Can Euclidean Symmetry Help in Reinforcement Learning and Planning?

Anonymous Authors[1]

Abstract

In robotic tasks, changes of reference frames typically do not affect the underlying physical meaning. These are isometric transformations, including translations, rotations, and reflections, called Euclidean group. In this work, we study reinforcement learning and planning tasks that have Euclidean group symmetry. We provide a theory that extends prior work (on symmetry in reinforcement learning, planning, and optimal control) to compact Lie groups and covers them as special cases, and show examples to explain the benefits of equivariance to Euclidean symmetry. We extend the 2D path planning with value-based planning to continuous MDPs and propose a pipeline for equivariant sampling-based planning algorithm with empirical evidence.

1. Introduction

Robot decision-making tasks often involve the movement of robots in two or three-dimensional Euclidean spaces. Different reference frames can be used to model the robot, but they do not change the underlying physical meaning of the task, indicating potential parameter sharing. Geometric transformations between reference frames in an Euclidean space $\mathbb{R}^d$ preserve distances and form a set of isometries, or Euclidean group $\mathrm{E}(d)$ of dimension d. This has been referred to as frame symmetry. In this work, we aim to answer: *Can Euclidean symmetry guarantee benefits in (model-based) RL algorithms?* Although the use of symmetry in decision-making has been studied in model-free or model-based reinforcement learning, planning, optimal control, and other related fields (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., 2022), they are not unified together. For example, equivariance has been studied in navigation on 2D grid, manipulation with top-down view, Atari games with discrete rotations and reflections, and optimal control on manifolds such as $\mathrm{SO}(3)$.

To address the aforementioned question, we restrict our attention to the symmetry of *changes of reference frame* using Euclidean symmetry groups (and their subgroups), such as 3D transformations $\mathrm{SE}(3)$ or, with reflections, $\mathrm{E}(3)$. We study a set of MDPs such that Euclidean symmetry can *transform* in both state and action spaces, naming them *Geometric MDPs*, which is motivated by the study of *geometric graphs* in geometric deep learning (Bronstein et al., 2021). This type of symmetry is derived from the physical space and is usually known *beforehand*, making it useful in solving MDPs. In addition to the previous definition of symmetry in MDPs (Ravindran and Barto, 2004; van der Pol et al., 2020a; Wang et al., 2021; Zhao et al., 2022), the group transformation is required to be *continuous*, enabling the study of quotient space and more equivariance properties. We present a theoretical framework that studies the linearized dynamics (LQR) and show that A, B matrices are G-steerable kernels, using tools from equivariant networks and representation theory. The theory covers prior work such that X is a homogeneous space or even a group and resumes to previous cases by restricting G to subgroups (such as reflections or discrete rotations).

Furthermore, we propose a *sampling-based* model-based RL algorithm for Geometric MDPs. It extends the prior work from planning on 2D grid with value-based planning to continuous state and action space, which requires the use of sampling-based planning. We take inspiration from geometric deep learning (Bronstein et al., 2021) and consider the features in neural networks to transform under Euclidean symmetry. The algorithm is constructed under the formalism and is shown to be equivariant. We validate our algorithm by analyzing a few tasks using the theory and demonstrating its efficacy through empirical evidence.

Our contributions are summarized as follows. (1) We study geometric structures under a specific set of MDPs, Geometric MDPs, and focus on Euclidean symmetry. (2) By analyzing the linearization of Geometric MDPs, our theory quantitatively shows the reduction of free parameters, supporting the use of Euclidean symmetry. (3) We propose a

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

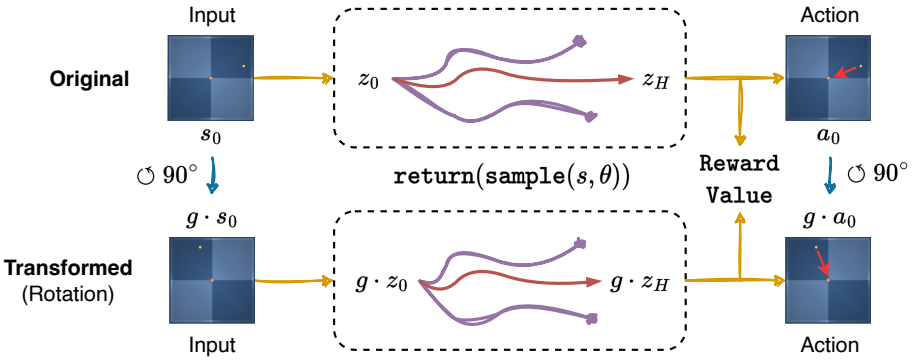

Figure 1. Illustration of equivariance in the proposed sampling-based planning algorithm $a_0 = \texttt{plan} = s_0$. The procedure is equivariant and the learned functions are G-equivariant networks. The sampling procedure produces equivariant trajectories, while the predicted quantities are G-invariant, such as values and rewards.

sampling-based model-based RL algorithm to integrate Euclidean symmetry, applicable to Geometric MDPs. (4) Our empirical evidence demonstrates its effectiveness in solving MDPs on a few continuous control tasks.

2. Problem Statement: Where does symmetry come from?

We focus on potential sharing in geometric transformations of MDPs between reference frames, which are isometric and form the Euclidean symmetry group $\mathrm{E}(d)$. These transformations can be written in semi-direct product form as $\left(\mathbb{R}^d, +\right) \rtimes G$, where G is the stabilizer group of origin and the action on a vector x has translation part t and rotations/reflections part g, i.e. $x \mapsto (tg) \cdot x := gx + t$. We focus on the compact group G, and translations can be implemented by relative position or careful choice of the coordinate system (Lang and Weiler, 2020; Brandstetter et al., 2021).

To transform an MDP (to a different reference frame), we require the MDP has the group G to act on a set X, such as the state or action space. This extends (Zhao et al., 2022) on 2D grid $\mathbb{Z}^2$ and is analogous to geometric graphs in geometric deep learning for supervised learning (Bronstein et al., 2021; Brandstetter et al., 2021). This definition unifies different types of prior work and allows X to be a homogeneous space, a group, or any other space as long as equipped with a G-action (van der Pol et al., 2020b; Wang et al., 2021; Zhao et al., 2022; Teng et al., 2023). The compact group $G \leq \mathrm{GL}(d)$ can be any group, including the group of proper 3D transformations $\mathrm{SO}(3)$ or finite subgroups like the icosahedral group or cyclic groups.

To this end, we define a class of MDPs that we can study their geometric structure and build the theory section upon. If the group action $\cdot_G : G \times X \to X$ is also *continuous*, there is a rich structure to study. This is not mandatory but is useful to relate symmetry with the linearized dynamics.

Definition 1 (Geometric MDP) *A Geometric MDP*

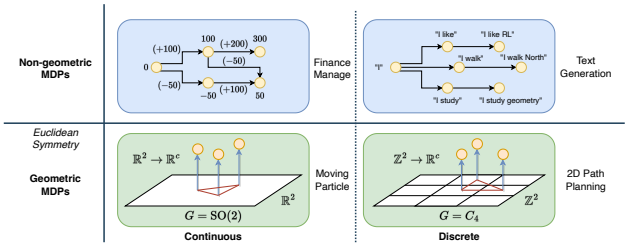

Figure 2. Illustration on MDPs with or without underlying geometric structures. The geometric structures underlying MDPs distinguish the tasks, no matter the underlying space is continuous or discrete.

(GMDP) $\mathcal{M}$ is an MDP with a (compact) symmetry group $G \leq \mathrm{GL}(d)$ that acts on the state and action space. It is written as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma, \mathcal{B}, G, \rho_{\mathcal{S}}, \rho_{\mathcal{A}} \rangle$. The state and action spaces $\mathcal{S}, \mathcal{A}$ have (continuous) group actions that transform them, defined by $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{A}}$.

3. Algorithm: How to Use Symmetry in Sampling-based Planning?

In this section, we aim to exploit the symmetry in Geometric MDPs $G \leq \mathrm{GL}(d)$, such as rotations and reflections, for sampling-based planning. We extend prior work (Zhao et al., 2022) that uses value-based planning on a discrete state space $\mathbb{Z}^2$ and discrete group D_4 to continuous case, necessitating sampling-based planning. The idea is to ensure that the algorithm $a_t = \texttt{plan}(s_t)$ produces same actions up to transformations, i.e., it is G-equivariant: $g \cdot a_t \equiv g \cdot \texttt{plan}(s_t) = \texttt{plan}(g \cdot s_t)$, shown in Figure 1. The principle is potentially applicable for MDPs with other groups.

We use TD-MPC (Hansen et al., 2022) as the backbone of our implementation and introduce their procedure. The principle of designing an equivariant sampling-based planning algorithm does not limit to a particular algorithm.

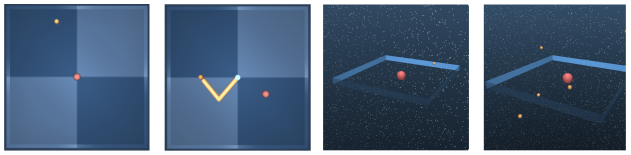

Figure 3. Sampled tasks that we use in experiments: (1) `PointMass` in 2D, (2) `Reacher`, (3) `PointMass` customized 3D version, and (4) `PointMass` customized 3D version with multiple particles to control.

Integrating symmetry. The major difference from (Zhao et al., 2022) is that we additionally need to consider how the sampling procedure "transforms" under symmetry. The equivariance in transition model 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 visualize in Figure 1.

1. **`dynamics` and `reward` model.** In the definition of symmetry in Geometric MDPs (and symmetric MDPs (Ravindran and Barto, 2004; van der Pol et al., 2020b; Zhao et al., 2022)) in Equation **??**, the transition and reward function are G-invariant. Therefore, in implementation, the transition network is deterministic and uses a G-equivariant MLP, and the reward network is constrained to be G-invariant.

2. **`value` and `policy` model.** The optimal value function produces a scalar for each state and is G-invariant, and the optimal policy function is G-equivariant (Ravindran and Barto, 2004). Assuming if we use equivariant/invariant transition and 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 policy from the value network, which is also equivariant (van der Pol et al., 2020b; Wang et al., 2021; Zhao et al., 2022).

3. **`MPC` procedure.** We consider the equivariance in MPC procedure in three parts: `sample` trajectories from the MDP, compute `return` of them, and use gradients of loss to `update`: $\theta' = \texttt{update}(\texttt{return}(\texttt{sample}(s, \theta)))$. We discuss the equivariance in the next subsection.

4. Evaluation: Sampling-based Planning

In this section, we present the setup and results for our proposed sampling-based planning algorithm: equivariant version of TD-MPC.

Tasks. We verify the algorithm on a few selected tasks from DeepMind Control suite (DMC) and several customized ones, visualized in Figure 3. One task is 2D particle moving in $\mathbb{R}^2$, named `PointMass`. We customize

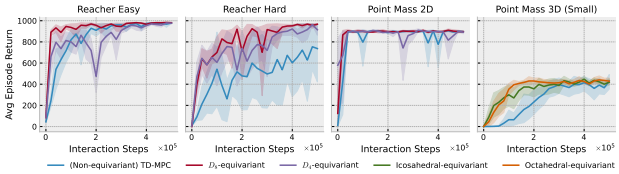

Figure 4. Results on `Reacher`, default `PointMass` 2D, and customized 3D `PointMass` with smaller target.

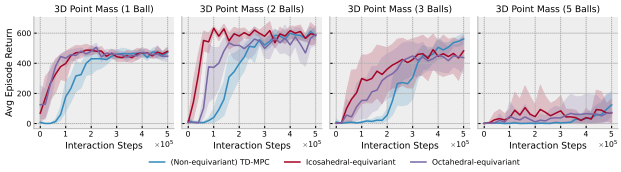

Figure 5. Results on a set of customized 3D N-ball `PointMass` tasks, with $N = 1, 2, 3, 5$.

tasks based on it: (1) 3D particle moving in $\mathbb{R}^3$ (disabled gravity), and (2) 3D N-point moving that has several particles to control simutaneously. The goal is still to move particle(s) to a target position (the origin). We also experiment two-arm manipulation tasks, `Reacher` (easy and hard), where the goal is to move the end-effector to a random position in a plane. In Section **??** , we additionally discuss tasks that Euclidean symmetry do not practically work better, which is related to the ratio of equivariant features discussed in theory.

Experimental setup. We compare against the non-equivariant version of TD-MPC (Hansen et al., 2022). Here, we by default make all components equivariant as described in the algorithm section. In appendix, we include ablation studies for disabling or enabling each equivariant component. The training procedure follows TD-MPC (Hansen et al., 2022). We use the state as input and for equivariant TD-MPC, we divide the orignal hidden dimension by $\sqrt{N}$ where N is the group order to keep the number of parameters roughly equal to the equivariant and non-equivariant version. 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 `escnn` package (Weiler and Cesa, 2021), as they perform more stably and For 2D case, we use $\mathrm{O}(2)$ subgroups: dihedral groups D_4 and D_8 (4 or 8 rotation components). For 3D case, we use Icosahedral group and Octahedral group, which are finite subgroups of $\mathrm{SO}(3)$ with order 60 and 24, respectively.

Results. In Figure 4 and 5, we show the reward curves in evaluation. `Reacher easy` and `hard` are top-down where the goal is to reach a random 2D position. If we rotate the MDP, the angle between the first and second links is not affect, i.e. G-invariant. The first joint and the tar-

get position are transformed under rotation, so we set to ρ_1 standard representation (2D rotation matrices). The complete state and action representations are given in Table **??**. The system has $\mathrm{O}(2)$ rotation and also reflection symmetry, and we use D_8 and D_4 groups. Shown in Figure 4, D_8 outperforms the non-equivariant TD-MPC by noticable margins, especially on `hard` one. D_4 is slightly worse than D_8 but still better than the baseline. With higher order discrete subgroups, the performance plateaus and does not worth the additional compute.

The default `PointMass` 2D version seems easy to solve, while D_8-equivariant version still learns faster. Thus, we design 3D version of `PointMass` and use $\mathrm{SO}(3)$ subgroups to implement 3D equivariant version of TD-MPC, because the implementation is significantly easier and the computational cost is lower compared to continuous version, which needs to convert between frequency domain and spatial domain. Figure 5 shows $N = 1, 2, 3, 5$ balls in 3D `PointMass`, and the rightmost figure in Figure 4 shows 1-ball 3D version with smaller target (0.02 compared to 0.03 in N-ball version). We find the Icosahedral (order 60) equivariant TD-MPC always learns faster and uses less samples to achieve best rewards. Octahedral (order 24) equivariant version is pretty close and is also mostly better. The best absolute rewards in 1-ball case is interestingly lower than 2- and 3-ball, which may be caused by higher possible reward due to 2 or 3 balls that can reach the goal.

We find TD-MPC is especially sensitive to a hyperparameter `seed_steps` that controls the number of warmup trajectories. In contrast, our equivariant version is robust to it and sometimes learn better with less warmup. We conjecture that this is related to the end-to-end learning of all components in the model-based RL algorithm (transition, reward, policy, value) with task-specific loss (purely reward-driven). Thus, the efficiency of training all components together matters, especially for sparse-reward goal-reaching tasks, where the equivariant network components start to shine.

References

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

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv:2104.13478 [cs, stat]*, Apr. 2021. URL http://arxiv.org/abs/2104.13478. arXiv: 2104.13478.

N. Hansen, X. Wang, and H. Su. Temporal Difference Learning for Model Predictive Control. Technical Report arXiv:2203.04955, arXiv, Mar. 2022. URL http://arxiv.org/abs/2203.04955. arXiv:2203.04955 [cs] type: article.

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

A. K. Mondal, P. Nair, and K. Siddiqi. Group Equivariant Deep Reinforcement Learning. *arXiv:2007.03437 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2007.03437. arXiv: 2007.03437.

J. Y. Park, O. Biza, L. Zhao, J. W. van de Meent, and R. Walters. Learning Symmetric Embeddings for Equivariant World Models. *arXiv:2204.11371 [cs]*, Apr. 2022. URL http://arxiv.org/abs/2204.11371. arXiv: 2204.11371.

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

S. Teng, D. Chen, W. Clark, and M. Ghaffari. An Error-State Model Predictive Control on Connected Matrix Lie Groups for Legged Robot Control, Jan. 2023. URL http://arxiv.org/abs/2203.08728. arXiv:2203.08728 [cs, eess].

E. van der Pol, D. Worrall, H. van Hoof, F. Oliehoek, and M. Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020a.

E. van der Pol, D. E. Worrall, H. van Hoof, F. A. Oliehoek, and M. 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.

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

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

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

M. Zinkevich and T. 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.