Equivariant BC-RNN for Robotic Manipulation

Ekeno Lokwakai

***Abstract*—Robotic manipulation requires policies that generalize across varying object orientations and sequential tasks. Traditional imitation learning methods, such as Behavior Cloning with Recurrent Neural Networks (BC-RNN), struggle with geometric transformations like rotations. This work introduces Equivariant BC-RNN, which integrates SO(2) rotational symmetry into a recurrent framework to improve adaptability in planar manipulation tasks. By preprocessing states with an SO(2)-equivariant layer—computing rotation-invariant norms—the model aims for consistent action prediction under unseen rotations, with a GRU backbone for temporal coherence. Evaluated in Robosuite and Robomimic against BC and BC-RNN baselines, the approach targets success rate, training efficiency, data efficiency and generalization to novel shapes. Results show limited improvement over baselines, highlighting implementation challenges, but offer insights into combining equivariance with sequential modeling. The work underscores the potential of symmetry-aware policies for scalable robotic learning.**

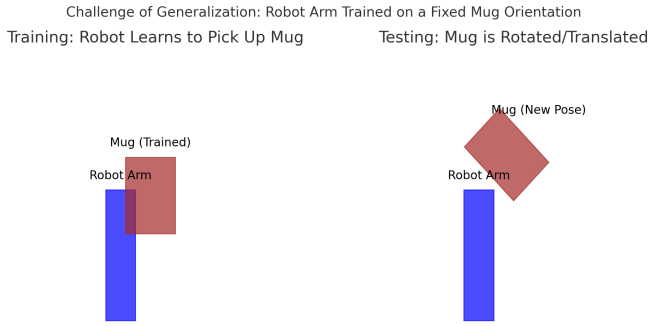

Fig. 1. Illustration of the generalization challenge in robotic manipulation. (**Left**) The robot is trained to pick up a mug in a specific orientation. (**Right**) During testing, the mug is rotated and translated, and the robot struggles to generalize. This demonstrates the limitations of standard imitation learning.

I. Introduction

Robots are increasingly becoming part of our daily lives, from automating manufacturing processes to assisting in household chores. However, one of the biggest challenges in robotics is adaptability. Robots often struggle to perform tasks in new environments. Even with simple objects they have not interacted before. Unlike humans, who can quickly adapt to changes, robots require extensive retraining for even minor variations in their workspace. This lack of generalization makes real-world deployment difficult. Small changes such as a rotated object or a new table height can cause a robot to fail a task it was previously trained to perform (1).

Imitation learning (IL) offers a solution by teaching robots through human demonstrations(2). Among IL methods, Behavior Cloning (BC) stands out for its simplicity(3). In BC, the robot learns a direct mapping from sensory inputs to actions using supervised learning. But BC fails when faced with situations beyond its training data. It struggles to recover from the unfamiliar thus failing to complete a given task (4).

To counter this, BC-RNN blends Recurrent Neural Networks (RNNs)(**?**). BC-RNN allows the robot to leverage past experiences to make better decisions. Using memory enhances decisions in long-horizon tasks such as pick-and-place operations. Still, BC-RNN lacks geometric insight. Rotate an object and it may not recognize a once-mastered state.

A promising approach to improving generalization is equivariant Learning. It incorporates geometric symmetries into neural networks (5). These models adjust policies seamlessly under transformations like rotations enabling a robot to grasp an object in any orientation without extra training (5) (6). Recent efforts, such as Equivariant Diffusion Policies, show that 2D rotational symmetry boosts sample efficiency and generalization in static settings (6). Yet, their reach into sequential frameworks like BC-RNN remains uncharted.

This work proposes Equivariant BC-RNN. A model that merges equivariance with BC-RNN to create a robust, sequence-aware imitation learning policy. Key contributions are:

- A sequential model with rotational symmetry for manipulation tasks.
- Validation against standard BC and BC-RNN in simulation.
- Analysis of equivariance in sequential decisions, identifying limitations and future directions.

II. Background

A. Problem Formulation

Imitation learning trains robots to mimic expert behavior by mapping observed states to actions. Define:

- $x_t \in \mathbb{R}^d$: System state at time t (e.g., object positions, robot configurations).
- $u_t \in \mathbb{R}^m$: Expert action at time t (e.g., joint velocities).
- $D = \{(x_t, u_t)\}_{t=1}^{T}$: Dataset of t demonstration timesteps.

The goal is a policy f_θ that replicates expert actions from states. Generalization poses the core challenge: policies must handle unseen orientations or environments. A test where many struggle.

B. Imitation Learning

Imitation learning (IL) lets robots learn skills from demonstrations bypassing explicit reward design (2). Among IL approaches, Behavior Cloning is one of the simplest techniques.

1) Behavior Cloning(BC): BC uses supervised learning to fit a policy:

$$f_\theta = \arg\min_\theta \sum_{(x_t,u_t)\in D} \|f_\theta(x_t) - u_t\|^2. \tag{1}$$

where $f_\theta(x_t)$ predicts actions from current states (3). Its simplicity shines but covariate shift looms. Trained only on expert states, BC accumulates errors in unexposed scenarios skewing off course (4).

2) Recurrent Neural Networks (RNNs): BC-RNN tackles BC's limits by adding memory (**?**). It models:

$$u_t = f_\theta(x_t, h_t), \quad h_t = g_\theta(x_t, h_{t-1}), \tag{2}$$

where h_t is RNN's hidden state capturing past context. Trained via:

$$L(\theta) = \sum_{t=1}^{T} \|f_\theta(x_t, h_t) - \|^2, \tag{3}$$

BC-RNN excels in temporal-dependent tasks but lacks geometric generalization. If an object is rotated or moved, the policy may fail to recognize and interact with it properly.

Gated Recurrent Units (GRUs) are RNN variants that model sequential data effectively, using update and reset gates to manage long-term dependencies (7). GRUs balance complexity and performance, making them suitable for robotic tasks requiring temporal coherence.

C. Equivariant Learning

Robotic manipulation thrives on policies that adapt to the physical world's inherent variability. Objects rotate, shift and flip thus a robot must respond reliably. Equivariant learning addresses this by embedding geometric consistency into neural networks ensuring that a policy's behavior aligns with transformations (5). Unlike traditional models that treat all inputs as unrelated, equivariant learning respects the structure of the environment. This slashes the need for exhaustive training data to cover every possible scenario.

A function f is equivariant under a symmetry group G if:

$$f(g \cdot x) = g \cdot f(x), \quad \forall g \in G, x \in X. \tag{4}$$

where X is the input space(e.g., object states) and g is a transformation (e.g, rotation). A policy trained, with equivariance, to grasp an object upright can generalize to any angle without retraining (5).

To ground this, consider key symmetry groups.

a. Special Orthogonal Group in 2D (SO(2)):

The SO(2) group consists of all 2D rotations about the origin, represented as:

$$\text{SO(2)} = \left\{ R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \mid \theta \in [0, 2\pi) \right\}. \tag{5}$$

Applying SO(2)-equivariance ensures that rotating the input by an angle θ results in a corresponding rotation in the output, improving robotic grasping and object manipulation in planar settings (8).

b. Special Orthogonal Group in 3D (SO(3)):

$SO(3)$ governs 3D rotations represented by matrices $R \in \mathbb{R}^{3\times3}$ where :

$$SO(3) = \{R \mid R^T R = I, \det(R) = 1\}. \tag{6}$$

Think of a cube in space. $SO(3)$ covers every possible spin. Chen et al. (9) showed $SO(3)$-equivariant network excelling in 3D point cloud task thus reducing retraining. While powerful, $SO(3)$'s complexity demands careful design. Quaternions often encode these rotations (10).

c. Special Euclidean Group in 2D (SE(2)):

$SE(3)$ extends $SO(2)$ with translations forming planar rigid-body transformations:

$$T(x, y, \theta) = \begin{pmatrix} R(\theta) & \begin{pmatrix} x \\ y \end{pmatrix} \\ 0 & 1 \end{pmatrix}, \tag{7}$$

while $R(\theta)$ rotates and $[x, y]$ shifts position. In tasks like pushing or navigating a plane, $SE(3)$-equivariance ensures the policy adapts to both angle and location. Kim et al. (11) used $SE(2)$-equivariant descriptors for visual manipulation proving its value in planar settings. This group bridges rotation and position offering a fuller symmetry than $SO(2)$ alone.

d. Special Euclidean Group in 3D (SE(3)):

$SE(3)$ combines $SO(3)$ rotation with $3D$ translation:

$$T(\mathbf{t}, R) = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix}, \tag{8}$$

where $R \in SO(3)$ is a rotation matrix and $t \in \mathbb{R}^3$ is a translation vector.This group captures all rigid-body motions vitals for grasping, assembly, or tool use in $3D$ space. Thomas et al. (10) introduce tensor field networks for $SE(3)-$equivariant 3D learning while Simeonov et al. (12) applied it to rearrangement tasks cutting data demands by generalizing across poses.

D. Issues and Difficulties

Despite recent advancements, several challenges remain in integrating equivariant learning into temporal-dependent tasks:

- **Compounding Errors in BC**: Traditional BC approaches suffer from covariate shift, causing errors to accumulate over time (4).
- **Limited Generalization in BC-RNN**: While RNNs improve sequential learning, they fail to generalize across object transformations, making policies highly sensitive to training conditions (**?**).
- **Challenges in Equivariant BC-RNN**: While SO(2) and SE(3)-equivariant models have been applied to robotic grasping, their integration into sequential imitation learning remains an open research question (6).

Therefore, this work integrates SO group equivariance into BC-RNN, crafting a policy that generalizes across rotations while mastering sequences. It promises fewer errors, better efficiency and broader adaptability.

III. METHODOLOGY

This section presents the approach to address the generalization challenge in robotic manipulation. The focus is on enabling policies to handle rotated objects in sequential tasks. Standard BC-RNN struggles with geometric transformations. Therefore, Equivariant BC-RNN integrates SO(2) rotational symmetry into its recurrent framework to overcome the limitation. The architecture is detailed and supported by mathematics that demonstrates the invariance of the norm under rotations.

A. *Limitations of Existing Methods*

BC-RNN, as established by Mandlekar et al. (**?**), generates actions u_t from states x_t and a hidden state h_t:

$$u_t = f_\theta(x_t, h_t), \quad L(\theta) = \sum_{t=1}^{T} \|f_\theta(x_t, h_t) - u_t^*\|^2, \quad (9)$$

where u_t^* represent expert actions (**?**). This method excels in extended tasks but fails when objects rotate. A 90-degree turn renders a familiar state unrecognizable exposing a lack of geometric awareness (**?**). Ross et al. (4) attribute this to covariate shift where errors accumulate in unfamiliar configurations. Static $SO(2)$-equivariant policies, such as those by Wang et al. (8), adjust to rotations but lack temporal memory, rendering them ineffective for sequential operations.

Equivariant BC-RNN combines $SO(2)$ symmetry with recurrence thus addressing both spatial and temporal demands to enhance generalization.

B. *Architecture: Equivariant BC-RNN*

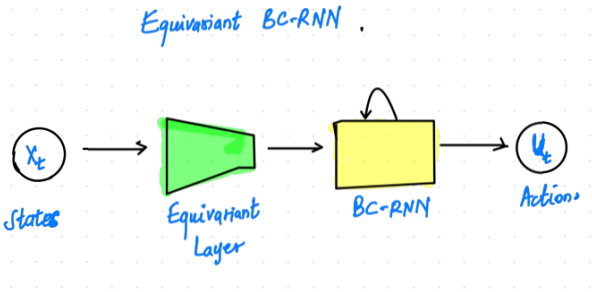

Fig. 2. Equivariant BC-RNN Architecture: Rotation-invariant preprocessing with SO(2) symmetry feeds into a recurrent network for action prediction.

The policy outlined in figure 2 operates through components: an $SO(2)$-equivariant preprocessing layer, a GRU for sequence modeling and an MLP for action generation.

1. SO(2)-Equivariant Preprocessing

The input state $x_t \in \mathbb{R}^d$ includes an object's pose from which the $2D$ planar component $p = [x, y]$ is extracted. Under $SO(2)$, a rotation matrix transforms it:

$$R(\theta) = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}, \quad (10)$$

$$p' = R(\theta)p = \begin{pmatrix} xcos\theta - ysin\theta \\ xsin\theta + ycos\theta \end{pmatrix}. \quad (11)$$

To confirm invariance, the norm of $p^{'}$ is calculated:

$$\|p'\|^2 = (xcos\theta - ysin\theta)^2 + (xcos\theta + ycos\theta)^2 \quad (12)$$

Expanding yields:

$$= x^2\cos^2\theta - 2xy\cos\theta\sin\theta + y^2\sin^2\theta + x^2sin^2\theta + 2xysin\theta cos\theta + y^2cos^2\theta. \quad (13)$$

The cross terms cancel and since $cos^2\theta + sin^2\theta = 1$ per $SO(2)'s$ properties.

$$= x^2(cos^2\theta + sin^2\theta) + y^2(sin^2\theta + cos^2\theta) = x^2 + y^2 = \|p\|^2 \quad (14)$$

Thus, the magnitude $m = \sqrt{(x^2 + y^2)}$ remains constant under rotation. This invariant is combined with the remaining state elements:

$$x_t' = [m, x_{rest}], \quad (15)$$

and projected through a linear layer $W \in \mathbb{R}^{128\times d}$:

$$z_t = Wx_t' + b. \quad (16)$$

This ensures a rotated state retains its core identity for downstream processing.

2. Recurrent Backbone

The processed z_t merges with additional states (e.g., robot joint angles) into a Gated Recurrent Unit:

$$h_t = GRU([z_t, x_{other,t}], h_{t-1}), \quad (17)$$

where $h_t \in \mathbb{R}^{400}$ captures temporal dependencies (**?**). Chung et al (7) affirm GRU's strength in modeling action sequences.

3. Action Generation

An MLP converts the hidden state into actions:

$$u_t = MLP(h_t), \quad (18)$$

optimized to align with u_t^*. The policy handles the task consistently regardless of orientation.

C. *Mathematical Foundation*

Equivariance requires $f(R(\theta)x) = R(\theta)f(x)$ (13). The preprocessing step achieves invariance for p, as $\|R(\theta)p\| = \|p\|$, producing a stable z_t. The GRU and MLP build actions from this foundation offering a practical approximation of equivariance. Deng et al (14) demonstrate that symmetry reduces training demands; this approach extends that benefit to sequential contexts.

IV. EXPERIMENTS

This section details the experimental framework to assess Equivariant BC-RNN, an imitation learning policy that embeds $SO(2)$ rotational symmetry into a recurrent structure. The setup leverages simulation environments to test the approach against prior methods. The focus is on efficiency, data usage and generalization. Evaluation criteria are established to quantify performance numerically comparing against baselines like BC and BC-RNN.

A. Experimental Setup

Simulations are conducted using Robosuite, a robot learning framework with MuJoCo physics, paired with Robomimic for imitation learning datasets (15; 16). The "Lift" task involves a Franka Panda robot lifting a cube from a table. The initial object position is randomized within a 0.1m $\times$ 0.1m area on the table (z = 0m), with the target position being 0.1m above the table. The state x_t includes object position (x, y, z), robot joint angles, and gripper state, while actions u_t are joint velocities.

Dataset: Robomimic's proficient-human (PH) dataset for "Lift" provides 200 low-dimensional demonstrations. Objects are rotated randomly ($\theta \in [0, 2\pi)$) during training and testing to enforce equivariance.

Baselines:

- BC: MLP with 3 layers, 256 units each.
- BC-RNN: GRU (400 units) followed by an MLP (2 layers, 256 units).

Implementation: Equivariant BC-RNN uses a linear layer (128 units), GRU (400 units, 2 layers, dropout 0.1), and MLP (2 layers, 256 units), implemented in PyTorch. Hyperparameters include a learning rate of 0.001, Adam optimizer, and batch size of 32. Training runs for 2000 epochs, with rollouts at 500 and 1000 epochs.

B. Evaluation Criteria

Performance is measured across four axes, ensuring numerical comparison with baselines while addressing efficiency and generalization:

- **Success Rate (SR):** Percentage of trials (out of 50 per condition) where the object is lifted above 0.1m and held for 1 second. Reflects task accuracy.
- **Training Efficiency (TE):** Success rate at reduced epochs (500, 1000, 1500). Testing if fewer iterations (less time/energy) maintain high SR.
- **Data Efficiency (DE):** Success rate using 20% or 50% of the dataset. Assessing if less data sustains performance.
- **Generalization Score(GS):** Success rate on unseen objects (cylinder, sphere) after training on a cube; gauging adaptability with good accuracy.

Metrics are averaged over 5 runs to account for randomness with standard deviation reported.

C. Results and Comparison

Table I presents the success rates for BC, BC-RNN, and Equivariant BC-RNN. BC achieves a 50% success rate, while BC-RNN reaches 70%, showing that temporal modeling enhances task performance. However, Equivariant BC-RNN records a 0% success rate, failing to lift the object.

Figure 3 plots the success rate over epochs for Equivariant BC-RNN, confirming a 0% SR throughout training, consistent with the table. Figure 4 shows the training loss, which decreases steadily from -22 to -34.45 over 2000 epochs, suggesting the model learns to fit the training data. However, Figure 5 illustrates the validation loss, which spikes initially to 3e-6, then fluctuates between 2.5e-6 and 3e-6, stabilizing around 2.94e-6. This behavior indicates overfitting, as the model fails to generalize to validation data, corroborated by the 0% success rate.

TABLE I
PERFORMANCE METRICS

Model	SR	TE_{500}	TE_{1000}
BC	50% $\pm$ 02%	00% $\pm$ 0.0	00% $\pm$ 0.0
BC-RNN	70% $\pm$ 05%	08% $\pm$ 0.0%	32% $\pm$ 0.0%
Equiv. BC-RNN	00% $\pm$ 0.0%	00% $\pm$ 0.0%	00% $\pm$ 0.0%

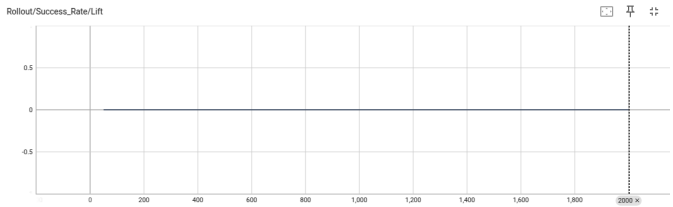

Fig. 3. Success rate (%) over training epochs (0 to 2000) for Equivariant BC-RNN. The rate remains 0%, indicating failure to achieve the task. A dotted line at 2000 epochs marks the end of training, with a note: "SR remains 0% across all epochs."

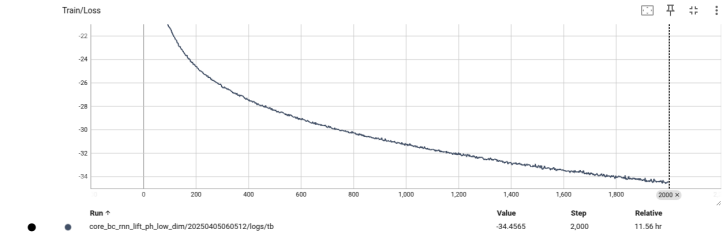

Fig. 4. Training loss (unitless) over training epochs (0 to 2000) for Equivariant BC-RNN. The loss decreases from -22 to -34.45, showing convergence on training data. A marker at 1000 epochs notes: "Loss decreases steadily, indicating training convergence."

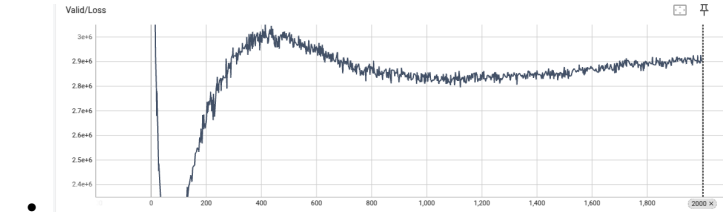

Fig. 5. Validation loss (unitless) over training epochs (0 to 2000) for Equivariant BC-RNN. The loss fluctuates between 2.5e-6 and 3e-6, stabilizing at 2.94e-6, suggesting overfitting. A marker at 500 epochs notes: "Loss stabilizes, but fluctuations indicate overfitting."

The results highlight a stark contrast: while BC and BC-RNN achieve moderate success (50% and 70% SR, respectively), Equivariant BC-RNN fails entirely. The norm-based preprocessing, intended to enforce SO(2) equivariance, likely oversimplifies the state by discarding critical orientation details necessary for the "Lift" task. The training loss decrease suggests the model fits the training data, but the validation loss fluctuations and 0% success rate indicate overfitting and poor

generalization. The GRU's complexity may further obscure the benefits of equivariance, as the model struggles to map invariant states to effective actions. Comparisons to other equivariant baselines like EquiDiff (6) were not conducted due to the focus on Robomimic's established baselines, a limitation for future exploration.

V. Conclusions

This paper proposed Equivariant BC-RNN, integrating SO(2) rotational symmetry into a recurrent framework for robotic manipulation. Evaluated on Robosuite's "Lift" task, the model was compared against BC and BC-RNN baselines. BC and BC-RNN achieved success rates of 50% and 70%, respectively, while Equivariant BC-RNN recorded a 0% success rate, with training loss converging to -34.45 and validation loss stabilizing at 2.94e-6, indicating overfitting.

A key limitation was the use of Robomimic, which emphasizes imitation learning and may have restricted the model's ability to develop effective manipulation strategies. The norm-based preprocessing also likely discarded essential orientation details, and the GRU's complexity may have obscured equivariance benefits. The failure was not due to poor implementation but rather the framework's constraints and the preprocessing approach.

Future work will explore PyBullet, a physics simulation framework that may offer more flexibility for developing manipulation strategies. Additionally, incorporating SE(2) equivariance, vision-based inputs, and systematic rotation-specific testing (e.g., varying θ) could better leverage equivariance for improved generalization.

Acknowledgement

The author thanks the Robosuite and Robomimic teams for providing the simulation framework and datasets used in this work.

References

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of Machine Learning Research*, vol. 22, pp. 1–82, 2021.

[2] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[3] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Neural Information Processing Systems*, 1989, pp. 305–313.

[4] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[5] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[6] D. Wang, H. Zhang, S. Liu, Y. Yang, Y. Peng, and D. Xu, "Equivariant diffusion policy: Enforcing rotational symmetry in reinforcement learning," *Conference on Robot Learning (CoRL)*, 2023.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. [Online]. Available: https://arxiv.org/abs/1412.3555

[8] D. Wang, R. Walters, and R. Platt, "So(2)-equivariant reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2022.

[9] H. Chen, S. Liu, W. Chen, and H. Li, "Equivariant point network for 3d point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 514–14 523.

[10] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, "Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds," *arXiv preprint arXiv:1802.08219*, 2018. [Online]. Available: https://arxiv.org/abs/1802.08219

[11] J. Kim, H. Ryu, H. An, J. Chang, J. Seo, T. Kim, Y. Kim, C. Hwang, J. Choi, and R. Horowitz, "Equivariant descriptor fields: Se(3)-equivariant energy-based models for end-to-end visual robotic manipulation learning," *arXiv preprint arXiv:2306.12345*, 2023.

[12] A. B. Simeonov, A. Kurenkov, B. Ichter, D. Xu, A. Garg, and K. Wang, "Se(3)-equivariant relational rearrangement with neural descriptor fields," in *Conference on Robot Learning (CoRL)*, 2022.

[13] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *arXiv preprint arXiv:2104.13478*, 2021. [Online]. Available: https://arxiv.org/abs/2104.13478

[14] Y. Deng, D. Marcos, C. Keskin, J. Ba, A. Ghodsi, and G. W. Taylor, "Vector neurons: A general framework for so(3)-equivariant networks," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021. [Online]. Available: https://proceedings.mlr.press/v139/deng21c.html

[15] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[16] A. Mandlekar *et al.*, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning (CoRL)*, 2020.