DEEP SE(3)-EQUIVARIANT GEOMETRIC REASONING FOR PRECISE PLACEMENT TASKS

Ben Eisner *
Carnegie Mellon University

Yi Yang, Todor Davchev, Mel Veceric, Jon Scholz
Google DeepMind

David Held
Carnegie Mellon University

ABSTRACT

Many robot manipulation tasks can be framed as geometric reasoning tasks, where an agent must be able to precisely manipulate an object into a position that satisfies the task from a set of initial conditions. Often, task success is defined based on the relationship between two objects - for instance, hanging a mug on a rack. In such cases, the solution should be equivariant to the initial position of the objects as well as the agent, and invariant to the pose of the camera. This poses a challenge for learning systems which attempt to solve this task by learning directly from high-dimensional demonstrations: the agent must learn to be both equivariant as well as precise, which can be challenging without any inductive biases about the problem. In this work, we propose a method for precise relative pose prediction which is provably SE(3)-equivariant, can be learned from only a few demonstrations, and can generalize across variations in a class of objects. We accomplish this by factoring the problem into learning an SE(3) invariant task-specific representation of the scene and then interpreting this representation with novel geometric reasoning layers which are provably SE(3) equivariant. We demonstrate that our method can yield substantially more precise predictions in simulated placement tasks than previous methods trained with the same amount of data, and can accurately represent relative placement relationships data collected from real-world demonstrations. Supplementary information and videos can be found at this URL.

1 INTRODUCTION

A critical component of many robotic manipulation tasks is deciding how objects in the scene should move to accomplish the task. Many tasks are based on the relative relationship between a set of objects, sometimes referred to as "relative placement" tasks (Simeonov et al. (2022); Pan et al. (2023); Simeonov et al. (2023); Liu et al. (2022)). For instance, in order for a robot to set a table, the position of the silverware has a desired relationship relative to the plate. These types of problems can be described as geometric reasoning tasks - if you know the locations of each object in the scene, and you know what kind of relationship the objects should have to accomplish the task, then you can logically infer the target position of the objects. We would like our robotic agents to also possess these geometric reasoning faculties to solve such relative placement problems.

In this work, we consider the task of training robotic agents to perform relative placement tasks directly from high-dimensional inputs by watching a small number of demonstrations. Although there have been many recent advances in predicting complex robot behaviors from raw sensor observations (Lee et al. (2020); Miki et al. (2022); Yang et al. (2023); Ha & Song (2022); Akkaya et al. (2019)), high-dimensional observations pose particular challenges in geometric reasoning tasks. For example, the amount by which each object should be moved to reach the goal configuration should be SE(3)-equivariant to the initial locations of the object. However, prior work (Pan et al. (2023)) demonstrates that if a general-purpose neural network is trained on a small number of high-dimensional demonstrations with no additional inductive biases, it will typically not learn to be

*Corresponding author: baeisner@andrew.cmu.edu

robust to the initial object configurations. Prior work to address this issue (Pan et al. (2023)) incorporates geometric reasoning into the network, but is not provably equivariant. Other work that is provably equivariant (Simeonov et al. (2022)) is outperformed both by (Pan et al. (2023)) and by our method.

To achieve both strong empirical performance and provable equivariance, we propose a visual representation which can be used for equivariant manipulation in relative placement tasks. Our key insight is to decouple representation learning into an invariant representation learning step and an equivariant reasoning step. We achieve this using a novel layer for differentiable multilateration, inspired by work in true-range multilateration (Zhou (2009)). In this work, we propose the following contributions:

- A novel dense representation for relative object placement tasks, which is geometrically interpretable and fully invariant under SE(3) transformations to objects in the scene.
- A method for solving relative placement tasks with differentiable geometric reasoning using a novel layer for differentiable multilateration, which can be trained end-to-end from observations of a small number of demonstrations without additional labels.
- A **provably SE(3)-equivariant** neural network architecture predicting this representation directly from high-dimensional raw inputs.
- A set of simulated experiments which demonstrate superior placement performance in several relative placement tasks - both on precision metrics and on overall success. Our experiments also demonstrate that our method generalizes within a class of objects with reasonable variation, and can be applied to real-world demonstration datasets.

2 Related Work

Object Pose Estimation: One approach to solving relative placement tasks is via object pose estimation. Recently several approaches have proposed using test-time optimization or correspondence detection to align current observations with demonstration observations with known pose (Florence et al. (2018); Simeonov et al. (2023; 2022)). However, (Simeonov et al. (2022)) requires one of the objects to remain in a fixed location, while (Simeonov et al. (2023)) requires user input to specify relationships of interest - our method has no such restriction and requires no user input. Most similar to our work is TAX-Pose (Pan et al. (2023)), which computes cross-object correspondences to estimate a task-specific alignment between a pair of objects, correcting these correspondences with a learned residual. Both TAX-Pose and our work attempt to learn to predict, for each point on an object, where that point should end up to accomplish the relative placement task. However, we compute this mapping quite differently - while TAX-Pose establishes "cross-correspondence" and updates with a residual vector to regress the next point's location, we learn an SE(3)-invariant representation (which we call RelDist) of the pair of objects, and infer the point's location from this representation using differentiable multilateration. By construction, our framework is fully $SE(3)$-invariant by construction, whereas TAX-Pose is not.

SE($\mathbf{3}$) Equivariance in Visual Prediction: Equivariance of geometric predictions under $SE(3)$ transformations of objects in a scene (including invariance to camera transformations) is a desirable property of many visual prediction systems. Several works design provably equivariant prediction methods (Cohen & Welling (2016); Weiler & Cesa (2019); van der Pol et al. (2020)) and use such methods for robot manipulation (Wang et al. (2022b); Huang et al. (2022); Wang & Walters (2022); Wang et al. (2022a)). Vector Neurons (Deng et al. (2021)) are used to design provably-SE(3) invariant and equivariant versions of standard point cloud analysis architectures like DGCNN (Wang et al. (2019)) and PointNet (Qi et al. (2017)), which we incorporate in the feature-encoder in this work. Neural Descriptor Fields (NDF) (Simeonov et al. (2022)) also use Vector Neurons (Deng et al. (2021)) to achieve provable equivariance, but suffer from suboptimal performance; NDF is outperformed by TAXPose (Pan et al. (2023)) which we compare to in this work. Other methods attempt to learn equivariance via training (Pan et al. (2023)). Another approach is to search over $SO(2)$ or $SO(3)$ (Zeng et al. (2021); Lin et al. (2023)), evaluating candidates with a scoring network to align objects in a scene for a given task. However, such methods have not been demonstrated to perform the precise SE(3) relative placement tasks used in this work.

3 BACKGROUND

We consider deep geometric reasoning tasks based on point cloud observations, and build on top of several separate lines of prior work:

Multilateration: Given a receiver with an unknown position p in 3D, a set of K beacons with known positions q_k in 3D space, and a set of K measurements r_k of the scalar distance from each beacon to the receiver, the multilateration task is to estimate the position of the receiver p. Multilateration is similar to triangulation, except that the direction of the beacons to the receiver is unknown. If the squared error cost is used, this reasoning task reduces to a non-linear least-squares optimization:

$$\min_{p} \quad \sum_{k=1}^{K} \left(||q_k - p||_2^2 - r_k^2 \right)^2 \tag{1}$$

There are several different approaches to solving this problem, as discussed in a survey paper on multilateration (Sirola (2010)). We leverage the closed-form solution to this problem; see (Zhou (2009)) for details.

The orthogonal Procrustes problem: Given two sets of N corresponding 3D points A and B, the orthogonal Procrustes problem is to find a rigid transform $T_{AB} \in (R_{AB}, t_{AB})$ which best aligns them. Specifically, if the squared error cost is used, this reasoning task reduces to a constrained least-squares optimization task:

$$\min_{R_{AB}, t_{AB}} \quad \sum_{i=1}^{N} ||R_{AB} A_i + t_{AB} - B_i||_2^2, \qquad \text{s.t.} \quad R_{AB} \in SO(3) \tag{2}$$

This problem has a well-known closed-form solution based on Singular Value Decomposition (SVD) (Sorkine-Hornung & Rabinovich (2017)).

Equivariance/Invariance: This work considers two geometric functional properties: $SE(3)$ equivariance (Eq. 3) and invariance (Eq. 4), under transform $T \in SE(3)$, which respectively satisfy:

$$f(x) = y \implies f(Tx) = Ty \tag{3}$$

$$f(x) = y \implies f(Tx) = y \tag{4}$$

Point Cloud Analysis Architectures: We make use of Dynamic Graph Convolutional Neural Networks (DGCNN) (Wang & Solomon (2019)) as the primary backbone architecture for learning per-point features. Roughly, each layer of DGCNN performs a two-step operation: 1) it computes a per-point connectivity graph, roughly its neighborhood (in either Euclidean space or latent space), typically using K-Nearest Neighbors; and 2) for each point, performs some sort of learned aggregation of the local neighborhood, which is permutation-invariant. Stacking such layers, an architecture can flexibly capture global and local context of a point cloud.

On top of DGCNN, we also make use of Vector Neurons (Deng et al. (2021)), which is a set of deep learning primitives which are designed to be SE(3)-Equivariant. These primitives can be directly dropped into DGCNN to replace existing layers. Roughly, Vector Neurons accomplish SE(3) equivariance by treating each point's Euclidean coordinate independently (through a batching operation), and constructing primitives that operate on the coordinates independently.

4 PROBLEM STATEMENT

Relative Placement Tasks: In this work, we consider relative placement tasks, as defined in prior work (Simeonov et al. (2022); Pan et al. (2023); Simeonov et al. (2023)). We borrow mathematical definitions of the task from (Pan et al. (2023)). Relative placement tasks require predicting a rigid body transform which transports a specific rigid object $\mathcal{A}$ from an initial position to a desired, task-specific final position with respect to another rigid object $\mathcal{B}$. Suppose $\mathbf{T}^*_{\mathcal{A}}$ and $\mathbf{T}^*_{\mathcal{B}}$ are $SE(3)$ poses for objects $\mathcal{A}$ and $\mathcal{B}$ respectively such that the placement task is

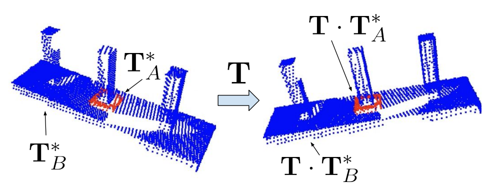


Figure 1: Invariance of relative placement tasks under transformations. In this case, a ring on peg maintains the same relative position under a rigid transformation $\mathbf{T}$.

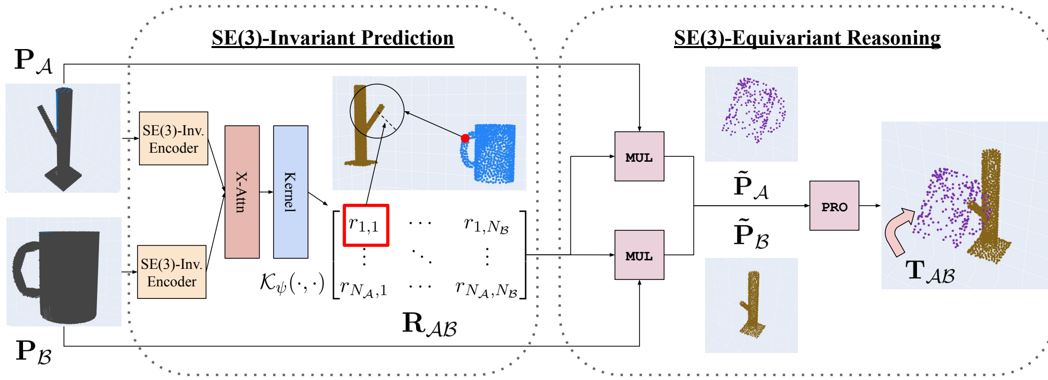


Figure 2: Method overview. First, the point clouds $\mathbf{P}_\mathcal{A}, \mathbf{P}_\mathcal{B}$ are each encoded with a dense $SE(3)$-equivariant encoder, after which cross-attention is applied to yield task-specific dense representations. Then, the kernel matrix $\mathbf{R}_{\mathcal{AB}}$ is constructed through the learned kernel $\mathcal{K}_\psi$. This matrix is then passed into `MUL` to infer the desired final point clouds, and then passed into `PRO` to extract a final transform which moves object $\mathcal{A}$ into its goal position.

complete, expressed with respect to some arbitrary world frame. Following Pan et al. (2023), the task is also considered complete if $\mathcal{A}$ and $\mathcal{B}$ are positioned such that, for some $\mathbf{T} \in SE(3)$:

$$\text{RelPlace}(\mathbf{T}_\mathcal{A}, \mathbf{T}_\mathcal{B}) = \textsc{Success} \text{ iff } \exists \mathbf{T} \in SE(3) \text{ s.t. } \mathbf{T}_\mathcal{A} = \mathbf{T} \circ \mathbf{T}^*_\mathcal{A} \text{ and } \mathbf{T}_\mathcal{B} = \mathbf{T} \circ \mathbf{T}^*_\mathcal{B}. \tag{5}$$

Cross-Pose: We are interested in predicting a transform which brings object $\mathcal{A}$ to a goal position relative to object $\mathcal{B}$, to satisfy the placement condition defined in Eq. 5. Rather than estimating each object's pose independently and computing a desired relative transform, we would like to learn a function $\mathbf{T}_{\mathcal{AB}} = f(\mathbf{P}_\mathcal{A}, \mathbf{P}_\mathcal{B})$ which takes as input point cloud observations of the two objects, $\mathbf{P}_\mathcal{A}, \mathbf{P}_\mathcal{B}$ and outputs an $SE(3)$ transform which would transport $\mathbf{P}_\mathcal{A}$ into their desired goal position with respect to $\mathcal{B}$. Pan et al. (2023) define this transform as the **cross-pose** for a specific instance of a relative placement task, as follows: when objects $\mathcal{A}$ and $\mathcal{B}$ have been rigidly transformed from their goal configuration by $\mathbf{T}_\alpha$ and $\mathbf{T}_\beta$, respectively, the cross-pose is defined as:

$$f(\mathbf{T}_\alpha \circ \mathbf{T}^*_\mathcal{A}, \mathbf{T}_\beta \circ \mathbf{T}^*_\mathcal{B}) = \mathbf{T}_{\mathcal{AB}} := \mathbf{T}_\beta \circ \mathbf{T}_\alpha^{-1} \tag{6}$$

meaning that applying this transformation to object $\mathcal{A}$ will then transform it back into a goal configuration relative to object $\mathcal{B}$. Note that the cross-pose function is **equivariant** in the sense that $f(T_\mathcal{A} \circ \mathbf{P}_\mathcal{A}, T_\mathcal{B} \circ \mathbf{P}_\mathcal{B}) = T_\mathcal{B} \circ f(\mathbf{P}_\mathcal{A}, \mathbf{P}_\mathcal{B}) \circ T_\mathcal{A}^{-1}$.

Our method assumes that we receive point cloud observations of the scene (e.g. from LiDAR/depth cameras/stereo), and that the pair of objects being manipulated (e.g. the mug and the rack) have been segmented. For training the cross-pose prediction module, we assume access to a set of demonstrations of pairs of objects in their final goal configuration (e.g. demonstrations in which objects $\mathcal{A}$ and $\mathcal{B}$ have poses $\mathbf{T}_\mathcal{A}$, $\mathbf{T}_\mathcal{B}$ such that $\text{RelPlace}(\mathbf{T}_\mathcal{A}, \mathbf{T}_\mathcal{B}) = \textsc{Success}$, according to Eqn. 6).

5 A Framework for SE(3) Equivariant Manipulation

Method Overview: We now describe our method for precise relative object placement which is provably $SE(3)$-equivariant under transformations to individual objects. At a high level, we: 1) Generate SE(3)-invariant per-point features for objects in the scene; 2) Use cross-attention to augment these features with task-specific information; 3) Apply a learned kernel function, which takes in a pair of points on two different objects and predicts how far apart they should be to satisfy the task; 4) Use a differentiable closed-form estimator of the least-squares solution to the multilateration problem to estimate the desired task-specific final location of each point; and 5) Compute the least-squares transform between the estimated point locations and the input locations, using differentiable SVD. See Figure 2 for a high-level overview of our approach.

5.1 RELDIST : AN SE(3)-INVARIANT REPRESENTATION FOR RELATIVE PLACEMENT

The key insight of this work for solving relative placement tasks is as follows: suppose that we wish to predict the desired goal position of object $\mathcal{A}$, which should be placed in some location relative to object $\mathcal{B}$. One approach is to predict the desired pose of object $\mathcal{A}$ directly, relative to object $\mathcal{B}$; previous work Pan et al. (2023) shows that this approach has poor performance. Another approach used in prior work that performs better Pan et al. (2023) is to predict the desired location of each point $p_i^{\mathcal{A}}$ in object $\mathcal{A}$. Instead, our approach is to predict a set of desired distance relationships between $p_i^{\mathcal{A}}$ and at least 3 points on object $\mathcal{B}$. Specifically, for each point $p_i^{\mathcal{A}}$ in object $\mathcal{A}$, and for some set of points $\{p_1^{\mathcal{B}}, \ldots, p_K^{\mathcal{B}}\}$ on object $\mathcal{B}$, we predict the desired Euclidean distance $r_{ij} = d(p_i^{\mathcal{A}}, p_j^{\mathcal{B}})$ between points $p_i^{\mathcal{A}}$ and $p_j^{\mathcal{B}}$ in the desired goal configuration. As long as the number of points K used from object $\mathcal{B}$ is at least 3, we can use multilateration (Section 3) to estimate the desired location of $p_i^{\mathcal{A}}$.

Note that these desired distance between point $p_i^{\mathcal{A}}$ on object $\mathcal{A}$ and some point $p_j^{\mathcal{B}}$ on object $\mathcal{B}$ is invariant to the current poses of objects $\mathcal{A}$ and $\mathcal{B}$. Specifically, if object $\mathcal{A}$ is transformed by $\mathbf{T}_\alpha$ and object $\mathcal{B}$ is transformed by $\mathbf{T}_\beta$, then the desired distance between these points is constant:

$$r_{ij} = d(p_i^{\mathcal{A}}, p_j^{\mathcal{B}}) = d(\mathbf{T}_\alpha \circ p_i^{\mathcal{A}}, \mathbf{T}_\beta \circ p_j^{\mathcal{B}}) \tag{7}$$

Further, these distances are scalar values, so they are not defined in any reference frame, which means that they are invariant to shifts in the origin of the coordinate system. We refer to r_{ij} as **RelDist**. This representation gives a convenient target for neural network prediction - one simply needs to predict a set of pairwise invariant relationships $d(p_i^{\mathcal{A}}, p_j^{\mathcal{B}})$ across two objects.

5.2 PREDICTING **RELDIST**

We now describe a provably SE(3)-invariant architecture for predicting RelDist :

Dense SE(3)-Invariant Features: We assume that we are given a point cloud for objects $\mathcal{A}$ and $\mathcal{B}$, given by $\mathbf{P}_{\mathcal{A}} \in \mathbb{R}^{N_{\mathcal{A}} \times 3}$ and $\mathbf{P}_{\mathcal{B}} \in \mathbb{R}^{N_{\mathcal{B}} \times 3}$, respectively. The first step of our approach is to compute an invariant feature for each point in these point clouds. In other words, the feature should not change if either object is transformed by an SE(3) transformation. Further, we wish each point to have a unique feature. Given point clouds $\mathbf{P}_{\mathcal{A}}$ and $\mathbf{P}_{\mathcal{B}}$, we can use a provably $SE(3)$-Invariant point cloud neural network such as DGCNN Wang & Solomon (2019) implemented using Vector Neuron layers Deng et al. (2021). Alternatively, one could train a standard DGCNN to produce $SE(3)$-Invariant representations, although the resulting features would not be guaranteed to be invariant. We consider both approaches in our experiments.

Specifically, we learn functions $f_{\theta_{\mathcal{A}}}$ and $f_{\theta_{\mathcal{B}}}$ which map the point clouds $\mathbf{P}_{\mathcal{A}}$ and $\mathbf{P}_{\mathcal{B}}$ to a set of invariant d-dimensional per-point features $\Phi_{\mathcal{A}}$ and $\Phi_{\mathcal{B}}$:

$$\Phi_{\mathcal{A}} = f_{\theta_{\mathcal{A}}}(\mathbf{P}_{\mathcal{A}}) \in \mathbb{R}^{N_{\mathcal{A}} \times d}, \quad \Phi_{\mathcal{B}} = f_{\theta_{\mathcal{B}}}(\mathbf{P}_{\mathcal{B}}) \in \mathbb{R}^{N_{\mathcal{B}} \times d} \tag{8}$$

Task-Specific Cross-Attention: Although the representations $\Phi_{\mathcal{A}}, \Phi_{\mathcal{B}}$ are object specific, the relative placement task requires reasoning about cross-object relationships. For instance, for hanging a mug on a rack, the network needs to reason about the relative location of the mug handle and the peg on the rack. To reason about these cross-object relationships, we learn a cross-attention module, using multi-head cross attention Vaswani et al. (2017), which we sum with the pre-attention embeddings from Eq. 8:

$$\Phi'_{\mathcal{A}}, \Phi'_{\mathcal{B}} = \text{X-Attn}_\gamma(\Phi_{\mathcal{A}}, \Phi_{\mathcal{B}}) \tag{9}$$

$$\hat{\Phi}_{\mathcal{A}} = \Phi_{\mathcal{A}} + \Phi'_{\mathcal{A}}, \quad \hat{\Phi}_{\mathcal{B}} = \Phi_{\mathcal{B}} + \Phi'_{\mathcal{B}} \tag{10}$$

A Kernel Function for predicting RelDist : Let $\phi_i^{\mathcal{A}}$ and $\phi_j^{\mathcal{B}}$ represent the learned embeddings for point $p_i^{\mathcal{A}}$ on object $\mathcal{A}$ and point $p_j^{\mathcal{B}}$ on object $\mathcal{B}$, respectively. We now wish to define a symmetric function $\mathcal{K}$ which takes a pair of embeddings $\phi_i^{\mathcal{A}}$, $\phi_j^{\mathcal{B}}$ and computes a positive scalar $r_{ij} \in \mathbb{R}^+$ which predicts the desired distance between points $p_i^{\mathcal{A}}$ and $p_j^{\mathcal{B}}$ when objects A and B are placed in the goal configuration: $r_{ij} = \mathcal{K}(\phi_i^{\mathcal{A}}, \phi_j^{\mathcal{B}})$. This function $\mathcal{K}$ can be thought of as a type of kernel, for which there are various choices of functions. In this work, we select a simple learned kernel with the following form:

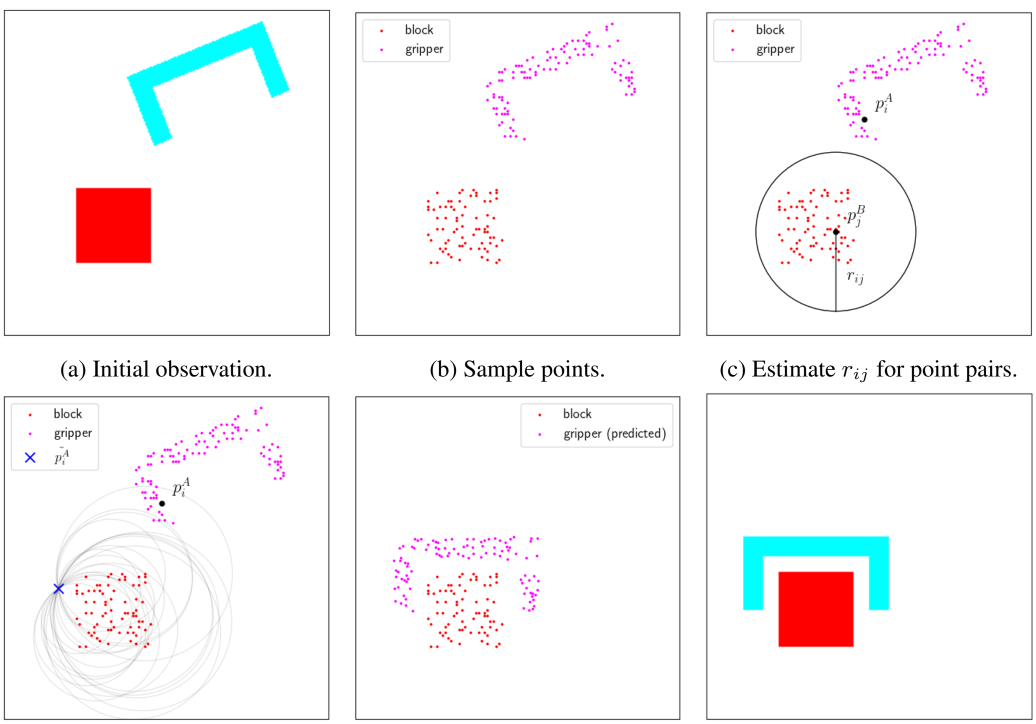
$$\mathcal{K}_\psi(\phi_i^{\mathcal{A}}, \phi_j^{\mathcal{B}}) = \text{softplus}\left(\frac{1}{2}\big(\text{MLP}_\psi(\phi_i^{\mathcal{A}}, \phi_j^{\mathcal{B}}) + \text{MLP}_\psi(\phi_j^{\mathcal{B}}, \phi_i^{\mathcal{A}})\big)\right) \tag{11}$$

where ψ are a set of learned parameters. Details can be found in the Appendix C.1. This allows us to build the kernel matrix $\mathbf{R}_{\mathcal{AB}}$:

$$\mathbf{R}_{\mathcal{AB}} = \begin{bmatrix} \mathcal{K}_\psi(\phi_1^{\mathcal{A}}, \phi_1^{\mathcal{B}}) & \cdots & \mathcal{K}_\psi(\phi_1^{\mathcal{A}}, \phi_{N_{\mathcal{B}}}^{\mathcal{B}}) \\ \vdots & \ddots & \vdots \\ \mathcal{K}_\psi(\phi_{N_{\mathcal{A}}}^{\mathcal{A}}, \phi_1^{\mathcal{B}}) & \cdots & \mathcal{K}_\psi(\phi_{N_{\mathcal{A}}}^{\mathcal{A}}, \phi_{N_{\mathcal{B}}}^{\mathcal{B}}) \end{bmatrix} \in (\mathbb{R}^+)^{(N_{\mathcal{A}} \times N_{\mathcal{B}})} \tag{12}$$

which can be interpreted exactly as the set of RelDist values for all pairs of points $p_i^{\mathcal{A}}$ and $p_j^{\mathcal{B}}$ between objects $\mathcal{A}$ and $\mathcal{B}$. Note again that this prediction is $SE(3)$-Invariant for any individual transformations of objects $\mathcal{A}$ or $\mathcal{B}$.

5.3 From RelDist to Cross-Pose



(a) Initial observation. (b) Sample points. (c) Estimate r_{ij} for point pairs.

(d) Infer $p_i'^{\mathcal{A}} = \texttt{MUL}(R_{i\mathcal{B}}, \mathbf{P}_{\mathcal{B}})$. (e) Repeat $\texttt{MUL}(R_{i\mathcal{B}}, \mathbf{P}_{\mathcal{B}})$ for all i. (f) Infer gripper goal pose.

Figure 3: Reasoning with multilateration (a) in a 2D environment with block and gripper. (b) For each sampled point on the gripper, (c) we estimate the desired distances between the gripper point and points on the block. We then use multilateration (d) to extract a least-squares solution to compute the desired gripper point location. Doing this for every point on the gripper, (e) we can reconstruct the desired position for each gripper point. (f) These corresponding points can be used to infer a rigid transform which brings the gripper to the final goal position.

We now describe how we can turn our invariant representation into a Cross-Pose prediction with the desired properties above. A visual walkthrough can be found in Figure 3.

Estimating Corresponding Points with Differentiable Multilateration: Given the estimated set of RelDist values $\mathbf{R}_{\mathcal{AB}}$, we would like to estimate the intended position of each point in object $\mathcal{A}$

with respect to object $\mathcal{B}$ (or vice versa). The ith row $R_{i\mathcal{B}}$ of $\mathbf{R}_{\mathcal{AB}}$ contains a set of scalars indicating the desired distances from points on object $\mathcal{B}$ to the point $p_i^{\mathcal{A}}$ on object $\mathcal{A}$ when placed in the goal configuration. We would like to use these relative distances to estimate the goal position of this point. Since $\mathbf{R}_{\mathcal{AB}}$ is a predicted quantity and may have some noise, we wish to find the point $p_i'^{\mathcal{A}}$ which minimizes the Mean-Squared error with respect to that row:

$$p_i'^{\mathcal{A}} = \arg\min_{p_i'^{\mathcal{A}}} \sum_{j=1}^{N_{\mathcal{B}}} \left|\left| ||p_j^{\mathcal{B}} - p_i'^{\mathcal{A}}||_2^2 - r_{ij}^2 \right|\right|_2^2 \tag{13}$$

This describes a nonlinear least-squares optimization problem, a class of problems which has no general closed-form global minimizer. However, it has been shown that there is a differentiable, closed-form global minimizer Zhou (2009) for this specific problem in Equation 13, which we will refer to as the function MUL: $p_i'^{\mathcal{A}} = \texttt{MUL}(R_{i\mathcal{B}}, \mathbf{P}_{\mathcal{B}})$. See Figures 3c and 3d for a visualization of this point estimation process, and Appendix A.1 for details. By computing this function on each row of $\mathbf{R}_{\mathcal{AB}}$, we can compute the desired goal position for each point on $\mathcal{A}$, as: $\tilde{\mathbf{P}}_{\mathcal{A}} = \{\texttt{MUL}(R_{i\mathcal{B}}, \mathbf{P}_{\mathcal{B}})\}_{i\in[N_{\mathcal{A}}]}$ (see Figure 3e). Importantly, this solution to the multilateration problem is provably SE(3)-equivariant to transformations on object $\mathcal{B}$, given a fixed $\mathbf{R}_{\mathcal{AB}}$ (see Appendix B for proof).

Estimating Cross-Pose with SVD: Given a set of initial points P_A and their corresponding estimated final task-specific goal positions $\tilde{P}_{\mathcal{A}}$, we can set up a classic weighted Procrustes problem to compute the estimated cross-pose $\mathbf{T}_{\mathcal{AB}}$ between objects $\mathcal{A}$ and $\mathcal{B}$, as defined in Section 3. This has a known closed-form solution based on the Singular Value Decomposition (SVD) (see Appendix A.2) which can be implemented to be differentiable Papadopoulo & Lourakis (2000); Pan et al. (2023). For the relative placement task, once we estimate $\mathbf{T}_{\mathcal{AB}}$, we use motion planning to move object $\mathcal{A}$ by $\mathbf{T}_{\mathcal{AB}}$ into the goal pose. Because MUL and PRO are all differentiable functions, the whole method can be trained end-to-end, using the same loss functions as defined in Pan et al. (2023) (see Appendix C for details).

6 Experiments

6.1 RLBench Tasks - Achieving Precise Placement

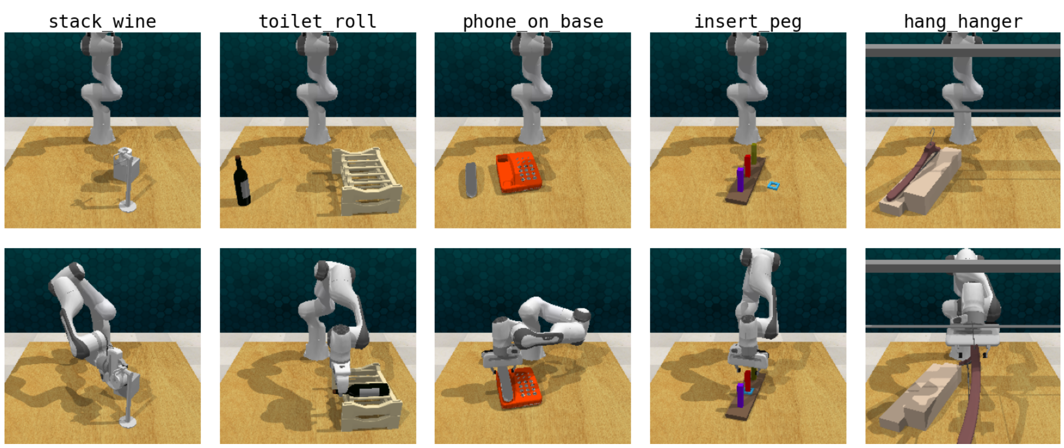


Figure 4: RLBench (James et al. (2020)) relative placement tasks. Top: the initial state of a demonstration. Bottom: the final state of a demonstration, where a successful placement has been achieved.

Experiment Setup: To assess the ability of our method to make precise predictions in relative placement tasks, we select 5 manipulation tasks tasks from the RLBench benchmark (James et al. (2020) which require varying amounts of precision to successfully accomplish each task. Specifically, we choose the following tasks: stack_wine, toilet_roll, phone_on_base, insert_peg, and hang_hanger. Each of these tasks involves placing a singular object in a task-specific configuration relative to some other object. We collect 10 expert demonstrations for each task, and extract point clouds for initial and final keyframes using ground truth segmentation from multiple simulated RGB-D images. This forms the basis for a per-task relative placement task, where the goal is to predict the desired goal configuration for each object given the initial point cloud. To evaluate the performance of a prediction system, we generate 1000 unseen initial starting positions for the object in the scene, and measure the both the rotation error ϵ_{r} and the centroid translation error ϵ_{p} of the predictions compared to the goal poses reached in expert demonstrations. We evaluate both TAX-Pose Pan et al. (2023) and our method on each task. Results are presented in Table 1.

RLBench Placement Tasks										
	`stack_wine`		`toilet_roll`		`hang_hanger`		`phone_on_base`		`insert_peg`	
Method	ϵ_{r} (°)	ϵ_{p} (m)	ϵ_{r} (°)	ϵ_{p} (m)	ϵ_{r} (°)	ϵ_{p} (m)	ϵ_{r} (°)	ϵ_{p} (m)	ϵ_{r} (°)	ϵ_{p} (m)
TAX-Pose	1.485	0.003	1.173	**0.001**	5.471	0.012	4.144	0.005	7.098	0.004
Ours	**0.764**	**0.001**	**1.150**	**0.001**	**0.624**	**0.002**	**0.804**	**0.001**	**1.209**	**0.003**

Table 1: RLBench Placement Tasks, prediction error (↓ is better): We measure the precision of our method when predicting the cross-pose which brings each object into the goal position. We report both the angular error ϵ_{r} (°)and the translational error ϵ_{p} (m), compared to the goal poses achieved by expert demonstrations.

Analysis: For every task in our evaluation, our method achieves highly precise placements, with rotational errors less than 1.25°and translational errors less than 3 millimeters (and frequently only 1mm). For each task, this level of precision is well within the tolerances for successfully completing the task. Additionally, we substantially outperform our closest baseline (TAX-Pose) in nearly every metric (which matches our translational precision in only on `toilet_roll`). In particular, our method is 2-9x more precise in rotation on 4 of the tasks. Because each task includes only a single object which varies only in initial pose, this evaluation can be interpreted roughly as a ceiling for how precisely these relative relationships can be represented. We believe that this is strong evidence that our invariant-equivariant architecture is capable of representing significantly more precise relationships in general than prior work. Furthermore, we show in Appendix D that our method can learn highly precise relationships from only a single demonstration.

6.2 NDF Tasks - Category-Level Generalization

Experiment Setup: To evaluate the ability of our method to generalize precise placements across a class of objects with reasonable variation, we evaluate our method on the NDF relative placement tasks, as proposed in Simeonov et al. (2022). These tasks have the following structure: an object is positioned in a tabletop scene, along with a robot arm with parallel-jaw gripper. The task is to predict two poses for the robot to execute: first, a **Grasp** pose which will lead to a stable grasp on the object when the gripper is actuated; second, a **Place** pose, where the grasped object should be placed to accomplish the task. At training time, the agent is given 10 demonstrations of each stage of the task (grasp and place), with a variety of objects from the same object category. The agent must then learn a relative pose prediction function for each scenario which can generalize to novel instances of objects. There are two variants of each task evaluation: when the object is initialized to be resting upright on the table, bottom-side down ("Upright"), and when the object is initialized at an arbitrary orientation above the table for the robot to grasp ("Arbitrary"). At test time, we randomly sample 100 initial configurations from each distribution and report average metrics across all trials. We report success rates at various thresholds of penetration between the placed object and the scene as a proxy for precision. See Appendix E for more details and motivation for this metric.

We compare our method to the following baselines: Dense Object Nets (Florence et al. (2018)), Neural Descriptor Fields (Simeonov et al. (2022)), and TAX-Pose (Pan et al. (2023)) (see Appendix E.2 for details).

Analysis: Our numerical results are presented in Table 3. Our method achieves substantially more precise placements in both the upright and arbitrary mug hanging tasks: while other methods perform well when geometric feasibility is not considered in the evaluation (infinity column), our method yields substantially more feasible predictions in both settings, when penalizing for collisions (penetration thresholds of 1 or 3 cm). Results for the remaining NDF tasks can be found in

Mug Hanging (Upright/Arbitrary)							
	Grasp	**Place**			**Overall**		
Method		<1cm	<3cm	∞	<1cm	<3cm	∞
DON[1]	0.91/0.35	-	-	0.50/0.45	-	-	0.45/0.17
NDF[2]	0.96/0.78	-	-	0.92/0.75	-	-	0.88/**0.58**
TAX-Pose[3]	**1.00/0.50**	0.15/0.32	0.70/**0.75**	0.93/0.85	0.15/0.14	0.70/**0.40**	0.93/0.44
Ours	**1.00**/0.43	**0.35/0.36**	**0.78**/0.66	**0.99/0.88**	**0.35/0.19**	**0.78**/0.32	**0.99**/0.41

Table 2: NDF Mug Hanging, Success Rate (↑ is better): The success rate of each method when running the relative placement tasks with the objects starting either upright on the table (left number) or in arbitrary poses above the table (right number). Each column group shows the success rates for the **Grasp** phase and **Place** phase, as well as **Overall** performance (when both grasp and place are successful in a trial). Additionally, for **Place** and **Overall** we report success when maximum penetration is thresholded at the given distance (1cm, 3cm, ∞).

Appendix E; however, our results on the Bottle and Bowl tasks are inconclusive, as implementations of the symmetry-breaking techniques proposed by Pan et al. (2023) - which are crucial to good performance for our closest baseline, TAX-Pose - were not released, and we were not able to reproduce the performance reported by Pan et al. (2023). We also performed multiple ablations to understand the effect of different design decisions of our method; see Appendix F for details.

6.3 Real-World Sensor Experiments

Experiment Description: To demonstrate the ability of our method to generalize to real-world sensors, we design an experiment that closely follows the real-world experiments proposed in Pan et al. (2023). Specifically, we choose the real-world mug-hanging experiment from this work. The authors from Pan et al. (2023) have provided us with a dataset real-world mug-hanging demonstrations, collected in the setup displayed in Figure 9. We then train our method in the same way as we did simulated mugs. See Figure 10a and 11 and Table 7 in the Appendix for qualitative and quantitative evaluations of this offline dataset. We find that our method produces predictions consistent with successful mug-hanging.

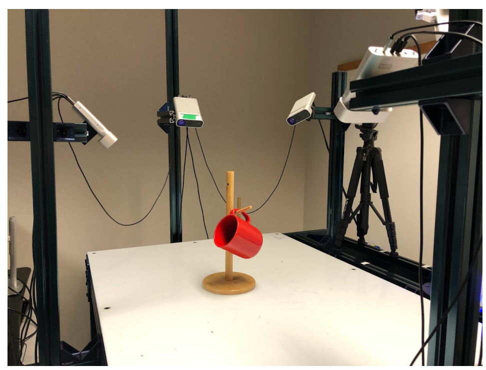


Figure 5: A real-world demonstration of mug-hanging.

7 Conclusion & Limitations

In this work, we present a provably $SE(3)$-Equivariant system for predicting task-specific object poses for relative placement tasks. We propose a novel cross-object representation, RelDist, which is an $SE(3)$-Invariant geometric representation for cross-object relationships. We also propose a pair of $SE(3)$-Equivariant geometric reasoning layers that use multilateration and SVD, respectively, to extract a relative pose prediction. Finally, we demonstrate that this representation provides superior performance in high-precision tasks in simulated environments, and is applicable to point cloud data collected in real-world manipulation demonstrations.

This method has several limitations. First, it cannot handle symmetric objects or multimodal placement tasks without an external mechanism for symmetry-breaking, as only a single pose is predicted even when multiple poses may be valid. This might be addressed by exploring generative versions of this model. Further, our method assumes that we have segmented the two task-relevant objects (e.g. the mug and the rack). Still, we hope that our method provides a foundation for future work on SE(3)-Equivariant learning for relative placement tasks.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DGE-1745016 (NSF GRFP), as well as by NIST under Grant No. 70NANB23H178.

References

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, October 2021.

Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto (eds.), *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 373–385. PMLR, 2018.

Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pp. 24–33. PMLR, 2022.

Haojie Huang, Dian Wang, Robin Walters, and Robert Platt. Equivariant transporter network. *arXiv preprint arXiv:2202.09400*, 2022.

S James, Z Ma, D R Arrojo, and A J Davison. RLBench: The robot learning benchmark learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, April 2020.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

Yen-Chen Lin, Pete Florence, Andy Zeng, Jonathan T Barron, Yilun Du, Wei-Chiu Ma, Anthony Simeonov, Alberto Rodriguez Garcia, and Phillip Isola. MIRA: Mental imagery for robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1916–1927. PMLR, 2023.

Weiyu Liu, Tucker Hermans, Sonia Chernova, and Chris Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv preprint arXiv:2211.04604*, 2022.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. TAX-Pose: Task-Specific Cross-Pose estimation for robot manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1783–1792. PMLR, 2023.

Théodore Papadopoulo and Manolis IA Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Computer Vision-ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part I 6*, pp. 554–570. Springer, 2000.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: SE(3)-Equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6394–6400, May 2022.

Anthony Simeonov, Yilun Du, Yen-Chen Lin, Alberto Rodriguez Garcia, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Pulkit Agrawal. SE(3)-Equivariant relational rearrangement with neural descriptor fields. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 835–846. PMLR, 2023.

Niilo Sirola. Closed-form algorithms in mobile positioning: Myths and misconceptions. In *2010 7th Workshop on Positioning, Navigation and Communication*, pp. 38–44, March 2010.

Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1):1–5, 2017.

Elise van der Pol, Thomas Kipf, Frans A Oliehoek, and Max Welling. Plannable approximations to mdp homomorphisms: Equivariance under actions. *arXiv preprint arXiv:2002.11963*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł Ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I Guyon, U Von Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Dian Wang and Robin Walters. So (2) equivariant reinforcement learning. In *International Conference on Learning Representations*, 2022.

Dian Wang, Jung Yeon Park, Neel Sortur, Lawson LS Wong, Robin Walters, and Robert Platt. The surprising effectiveness of equivariant models in domains with latent symmetry. *arXiv preprint arXiv:2211.09231*, 2022a.

Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant q learning in spatial action spaces. In *Conference on Robot Learning*, pp. 1713–1723. PMLR, 2022b.

Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3523–3532, 2019.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):1–12, October 2019.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019.

Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1430–1440, 2023.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. In Jens Kober, Fabio Ramos, and Claire Tomlin (eds.), *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 726–747. PMLR, 2021.

Yu Zhou. An efficient least-squares trilateration algorithm for mobile robot localization. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3474–3479, October 2009.