arXiv:2504.06961v2 [cs.RO] 15 Aug 2026

Two by Two ✌: Learning Multi-Task Pairwise Objects Assembly for Generalizable Robot Manipulation

Yu Qi [2, 1*] Yuanchen Ju[1, 3*] Tianming Wei [1, 4] Chi Chu [1] Lawson L.S. Wong [2] Huazhe Xu [1, 3, 5 †]
[1] Shanghai Qi Zhi Institute [2] Northeastern University [3] IIIS, Tsinghua University
[4] Shanghai Jiao Tong University [5] Shanghai AI Laboratory



Figure 1. **Overview of the *2BY2* Dataset.** We propose the first large-scale daily pairwise object assembly dataset **2BY2**, which contains 1,034 instances and 517 pairwise objects with pose and symmetry annotations.

Abstract

*3D assembly tasks, such as furniture assembly and component fitting, play a crucial role in daily life and represent essential capabilities for future home robots. Existing benchmarks and datasets predominantly focus on assembling geometric fragments or factory parts, which fall short in addressing the complexities of everyday object interactions and assemblies. To bridge this gap, we present **2BY2**, a large-scale annotated dataset for daily pairwise objects assembly, covering 18 fine-grained tasks that reflect real-life scenarios, such as plugging into sockets, arranging flowers in vases, and inserting bread into toasters. 2BY2 dataset includes 1,034 instances and 517 pairwise objects with pose and symmetry annotations, requiring approaches that align geometric shapes while accounting for functional and spatial relationships between objects. Leveraging the 2BY2 dataset, we propose a two-step SE(3) pose estimation method with equivariant features for assembly constraints. Compared to previous shape assembly methods, our approach achieves state-of-the-art performance across all 18 tasks in the 2BY2 dataset. Additionally, robot experiments further validate the reliability and generalization ability of our method for complex 3D assembly tasks.*

1. Introduction

Assembly tasks are ubiquitous, such as assembling furniture, repairing household appliances, or putting together electronics. Successfully completing these tasks requires precise reasoning about the spatial relationships between pairs of objects. For robots to assist in these activities, they need to accurately estimate the 6D pose of each objects—including both their orientation and position in space. This capability is essential for domestic robots to help humans with various tasks, as it enables them to interact with their environment in a meaningful way.

Daily object pairwise assembly not only requires considering the geometric constraints and spatial relationships between objects to achieve precise alignment but also

needs to exhibit a certain level of generalization. Existing methods and benchmarks for solving assembly problems [35, 36, 56, 57, 70], typically focus on matching local geometric shapes, which often results in suboptimal performance in everyday assembly scenarios which require semantic and spatial alignment. This is because they are primarily trained and tested on existing assembly datasets that consist of large-scale geometric fragments, such as Breaking Bad [46] and Neural Shape Mating [7]. Compared with existing assembly tasks that focus on putting together fractures of objects, daily pairwise assembly tasks are more challenging and hold greater practical significance in human life.

To bridge this gap, we introduce ***2BY2***, the first large-scale daily pairwise assembly dataset comprising 18 fine-grained tasks, shown as Figure 1. Compared to previous datasets and benchmarks, *2BY2* contains 1,034 instances and 517 pairwise objects with pose and symmetry annotations, covering a variety of pairwise assembly tasks reflecting everyday scenarios, which require approaches that align geometric shapes while accounting for functional and spatial relationships between objects, as shown in Table 1.

Building on this dataset, we propose a two-step pairwise network architecture for assembly tasks. Mimicking the human assembly process like we firstly put the vase on the table and then arrange flower in it, our approach predicts the pose of each object in a step-by-step manner to assemble them to a predefined canonical space, which refers to a standard coordinate system that aligns with the principles of the human world, with detailed definition in Section 3.2. The network leverages a custom two-scale Vector Neuron DGCNN [11] encoder with spherical convolution [9] to extract SE(3) equivariant and SO(3) invariant features from point cloud inputs. Additionally, a feature fusion module and a two-step training and evaluation strategy are used to improve pose prediction accuracy.

We evaluate our approach on 18 tasks in *2BY2* dataset to demonstrate the effectiveness on multi-task object pairwise assembly prediction. Compared to existing baselines, our method achieves an average improvement of *0.046* in translation RMSE and *8.97* in rotation RMSE. Moreover, we validate the effectiveness of our approach on three multi-category task, namely *Lid Covering*, *Inserting* and *High Precision Placing*, as well as the *All* task, which is defined in Section 2. Besides, real-world robot experiments validate the practical applicability of our approach.

Our main contributions are listed as follows:

1. We introduce ***2BY2***, the first large-scale daily pairwise object assembly dataset. By providing comprehensive pose and symmetry annotations for 517 pairwise objects across 18 fine-grained tasks, 2BY2 pushes the boundaries of real-world 3D assembly challenges and establishes a benchmark for pairwise assembly tasks.

2. Our two-step pairwise SE(3) pose estimation method, leveraging equivariant geometric features, demonstrates superior performance compared to existing shape assembly methods, significantly reducing translation and rotation errors and enhancing the accuracy of 6D pose estimation.

3. Our approach achieves state-of-the-art performance on the benchmark, with real-world robot experiments demonstrating its capability, providing a generalizable solution for robot manipulation using pairwise object assembly.

2. Related Work

2.1. Object Assembly Benchmarks and Datasets

Object reassembly has led to various datasets in computer vision and robotics. In computer vision, datasets like AutoMate [28] and JoinABLe [58] focus on reassembling fragments using geometric clues, while early datasets [5, 15, 25, 47] were limited in scale. Recent efforts, such as Neural Shape Mating [7] and Breaking Bad [46], generate large-scale fractured object data using parametric segmentation. In robotics, benchmarks like Factory [39], RLBench [27], and RoboSuite [77] lack diverse shapes and assembly tasks under varying initial poses. In contrast, our dataset includes over 500 diverse object pairs across 3 categories and 18 assembly tasks, providing a comprehensive benchmark for pairwise object assembly, supporting the development of generalizable methods for real-world applications.

2.2. 3D Shape Assembly

3D shape assembly [14, 34, 37, 63, 68], also known as part assembly, involves reconstructing objects from fragments, such as shattered sculptures or disassembled furniture. Existing methods use graphical models [6, 26, 30] and neural networks [8, 29, 32, 54, 59, 64, 65, 67, 71, 74] to capture geometric and semantic relationships. Approaches such as [7, 40, 69, 70] focus on pose estimation and part assembly without relying on predefined semantic information. Few-shot learning has been applied to assembly tasks [33], while jigsaw puzzle techniques [36, 41] leverage shape completion strategies. Recent works [19, 23, 45, 57] utilize diffusion models to refine poses or point clouds for assembly. In contrast, our method introduces a two-step pairwise network for step-by-step assembly, tailored to pairwise object alignment.

2.3. 6D Pose Estimation for Robot Manipulation

6D pose estimation is crucial in robotics and computer vision for object interaction in unstructured environments [18, 50, 62]. Early handcrafted feature-based methods struggled in cluttered scenes [17, 18], while CNN-based approaches improved performance but lacked generalization [31, 62]. Domain randomization enhances robustness by varying synthetic datasets [49, 50]. In assembly tasks, 6D pose estimation aids manipulation planning with predefined objects [44, 51]. Like [16, 20–24, 43, 48, 66, 72, 73, 75, 76, 78], our

Dataset	#OC	#OS	Task Number	Pair	Task Hierarchy	Everyday Scenario	Symmetry	Assemble Type
PartNet [38]	24	26,671	-	No	No	No	No	Semantic
AutoMate [28]	2	92,529	1	No	No	No	No	Geometric
JoinABLe [58]	6	8,251	1	No	No	No	No	Semantic
NSM dataset[7]	11	1,246	1	Yes	No	No	No	Geometric
Breaking Bad[46]	-	10,474	1	No	No	No	No	Geometric
Factory[39]	8	60	8	Yes	No	No	No	Geometric
2BY2 Dataset (Ours)	**36**	**1034**	**18**	**Yes**	**Yes**	**Yes**	**Yes**	**Geometric and Semantic**

Table 1. **Dataset Comparison.** We compare *2BY2* dataset with exsiting datasets and benchmarks. **#OC** stands for the number of object categories. **#OS** stands for the number of object shapes. **Pair** denotes whether the dataset is pairwise. **Task Number** refers to the number of distinct assembly tasks, with the assembly of fractured pieces considered as a single task. **Task Hierarchy** stands for the different categories of task from coarse to fine, with ours shown in Section 3.1. **Everyday Scenario** means whether the assemble task has practical significance in real-world human applications. **Symmetry** denotes whether the dataset contains part symmetry annotation.

method leverages equivariant features for efficient 6D pose learning and improved generalization.

3. 2BY2 Dataset

3.1. 2BY2 Dataset Overview

We present the first large-scale 3D pairwise object assembly dataset for everyday scenarios, with detailed annotations for each object pair. The meshes in our dataset come from 3D Warehouse [1], SAPIEN PartNet-Mobility [61], Google SketchUp 3D Challenge [2], and Objaverse [10]. These meshes are manually paired, cleaned, annotated, and scaled uniformly. The 2BY2 dataset contains 517 unique pairs across three main tasks: *Lid Covering*, *Inserting*, and *High Precision Placing*, further subdivided into multiple subcategories, as shown in Table 2.

3.2. Data Annotation

To ensure high quality and reliability of our dataset, we conducted systematic cleaning and annotation of the collected meshes. First, we manually segment, integrate, and pair the meshes, classify them into *Object B* and *Object A*. *Object B* is the base or the receiving component, such as the nut, the vase, the postbox. *Object A*, is the fitting component, such as the bolt, the flower, the mail. This classification aligns with intuitive human assembly logic and supports our network's prediction strategy, such as positioning a nut before the bolt, as detailed in Sections 4 and 5.1. Automated scripts were used to uniformly scale meshes and align each pair to a canonical pose in world frame, defined as the object resting stably on the *XY* plane with its lowest point aligned to *Z=0*. For instance, bottles and vases are aligned as if placed on a table, and mailboxes on the ground.

During point cloud generation, we utilized blue noise sampling method [46] to extract point clouds uniformly from each mesh surface with dimension $(1024, 3)$. We also annotated each object category with its inherent symmetry properties, specifically considering rotational symmetry along the *Z*-axis, such as bottles, screws, and mirror symmetry along the *X*-axis, such as bread, letters.

3.3. Data Division and Task Diversity Analysis

Our dataset provides diverse task coverage across categories, with each further divided into specific sub-categories, see Table 2. Objects within each category vary in shape, size, and type. To enhance generalization, the testing set includes objects with unseen geometric shapes, as shown in Figure 3. We also compute Chamfer Distance on point clouds between training and testing sets to quantify geometry differences, as shown in Figure 2. This diversity ensures generalization ability and applicability in real world scenarios and supports robust 3D matching and assembly tasks.

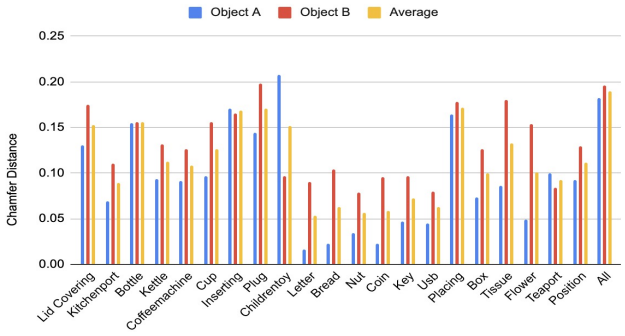


Figure 2. **Chamfer Distance Between Training and Testing Set.** We normalize point clouds and compute the Chamfer Distance. For each task we calculate the distance separately between point cloud of *Object A* and *Object B* in the training set and test set.

4. Problem Formulation

The task takes two point clouds as input, namely $\mathcal{P}_A$ and $\mathcal{P}_B$, each with dimension $(1024, 3)$. These point clouds are derived from objects $\mathcal{O}_A$ and $\mathcal{O}_B$ from predefined canonical pose, as detailed in 3.2, respectively, and is randomly augmented with SO(3) rotation and being translated to its centroid. The desired output would be two individual SE(3) pose two assemble $\mathcal{O}_A$ and $\mathcal{O}_B$ to the canonical pose.

Task	Lid Covering					Inserting								High Precision Placing				
	Kit	**Bot**	**Ket**	**Cof**	**Cup**	**Plu**	**Chi**	**Let**	**Bre**	**Nut**	**Coi**	**Key**	**Usb**	**Box**	**Tis**	**Flo**	**Tea**	**Pos**
Pair Num	24	86	28	26	16	14	19	32	24	20	21	20	20	25	20	60	42	21

Table 2. **2BY2 Dataset Statistics Overview.** The figure presents the number of object pairs across all task categories in the 2BY2 dataset, where each pair consists of two unique objects. The first row categorizes tasks into three major groups, while the second row provides a detailed breakdown of specific task categories. Specifically, **Kit** = Kitchenport, **Bot** = Bottle, **Ket** = Kettle, **Cof** = Coffee machine, **Cup** = Cup, **Plu** = Plug into socket, **Chi** = Children's toy, **Let** = Letter into mailbox, **Bre** = Bread into toaster, **Nut** = Bolt into nut, **Coi** = Coin into piggy bank, **Key** = Key into lock, **Usb** = USB cap, **Box** = Shoe boxing, **Tis** = Tissue placement on rack, **Flo** = Flower into vase, **Tea** = Teaware arrangement on tray, **Pos** = Positioning a cup on the coffee machine for coffee dispensing.

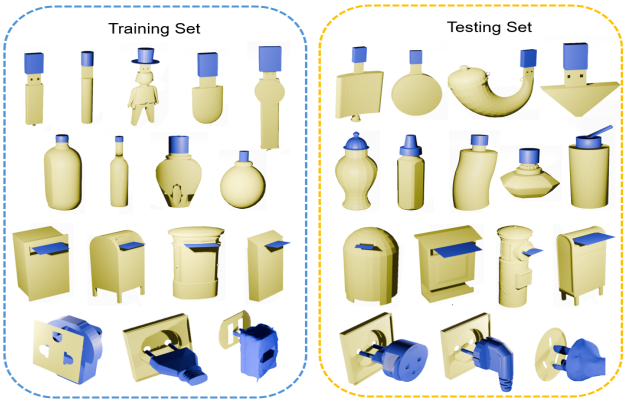


Figure 3. **Task Diversity Visualization.** The image shows selected objects from four different tasks: *USB*, *Bottle*, *Letter*, and *Plug in Socket*. On the left are the objects selected on training set, and on the right is the testing set. As seen in the legend, object geometry varies in both the training and testing set, with the testing set containing novel shapes not seen in the training set.

5. Method

5.1. Two-step Pairwise Network Architecture

To effectively learn pairwise object assembly, we propose a two-step pairwise network architecture with two branches: Branch B ($\mathcal{B}_\mathcal{B}$) and Branch A ($\mathcal{B}_\mathcal{A}$), as shown in Figure 4. Branch B predicts the pose of $\mathcal{P}_\mathcal{B}$, which is the socket, using a two-scale Vector Neuron DGCNN encoder [11] to extract SE(3) equivariant features, denoted as $\mathcal{E}_\mathcal{B}$, followed by MLP-based pose prediction heads for translation and rotation. The transformed $\mathcal{P}_\mathcal{B}$ and inserter object $\mathcal{P}_\mathcal{A}$, which is the plug, are then passed to Branch A, which extracts SE(3) equivariant features, denoted as $\mathcal{E}_\mathcal{A}$ and SO(3) invariant features ($\mathcal{I}_\mathcal{B}$). The features are fused through element-wise multiplication, allowing $\mathcal{B}_\mathcal{A}$ to predict the pose of $\mathcal{P}_\mathcal{A}$ using information from both objects. This architecture ensures geometric alignment and matching by leveraging shared feature representations while reducing feature interference.

Our two-step pairwise network is inspired by the human approach to pairwise assembly tasks. For example, when arranging a vase with flowers, one intuitively first positions the vase correctly before placing the flowers inside. Similarly, inserting an envelope into a mailbox requires identifying the mailbox slot's pose first. By mimicking this sequential strategy, our model simulates human decision-making process, enabling more efficient and accurate assembly tasks.

5.2. Two-scale SE(3) Equivariant and SO(3) Invariant Feature Extraction

We employ a two-scale SE(3) Vector Neuron DGCNN, an enhanced variant of the original Vector Neuron DGCNN [11], as our encoder to extract SE(3) equivariant and SO(3) invariant features. This architecture leverages equivariance to improve sample efficiency of the model, while incorporating a two-scale information fusion mechanism to capture geometric features at two different scales.

SE(3) Equivariance and SO(3) Invariance. SE(3) equivariance combines SO(3) rotation and T(3) translation equivariance: rotation equivariance ensures that a network's output rotates with the input, while translation equivariance shifts the output accordingly. SO(3) invariance means the network's output remains unchanged under any 3D rotation. By leveraging SE(3) equivariance, the model benefits from improved sample efficiency and generalization. This is particularly advantageous for assembly tasks, where objects may appear in arbitrary poses.

Vector Neuron DGCNN. The Vector Neuron Network [11] extends traditional neurons from scalars to 3D vectors, designs vector-based convolutional layers and nonlinear functions like pooling and ReLU to support SO(3) equivariant and SO(3) invariant feature extraction. VNN operates in vector space, captures richer geometric relationships and ensures more robust feature representations for downstream tasks.

Two-scale Vector Neuron DGCNN. We propose a two-scale Vector Neuron DGCNN for extracting SE(3) equivariant and SO(3) invariant features $\mathcal{E}_\mathcal{B}$, $\mathcal{I}_\mathcal{B}$, and $\mathcal{E}_\mathcal{A}$. As shown in Figure 4, the encoder comprises two branches with different K values, each consisting of multiple Vector Neuron convolutional layers followed by pooling. The extracted features from both branches are concatenated and further pro-

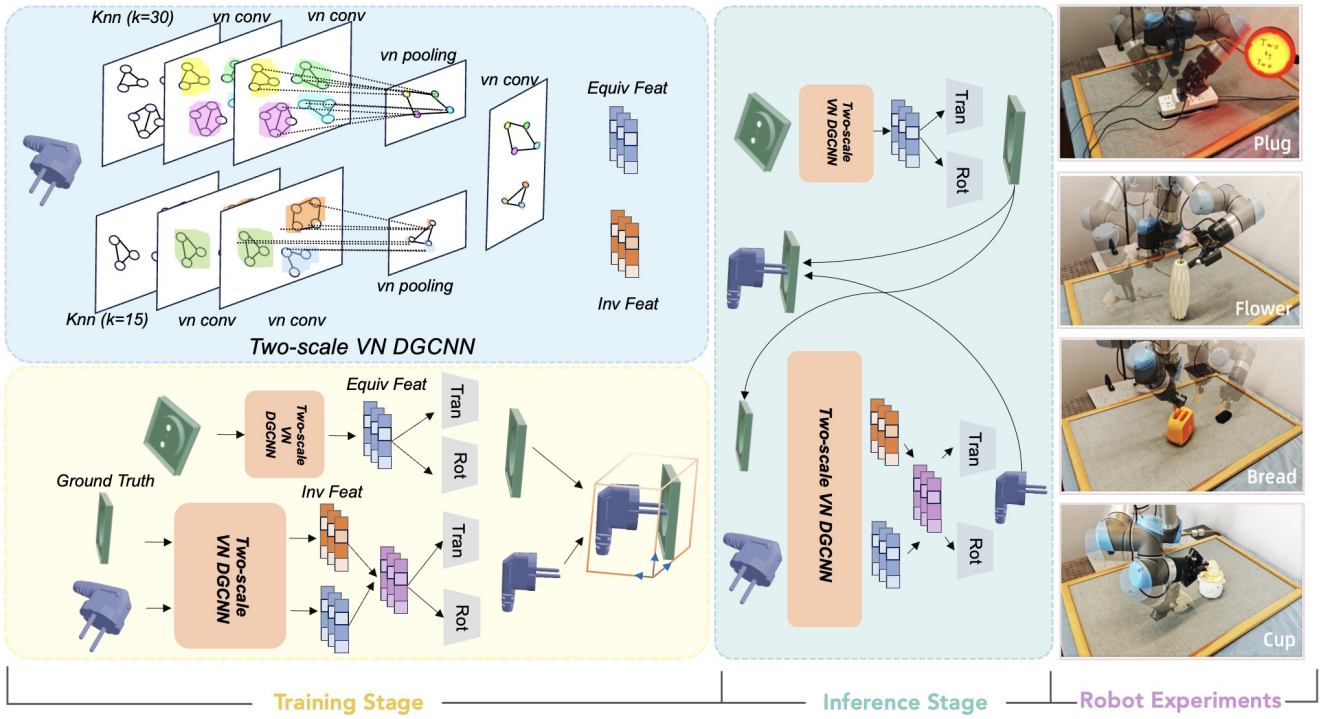


Figure 4. **Our Two-Step Pairwise Network.** We utilize two-scale VN DGCNN as our encoder to extract equivariant and invariant feature. We first predict the canonical pose of $\mathcal{O}_\mathcal{B}$ and then predict the pose of $\mathcal{O}_\mathcal{A}$ according to it.

cessed through an additional Vector Neuron convolutional layer. Point clouds $\mathcal{P}_\mathcal{B}$ and $\mathcal{P}_\mathcal{A}$ are independently processed, forming graphs that propagate through both branches.

The SO(3) rotation equivariance of our encoder is ensured by the inherent equivariant properties of the Vector Neuron layers. To achieve T(3) translation equivariance, with an input point cloud $P = (p_1, p_2, ..., p_n), p_i \in R^3$, we compute its centroid $x = (\Sigma_{i=1}^{n} p_i)/n$, and get the input point cloud as $P' = P - x$. In this way, our prediction is T(3) translation equivariant, i.e., f is our encoder and $\mathcal{P}$ is the original point cloud.

$$f(\mathcal{P} + \mathcal{T}) = f(\mathcal{P}) + \mathcal{T}, \quad T \in R^3 \tag{1}$$

Our two-scale VN DGCNN employs dual K-nearest neighbor (KNN) values to extract features across two distinct scales, enhancing its ability to capture both local and global information. This pyramid structure enables the network to simultaneously grasp overall object shapes and fine-grained details, improving feature extraction.

5.3. Cross Object Fusion Module

We utilize point-wise multiplication, shown in Figure 4, as our cross object fusion module designed in B_A. We fuse the feature of $\mathcal{P}_\mathcal{B}$ and $\mathcal{P}_\mathcal{A}$ by multiplying $\mathcal{I}_\mathcal{B}$ and $\mathcal{E}_\mathcal{A}$, so that each point in $\mathcal{P}_A$ will have the geometry feature of both $\mathcal{P}_A$ and $\mathcal{P}_B$. This approach integrates the geometric feature of $\mathcal{P}_B$ in each point while preserving the rotation equivariance of $\mathcal{P}_A$, i.e., f is an equivariant neural network, R is random rotation matrix,

$$f(R \cdot (\mathcal{I}_\mathcal{B} * \mathcal{E}_\mathcal{A})) = R \cdot f(\mathcal{I}_\mathcal{B} * \mathcal{E}_\mathcal{A}), \quad R \in R^{3\times3} \tag{2}$$

5.4. Pose Prediction

At both branches, we utilizes two seperate MLPs as our pose prediction head, to separately predict the translation $T \in R^3$ and rotation $R \in R^{3\times3}$. Compared to predicting translation and rotation within a single prediction head, this approach helps mitigate the issue of differing convergence speeds between the two components.

5.5. Training and Evaluation Strategy

We adopt a separate training and evaluation strategy for our network. To minimize the impact of pose prediction errors of $\mathcal{P}_\mathcal{B}$ on $\mathcal{P}_\mathcal{A}$, we train $\mathcal{B}_\mathcal{A}$ and $\mathcal{B}_\mathcal{B}$ independently. Specifically, for $\mathcal{B}_\mathcal{A}$, during training, we utilize $\mathcal{P}_\mathcal{B}$ under canonical pose, which is our ground truth point cloud of $\mathcal{P}_\mathcal{B}$, to train our model. During testing, we first predict the pose of $\mathcal{P}_\mathcal{B}$, then use the transformed $\mathcal{P}_\mathcal{B}$, along with the initial $\mathcal{P}_\mathcal{A}$ to predict A's pose, as shown in Figure 4. This phased, two-step training and evaluation strategy reduces errors caused by joint training of object poses, ensuring more accurate predictions.

5.6. Loss Function

To train our network to robustly predict poses, we use the following equation as our loss function:

$$\mathcal{L} = \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}} \tag{3}$$

Specifically, for predicted pose translation $T_{\text{pred}} \in R^3$, rotation $\mathcal{R}_{\text{pred}} \in R^{3\times3}$ and ground truth pose translation $T_{\text{gt}} \in R^3$ and rotation $\mathcal{R}_{\text{gt}} \in R^{3\times3}$, we use $\mathcal{L}_1$ loss to compute our $\mathcal{L}_{\text{trans}}$:

$$\mathcal{L}_{\text{trans}} = \mathcal{L}_1(T_{\text{pred}}, T_{\text{gt}}) \tag{4}$$

As for the rotation, we utilize Geodesic Distance, which measures the shortest path between two rotations on the rotation manifold. It offers a smooth and bounded angular error, ensuring stable gradients and accurately achieving precise rotation alignment.

$$\mathcal{L}_{\text{rot}} = \arccos\left(\frac{\text{tr}(\mathcal{R}_{\text{gt}}\mathcal{R}_{\text{pred}}^T) - 1}{2}\right) \tag{5}$$

6. Experiments

In this section, we present a comprehensive evaluation and analysis of our two-step pairwise network architecture by addressing the following key questions:

1. How does our network perform on 2BY2 tasks compared to existing baseline approaches, including matching-based, graph-network-based, and diffusion-based assembly methods?

2. How well does our network generalize across multiple tasks within the 2BY2 dataset? Can our network effectively handle a diverse set of tasks simultaneously?

3. Can our network generalize to real-world robot tasks?

6.1. 2BY2 Dataset Main Experiment

6.1.1 Experiment Set Up

Tasks. We divide the 18 assembly tasks in the 2BY2 dataset into training and testing sets individually and compared the performance of our method with various baseline approaches. To further evaluate its cross-task generalization ability, we conducted additional experiments on tasks such as *Lid Covering*, *Insertion*, and *High Precision Placement*, as well as *All* task, which requires the method to handle all tasks in the entire dataset. See Table 2 for task details.

Evaluation metrics. Following metrics from datasets like Breaking Bad [46] and Neural Shape Mating [7], we use Root Mean Squared Error (RMSE) to evaluate both rotation and translation of the predicted SE(3) pose. Specifically, rotations are represented using Euler angles with symmetry considerations, see Section 3.2 for symmetry details.

Training parameters. We set batch size to be 4, and the initial learning rate of Adam Optimizer [12] to be 1e-4. We train models for 1000 epochs for them to fully converge.

6.1.2 Baselines

We compare our method with SE-3 assembly [60], Puzzlefusion++ [57], Jigsaw [36] and Neural Shape Mating [7].

- **SE-3 Assembly [60]** proposes a network architecture to leverage SE(3) equivariance for representations considering multi-part correlations, and predict the pose of each part jointly.
- **Puzzlefusion++ [57]** proposes an auto-agglomerative 3D fracture assembly framework. It iteratively aligns and merges fragments using a diffusion model for 6-DoF alignment and a transformer model for verification.
- **Jigsaw [36]** leverages hierarchical features of global and local geometry to match and align the fracture surfaces, and recovers the global pose of each piece to restore the underlying object.
- **Neural Shape Mating [7]** utilizes PointNet for feature encoding and a transformer for feature fusion to learn the correlations between assembly parts, enabling joint prediction of their poses.

6.1.3 2BY2 Benchmark Results and Analysis

Table 3 presents the quantitative performance of our method compared to all baselines. The results show that our approach outperforms the baselines across 18 fine-grained assembly tasks, with an average improvement of *0.046* in translation RMSE and *8.97* in rotation RMSE.

Additionally, we evaluate our method on three cross-category tasks defined in Section 3.1, namely *Lid Covering*, *Inserting* and *High Precision Placing*, and achieve the state-of-the-art performance. Moreover, in the most comprehensive *All* task, we outperform the baseline by *0.123* in translation and *10.90* in rotation, demonstrating strong generalization across tasks and object shapes. In the meantime, baseline comparisons confirm the rigor and challenge of our tasks. Results on challenging tasks like *Plug* and *Key* highlight our framework's effectiveness in complex scenarios.

We analyze that the superior performance of our designed network is due to the approach of separately predicting the poses of the two objects in a step-by-step manner. This prevents the pose errors from interfering with each other, which often occurs in other baselines when predicting both poses simultaneously. Additionally, the design of our encoder makes our network more sensitive to subtle changes in rotation and translation, resulting better performance.

6.2. Real-World Robot Experiment

Real-world robot experiment setup. As shown in Figure 5, we conduct our real-world robot experiments using a UR5 robotic arm, equipped with a Robotiq 2F-85 Gripper. We select four tasks, *Cup*, *Flower*, *Bread* and *Plug*, demonstrating that our model exhibits strong generalization ability on unseen real-world objects.

Task	Jigsaw [36]		Puzzlefusion++ [57]		NSM [7]		SE(3)-Assembly [60]		Ours	
	RMSE(T)	RMSE(R)	RMSE(T)	RMSE(R)	RMSE(T)	RMSE(R)	RMSE(T)	RMSE(R)	RMSE(T)↓	RMSE(R)↓
Lid Covering	0.398	33.33	0.408	37.74	0.184	33.45	0.125	21.37	**0.090**	**16.12**
Kitchenport	0.477	45.80	0.423	47.23	0.237	57.47	0.093	17.29	**0.068**	**16.60**
Bottle	0.411	34.71	0.385	35.23	0.227	78.58	0.147	36.92	**0.076**	**27.70**
Kettle	0.335	43.71	0.372	38.38	0.215	61.10	0.133	13.15	**0.111**	**11.56**
Coffeemachine	0.527	32.67	0.437	34.64	0.253	50.66	0.142	24.43	**0.076**	**22.83**
Cup	0.408	33.55	0.439	33.58	0.260	67.35	0.160	46.46	**0.122**	**23.18**
Inserting	0.364	53.58	0.327	57.83	0.275	69.93	0.199	46.3	**0.142**	**38.03**
Plug	0.372	56.89	0.348	48.89	0.303	52.26	0.176	18.58	**0.094**	**9.74**
Childrentoy	0.268	59.88	0.245	63.21	0.271	93.77	0.302	80.52	**0.242**	**57.81**
Letter	0.409	67.99	0.357	72.08	0.317	76.48	0.121	39.24	**0.094**	**33.74**
Bread	0.220	57.84	0.201	60.92	0.171	65.50	0.111	51.13	**0.090**	**36.40**
Nut	0.476	40.08	0.323	47.29	0.271	55.32	0.102	46.68	**0.051**	**35.60**
Coin	0.406	39.62	0.348	51.40	0.289	62.58	0.111	28.69	**0.107**	**22.88**
Key	0.384	42.85	0.348	50.38	0.290	63.60	0.087	17.28	**0.045**	**16.32**
Usb	0.463	67.41	0.342	58.23	0.252	69.90	0.215	32.28	**0.128**	**28.98**
Precision Placing	0.375	73.94	0.287	67.81	0.211	85.02	0.134	57.86	**0.115**	**44.84**
Box	0.137	33.72	0.134	40.47	0.130	72.47	0.071	25.08	**0.066**	**21.53**
Tissue	0.292	82.39	0.265	85.18	0.175	79.78	0.183	73.02	**0.115**	**64.37**
Flower	0.328	64.39	0.283	59.04	0.246	87.18	0.213	64.89	**0.125**	**42.23**
Teaport	0.302	68.33	0.324	61.01	0.288	56.32	0.085	40.59	**0.050**	**26.11**
Position	0.423	58.07	0.389	57.55	0.257	70.11	0.166	28.17	**0.141**	**24.46**
ALL	0.360	53.34	0.342	58.23	0.284	70.30	0.233	52.34	**0.110**	**41.44**

Table 3. **Quantitative Evaluation on *2BY2* for Pairwise Object Assembly.** Our method outperforms the baseline across all 18 fine-grained assembly tasks, as well as demonstrating significant improvement on three cross-category assembly tasks. It achieves a lower task average with a reduction of *0.046* in translation RMSE and *8.97* in rotation RMSE.

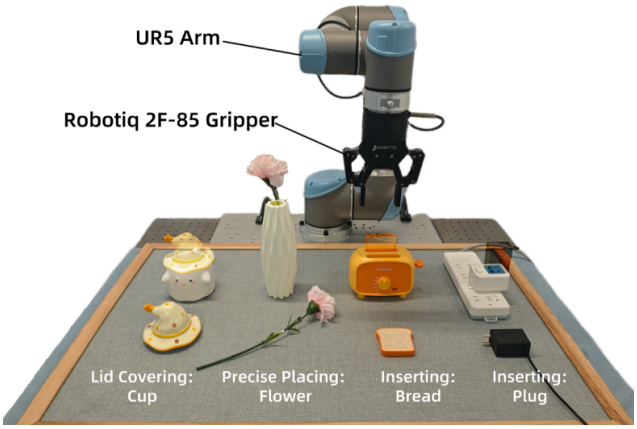


Figure 5. **Real Robot Setup.** We conduct real-world robot experiments on *Cup*, *Flower*, *Bread* and *Plug* tasks.

We place objects in the scene with random initial poses and scan them to obtain their point clouds. Using pre-trained models on selected data of *2BY2* dataset, we predict the pose of each object. A manually designed grasping pose is then applied to pick up each object, and based on the predicted poses, the robotic arm plans a trajectory to complete the assembly. We use SE(3) assembly [60] as the baseline and test our approach on 10 different initial poses. As shown in Table 4, our method significantly outperforms the baseline.

Task	Cup	Flower	Bread	Plug	Overall
SE(3) [60]	2/10	4/10	1/10	2/10	22.5%
Ours	8/10	10/10	6/10	7/10	77.5%

Table 4. **Real-World Robot Experiment Success Rate Results.**

7. Ablation Study

In this section, we conduct comprehensive experiments to demonstrate the rationality of our network design and the effectiveness of each module.

Encoder. To validate the effectiveness of two-scale Vector Neuron(VN) DGCNN, we compare it with other encoders: VN DGCNN [11], DGCNN [55], and PointNet [42].

Two-step network design. We compare our method with an end-to-end approach, which jointly predicts the pose of $\mathcal{P}_\mathcal{A}$ and $\mathcal{P}_\mathcal{B}$. Specifically, we utilize $\mathcal{B}_\mathcal{A}$ and the input point cloud of $\mathcal{P}_\mathcal{A}$ and $\mathcal{P}_\mathcal{B}$ to get the 6D pose of P_A, while using the same encoder to extract $\mathcal{E}_\mathcal{B}$ and pass it through the same pose prediction head to predict the pose of P_B.

As shown in in Table 5, we show the results of ablation studies on *Lid covering*, *Inserting* and *Precision Placing*, which are more comprehensive and require cross-task generalization abilities. The performance declines in both translation and rotation when we removing our two-scale VN DGCNN encoder and change it to Vector Neuron DGCNN [11], DGCNN [55], Pointnet [42]. It demonstrates

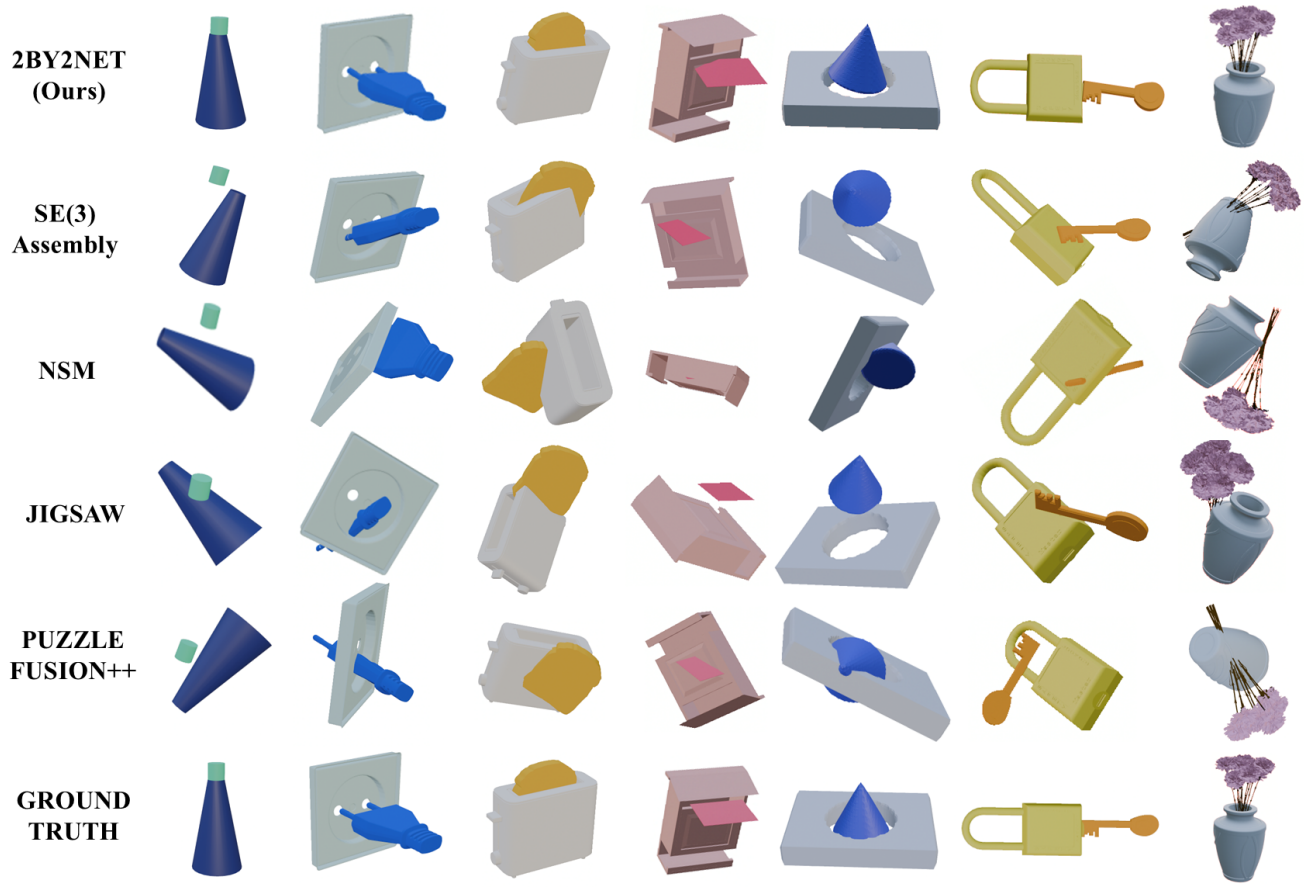


Figure 6. **Qualitative Results Comparison.** We highlight *Bottle*, *Plug*, *Bread*, *Letter*, *Childrentoy*, *Key*, and *Flower* tasks to demonstrate our improved translation and rotation predictions compared to baseline methods.

Task	Vector Neuron DGCNN [11]		DGCNN [55]		PointNet [42]		w/o Two-step		Ours	
	RMSE(T)	RMSE(R)	RMSE(T)	RMSE(R)	RMSE(T)	RMSE(R)	RMSE(T)	RMSE(R)	RMSE(T) ↓	RMSE(R)↓
Lid Covering	0.098	18.23	0.245	70.06	0.234	65.32	0.117	18.74	**0.090**	**16.12**
Inserting	0.157	41.22	0.234	62.25	0.234	66.80	0.164	41.11	**0.142**	**38.03**
Precision Placing	0.121	48.01	0.245	65.19	0.211	72.47	0.137	46.38	**0.115**	**44.84**
ALL	0.123	44.67	0.277	72.46	0.264	75.38	0.139	45.20	**0.110**	**41.44**

Table 5. **Ablation Study Results.** We compare various encoders including Vector Neuron DGCNN [11], DGCNN [55], PointNet [42], and our proposed two-scale Vector Neuron DGCNN. We also compare end-to-end networks with two-step networks to demonstrate the effectiveness of each component in our network design.

that our encoder effectively exploits the advantage of SE(3) equivariance, enabling greater sample efficiency and more robust generalization abilities. Compared with version in Figure 4, the experiment performance declines when we change our two-step network in a joint-learning manner, proving that our two-step network design can reduce error caused by jointly predictions and thereby is more effective.

8. Conclusion

2BY2 is a significant step in bridging the gap between geometry-based assembly tasks and everyday object assemblies. With pose and symmetry annotations for 517 object pairs across 18 fine-grained tasks, 2BY2 sets a new benchmark for 3D assembly challenges. Our two-step pairwise SE(3) pose estimation framework, which leverages equivariant features, demonstrates superior performance over existing approaches in reducing both translation and rotation errors. Robot experiments further validate the method's generalizability in practical 3D assembly scenarios. In conclusion, 2BY2 provides both a comprehensive benchmark and an effective framework, with the aim of inspiring and supporting more generalizable solution in robot manipulation.

References

[1] 3dwarehouse. https://3dwarehouse.sketchup.com/by/su3dchallenge, 2014. 3

[2] google-sketchup-3d-challenge. https://3dwarehouse.sketchup.com/by/su3dchallenge, 2014. 3

[3] Chamfer distance pytorch. https://github.com/ThibaultGROUEIX/ChamferDistancePytorch/tree/master, 2020. 13

[4] Blender 4.3. https://www.blender.org/, 2024. 12

[5] Benedict J Brown, Corey Toler-Franklin, Diego Nehab, Michael Burns, David Dobkin, Andreas Vlachopoulos, Christos Doumas, Szymon Rusinkiewicz, and Tim Weyrich. A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings. *ACM transactions on graphics (TOG)*, 27(3):1–9, 2008. 2

[6] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011. 2

[7] Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12724–12733, 2022. 2, 3, 6, 7, 12, 13, 15

[8] Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-pa: Score-based 3d part assembly. In *British Machine Vision Conference*, 2023. 2

[9] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 2

[10] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[11] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021. 2, 4, 7, 8, 14, 15

[12] P Kingma Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014. 6

[13] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39 (5):3929–3945, 2023. 14

[14] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM transactions on graphics (TOG)*, 23(3):652–663, 2004. 2

[15] Thomas Funkhouser, Hijung Shin, Corey Toler-Franklin, Antonio García Castañeda, Benedict Brown, David Dobkin, Szymon Rusinkiewicz, and Tim Weyrich. Learning how to match fresco fragments. *Journal on Computing and Cultural Heritage (JOCCH)*, 4(2):1–13, 2011. 2

[16] Chongkai Gao, Zhengrong Xue, Shuying Deng, Tianhai Liang, Siqi Yang, Lin Shao, and Huazhe Xu. Riemann: Near real-time se (3)-equivariant robot manipulation without point cloud segmentation. *arXiv preprint arXiv:2403.19460*, 2024. 2, 13

[17] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 International Conference on Computer Vision*, pages 858–865, 2011. 2

[18] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision – ACCV 2012*, pages 548–562, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 2

[19] Sepidehsadat Sepid Hossieni, Mohammad Amin Shabani, Saghar Irandoust, and Yasutaka Furukawa. Puzzlefusion: unleashing the power of diffusion models for spatial puzzle solving. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[20] Boce Hu, Xupeng Zhu, Dian Wang, Zihao Dong, Haojie Huang, Chenghao Wang, Robin Walters, and Robert Platt. Orbitgrasp: Se (3)-equivariant grasp learning. In *8th Annual Conference on Robot Learning*, 2024. 2

[21] Haojie Huang, Dian Wang, Xupeng Zhu, Robin Walters, and Robert Platt. Edge grasp network: A graph-based se (3)-invariant approach to grasp detection. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3882–3888. IEEE, 2023. 14

[22] Haojie Huang, Haotian Liu, Dian Wang, Robin Walters, and Robert Platt. Match policy: A simple pipeline from point cloud registration to manipulation policies. *arXiv preprint arXiv:2409.15517*, 2024.

[23] Haojie Huang, Karl Schmeckpeper, Dian Wang, Ondrej Biza, Yaoyao Qian, Haotian Liu, Mingxi Jia, Robert Platt, and Robin Walters. IMAGINATION POLICY: Using generative point cloud models for learning manipulation policies. In *8th Annual Conference on Robot Learning*, 2024. 2, 13

[24] Haojie Huang, Linfeng Zhao, Haotian Liu, Zhang Ye, Si-Yuan Huang, Mingxi Jia, Boce Hu, Fangzhou Lin, Yu Qi, Dian Wang, et al. Pix2act: Image-space manipulation policies with equivariant augmentation. *arXiv preprint arXiv:2607.11167*, 2026. 2

[25] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. In *ACM Siggraph 2006 papers*, pages 569–578. 2006. 2

[26] Prakhar Jaiswal, Jinmiao Huang, and Rahul Rai. Assembly-based conceptual 3d modeling with unlabeled components using probabilistic factor graph. *Computer-Aided Design*, 74: 45–54, 2016. 2

[27] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2

[28] Benjamin Jones, Dalton Hildreth, Duowen Chen, Ilya Baran, Vladimir G Kim, and Adriana Schulz. Automate: A dataset

and learning approach for automatic mating of cad assemblies. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2, 3

[29] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239. Springer, 2025. 2

[30] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *Acm Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 2

[31] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. *CoRR*, abs/1711.10006, 2017. 2

[32] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3d part assembly from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 664–682. Springer, 2020. 2

[33] Yulong Li, Andy Zeng, and Shuran Song. Rearrangement planning for general part assembly. In *7th Annual Conference on Robot Learning*, 2023. 2

[34] Yuval Litvak, Armin Biess, and Aharon Bar-Hillel. Learning pose estimation for high-precision robotic assembly using simulated depth images. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3521–3527. IEEE, 2019. 2

[35] Hao-Yu Liu, Jian-Wei Guo, Hai-Yong Jiang, Yan-Chao Liu, Xiao-Peng Zhang, and Dong-Ming Yan. Puzzlenet: boundary-aware feature matching for non-overlapping 3d point clouds assembly. *Journal of Computer Science and Technology*, 38 (3):492–509, 2023. 2

[36] Jiaxin Lu, Yifan Sun, and Qixing Huang. Jigsaw: Learning to assemble multiple fractured objects. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6, 7, 13, 15

[37] Jianlan Luo, Eugen Solowjow, Chengtao Wen, Juan Aparicio Ojea, Alice M Agogino, Aviv Tamar, and Pieter Abbeel. Reinforcement learning on variable impedance controller for high-precision robotic assembly. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3080–3087. IEEE, 2019. 2

[38] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 3

[39] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, et al. Factory: Fast contact for robotic assembly. *arXiv preprint arXiv:2205.03532*, 2022. 2, 3

[40] Abhinav Narayan, Rajendra Nagar, and Shanmuganathan Raman. Rgl-net: A recurrent graph learning framework for progressive part assembly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 78–87, 2022. 2

[41] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[42] 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*, pages 652–660, 2017. 7, 8, 14, 15

[43] Yu Qi, Haibo Zhao, Ziyu Guo, Siyuan Ma, Ziyan Chen, Yaokun Han, Renrui Zhang, Zitiantao Lin, Shiji Xin, Yijian Huang, et al. Bear: Benchmarking and enhancing multimodal language models for atomic embodied capabilities. *arXiv e-prints*, pages arXiv–2510, 2025. 2

[44] Ismael Rodrıguez, Korbinian Nottensteiner, Daniel Leidner, Michael Kaßecker, Freek Stulp, and Alin Albu-Schäffer. Iteratively refined feasibility checks in robotic assembly sequence planning. *IEEE Robotics and Automation Letters*, 4(2):1416–1423, 2019. 2

[45] Gianluca Scarpellini, Stefano Fiorini, Francesco Giuliari, Pietro Moreiro, and Alessio Del Bue. Diffassemble: A unified graph-diffusion model for 2d and 3d reassembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28098–28108, 2024. 2

[46] Silvia Sellán, Yun-Chun Chen, Ziyi Wu, Animesh Garg, and Alec Jacobson. Breaking bad: A dataset for geometric fracture and reassembly. *Advances in Neural Information Processing Systems*, 35:38885–38898, 2022. 2, 3, 6, 12, 13

[47] Hijung Shin, Christos Doumas, Thomas Funkhouser, Szymon Rusinkiewicz, Kenneth Steiglitz, Andreas Vlachopoulos, and Tim Weyrich. Analyzing and simulating fracture patterns of theran wall paintings. *Journal on Computing and Cultural Heritage (JOCCH)*, 5(3):1–14, 2012. 2

[48] Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. ET-SEED: EFFICIENT TRAJECTORY-LEVEL SE(3) EQUIVARIANT DIFFUSION POLICY. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[49] Joshua Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *CoRR*, abs/1703.06907, 2017. 2

[50] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *ArXiv*, abs/1809.10790, 2018. 2

[51] Weiwei Wan, Kensuke Harada, and Kazuyuki Nagata. Assembly sequence planning for motion planning. *CoRR*, abs/1609.03108, 2016. 2

[52] 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*, 2022. 13

[53] Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang,

Robin Walters, and Robert Platt. Equivariant diffusion policy. In *8th Annual Conference on Robot Learning*, 2024. 13

[54] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Ran Zhang, Chin-Yi Cheng, and Jiajun Wu. Ikea-manual: Seeing shape assembly step by step. *Advances in Neural Information Processing Systems*, 35:28428–28440, 2022. 2

[55] 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 Transactions on Graphics (tog)*, 38(5):1–12, 2019. 7, 8, 14, 15

[56] Ziming Wang and Rebecka Jörnsten. Se (3)-bi-equivariant transformers for point cloud assembly. *arXiv preprint arXiv:2407.09167*, 2024. 2

[57] Zhengqing Wang, Jiacheng Chen, and Yasutaka Furukawa. Puzzlefusion++: Auto-agglomerative 3d fracture assembly by denoise and verify. *arXiv preprint arXiv:2406.00259*, 2024. 2, 6, 7, 13, 15

[58] Karl DD Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G Lambourne, Armando Solar-Lezama, et al. Joinable: Learning bottom-up assembly of parametric cad joints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15849–15860, 2022. 2, 3

[59] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838, 2020. 2

[60] Ruihai Wu, Chenrui Tie, Yushi Du, Yan Zhao, and Hao Dong. Leveraging se (3) equivariance for learning 3d geometric shape assembly. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14311–14320, 2023. 6, 7, 13, 15

[61] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[62] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 2

[63] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning compact representations for lidar completion and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2023. 2

[64] Boshen Xu, Sipeng Zheng, and Qin Jin. Spaformer: Sequential 3d part assembly with transformers. *arXiv preprint arXiv:2403.05874*, 2024. 2

[65] Xianghao Xu, Paul Guerrero, Matthew Fisher, Siddhartha Chaudhuri, and Daniel Ritchie. Unsupervised 3d shape reconstruction by part retrieval and assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8559–8567, 2023. 2, 13

[66] Zhengrong Xue, Zhecheng Yuan, Jiashun Wang, Xueqian Wang, Yang Gao, and Huazhe Xu. Useek: Unsupervised se (3)-equivariant 3d keypoints for generalizable manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1715–1722. IEEE, 2023. 2, 13

[67] Kangxue Yin, Zhiqin Chen, Siddhartha Chaudhuri, Matthew Fisher, Vladimir G Kim, and Hao Zhang. Coalesce: Component assembly by learning to synthesize connections. In *2020 International Conference on 3D Vision (3DV)*, pages 61–70. IEEE, 2020. 2

[68] Kevin Zakka, Andy Zeng, Johnny Lee, and Shuran Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9404–9410. IEEE, 2020. 2

[69] Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 33:6315–6326, 2020. 2

[70] Rufeng Zhang, Tao Kong, Weihao Wang, Xuan Han, and Mingyu You. 3d part assembly generation with instance encoded transformer. *IEEE Robotics and Automation Letters*, 7(4):9051–9058, 2022. 2

[71] Ruiyuan Zhang, Jiaxiang Liu, Zexi Li, Hao Dong, Jie Fu, and Chao Wu. Scalable geometric fracture assembly via co-creation space among assemblers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7269–7277, 2024. 2

[72] Haibo Zhao, Dian Wang, Yizhe Zhu, Xupeng Zhu, Owen Howell, Linfeng Zhao, Yaoyao Qian, Robin Walters, and Robert Platt. Hierarchical equivariant policy via frame transf. *arXiv preprint arXiv:2502.05728*, 2025. 2

[73] Haibo Zhao, Yu Qi, Boce Hu, Yizhe Zhu, Ziyan Chen, Heng Tian, Xupeng Zhu, Owen Howell, Haojie Huang, Robin Walters, et al. Generalizable hierarchical skill learning via object-centric representation. *IEEE Robotics and Automation Letters*, 11(6):6536–6543, 2026. 2

[74] Junzhe Zhu, Yuanchen Ju, Junyi Zhang, Muhan Wang, Zhecheng Yuan, Kaizhe Hu, and Huazhe Xu. Densematcher: Learning 3d semantic correspondence for category-level manipulation from a single demo. *International Conference on Learning Representations (ICLR) 2025*, 2024. 2

[75] Xupeng Zhu, Dian Wang, Guanang Su, Ondrej Biza, Robin Walters, and Robert Platt. On robot grasp learning using equivariant models. *Autonomous Robots*, 47(8):1175–1193, 2023. 2, 13

[76] Xupeng Zhu, Yu Qi, Yizhe Zhu, Robin Walters, and Robert Platt. Equact: An se (3)-equivariant multi-task transformer for open-loop robotic manipulation. *arXiv preprint arXiv:2505.21351*, 2025. 2

[77] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020. 2

[78] Yizhe Zhu, Zhang Ye, Boce Hu, Haibo Zhao, Yu Qi, Dian Wang, and Robert Platt. Residual rotation correction using tactile equivariance. *IEEE Robotics and Automation Letters*, 2026. 2

A. Appendix Section

A.1. 2BY2 Dataset

Unlike previous datasets like Breaking Bad and Neural Shape Mating [7, 46] which focus on assembly of object fragments, our ***2BY2*** dataset focuses on pairwise assembly of daily objects with geometry and task variety, includes tasks that can be quite challenging for robot manipulation. For example *Plug*, *Bread*, *flower* are very challenging in real world becuase they require precise pose alignment to achieve assembly success.

In previous datasets such as Breaking Bad, the pose of each fragment depends on all the other fragments. However, in daily pairwise assembly task, the pose of the *Object B*, such as bottle and toaster, is not affected by *Object A*, such as cap and bread, and is only determined by the canonical space. In contrast, the pose of *Object A* is influenced by the geometry and pose of *Object B*. For instance, the pose of a cap is determined by the rim of the cup, while the pose of a piece of bread is dictated by the slot of the toaster. Consequently, previous methods that jointly predict the poses of two objects are not well-suited for daily pairwise assembly tasks. To address this, we propose a two-step paired network architecture that sequentially predicts the pose of each object, effectively mitigating pose errors introduced by joint pose prediction in prior approaches.

A.1.1 Dataset Collection

We segment, integrate, and pair meshes obtained online, scaling them to a global scale of *3.0*. Each mesh pair is categorized into *Object B* and *Object A*, where *Object B* serves as the receiving component, and *Object A* functions as the fitting component. Similar to Breaking Bad [46], we triangulate each mesh using blender [4] and use blue noise sampling method to extract the point cloud from the surface of each mesh, and use padding to make sure each dimension aligns with $(1024, 3)$.

A.1.2 Symmetry Annotation

Each object is associated with a JSON file specifying its symmetry type. In this work, we account for two types of symmetry: axis symmetry along the *x, y, z* axes, and rotational symmetry around the *x, y, z* axes.

A.1.3 Task Definition

In the *Lid Covering* category, *Object A* refers to the lid, and *Object B* refers to the corresponding body, including *Kitchen*, *Bottle*, *Kettle*, *Coffeemachine*, and *Cup*.

In the *Inserting* category:

- In *Plug*, *Object A* is the plug, and *Object B* is the socket.

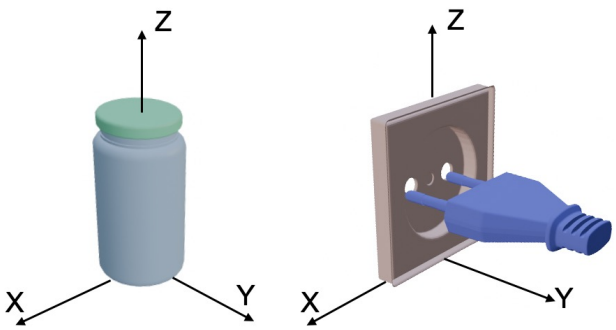


Figure 7. **The Definition of Canonical Pose.** The left image illustrates the canonical pose of the task *bottle*, while the right image represents the canonical pose of *plug*.

- In *Children's Toy*, *Object A* is the block, such as cylinder and cone, and *Object B* is the board with slots.
- In *Letter*, *Object A* is the mail, and *Object B* is the postbox.
- In *Bread*, *Object A* is the bread, and *Object B* is the toaster.
- In *Nut*, *Object A* is the bolt, and *Object B* is the nut.
- In *Coin*, *Object A* is the coin, and *Object B* is the piggy bank.
- In *Key*, *Object A* is the key, and *Object B* is the lock.
- In *USB*, *Object A* is the cap, and *Object B* is the USB body.

In the *High Precision Placing* category:

- In the *Box* task, *Object A* refers to the shoes, and *Object B* refers to the box. The goal is to neatly place the shoes in the shoebox.
- In the *Tissue* task, *Object A* refers to the tissue, and *Object B* refers to the tissue rack. The goal is to place the tissue on the rack.
- In the *Flower* task, *Object A* refers to the flower, and *Object B* refers to the vase.
- In the *Teapot* task, *Object A* refers to the teapot, and *Object B* refers to the tea tray. The goal is to neatly place the teapot on the tray.
- In the *Position* task, *Object A* refers to the cup, and *Object B* refers to the coffee machine. The goal is to place the cup underneath the spout of the coffee machine.

A.1.4 Definition of Canonical Pose in Different Tasks

In all tasks except for *Plug*, the canonical pose refers to the assembled state where the two objects are placed on the *XY* plane under the influence of gravity, ensuring stable contact with the plane. Additionally, the positive Z-axis passes through the geometric center of the object's base, ensuring proper central and vertical alignment, as shown in Figure 9.

In the *Plug* task, the canonical pose is defined as the state where the socket is placed on the *XZ* plane, representing the wall, as shown in Figure 9.

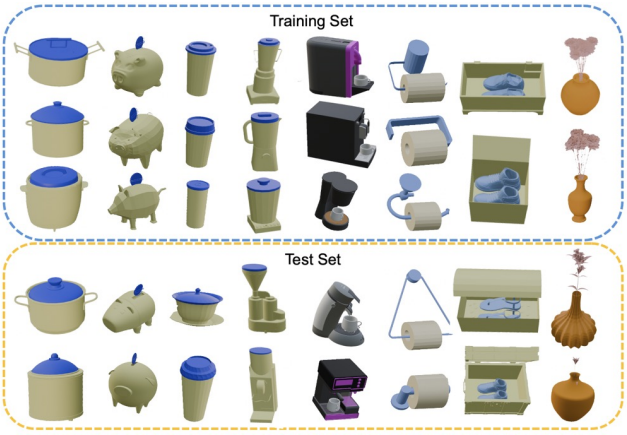


Figure 8. **Task Diversity Visualization.** From left to right, each column shows selected meshes from training set and test set of *Kitchenport*, *Coin*, *Cup*, *Coffeemachine*, *Position*, *Toilet*, *Shoes*, *Flower*.

Notably, in tasks where only a single relative pose is required—such as plugging into a socket which is fixed on the wall—the plug's pose can be determined through coordinate transformation, as illustrated in Section A.3.3.

A.1.5 Data Splition

As described in the main paper, our ***2BY2*** dataset includes *18* fine-grained tasks, such as *Bottle* and *Children's Toy*, and *4* tasks which require cross-category generalization ability, which is *Lid Covering*, *Inserting*, *High Precision Placing* and *All*. We ensure geometric diversity when assigning each object exclusively to either the training or test set, as shown in Figure 8.

For cross-category tasks like *Lid Covering*, the training and test sets both include objects from its own categories, such as *Kitchen*, *Bottle*, *Kettle*, *Coffeemachine*, and *Cup*. Similar applies to the *Inserting* and *High Precision Placing* tasks. For the *All* task, both the training and test sets include all *18* fine-grained tasks.

For each of the *18* fine-grained task, we maintain a training-to-test set ratio of approximately *3:2*. For *Lid Covering*, *Inserting*, *High Precision Placing* and *All*, the ratio is controlled at roughly *5:2*.

A.2. Methodology

A.2.1 SE(3) Equivariant and SO(3) Invariant Feature

Robots operate within a three-dimensional Euclidean space, where manipulation tasks inherently encompass geometric symmetries such as rotations. Recent works [16, 23, 53, 65, 66, 75] leverage symmetry to enable robust learning and generalization. As illustrated in the main paper, SE(3) equivariant feature, which is extracted by our designed encoder, leverage symmetry to improve sample efficiency. In both branch, SE(3) equivariant features of $\mathcal{O}_B$ and $\mathcal{O}_A$ are used for object pose estimation.

SO(3) invariant features encode geometric shape information in the latent space, independent of the input point cloud's orientation. In $\mathcal{B}_A$, the SO(3) invariant feature of $\mathcal{P}_B$ is extracted to facilitate the pose estimation of $\mathcal{P}_A$. Intuitively, the predicted pose of the bread is determined by the geometry of the toaster slot.

A.3. Experiment

A.3.1 Data Augmentation

During training, we apply SO(3) data augmentation to all methods, including both our approach and the baselines, which provides sufficient data for network convergence and ensures fair comparison. Notably, as pointed out by [52], although our network exhibits SE(3) equivariance, SO(3) data augmentation still benefits the learning process.

A.3.2 2BY2 Dataset Experiment

Similar to Breaking Bad [46], we also use Chamfer Distance (CD) as our additional evaluation metric to validate the effectiveness our multi-step pairwise network.

Evaluation Metric. Chamfer Distance (CD) [3] is a common metric used to measure the similarity between two point clouds or sets. It is widely applied in computer vision, 3D shape matching, point cloud alignment. More specifically, given two point clouds $P = \{p_1, p_2, \ldots, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$, Chamfer Distance between P and Q is defined as:

$$CD(P,Q) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|_2^2 + \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|_2^2 \tag{6}$$

More specifically, we use the average Chamfer Distance between the predicted P'_B and ground truth P_B, and the predicted P'_A and ground truth P_A:

$$CD = \frac{1}{2}(CD(P'_B, P_B) + CD(P'_A, P_A)) \tag{7}$$

Results and Analysis. As detailed in the main paper, we compare our multi-step pairwise network with SE-3 assembly [60], Puzzlefusion++ [57], Jigsaw [36] and Neural Shape Mating [7]. As shown in Table 6 and Figure , our method consistently outperforms all baselines across 18 fine-grained tasks, demonstrating significantly improved alignment and geometric matching accuracy. This highlights the superior precision and effectiveness of our multi-step pairwise network. Moreover, in tasks such as *Lid Covering*, *Inserting*, *Precision Placing*, and the overall *All* category, our method achieves a substantial margin of improvement over the baselines, further indicating its robust generalization ability.

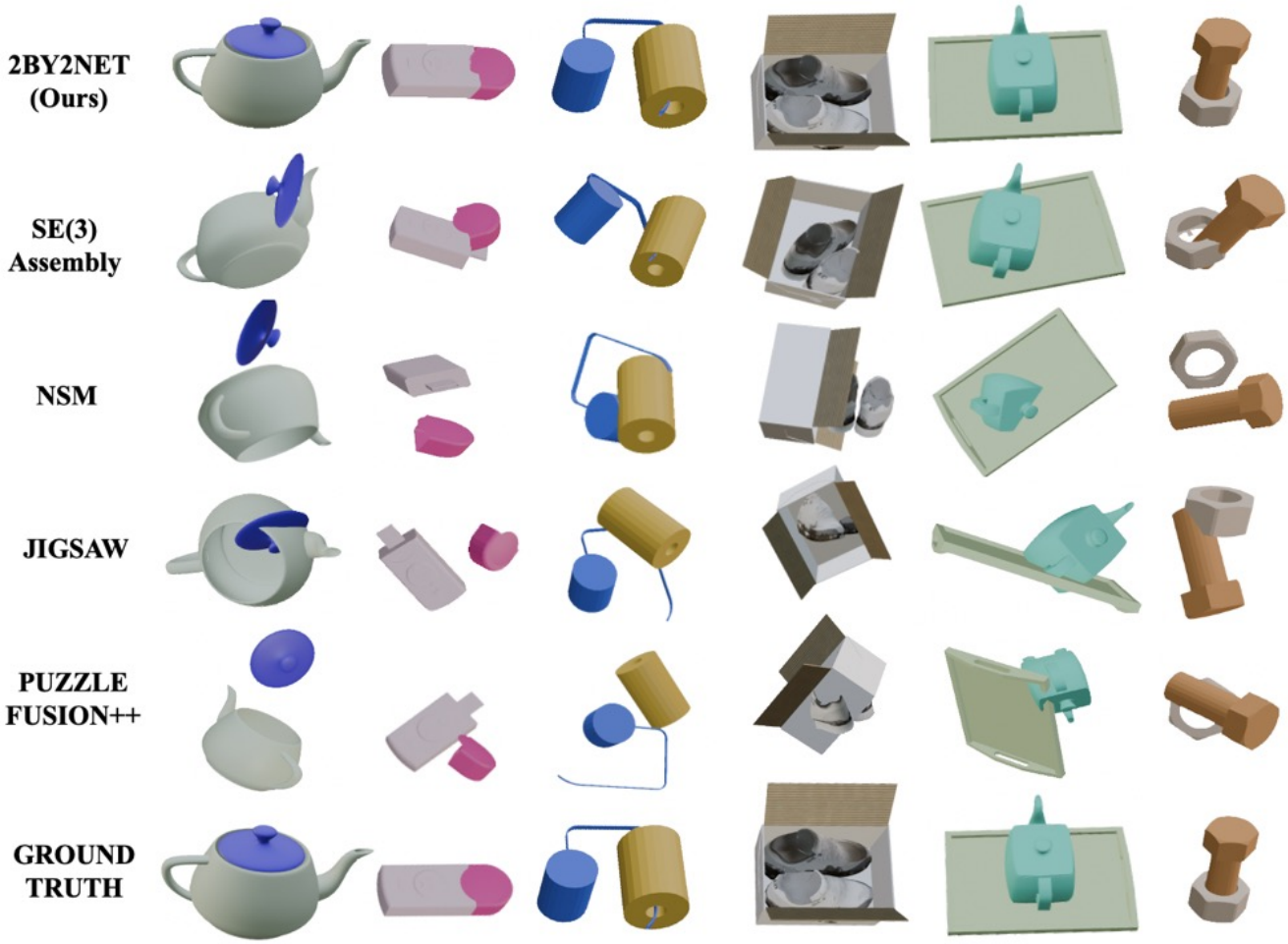


Figure 9. **Qualitative Results Comparison.** We highlight *Kettle*, *USB*, *Toilet*, *Shoes*, *Teapot*, *Nut* tasks to demonstrate our improved translation and rotation predictions compared to baseline methods.

A.3.3 Real-robot Experiment

In some tasks in the real world, instead of two poses, only one relative pose is needed to solve the pairwise assembly task. For example, when plugging into the socket that is fixed to the wall, only the pose of the plug is needed. To resolve tasks like these, we first infer the socket's pose in our defined world frame. In this step, we are not rotating socket arbitrarily. Then estimate the plug's target pose in defined world frame. The plug's target pose in the real world can be calculated using a coordinate transformation.

Moreover, rather than relying on pre-defined grasping poses, numerous existing grasping methods, such as [13, 21], can generate adaptable grasps efficiently. The motion trajectory can then be computed using motion planning library.

A.4. Ablation Study

As detailed in the main paper, we compare our method on *Lid covering*, *Inserting*, and *High precision placing* and *All* task in ***2BY2*** dataset with other encoders: Vector Neuron DGCNN [11], DGCNN [55], PointNet [42] and an end-to-end approach which jointly predicts the pose of P_A and P_B.

Evaluation Metric. Similar to Section A.3.2, We choose Chamfer Distance (CD) as our additional evalution metric.

Results and Analysis. As shown in Table 7, replacing our multi-scale VN DGCNN encoder with Vector Neuron DGCNN [11], DGCNN [55], or PointNet [42] results in a performance drop, highlighting that our encoder better captures geometric features and exhibits greater sensitivity to pose transformations. Additionally, substituting our multi-step network with a joint-learning approach leads to an increase in Chamfer Distance, underscoring the effectiveness of our multi-step network design.

A.5. Limitations and Future Works

The current design of our network is primarily constrained by the scope of the ***2BY2*** dataset, which could be further ex-

Task	Jigsaw [36] CD	Puzzlefusion++ [57] CD	NSM [7] CD	SE(3)-Assembly [60] CD	Ours CD ↓
Lid Covering	1.665	1.809	1.082	0.453	**0.362**
Kitchenport	1.100	1.169	0.772	0.323	**0.230**
Bottle	1.640	1.738	1.194	0.601	**0.321**
Kettle	1.277	1.425	0.903	0.428	**0.163**
Coffeemachine	1.290	1.394	1.178	0.394	**0.189**
Cup	1.336	1.260	1.093	0.493	**0.268**
Inserting	0.712	0.842	0.860	0.431	**0.278**
Plug	0.752	0.746	0.411	0.194	**0.085**
Childrentoy	1.037	0.917	0.874	0.814	**0.791**
Letter	1.296	0.862	0.341	0.191	**0.140**
Bread	0.406	0.301	0.139	0.144	**0.105**
Nut	0.131	0.665	0.946	0.368	**0.059**
Coin	0.946	0.921	0.756	0.146	**0.134**
Key	0.603	0.829	0.441	0.149	**0.032**
Usb	0.541	0.656	0.508	0.327	**0.266**
Precision Placing	0.888	0.472	0.366	0.306	**0.255**
Box	0.263	0.234	0.205	0.102	**0.093**
Tissue	0.462	0.644	0.335	0.349	**0.232**
Flower	0.463	0.361	0.371	0.376	**0.295**
Teaport	0.577	0.475	0.345	0.157	**0.069**
Position	0.759	0.735	0.585	0.548	**0.302**
ALL	1.223	1.469	1.100	0.679	**0.268**

Table 6. **Quantitative Evaluation on *2BY2* for Pairwise Object Assembly.** Our method outperforms the baseline across all 18 fine-grained assembly tasks, as well as demonstrating significant improvement on *4* cross-category assembly tasks, including *Lid covering*, *Inserting*, *Precision Placing* and *All*. It achieves an average reduction of *0.138* in Chamfer Distance.

Task	Vector Neuron DGCNN [11] Chamfer Distance	DGCNN [55] Chamfer Distance	PointNet [42] Chamfer Distance	w/o Multi-step Chamfer Distance	Ours Chamfer Distance ↓
Lid Covering	0.387	0.873	0.875	0.439	**0.362**
Inserting	0.297	0.483	0.489	0.290	**0.278**
Precision Placing	0.274	0.864	0.729	0.283	**0.255**
ALL	0.294	0.806	0.816	0.307	**0.268**

Table 7. **Ablation Study Results.** We compare various encoders including Vector Neuron DGCNN [11], DGCNN [55], PointNet [42], and our proposed multi-scale Vector Neuron DGCNN. We also compare end-to-end networks with multi-step networks to demonstrate the effectiveness of each component in our network design.

panded to include a wider range of tasks and more complex everyday scenarios. Additionally, rather than hardcoding the grasping pose, a policy network for robotic manipulation could be trained using the ***2BY2*** dataset. Furthermore, the network architecture can be optimized to reduce computational overhead, improving its suitability for real-time robotic operations.