Equivariant Action Sampling for Reinforcement Learning and Planning

Linfeng Zhao[1], Owen Howell[1], Xupeng Zhu[1], Jung Yeon Park[1], Zhewen Zhang[1], Robin Walters[†1], and Lawson L.S. Wong[†1]

Northeastern University, Boston, MA, USA
zhao.linf@northeastern.edu, robin.walters@northeastern.edu, lsw@ccs.neu.edu

Abstract. Reinforcement learning (RL) algorithms for continuous control tasks require accurate sampling-based action selection. Many tasks, such as robotic manipulation, contain inherent problem symmetries. However, correctly incorporating symmetry into sampling-based approaches remains a challenge. This work addresses the challenge of preserving symmetry in sampling-based planning and control, a key component for enhancing decision-making efficiency in RL. We introduce an action sampling approach that enforces the desired symmetry. We apply our proposed method to a coordinate regression problem and show that the symmetry aware sampling method drastically outperforms the naive sampling approach. We furthermore develop a general framework for sampling-based model-based planning with Model Predictive Path Integral (MPPI). We compare our MPPI approach with standard sampling methods on several continuous control tasks. Empirical demonstrations across multiple continuous control environments validate the effectiveness of our approach, showcasing the importance of symmetry preservation in sampling-based action selection.

Keywords: Symmetry · Continuous Control · Model-based Planning.

1 Introduction

In reinforcement learning (RL) for continuous control, the need for effective sampling-based action selection is paramount. Many control environments, especially in robotic manipulation and navigation, exhibit symmetries due to their operation within Euclidean space. While previous explorations of equivariance have primarily focused on deterministic RL policies [Ravindran and Barto, 2004, Zinkevich and Balch, 2001, van der Pol et al., 2020a, Mondal et al., 2020, Wang et al., 2021, Zhao et al., 2022a], the inherently multimodal nature of these control tasks demands sampling-based approaches. This crucial integration of symmetry into sampling-based methods remains largely under-explored.

In general, sampling methods will break the exact symmetries of action selection. This is an issue as the breaking of symmetry prevents the use of equivariant reinforcment learning methods [van der Pol et al., 2020b]. This paper addresses

the challenge of preserving symmetry in sampling-based planning and control, a vital aspect for enhancing decision-making efficiency in RL. Specifically, existing sampling methods maintain symmetry only in the limit of an infinite number of samples. In the case of finite samples, problem symmetries are only conserved approximately.

We formulate the action optimization challenge as a two-step procedure: first, we estimate the performance of state-action pairs or trajectories using a neural network like Q-values, energy functions, or returns; then, we engage in sampling to optimize optimal actions or action sequences. This paper investigates the equivariance properties of this formulation, crucial for the effectiveness of sampling-based strategies in symmetric environments.

In this study, we first study the coordinate regression problem, providing profound insights into the mechanisms of equivariant action sampling. We find that using a symmetry invariant energy function alongside our novel sampling strategy significantly enhances algorithm performance. Our strategy deviates from conventional sampling methods by augmenting sampled actions with the symmetry group G. This ensures that the sampling procedure always preserves equivariance, irrespective of the number of samples. Without our proposed sampling strategy, the symmetry of the two-step procedure holds only in the infinite sample limit. We extend our sampling strategy to multi-step action selection for sampling-based planning. We propose an equivariant version of Model Predictive Path Integral (MPPI) [Williams et al., 2017a] and derive that it needs equivariant dynamics and reward model, and equivariant policy and value network, analogous to the need of equivariant energy function. Based on it, we implement an equivariant model-based RL algorithm, TDMPC [Hansen et al., 2022], for continuous control tasks. This adaptation allows for a comprehensive preservation of symmetry across the entire trajectory planning process.

The contributions of this paper include offering both theoretical insights into equivariance in sampling-based planning and practical demonstrations of a novel equivariant sampling methodology's effectiveness. Empirical validations across various continuous control environments underscore the significance of our findings, illuminating the path for future research in symmetric RL tasks.

2 Related Work

Symmetry in Reinforcement Learning Symmetry in decision-making tasks has been studied in the context of reinforcement learning and control. Early research focused on symmetry in MDPs without function approximation [Ravindran and Barto, 2004, Zinkevich and Balch, 2001, Ravindran and Barto], while more recent work has explored symmetry in model-free (deep) RL and imitation learning using equivariant policy networks [van der Pol et al., 2020a, Mondal et al., 2020, Wang et al., 2021, Huang et al., 2024, Wang et al., 2022, Xie et al., 2020, Jia et al., 2024, Sortur et al., 2023, Zhao et al., 2023a]. Park et al. [2022] investigated equivariance in learning world models. Additionally, Zhao et al. [2022a] analyzed the use of symmetry in value-based planning on a 2D grid. Our work extends

these studies by focusing on continuous-action MDPs and sampling-based planning algorithms.

Geometric Graphs and Geometric Deep Learning 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 and have been used to examine algorithmic alignments and dynamic programming [Xu et al., 2019, Dudzik and Veličković, 2022]. We extend this concept by embedding MDPs into geometric spaces such as $\mathbb{R}^2$ or $\mathbb{R}^3$, focusing on 2D and 3D Euclidean symmetry [Brandstetter et al., 2021, Lang and Weiler, 2020, Weiler and Cesa, 2021] with corresponding symmetry groups E(2) and E(3). Geometric deep learning, which maintains geometric properties like symmetry and curvature in data analysis [Bronstein et al., 2021], has developed equivariant neural networks to preserve these symmetries. Notable contributions include G-CNNs [Cohen and Welling, 2016a], which introduced group convolutions, and Steerable CNNs [Cohen and Welling, 2016b], which generalize scalar feature fields to vector fields and induced representations. Additionally, $E(2)$-CNNs [Weiler and Cesa, 2021] solve kernel constraints for $E(2)$ and its subgroups by decomposing into irreducible representations. Researchers have also explored steerable message-passing GNNs [Brandstetter et al., 2022], $E(n)$-equivariant graph networks [Satorras et al., 2021], and the theory of equivariant maps and convolutions for scalar and vector fields [Kondor and Trivedi, 2018, Cohen et al., 2020]. Recent work has focused on designing latent equivariant architectures for 3D scene renderings [Klee et al., 2023, Howell et al., 2023].

Planning in Reinforcement Learning Planning in RL involves devising strategies to achieve long-term goals by predicting future states and rewards. MuZero [Schrittwieser, Antonoglou, Hubert, Simonyan, Sifre, Schmitt, Guez, Lockhart, Hassabis, Graepel, Lillicrap, and Silver, 2019] uses TDMPC [Hansen et al., 2022] integrates planning using Model Predictive Path Integral [Williams et al., 2017b] with temporal-difference learning [Sutton and Barto, 2018]. Planning has also played vital roles in robotics, where high-level task planning and geometric-level motion planning are crucial for enabling robots to perform complex tasks autonomously [Garrett et al., 2020, Kumar et al., 2024, Zhao and Wong, 2024]. These planning techniques are crucial for enabling agents and robots to perform complex tasks autonomously. However, they typically do not consider symmetry, which can lead to inefficiencies and suboptimal performance in symmetric environments.

3 MDPs with Geometric Structure

3.1 Formulation of Geometric MDPs

Markov Decision Processes (MDPs) are fundamental in modeling decision-making in interactive environments. In robotic control applications, MDPs often involve state spaces defined over Euclidean spaces $\mathbb{R}^d$ or on groups like SE(d). These

state spaces might directly represent physical spaces, such as the position of a robot, or embody latent structures in sensor inputs, such as camera images.

The study of isometric changes, which are transformations that preserve distances in the state space, introduces the Euclidean symmetry group $\mathrm{E}(d)$. This group and its subgroups, which can be expressed in semi-direct product form as $\left(\mathbb{R}^d, +\right) \rtimes G$, play a crucial role in how we understand and manipulate these state spaces. The group action, consisting of translations and rotations or reflections, transforms a vector x in the space according to $x \mapsto (tg) \cdot x := gx + t$ [Lang and Weiler, 2020, Weiler and Cesa, 2021]. These transformations are critical for defining symmetry properties in the system, which lead to more efficient problem-solving strategies [Wang et al., 2021, Zhao et al., 2022a, van der Pol et al., 2020b, Teng et al., 2023].

We define a class of MDPs with internal geometric structure, where the ground state space or a latent space of the MDP can be transformed by a Euclidean group. This extends a previously studied discrete case [Zhao et al., 2022a].

Definition 1 (Geometric MDP). *A Geometric MDP (GMDP) $\mathcal{M}$ is an MDP with internal geometric structure: there is a symmetry group $G \leq \mathrm{GL}(d)$ that acts on the ground or latent state space $\mathcal{S}$ and action space $\mathcal{A}$. It is written as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma, G, \rho_{\mathcal{S}}, \rho_{\mathcal{A}} \rangle$. The state and action spaces $\mathcal{S}, \mathcal{A}$ have group actions that transform them, defined by $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$.*

3.2 Symmetry in Geometric MDPs

In this subsection, we explore how Euclidean symmetry influences the internal geometric structure of MDPs. The symmetry properties in MDPs are characterized by the equivariance and invariance of the transition and reward functions, respectively [Ravindran and Barto, 2004, Zinkevich and Balch, 2001, van der Pol et al., 2020a, Wang et al., 2021, Zhao et al., 2022a, 2024]:

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

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

Here, g acts on the state and action spaces through group representations $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$, respectively. For instance, the standard representation $\rho_{\mathrm{std}}(g)$ of $\mathrm{SO}(2)$ assigns each rotation $g \in \mathrm{SO}(2)$ a 2D rotation matrix $R_{2\times 2}(g)$ given by:

$$R(g) = \begin{bmatrix} \cos(g) & \sin(g) \\ -\sin(g) & \cos(g) \end{bmatrix}$$

This matrix represents a rotation by an angle g. The trivial representation $\rho_{\mathrm{tri}}(g)$ assigns the one-dimensional identity matrix $\rho_{tri}(g) = \mathbf{1}_{1\times 1}$ to all g.

Properties. In a geometric MDP with a discrete symmetry group, the optimal policy mapping is G-equivariant, as demonstrated in [Ravindran and Barto, 2004]. To incorporate symmetry constraints, one strategy is to ensure the entire

Table 1: Examples of tasks modeled as geometric MDPs. G denotes the MDP symmetry group, $\mathcal{S}$ denotes the MDP state space, and $\mathcal{A}$ denotes the MDP action space. We can quantitatively measure the savings 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 provide intuition on savings. The $Gx = \{g \cdot x \mid g \in G\}$ 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., 2022a]
3	SO(2)	$\mathbb{R}^2$	$\mathbb{R}^2$	$\mathbb{R}^+$	S^1	2D Continuous Navigation [Zhao et al., 2024]
4	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	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	SO(2)	SO(2)	$\mathbb{R}^2$	$\{e\}$	S^1	Free Particle on SO(2) $\cong S^1$ manifold
7	SO(3)	SO(3)	$\mathbb{R}^3$	$\{e\}$	S^2	Free Particle on SO(3) [Teng et al., 2023]
8	SO(2)	SE(2)	SE(2)	$\mathbb{R}^2$	S^1	Top-down Grasping [Zhu et al., 2022]
9	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]

policy mapping is equivariant: $a_t = \texttt{policy}(s_t)$ [Wang et al., 2021, Zhao et al., 2022a, van der Pol et al., 2020b, Zhao et al., 2024], as illustrated in Figure 1.

Many model-based RL algorithms rely on iteratively applying Bellman operations [Sutton and Barto, 2018]. We show that the symmetry G in a Geometric MDP (GMDP) results in a G-equivariant Bellman operator. This implies that the iterative process in model-based RL algorithms can be constrained to be G-equivariant to exploit symmetry.

Additionally, for GMDPs, a specific instance of a dynamic programming (DP)-based algorithm, value iteration, can be connected with geometric graph neural networks [Bronstein et al., 2021]. For non-geometric graphs, Dudzik and Veličković [2022] demonstrated the equivalence between dynamic programming on a general non-geometric MDP and a message-passing GNN.

Theorem 1. *The Bellman operator of a geometric MDP is equivariant under the Euclidean group* $\mathrm{E}(d)$*, which includes d-dimensional isometric transformations.*

We provide proofs and derivations in Appendix D. This extends the theorems in [Zhao et al., 2022a] on 2D discrete groups, where they showed that value iteration is equivariant under discrete subgroups of the Euclidean group, such as discrete translations, rotations, and reflections. We generalize this result to groups of the form $\left(\mathbb{R}^d, +\right) \rtimes G$, where G includes continuous rotations and translations.[1]

3.3 Illustration and Examples of Geometric MDPs

Geometric MDP examples include moving a point robot in a 2D continuous space ($\mathbb{R}^2$, Example 3 in Table 1) or a discrete space ($\mathbb{Z}^2$, Example 1 [Tamar

[1] For the translation part, one may use relative/normalized positions or induced representations [Lang and Weiler, 2020, Cohen et al., 2020].

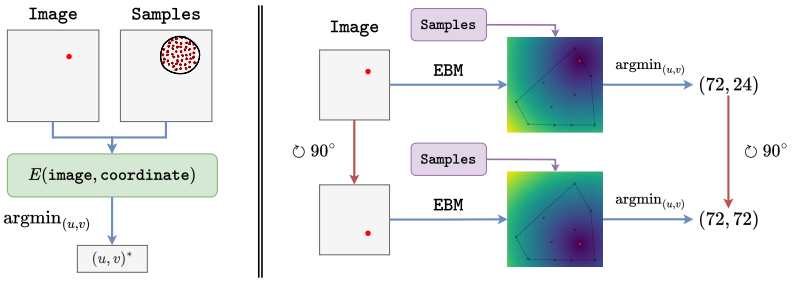

Fig. 1: Illustration of the coordinate regression problem (Sec 4.2) and its equivariance. (Left) The energy function `EBM` takes **image** and coordinate **samples** and outputs scalar energy value. (Right) Equivariance in coordinate regression: rotating the image and augmenting samples results in rotated coordinate prediction.

et al., 2016]), which is the abstraction of 2D discrete or continuous navigation. Table 1 includes more relevant examples. We use visual navigation over a 2D grid ($\mathbb{Z}^2 \rtimes C_4$, Example 2 [Zhao et al., 2022a, Lee et al., 2018]) 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 includes linear velocity $\mathbb{R}^3$. Alternatively, we can consider (5) moving a rigid body with SO(3) rotation. In (6) and (7), we consider moving free particle positions on SO(2), SO(3), which are examples of optimal control *on manifold* in [Teng et al., 2023, Lu 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 SE(2) action on grasping an object on a plane with 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, where the agent controls a two-joint arm. It is easy to see that because two links are connected, kinematic constraints restrict the potential possible improvement with an equivariant algorithm. Additionally, Example (3) is later implemented as `PointMass`, which is (8) top-down grasping without SO(2) rotation.

4 Equivariant Sampling-Based Action Selection

4.1 Action Selection via Sampling

Many real-world reinforcement learning (RL) and imitation learning problems involve continuous actions $\boldsymbol{a}_t \sim \pi(a \mid \boldsymbol{s}_t)$. When the action space $\mathcal{A}$ is infinite, the policy function may need to employ stochastic sampling.

Concretely, we focus on a two-step "implicit policy" strategy [Hansen et al., 2022, Kalashnikov et al., 2018, Florence et al., 2021], illustrated in Figure 1 (left), which solves the action optimization problem $a^* = \arg\min_a E(s, a)$ via sampling in two steps: (1) A neural network evaluates state-action pairs $(\boldsymbol{s}_t, \boldsymbol{a}_t)$ or trajectories $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}, \ldots)$; (2) Online optimization is performed over the

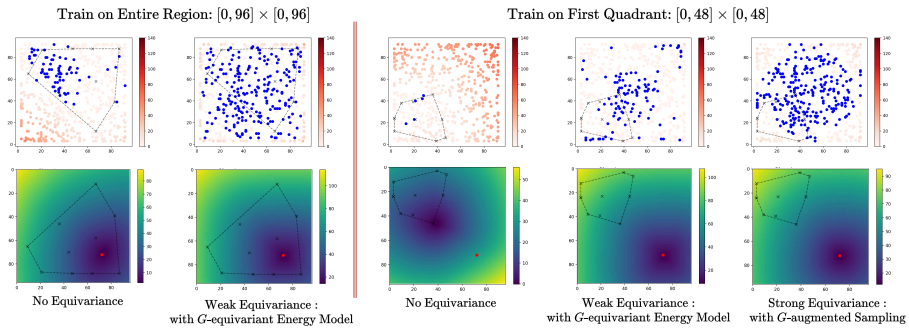

Fig. 2: Demonstration of results on coordinate regression problem: left two columns for training on entire region, and right three columns for training only on coordinates in first quadrant.

action space $\boldsymbol{a} \in \mathcal{A}$ to select better actions for each state $\boldsymbol{s}_t$ using methods such as the Cross-Entropy Method (CEM) [de Boer et al., 2005] or Model-Predictive Path Integral [Williams et al., 2017b]. In step (1), the neural network can be parameterized as an energy function $E_\theta(s, a)$ being minimized (Implicit Behavior Cloning, IBC; Florence et al. [2021]), a Q-value network $Q_\theta(s, a)$ being maximized (QT-Opt; Kalashnikov et al. [2018]), or the N-step return of a trajectory being maximized (TD-MPC; Hansen et al. [2022]).

4.2 Coordinate Regression Problem

In learning visuo-motor policies, a key challenge is converting high-dimensional image data into continuous action outputs. This challenge is exemplified in the coordinate regression problem [Florence et al., 2021], where the goal is to predict the xy coordinates of a specific target marker within an image, as shown in Figure 1 (left). We use this problem to demonstrate and study the equivariance properties of action sampling.

In the coordinate regression problem, the objective is to predict the (x, y) coordinate value v of a (green) marker on an image I: $v^* = \arg\max_{v \in \mathbb{R}^2} E_\theta(I, v)$, which can be written as a function $h(\cdot)$ of the image $v^* = h(I)$.

The coordinate regression problem exhibits both rotation and reflection symmetry (or $g \in D_4$ dihedral group). Specifically, if the input image is rotated/flipped $g \cdot I$, the network should predict the rotated/flipped coordinate value $g \cdot v$: $h_\theta(g \cdot I) = g \cdot h_\theta(I) \equiv g \cdot v$. Although simplified, the coordinate regression problem captures many fundamental challenges inherent in visuomotor control problems like robotic manipulation and control. Understanding the interplay between symmetry and sampling in this problem will help build intuition for equivariant action sampling in real-world tasks.

4.3 Strong and Weak Equivariance in Sampling

We can classify sampling methods that satisfy symmetry constraints as either **weak equivariance** or **strong equivariance**. Suppose that we are trying to

estimate the function $f(x)$ via sampling

$$f(x) = \mathbb{E}_\omega[q(x, \omega)],$$

where ω is drawn from some probability distribution and $q(x, \omega)$ averaged over the distribution of ω returns $f(x)$. Now, suppose that we know a-priori that the function f satisfies some equivariant constraints,

$$\forall g \in G, \quad f(g \cdot x) = \rho(g^{-1}) f(x).$$

This constraint arises naturally in many energy based models, where the energy function is invariant under some spatial symmetry, see Fig. 1. In the naive approach, we can drawn m sample points $\omega_1, \omega_2, ..., \omega_m$ i.i.d. and estimate the sample average $\hat{f}(x)$ of the function $f(x)$ as

$$\hat{f}(x) = \frac{1}{m} \sum_{i=1}^{m} q(x, \omega_i).$$

However, this approximation $\hat{f}(x)$ does not need to satisfy the original equivariance property. We will say that a sample estimator $\hat{f}$ of a G-equivariant function f is **weakly equivariant** if

$$\forall g \in G, \quad \mathbb{E}_\omega[q(g \cdot x, \omega)] = \rho(g^{-1}) \mathbb{E}_\omega[q(x, \omega)] \tag{3}$$

so that the G-equivariance properties of the estimator are recovered after averaging. Note that a weakly equivariant estimator in Eq. 3 will have sample averages that are not guaranteed to be G-equivariant. Analogously, we will say that a sample estimator $\hat{f}$ of a G-equivariant function f is **strongly equivariant** if

$$\forall g \in G, \quad q(g \cdot x, \omega) = \rho(g^{-1}) q(x, \omega) \tag{4}$$

holds identically. A strongly equivariant estimator in Eq. 3 has sample averages that always satisfies G-equivariant condition, regardless of the number of samples. Note that any strongly equivariant estimator is a weakly equivariant estimator, but the converse is not true.

We show in the appendix Sec. D.1 that for any compact group G it is possible to construct a strongly equivariant estimator $\hat{f}_G$ of f from a weakly equivariant estimator $\hat{f}$ of f. Furthermore, the strongly equivariant estimator $\hat{f}_G$ is guaranteed to be a better estimator of f than the weakly equivariant estimator $\hat{f}$.

4.4 Constructing Fully Equivariant Version of Action Sampling

Following the insight, we construct an equivariant action sampling approach based on Implicit Behavior Cloning method (IBC) [Florence et al., 2021], an energy-based approach that samples actions from a learned energy model. We use it to illustrate how to design a sampling-based policy that is not only weakly equivariant but also strongly equivariant in generating action samples.

We propose an equivariant action sampling strategy which has two steps can be used to produce action $\boldsymbol{a}_t$ for one time step. We must enforce equivariance in both the energy function $E(\boldsymbol{s}_t, \boldsymbol{a}_t)$, and (2) equivariance in sampling of actions $\boldsymbol{a}_t \sim \pi(\boldsymbol{a} \mid \boldsymbol{s}_t)$. More concretely, we need to make these two steps respect symmetry:

- *Step 1: G-Invariant Energy Function: Weak Equivariance.* We enforce the constraint $E(\boldsymbol{s}, \boldsymbol{a}) = E(g \cdot \boldsymbol{s}, g \cdot \boldsymbol{a})$ to maintain G-invariance. Although this condition guarantees equivariance under an infinite sampling regime, finite sample sets can still introduce deviations due to sampling disparities, particularly noticeable when the sample size is small.
- *Step 2: Symmetry Preservation in Sampling: Strong Equivariance.* As CEM samples actions from a Gaussian distribution, this randomness can disrupt the underlying symmetries. To counteract this, we need to additionally introduce a symmetry-preserving mechanism in the sampling process.

***G-Augmented Sampling**: Sampling on G-orbits.* We propose a strategy to preserve equivariance in sampling for *strong equivariance*[2]. 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 score (or "return" in RL terminology) is simply a scalar value $Q(s, a)$, also called a Q-value in RL literature Sutton and Barto [2018]. To develop analogy with energy models, we will sometime denote the return, as a function of state s and action a, as $E(s, a)$. Assuming we only select the best trajectory ($K = 1$), we require the sampling algorithm to be equivariant: $a_0 = \arg\min_a E(s_0, a)$. In other words, if we rotate the state $s_0 \to g \cdot s_0$, the selected action is also rotated $a_0 \to g \cdot a_0$.

We can enforce this condition via the following simple strategy: for each single sample $\boldsymbol{a}$, we augment the action sampling via $g \cdot \boldsymbol{a}$, i.e., left to the orbit of G: $\{g \cdot \boldsymbol{a} \mid g \in G\}$. Thus, the G-augmented sample set is $G\mathbb{A} = \{g \cdot a_i \mid g \in G\}_{i=1}^N$. We indicate the sampling strategy as G-`sample`. The equivariance condition for the reward function can be written as

$$g \cdot a_0 = g \cdot \arg\min_{a \in G\mathbb{A}} E(s_0, a) = \arg\min_{a \in G\mathbb{A}} E(g \cdot s_0, a). \tag{5}$$

For a more detailed account, please refer to Sec E.

4.5 Evaluation of Equivariant Sampling on Coordinate Regression

Setup. In our experiments, we processed images at a resolution of 96×96 pixels. The dataset consisted of 10 training images from either (1) the entire region $[0, 96] \times [0, 96]$ or (2) only the first quadrant $[0, 48] \times [0, 48]$. Each image features a single red marker with randomly assigned coordinates (as shown in Fig 1), using a fixed random seed to ensure consistency across models. The coordinates

[2] We can remove randomness while keeping the correct distribution by reparameterizing the input with standard Gaussian noise.

input to the energy function $E(I, v)$ are normalized to $[-1, 1] \times [-1, 1]$. For the equivariant $E(I, v)$, we use D_4-equivariance[3].

(1) The **upper row** of Figure 2 evaluates spatial generalization, presenting the model's test performance on 500 random coordinates. A blue marker indicates the model's accurate prediction within a 1-pixel error range. (2) The **lower row** shows the learned energy function's landscape across a 96×96 grid visualized as a color map, with a test marker fixed at coordinate $(72, 72)$. The color intensity corresponds to the energy levels, guiding the CEM in identifying potential coordinate predictions.

Additionally, we visualized the training data points as **crosses** $(\times)$ within the prediction error (upper row) and energy color map (lower row) and delineated the convex hull of these training points. The convex hull represents the smallest convex set that contains all the training data points and serves as a boundary for evaluating the model's extrapolation capabilities.

Evaluation: Coordinate Regression Spatial Generalization. (1) When training across the entire region, the use of equivariance in both energy and sampling resulted in more blue points with errors less than 1 pixel, indicating a well-trained energy function that supports accurate predictions during sampling. (2) When training was limited to the first quadrant, the non-equivariant model struggled to generalize beyond the convex hull, especially in regions outside $[0, 48] \times [0, 48]$. However, the implementation of a G-invariant energy function combined with G-augmented sampling significantly enhanced the model's extrapolation capabilities. This underscores the importance of equivariance in improving model generalization, particularly through the use of augmented sampling.

Analysis: Equivariance Error in 1-Step CEM. To explore the relationship between equivariance and finite samples, we simulate CEM using *untrained* equivariant and non-equivariant energy functions $E(s, a)$, thereby avoiding any learned equivariance from data. We measure the equivariance error under two conditions: (1) using a D_4-equivariant ($90°$ rotations and reflections) or non-equivariant energy-based model $E(s, a)$, and (2) employing a G-augmented equivariant action sampling strategy. We utilize randomly initialized models and random action samples with $\mu = 2$ (shared across four variants but resampled between runs) and average the results over 50 seeds, as shown in Fig 3. The results indicate that while the equivariant EBM without G-augmentation requires more samples to achieve a low equivariance error, the G-augmented sampling consistently

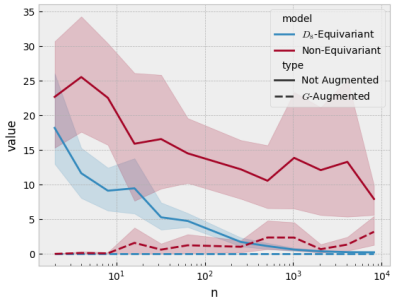

Fig. 3: Measuring the equivariance error of using whether G-invariant $E(s, a)$ and whether augment action with G.

[3] (1) The image I has dimensions $3 \times 96 \times 96$, with group actions involving *i)* spatial rotation of the image, and *ii)* no action on the RGB channels. (2) The coordinate input u to the energy function is essentially a $2 \times 1 \times 1$ vector (no spatial dimension), with group actions involving *i)* no spatial rotation, and *ii)* standard G-representation (2×2 rotation matrix).

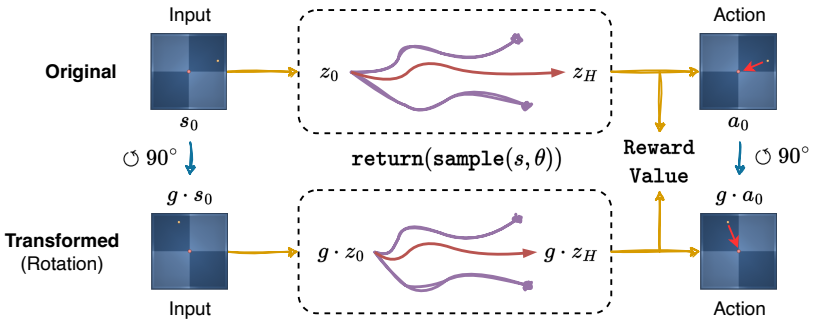

Fig. 4: The proposed sampling-based planning algorithm $a_0 = \texttt{plan}(s_0)$: if the input state is rotated, the output action should be rotated accordingly. This requires (1) the learned functions to be G-equivariant or G-invariant networks and (2) a specialized sampling strategy, as introduced in our method.

maintains perfect equivariance. This confirms the superiority of our proposed algorithm over sampling algorithms that lack G-action augmentation and those that do not employ a G-equivariant model.

5 Equivariant Sampling-based Planning Algorithm

In this section, we present an equivariant model-based RL algorithm designed for continuous action spaces, leveraging continuous symmetry through symmetric sampling. To plan in continuous spaces, we employ *sampling-based* methods such as MPPI [Williams et al., 2017a, 2015], extending them to maintain equivariance. Our approach builds on prior work [Zhao et al., 2022a] that utilized value-based planning in a discrete state space $\mathbb{Z}^2$ with the discrete group D_4, extending these concepts to continuous domains.

The core idea is to ensure that the algorithm $a_t = \texttt{plan}(s_t)$ produces actions that are consistent under 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 illustrated in Figure 1. This principle is applicable to MDPs with various symmetry groups.

5.1 Components

We use TD-MPC [Hansen et al., 2022] as the foundation of our implementation. Here, we introduce the procedure and demonstrating how to incorporate symmetry into sampling-based planning algorithms.

– *Planning with learned models.* We utilize the MPPI (Model Predictive Path Integral) control method [Williams et al., 2017a,b, 2015, 2016], as adopted in TD-MPC [Hansen et al., 2022]. We sample N trajectories with a horizon H using the learned dynamics model, with actions derived from a learned policy, and estimate the expected total return.

– *Training models.* The learnable components in equivariant TD-MPC include: an `encoder` that processes input observations, `dynamics` and `reward` networks that simulate the MDP, and `value` and `policy` networks that guide the planning process.
– *Loss.* The only requirement is that the loss function is G-invariant. The loss terms in TD-MPC include value-prediction MSE loss and dynamics/reward-consistency MSE loss, all of which satisfy invariance.

5.2 Integrating Symmetry

Zhao et al. [2022a] 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 approaches 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 and Barto, 2004, Zhao et al., 2022a, van der Pol et al., 2020b]) 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 $\bar{f}(\boldsymbol{z}, \boldsymbol{a}) = \boldsymbol{z}'$.
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 and 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 [Wang et al., 2021, Zhao et al., 2022a, van der Pol et al., 2020b].
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: $\rho_{\text{out}}(g) \cdot y = \rho_{\text{out}}(g) \cdot \texttt{MLP}(x) = \texttt{MLP}(\rho_{\text{in}}(g) \cdot x)$.

$$f_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S} : \quad \rho_\mathcal{S}(g) \cdot f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = f_\theta(\rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t, \rho_\mathcal{A}(g) \cdot \boldsymbol{a}_t) \tag{6}$$

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

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

$$\pi_\theta : \mathcal{S} \to \mathcal{A} : \quad \rho_\mathcal{A}(g) \cdot \pi_\theta(\cdot \mid \boldsymbol{s}_t) = \pi_\theta(\cdot \mid \rho_\mathcal{S}(g) \cdot \boldsymbol{s}_t) \tag{9}$$

5.3 Equivariance of `MPC`

Analogous to equivariant action selection for single-step case, we constrain the underlying MPC planner to be equivariant. We use MPPI (Model Predictive Path Integral) [Williams et al., 2017b, 2015], 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:

$$\texttt{return}(\tau) = \mathbb{E}_\tau \left[\gamma^H Q_\theta \left(\boldsymbol{s}_H, \boldsymbol{a}_H \right) + \sum_{t=0}^{H-1} \gamma^t R_\theta \left(\boldsymbol{s}_t, \boldsymbol{a}_t \right) \right] = \mathbb{E}_\tau \left[U(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H-1}) \right] \tag{10}$$

A trajectory is transformed element-wise by 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, similar to CEM. Thus, we can (1) constrain return to be G-invariant and (2) use G to augment the sampling of action sequence $(\boldsymbol{a}_t, \ldots, \boldsymbol{a}_{t+H})$.

Proposition 1. *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, the sampling procedure satisfies the following conditions, indicating that the procedure $\texttt{return}(G\text{-}\texttt{sample}(s, \theta))$ 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.

$$G\text{-}\texttt{sample} : \boldsymbol{s}_t, \theta \mapsto \tau_i : \quad g \cdot \tau_i \sim G\text{-}\texttt{sample}(g \cdot \boldsymbol{s}_t; f_\theta, R_\theta, Q_\theta, \pi_\theta) \tag{11}$$

$$\texttt{return} : \tau_i \mapsto \mathbb{R} : \quad \texttt{return}(\tau_i) = \texttt{return}(g \cdot \tau_i) \tag{12}$$

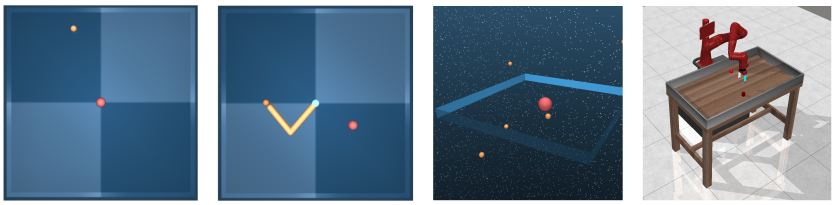

Fig. 5: Tasks used in experiments: (1) `PointMass` in 2D, (2) `Reacher`, (3) Customized 3D version of `PointMass` with multiple particles to control, and (4) MetaWorld task to reach an object with gripper.

6 Evaluation: Sampling-Based Planning

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

6.1 Experimental Setup

Tasks. We verify the algorithm on a few selected and customized tasks using the DeepMind Control suite (DMC) [Tassa et al., 2018], visualized in Figure 5. One task is a 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 simultaneously. 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 tasks 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 system has O(2) rotation and 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 includes (1) gripper position, (2) 3D position plus 4D quaternion of at most 2 relevant objects, and (3) 3D randomized goal position, depending on tasks. If we consider tasks with gravity, the MDP itself should exhibit SO(2) symmetry about the gravity axis. In implementation, the symmetry also depends on the data distribution, so we make the origin at the workspace center and the gripper initialized at the origin, so the task respects rotation equivariance around the origin. We add SO(2) equivariance to the algorithm about the gravity axis.

Experimental setup. We compare against the non-equivariant version of TD-MPC [Hansen et al., 2022]. By default, we 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.

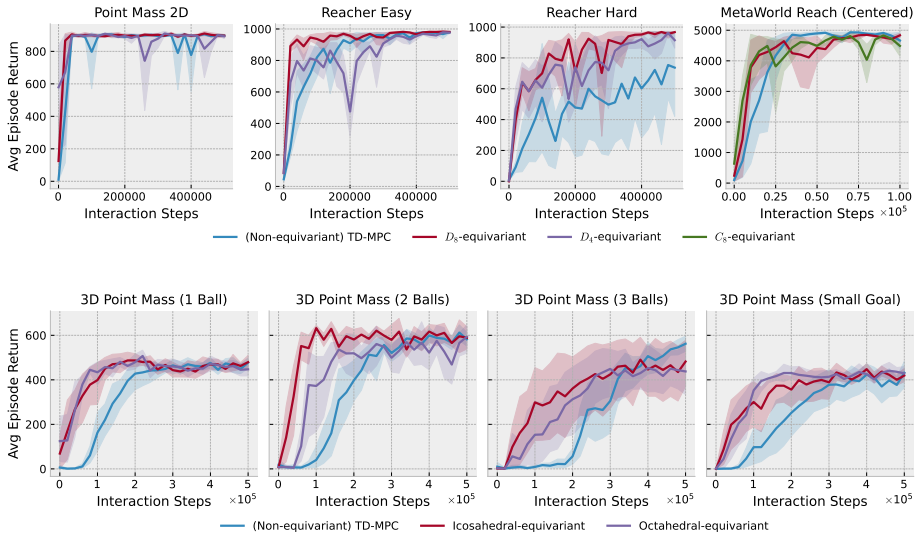

Fig. 6: (Upper) Results on `PointMass`, `Reacher`, and MetaWorld `Reach` task. (Lower) A set of customized 3D N-ball `PointMass` tasks, with $N = 1, 2, 3$, and a customized 3D `PointMass` with a smaller target.

The training procedure follows TD-MPC [Hansen et al., 2022]. We use the state as input and for equivariant TD-MPC, we divide the original 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 and Cesa, 2021], which are more stable and easier to implement than continuous equivariance. For the 2D case, we use 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 SO(3) with orders 60 and 24 respectively. On `Reacher` tasks, we also compare against a *planning-free* baseline by removing MPPI planning with the learned model and only keeping the policy learning, as shown in Fig 8.

6.2 Results

Figures 6 (upper and lower rows) present the reward curves, demonstrating that our equivariant methods can achieve near-optimal performance 2 to 3 times faster in terms of training interaction steps for several tasks. For the default 2D `PointMass` task, the D_8-equivariant version learns slightly faster than the non-equivariant version. In the `Reacher` task, as shown in the lower part of Figure 6, the D_8-equivariant version significantly outperforms the non-equivariant TD-MPC, especially in the `Hard` domain. The D_4-equivariant version also performs

better than the baseline, though not as well as D_8. The rightmost plot illustrates the MetaWorld `Reach` task, which involves reaching a button on a desk using a parallel gripper. We added SO(2) equivariance to the algorithm about the gravity axis and evaluated the C_8 and D_8-equivariant versions, both of which demonstrated more efficient learning.

In the `Reacher` tasks, we also compared against a *planning-free* baseline by removing MPPI planning with the learned model and retaining only policy learning, as shown in Figure 8. This approach is effectively similar to the DDPG algorithm [Lillicrap et al., 2016].

We designed a set of more challenging 3D versions of `PointMass` and used SO(3) subgroups to implement 3D equivariant versions of TD-MPC, utilizing icosahedral- and octahedral-equivariant MLPs. Figure 6 (lower) shows tasks with $N = 1, 2, 3$ balls in 3D `PointMass`, and the rightmost figure depicts a 1-ball 3D version with a smaller target (0.02 compared to 0.03 in the N-ball version).

We found that the icosahedral (order 60) equivariant TD-MPC consistently learns faster and uses fewer samples to achieve the best rewards compared to the non-equivariant version. The octahedral (order 24) equivariant version also performs similarly. Interestingly, the best absolute rewards in the 1-ball case are lower than in the 2- and 3-ball cases, which may be due to the higher possible return from having 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 hyper-parameter `seed_steps` that controls the number of warm-up trajectories. In contrast, our equivariant version is robust to it and sometimes learns better with less warm-up. In the shown curves, we do not use warm-up across non-equivariant and equivariant ones and present additional results in Appendix F. Most of these tasks are goal-reaching and have optimal rewards, thus the number of transitions used to reach near-optimal is a proxy of sample efficiency. The reward curves show the superiority of our equivariant sampling-based approach.

7 Conclusion and Discussion

This work introduces a two-step approach to preserve symmetry in sampling-based planning and control for continuous tasks. Using equivariant sampling, our method improves decision-making efficiency and performance in various control environments. Our findings highlight the benefits of integrating symmetry into sampling-based model-based RL algorithms, enhancing current practices and opening avenues for future research in continuous control and robotics applications.

Acknowledgements This work was partially supported by NSF Grant 2107256. Owen Howell is indebted to the National Science Foundation Graduate Research Fellowship Program (NSF-GRFP) for financial support. Robin Walters is supported by NSF DMS-2134178.

Bibliography

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

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*, pages 632–640. Morgan Kaufmann, 2001.

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.

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.

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

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

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.

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)*, pages 1714–1721, Singapore, May 2017a. IEEE. ISBN 978-1-5090-4633-1. https://doi.org/10/ggdv8n. URL https://ieeexplore.ieee.org/document/7989202/.

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.

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

Haojie Huang, Owen Howell, Dian Wang, Xupeng Zhu, Robin Walters, and Robert Platt. Fourier transporter: Bi-equivariant robotic manipulation in 3d, 2024. URL https://arxiv.org/abs/2401.12046.

Dian Wang, Mingxi Jia, Xupeng Zhu, Robin Walters, and Robert Platt. On-robot learning with equivariant models, 2022. URL https://arxiv.org/abs/2203.04923.

Fan Xie, Alexander Chowdhury, M. Clara De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep imitation learning for bimanual robotic manipulation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2327–2337. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/18a010d2a9813e91907ce88cd9143fdf-Paper.pdf.

Mingxi Jia, Haojie Huang, Zhewen Zhang, Chenghao Wang, Linfeng Zhao, Dian Wang, Jason Xinyu Liu, Robin Walters, Robert Platt, and Stefanie Tellex. Open-vocabulary pick and place via patch-level semantic maps. 2024. URL https://openreview.net/forum?id=cY3jXubzpR&referrer=%5BAuthor%20Console%5D(%2Fgroup%3Fid%3Drobot-learning.org%2FCoRL%2F2024%2FConference%2FAuthors%23your-submissions).

Neel Sortur, Linfeng Zhao, and Robin Walters. Sample efficient modeling of drag coefficients for satellites with symmetry. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=u7r2160QiP.

Linfeng Zhao, Owen Howell, Jung Yeon Park, Xupeng Zhu, Robin Walters, and Lawson L. S. Wong. Can euclidean symmetry be leveraged in reinforcement learning and planning? *arXiv preprint arXiv: 2307.08226*, 2023a.

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.

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.

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.

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.

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.

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

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.

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

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

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.

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.

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.

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

David Klee, Ondrej Biza, Robert Platt, and Robin Walters. Image to sphere: Learning equivariant features for efficient pose prediction. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_2bDpAtr7PI.

Owen Howell, David Klee, Ondrej Biza, Linfeng Zhao, and Robin Walters. Equivariant single view pose prediction via induced and restriction representations. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 47251–47263. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/93b3d975f9a2448964a906199db98a9d-Paper-Conference.pdf.

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.

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 2017b. ISSN 0731-5090, 1533-3884. https://doi.org/10/f9vx74. URL https://arc.aiaa.org/doi/10.2514/1.G001921.

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.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated Task and Motion Planning. *arXiv:2010.01083 [cs]*, October 2020. URL http://arxiv.org/abs/2010.01083. arXiv: 2010.01083.

Nishanth Kumar, Tom Silver, Willie McClinton, Linfeng Zhao, Stephen Proulx, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Jennifer Barry. Practice makes perfect: Planning to learn skill parameter policies. In *Robotics: Science and Systems (RSS)*, 2024.

Linfeng Zhao and Lawson L.S. Wong. Learning to navigate in mazes with novel layouts using abstract top-down maps. *Reinforcement Learning Journal*, 5: 2359–2372, 2024.

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].

Linfeng Zhao, Hongyu Li, Taşkın Padır, Huaizu Jiang, and Lawson LS Wong. E(2)-equivariant graph planning for navigation. *IEEE Robotics and Automation Letters*, 2024.

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.

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.

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].

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.

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. https://doi.org/10.1109/TIE.2022.3212397. Conference Name: IEEE Transactions on Industrial Electronics.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. *arXiv:1806.10293 [cs, stat]*, November 2018. URL http://arxiv.org/abs/1806.10293. arXiv: 1806.10293.

Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson.

Implicit Behavioral Cloning, August 2021. URL http://arxiv.org/abs/2109.00137. arXiv:2109.00137 [cs].

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, 134 (1):19–67, February 2005. ISSN 0254-5330, 1572-9338. https://doi.org/10/fkbjf3. URL http://link.springer.com/10.1007/s10479-005-5724-z.

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)*, pages 1433–1440, May 2016. https://doi.org/10/gf9knc.

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.

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

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

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

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.

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

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. https://doi.org/10/gq5m5s. URL https://ojs.aaai.org/index.php/AAAI/article/view/9649. Number: 1.

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, editors, *KI 2015: Advances in Artificial Intelligence*, volume 9324, pages 166–180. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24488-4 978-3-319-24489-1. https://doi.org/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. Structural Symmetries of the Lifted Representation of Classical Planning Tasks. page 8.

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. https://doi.org/10/gq5m5t. URL https://ojs.aaai.org/index.php/ICAPS/article/view/3509.

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. https://doi.org/10/ghkkbq. URL https://ojs.aaai.org/index.php/AAAI/article/view/4751. Number: 01.

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*, pages 688–695, Helsinki, Finland, 2008. ACM Press. ISBN 978-1-60558-205-4. https://doi.org/10/bkswc2. URL http://portal.acm.org/citation.cfm?doid=1390156.1390243.

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

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

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

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.

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

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

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.

Table of 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 and 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 and 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 and 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 and 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 and 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 and 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. [2022b] focuses on permutation symmetry in object-oriented transition models. Recent research by [Zhao et al., 2022a, 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 and Barto, 2004]. Furthermore, from equivariant network literature [Elesedy and Zaidi, 2021], the generalization gap for learned equivariant policy and value networks are smaller, which allows them to generalize better.

B.2 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.

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

$$\text{Existence of identity: } \exists e \in G, \text{ s.t. } \forall g \in G, \;\; e \cdot g = g \cdot e = g$$

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

For a complete reference on group theory, please see Zee [2016].

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

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

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 $\hat{G}$.

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

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

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

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

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

Fig. 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

$$\forall g \in G, \forall \omega \in \Omega, \quad \Psi(\Phi(g, \omega)) = \Phi'(g, \Psi(\omega)) \tag{13}$$

Diagrammatically, Ψ is G-equivariant if and only if the diagram C.2 is commutative.

D Theory and Proofs

This section includes more insights and explanation of equivariant sampling and its benefits, as well as the equivariance properties of Geometric MDP.

D.1 Toy Models of Equivariant Sampling

Let us consider a toy model of equivariant sampling. This will illustrate the importance of symmetry considerations in sampling methods. Specifically, this example illustrates that if sampling is performed incorrectly, the finite sample averages will not have the same symmetries as the infinite sample average. We show how the desired symmetries can be recovered via a 'group averaging'. Let $f : \mathbb{R} \to \mathbb{R}$ be any smooth function. Let us define the 'energy' function $H : \mathbb{R}^d \to \mathbb{R}$ as

$$H(x) = \mathbb{E}_\omega[f(\omega^T x)]$$

where the random vector $\omega \sim \mathcal{N}(0, \mathbb{I}_d)$ is drawn from a normal distribution with zero mean and identity matrix covariance $\mathbb{I}_d$. The random variable ω is isotropic and both ω and $O\omega$ are drawn from the same distribution for any orthogonal matrix O. Thus, we have that

$$\forall O \in O(d), \quad H(O \cdot x) = \mathbb{E}_\omega[f(\omega^T O x)] = \mathbb{E}_\omega[f((O\omega)^T x)] = \mathbb{E}_\omega[f(\omega x)] = H(x)$$

where we have used the fact that the random variable ω satisfies the property $\omega = O\omega$. Thus, $H(Ox) = H(x)$ is a left $O(d)$-invariant quantity. Now, suppose that we try to approximate $H(x)$ by random sampling. In the naive approach to estimation of $H(x)$, we can drawn m iid sample points $\omega_1, \omega_2, ..., \omega_m$ iid from $\mathcal{N}(0, \mathbb{I}_d)$ and estimate the function $H(x)$ as

$$\hat{H}(x) = \frac{1}{M} \sum_{m=1}^{M} f(\omega_i x)$$

However, this approximation $\hat{H}(x)$ does not need to satisfy the original symmetry property and $\hat{H}(Ox) = \hat{H}(x)$ is not guaranteed to hold. Because we know that the true function $H(x)$ is left $O(d)$-invariant, it seems like we should be able to

construct and estimate to $H(x)$ that is always left $O(d)$-invariant. Let G be a compact group, we can always 'symmetrize' the sample estimate $\hat{H}(x)$ via

$$\hat{H}_G(x) = \int_{g \in O(d)} dg \; \hat{H}(g \cdot x)$$

The symmetrized $\hat{H}_G(x)$ is then guaranteed to satisfy $\hat{H}_G(g \cdot x) = H_G(x)$ for all $g \in O(d)$. Furthermore, the symmetrized estimate $\hat{H}_G$ is always guaranteed to be a better estimate of H than $\hat{H}$. To see this, note that, for any function $F : G \to \mathbb{C}$,

$$|\int_{g \in G} dg \; F(g)| \leq \int_{g \in G} dg \; |F(g)|$$

holds via the triangle inequality. Thus, letting $F(g) = H(x) - \hat{H}(g \cdot x)$ we have that

$$|\int_{g \in G} dg \; H(x) - \hat{H}(g \cdot x)| \leq \int_{g \in G} dg \; |H(x) - \hat{H}(g \cdot x)|$$

Ergo, by definition of the symmetrized estimate $\hat{H}^G$, we have that

$$\int_{g \in G} dg \; |H(g \cdot x) - \hat{H}^G(g \cdot x)| \leq \int_{g \in G} dg \; |H(g \cdot x) - \hat{H}(g \cdot x)|$$

so that the error $|H(x) - \hat{H}^G(x)|$ is always less than the error $|H(x) - \hat{H}(x)|$ when averaged on G orbits. This is example is of course artificial. However, the sampling methodology developed in (ref main text) ensures that sample averages always have the same symmetries of the true energy functional is based on the same idea. It is easy to see that this can be extended from G-invariant to G-equivariant functions by modifying the averaging operator,

$$\int_{g \in O(d)} dg \; \hat{H}(g \cdot x) \to \int_{g \in O(d)} dg \; \rho(g^{-1}) \hat{H}(g \cdot x)$$

D.2 Equivariant Sampling Is Always Better

Let G be a compact group. Suppose that we have an MDP with symmetry with optimal policy $\pi^\star(a|s)$. Using a result of [van der Pol et al., 2020b], the optimal policy satisfies the relation $\forall g \in G, \quad \pi^\star(ga \mid gs) = \pi^\star(a \mid s)$.

Let us suppose that we have an learning policy π, which may or may not satisfy the $\pi(ga|gs) = \pi(a|s)$ condition derived in [van der Pol et al., 2020b]. Given any policy $\pi(a|s)$ we can always 'symmetrize' the policy by defining a new policy $\mathit{\Pi}_G[\pi] : S \to A$ defined as

$$\mathit{\Pi}_G[\pi](a|s) = \int_{g \in G} dg \; \pi(g \cdot a | g \cdot s)$$

Then, the symmetrized policy $\Pi_G[\pi]$ satisfies the condition

$$\forall g \in G, \quad \Pi_G[\pi](g \cdot a | g \cdot s) = \Pi_G[\pi](a|s)$$

The operator Π_G can be viewed as a operator which takes as input an arbitrary policy π and returns a policy $\Pi_G[\pi]$ which is G-invariant. Under the assumption of un-biasedness, the symmetrized policy $\Pi_G[\pi]$ is always better than the policy π. To see this, let D be a metric on the action space and consider the G-averaged error

$$\int_{g \in G} dg \; D(\pi^\star(ga|gs), \pi(ga|gs))$$

where $\pi^\star(ga|gs) = \pi^\star(a|s)$ is the true optimal policy. Now, using the triangle inequality, we have that

$$D(\pi^\star(a|s), \int_{g \in G} dg \; \pi(ga|gs)) \leq \int_{g \in G} dg \; D(\pi^\star(ga|gs), \pi(ga|gs))$$

Thus, using the definition of the G-averaged policy, we have that

$$D(\pi^\star(a|s), \Pi_G[\pi](a|s)) \leq \int_{g \in G} dg \; D(\pi^\star(ga|gs), \pi(ga|gs))$$

Using the fact that both $\pi^\star$ and $\Pi_G[\pi]$ are G-invariant, we can rewrite this as,

$$\int_{g \in G} dg \; D(\pi^\star(ga|gs), \Pi_G[\pi](ga|gs)) \leq \int_{g \in G} dg \; D(\pi^\star(ga|gs), \pi(ga|gs))$$

Ergo, the G-averaged policy $\Pi_G[\pi]$ is always closer to the true policy than the policy π. Thus, a arbitrary policy is always worse than its symmetrized counterpart.

D.3 Theorem 1: 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

$$\mathcal{T}[V](\boldsymbol{s}) := \max_{\boldsymbol{a}} R(\boldsymbol{s}, \boldsymbol{a}) + \int d\boldsymbol{s}' P(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}) V(\boldsymbol{s}'), \tag{14}$$

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:

$$[L_g f](x) = \left[f \circ g^{-1}\right](x) = \rho_{\text{out}}(g) \cdot f\left(g^{-1} x\right), \tag{15}$$

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 $\mathrm{E}(d)$:

$$L_g\left[\mathcal{T}[V]\right](\boldsymbol{s}) \overset{(1)}{=} \mathcal{T}[V](g^{-1}\boldsymbol{s}) \tag{16}$$

$$\overset{(2)}{=} \max_{\boldsymbol{a}} R(g^{-1}\boldsymbol{s}, \boldsymbol{a}) + \int d\boldsymbol{s}' \cdot P(\boldsymbol{s}' \mid g^{-1}\boldsymbol{s}, \boldsymbol{a}) V(\boldsymbol{s}') \tag{17}$$

$$\overset{(3)}{=} \max_{\bar{\boldsymbol{a}}} R(g^{-1}\boldsymbol{s}, g^{-1}\bar{\boldsymbol{a}}) + \int d(g^{-1}\bar{\boldsymbol{s}}) \cdot P(g^{-1}\bar{\boldsymbol{s}} \mid g^{-1}\boldsymbol{s}, g^{-1}\boldsymbol{a}) V(g^{-1}\bar{\boldsymbol{s}}) \tag{18}$$

$$\overset{(4)}{=} \max_{\bar{\boldsymbol{a}}} R(\boldsymbol{s}, \bar{\boldsymbol{a}}) + \int d(g^{-1}\bar{\boldsymbol{s}}) \cdot P(\bar{\boldsymbol{s}} \mid \boldsymbol{s}, \boldsymbol{a}) V(g^{-1}\bar{\boldsymbol{s}}) \tag{19}$$

$$\overset{(5)}{=} \max_{\bar{\boldsymbol{a}}} R(\boldsymbol{s}, \bar{\boldsymbol{a}}) + \int d\bar{\boldsymbol{s}} \cdot P(\bar{\boldsymbol{s}} \mid \boldsymbol{s}, \boldsymbol{a}) V(g^{-1}\bar{\boldsymbol{s}}) \tag{20}$$

$$\overset{(6)}{=} \mathcal{T}[L_g[V]](\boldsymbol{s}) \tag{21}$$

For each step:

- (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., 2022a], 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 and Balch, 2001], MDP homomorphism induced from symmetry group [Ravindran and Barto, 2004], and later work on symmetry in deep RL [Wang et al., 2021, van der Pol et al., 2020b].

We additionally discuss another theorem.

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 and Veličković, 2022].

Notice that they satisfy the following equivariance conditions:

$$P_\theta : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+ : \quad P_\theta(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t) = P_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_{t+1} \mid \rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{22}$$

$$R_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad R_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{23}$$

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

$$V_\theta : \mathcal{S} \to \mathbb{R} : \quad V_\theta(\boldsymbol{s}_t) = V_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t) \tag{25}$$

$$\tag{26}$$

We construct geometric message passing such that it uses *scalar* messages and features and resembles value iteration.

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.

$$\tilde{\mathbf{m}}_{ij} = \phi_m\left(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \tilde{\mathbf{a}}_{ij}\right) \tag{27}$$

$$\tilde{\mathbf{f}}_i' = \phi_f\left(\tilde{\mathbf{f}}_i, \sum_{j \in \mathcal{N}(i)} \tilde{\mathbf{m}}_{ij}, \tilde{\mathbf{a}}_i\right). \tag{28}$$

The tilde means they are steerable under G transformations.

We want to construct value iteration:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \tag{29}$$

$$V'(s) = \sum_a \pi(a|s) Q(s,a) \tag{30}$$

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., 2022a], while we leave it for future work.

E Algorithm Design of Equivariant TD-MPC

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:

$$\texttt{return}(\tau) = \mathbb{E}_\tau \left[\gamma^H Q_\theta \left(\boldsymbol{s}_H, \boldsymbol{a}_H \right) + \sum_{t=0}^{H-1} \gamma^t R_\theta \left(\boldsymbol{s}_t, \boldsymbol{a}_t \right) \right] = \mathbb{E}_\tau \left[U(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H-1}) \right] \tag{31}$$

$$\texttt{return}(g \cdot \tau) = \mathbb{E}_{g \cdot \tau} \left[\gamma^H \rho_0(g) \cdot Q_\theta \left(g \cdot \boldsymbol{s}_H, g \cdot \boldsymbol{a}_H \right) + \sum_{t=0}^{H-1} \gamma^t \rho_0(g) \cdot R_\theta \left(g \cdot \boldsymbol{s}_t, g \cdot \boldsymbol{a}_t \right) \right] \tag{32}$$

$$= \int_{g \in G} \rho_0(g) dg \cdot \mathbb{E}_\tau \left[U(g \cdot \boldsymbol{s}_{1:H}, g \cdot \boldsymbol{a}_{1:H-1}) \right] \tag{33}$$

$$= \mathbf{1} \cdot \mathbb{E}_\tau \left[U(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H-1}) \right] = \texttt{return}(\tau) \tag{34}$$

In Equation 32, we use $\rho_0(g) = \mathbf{1}$ to denote that the output is not transformed, so we may extract the term out. In Equation 33, dg is a Haar measure that absorbs the normalization factor, and we can extract the term from expectation. Equation 34 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., 2017a], 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., 2017a, 2015]) 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 and 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 $|D_8| = 16$).

We use two strategies: `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 $|D_8| = 16$)

The hidden space uses *regular* representation, which is common for discrete equivariant network [Zhao et al., 2022a, Weiler and Cesa, 2021, Cohen and Welling, 2016a].

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

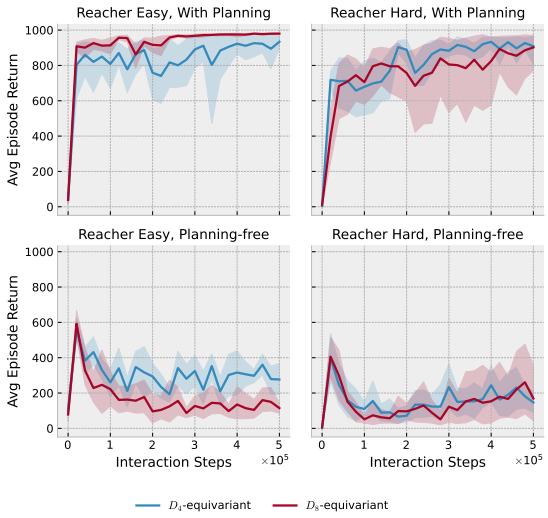

Fig. 8: Ablation study on planning component.

we aim to build model-based version of equivariant RL algorithm over model-free counterparts. The results are shown in Figure 8. On both `Reacher` Easy and Hard, with planning, the performance is much better.

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

$$f_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S} : \quad \rho_{\mathcal{S}}(g) \cdot f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = f_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{35}$$

$$R_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad R_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{36}$$

$$Q_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : \quad Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q_\theta(\rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t, \rho_{\mathcal{A}}(g) \cdot \boldsymbol{a}_t) \tag{37}$$

$$\pi_\theta : \mathcal{S} \to \mathcal{A} : \quad \rho_{\mathcal{A}}(g) \cdot \pi_\theta(\cdot \mid \boldsymbol{s}_t) = \pi_\theta(\cdot \mid \rho_{\mathcal{S}}(g) \cdot \boldsymbol{s}_t) \tag{38}$$

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 9. 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

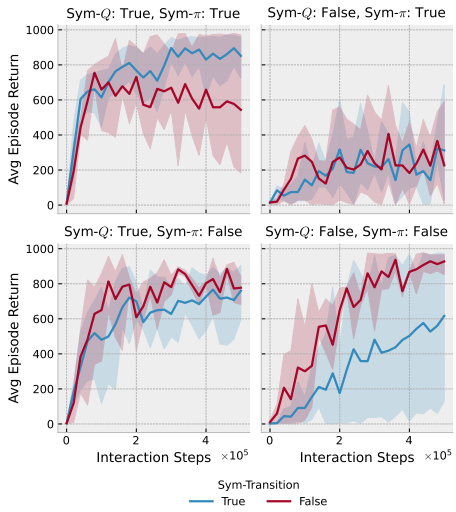

Fig. 9: Ablation study on equivariant components, using `Reacher` Hard with D_8 symmetry group.

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 10, 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.

The results are shown in Fig 11. 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 the-

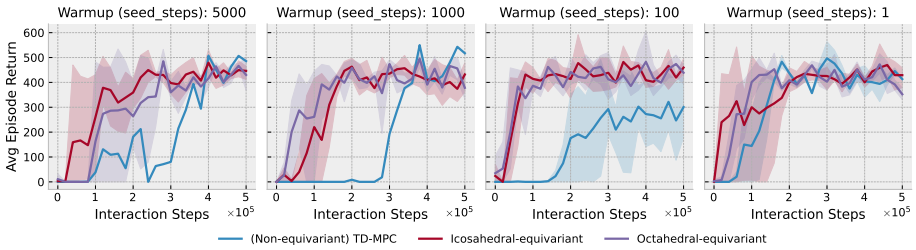

Fig. 10: Ablation study on number of warmup episodes on `PointMass` *3D with small target.*

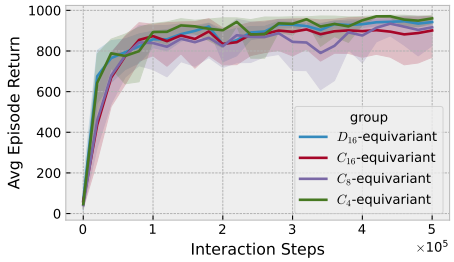

Fig. 11: Ablation study on symmetry group on `Reacher` Hard.

ory 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

$$\Big(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, x_g - x_f, y_g - y_f\Big) \Rightarrow \Big(\cos\theta_1, \sin\theta_1, \cos\theta_2, \sin\theta_2, \dot{\theta}_1, \dot{\theta}_2, x_g - x_f, y_g - y_f\Big) \tag{39}$$

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 12. 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 kine-

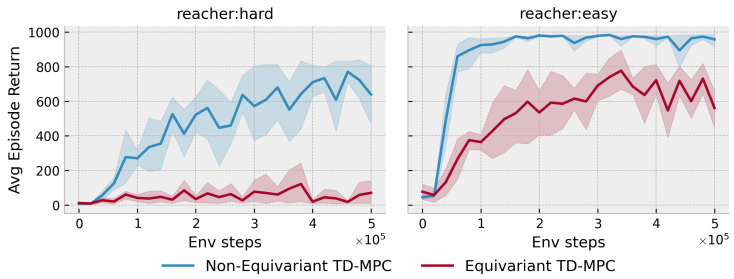

Fig. 12: Results for global reference frame on `Reacher`.

matic constraints (e.g., the length of the second link is preserved in $\cos\theta_2, \sin\theta_2$), compared to using equivariant feature.