



# Equivariant diffusion policy for sample-efficient robotic manipulation

Dian Wang<sup>1,2</sup> , Stephen Hart<sup>3</sup>, David Surovik<sup>3</sup>, Tarik Kelestemur<sup>3</sup>, Haojie Huang<sup>1</sup> , Haibo Zhao<sup>1</sup>, Mark Yeatman<sup>3</sup>, Xupeng Zhu<sup>1</sup>, Boce Hu<sup>1</sup>, Mingxi Jia<sup>4</sup>, Jiuguang Wang<sup>3</sup>, Robin Walters<sup>1</sup> and Robert Platt<sup>1</sup>

## Abstract

Recent work has shown diffusion models are an effective approach to learning the multimodal distributions arising from demonstration data in behavior cloning. However, a drawback of this approach is the need to learn a denoising function, which is significantly more complex than learning an explicit policy. In this work, we propose Equivariant Diffusion Policy, a novel diffusion policy-learning method that leverages domain symmetries to obtain better sample efficiency and generalization in the denoising function. We theoretically characterize when a diffusion policy is equivariant and analyze the SO(2) symmetry of full 6-DoF control. We furthermore evaluate the method empirically on a set of 12 simulation tasks in MimicGen, and show that it obtains a success rate that is, on average, 34.5% higher than the baseline Diffusion Policy. We also evaluate the method on a real-world system to show that effective policies can be learned with relatively few training samples, whereas the baseline Diffusion Policy cannot.

## Keywords

equivariance, diffusion model, robotic manipulation

## Introduction

The recently proposed *Diffusion Policy* (Chi et al., 2023) formulates robotic manipulation action prediction as a diffusion model that denoises the action conditioned on the observation, thereby better capturing the multimodal action distribution of the demonstration data in Behavior Cloning (BC). Although Diffusion Policy often outperforms baselines on various benchmarks (Gupta et al., 2020; Mandlekar et al., 2022), a significant limitation is that the denoising function is inherently more complex than a standard policy function. Specifically, for a single state-action pair  $(s, a)$ , the denoising process utilizes a mapping  $(s, a + \varepsilon^k, k) \mapsto \varepsilon^k$  for all possible  $k$  and  $\varepsilon^k$ , where  $\varepsilon^k$  is Gaussian noise conditioned on step  $k$ . This formulation creates a considerably more challenging learning problem compared with an explicit BC  $s \mapsto a$ .

In this paper, we leverage equivariant neural network models to incorporate task symmetry as an inductive bias in the diffusion process, substantially simplifying the denoising function learning problem. Although equivariant diffusion models have been studied by a number of prior works (Brehmer et al., 2023; Chen et al., 2023; Guan et al., 2023; Hoogeboom et al., 2022; Ryu et al., 2023b), our work

is the first to comprehensively study and implement this concept for visuomotor policy learning. As illustrated in Figure 1, when a state and noisy trajectory action are rotated about the gravity axis, the corresponding denoised trajectory undergoes an equivalent transformation. This symmetry-aware approach enables our model to achieve significantly greater data efficiency and generalization capabilities than non-symmetric baselines, effectively addressing the high data requirements typically associated with diffusion-based methods.

Our contributions are as follows:

- We propose Equivariant Diffusion Policy, a novel BC approach based on equivariant diffusion.

<sup>1</sup>Northeastern University, Boston, MA, USA

<sup>2</sup>Stanford University, Stanford, CA, USA

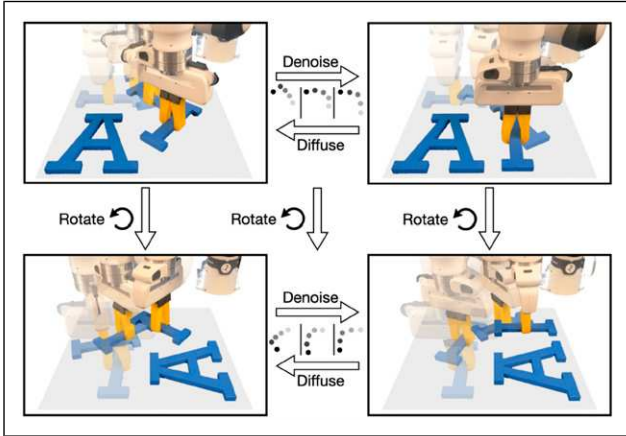
<sup>3</sup>Robotics and AI Institute, Cambridge, MA, USA

<sup>4</sup>Brown University, Providence, RI, USA

## Corresponding author:

Dian Wang, Computer Science Department, Stanford University, 353 Jane Stanford Way, Stanford, CA 94305, USA.

Email: [wang.dian@northeastern.edu](mailto:wang.dian@northeastern.edu)



**Figure 1.** Equivariance in diffusion policy. Top left: a randomly sampled trajectory. Top right: a valid trajectory after denoising. If the state and the random trajectory are both rotated (bottom left), and we rotate the noise accordingly in the denoising process, we will end up with a successful trajectory in the rotated state (bottom right).

- We theoretically show that the diffusion policy is equivariant when the denoising function is equivariant, justifying modeling the denoising function using an equivariant network.
- We theoretically demonstrate the use of  $SO(2)$ -equivariance in the context of 6-DoF control for robotic manipulation, which prior methods (Jia et al., 2023; Wang et al., 2022b) leveraged in a less expressive  $SE(2)$  action space.
- We provide a thorough demonstration of our method in both simulated and physical systems. In simulation, we evaluate on 12 manipulation tasks in the MimicGen benchmark (Mandlekar et al., 2023) and outperform the baseline Diffusion Policy by an average success rate of 34.5% when trained with 100 demos. On hardware, we show that successful policies can be learned with a small number of demonstrations for 12 different manipulation tasks, including long-horizon tasks like bagel baking, coffee making, etc.

This work is an extended version of our conference paper (Wang et al., 2024b), substantially expanding both the theoretical foundations and practical implementations of Equivariant Diffusion Policy. Particularly, while our conference version focused on  $SO(2)$  rotational equivariance, here we include  $T(3)$  (the group of 3D translations) translational symmetry alongside rotational symmetry. This extension is enabled by our novel Equivariant Point Transformer architecture, which processes point cloud inputs in a manner that preserves both types of symmetries. Concretely, we extend the content of the prior work in the following ways:

- We update the theoretical analysis of Equivariant Diffusion Policy, refining the proposition and proof

from a probability density perspective that is more comprehensive. See Section Theory of Equivariant Diffusion Policy.

- We propose a new version of Equivariant Diffusion Policy using point cloud input, referred to as EquiDiff (PC), powered by our novel Equivariant Point Transformer architecture. See Section Equivariant Point Transformer and Translation Symmetry.
- EquiDiff (PC) achieves additional translational symmetry (and is thus  $SO(2) \times T(3)$ -equivariant) and reaches a performance that is 12.6% higher than the conference version. See Section Standard Baseline Comparison.
- We include an ablation study for EquiDiff (PC) in Section Ablation Study studying the effect of each design choice of our work.
- We include a new real-world experiment with six new challenging manipulation tasks, demonstrating the advantage of EquiDiff (PC) compared with the version from the conference paper. See Section EquiDiff with Point Cloud Input.

## Related work

### Diffusion models

Diffusion models (Sohl-Dickstein et al., 2015) learn distributions by modeling the reverse of a diffusion process, which is a Markov chain that gradually adds Gaussian noise to the data until it transitions to a Gaussian distribution. Denoising diffusion models (Ho et al., 2020; Song and Ermon, 2019) can be interpreted as learning the gradient field of an implicit score during training, where inference applies a sequence of score optimization steps. This new family of generative methods has proven to be effective for capturing multimodal distributions in planning (Janner et al., 2022; Liang et al., 2023) and policy learning (Chi et al., 2023; Pearce et al., 2022; Wang et al., 2022c; Xian et al., 2023; Ze et al., 2024). However, these methods did not leverage the geometric symmetries underlying the task and the diffusion process. Xu et al. (2022); Hoogeboom et al. (2022) show that leveraging  $SO(3)$  symmetries from the domain in the diffusion process dramatically improves sample efficiency and generalization ability in molecular generation. EDGI (Brehmer et al., 2023) extends diffuser (Janner et al., 2022) to equivariant diffusion planning with improved performance, but relies on the ground-truth state as the input. Ryu et al. (2023b) propose bi-equivariant diffusion models for visual robotic manipulation, while limited to open-loop settings. Yang et al. (2024a) integrate  $SIM(3)$ -equivariant networks with diffusion models to enable scalable, generalizable policy, but is limited to tasks involving a single object. By contrast, we exploit domain

symmetries during the diffusion process to attain an effective closed-loop visuomotor policy for complex manipulation tasks.

### Equivariance in manipulation policies

Robots operate within a three-dimensional Euclidean space, where manipulation tasks inherently encompass geometric symmetries such as rotations. Recent works (Eisner et al., 2024; Gao et al., 2024; Hu et al., 2024; Huang et al., 2023a, 2023b; Jia et al., 2023; Kim et al., 2023; Kohler et al., 2023; Lim et al., 2024; Liu et al., 2023; Nguyen et al., 2023, 2024; Pan et al., 2023; Simeonov et al., 2023; Wang et al., 2021a, 2022b; Yang et al., 2024b) compellingly show that improvement in sample efficiency and performance can be obtained by leveraging symmetries in policy learning. (Wang et al., 2022a; Zhu et al., 2022, 2023) show the efficiency of equivariant models for on-robot learning. (Huang et al., 2022, 2023; 2024a; 2024b; Ryu et al., 2023a; Simeonov et al., 2022) learn an open-loop pick and place policy with few demonstrations. While this prior work either considers symmetries in SE (3) open-loop or SE (2) closed-loop action spaces, our paper studies symmetries in an SE (3) closed-loop action space, and is the first one to study the symmetries in diffusion policy.

### Closed-loop visuomotor control

Closed-loop visuomotor policies are more robust and responsive but struggle with learning from diverse trajectories and predicting long-horizon actions. Previous methods (Florence et al., 2019; Rahmatizadeh et al., 2018; Toyer et al., 2020; Zhang et al., 2018) directly map from observations to actions. However, this type of explicit policy-learning struggles to learn multimodal behavior distributions and may not be expressive enough to capture the full range and fidelity of trajectory data (Orsini et al., 2021; Pearce et al., 2022). Several works propose implicit policies (Florence et al., 2021; Jarrett et al., 2020) with energy-based models (Du and Mordatch, 2019; Grathwohl et al., 2020). However, training is challenging due to the necessity of a substantial volume of negative samples to effectively learn an optimal energy score function for state-action pairs. Recently, (Chi et al., 2023; Pearce et al., 2022) model action generation as a conditional denoising diffusion process and demonstrate strong performance by adapting diffusion models to sequential environments. Our work builds on Chi et al. (2023) but focuses on equivariance in the diffusion process.

## Background

### Problem statement

We study policy learning using behavior cloning. The agent is required to learn a mapping from the observation  $\mathbf{o}$  to the

action  $\mathbf{a}$  that mimics an expert policy. Both  $\mathbf{o}$  and  $\mathbf{a}$  can contain a number of time steps, that is,  $\mathbf{o} = \{\mathbf{o}_{t-(m-1)}, \dots, \mathbf{o}_{t-1}, \mathbf{o}_t\}$ ,  $\mathbf{a} = \{\mathbf{a}_t, \mathbf{a}_{t+1}, \dots, \mathbf{a}_{t+(n-1)}\}$  where  $m$  is the number of history steps observed and  $n$  is the number of future action steps. The observation contains both visual information (images or voxels) and the pose vector of the gripper.

Let  $T_t \in \mathbb{R}^{4 \times 4}$  be the current SE (3) pose of the gripper in the world frame, the actions  $\mathbf{a}_t$  specify a desired pose  $\mathbf{A}_t \in \mathbb{R}^{4 \times 4}$  of the gripper and an open-width command  $w_t \in \mathbb{R}$ . The pose can be either absolute ( $T_{t+1} = \mathbf{A}_t$ , also called position control) or relative ( $T_{t+1} = \mathbf{A}_t T_t$ , also called velocity control). In order to noise and denoise via addition and subtraction as in the standard diffusion process, we vectorize the SE (3) pose  $\mathbf{A}_t$  into a vector  $\mathbf{a}_t$  during diffusion and denoising, and orthogonalize the noise-free action vector after denoising.

### Diffusion policy

Chi et al. (2023) proposed Diffusion Policy to model the multimodal distribution in behavior cloning using Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020). Diffusion Policy learns a noise prediction function  $\varepsilon_\theta(\mathbf{o}, \mathbf{a} + \varepsilon^k, k) = \varepsilon^k$  using a network  $\varepsilon_\theta$  parameterized by  $\theta$ . The network is expected to predict the noise component of the input  $\mathbf{a} + \varepsilon^k$ . During training, transitions  $(\mathbf{o}, \mathbf{a})$  are sampled from the expert dataset. Then, random noise  $\varepsilon^k$  (conditioned on a randomly sampled denoising step  $k$ ) is added to  $\mathbf{a}$ . The loss is  $\mathcal{L} = \|\varepsilon_\theta(\mathbf{o}, \mathbf{a} + \varepsilon^k, k) - \varepsilon^k\|^2$ . During inference, given an observation  $\mathbf{o}$ , DDPM performs a sequence of  $K$  denoising steps starting from a random action  $\mathbf{a}^k \sim \mathcal{N}(0, 1)$  to generate an action  $\mathbf{a}^0$  defined inductively by

$$\mathbf{a}^{k-1} = \alpha(\mathbf{a}^k - \gamma \varepsilon_\theta(\mathbf{o}, \mathbf{a}^k, k) + \epsilon), \quad (1)$$

where  $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ .  $\alpha, \gamma, \sigma$  are functions of the denoising step  $k$  (also known as the noise schedule). The action  $\mathbf{a}^0$  is expected to be a sample from the expert policy  $\pi$ .

### Equivariance

A function  $f$  is equivariant if it commutes with the transformations of a symmetry group  $G$ . Specifically,  $\forall g \in G$ ,

$$f(\rho_x(g)x) = \rho_y(g)f(x), \quad (2)$$

where  $\rho: G \rightarrow GL(n)$  is called the group representation that maps each group element to an  $n \times n$  invertible matrix that acts on the input and output through matrix multiplication. We sometimes leave the actions implicit and write  $f(gx) = gf(x)$ .

We mainly focus on the group  $\text{SO}(2) \times T(3)$ , where  $\text{SO}(2)$  is a group of planar rotations (i.e., rotation around the z-axis of the world) and  $T(3)$  is a group of 3D translations. This group

captures the symmetry in many robotic tasks without enforcing unnecessary out-of-plane rotation equivariance (which is often invalid due to gravity and the canonical pose of objects). Notice that  $\text{SO}(2) \times T(3)$  can be decomposed and a function that is both  $\text{SO}(2)$ -equivariant and  $T(3)$ -equivariant would be  $\text{SO}(2) \times T(3)$ -equivariant.

We sometimes approximate  $\text{SO}(2)$  with the discrete subgroup  $C_u \subset \text{SO}(2)$  containing  $u$  discrete rotations, and there are three particular representations of  $\text{SO}(2)$  or  $C_u$  that are of interest in this paper:

- 1) the trivial representation  $\rho_0$  defines  $\text{SO}(2)$  or  $C_u$  acting on an invariant scalar  $x \in \mathbb{R}$  by  $\rho_0(g)x = x$ .
- 2) the irreducible representation  $\rho_\omega$  defines  $\text{SO}(2)$  or  $C_u$  acting on a vector  $v \in \mathbb{R}^2$  by a  $2 \times 2$  rotation matrix with frequency  $\omega$ ,  $\rho_\omega(g)v = \begin{pmatrix} \cos \omega g & -\sin \omega g \\ \sin \omega g & \cos \omega g \end{pmatrix} v$ .
- 3) the regular representation  $\rho_{\text{reg}}$  that defines  $C_u$  acting on a vector  $x \in \mathbb{R}^u$  by  $u \times u$  permutation matrices. Let  $g = r^m \in C_u = \{1, r^1, \dots, r^{u-1}\}$  and  $(x_1, \dots, x_u) \in \mathbb{R}^u$ . Then  $\rho_{\text{reg}}(g)x = (x_{u-m+1}, \dots, x_u, x_1, x_2, \dots, x_{u-m})$  cyclically permutes the coordinates of  $\mathbb{R}^u$ .

A representation  $\rho$  can also be a combination of different representations, that is,  $\rho = \rho_0^{n_0} \oplus \rho_1^{n_1} \oplus \rho_2^{n_2} \in \text{GL}(n_0 + 2n_1 + 2n_2)$ . In such a case,  $\rho(g)$  is an  $(n_0 + 2n_1 + 2n_2) \times (n_0 + 2n_1 + 2n_2)$  block diagonal matrix that acts on  $x \in \mathbb{R}^{n_0+2n_1+2n_2}$ .

## Method

### Theory of equivariant diffusion policy

The main contribution of this paper is a method that incorporates equivariance in the diffusion process for policy learning. As a theoretical justification, we analyze the noise prediction function  $\varepsilon$  and show that if  $\varepsilon$  is equivariant, then the policy being modeled is also equivariant. This implies equivariant neural networks have the correct inductive bias to model this function.

**Proposition 1.** *If the noise prediction function  $\varepsilon: \mathbf{o}, \mathbf{a}^k \rightarrow \varepsilon^k$  is  $\text{SO}(2)$ -equivariant, that is, for all  $g \in \text{SO}(2)$ ,*

$$\varepsilon(g\mathbf{o}, g\mathbf{a}^k, k) = g\varepsilon(\mathbf{o}, \mathbf{a}^k, k), \quad (3)$$

then the policy function is  $\text{SO}(2)$ -equivariant, that is, for all  $g \in \text{SO}(2)$  and any measurable set  $\mathcal{A}$ ,

$$\pi(g\mathcal{A} | g\mathbf{o}) = \pi(\mathcal{A} | \mathbf{o}), \quad (4)$$

where  $g\mathcal{A} = \{g\mathbf{a} | \mathbf{a} \in \mathcal{A}\}$ .

**Proof.** Notation and Setup.

Let  $q(\mathbf{a}^{k-1} | \mathbf{a}^k, \varepsilon^k)$  denote the transition density induced by the DDPM update,

$$\mathbf{a}^{k-1} = \alpha(\mathbf{a}^k - \gamma\varepsilon^k + \epsilon),$$

where the noise  $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$  is sampled from an isotropic Gaussian and  $\alpha$ ,  $\gamma$ , and  $\sigma$  depend only on  $k$  (and are invariant under the group action).

Let  $p_k(\mathbf{a}^k | \mathbf{o})$  denote the probability density of the intermediate state  $\mathbf{a}^k$  given the observation  $\mathbf{o}$ . The policy  $\pi$  is defined by probability density  $p_\pi(\mathbf{a} | \mathbf{o}) = p_0(\mathbf{a} | \mathbf{o})$ . For any measurable set  $\mathcal{A}$ ,

$$\pi(\mathcal{A} | \mathbf{o}) = \int_{\mathcal{A}} p_\pi(\mathbf{a} | \mathbf{o}) d\mathbf{a}.$$

Invariance of  $q$ .

Since the group action is linear and acts identically on  $\mathbf{a}^k$ ,  $\varepsilon^k$ , and  $\epsilon$ , applying a rotation  $g$  to the DDPM update gives

$$g\mathbf{a}^{k-1} = \alpha(g\mathbf{a}^k - \gamma g\varepsilon^k + g\epsilon).$$

Because the Gaussian density is invariant under rotations (i.e., the probability density at  $g\epsilon$  equals that at  $\epsilon$ ) and rotations preserve volume, we have

$$q(\mathbf{a}^{k-1} | \mathbf{a}^k, \varepsilon^k) = q(g\mathbf{a}^{k-1} | g\mathbf{a}^k, g\varepsilon^k). \quad (5)$$

Substituting the equivariance condition (Equation (3)) into Equation (5) yields

$$q(g\mathbf{a}^{k-1} | g\mathbf{a}^k, \varepsilon(g\mathbf{o}, g\mathbf{a}^k, k)) = q(\mathbf{a}^{k-1} | \mathbf{a}^k, \varepsilon(\mathbf{o}, \mathbf{a}^k, k)).$$

Since  $\mathbf{o}$  influences the density only via  $\varepsilon$ , we can write

$$q(g\mathbf{a}^{k-1} | g\mathbf{a}^k, g\mathbf{o}) = q(\mathbf{a}^{k-1} | \mathbf{a}^k, \mathbf{o}). \quad (6)$$

Proof by Induction.

For  $k = 0, 1, \dots, K$ , we prove by backward induction that

$$p_k(g\mathbf{a}^k | g\mathbf{o}) = p_k(\mathbf{a}^k | \mathbf{o}) \quad \text{for all } g \in \text{SO}(2).$$

Base Case ( $k = K$ ):

Since the initial state  $\mathbf{a}^K$  is drawn from an isotropic Gaussian (and is independent of  $\mathbf{o}$ ), its density is given by

$$p_K(\mathbf{a}^K | \mathbf{o}) = p_{\mathcal{N}(0, I)}(\mathbf{a}^K),$$

where  $p_{\mathcal{N}(0, I)}$  is the density function of the Gaussian  $\mathcal{N}(0, I)$ . Because the Gaussian density is invariant under rotations, for any  $g \in \text{SO}(2)$  we have

$$\begin{aligned} p_K(\mathbf{a}^K | \mathbf{o}) &= p_{\mathcal{N}(0, I)}(\mathbf{a}^K) \\ &= p_{\mathcal{N}(0, I)}(g\mathbf{a}^K) \\ &= p_K(g\mathbf{a}^K | g\mathbf{o}). \end{aligned}$$

Inductive Step:

Assume that for some  $k \in \{1, \dots, K\}$ ,

$$p_k(g\mathbf{a}^k | g\mathbf{o}) = p_k(\mathbf{a}^k | \mathbf{o})$$

holds for all  $g \in \text{SO}(2)$ . Then the density at step  $k - 1$  is given by

$$p_{k-1}(\mathbf{a}^{k-1} | \mathbf{o}) = \int q(\mathbf{a}^{k-1} | \mathbf{a}^k, \mathbf{o}) p_k(\mathbf{a}^k | \mathbf{o}) d\mathbf{a}^k.$$

Similarly, for transformed observation and action,

$$p_{k-1}(g\mathbf{a}^{k-1} | g\mathbf{o}) = \int q(g\mathbf{a}^{k-1} | \mathbf{a}^k, g\mathbf{o}) p_k(\mathbf{a}^k | g\mathbf{o}) d\mathbf{a}^k.$$

We change variables of integration from  $\mathbf{a}^k$  to  $g\mathbf{a}^k$ . Since rotations preserve the set  $\{\mathbf{a}^{k-1}\}$  and preserve volume ( $d(g\mathbf{a}^k) = d\mathbf{a}^k$ ),

$$\begin{aligned} p_{k-1}(g\mathbf{a}^{k-1} | g\mathbf{o}) \\ = \int q(g\mathbf{a}^{k-1} | g\mathbf{a}^k, g\mathbf{o}) p_k(g\mathbf{a}^k | g\mathbf{o}) d\mathbf{a}^k. \end{aligned} \quad (7)$$

Substituting Equation (6) into Equation (7) and applying the inductive hypothesis,

$$\begin{aligned} p_{k-1}(g\mathbf{a}^{k-1} | g\mathbf{o}) \\ = \int q(\mathbf{a}^{k-1} | \mathbf{a}^k, \mathbf{o}) p_k(\mathbf{a}^k | \mathbf{o}) d\mathbf{a}^k \\ = p_{k-1}(\mathbf{a}^{k-1} | \mathbf{o}). \end{aligned}$$

as desired. In particular, for  $k = 0$  we have

$$p_\pi(g\mathbf{a} | g\mathbf{o}) = p_\pi(\mathbf{a} | \mathbf{o}).$$

Equivariance of  $\pi$ .

By the invariance of the density  $p_\pi$ , for any measurable set  $\mathcal{A}$  we obtain

$$\begin{aligned} \pi(\mathcal{A} | \mathbf{o}) &= \int_{\mathcal{A}} p_\pi(\mathbf{a} | \mathbf{o}) d\mathbf{a} \\ &= \int_{g\mathcal{A}} p_\pi(\mathbf{a} | g\mathbf{o}) d\mathbf{a} \\ &= \pi(g\mathcal{A} | g\mathbf{o}). \end{aligned}$$

Thus, the policy function is SO (2)-equivariant.

Figure 2 illustrates the equivariance property of  $\varepsilon$ . If we infer  $\varepsilon$  for all actions in the action space, we effectively acquire a gradient field towards the expert trajectory. The figure shows that if the function  $\varepsilon$  is equivariant, such a gradient field would also be equivariant. Thus, the expert policy is equivariant. Notice that the figure shows the average of all action time steps.

### SO (2) representation on 6DoF action

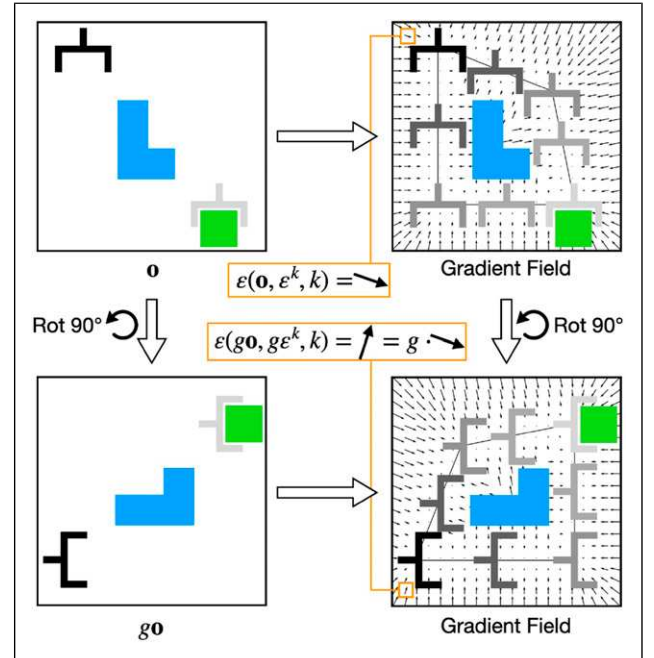
A key step in defining an Equivariant Diffusion Policy is to define how actions  $\mathbf{a}$ , transform linearly under SO (2). We describe this SO (2) transformation in terms of irreducible SO (2) representations, which allows us to build the equivariance constraint into the denoising network.

**Proposition 2.** *There exist irreducible representations that describe how SO (2) acts on an SE (3) gripper action  $\mathbf{a}_t$ . In absolute pose control, let  $\mathbf{a}_t = \text{Vec}_c(\mathbf{A}_t)$  where  $\text{Vec}_c$  flattens an SE (3) pose  $\mathbf{A}_t \in \mathbb{R}^{4 \times 4}$  into a vector by column,  $g\mathbf{a}_t = (\rho_1 \oplus \rho_0^2)^4(g)\mathbf{a}_t$ . In relative-pose control, let  $\mathbf{a}_t = \text{Vec}_r(\mathbf{A}_t)$  where  $\text{Vec}_r$  flattens  $\mathbf{A}_t$  into a vector by row,  $g\mathbf{a}_t = P^{-1}[(\rho_0^6 \oplus \rho_1^4 \oplus \rho_2)(g)]P\mathbf{a}_t$ , where  $P$  is a fixed change-of-basis matrix.*

**Absolute control.** We first consider absolute pose control, where the model infers the absolute pose to which the gripper is to move, that is,  $T_{t+1} = \mathbf{A}_t$ . Let  $T_g$  be the transformation matrix corresponding to the SO (2) rotation along the z-axis of the world frame,

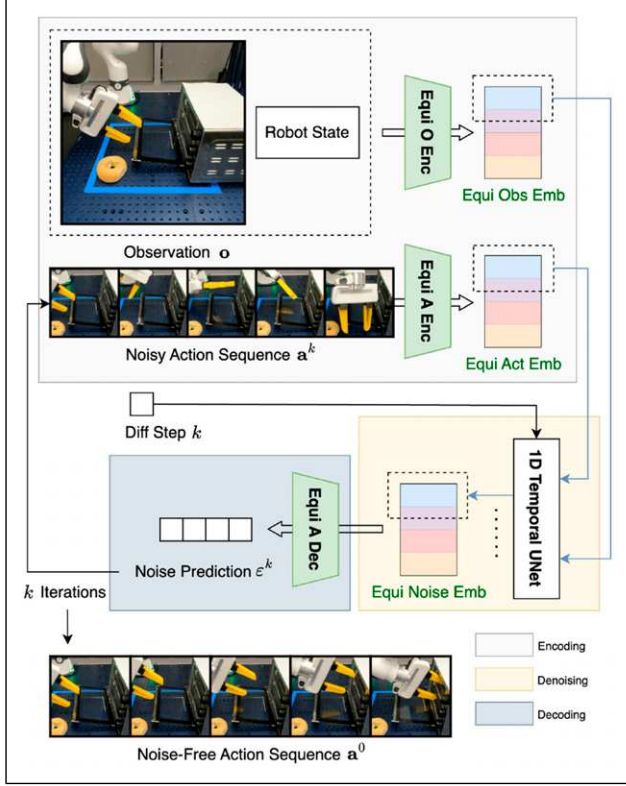
$$T_g = \begin{bmatrix} \cos g & -\sin g & 0 & 0 \\ \sin g & \cos g & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \rho_1(g) & & & \\ & \rho_0(g) & & \\ & & \rho_0(g) & \\ & & & \rho_0(g) \end{bmatrix}, \quad (8)$$

where  $\rho_1(g) = \begin{pmatrix} \cos g & -\sin g \\ \sin g & \cos g \end{pmatrix}$ . The SO (2) action on  $\mathbf{A}_t$  is  $g\mathbf{A}_t = T_g\mathbf{A}_t = (\rho_1 \oplus \rho_0^2)(g)\mathbf{A}_t$ . Vectorizing  $\mathbf{A}_t$  by column



**Figure 2.** Equivariance of the denoising function  $\varepsilon$ . Left: In observation  $\mathbf{o}$ , the goal for the gripper is to reach the green block while avoiding the blue obstacle. Right: The expert trajectory and the gradient field associated with the denoising function. If the policy is equivariant, both the denoising function and the entire gradient field are equivariant. The orange boxes show the equivariance of  $\varepsilon$  with a particular input  $\varepsilon^k$ .



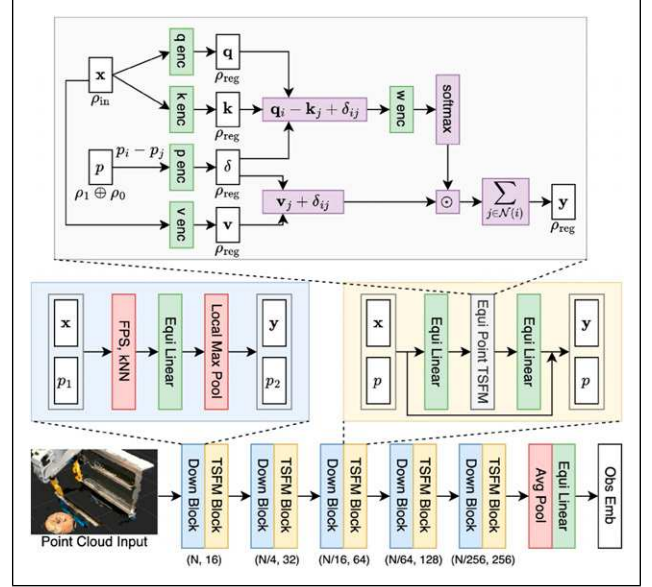


**Figure 3.** Overview of our Equivariant Diffusion Policy architecture. The first two equivariant encoders take the input observation  $\mathbf{o}$  and the noisy action sequence  $\mathbf{a}^k$  and generate the equivariant observation and action embeddings. Second, a 1D Temporal U-Net will generate the equivariant noise embedding, which is further decoded into the noise prediction  $\varepsilon^k$ . We can then perform a denoising step using Equation (1) to get  $\mathbf{a}^{k-1}$ . The above process is iterated  $k$  times until we acquire the noise-free action sequence  $\mathbf{a}^0$ .

equivariant embedding of the noise in the regular representation. We refer to this strategy of processing each element of the regular representation as the “pointwise equivariant processing.” Finally, an equivariant decoder will decode the noise  $\varepsilon^k$ .

### Equivariant point transformer and translation symmetry

We consider different input modalities in our work (i.e., images, voxel grids, and point clouds). Although images and voxel grids can be effective visual observations, point clouds have the advantage of directly capturing the 3D geometry of objects without the limited resolution problem that typically exists with voxel grids. For image and voxel inputs, we can use a simple 2D or 3D equivariant CNN as the equivariant observation encoder (as in Figure 3). However, with point cloud inputs, we found that a simple equivariant version of PointNet (Qi



**Figure 4.** Bottom: Our equivariant point transformer architecture. Top (gray): equivariant point transformer layer; middle left (blue): down sample block; middle right (yellow): equivariant point transformer block.

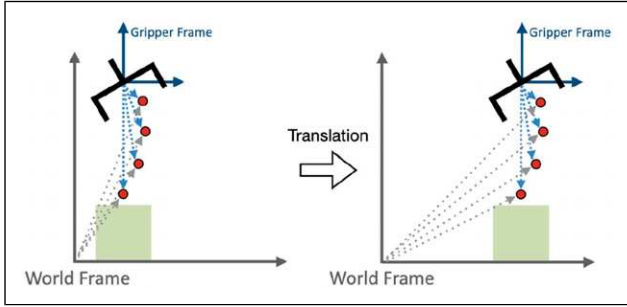
et al., 2017) does not perform well enough in complex manipulation tasks. To address this limitation, we propose an Equivariant Point Transformer that effectively captures geometric relationships while maintaining equivariance properties.

Our architecture is based on point transformer (Zhao et al., 2021). Let  $\mathbf{x}$  be the input feature vector;  $\mathbf{p}$  be the input point coordinates;  $\mathbf{y}$  be the output feature vector;  $i, j$  be the indices of points; and  $\mathcal{N}(i)$  be the set of neighbor points of  $\mathbf{x}_i$ . We use the same point transformer layer as in the prior work:

$$\mathbf{y}_i = \sum_{\mathbf{x}_j \in \mathcal{N}(i)} \text{softmax}(w(\mathbf{q}_i - \mathbf{k}_j + \delta)) \odot (\mathbf{v}_j + \delta), \quad (11)$$

where  $\mathbf{q}, \mathbf{k}, \mathbf{v}$  are the outputs of three MLPs with input  $\mathbf{x}$  (i.e.,  $\mathbf{q} = q(x)$ ,  $\mathbf{k} = k(x)$ ,  $\mathbf{v} = v(x)$ );  $w$  is another MLP;  $\odot$  is pointwise multiplication; and  $\delta = \theta(\mathbf{p}_i - \mathbf{p}_j)$  is the relative position embedding where  $\theta$  is an MLP. In order to make Equation (11) equivariant, we need to make all the MLPs  $q, k, v, \theta, w$  equivariant. Moreover, we need to ensure that the other operations in Equation (11) do not break the equivariance. To achieve this, we use the regular representation  $\rho_{\text{reg}}$  as the output types of all MLPs, since the regular representation is compatible with all pointwise operations such as addition, subtraction, and pointwise multiplication. The full architecture is shown in Figure 4.

One advantage of our equivariant point transformer is that the relative position embedding  $\delta = \theta(\mathbf{p}_i - \mathbf{p}_j)$  makes the encoder translationally invariant, that is, translating



**Figure 5.** Translation invariance via predicting the action in the gripper translation frame.

the entire point cloud will not change the observation embedding. We further translate the action from the world frame to the gripper frame, making the entire policy  $T(3)$ -invariant (recall that  $T(3)$  is the group of 3D translations). Specifically, when the point cloud and the action are translated by the same amount, both the observation embedding and the action in the gripper’s translation frame are invariant, as shown in Figure 5. As a result, the point cloud version of our method has  $SO(2) \times T(3)$  symmetry rather than just  $SO(2)$ . (A similar process called “Relative Trajectory” is proposed in Chi et al. (2024).) This  $SO(2) \times T(3)$  symmetry allows our model to generalize across both position and orientation changes in the environment, a capability that neither image-based nor voxel-based methods can fully achieve.

In our experiments in Sections Standard Baseline Comparison and Ablation Study, this comprehensive equivariance contributes significantly to the superior performance of our point cloud-based approach, particularly in environments with high variability in object positions and orientations.

We also replace the pointwise equivariant processing in the U-Net (Section Implementation of Equivariant Diffusion Policy) with a Frame Averaging interface (Puny et al., 2022). Specifically, let  $U$  be the U-Net,  $k$  be the denoising step,  $e_a$  and  $e_o$  be the action and observation embeddings (in the form of regular representations), and  $G$  be a symmetry group (e.g.,  $C_8$ ). The noise embedding  $z$  is calculated as

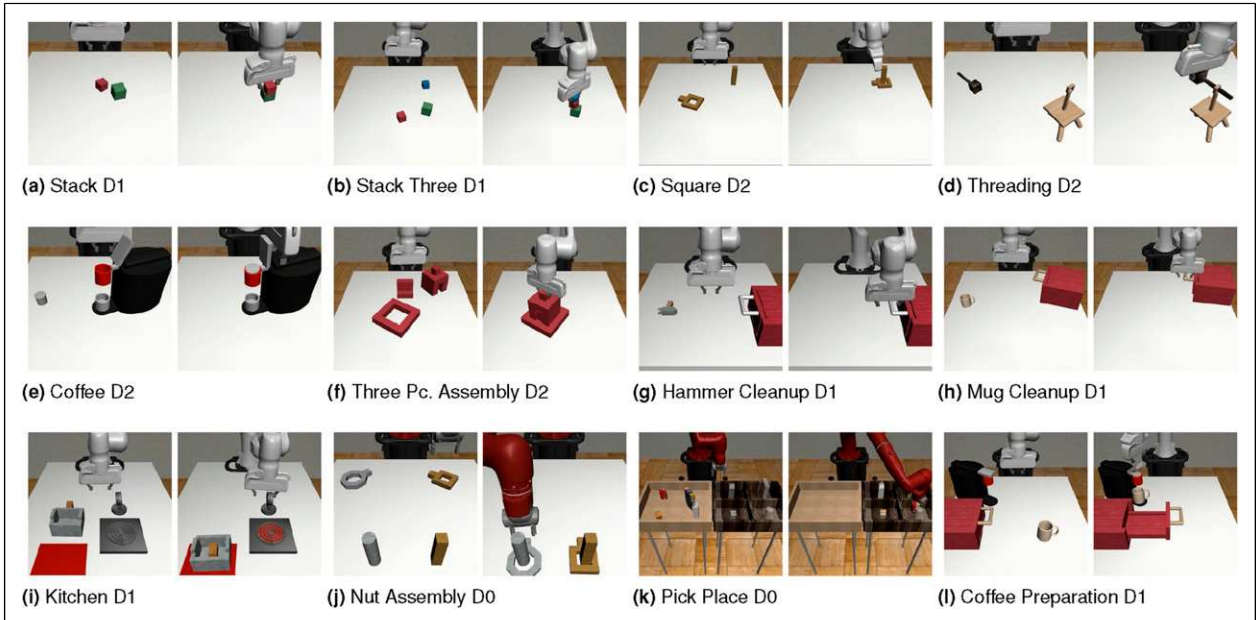
$$z = \frac{1}{|G|} \sum_{g \in G} \rho_{\text{reg}}(g) U(\rho_{\text{reg}}^{-1}(g) e_o, \rho_{\text{reg}}^{-1}(g) e_a, k).$$

Compared with the pointwise equivariant processing, Frame Averaging enables the U-Net to access the entire regular representation (rather than one element at a time) while also ensuring the equivariance. See Puny et al. (2022) for details.

## Simulation experiments

### Experimental settings

We first evaluate our Equivariant Diffusion Policy (Equi-Diff) with image (Im), voxel (Vo), or point cloud (PC) input



**Figure 6.** The experimental environments from MimicGen (Mandlekar et al., 2023). The left image in each subfigure shows the initial state of the environment; the right image shows the goal state. (a) Stack D1, (b) Stack three D1, (c) Square D2, (d) Threading D2, (e) Coffee D2, (f) Three Pc, Assembly D2, (g) Hammer cleanup D1, (h) Mug cleanup D1, (i) Kitchen D1, (j) Nut assembly D0, (k) Pick place D0, and (l) Coffee preparation D1.

**Table 1.** The performance of our method in absolute control compared with the baselines in simulation.

Method	Stack D1				Stack Three D1				Square D2				Threading D2			
	Obs	100	200	1000	100	200	1000	1000	100	200	1000	1000	100	200	1000	1000
EquiDiff (PC)	PCD	98 (+22)	100 (+3)	100 (=)	90 (+52)	96 (+24)	97 (+3)	97 (+3)	67 (+59)	81 (+62)	75 (+26)	55 (+38)	60 (+25)	59 (=)		
EquiDiff (Vo)	Voxel	99 (+23)	100 (+3)	100 (=)	75 (+37)	91 (+19)	91 (-3)	91 (-3)	39 (+31)	48 (+29)	63 (+14)	39 (+22)	53 (+18)	55 (-4)		
EquiDiff (Im)	RGB	93 (+17)	100 (+3)	100 (=)	55 (+17)	77 (+5)	96 (+2)	96 (+2)	25 (+17)	41 (+22)	60 (+11)	22 (+5)	40 (+5)	59 (=)		
DiffPo-C	RGB	76	97	100	38	72	94	94	8	19	46	17	35	59		
DiffPo-T	RGB	51	83	99	17	41	84	84	5	11	45	11	18	41		
DP3	PCD	69	87	99	7	23	65	65	7	6	19	12	23	40		
ACT	RGB	35	73	96	6	37	78	78	6	18	49	10	21	35		
Coffee D2																
				Three Pc. assembly D2				Hammer cleanup D1				Mug cleanup D1				
Method	Obs	100	200	1000	100	200	1000	1000	100	200	1000	1000	100	200	1000	1000
EquiDiff (PC)	PCD	78 (+31)	74 (+8)	75 (-4)	66 (+62)	72 (+66)	69 (+26)	69 (+26)	81 (+27)	81 (+10)	82 (-5)	65 (+22)	71 (+12)	71 (+6)		
EquiDiff (Vo)	Voxel	65 (+18)	73 (+7)	76 (-3)	37 (+33)	58 (+52)	71 (+28)	71 (+28)	70 (+16)	66 (-5)	73 (-14)	53 (+10)	65 (+6)	68 (+3)		
EquiDiff (Im)	RGB	60 (+13)	79 (+13)	76 (-3)	15 (+11)	39 (+33)	69 (+26)	69 (+26)	65 (+11)	63 (-8)	77 (-10)	49 (+6)	64 (+5)	67 (+2)		
DiffPo-C	RGB	44	66	79	4	6	30	30	52	59	73	43	59	65		
DiffPo-T	RGB	47	61	75	1	4	43	43	48	60	76	30	43	63		
DP3	PCD	34	45	69	0	1	3	3	54	71	87	21	33	53		
ACT	RGB	19	33	64	0	3	24	24	38	54	71	23	31	56		
Kitchen D1																
				Nut assembly D0				Pick place D0				Coffee preparation D1				
Method	Obs	100	200	1000	100	200	1000	1000	100	200	1000	1000	100	200	1000	1000
EquiDiff (PC)	PCD	84 (+17)	86 (+1)	86 (-5)	91 (+36)	94 (+26)	95 (+11)	95 (+11)	59 (+24)	79 (+14)	90 (+7)	84 (+19)	85 (+23)	88 (+12)		
EquiDiff (Vo)	Voxel	85 (+18)	89 (+4)	88 (-3)	67 (+12)	77 (+9)	83 (-1)	83 (-1)	58 (+23)	68 (+3)	82 (-1)	80 (+15)	83 (+21)	85 (+9)		
EquiDiff (Im)	RGB	67 (=)	77 (-8)	81 (-10)	74 (+19)	85 (+17)	94 (+10)	94 (+10)	42 (+7)	74 (+9)	92 (+9)	77 (+12)	83 (+21)	85 (+9)		
DiffPo-C	RGB	67	85	87	55	68	83	83	35	65	83	65	62	58		
DiffPo-T	RGB	54	75	81	31	32	46	46	15	37	50	38	51	76		
DP3	PCD	45	71	91	16	24	58	58	12	15	34	10	22	63		
ACT	RGB	37	61	87	42	64	84	84	7	17	50	32	46	65		

We experiment with 100, 200, and 1000 demos in each environment and report the maximum task success rate among 50 evaluations throughout training. Results averaged over three seeds. Number in parentheses shows the difference between our method and the best baseline (with decrement in italic format). Bold performance indicates the best, bold difference is greater than 10%.

**Table 2.** The average performance over 12 tasks of Equivariant Diffusion Policy compared with baselines.

Method	Ctrl	Average over 12 environments		
		100	200	1000
EquiDiff (PC)	Abs	76.5 ( <b>+34.5</b> )	81.6 ( <b>+23.8</b> )	82.3 ( <b>+10.9</b> )
EquiDiff (Vo)		63.9 ( <b>+21.9</b> )	72.6 ( <b>+14.8</b> )	77.9 (+6.5)
EquiDiff (Im)		53.7 ( <b>+11.7</b> )	68.5 ( <b>+10.7</b> )	79.7 (+8.3)
DiffPo-C		42.0	57.8	71.4
DiffPo-T		29.0	43.0	64.9
DP3		23.9	35.1	56.8
ACT		21.3	38.2	63.3
EquiDiff (Vo)	Rel	48.8 ( <b>+15.5</b> )	58.0 ( <b>+10.7</b> )	70.2 ( $-0.1$ )
EquiDiff (Im)		35.4 (+2.1)	50.4 (+3.1)	74.0 (+3.7)
DiffPo-C		33.3	47.3	63.2
BC RNN		22.9	41.2	70.3

on 12 manipulation tasks from MimicGen (Mandlekar et al., 2023) (Figure 6). The RGB observation is an agent-view image and an eye-in-hand image with a size of  $3 \times 84 \times 84$ . The voxel grid observation has a size of  $4 \times 64 \times 64 \times 64$  where the first channel is binary occupancy and the remaining three channels are RGB. The point cloud observation has a size of  $1024 \times 6$  (i.e., xyzrgb). The point cloud only contains points above the table, as suggested in Ze et al. (2024). All tasks have a full 6 DoF SE (3)

action space. We define the rotation of the observation as a point cloud rotation, a voxel grid rotation, or an image rotation. Notice that in the image version of our method, there is a mismatch between the rotation of the agent-view image and the rotation of the ground-truth state since the agent view is not orthogonally top-down. Although top-down observations could be captured, we use the observation settings in the published dataset from MimicGen (Mandlekar et al., 2023) to demonstrate the generalizability of our method (Notice that

**Table 3.** Same experiment as Table 1 in relative control.

Method	Obs	Stack D1			Stack Three D1			Square D2			Threading D2		
		100	200	1000	100	200	1000	100	200	1000	100	200	1000
EquiDiff (Vo)	Voxel	<b>95 (+14)</b>	<b>100 (+5)</b>	<b>100 (=)</b>	<b>59 (+33)</b>	<b>76 (+24)</b>	83 ( $-9$ )	<b>25 (+17)</b>	<b>35 (+14)</b>	52 ( $-7$ )	<b>33 (+20)</b>	<b>39 (+13)</b>	46 ( $-1$ )
EquiDiff (Im)	RGB	75 ( $-6$ )	96 (+1)	<b>100 (=)</b>	25 ( $-1$ )	63 ( <b>+11</b> )	<b>92 (=)</b>	11 (+3)	21 (=)	48 ( $-11$ )	11 ( $-2$ )	22 ( $-4$ )	<b>49 (+2)</b>
DiffPo-C	RGB	81	93	99	26	52	86	6	13	37	13	26	40
BC RNN	RGB	59	95	<b>100</b>	12	48	<b>92</b>	8	21	<b>59</b>	7	13	47
Method	Obs	Coffee D2			Three Pc. Assembly D2			Hammer Cleanup D1			Mug Cleanup D1		
		100	200	1000	100	200	1000	100	200	1000	100	200	1000
EquiDiff (Vo)	Voxel	<b>55 (+12)</b>	<b>59 (+7)</b>	64 ( $-12$ )	<b>5 (+3)</b>	<b>5 (=)</b>	55 ( <b>+28</b> )	<b>64 (+21)</b>	<b>62 (+8)</b>	67 ( $-5$ )	<b>39 (+14)</b>	<b>43 (+4)</b>	62 ( $-5$ )
EquiDiff (Im)	RGB	41 ( $-2$ )	<b>59 (+7)</b>	66 ( $-10$ )	1 ( $-1$ )	<b>5 (=)</b>	<b>59 (+32)</b>	49 (+6)	52 ( $-2$ )	69 ( $-3$ )	29 (+4)	36 ( $-3$ )	65 ( $-2$ )
DiffPo-C	RGB	43	51	67	2	2	20	43	54	65	25	39	55
BC RNN	RGB	37	52	<b>76</b>	0	<b>5</b>	27	32	43	<b>72</b>	19	39	<b>67</b>
Method	Obs	Kitchen D1			Nut assembly D0			Pick place D0			Coffee preparation D1		
		100	200	1000	100	200	1000	100	200	1000	100	200	1000
EquiDiff (Vo)	Voxel	<b>69 (+27)</b>	<b>83 (+19)</b>	<b>89 (+8)</b>	<b>53 (+11)</b>	<b>65 (+3)</b>	72 ( $-13$ )	<b>40 (+5)</b>	58 ( $-1$ )	79 ( $-3$ )	48 (+6)	<b>71 (+18)</b>	73 ( <b>+12</b> )
EquiDiff (Im)	RGB	61 ( <b>+19</b> )	72 (+8)	83 (+2)	44 (+2)	<b>65 (+3)</b>	<b>87 (+2)</b>	29 ( $-6$ )	55 ( $-4$ )	<b>91 (+9)</b>	<b>49 (+7)</b>	59 (+6)	<b>79 (+18)</b>
DiffPo-C	RGB	42	64	81	42	62	75	35	<b>59</b>	82	42	53	51
BC RNN	RGB	31	47	81	35	58	85	21	41	77	14	32	61

the prior work (Wang et al., 2023) has demonstrated that the equivariant CNN is still able to capture symmetry in such a scenario.). On the other hand, the point cloud and the voxel versions eliminate this symmetry mismatch as the rotation of the point cloud or the voxel grid aligns with the rotation of the ground-truth state. To better leverage the equivariance, we also add a rotation augmentation in the point cloud and voxel versions of our method following our analysis in Section Method.

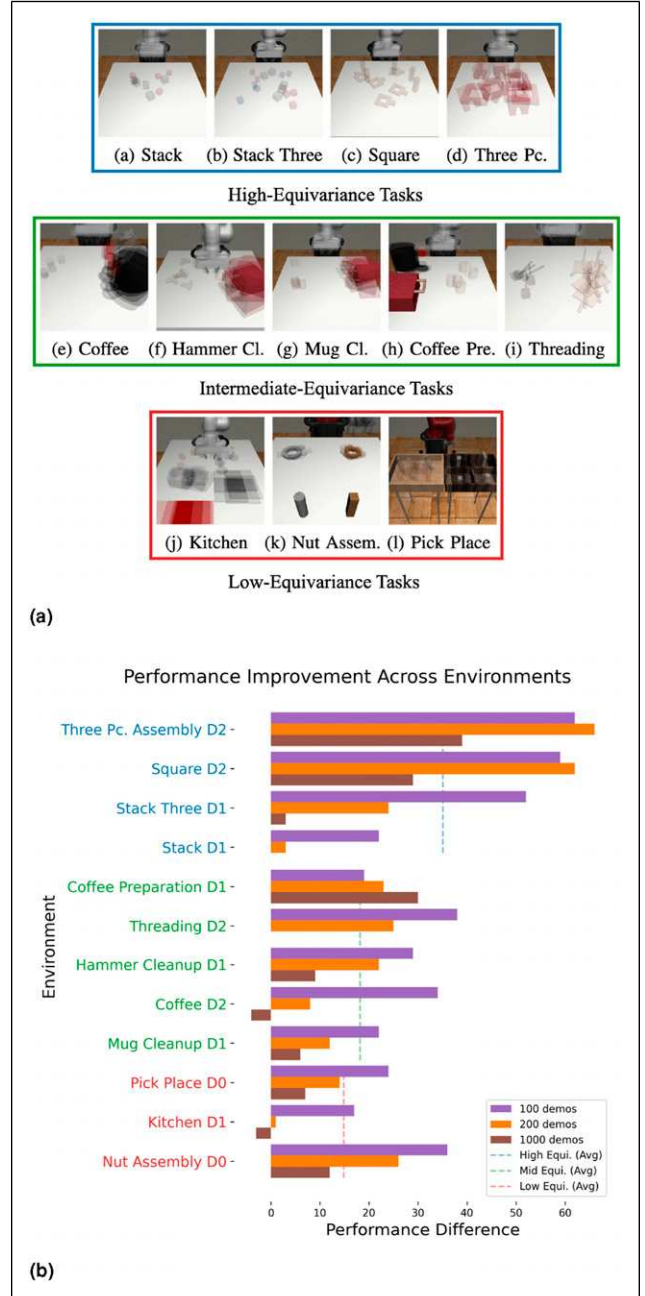
### Standard baseline comparison

We evaluate our Equivariant Diffusion Policy for both absolute pose control and relative-pose control. We compare our method with the following baselines:

1. DiffPo-C: the original diffusion policy (Chi et al., 2023) trained with the 1D Temporal U-Net (Janner et al., 2022). Notice that the baseline shares the same U-Net architecture as our method, but it does not have any equivariant structure.
2. DiffPo-T: same as above, but trained with a transformer.
3. DP3: the 3D diffusion policy (Ze et al., 2024) trained with a point net encoder.
4. ACT: the Action Chunking Transformer (Zhao et al., 2023) trained as a conditional VAE.
5. BC RNN: a recurrent architecture from Mandlekar et al. (2022).

Notice that the voxel version of our method and DP3 utilizes the 3D inputs constructed from four cameras, while the image version of our method and the other baselines directly use the RGB images from two cameras. As our main baseline, we evaluate DiffPo-C in both absolute and relative-pose control. We evaluate the other baselines in the same control mode as in the original work (absolute for DiffPo-T, DP3 and ACT, and relative for BC RNN). See Appendix Simulation Environments and Training Detail for the details.

**Results.** Table 1 shows the experimental results of different methods using absolute control in terms of the maximum success rate among 50 evaluations throughout the training. Our Equivariant Diffusion Policy with point cloud input (EquiDiff (PC)) consistently achieves the best overall performance, significantly outperforming all baselines in most environments. The performance advantage is particularly pronounced in the low-data regime (i.e., with 100 or 200 demonstrations). Specifically, as shown in Table 2, EquiDiff (PC) trained with just 100 demos achieves an average success rate of 76.5% across all environments, outperforming the best baseline with the same amount of data by 34.5%. Remarkably, this performance even exceeds that of all baselines trained with 1000 demos, clearly



**Figure 7.** (a) The three task groups are based on the level of equivariance and their initial object distribution. Images were generated by taking the average of five random initialization states. (b) The performance improvement of our Equivariant Diffusion Policy (PC) compared with the original diffusion policy in absolute pose control. Blue environments are high-equivariance tasks; green environments are intermediate-equivariance tasks; red environments are low-equivariance tasks.

demonstrating the exceptional sample efficiency of our approach. Compared to our conference version (EquiDiff (Vo)), the point cloud model delivers an additional 12.6% performance improvement, validating the benefits of incorporating both rotational and translational symmetries.

**Table 4.** The performance of our method compared with two sample-efficient baselines in simulation.

Method	Obs	Mean	Stack D1	Stack three D1	Square D2	Threading D2	Coffee D2	Three Pc. D2
EquiDiff (PC)	PCD	<b>76.5</b>	98	<b>90</b>	<b>67</b>	<b>55</b>	<b>78</b>	<b>66</b>
EquiDiff (Vo)	Voxel	63.9	99	75	39	39	65	37
EquiDiff (Im)	RGB	53.7	93	55	25	22	60	15
RISE	PCD	56.5	<b>100</b>	<b>87</b>	34	38	48	45
ISP	RGB	62.9	98	81	56	19	67	39

Method	Obs	Hammer Cl. D1	Mug Cl. D1	Kitchen D1	Nut Asse. D0	Pick Place D0	Coffee prep. D1
EquiDiff (PC)	PCD	<b>81</b>	<b>65</b>	84	<b>91</b>	<b>59</b>	<b>84</b>
EquiDiff (Vo)	Voxel	70	53	<b>85</b>	67	58	80
EquiDiff (Im)	RGB	65	49	67	74	42	77
RISE	PCD	73	54	71	44	31	53
ISP	RGB	73	54	64	85	56	63

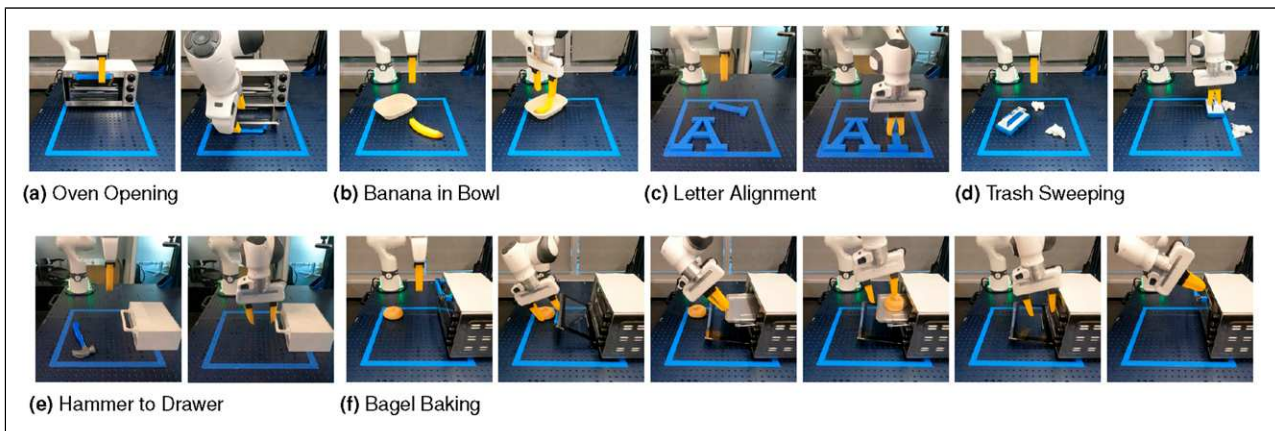
We experiment with 100 demos in each environment and report the maximum task success rate among 50 evaluations throughout training. Results averaged over three seeds. Bold indicates best performance.

Table 3 shows the results of different methods using relative control. Our method with voxel input achieves the best performance, while our method with RGB input is only marginally better than the baselines.

### Improvement with different levels of equivariance

We further analyze the performance improvement of our method when the tasks have different levels of equivariance. Since equivariant models generalize automatically across different object poses, equivariance should hypothetically be more useful when there is greater variance in the distribution of initial object poses. We

qualitatively group the tasks into three levels: (1) high-equivariance tasks where the poses of the objects are initialized randomly within the workspace; (2) intermediate-equivariance tasks where each object is initialized in a certain range, but with some randomness inside the range; (3) low-equivariance tasks where there is no randomness for the position and/or orientation of certain objects. Figure 7(a) shows the three task groups. We show the performance improvement of our Equivariant Diffusion Policy with point cloud in absolute pose control compared with the standard diffusion policy in Figure 7(b). Generally, the high-equivariance tasks benefit more from injecting symmetry in the network architecture. Moreover, our method’s strong performance



**Figure 8.** The real-world environments. The left image of each subfigure shows the initial state of the environment; the right image shows the goal state. See Appendix Real-Robot Environment Details for a detailed task description. (a) Oven opening, (b) Banana in bowl, (c) Letter alignment, (d) Trash sweeping, (e) Hammer to drawer, and (f) Bagel baking.

**Table 5.** The performance of our EquiDiff (PC) compared with different ablated variations.

Ablation	Average	Stack		Square	Threading	Coffee	Three Pc.	Hammer C.	Mug C.	Kitchen	Nut Asse.	Pick Place	Coffee Prep.
		Stack	three										
EquiDiff (PC)	76.5	98	90	67	55	78	66	81	65	84	91	59	84
No FA	74.8 (-1.7)	96	94	63	50	71	69	82	59	85	92	51	85
No TSFM	60.8 (-15.7)	100	93	30	35	73	5	85	58	85	57	49	59
No trans equi	59.0 (-17.5)	95	65	44	29	48	49	61	57	71	84	34	71
No rot equi	37.2 (-39.3)	97	41	13	21	51	2	67	48	61	13	17	15

We experiment with 100 demos in each environment. Results averaged over three seeds.

in the intermediate and low-equivariance tasks indicates its robustness and generalizability, as the model’s symmetry is helpful even when the task is partially symmetric.

### Sample-efficient baseline comparison

In this section, we evaluate our method against two sample-efficient baselines:

1. RISE: a diffusion transformer architecture with a sparse 3D encoder, taking point clouds as inputs (Wang et al., 2024a).
2. ISP: an equivariant policy using spherical projection to project the input eye-in-hand RGB image onto a sphere for equivariant reasoning (Hu et al., 2025).

We experiment with 100 demos (i.e., the low-data regime) in this evaluation. As shown in Table 4, although ISP achieves a comparable performance as EquiDiff (Vo), and RISE slightly outperforms EquiDiff (Im), EquiDiff (PC) remains the best performing method. Across all 12 tasks with 100 demos, EquiDiff (PC) outperforms ISP and RISE by 13.6–20.0 absolute points on average (76.5 vs 62.9 and 56.5, respectively).

### Ablation study

We perform an ablation study to understand the importance of different components of EquiDiff (PC). Specifically, we consider the following variations:

1. EquiDiff (PC): the complete model.
2. No FA: replaces the Frame Averaging with the pointwise equivariant processing.
3. No TSFM: replaces the equivariant point transformer with a simple equivariant point net.
4. No Trans Equi: does not translate the action to the gripper frame (thus the policy does not have the translation symmetry described at the end of

Section Equivariant Point Transformer and Translation Symmetry).

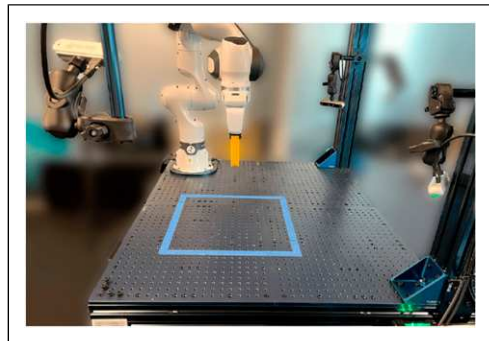
5. No Rot Equi: removes all the SO (2)-equivariant structure in the model. This is essentially DP3 but with translation symmetry.

As shown in Table 5, removing the rotational symmetry makes the most significant negative impact on the model, decreasing the average performance by nearly 40%. Removing the translation symmetry and the point transformer architecture decreases the overall performance by 17.5% and 15.7%, respectively. This result demonstrates the importance of all the key pieces of our model: rotational and translational symmetry, and the equivariant point transformer architecture.

## Real-robot experiment

### Experimental settings

In this section, we evaluate our method on a real robot system containing a Franka Emika robot arm (Haddadin et al., 2022) equipped with a pair of fin-ray (Crooks et al., 2016) fingers and three Intel Realsense (Keselman et al., 2017) D455 cameras. Demonstrations were



**Figure 9.** Our real-robot platform contains a Franka Emika robot arm equipped with a pair of fin-ray fingers, and three Intel Realsense D455 cameras.

**Table 6.** Performance of Equivariant Diffusion Policy in real-world robot experiments.

	Oven opening	Banana in bowl	Letter alignment	Trash sweeping	Hammer to drawer	Bagel baking
# Demos	20	40	40	40	60	58
EquiDiff (Vo)	95% (19/20)	95% (19/20)	95% (19/20)	90% (18/20)	85% (17/20)	80% (16/20)
DiffPo-C (Vo)	60% (12/20)	30% (6/20)	0% (0/20)	5% (1/20)	5% (1/20)	10% (2/20)

gathered by an operator using a 6DoF 3DConnexion mouse. Observations and demonstration actions were recorded at 5 Hz. Similarly to prior work (Chi et al., 2023), we use DDIM (Song et al., 2020) in this experiment to reduce the number of denoising steps to 16.

### EquiDiff with voxel input

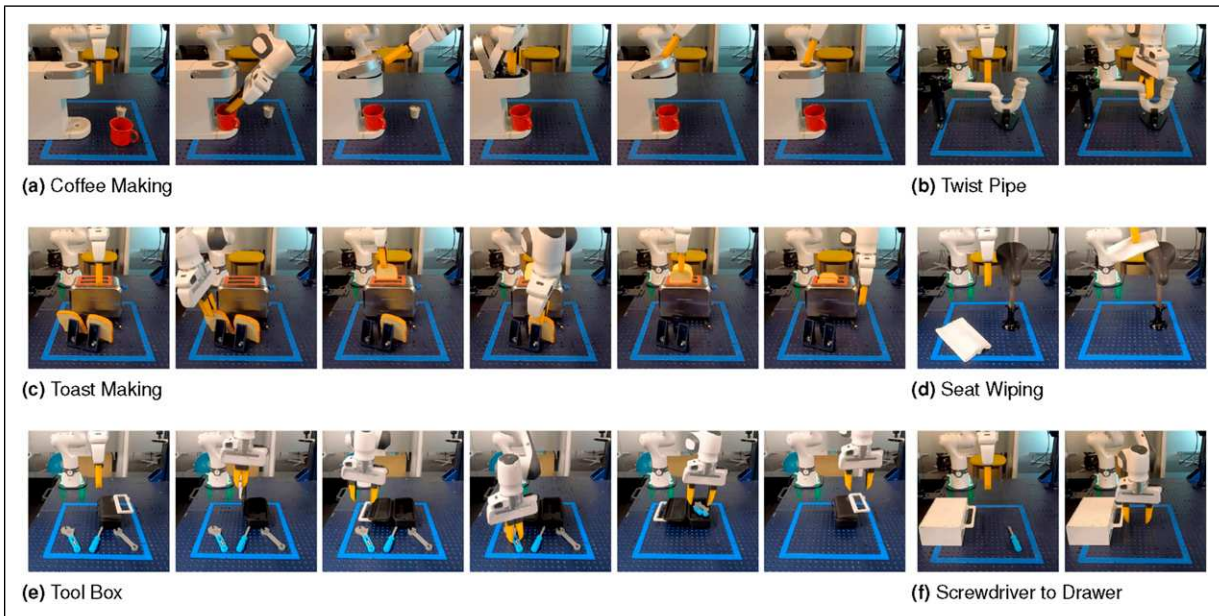
We first compare our Equivariant Diffusion Policy with voxel input against a baseline Diffusion Policy, which uses the same voxel grid as the vision input and employs a non-equivariant 3D convolutional encoder with approximately the same number of trainable parameters as ours. As we show in the ablation study (Appendix Ablation Study of EquiDiff (Vo)), this baseline works better than the original diffusion policy with image input.

Figure 8 shows the six tasks in this experiment. Figure 9 shows the robot system.

**Results.** We evaluate the trained models over 20 test trials for each task. The results are shown in Table 6. Our Equivariant Diffusion Policy can solve those tasks with only 20 to 60 demonstrations. Notably, our method achieves an 80% success rate in bagel baking, where the failures were all due to the joint limits of the robot. In comparison, the baseline performs poorly in all six tasks.

### EquiDiff with point cloud input

This experiment evaluates our Equivariant Diffusion Policy with point cloud or voxel input in more advanced



**Figure 10.** The more advanced real-world environments. (a) Coffee making, (b) Twist pipe, (c) Toast making, (d) Seat wiping, (e) Tool box, and (f) Screwdriver to drawer.

**Table 7.** Performance of Equivariant Diffusion Policy in more advanced real-world environments.

	Twist pipe	Seat wiping	Screwdriver to drawer	Trash sweeping	Toast making	Bagel baking	Tool box	Coffee making
# Demos	20	20	40	50	50	60	99	159
EquiDiff (PC)	100% (20/20)	95% (19/20)	90% (18/20)	100% (20/20)	70% (14/20)	85% (17/20)	85% (17/20)	70% (14/20)
EquiDiff (Vo)	100% (20/20)	70% (14/20)	85% (17/20)	80% (18/20)	65% (13/20)	85% (17/20)	55% (11/20)	55% (11/20)

tasks. We consider the Bagel Baking and Trash Sweeping tasks in Figure 8, as well as six new tasks in Figure 10. As is shown in Table 7, EquiDiff (PC) can solve those more advanced tasks with significantly higher success rates compared to the voxel version, which aligns with our simulation experiment.

### Generalization experiment

In this experiment, we evaluate the generalizability of our Equivariant Diffusion Policy to unseen object poses. We conduct this evaluation in the Bagel Baking experiment in the real world, where the oven is initialized in three different poses during training (Figure 11(a)). At test time, we rotate the oven to eight different, unseen poses (Figure 11(b)). We found that the learned policy can zero-shot generalize to these unseen rotations, with the exception of the scenario where the oven is rotated to the bottom-right corner. In this case, the policy is constrained by the robot’s joint limits. Specifically, the policy was able to open the oven and pull out the tray, but when

picking up the bagel, although the policy could generate good gripper poses, the actions were infeasible for the robot due to joint limits. This generalization demonstrates the power of the equivariant structure in our policy.

### Conclusion

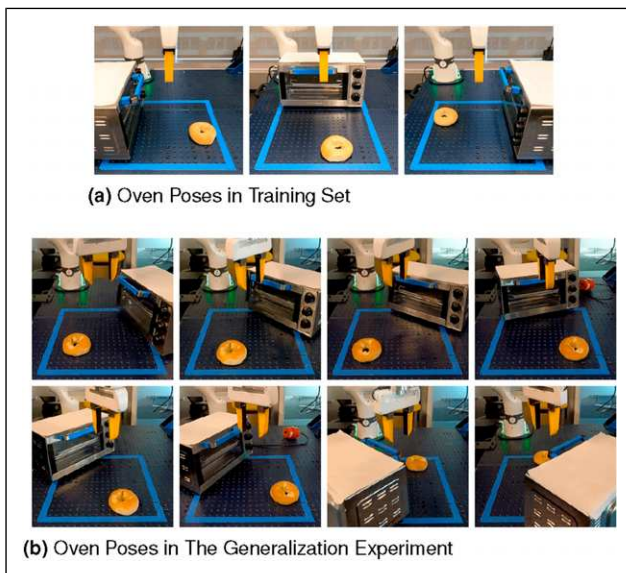
This paper studies leveraging symmetries in visuomotor policy learning. We propose the novel Equivariant Diffusion Policy method and provide a theoretical analysis identifying the conditions under which the diffusion policies are equivariant. We extend our previous work by incorporating both SO (2) rotational and  $T$  (3) translational symmetries through our Equivariant Point Transformer architecture, demonstrating a general framework for using these symmetries in 6DoF control for robotic manipulation. Our comprehensive evaluation in both simulation and real-world environments shows that our extended method substantially outperforms both our conference version and the baseline Diffusion Policy, achieving significantly higher success rates with notably fewer demonstrations.

### Training stability

We trained our equivariant models with the *same* optimizer, noise schedule, and U-Net/transformer depth as their non-equivariant counterparts, with no per-task tuning. Across seeds and data regimes, we did not observe gradient vanishing/exploding or mode-collapse behavior. In Appendix Performance under Hyper-Parameter Changes, varying batch size, warm-up, weight decay, and learning rate left Stack D1 at 100% success, indicating our equivariant layers are plug-and-play replacements that preserve Diffusion Policy training dynamics while improving data efficiency and generalization.

### Symmetry breakings

One limitation of this work is the partial utilization of equivariance due to symmetry mismatch in the vision system. Even with voxel or point cloud inputs, factors such as occasional arm visibility in the observation and camera noise can introduce imperfect symmetric transformations, leading to an “extrinsic equivariance” (Wang et al., 2023) setting where the symmetry in the architecture transforms the data out of distribution, and the benefit of equivariance degrades. Future



**Figure 11.** (a) The initial oven poses in the training set. (b) The oven poses in the generalization experiment. Those poses are unseen during training. In both training and testing, the pose of the bagel is random.

work could address this by designing a vision system that avoids such symmetry corruption. Additionally, “incorrect equivariance,” as shown in prior work (Wang et al., 2024c), may harm performance when the model’s symmetry conflicts with the demonstrations. For example, reachability and kinematic constraints of the robot arm are not always symmetric, potentially yielding infeasible symmetric transformations of demonstrated actions. Another example is tasks requiring actions tied to the world frame without visual cues (e.g., “push object to the left”); applying a symmetric transformation could produce the opposite behavior.

It is worth noticing that although both “extrinsic equivariance” and “incorrect equivariance” can be viewed as forms of symmetry breaking, their effects differ fundamentally. Intuitively, extrinsic equivariance can shift inputs slightly out of distribution, but the similarity between the symmetrically transformed data and the in-distribution data can help the network learn the true decision boundary. In such cases, equivariant models often remain beneficial, and the good performance of our method in the intermediate- and low-equivariance tasks in Figure 7(a) confirms this. In contrast, incorrect equivariance places the model in direct conflict with the ground-truth mapping and is therefore harmful and should be avoided. See Wang et al. (2023, 2024c) for further discussion.

### Other limitations

While the theory in Section SO(2) Representation on 6DoF Action is not restricted to diffusion policies and in principle applies to other policy-learning pipelines, we have not demonstrated this empirically. Given the strong performance of BC-RNN with relative-pose control in Table 1, an equivariant BC-RNN is a promising direction. Finally, extending our approach to other robotic settings, such as navigation, locomotion, and mobile manipulation, remains important future works.

### Acknowledgments

The authors would like to thank Dr Osman Dogan Yirmibesoglu for the design of the fin-ray gripper fingers, Dr Andy Park for building the teleop system for data collection, Emmanuel Panov for collecting demonstration data in the robot experiment, Dr Thomas Weng for the proofreading of the paper, and Dr Cheng Chi for the helpful discussion.

### ORCID iDs

Dian Wang  <https://orcid.org/0000-0002-0546-0175>

Haojie Huang  <https://orcid.org/0000-0001-8737-7959>

### Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Part of the work was done when Dian Wang was an intern at the Robotics and AI Institute. This work is supported in part by NSF 1750649, NSF 2107256, NSF 2314182, NSF 2134178, NSF 2409351,

2442658, and NASA 80NSSC19K1474. Dian Wang is supported in part by the JPMorgan Chase PhD fellowship.

### Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### References

- Brehmer J, Bose J, De Haan P, et al. (2023) EDGI: equivariant diffusion for planning with embodied agents. ArXiv Preprint arXiv:2303.12410.
- Cesa G, Lang L and Weiler M (2021) A program to build  $E(N)$ -equivariant steerable CNNs. In: International conference on learning representations, Virtual Conference, 3–7 May 2021.
- Chen K, Chen X, Yu Z, et al. (2023) Equidiff: a conditional equivariant diffusion model for trajectory prediction. In: 2023 IEEE 26th international conference on intelligent transportation systems (ITSC), Bilbao, Spain, 24–28 September 2023, pp. 746–751. IEEE.
- Chi C, Feng S, Du Y, et al. (2023) Diffusion policy: visuomotor policy learning via action diffusion. In: Proceedings of robotics: science and systems (RSS), Daegu, Republic of Korea, 10–14 July 2023.
- Chi C, Xu Z, Pan C, et al. (2024) Universal manipulation interface: in-The-wild robot teaching without in-the-wild robots. In: Proceedings of robotics: science and systems (RSS), Delft, Netherlands, 15–19 July 2024.
- Crooks W, Vukasin G, O’Sullivan M, et al. (2016) Fin ray® effect inspired soft robotic gripper: from the roboSoft grand challenge toward optimization. *Frontiers in Robotics and AI* 3: 70.
- Du Y and Mordatch I (2019) Implicit generation and generalization in energy-based models. In: Advances in neural information processing systems, Vancouver, BC, 8–14 December 2019.
- Eisner B, Yang Y, Davchev T, et al. (2024) Deep SE(3)-equivariant geometric reasoning for precise placement tasks. In: The twelfth international conference on learning representations, Vienna, Austria, 7–11 May 2024. URL: <https://openreview.net/forum?id=2inBuWTyL2>
- Florence P, Manuelli L and Tedrake R (2019) Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters* 5(2): 492–499.
- Florence P, Lynch C, Zeng A, et al. (2021) Implicit behavioral cloning. In: Conference on robot learning (CoRL), London, UK, 8–11 November 2021.
- Gao C, Xue Z, Deng S, et al. (2024) RiEMann: near real-time SE(3)-equivariant robot manipulation without point cloud segmentation. In: 8th annual conference on robot learning, Munich, Germany, 6–9 November 2024. URL: <https://openreview.net/forum?id=eJHy0AF5TO>
- Grathwohl W, Wang KC, Jacobsen JH, et al. (2020) Learning the stein discrepancy for training and evaluating energy-based models without sampling. In: International conference on machine learning, Virtual Meeting, 13–18 July 2020, pp. 3732–3747. PMLR.

- Guan J, Qian WW, Peng X, et al. (2023) 3D equivariant diffusion for target-aware molecule generation and affinity prediction. In: The eleventh international conference on learning representations, Kigali, Rwanda, 1–5 May 2023.
- Gupta A, Kumar V, Lynch C, et al. (2020) Relay policy learning: solving long-horizon tasks via imitation and reinforcement learning. In: Conference on robot learning, Virtual Meeting, 16–18 November 2020, pp. 1025–1037. PMLR.
- Haddadin S, Parusel S, Johannsmeier L, et al. (2022) The Franka Emika robot: a reference platform for robotics research and education. *IEEE Robotics and Automation Magazine* 29(2): 46–64.
- He K, Zhang X, Ren S, et al. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, 27–30 June 2016, pp. 770–778.
- Ho J, Jain A and Abbeel P (2020) Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33: 6840–6851.
- Hoogeboom E, Satorras VG, Vignac C, et al. (2022) Equivariant diffusion for molecule generation in 3D. In: International conference on machine learning, Baltimore, MD, 17–23 July 2022, pp. 8867–8887. PMLR.
- Hu B, Zhu X, Wang D, et al. (2024) Orbitgrasp: Se (3)-equivariant grasp learning. In: 8th annual conference on robot learning, Munich, Germany, 6–9 November 2024.
- Hu B, Wang D, Klee D, et al. (2025) 3d equivariant visuomotor policy learning via spherical projection. In: The thirty-ninth annual conference on neural information processing systems, San Diego, CA, 2–7 December 2025. URL: <https://openreview.net/forum?id=kXJd4JxF34>
- Huang H, Wang D, Walters R, et al. (2022) Equivariant transporter network. In: Robotics: science and systems, New York City, NY, 27 June 2022–1 July 2022.
- Huang H, Wang D, Tangri A, et al. (2023a) Leveraging symmetries in pick and place. *The International Journal of Robotics Research* 43(4): 550–571.
- Huang H, Wang D, Zhu X, et al. (2023b) Edge grasp network: a graph-based SE(3)-invariant approach to grasp detection. In: International conference on robotics and automation (ICRA), London, UK, 29 May 2023–2 June 2023.
- Huang H, Howell OL, Wang D, et al. (2024a) Fourier transporter: bi-equivariant robotic manipulation in 3d. In: The twelfth international conference on learning representations, Vienna, Austria, 7–11 May 2024. URL: <https://openreview.net/forum?id=UulwvAU1W0>
- Huang H, Schmeckpeper K, Wang D, et al. (2024b) Imagination policy: using generative point cloud models for learning manipulation policies. In: 8th annual conference on robot learning, Munich, Germany, 6–9 November 2024. URL: <https://openreview.net/forum?id=56IzghzjFZ>
- Janner M, Du Y, Tenenbaum J, et al. (2022) Planning with diffusion for flexible behavior synthesis. In: International conference on machine learning, Baltimore, MD, 17–23 July 2022, pp. 9902–9915. PMLR.
- Jarrett D, Bica I and van der Schaar M (2020) Strictly batch imitation learning by energy-based distribution matching. *Advances in Neural Information Processing Systems* 33: 7354–7365.
- Jia M, Wang D, Su G, et al. (2023) SEIL: simulation-augmented equivariant imitation learning. In: International conference on robotics and automation (ICRA), London, UK, 29 May 2023–2 June 2023.
- Keselman L, Iselin Woodfill J, Grunnet-Jepsen A, et al. (2017) Intel realsense stereoscopic depth cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, Honolulu, HI, 21–26 July 2017, pp. 1–10.
- Kim S, Lim B, Lee Y, et al. (2023) Se (2)-equivariant pushing dynamics models for tabletop object manipulations. In: Conference on robot learning, Atlanta, GA, 6–9 November 2023, pp. 427–436. PMLR.
- Kohler C, Srikanth AS, Arora E, et al. (2023) Symmetric models for visual force policy learning. ArXiv Preprint arXiv: 2308.14670.
- Liang Z, Mu Y, Ding M, et al. (2023) AdaptDiffuser: diffusion models as adaptive self-evolving planners. In: International conference on machine learning, Honolulu, HI, 23–29 July 2023.
- Lim B, Kim J, Kim J, et al. (2024) Equigraspflow: SE(3)-equivariant 6-dof grasp pose generative flows. In: 8th annual conference on robot learning, Munich, Germany, 6–9 November 2024. URL: <https://openreview.net/forum?id=5ISkn5v4LK>
- Liu S, Xu M, Huang P, et al. (2023) Continual vision-Based reinforcement learning with group symmetries. In: Conference on robot learning, Atlanta, GA, 6–9 November 2023, pp. 222–240. PMLR.
- Loshchilov I and Hutter F (2018) Decoupled weight decay regularization. In: International conference on learning representations, Vancouver, BC, 30 April 2018–3 May 2018.
- Mandlekar A, Xu D, Wong J, et al. (2022) What matters in learning from offline human demonstrations for robot manipulation. In: Faust A, Hsu D and Neumann G (eds) Proceedings of the 5th conference on robot learning, proceedings of machine learning research, Auckland, New Zealand, 14–18 December 2022, pp. 1678–1690. PMLR.
- Mandlekar A, Nasiriany S, Wen B, et al. (2023) MimicGen: a data generation system for scalable robot learning using human demonstrations. In: 7th annual conference on robot learning, Atlanta, GA, 6–9 November 2023.
- Nguyen HH, Baisero A, Klee D, et al. (2023) Equivariant reinforcement learning under partial observability. In: Conference on robot learning, Atlanta, GA, 6–9 November 2023, pp. 3309–3320. PMLR.
- Nguyen H, Kozuno T, Beltran-Hernandez CC, et al. (2024) Symmetry-aware reinforcement learning for robotic assembly under partial observability with a soft wrist. ArXiv Preprint arXiv:2402.18002.
- Orsini M, Raichuk A, Hussenot L, et al. (2021) What matters for adversarial imitation learning? *Advances in Neural Information Processing Systems* 34: 14656–14668.
- Pan C, Okorn B, Zhang H, et al. (2023) TAX-Pose: task-specific cross-pose estimation for robot manipulation. In: Conference

- on robot learning, Atlanta, GA, 6–9 November 2023, pp. 1783–1792. PMLR.
- Pearce T, Rashid T, Kanervisto A, et al. (2022) Imitating human behaviour with diffusion models. In: The eleventh international conference on learning representations, Virtual Meeting, 25–29 April 2022.
- Puny O, Atzmon M, Smith EJ, et al. (2022) Frame averaging for invariant and equivariant network design. In: International conference on learning representations, Virtual Meeting, 25–29 April 2022. URL: <https://openreview.net/forum?id=ziUyj55nXR>
- Qi CR, Su H, Mo K, et al. (2017) Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, 21–26 July 2017, pp. 652–660.
- Rahmatizadeh R, Abolghasemi P, Bölöni L, et al. (2018) Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In: 2018 IEEE international conference on robotics and automation (ICRA), Brisbane, QLD, 21–25 May 2018, pp. 3758–3765. IEEE.
- Ryu H, Kim J, Chang J, et al. (2023b) Diffusion-EDFs: bi-equivariant denoising generative modeling on SE(3) for visual robotic manipulation. ArXiv Preprint arXiv:2309.02685.
- Simeonov A, Du Y, Tagliasacchi A, et al. (2022) Neural descriptor fields: SE(3)-equivariant object representations for manipulation. In: 2022 international conference on robotics and automation (ICRA), Philadelphia, PA, 23–27 May 2022, pp. 6394–6400. IEEE.
- Ryu H, Lee H, Lee JH, et al. (2023a) Equivariant descriptor fields: SE(3)-equivariant energy-based models for end-to-end visual robotic manipulation learning. In: *The eleventh international conference on learning representations*, Kigali, Rwanda, 1–5 May 2023.
- Simeonov A, Du Y, Lin YC, et al. (2023) SE(3)-Equivariant relational rearrangement with neural descriptor fields. In: Conference on robot learning, Atlanta, GA, 6–9 November 2023, pp. 835–846. PMLR.
- Sohl-Dickstein J, Weiss E, Maheswaranathan N, et al. (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: Bach F and Blei D (eds) Proceedings of the 32nd international conference on machine learning, proceedings of machine learning research, Lille, France, 6–11 July 2015, pp. 2256–2265. PMLR.
- Song Y and Ermon S (2019) Generative modeling by estimating gradients of the data distribution. In: Advances in neural information processing systems, Vancouver, BC, 8–14 December 2019.
- Song J, Meng C and Ermon S (2020) Denoising diffusion implicit models. In: International conference on learning representations, Virtual Meeting, 26–30 April 2020.
- Toyer S, Shah R, Critch A, et al. (2020) The magical benchmark for robust imitation. *Advances in Neural Information Processing Systems* 33: 18284–18295.
- Wang D, Walters R, Zhu X, et al. (2021a) Equivariant  $Q$  learning in spatial action spaces. In: 5th annual conference on robot learning, London, UK, 8–11 November 2021.
- Wang R, Walters R and Yu R (2021b) Incorporating symmetry into deep dynamics models for improved generalization. In: International conference on learning representations (ICLR), Virtual Meeting, 3–7 May 2021.
- Wang D, Jia M, Zhu X, et al. (2022a) On-robot learning with equivariant models. In: 6th annual conference on robot learning, Auckland, New Zealand, 14–18 December 2022.
- Wang D, Walters R and Platt R (2022b) SO(2)-Equivariant reinforcement learning. In: International conference on learning representations, Virtual Meeting, 25–29 April 2022.
- Wang Z, Hunt JJ and Zhou M (2022c) Diffusion policies as an expressive policy class for offline reinforcement learning. In: The eleventh international conference on learning representations, Virtual Meeting, 25–29 April 2022.
- Wang D, Park JY, Sortur N, et al. (2023) The surprising effectiveness of equivariant models in domains with latent symmetry. In: International conference on learning representations, Kigali, Rwanda, 1–5 May 2023.
- Wang C, Fang H, Fang HS, et al. (2024a) Rise: 3d perception makes real-world robot imitation simple and effective. In: 2024 IEEE/RSJ international conference on intelligent robots and systems (IROS), Abu Dhabi, UAE, 14–18 October 2024, pp. 2870–2877. IEEE.
- Wang D, Hart S, Surovik D, et al. (2024b) Equivariant diffusion policy. In: 8th annual conference on robot learning, Munich, Germany, 6–9 November 2024. URL: <https://openreview.net/forum?id=wD2kUVL1g>
- Wang D, Zhu X, Park JY, et al. (2024c) A general theory of correct, incorrect, and extrinsic equivariance. In: Advances in neural information processing systems, Vancouver, BC, 10–15 December 2024.
- Xian Z, Gkanatsios N, Gervet T, et al. (2023) ChainedDiffuser: unifying trajectory diffusion and keypose prediction for robotic manipulation. In: 7th annual conference on robot learning, Atlanta, GA, 6–9 November 2022.
- Xu M, Yu L, Song Y, et al. (2022) GeoDiff: a geometric diffusion model for molecular conformation generation. In: International conference on learning representations, Virtual Meeting, 25–29 April 2022.
- Yang J, Cao Z, Deng C, et al. (2024a) Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. In: 8th annual conference on robot learning, Munich, Germany, 6–9 November 2024.
- Yang J, Deng C, Wu J, et al. (2024b) Equivact: Sim (3)-equivariant visuomotor policies beyond rigid object manipulation. In: 2024 IEEE international conference on robotics and automation (ICRA), Yokohama, Japan, 13–17 May 2024, pp. 9249–9255. IEEE.
- Ze Y, Zhang G, Zhang K, et al. (2024) 3d diffusion policy: generalizable visuomotor policy learning via simple 3d representations. In: Robotics: science and systems, Delft, Netherlands, 15–19 July 2024.
- Zhang T, McCarthy Z, Jow O, et al. (2018) Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In: 2018 IEEE international conference on robotics



$$\rho_{\mathbf{R}} = \begin{bmatrix} c^2 & -cs & 0 & -cs & s^2 & 0 & 0 & 0 & 0 \\ cs & c^2 & 0 & -s^2 & -cs & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 & -s & 0 & 0 & 0 \\ cs & -s^2 & 0 & c^2 & -cs & 0 & 0 & 0 & 0 \\ s^2 & cs & 0 & cs & c^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & s & 0 & 0 & c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c & -s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (18)$$

where  $c = \cos g$ ,  $s = \sin g$ . To decompose it into the irreducible representations of SO (2), we define

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (19)$$

Such that  $P\rho_{\mathbf{R}}P^{-1}$  is a block diagonal matrix consisting of irreducible representations

$$P\rho_{\mathbf{R}}P^{-1} = \begin{bmatrix} \rho_0(g) & & & & & & & & \\ & \rho_0(g) & & & & & & & \\ & & \rho_0(g) & & & & & & \\ & & & \rho_1(g) & & & & & \\ & & & & \rho_1(g) & & & & \\ & & & & & \rho_2(g) & & & \end{bmatrix}. \quad (20)$$

We can then use  $\rho(g) = P\rho_{\mathbf{R}}P^{-1} = \rho_0^3(g) \oplus \rho_1^2(g) \oplus \rho_2(g) \in \mathbb{R}^{9 \times 9}$  as the group representation of the output of the equivariant network, then construct the  $3 \times 3$  rotation matrix  $\mathbf{R}_t$  using  $P$ . Specifically, let  $V \in \mathbb{R}^9$  be the output of the network associated with the representation  $\rho(g)$  (i.e.,  $g$  acts on  $V$  through  $\rho(g)V$ ). Define

$$\text{Vec}_r(\mathbf{R}_t) = P^{-1}V. \quad (21)$$

Applying  $\rho(g)$  on  $V$  will lead to

$$P^{-1}\rho(g)V \quad (22)$$

$$= P^{-1}P\rho_{\mathbf{R}}P^{-1}V \quad (23)$$

$$= \rho_{\mathbf{R}}\text{Vec}_r(\mathbf{R}), \quad (24)$$

which is the desired property in the equivariant network.

In the end, adding the group action for the translation ( $\rho_1 \oplus \rho_0$ ) and gripper open width ( $\rho_0$ ), we have  $\rho_a = \rho_0^5 \oplus \rho_1^3 \oplus \rho_2$ .

## Network architecture detail

In the image version, we implement the equivariant observation encoder with an equivariant ResNet (Wang et al., 2021b) for the agent-view image, a standard ResNet (He et al., 2016) for the eye-in-hand image, and an equivariant MLP for the robot states. We implement the equivariant layers in the group  $C_8$ . Figure 12 shows the detailed network architecture of our Equivariant Diffusion Policy with image input, that is, EquiDiff (Im), in the simulation experiments. The network is defined under group  $C_8$ . First, in the encoding phase, the agent-view image is processed with an equivariant ResNet-18, whose output is a  $128 \times 8$ -dimensional regular representation vector of group. A non-equivariant ResNet-18 with a spatial maxpool at the end processes the eye-in-hand image and outputs a 128-dimensional representation vector that uses the trivial invariance representation. Those two vectors are concatenated with the gripper position (represented using  $\rho_1 \oplus \rho_0$ ), gripper orientation (in the format of 6D rotation, represented using  $\rho_1^3$ ), and the gripper finger position (represented using  $\rho_0^2$ ). The concatenated mixed-representation vector is sent to an equivariant linear layer, whose output is a  $128 \times 8$ -dimensional regular representation observation embedding. The noisy action is also encoded using an equivariant linear layer, whose output is a  $64 \times 8$ -dimensional regular representation action embedding. Second, in the denoising phase, we process each part of the observation embedding and the action embedding that corresponds to the same group element with a 1D Temporal U-Net with hidden dimensions of [512, 1024, 2048] to get a 64-dimensional vector. Doing so for each pair, we will recover a  $64 \times 8$ -dimensional regular representation noise embedding. In the end, an equivariant linear layer will decode the noise.

In the voxel version, that is, EquiDiff (Vo), the agent-view image is replaced with a voxel grid, and we replace the equivariant ResNet with an 8-layer 3D equivariant convolutional encoder. The 1D Temporal U-Net has hidden dimensions of [256, 512, 1024]. The other part of the network stays the same.

In the point cloud version, that is, EquiDiff (PC), the agent-view image is replaced with a point cloud, whereas we use our Equivariant Point Transformer (Section Equivariant Point Transformer and Translation Symmetry) as the encoder. The 1D Temporal U-Net has hidden dimensions of [512, 1024, 2048], and the other part of the network stays the same.

In all real-world experiments, we remove the eye-in-hand image and only use the voxel grid or point cloud as vision input (the gripper state vector stays the same).

## Simulation environments

Figure 6 shows the initial and goal states of each tasks. Figure 13 shows an example trajectory for finishing the



**Table 8.** The maximum number of time steps and the maximum out-of-plane rotation (in degrees) in the demo for each simulation environments. The maximum out-of-plane rotation in the demo is the maximum angular difference between the SO (3) rotation and the SO (2) rotation (i.e., only rotating around the z axis) over all demonstration steps, averaged over 1000 demonstration episodes.

Task	Max steps	Max out of plane rot in demo
Stack D1	400	11.2
Stack three D1	400	13.2
Square D2	400	14.7
Threading D2	400	13.4
Coffee D2	400	14.1
Three piece assembly D2	500	16.2
Hammer cleanup D1	500	16.4
Mug cleanup D1	500	13.0
Kitchen D1	800	16.2
Nut assembly D0	500	15.5
Pick place D0	1000	18.0
Coffee preparation D1	800	59.0

and the SO (2) rotation (i.e., only rotating around the z axis) across the entire demonstration episode. Results averaged for 1000 demonstrations.

### Training detail

In the simulation experiments, we follow the hyper-parameters of the prior work (Chi et al., 2023) for the image version of our method, where the only change is that we increase the batch size to 128 for faster training. Specifically, the observation contains two steps of history observation, and the output of the denoising process is a sequence of 16 action steps. We use all 16 steps for training but only execute eight steps in evaluation. We train our models with the AdamW (Loshchilov and Hutter, 2018) optimizer (with a learning rate of  $10^{-4}$  and weight decay of  $10^{-6}$ ) and Exponential Moving

Average (EMA). We use a cosine learning rate scheduler with 500 warm-up steps. We use DDPM (Ho et al., 2020) with 100 denoising steps for both training and evaluation. For each different number of demos (100, 200, 1000), we maintain roughly the same number of training steps by training for  $50,000/n$  epochs where  $n$  is the number of demos. Evaluations are conducted every  $1000/n$  epochs (50 evaluations in total). In the voxel version, we use only one step of history observation, and keep the other hyper-parameters the same.

The hyper-parameters for the diffusion policy and BC RNN baselines exactly follow Chi et al. (2023). We follow the original work (Ze et al., 2024) for the hyper-parameters of DP3, except that we use the same action sequence length (16 for training and 8 for evaluation) as Chi et al. (2023) and our method. For the ACT baseline, we follow the hyper-parameters provided in

**Table 9.** The observation format, observation step, action prediction step, and action execution step for all methods. The gripper state is a vector including a 3 dimensional position vector, a rotation vector in the format of 6D rotation representation or 4D quaternion, and a 2 dimensional finger position.

Method	Obs	Obs step	Action pred. step	Action exec. step
EquiDiff (PC)	Point cloud, eye-in-hand image, gripper state	1	16	8
EquiDiff (Vo)	Voxel grid, eye-in-hand image, gripper state	1	16	8
EquiDiff (Im)	Agent-view image, eye-in-hand image, gripper state	2	16	8
DiffPo-C	Agent-view image, eye-in-hand image, gripper state	2	16	8
DiffPo-T	Agent-view image, eye-in-hand image, gripper state	2	10	8
DP3	Point cloud, gripper state	2	16	8
ACT	Agent-view image, eye-in-hand image, gripper state	1	10	10
BC-RNN	Agent-view image, eye-in-hand image, gripper state	1	1	1
RISE	Point cloud/point cloud, eye-in-hand image, gripper state	1	20/16	10/8
ISP	Large FOV eye-in-hand image, gripper state	1	16	8
EquiDiff (PC) real	Point cloud, gripper state	1	16	8
EquiDiff (Vo) real	Voxel grid, gripper state	1	16	8
DiffPo-C (Vo) real	Voxel grid, gripper state	1	16	8

**Table 10.** The performance of our Equivariant Diffusion Policy compared with the baselines in simulation.

Method	Stack DI			Stack Three DI			Square D2			Threading D2			
	Ctrl	100	200	1000	200	1000	100	1000	200	1000	100	200	1000
EquiDiff (PC)	Abs	98.0 ± 1.2	100.0 ± 0.0	100.0 ± 0.0	90.0 ± 1.2	96.0 ± 1.2	96.7 ± 0.7	66.7 ± 7.5	80.7 ± 3.7	74.7 ± 3.7	55.3 ± 0.7	60.0 ± 1.2	59.3 ± 1.8
EquiDiff (Vo)		98.7 ± 0.7	100.0 ± 0.0	100.0 ± 0.0	74.7 ± 4.4	91.3 ± 0.7	90.7 ± 1.3	38.7 ± 1.3	48.0 ± 3.1	63.3 ± 1.3	38.7 ± 0.7	52.7 ± 2.9	54.7 ± 2.9
EquiDiff (Im)		93.3 ± 0.7	100.0 ± 0.0	100.0 ± 0.0	54.7 ± 5.2	77.3 ± 1.8	96.0 ± 1.2	25.3 ± 8.7	41.3 ± 9.8	60.0 ± 4.2	22.0 ± 1.2	40.0 ± 1.2	59.3 ± 1.8
DiffPo-C		76.0 ± 4.0	97.3 ± 0.7	100.0 ± 0.0	38.0 ± 0.0	72.0 ± 2.0	94.0 ± 1.2	8.0 ± 1.2	19.3 ± 5.3	46.0 ± 7.2	17.3 ± 1.8	35.3 ± 1.3	58.7 ± 0.7
DiffPo-T		51.3 ± 1.8	82.7 ± 0.7	98.7 ± 0.7	16.7 ± 0.7	41.3 ± 2.9	84.0 ± 1.2	4.7 ± 1.8	11.3 ± 2.4	44.7 ± 4.7	10.7 ± 0.7	18.0 ± 1.2	40.7 ± 0.7
DP3		69.3 ± 3.7	86.7 ± 4.7	99.3 ± 0.7	7.3 ± 0.7	22.7 ± 3.7	65.3 ± 1.8	6.7 ± 0.7	6.0 ± 0.0	19.3 ± 3.3	12.0 ± 3.1	23.3 ± 3.3	40.0 ± 2.0
ACT		34.7 ± 0.7	72.7 ± 7.7	96.0 ± 1.2	6.0 ± 2.3	36.7 ± 2.7	78.0 ± 1.2	6.0 ± 0.0	18.0 ± 1.2	49.3 ± 4.7	10.0 ± 1.2	20.7 ± 2.9	35.3 ± 2.4
EquiDiff (Vo)	Rel	94.7 ± 1.3	100.0 ± 0.0	100.0 ± 0.0	59.3 ± 0.7	76.0 ± 0.0	82.7 ± 0.7	24.7 ± 1.8	34.7 ± 5.2	52.0 ± 2.3	33.3 ± 1.8	38.7 ± 2.9	46.0 ± 1.2
EquiDiff (Im)		74.7 ± 5.8	96.0 ± 0.0	100.0 ± 0.0	25.3 ± 3.3	62.7 ± 3.5	92.0 ± 1.2	11.3 ± 1.3	20.7 ± 4.1	48.0 ± 4.0	11.3 ± 1.3	22.0 ± 1.2	49.3 ± 2.4
DiffPo-C		80.7 ± 2.4	93.3 ± 0.7	99.3 ± 0.7	26.0 ± 4.0	52.0 ± 2.0	86.0 ± 1.2	6.0 ± 1.2	13.3 ± 1.3	36.7 ± 4.8	13.3 ± 1.8	26.0 ± 3.1	40.0 ± 2.3
BC RNN		59.3 ± 7.0	94.7 ± 1.3	100.0 ± 0.0	12.0 ± 2.5	48.0 ± 5.3	92.0 ± 2.3	8.0 ± 1.2	20.7 ± 2.7	58.7 ± 3.5	7.3 ± 0.7	13.3 ± 2.4	46.7 ± 0.7
	Coffee D2				Three Pc. Assembly D2			Hammer Cleanup D1			Mug Cleanup D1		
EquiDiff (PC)	Abs	78.0 ± 1.2	74.0 ± 1.2	75.3 ± 2.9	66.0 ± 3.5	72.0 ± 1.2	69.3 ± 0.7	80.7 ± 1.8	80.7 ± 1.8	82.0 ± 0.0	64.7 ± 1.3	70.7 ± 1.3	71.3 ± 1.8
EquiDiff (Vo)		64.7 ± 0.7	73.3 ± 1.8	76.0 ± 0.0	37.3 ± 2.7	58.0 ± 5.0	71.3 ± 3.3	70.0 ± 2.0	66.0 ± 2.3	72.7 ± 0.7	52.7 ± 1.3	64.7 ± 2.4	68.0 ± 1.2
EquiDiff (Im)		60.0 ± 2.0	79.3 ± 1.3	76.0 ± 2.0	15.3 ± 1.8	39.3 ± 1.8	69.3 ± 3.5	65.3 ± 0.7	63.3 ± 4.4	76.7 ± 0.7	49.3 ± 0.7	64.0 ± 1.2	66.7 ± 0.7
DiffPo-C		44.0 ± 1.2	66.0 ± 2.3	78.7 ± 0.7	4.0 ± 0.0	6.0 ± 1.2	30.0 ± 1.2	52.0 ± 1.2	58.7 ± 1.3	73.3 ± 2.4	42.7 ± 0.7	58.7 ± 1.3	65.3 ± 2.4
DiffPo-T		47.3 ± 0.7	60.7 ± 1.8	74.7 ± 2.7	0.7 ± 0.7	4.0 ± 0.0	42.7 ± 1.3	48.0 ± 1.2	60.0 ± 1.2	76.0 ± 1.2	30.0 ± 1.2	42.7 ± 2.9	63.3 ± 0.7
DP3		34.0 ± 4.0	45.3 ± 4.1	68.7 ± 2.4	0.0 ± 0.0	0.7 ± 0.7	3.3 ± 0.7	54.0 ± 3.1	70.7 ± 4.1	86.7 ± 0.7	21.3 ± 2.7	32.7 ± 1.8	52.7 ± 4.4
ACT		19.3 ± 2.4	33.3 ± 2.4	64.0 ± 2.3	0.0 ± 0.0	3.3 ± 0.7	24.0 ± 3.1	38.0 ± 4.2	54.0 ± 1.2	70.7 ± 1.3	23.3 ± 0.7	31.3 ± 1.3	56.0 ± 2.0
EquiDiff (Vo)	Rel	55.3 ± 0.7	59.3 ± 0.7	64.0 ± 0.0	4.7 ± 0.7	5.3 ± 0.7	54.7 ± 3.5	64.0 ± 1.2	62.0 ± 1.2	67.3 ± 1.3	39.3 ± 0.7	43.3 ± 1.8	62.0 ± 1.2
EquiDiff (Im)		40.7 ± 0.7	58.7 ± 1.8	66.0 ± 1.2	1.3 ± 0.7	4.7 ± 0.7	59.3 ± 4.8	48.7 ± 2.7	52.0 ± 3.5	68.7 ± 2.4	29.3 ± 2.9	36.0 ± 1.2	65.3 ± 2.4
DiffPo-C		42.7 ± 1.8	50.7 ± 1.8	66.7 ± 2.9	2.0 ± 1.2	2.0 ± 0.0	20.0 ± 1.2	43.3 ± 1.8	54.0 ± 1.2	65.3 ± 1.8	25.3 ± 0.7	39.3 ± 1.8	54.7 ± 0.7
BC RNN		37.0 ± 1.0	52.0 ± 2.0	76.0 ± 2.3	0.0 ± 0.0	5.3 ± 0.7	27.0 ± 1.0	32.0 ± 0.0	42.7 ± 0.7	72.0 ± 2.3	19.3 ± 0.7	39.0 ± 1.0	66.7 ± 0.7
	Kitchen D1				Nut Assembly D0			Pick Place D0			Coffee Preparation D1		
EquiDiff (PC)	Abs	84.0 ± 2.0	86.0 ± 1.2	86.0 ± 1.2	91.3 ± 0.3	94.3 ± 0.9	94.7 ± 0.7	58.8 ± 1.1	78.7 ± 1.4	90.0 ± 1.6	84.0 ± 1.2	84.7 ± 1.3	88.0 ± 1.2
EquiDiff (Vo)		85.3 ± 0.7	89.3 ± 0.7	88.0 ± 2.3	67.3 ± 0.9	77.0 ± 0.0	83.3 ± 0.7	57.7 ± 1.8	68.5 ± 0.6	82.2 ± 0.8	80.0 ± 1.2	83.3 ± 1.8	85.3 ± 1.8
EquiDiff (Im)		67.3 ± 0.7	76.7 ± 3.3	81.3 ± 0.7	74.0 ± 1.2	85.0 ± 1.5	93.7 ± 0.9	41.7 ± 3.2	74.2 ± 3.2	92.0 ± 1.2	76.7 ± 0.7	82.7 ± 0.7	85.3 ± 0.7
DiffPo-C		66.7 ± 2.4	84.7 ± 0.7	86.7 ± 1.8	54.7 ± 2.3	68.0 ± 2.6	83.0 ± 1.5	35.3 ± 2.2	65.0 ± 2.8	82.7 ± 0.6	65.3 ± 0.7	62.0 ± 4.2	58.0 ± 3.1
DiffPo-T		54.0 ± 2.3	75.3 ± 0.7	81.3 ± 2.4	30.7 ± 5.0	32.3 ± 5.2	45.7 ± 5.9	14.7 ± 1.5	36.5 ± 1.3	50.0 ± 6.0	38.0 ± 2.0	51.3 ± 1.8	76.0 ± 6.0
DP3		44.7 ± 1.8	71.3 ± 2.4	91.3 ± 2.4	15.7 ± 1.3	23.7 ± 3.4	57.7 ± 1.9	11.7 ± 0.9	15.0 ± 1.7	34.0 ± 0.0	10.0 ± 2.3	22.0 ± 5.3	63.3 ± 4.1
ACT		37.3 ± 3.5	60.7 ± 3.5	87.3 ± 3.5	42.3 ± 2.9	63.7 ± 3.5	84.3 ± 0.9	7.2 ± 0.9	17.2 ± 1.1	50.0 ± 2.9	32.0 ± 2.0	46.0 ± 3.1	64.7 ± 2.4
EquiDiff (Vo)	Rel	69.3 ± 1.8	82.7 ± 1.3	89.3 ± 1.8	53.0 ± 1.0	65.0 ± 2.0	72.0 ± 2.0	40.3 ± 1.6	58.2 ± 0.9	78.8 ± 0.8	48.0 ± 1.2	70.7 ± 2.9	73.3 ± 1.8
EquiDiff (Im)		60.7 ± 1.3	72.0 ± 3.1	82.7 ± 2.7	44.3 ± 1.2	65.3 ± 1.5	87.3 ± 0.9	29.3 ± 3.1	54.7 ± 1.5	91.3 ± 1.2	48.7 ± 1.3	59.3 ± 2.4	79.3 ± 0.7
DiffPo-C		42.0 ± 2.3	64.0 ± 5.0	81.3 ± 1.3	41.7 ± 2.7	62.0 ± 1.5	75.3 ± 1.2	34.7 ± 1.1	58.7 ± 1.0	82.2 ± 2.5	42.0 ± 3.1	52.7 ± 3.3	51.3 ± 1.8
BC RNN		31.3 ± 2.9	46.7 ± 6.7	80.7 ± 1.3	35.3 ± 0.7	58.0 ± 2.1	85.0 ± 1.2	21.2 ± 0.7	41.0 ± 9.0	77.3 ± 2.5	14.0 ± 1.2	32.0 ± 1.2	60.7 ± 4.1

We experiment with 100, 200, and 1000 demos in each environment and report the maximum task success rate among 50 evaluations throughout training. Results averaged over three seeds. ± indicates standard error.

**Table 11.** The ablation study that ablates the voxel input and the equivariant structure in our method.

Ablation	Method	Ctrl	Obs	100	200	1000	100	200	1000	100	200	1000	100	200	1000
—	EquiDiff (Vo)	Abs	Voxel	Stack D1			Stack Three D1			Square D2			Threading D2		
No Voxel	EquiDiff (Im)		RGB	99	100	100	75	91	91	39	48	63	39	53	55
No Equi.	DiffPo-C (Vo)		Voxel	93	100	100	55	77	96	25	41	60	22	40	59
No Voxel No Equi.	DiffPo-C		RGB	87	99	100	33	79	94	10	24	60	19	43	54
—	EquiDiff (Vo)	Abs	Voxel	Coffee D2			Three Pc. Asse. D2			Hammer Cleanup D1			Mug Cleanup D1		
No Voxel	EquiDiff (Im)		RGB	65	73	76	37	58	71	70	66	73	53	65	68
No Equi.	DiffPo-C (Vo)		Voxel	60	79	76	15	39	69	65	63	77	49	64	67
No Voxel No Equi.	DiffPo-C		RGB	50	72	75	2	5	50	54	64	76	47	58	66
—	EquiDiff (Vo)	Abs	Voxel	Kitchen D1			Nut Assembly D0			Pick Place D0			Coffee Prep. D1		
No Voxel	EquiDiff (Im)		RGB	85	89	88	67	77	83	58	69	82	80	83	85
No Equi.	DiffPo-C (Vo)		Voxel	67	77	81	74	85	94	42	74	92	77	83	85
No Voxel No Equi.	DiffPo-C		RGB	82	87	87	66	77	84	41	67	84	65	75	77
—	EquiDiff (Vo)	Abs	Voxel	Kitchen D1			Nut Assembly D0			Pick Place D0			Coffee Prep. D1		
No Voxel	EquiDiff (Im)		RGB	67	77	81	74	85	94	42	74	92	77	83	85
No Equi.	DiffPo-C (Vo)		Voxel	82	87	87	66	77	84	41	67	84	65	75	77
No Voxel No Equi.	DiffPo-C		RGB	67	85	87	55	68	83	35	65	83	65	62	58

We experiment with 100, 200, and 1000 demos in each environment and report the maximum task success rate among 50 evaluations throughout training. Results averaged over three seeds.

the prior work (Zhao et al., 2023), except that we use a chunk size of 10, KL weight of 10, batch size of 64 with a learning rate of  $5 \times 10^{-5}$ , and no temporal aggregation, following the tuning tips provided by the authors. The hyper-parameters for RISE (Wang et al., 2024a) and ISP (Hu et al., 2025) also exactly follow the prior works. For RISE, we use the point-cloud-only version (with 20 action prediction steps and 10 action execution steps) for the first 9 tasks, and the hybrid version (with 16 action prediction steps and 8 action execution steps) for Nut Assembly D0, Pick Place D0, and Coffee Preparation D1, because we empirically found that the point-cloud-only version works better in short-horizon tasks and the hybrid version works better in long-horizon tasks. See Table 9 for the observation format, observation step, action prediction step, and action execution step for all methods.

In the real-world experiments, we use a batch size of 64, one step of observation, and disable the EMA. We use DDIM (Song et al., 2020) with 100 denoising steps for training and 16 denoising steps for evaluation. The other hyper-parameters stay the same as in simulation.

### Simulation experiment result with standard error

Table 10 shows the same result in Table 1 with the standard error.

### Ablation study of EquiDiff (Vo)

We perform an ablation study regarding the equivariant structure and the voxel input in our method. We consider the following four candidates: (1) Ours: our Equivariant Diffusion Policy with voxel input; (2) Ours no Voxel: our Equivariant Diffusion Policy with RGB input; (3) Ours no Equi.: the baseline Diffusion Policy with voxel input; (4) Ours no Voxel no Equi.: the baseline Diffusion Policy with RGB input, same as Chi et al. (2023). Table 11 shows the result and Table 12 shows the average over all 12 environments. Though both the equivariant structure and the voxel input contribute to the performance improvement of our method, the equivariant structure plays a more important rule, as removing it (No Equi.) lead to a more significant performance drop compared with removing the voxel input

**Table 12.** The average performance over 12 tasks of the ablation study.

Ablation	Method	Ctrl	Average over 12 environments		
			100	200	1000
—	EquiDiff (Vo)	Abs	63.9	72.6	77.9
No Voxel	EquiDiff (Im)		53.7 (−10.3)	68.5 (−4.1)	79.7 (+1.8)
No Equi.	DiffPo-C (Vo)		46.3 (−17.6)	62.5 (−10.1)	75.6 (−2.3)
No Voxel No Equi.	DiffPo-C		42.0 (−21.9)	57.8 (−14.8)	71.4 (−6.5)

Number in parenthesis shows the performance difference after removing different components in our Equivariant Diffusion Policy with voxel input.

**Table 13.** Comparing implementing equivariance via equivariant network or data augmentation.

Method	Obs	Stack D1	Stack three D1	Square D2	Threading D2
Equi. Net (Vo)	Voxel	98.7 ± 0.7	74.7 ± 4.4	38.7 ± 1.3	38.7 ± 0.7
CNN + Aug (Vo)	Voxel	99.3 ± 0.7	84.0 ± 1.2	36.0 ± 1.2	30.7 ± 1.8
CNN (Vo)	Voxel	86.7 ± 1.3	33.3 ± 1.8	10.0 ± 2.0	19.3 ± 2.4
Equi. Net (Im)	RGB	93.3 ± 0.7	54.7 ± 5.2	25.3 ± 8.7	22.0 ± 1.2
CNN + Aug (Im)	RGB	98.7 ± 0.7	68.0 ± 1.2	26.7 ± 1.8	22.0 ± 1.2
CNN (Im)	RGB	76.0 ± 4.0	38.0 ± 0.0	8.0 ± 1.2	17.3 ± 1.8
Method	Obs	Coffee D2	Three Pc. Asse. D2	Hammer Cleanup D1	Mug Cleanup D1
Equi. Net (Vo)	Voxel	64.7 ± 0.7	37.3 ± 2.7	70.0 ± 2.0	52.7 ± 1.3
CNN + Aug (Vo)	Voxel	56.7 ± 2.9	7.3 ± 0.7	70.7 ± 1.8	52.0 ± 1.2
CNN (Vo)	Voxel	50.0 ± 3.1	2.0 ± 0.0	54.0 ± 3.1	46.7 ± 0.7
Equi. Net (Im)	RGB	60.0 ± 2.0	15.3 ± 1.8	65.3 ± 0.7	49.3 ± 0.7
CNN + Aug (Im)	RGB	58.0 ± 1.2	5.3 ± 0.7	61.3 ± 2.9	50.0 ± 1.2
CNN (Im)	RGB	44.0 ± 1.2	4.0 ± 0.0	52.0 ± 1.2	42.7 ± 0.7
Method	Obs	Kitchen D1	Nut assembly D0	Pick Place D0	Coffee Prep. D1
Equi. Net (Vo)	Voxel	85.3 ± 0.7	67.3 ± 0.9	57.7 ± 1.8	80.0 ± 1.2
CNN + Aug (Vo)	Voxel	62.0 ± 2.0	51.7 ± 1.5	39.5 ± 2.8	48.7 ± 2.4
CNN (Vo)	Voxel	82.0 ± 2.3	66.0 ± 1.7	40.8 ± 1.9	65.3 ± 0.7
Equi. Net (Im)	RGB	67.3 ± 0.7	74.0 ± 1.2	41.7 ± 3.2	76.7 ± 0.7
CNN + Aug (Im)	RGB	47.3 ± 2.9	53.7 ± 0.7	27.7 ± 0.8	34.7 ± 2.9
CNN (Im)	RGB	66.7 ± 2.4	54.7 ± 2.3	35.3 ± 2.2	65.3 ± 0.7
Method	Obs	Average over 12 environments			
Equi. Net (Vo)	Voxel	63.9			
CNN + Aug (Vo)	Voxel	53.3			
CNN (Vo)	Voxel	46.3			
Equi. Net (Im)	RGB	53.7			
CNN + Aug (Im)	RGB	46.2			
CNN (Im)	RGB	42.0			

We experiment with 100 demos in each environment and report the maximum task success rate among 50 evaluations throughout training. Results averaged over three seeds.

(No Voxel). Note that by using the voxel input, Diffpo-C (Vo) is marginally better than the original Diffusion Policy (DiffPo-C), thus we use Diffpo-C (Vo) as the baseline in our robot experiment in Section Real-Robot Experiment.

### Implementing equivariance via data augmentation

In this section, we evaluate implementing equivariance through data augmentation instead of using equivariant networks. Specifically, we applied random rotation data augmentation based on our analysis in Sections Theory of Equivariant Diffusion Policy and SO(2) Representation on 6DoF Action to a standard, unconstrained CNN. We then compared this CNN + Aug baseline against using equivariant neural networks (Equi. Net) and not implementing equivariance at all (CNN).

As is shown in Table 13, CNN + Aug can achieve good performance, occasionally even outperforming equivariant

networks in simpler tasks like Stack and Stack Three. However, it performs poorly in more challenging tasks. When averaged across 12 environments, CNN + Aug performs better than CNN but still underperforms compared to Equi. Net by a significant margin.

### SE (2) action space variation

In this section, we evaluate a variation of our Equivariant Diffusion Policy in an SE (2) (with  $z$  translation) action space to demonstrate the necessity of leveraging an SE (3) action space. Specifically, the SE (2) agent only learns the top-down rotation and the out-of-plane rotations will be constantly set to 0. As is shown in Table 14, the SE (2) variation achieves a similar performance as the SE (3) version in Stack Three, as the demonstration data in this task has the least amount of out-of-plane rotation (as shown in Table 8). On the other hand, the SE (2) variation

**Table 14.** Performance of Equivariant Diffusion Policy in SE (2) action space compared with SE (3) action space. 200 demos are used in this experiment.

	Stack three D1	Threading D2	Coffee Preparation D1
EquiDiff (Im), SE (3) Action	77.3	40.0	85.3
EquiDiff (Im), SE (2) Action	75.3	12.7	0.0

significantly underperforms in Threading, since the ability of wiggling the out-of-plane rotation helps the agent to precisely insert the tool. In the end, the SE (2) agent cannot solve Coffee Preparation at all, because the task requires a significant amount of out-of-plane rotation (as shown in Figure 13(b)).

### Robomimic experiment

In this section, we compare our Equivariant Diffusion Policy with the original Diffusion Policy across four Robomimic tasks (Figure 14). Both methods are trained with 100 Proficient-Human (PH) demonstrations or 100 Multi-Human (MH) demonstrations. Other hyper-parameters mirror those used in our MimicGen experiment.

Table 15 shows the result. Our method achieves similar or slightly better performance compared to the baseline Diffusion Policy. The improvements in Robomimic tasks are smaller than in MimicGen tasks. This is because the Robomimic tasks can be classified as Low-Equivariance Tasks (as illustrated in Figure 5, bottom), with minimal randomness in the initial distribution (except for Lift), making the symmetry in our method less advantageous.

### Performance under hyper-parameter changes

In this section, we test the training stability of our method by modifying some hyper-parameters for training. Specifically, we experiment with our EquiDiff (PC) method, and consider the following hyper-parameter changes:

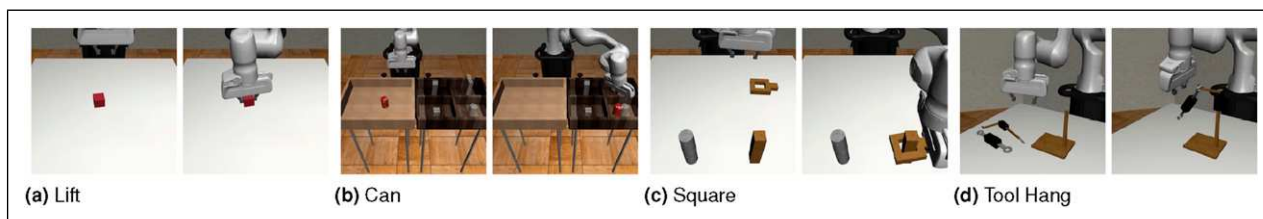
1. Batch size: 128  $\rightarrow$  64
2. Learning rate warm-up: 1000  $\rightarrow$  500
3. Equivariant point transformer optimizer weight decay: 0.0001  $\rightarrow$  0.01

4. Equivariant point transformer optimizer learning rate: 0.0005  $\rightarrow$  0.0001

In all variations, EquiDiff (PC) achieves a 100% test success rate in the Stack D1 task, demonstrating the robustness and stability of our method.

### Real-robot environment details

Figure 9 shows our real-world experimental platform containing a Franka Emika (Haddadin et al., 2022) and three Intel Realsense (Keselman et al., 2017) D455 cameras. Compared with simulation, we use a pair of fin-ray (Crooks et al., 2016) gripper fingers instead of the original Franka fingers. Figure 8 shows the five tasks in this experiment. In Oven Opening, the oven is randomly initialized at one of the four borders of the workspace. In Banana in Bowl, the initial poses of the banana and the bowl are both randomly sampled. In Trash Sweeping, the robot needs to use a tool brush to sweep two pieces of crumpled paper out of its workspace. The initial poses of the objects are randomly sampled. In Letter Alignment, the robot needs to align the letters to form “AI.” The letter A is randomly initialized at one of the four corners of the workspace, and the pose of the I is randomly sampled. In Hammer to Drawer, the robot needs to open a drawer, pick up a hammer, place it inside the drawer, and close the drawer. The drawer is initialized at one of the four borders of the workspace, and the hammer is randomly initialized at the opposite side of the drawer. Lastly, we also evaluate a Bagel Baking task with an extremely long time horizon, where the robot needs to open the oven, pull out the tray inside the oven, pick up the bagel, place it inside the tray, close the tray, and close the oven. In this task, the oven is randomly initialized at one of the three borders of the workspace (where we eliminate the side that is furthest from the robot to avoid joint limits of the robot), and

**Figure 14.** The experimental environments from Robomimic. The left image in each subfigure shows the initial state of the environment; the right image shows the goal state. (a) Lift, (b) can, (c) square, and (d) tool hang.

**Table 15.** The performance of our Equivariant Diffusion Policy compared with the Diffusion Policy baseline in Robomimic.

	Lift		Can		Square		Tool hang	Average
	100 PH	100 MH	100 PH	100 MH	100 PH	100 MH	100 PH	
EquiDiff	100.0 ± 0.0	100.0 ± 0.0	99.3 ± 0.7	96.7 ± 0.7	84.0 ± 1.2	76.7 ± 1.3	76.0 ± 0.0	90.4 ± 2.3
DiffPo	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	95.3 ± 0.7	85.3 ± 0.7	70.7 ± 0.7	64.0 ± 5.8	87.9 ± 3.2

We experiment with 100 Proficient-Human (PH) or Multi-Human (MH) demos in each environment and report the maximum task success rate among 50 evaluations throughout training. Results averaged over three seeds.  $\pm$  indicates standard error.

the bagel is randomly initialized along the opposite side of the oven. The observation is a voxel grid with a resolution of  $64 \times 64 \times 64$  and the gripper pose and open width. The voxel grid covers the  $(0.4 \text{ m})^3$  workspace. During training, we apply a random crop augmentation to crop the voxel grid to  $58 \times 58 \times 58$ . In Banana in Bowl and Trash Sweeping, we train the model with an additional random rotation augmentation. The baseline is trained with the same data augmentation as our method.

In Section EquiDiff with Point Cloud Input, we experiment with six additional environments (as shown in Figure 10). In Coffee Making, the robot needs to pick up a mug, place it underneath the coffee machine, open up the lid of the coffee machine, pick up the k-cup, place it inside the coffee machine, close the lid, and press a button on the coffee machine. The coffee machine, the mug, and the k-cup are randomly initialized in the workspace. In Twist Pipe, the robot needs to twist the connector to detach the pipe, which is initialized in one of four fixed poses. In Toast Making, the

robot needs to pick-place two toasters inside the toast oven, then press to start the oven. The toaster oven is randomly initialized along one of the four borders in the workspace, and the toast holder is randomly placed inside the workspace. In Seat Wiping, the robot needs to pick up a cloth and wipe a bike seat. The bike seat is placed in one of eight poses, and the cloth is randomly placed inside the workspace. In Tool Box, the robot needs to open a tool box, pick-place three tools inside the tool box, then close the tool box. The workspace is separated into two rectangular regions (vertically or horizontally), where the tool box and the tools are placed randomly in each of the two regions. In Screwdriver to Drawer, the robot needs to open the drawer, pick up the screwdriver, place it inside the drawer, and close the drawer. The drawer is randomly placed with the handle pointing the center of the workspace, and the screwdriver is also randomly placed in the workspace. The observation is a point cloud with size  $1024 \times 6$  covering the  $(0.4 \text{ m})^3$  workspace.