A. Appendix Section

A.1. 2BY2 Dataset

Unlike previous datasets like Breaking Bad and Neural Shape Mating [3, 11] 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 [11], we triangulate each mesh using blender [2] 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.

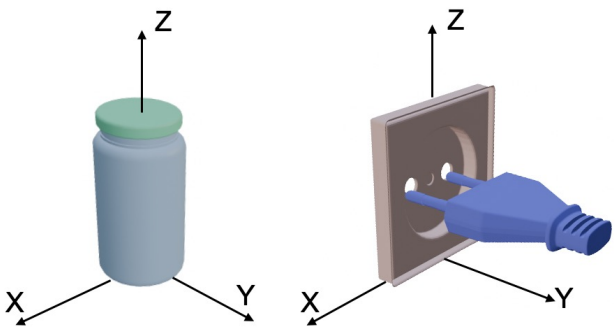

Figure 1. **The Definition of Canonical Pose.** The left image illustrates the canonical pose of the task *bottle*, while the right image represents the canonical pose of *plug*.

- 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 3.

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 3.

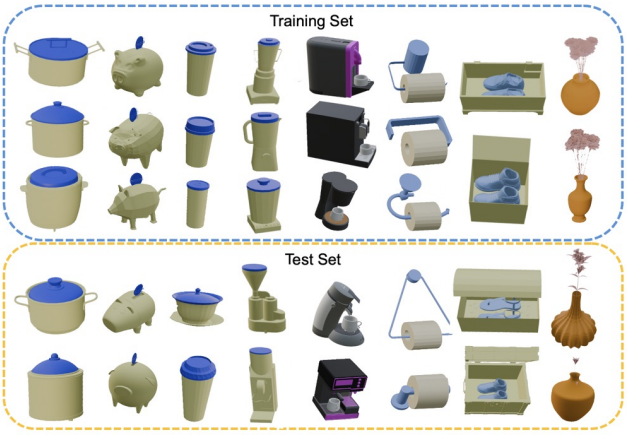

Figure 2. **Task Diversity Visualization.** From left to right, each column shows selected meshes from training set and test set of *Kitchenport*, *Coin*, *Cup*, *Coffeemachine*, *Position*, *Toilet*, *Shoes*, *Flower*.

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 2.

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 [6, 8, 13, 17–19] 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 [12], 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 [11], 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) [1] 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:

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

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:

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

Results and Analysis. As detailed in the main paper, we compare our multi-step pairwise network with SE-3 assembly [16], Puzzlefusion++ [15], Jigsaw [9] and Neural Shape Mating [3]. As shown in Table 1 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.

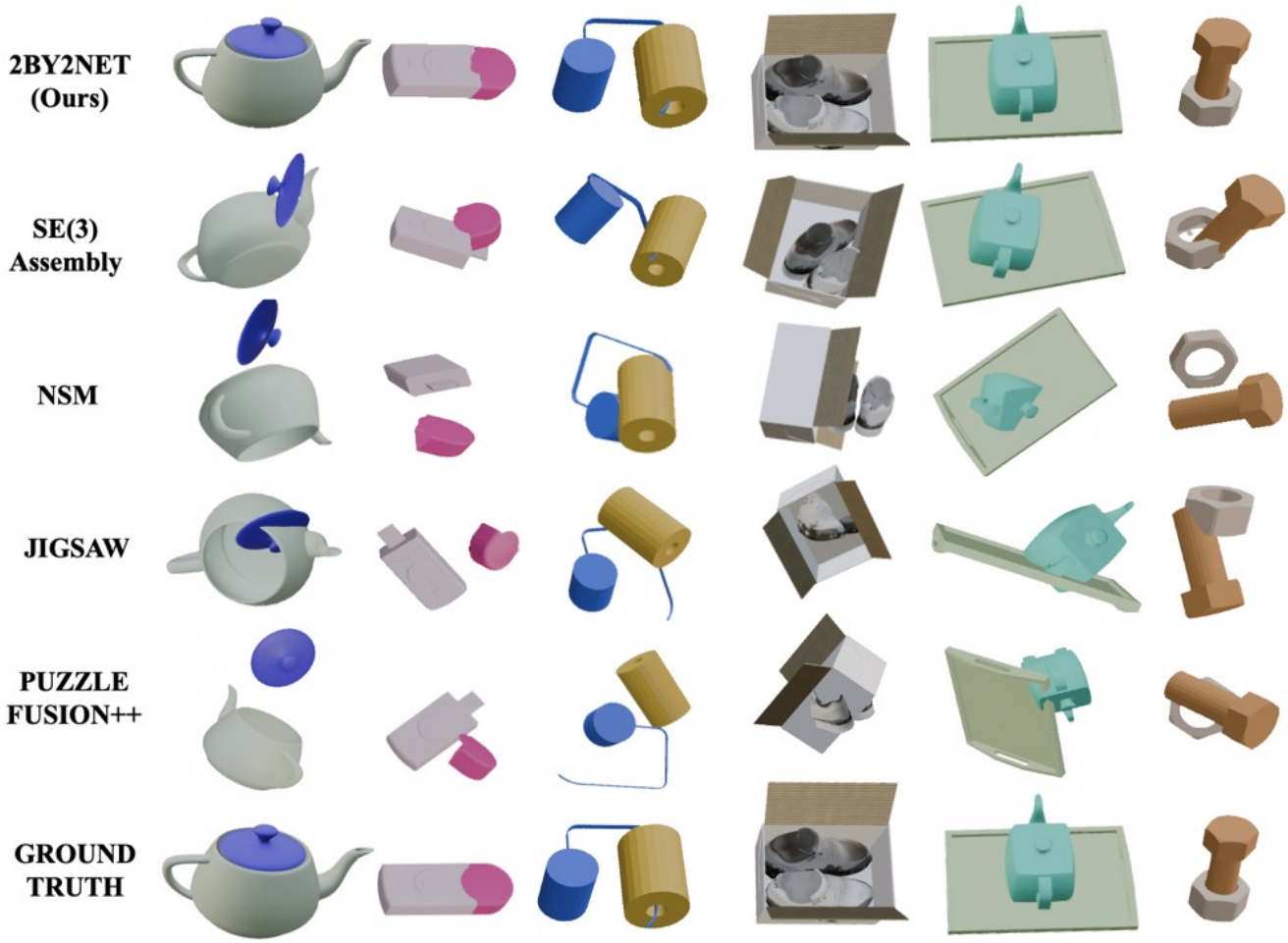

Figure 3. **Qualitative Results Comparison.** We highlight *Kettle*, *USB*, *Toilet*, *Shoes*, *Teapot*, *Nut* tasks to demonstrate our improved translation and rotation predictions compared to baseline methods.

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 [5, 7], 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 [4], DGCNN [14], PointNet [10] 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 2, replacing our multi-scale VN DGCNN encoder with Vector Neuron DGCNN [4], DGCNN [14], or PointNet [10] 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 expanded to include a wider range of tasks and more complex

Task	Jigsaw [9] CD	Puzzlefusion++ [15] CD	NSM [3] CD	SE(3)-Assembly [16] 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 1. **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 [4] Chamfer Distance	DGCNN [14] Chamfer Distance	PointNet [10] 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 2. **Ablation Study Results.** We compare various encoders including Vector Neuron DGCNN [4], DGCNN [14], PointNet [10], 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.

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.

References

[1] Chamfer distance pytorch. https://github.com/ThibaultGROUEIX/ChamferDistancePytorch/tree/master, 2020. 2

[2] Blender 4.3. https://www.blender.org/, 2024. 1

[3] 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. 1, 2, 4

[4] 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. 3, 4

[5] 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. 3

[6] 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

[7] 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. 3

[8] 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

[9] Jiaxin Lu, Yifan Sun, and Qixing Huang. Jigsaw: Learning to assemble multiple fractured objects. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 4

[10] 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. 3, 4

[11] 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. 1, 2

[12] 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. 2

[13] 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. 2

[14] 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. 3, 4

[15] Zhengqing Wang, Jiacheng Chen, and Yasutaka Furukawa. Puzzlefusion++: Auto-agglomerative 3d fracture assembly by denoise and verify. *arXiv preprint arXiv:2406.00259*, 2024. 2, 4

[16] 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. 2, 4

[17] 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

[18] 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.

[19] 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