $\rho_{\mathcal{S}}$ and B is a G-steerable kernel with input representation $\rho_{\mathcal{A}}$ and output representation $\rho_{\mathcal{S}}$ Cohen & Welling (2016a). The kernel constraints 33 relate $A(p)$ and $B(p)$ at different points. We use this constraint to understand why Euclidean symmetry is beneficial for decision-making: it *reduces* the number of free parameters and the dimensions of the solution space.

D.4 Characteristics of G-Steerable Kernels

Consider the G-steerable kernel constraints,

$$\forall g \in G, \quad A(g \cdot p) = \rho_{\mathcal{S}}(g)A(p)\rho_{\mathcal{S}}(g^{-1}), \quad B(g \cdot p) = \rho_{\mathcal{S}}(g)B(p)\rho_{\mathcal{A}}(g^{-1}) \tag{34}$$

To begin, note that this equation only relates A and B on points that can be related by G-transformation. As first observed in Weiler et al. (2018a), we can solve the constraints 34 over each orbit separately. Let us define the G-orbits of $\mathcal{S} \times \mathcal{A}$ as

$$O(x) = \{ \ y \ | \ \exists g \in G, \ \ y = g \cdot x \ \} \tag{35}$$

where $x \in \mathcal{S} \times \mathcal{A}$. Every G-orbit is a homogeneous space of G Serre (2005). The set of G-orbits form a partition of the space $\mathcal{S} \times \mathcal{A}$. Let us define the equivalence relation $\sim$ as

$$x \sim y \implies \exists g \in G, \text{ such that } x = g \cdot y$$

so that $x \sim y$ only if x and y are related by symmetry transformation. We then define the quotient space

$$\mathcal{B} = (\mathcal{S} \times \mathcal{A}) / \sim$$

where the space $\mathcal{B}$ consists of the space of all G-orbit representatives. Using a standard result in topology Husemöller (2013), there is then a canonical continuous projection map $\Pi : \mathcal{S} \times \mathcal{A} \to \mathcal{B}$ which projects each point in $\mathcal{S} \times \mathcal{A}$ to a canonically chosen orbit representative. Note that points related by G-action have the same projection and

$$\forall g \in G, \quad \Pi(g \cdot x) = \Pi(x)$$

holds for all $x \in \mathcal{S} \times \mathcal{A}$. We can decompose every point in the space $\mathcal{S} \times \mathcal{A}$ into an element of $\mathcal{B}$ and a element of a homogeneous space of G. Let $p \in \mathcal{S} \times \mathcal{A}$, we can always write

$$p = x_p \times p_\downarrow$$

where $p_\downarrow = \Pi(p) \in \mathcal{B}$ is the orbit representative of p and x_p is an element of $O(p_\downarrow)$ which is a homogeneous space of G. Now, using this decomposition, we may write the constraints 33 as

$$\forall g \in G, \quad A(g \cdot x, p_\downarrow) = \rho_{\mathcal{S}}(g) A(x, p_\downarrow) \rho_{\mathcal{S}}(g^{-1}), \quad B(g \cdot x, p_\downarrow) = \rho_{\mathcal{S}}(g) B(x, p_\downarrow) \rho_{\mathcal{A}}(g^{-1}),$$

A complete solution to this constraint for any compact group G was given in Lang & Weiler (2020a). Following Weiler et al. (2018a), we can simplify these kernel constraints based on the decomposition of $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$ into irreducibles. Let $\hat{G}$ denote a representative set of G-irreducibles. Let us suppose that $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$ decompose into irreducibles of G as

$$(\rho_{\mathcal{S}}, V_{\mathcal{S}}) = \bigoplus_{\sigma \in \hat{G}} n_\sigma (\sigma, V_\sigma), \quad (\rho_{\mathcal{A}}, V_{\mathcal{A}}) = \bigoplus_{\sigma \in \hat{G}} q_\sigma (\sigma, V_\sigma)$$

where (σ, V_σ) are the G-irreducibles and n_σ and q_σ count the multiplicity of the (σ, V_σ) irreducible in $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$, respectively. The dimensions of each irreducible are related to the dimensions of the $\mathcal{S}$ and $\mathcal{A}$ manifolds via

$$d_{\mathcal{S}} = \sum_{\sigma \in \hat{G}} d_\sigma n_\sigma, \qquad d_{\mathcal{A}} = \sum_{\sigma \in \hat{G}} d_\sigma q_\sigma$$

Now, by definition of an reducible representation, there exists a $d_{\mathcal{S}} \times d_{\mathcal{S}}$ unitary matrix U and a $d_{\mathcal{A}} \times d_{\mathcal{A}}$ unitary matrix V such that we change basis and write

$$\forall g \in G, \quad \rho_{\mathcal{S}}(g) = U \begin{bmatrix} n_1\sigma_1(g) & 0 & 0 & \dots & 0 & 0 \\ 0 & n_2\sigma_2(g) & 0 & \dots & 0 & 0 \\ 0 & 0 & n_3\sigma_3(g) & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & & 0 & \dots & 0 & n_{|\hat{G}|}\sigma_{|\hat{G}|}(g) \end{bmatrix} U^\dagger$$

and

$$\forall g \in G, \quad \rho_{\mathcal{A}}(g) = V \begin{bmatrix} q_1\sigma_1(g) & 0 & 0 & \dots & 0 & 0 \\ 0 & q_2\sigma_2(g) & 0 & \dots & 0 & 0 \\ 0 & 0 & q_3\sigma_3(g) & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & & 0 & \dots & 0 & q_{|\hat{G}|}\sigma_{|\hat{G}|}(g) \end{bmatrix} V^\dagger$$

where the notation $b_i\sigma_i(g)$ denotes a block matrix with b_i copies of the irreducible $(\sigma_i, V_i) \in \hat{G}$ on the diagonals,

$$\forall g \in G, \quad b_i\sigma_i(g) = \underbrace{\begin{bmatrix} \sigma_i(g) & 0 & 0 & ... & 0 & 0 \\ 0 & \sigma_i(g) & 0 & ... & 0 & 0 \\ 0 & 0 & \sigma_i(g) & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & & 0 & ... & 0 & \sigma_i(g) \end{bmatrix}}_{b_i \text{ copies}}$$

We can write down the solution explicitly in this basis,

$$A(x, p_\downarrow) = U \begin{bmatrix} n_1 n_1 K^A_{11}(x, p_\downarrow) & n_1 n_2 K^A_{12}(x, p_\downarrow) & n_1 n_3 K^A_{13}(x, p_\downarrow) & ... & n_1 n_{|\hat{G}|} K^A_{1|\hat{G}|}(x, p_\downarrow) \\ n_2 n_1 K^A_{21}(x, p_\downarrow) & n_2 n_2 K^A_{22}(x, p_\downarrow) & n_2 n_3 K^A_{23}(x, p_\downarrow) & ... & n_2 n_{|\hat{G}|} K^A_{2|\hat{G}|}(x, p_\downarrow) \\ ... & ... & ... & ... & ... \\ n_{|\hat{G}|} n_1 K^A_{|\hat{G}|1}(x, p_\downarrow) & n_{|\hat{G}|} n_2 K^A_{|\hat{G}|2}(x, p_\downarrow) & n_{|\hat{G}|} n_3 K^A_{|\hat{G}|3}(x, p_\downarrow) & ... & n_{|\hat{G}|} n_{|\hat{G}|} K^A_{|\hat{G}||\hat{G}|}(x, p_\downarrow) \end{bmatrix} U^\dagger$$

and

$$B(x, p_\downarrow) = U \begin{bmatrix} n_1 q_1 K^B_{11}(x, p_\downarrow) & n_1 q_2 K^B_{12}(x, p_\downarrow) & n_1 q_3 K^B_{13}(x, p_\downarrow) & ... & n_1 q_{|\hat{G}|} K^B_{1|\hat{G}|}(x, p_\downarrow) \\ n_2 q_1 K^B_{21}(x, p_\downarrow) & n_2 q_2 K^B_{22}(x, p_\downarrow) & n_2 q_3 K^B_{23}(x, p_\downarrow) & ... & n_2 q_{|\hat{G}|} K^B_{2|\hat{G}|}(x, p_\downarrow) \\ ... & ... & ... & ... & ... \\ n_{|\hat{G}|} q_1 K^B_{|\hat{G}|1}(x, p_\downarrow) & n_{|\hat{G}|} q_2 K^B_{|\hat{G}|2}(x, p_\downarrow) & q_{|\hat{G}|} n_3 K^B_{|\hat{G}|3}(x, p_\downarrow) & ... & n_{|\hat{G}|} q_{|\hat{G}|} K^B_{|\hat{G}||\hat{G}|}(x, p_\downarrow) \end{bmatrix} V^\dagger$$

where K^A_{ij} are the A-kernels with input representation (σ_i, V_i) and output representation (σ_j, V_j) and K^B_{ij} are the B-kernels with input representation (σ_i, V_i) and output representation (σ_j, V_j). The notation $b_i c_j K^C_{ij}(x)$ denotes $b_i \times c_j$ independent copies of the K^C_{ij} kernel,

$$b_i c_j K^C_{ij}(x, p_\downarrow) = \underbrace{\left.\begin{bmatrix} K^C_{ij}(x, p_\downarrow) & K^C_{ij}(x, p_\downarrow) & K^C_{ij}(x, p_\downarrow) & ... & K^C_{ij}(x, p_\downarrow) \\ K^C_{ij}(x, p_\downarrow) & K^C_{ij}(x, p_\downarrow) & K^C_{ij}(x, p_\downarrow) & ... & K^C_{ij}(x, p_\downarrow) \\ ... & ... & ... & ... & ... \\ K^C_{ij}(x, p_\downarrow) & K^C_{ij}(x, p_\downarrow) & K^C_{ij}(x, p_\downarrow) & ... & K^C_{ij}(x, p_\downarrow) \end{bmatrix}\right\} b_i \text{ copies}}_{c_j \text{ copies}}$$

Using Bra-Ket notation, let us define

$$K^{\tau ks}_{\sigma\rho}(x) = \sum_{i_\tau=1}^{d_\tau} \sum_{j_\sigma=1}^{d_\sigma} \sum_{i_\rho=1}^{d_\rho} |\sigma j_\sigma\rangle \underbrace{\langle s, \sigma j_\sigma | \tau i_\tau, \rho i_\rho\rangle}_{\text{Clebsch-Gordon}} \underbrace{Y^{i_\rho}_{\rho,k}(x)}_{\text{harmonics}} \langle \rho i_\rho|$$

Then, using the main result of Lang & Weiler (2020a), the matrices $K^{\tau ks}_{\sigma\rho}(x)$ form a basis for the space of G-steerable kernels with input representation ρ and output representation σ. Then, the kernels K^A and K^B can be written in the form

$$K^A_{\sigma\rho}(x) = \sum_{\tau \in \hat{G}} \sum_{k=1}^{m_\rho} \sum_{s=1}^{m_\sigma} c^A_{\tau ks}(p_\downarrow) K^{\tau ks}_{\sigma\rho}(x)$$

$$K^A_{\sigma\rho}(x) = \sum_{\tau \in \hat{G}} \sum_{k=1}^{m_\rho} \sum_{s=1}^{m_\sigma} c^B_{\tau ks}(p_\downarrow) K^{\tau ks}_{\sigma\rho}(x)$$

where $c^A_{\tau ks}(p_\downarrow) : \mathcal{B} \to \text{Hom}_G[(\sigma, V_\sigma), (\sigma, V_\sigma)]$ and $c^B_{\tau ks}(p_\downarrow) : \mathcal{B} \to \text{Hom}_G[(\sigma, V_\sigma), (\sigma, V_\sigma)]$ are maps from the quotient space $\mathcal{B}$ into (σ, V_σ)-endomorphisms.

Thus, when $p_\downarrow \in \mathcal{B}$ is fixed, the total number of free parameters in the $A(x, p_\downarrow)$ matrix is

$$\dim A(x, p_\downarrow) = \sum_{\rho\sigma \in \hat{G}} n_\rho n_\sigma \sum_{\tau \in \hat{G}} m_\sigma m_\rho C^\sigma_{\tau\rho} \times \dim \text{Hom}_G[(\sigma, V_\sigma), (\sigma, V_\sigma)] \leq 4 \sum_{\rho\tau\sigma \in \hat{G}} n_\rho n_\sigma C^\sigma_{\tau\rho} m_\sigma m_\rho$$

Similarly, at fixed $p_\downarrow \in \mathcal{B}$, the total number of free parameters in $B(x, p_\downarrow)$

$$\dim B(x,p_\downarrow) = \sum_{\rho\sigma\in\hat{G}} n_\rho q_\sigma \sum_{\tau\in\hat{G}} C^\sigma_{\tau\rho} m_\sigma m_\rho \times \dim \mathrm{Hom}_G[(\sigma,V_\sigma),(\sigma,V_\sigma)] \leq 4 \sum_{\rho\tau\sigma\in\hat{G}} n_\rho q_\sigma C^\sigma_{\tau\rho} m_\sigma m_\rho$$

Note that for fixed $p_\downarrow$,

$$\dim A(x,p_\downarrow) \leq d_\mathcal{S}^2, \quad \dim B(x,p_\downarrow) \leq d_\mathcal{S} d_\mathcal{A},$$

always hold. To summarize, symmetry constraints force the LQR matrices A and B to take the form

$$A : \mathcal{B} \to \mathbb{R}^{d_S \times d_S}, \quad B : \mathcal{B} \to \mathbb{R}^{d_S \times d_A}$$

Furthermore, the output matrices take the form of a G-steerable kernel Lang & Weiler (2020a) and are parameterized by only a small number of parameters. This should be contrasted with the non-equivarient case where

$$A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_S \times d_S}, \quad B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_S \times d_A}$$

The dimension of the space $\mathcal{B}$ can be significantly less than that of $\mathcal{S} \times \mathcal{A}$. Symmetry constraints thus highly restricts the allowed form of LQR.

D.5 Theorem 4: Symmetry in Solutions of LQR

Theorem 4: *The LQR feedback matrix in $a_t^\star = -K(p)s_t$ and value matrix in $V = s_t^\top P(p) s_t$ are G-steerable kernels, or matrix-valued functions: $K : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{A} \times d_\mathcal{S}}$ and $P : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}$.*

Proof. The solution of LQR is derived from Bellman equation. The results, or optimal value function $V_t(\boldsymbol{s}) = \boldsymbol{s}^\top P_t \boldsymbol{s}$ and optimal policy function (feedback control law) $\boldsymbol{a}^\star = -K_t \boldsymbol{s}$, are given by the discrete algebraic Riccati equation (DARE).

$$V_t(\boldsymbol{s}) = \boldsymbol{s}^\top P_t \boldsymbol{s}, \quad \boldsymbol{a}^\star = -K_t \boldsymbol{s}, \tag{36}$$

where

$$P_t = Q + A^\top P_{t+1} A - A^\top P_{t+1} B \left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A \tag{37}$$

$$K_t = \left(R + B^\top P_{t+1} B\right)^{-1} B^\top P_{t+1} A. \tag{38}$$

The goal is to prove that P_t and K_t are G-steerable kernels.

If we iteratively linearize the problem, similar to the linearized dynamics, we quadratize the cost function around a point p. The quadratic cost also depends on the point p and is given as follows.

$$\boldsymbol{s}_{t+1} = A(p) \cdot \boldsymbol{s}_t + B(p) \cdot \boldsymbol{a}_t, \quad A : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}, \quad B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{A}} \tag{39}$$

$$c(s,a) = \boldsymbol{s}^\top Q(p) \boldsymbol{s} + \boldsymbol{a}^\top R(p) \boldsymbol{a}, \quad Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{S} \times d_\mathcal{S}}, \quad R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\mathcal{A} \times d_\mathcal{A}} \tag{40}$$

Analogously, Q and R are also G-steerable kernels by assuming the scalar function $c(s,a)$ is G-invariant:

$$\forall g \in G, \quad Q(g \cdot p) = \rho_\mathcal{S}(g) Q(p) \rho_\mathcal{S}(g^{-1}), \quad R(g \cdot p) = \rho_\mathcal{A}(g) R(p) \rho_\mathcal{A}(g^{-1}) \tag{41}$$

We prove by induction. For simplicity, we denote $A_p := A(p)$ and similarly for B_p, Q_p, R_p. We start from $P_T = Q_p$. By the property of G-steerable kernel, it is $Q_p = \rho_\mathcal{S}(g) Q_{g \cdot p} \rho_\mathcal{S}(g^{-1})$, thus P_T is a steerable kernel.

By induction, we assume P_{t+1} is steerable kernel: $P_{t+1}(p) = \rho_\mathcal{S}(g) P_{t+1}(g \cdot p) \rho_\mathcal{S}(g^{-1})$. We show how each component is transformed under group transformation.

$$P_t(p) = \underbrace{Q_p}_{(1)} + \underbrace{A_p^\top P_{t+1}(p) A_p}_{(2)} - \underbrace{A_p^\top P_{t+1}(p) B_p}_{(3)} \underbrace{\left(R_p + B_p^\top P_{t+1}(p) B_p\right)^{-1}}_{(4)} \underbrace{B_p^\top P_{t+1}(p) A_p}_{(5)} \tag{42}$$

For (1), by the property of G-steerable kernel, it is $Q_p = \rho_\mathcal{S}(g) Q_{g \cdot p} \rho_\mathcal{S}(g^{-1})$.

Table 3: Equivariant dimension reduction for linearized dynamics. This table highlights the reduced dimensions of spaces of kernels. $\mathcal{X}(\mathcal{M})$ denotes the dimension of signals living on the manifold $\mathcal{M}$.

Task	$\mathcal{S}$	$\mathcal{A}$	G	$\rho_{\mathcal{S}}$	$\rho_{\mathcal{A}}$	$\mathcal{X}(\mathcal{S}\times\mathcal{A})$	$\mathcal{X}(\mathcal{B})$
Free Particle in 2D	$\mathbb{R}^2\times\mathbb{R}^2$	$\mathbb{R}^2$	$\mathrm{SO}(2)$	ρ_{std}	ρ_{std}	$\mathbb{R}^{16}$	$\mathbb{R}^+\times\mathbb{R}^{14}$
Reacher (in 2D)	$S^1\times S^1\times\mathbb{R}^2$	$\mathbb{R}^2$	$\mathrm{SO}(2)$	$\rho_{std}\oplus\rho_{triv}$	ρ_{triv}	$S^1\times S^1\times\mathbb{R}^{10}$	$S^1\times\mathbb{R}^{10}$
Single Free Particle in 3D	$\mathbb{R}^3\times\mathbb{R}^3$	$\mathbb{R}^3$	$\mathrm{SO}(3)$	ρ_{std}	ρ_{std}	$\mathbb{R}^{32}$	$(\mathbb{R}^+)^2\times\mathbb{R}^{10}$
N-Free Particles in 3D	$\mathbb{R}^{3N}\times\mathbb{R}^{3N}$	$\mathbb{R}^{3N}$	$\mathrm{SO}(3)$	ρ_{std}	ρ_{std}	$\mathbb{R}^{32N}$	$(\mathbb{R}^+)^2\times\mathbb{R}^{10N}$

For (2), we have

$$A_p^\top P_{t+1}(p)A_p = \left(\rho_{\mathcal{S}}(g)A_{g\cdot p}\rho_{\mathcal{S}}\left(g^{-1}\right)\right)^\top\left(\rho_{\mathcal{S}}(g)P_t(g\cdot p)\,\rho_{\mathcal{S}}(g^{-1})\right)\left(\rho_{\mathcal{S}}(g)A_{g\cdot p}\rho_{\mathcal{S}}\left(g^{-1}\right)\right) \tag{43}$$

$$= \rho_{\mathcal{S}}(g)A_{g\cdot p}^\top\rho_{\mathcal{S}}\left(g^{-1}\right)\rho_{\mathcal{S}}(g)P_t(g\cdot p\rho_{\mathcal{S}}(g^{-1}))\rho_{\mathcal{S}}(g)A_{g\cdot p}\rho_{\mathcal{S}}\left(g^{-1}\right) \tag{44}$$

$$= \rho_{\mathcal{S}}(g)\left(A_{g\cdot p}^\top P_t(g\cdot p)A_{g\cdot p}\right)\rho_{\mathcal{S}}\left(g^{-1}\right). \tag{45}$$

We can similarly show for the rest:

$$(3) = \rho_{\mathcal{S}}(g)\left(A_{g\cdot p}^\top P_{t+1}(g\cdot p)B_{g\cdot p}\right)\rho_{\mathcal{A}}(g^{-1}) \tag{46}$$

$$(4)^{-1} = \rho_{\mathcal{A}}(g)\left(R_{g\cdot p}+B_{g\cdot p}^\top P_{t+1}(g\cdot p)B_{g\cdot p}\right)\rho_{\mathcal{A}}(g^{-1}) \tag{47}$$

$$(5) = \rho_{\mathcal{A}}(g)\left(B_{g\cdot p}^\top P_{t+1}(g\cdot p)A_{g\cdot p}\right)\rho_{\mathcal{S}}(g^{-1}) \tag{48}$$

Thus, the multiplication (3)(4)(5) transforms under input representation $\rho_{\mathcal{S}}(g)$ and output representation $\rho_{\mathcal{S}}(g^{-1})$. Analogously, all terms (1), (2), and (3)(4)(5) transforms under input representation $\rho_{\mathcal{S}}(g)$ and output representation $\rho_{\mathcal{S}}(g^{-1})$, same for the sum. By moving the terms, we obtain $P_t(p)=\rho_{\mathcal{S}}(g)P_t(g\cdot p)\rho_{\mathcal{S}}(g^{-1})$ that P_t is a steerable kernel.

D.6 Illustration and Examples

We work out a few examples of how symmetry can reduce the dimensionality of the LQR problem. We specifically consider two simple problems: a free particle moving in the plane and a free particle moving in space. The agent can control an applied force and tries to steer the particle to the origin.

Examples. In Table 3, we show the dimensions of the spaces of G-steerable kernels for each task. $\mathcal{X}(\mathcal{S}\times\mathcal{A})$ refers to the space *without* equivariant constraints, while $\mathcal{X}(\mathcal{B})$ denotes the space with *G-steerable equivariant constraints*. We can see that the spaces of equivariant version are hugely reduced because of (1) smaller base space, as visualized in Figure **??**, and (2) constrained output matrix $\mathbb{R}^{d\times d}$ with less free parameters.

In the empirical results, we show for multiple free particles (N-ball `PointMass` 3D), which generalize the results of single balls. Compared to `Reacher` that has two connected links by a joint, multiple free particles can have much better saving. Note that for `Reacher`, in implementation we convert the angles to unit vectors:

$$\left(\theta_1,\theta_2,\dot\theta_1,\dot\theta_2,x_g-x_f,y_g-y_f\right)\Rightarrow\left(\cos\theta_1,\sin\theta_1,\cos\theta_2,\sin\theta_2,\dot\theta_1,\dot\theta_2,x_g-x_f,y_g-y_f\right). \tag{49}$$

D.6.1 Single Particle Control in the Plane

We work out the single particle control problem in the plane. The goal is to move a particle to origin. The state space is the particle position, represented as a two-vector $\vec{p}$, and the particle velocity represented as a two-vector $\vec{v}$. The action space consists of the applied force $\hat{F}$, which is also a two-vector. The state space and action space are thus given by

$$\mathcal{S}=\mathbb{R}^2\times\mathbb{R}^2,\qquad \mathcal{A}=\mathbb{R}^2$$

The group $SO(2)$ acts on the state and action space. Specifically, under a rotation $R \in SO(2)$, the state and action transform as

$$\text{State Transform: } (\vec{p}, \vec{v}) \to (R\vec{p}, R\vec{v})$$
$$\text{Action Transform: } \vec{F} \to R\vec{F}$$

The state space transforms in $\rho_{\mathcal{S}} = \rho_1 \oplus \rho_1$ and the action space transform in $\rho_{\mathcal{A}} = \rho_1$. The set of orbits are then given by states and actions where the angles between the vectors $\vec{p}$, $\vec{v}$ and $\vec{F}$ are fixed.

The base space is given by $\mathcal{B} = \mathbb{R}^+ \times \mathbb{R}^4$. Using Proposition E.6. in Lang & Weiler (2020a), a basis for the steerable kernels of input type (ρ_1, V_1) and output type (ρ_1, V_1) is given by

$$K_{11}(x) = c_1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + c_2 \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} + c_3 \begin{bmatrix} \cos(2x) & \sin(2x) \\ \sin(2x) & -\cos(2x) \end{bmatrix} + c_4 \begin{bmatrix} -\sin(2x) & \cos(2x) \\ \cos(2x) & -\sin(2x) \end{bmatrix}$$

where each $c_j \in \mathbb{R}$. This result was first derived in Weiler et al. (2018a). In more compact notation, the matrix $K_{11} : S^1 \to \mathbb{R}^{2\times 2}$ can be expanded as

$$K_{11}(x) = \begin{bmatrix} c_1 + c_3\cos(2x) - c_4\sin(2x), & -c_1 + c_3\sin(2x) - c_4\cos(2x) \\ c_2 + c_3\sin(2x) + c_4\cos(2x), & c_1 + c_3\sin(2x) - c_4\sin(2x) \end{bmatrix}$$

where each $c_i \in \mathbb{R}$. Then, using the form of the LQR matrices, we can write

$$A(p_\downarrow, x) = \begin{bmatrix} K_A^{(1,1)}(p_\downarrow, x), & K_A^{(1,2)}(p_\downarrow, x) \\ K_A^{(2,1)}(p_\downarrow, x), & K_A^{(2,2)}(p_\downarrow, x) \end{bmatrix}, \quad B(p_\downarrow, x) = \begin{bmatrix} K_B^{(1,1)}(p_\downarrow, x) \\ K_B^{(2,1)}(p_\downarrow, x) \end{bmatrix}$$

where each $K_A^{(i,j)}$ and $K_B^{(k,l)}$ take the form, $X \in \{A, B\}$,

$$K_X^{(i,j)}(p_\downarrow, x) = \begin{bmatrix} c_{1,X}^{(i,j)}(p_\downarrow) + c_{3,X}^{(i,j)}(p_\downarrow)\cos(2x) - c_{4,X}^{(i,j)}(p_\downarrow)\sin(2x), & -c_{1,X}^{(i,j)}(p_\downarrow) + c_{3,X}^{(i,j)}(p_\downarrow)\sin(2x) - c_{4,X}^{(i,j)}(p_\downarrow)\cos(2x) \\ c_{2,X}^{(i,j)}(p_\downarrow) + c_{3,X}^{(i,j)}(p_\downarrow)\sin(2x) + c_{4,X}^{(i,j)}(p_\downarrow)\cos(2x), & c_{1,X}^{(i,j)}(p_\downarrow) + c_{3,X}^{(i,j)}(p_\downarrow)\sin(2x) - c_{4,X}^{(i,j)}(p_\downarrow)\sin(2x) \end{bmatrix}$$

the coefficients $c_{kA}^{(i,j)} : \mathbb{R}^+ \times \mathbb{R}^4 \to \mathbb{R}$ and $c_{kB}^{(i,j)} : \mathbb{R}^+ \times \mathbb{R}^4 \to \mathbb{R}$ are not constrained by symmetry and can be parameterized by a neural network. If we amalgamate each of the coefficients $c_{kA}^{(i,j)}$ and $c_{kB}^{(i,j)}$ into a single vector output C, the system dynamics can be learned by specifying

$$C : \mathbb{R}^+ \times \mathbb{R}^4 \to \mathbb{R}^{16+8}$$

This should be contrasted with the non-equivarient case, where one needs to learn

$$A : \mathbb{R}^6 \to \mathbb{R}^{16} \text{ and } B : \mathbb{R}^6 \to \mathbb{R}^8$$

The dimensional reduction in the equivarient vs non-equivarient case is infinite. This is analogous to Wang et al. (2020) where equivarient methods can outperform non-equivarient methods by essentially an essentially infinite margin. In practice, due to discritization, this infinite gain is reduced to some large finite number.

D.6.2 Single Particle Control in Space

Let us consider the analogous single particle control problem in three-dimensional space. The state space is the particle position, represented as a three-vector $\vec{p}$, and the particle velocity represented as a three-vector $\vec{v}$. The action space consists of the applied force $\hat{F}$, which is also a three-vector. The state space and action space are thus given by

$$\mathcal{S} = \mathbb{R}^3 \times \mathbb{R}^3, \qquad \mathcal{A} = \mathbb{R}^3$$

The group $SO(3)$ acts on the state and action space. Specifically, under a rotation $R \in SO(3)$, the state and action transform as

$$\text{State Transform: } (\vec{p}, \vec{v}) \to (R\vec{p}, R\vec{v})$$
$$\text{Action Transform: } \vec{F} \to R\vec{F}$$

Position and Velocity are both vectors the state space transforms in $\rho_{\mathcal{S}} = \rho_1 \oplus \rho_1$. Force is a vector quantity and the action space transform in $\rho_{\mathcal{A}} = \rho_1$. The set of $SO(3)$-orbits are then given by states

and actions where the angles between the vectors $\vec{p}$, $\vec{v}$ and $\vec{F}$ are fixed. The base space is given by $\mathcal{B} = \mathbb{R}^+ \times \mathbb{R}^6$.

A complete characterization of $SO(3)$-kernels was first derived in Weiler et al. (2018b). In the basis that diogonalizes the representation, the vectored kernel matrix for a $SO(3)$-steerable kernel of input type (D^1, W^1) and (D^1, W^1) can be written as

$$\text{Vec}(K_{11}(x)) = \begin{bmatrix} \Phi_0(||x||)Y_0(\frac{x}{||x||}) \\ \Phi_1(||x||)Y_1(\frac{x}{||x||}) \\ \Phi_2(||x||)Y_2(\frac{x}{||x||}) \end{bmatrix}$$

where each $Y_\ell : S^2 \to \mathbb{R}^{(2\ell+1)}$ are spherical harmonics in vector form. Each $\Phi_i(||x||) : \mathbb{R}^+ \to \mathbb{R}$ are a set of radial functions. Unvectorizing, we have that

$$K_{11}(x) = \begin{bmatrix} \Phi_0(||x||)Y_0^0(\frac{x}{||x||}) & \Phi_1(||x||)Y_1^1(\frac{x}{||x||}) & \Phi_2(||x||)Y_2^2(\frac{x}{||x||}) \\ \Phi_1(||x||)Y_1^{-1}(\frac{x}{||x||}) & \Phi_1(||x||)Y_1^0(\frac{x}{||x||}) & \Phi_2(||x||)Y_2^1(\frac{x}{||x||}) \\ \Phi_1(||x||)Y_2^{-2}(\frac{x}{||x||}) & \Phi_2(||x||)Y_2^{-1}(\frac{x}{||x||}) & \Phi_2(||x||)Y_2^0(\frac{x}{||x||}) \end{bmatrix}$$

Now, using the results of G-steerable kernel constraints, the most general LQR matrices can be written in the form

$$A(p_\downarrow, x) = \begin{bmatrix} K_A^{(1,1)}(p_\downarrow, x), & K_A^{(1,2)}(p_\downarrow, x) \\ K_A^{(2,1)}(p_\downarrow, x), & K_A^{(2,2)}(p_\downarrow, x) \end{bmatrix}, \quad B(p_\downarrow, x) = \begin{bmatrix} K_B^{(1,1)}(p_\downarrow, x) \\ K_B^{(2,1)}(p_\downarrow, x) \end{bmatrix}$$

where each $K_A^{(i,j)}$ and $K_B^{(k,l)}$ take the form, $X \in \{A, B\}$,

$$K_X^{ij}(x) = \begin{bmatrix} \Phi_0^{ij}(||x||)Y_0^0(\frac{x}{||x||}) & \Phi_1^{ij}(||x||)Y_1^1(\frac{x}{||x||}) & \Phi_2^{ij}(||x||)Y_2^0(\frac{x}{||x||}) \\ \Phi_1^{ij}(||x||)Y_1^{-1}(\frac{x}{||x||}) & \Phi_2^{ij}(||x||)Y_{-2}^{-2}(\frac{x}{||x||}) & \Phi_2^{ij}(||x||)Y_2^1(\frac{x}{||x||}) \\ \Phi_1^{ij}(||x||)Y_3^1(\frac{x}{||x||}) & \Phi_2^{ij}(||x||)Y_2^{-1}(\frac{x}{||x||}) & \Phi_2^{ij}(||x||)Y_2^2(\frac{x}{||x||}) \end{bmatrix}$$

the coefficients $\Phi_{kA}^{(i,j)} : \mathbb{R}^+ \times \mathbb{R}^6 \to \mathbb{R}$ and $\Phi_{kB}^{(i,j)} : \mathbb{R}^+ \times \mathbb{R}^6 \to \mathbb{R}$ are not constrained by symmetry and can be parameterized by a neural network. If we amalgamate each of the coefficients $\Phi_{kB}^{(i,j)}$ and $\Phi_{kB}^{(i,j)}$ into a single vector output Φ, the system dynamics can be learned by specifying

$$\Phi : \mathbb{R}^+ \times \mathbb{R}^6 \to \mathbb{R}^{12+6}$$

This should be contrasted with the non-equivarient case where one needs to learn a function from $\mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}^{12+6}$. By utilizing symmetry, we are able to significantly reduce the domain of the function spaces.

E Algorithm Design

We elaborate on the algorithm design in this section.

Invariance of `return`. We compute the expected return of sampled trajectories, and study how it is transformed:

$$\texttt{return}(\tau) = \mathbb{E}_\tau \left[\gamma^H Q_\theta(\boldsymbol{s}_H, \boldsymbol{a}_H) + \sum_{t=0}^{H-1} \gamma^t R_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) \right] = \mathbb{E}_\tau \left[U(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H-1}) \right] \tag{50}$$

$$\texttt{return}(g \cdot \tau) = \mathbb{E}_{g\cdot\tau} \left[\gamma^H \rho_0(g) \cdot Q_\theta(g \cdot \boldsymbol{s}_H, g \cdot \boldsymbol{a}_H) + \sum_{t=0}^{H-1} \gamma^t \rho_0(g) \cdot R_\theta(g \cdot \boldsymbol{s}_t, g \cdot \boldsymbol{a}_t) \right] \tag{51}$$

$$= \int_{g\in G} \rho_0(g) dg \cdot \mathbb{E}_\tau \left[U(g \cdot \boldsymbol{s}_{1:H}, g \cdot \boldsymbol{a}_{1:H-1}) \right] \tag{52}$$

$$= \mathbf{1} \cdot \mathbb{E}_\tau \left[U(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H-1}) \right] = \texttt{return}(\tau) \tag{53}$$

In Equation 51, we use $\rho_0(g) = \mathbf{1}$ to denote that the output is not transformed, so we may extract the term out. In Equation 52, dg is a Haar measure that absorbs the normalization factor, and we can extract the term from expectation. Equation 53 uses the invariance of Q_θ and R_θ. In other words, the return under the G-orbit of trajectories is the same, thus `return` is G-invariant.

Equivariance of G-`sample`. In Model Predictive Path Integral (MPPI) (Williams et al., 2017b), we sample N actions from a random Gaussian distribution $\mathcal{N}(\mu, \sigma^2 I)$, denoted as $\mathbb{A} = \{a_i\}_{i=1}^N$. However, since μ and σ are not state-dependent, CEM/MPPI does not satisfy the condition of equivariance, which requires that rotating the input results in a rotated output. To address this, we propose a solution - augmenting the action sampling by transforming with all elements in the group G: $G\mathbb{A} = \{g \cdot a_i \mid g \in G\}_{i=1}^N$. This approach ensures that our method can handle different orientations and maintain the property of equivariance.

To validate our approach, we first demonstrate the equivariance condition mathematically. We assume that (s_0, a_0) gives the maximum value $a_0 = \arg\max_{a \in G\mathbb{A}} Q(s_0, a)$. If we consider $g \cdot a_0 = g \cdot \arg\max_{a \in G\mathbb{A}} Q(s_0, a) = \arg\max_{a \in G\mathbb{A}} Q(g \cdot s_0, a)$, it implies that if we rotate the state to $g \cdot s_0$, we expect $g \cdot a_0$ to still provide the maximum Q-value so that $\arg\max$ can select it. The proof is validated using the invariance of Q, $Q(g \cdot s, g \cdot a) = Q(s, a)$. Hence, $a_0' = \arg\max_{a \in G\mathbb{A}} Q(g \cdot s_0, a) = \arg\max_{a \in G\mathbb{A}} Q(s_0, g^{-1} \cdot a)$. By comparing these two equations, we find that $a_0' = g \cdot a_0$.

Note that when not augmenting $\mathbb{A}$, it is not guaranteed that $g \cdot a_0$ exists in $\mathbb{A}$. However, when the number of samples approaches infinity, $g \cdot a_0$ can get close to some element in $\mathbb{A}$.

The proof can be directly applied to multiple steps, as `return` is also G-invariant.

F Implementation Details and Additional Evaluation

F.1 Implementation Details: Equivariant TD-MPC

We mostly follow the implementation of TD-MPC (Hansen et al., 2022). The training of TD-MPC is end-to-end, i.e., it produces trajectories with a learned dynamics and reward model and predicts the values and optimal actions for those states. It closely resembles MuZero (Schrittwieser et al., 2019) while uses MPPI (Model Predictive Path Integral (Williams et al., 2015; 2017b)) for continuous actions instead of MCTS (Monte-Carlo tree search) for discrete actions. It inherits the drawbacks from MuZero - the dynamics model is trained only from reward signals and may collapse or experience instability on sparse-reward tasks. This is also the case for the tasks we use: `PointMass` and `Reacher` and their variants, where the objectives are to reach a goal position.

F.2 Experimental Details

We implement G-equivariant MLP using `escnn` (Weiler & Cesa, 2021) for policy, value, transition, and reward network, with 2D and 3D discrete groups. For all MLPs, we use two layers with 512 hidden units. The hidden dimension is set to be 48 for non-equivariant version, and the equivariant version is to keep the same number of free parameters, or `sqrt` strategy.

For example, for D_8 group, `sqrt` strategy (to keep same free parameters) has number of hidden units divided by $\sqrt{|D_8|} = \sqrt{16} = 4$. The other strategy is to make equivariant networks' input and output be compatible with non-equivariant ones: *`linear`* strategy, which keeps same input/output dimensions (number of hidden units divided by $|D_8| = 16$).

The hidden space uses *regular* representation, which is common for discrete equivariant network (Cohen & Welling, 2016c; Weiler & Cesa, 2021; Zhao et al., 2022b).

F.3 Additional Results

Ablation on model-based vs. model-free ("planning-free"). We ablate the use of planning component in equivariant version of TD-MPC, which is to justify why we aim to build model-based version of equivariant RL algorithm over model-free counterparts. The results are shown in Figure 11. On both `Reacher` Easy and Hard, with planning, the performance is much better.

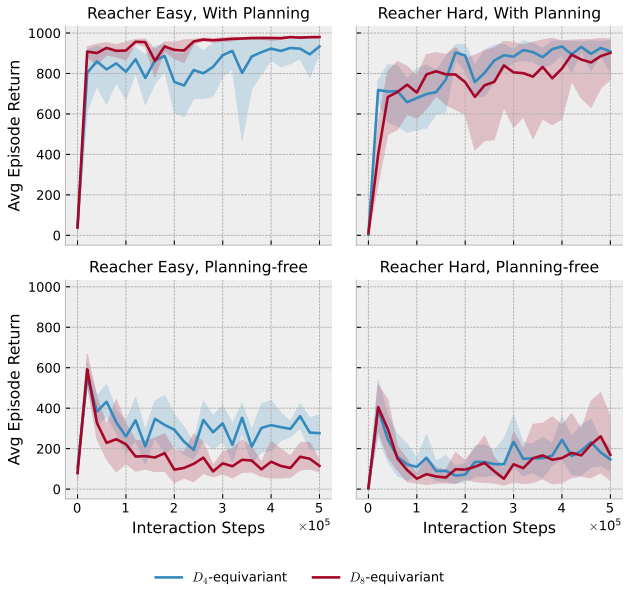

Figure 11: Ablation study on planning component.

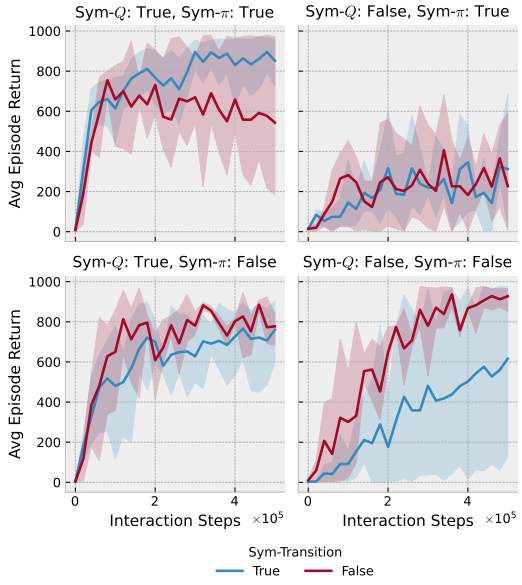

Figure 12: Ablation study on equivariant components, using `Reacher` Hard with D_8 symmetry group.

Ablation on equivariant components. Recall that we have several equivariant components in equivariant TD-MPC:

$$f_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S} : \quad \rho_\mathcal{S}(g) \cdot f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = f_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{54}$$

$$R_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad R_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{55}$$

$$Q_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{56}$$

$$\pi_\theta : \mathcal{S} \to \mathcal{A} : \quad \rho_\mathcal{A}(g) \cdot \pi_\theta(\cdot \mid \boldsymbol{s}_t) = \pi_\theta(\cdot \mid \rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t) \tag{57}$$

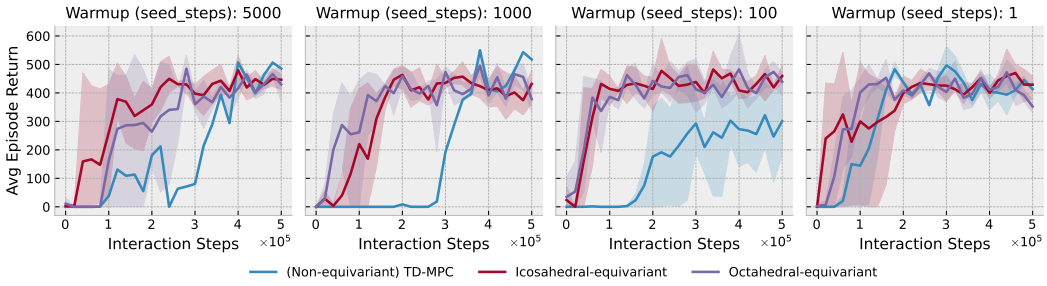

Figure 13: Ablation study on number of warmup episodes on `PointMass` *3D with small target*.

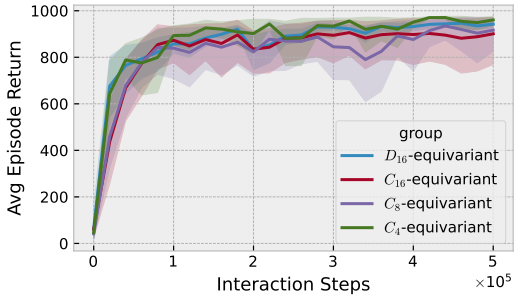

Figure 14: Ablation study on symmetry group on `Reacher` Hard.

We experiment to enable and disable each of them: (1) transition network: dynamics f and reward R, (2) value network: Q, and (3) policy network π. Note that to make equivariant and non-equivariant components compatible, we need to make sure the input and output dimensions match.

We show the results on `Reacher` Hard with D_8 symmetry group in Fig 12. Instead of using `sqrt` strategy (to keep same free parameters, number of hidden units divided by $\sqrt{|D_8|} = \sqrt{16} = 4$) on specifying the number of hidden units, we use `linear` strategy that keeps same input/output dimensions (number of hidden units divided by $|D_8| = 16$). Thus, the performance of fully non-equivariant model and fully equivariant model are not directly comparable, because the number of free parameters in fully equivariant one is much smaller.

The results show the relative importance of value, policy, and transition. It shows the most important equivariant component is Q-value network. It is reasonable because it has been used intensively in predicting into the future, where generalization and training efficiency are very important and benefit from equivariance.

Hyperparameter of amount of warmup. We experiment different number of warmup episodes, called `seed steps` in TD-MPC hyperparameter. We find this is a critical hyperparameter for (non-equivariant) TD-MPC. One possible reason is that TD-MPC highly relies on joint training and may collapse when the transition model is stuck at some local minima. This warmup hyperparameter controls how many episodes TD-MPC collects before starting actual training.

We test using different numbers on `PointMass` *3D with small target*. The results are shown in Figure 13, which demonstrate that our equivariant version is robust under all choices of warmup episodes, even with little to none warmup. The non-equivariant TD-MPC is very sensitive to the choice of warmup number.

Ablation on symmetry groups. We also do ablation study on the choice of discrete subgroups. We run experiments on `Reacher` Hard to compare 2D discrete rotation/dihedral groups: C_4, C_8, C_{16}, D_{16}, using 1 warmup episode.

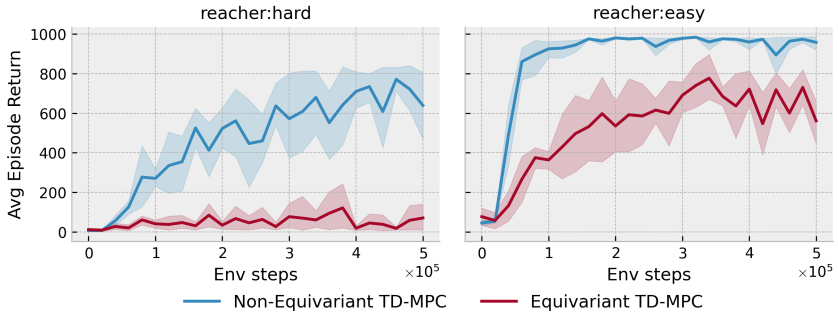

Figure 15: Results for global reference frame on `Reacher`.

The results are shown in Fig 14. We find using groups larger than C_8 does not bring additional improvement on this specific task, `Reacher` Hard. In the main paper, we thus use D_4, D_8 to balance the performance and computation time and memory use.

Comparing reference frames and state features. This experiment studies the balance between reference frames and the choice of state features. In the theory section, we emphasize that kinematic constraints introduce local reference frames.

Here, we study a specific example: `Reacher` (Easy and Hard). The second joint has angle θ_2 and angular velocity $\dot{\theta}_2$ relative to the first link.

For *local* reference frame version, we use

$$\Big(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, x_g - x_f, y_g - y_f\Big) \Rightarrow \Big(\cos\theta_1, \sin\theta_1, \cos\theta_2, \sin\theta_2, \dot{\theta}_1, \dot{\theta}_2, x_g - x_f, y_g - y_f\Big) \tag{58}$$

Thus, $\cos\theta_1, \sin\theta_1$ is transformed under standard representation ρ_1 and $\cos\theta_2, \sin\theta_2$ is transformed under trivial representation $\rho_0 \oplus \rho_0$.

For the *global* reference frame version, we compute the global location of the end-effector (tip) by adding the location of the first joint. Thus, the global position is transformed also under standard representation now ρ_1.

We show the results in Figure 15. Evaluation reward curves for non-equivariant and equivariant TD-MPC over 5 runs using global frames. Error bars denote 95% confidence intervals. Non-equivariant TD-MPC outperforms equivariant TD-MPC. Surprisingly, we find using global reference frame where the second joint is associated with standard representation (equivariant feature, instead of invariant feature) brings much worse results, compared to the local frame version in the main paper. One possibility is that it is more important to encode kinematic constraints (e.g., the length of the second link is preserved in $\cos\theta_2, \sin\theta_2$), compared to using equivariant feature.

G Additional Mathematical Background

We provide additional mathematical background on representation theory and its relation with our theoretical results.

G.1 Mathematical Exposition: Principal G-Bundles and LQR

Fiber bundles were introduced in the 1930s as a natural extension of the concept of tangent spaces in differential geometry (Milnor & Stasheff, 1974). Conceptually, fiber bundles attach additional information at each point on some underlying manifold. The connection between equivarient machine learning and the theory of fiber-bundles was first noted in (Cohen & Welling, 2016b; Weiler et al., 2018a). For continuous state-action manifolds, the dynamics can be viewed as some additional structure that depends on the state action manifold location. The results presented in the main text can be understood within the context of fiber bundle theory.

G.1.1 FIBER BUNDLE

We briefly comment on how Equivarient LQR can be understood in terms of fiber bundles. For a full exposition on the theory of fiber-bundles, please see (KARSTOFT, 1992; Husemöller, 2013).

Formally, a smooth fiber bundle is specified by (E, B, F, π) where E and B are smooth manifolds and F is a vector space. E is called the *total space* and B is called the *base space*. The vector space V is called the *fiber*. The map $\pi : E \to B$ is a smooth subjection called the projection map. The projection π must satisfy a trivialization condition (Husemöller, 2013).

Sections are generalization of vector fields. A smooth section s of a fiber bundle is a smooth map $s : B \to E$ such that

$$\forall x \in B, \quad \pi(s(x)) = x$$

so that a section does not change the base point. One can think of a section $s(x)$ as a vector with origin at point $x \in B$.

G.1.2 PRINCIPAL BUNDLE

Intuitively, a G-Principal Bundle is fiber bundle with an additional symmetry on the total space. Let G be a Lie group. Then, a smooth G-Principal Bundle is smooth fiber bundle that has continuous G action on the total space $G \times E \to E$ which preserves the fibers of E so that

$$\forall x \in E, \ \ \forall g \in G, \quad \pi(g \cdot x) = \pi(x)$$

Sections of G-Principal Bundles are defined analogously to the vector bundle case. Specifically, a section s is a map $s : B \to E$ such that

$$\forall x \in B, \quad \pi(s(x)) = x$$

Note that if $s : B \to E$ is a section then $g \cdot s : B \to E$ is another section as

$$\forall x \in B, \ \ \forall g \in G \quad \pi(g \cdot s(x)) = \pi(s(x)) = x$$

Thus, on G-Principal bundles we can speak of G-families of sections $s_g : G \times B \to E$ defined as $s_g(x) = g \cdot s(x)$. It is thus natural to consider equivalence classes of sections. These equivalence classes correspond to flows related by G action.

More formally, the LQR system in consideration is a G-bundle with total space given by $E = \mathcal{S} \times \mathcal{A}$ and base space $B = \mathcal{B}$ given by G-orbit representatives. The canonical projection $\pi : \mathcal{S} \times \mathcal{A} \to B$ is the map unto G-orbit representatives which satisfies

$$\forall g \in G, \quad \Pi(g \cdot x) = \Pi(x)$$

The LQR dynamical matrices A and B can be viewed as sections of the G-bundle. By utilizing equivarience methods, we only need to learn a section on the base space B as opposed to vector fields on the total space E. This corresponds to learning equivalence classes of sections, instead of individual sections.