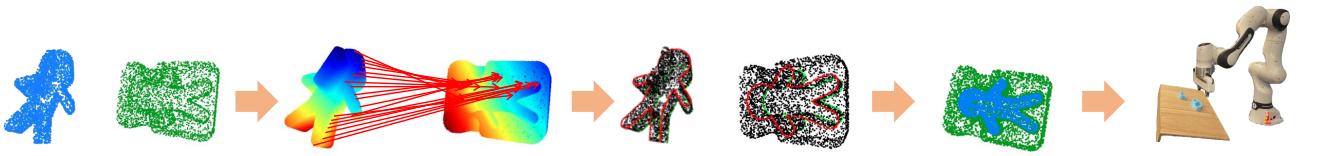
2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Learning Interaction Constraints for Robot Manipulation via Set Correspondences

Junyu Nan[1], Jessica Hodgins[2], Brian Okorn[2]



(a) Input Point Clouds (b) Cross-object Attention (c) Set Correspondences (d) Cross-Pose (e) Robot Manipulation

Fig. 1: We propose a new cross-pose estimation method that predicts set-level correspondences to avoid ambiguous point matches due to symmetries. Given (a) input point clouds, (b) cross-object attention is used to predict (c) set correspondences, which are aligned to obtain (d) cross-pose. The estimated cross-pose can then be used for (e) robot manipulation tasks.

***Abstract*— Cross-pose estimation between rigid objects is a fundamental building block for robotic applications. In this paper, we propose a new cross-pose estimation method that predicts correspondences on a set level as opposed to a point level. This contrasts methods that predict cross-pose from per-point correspondences, which can encounter optimization problems for objects with symmetries, since each point may have multiple valid correspondences. Our method, SCAlign, consists of a Set Correspondence Network (SCN) which predicts these sets and their correspondences, and an alignment module to compute their relative cross-pose. Taking point clouds of two objects as input, SCN predicts a set label for each point such that such that points that share a set label form a cross object correspondence. The alignment module then computes the cross-pose as the $SE(3)$ transformation that aligns these set correspondences. We compare SCAlign against other cross-pose estimation baselines on a synthetically generated dataset, SynWidth, which contains randomly generated width-mate objects with symmetric or near-symmetric intercepts. SCAlign significantly outperforms the baselines on this challenging dataset. Additionally, we show that set correspondences can be leveraged to distinguish positive and negative matches between pegs and holes. Robot experiments further validate the practical application of this approach.**

I. INTRODUCTION

Moving an object to a desired relative position with respect to another object is an important primitive in many robotics applications, especially in manufacturing and assembly. Effective cross-pose estimation, i.e., the estimation of the relative position between objects [1], ensures that robotic systems can accurately interact with a wide array of objects not seen at training time. However, many daily and industrial objects contain symmetries, leading to multiple, equivalent solutions for their desired relative poses. While point-correspondence based methods of predicting cross-pose have been shown to generalize over different object instances [1], [2], [3], they face challenges in predicting a single final pose due to these symmetries.

To address the challenge of object symmetries, we reformulate the problem of predicting pose from point-to-point correspondences to use set-to-set correspondences. We propose SCAlign, described in Figure 1, which consists of: 1) a **S**et **C**orrespondence **N**etwork (SCN) that takes point cloud data from two objects as input, and predicts set-level correspondences between both objects; 2) a set alignment module which computes a rigid transformation that aligns all set correspondences. SCN learns to identify each object's contact points, classify them into sets, and predict correspondences between these sets. The cross-pose is then determined by solving for the rigid transformation that best aligns the corresponding sets of points. We focus on predicting cross-pose of width-mate objects, as they are common parts in manufacturing and assembly scenarios.

To evaluate our method against other cross-pose estimation baselines, we generate a synthetic dataset, SynWidth, containing width-mate objects with symmetric or near-symmetric intercepts. SCAlign significantly reduces the prediction error as compared to other cross-pose estimation baselines [1], [2]. We show that the set correspondences can be used to improve the accuracy in identifying positive and negative width-mate pairs and conduct experiments on Franka arm in both insertion and matching scenarios.

The primary contributions of this paper are as follows:

1) A new method, SCAlign, for cross-pose estimation that handles object symmetries by predicting set correspondences.
2) A synthetic dataset, SynWidth, for cross-pose evaluation on objects with symmetries or near-symmetries, where SCAlign significantly outperforms other cross-pose estimation baselines.
3) A new method, SCMatcher, for relational inference of width-mate compatibility across multiple objects utilizing set correspondences.
4) Real robot experiments showcase our method's applicability in real-world scenarios.

[1]Authors are with the Robotics Institute, Carnegie Mellon University. This work was done during an internship with the Boston Dynamics AI Institute. jnan1@andrew.cmu.edu

[2]Authors are with the Boston Dynamics AI Institute. {jkh, bokorn}@theaiinstitute.com

II. RELATED WORKS

Point Cloud Registration: Traditional point cloud registration relied on optimizing the distance between two point clouds of the same object or scene. Iterative Closest Point (ICP) [4] iteratively approximates point correspondences using the closest points across point clouds and optimize the transform that reduces this distance. Deep Closest Point (DCP) [5] improves the correspondence estimation by utilizing a transformer architecture to learn correspondences in an end-to-end fashion. These methods, however, are designed to estimate the transform between multiple scans of the same object, whereas we focus on the the cross-pose estimation problem, which estimates the relative transform between different objects to achieve a desired configuration.

Cross-pose Estimation: Cross-pose, or cross object relative pose estimation, reframes the registration problem to better facilitate learning object interactions. TAX-Pose [1] used soft-correspondences between different objects to estimate this cross-pose from a limited number of examples. While this method predicts cross object correspondences, it relies on symmetry breaking tricks to handle ambiguities associated with symmetric objects which are infeasible for nearly symmetric objects. Neural Descriptor Fields (NDFs) [3], [6], [7] solve the cross-pose problem by learning a dense embedding fields to predict relative configurations of objects through gradient descent. Local Neural Descriptor Fields (LNDFs) [2] extends this method to allow both objects in the scene to be freely moved. In addition to NDF, relative configuration of inference objects to canonical object models can also be inferred from other representations, such as Coherent Point Drift (CPD) [8] and NUNOCS [9]. While these methods generalize well over object instances, they all attempt to learn point-to-point correspondences and do not produce the accuracy required for assembly tasks when the objects contain symmetries or near symmetries.

Part Assembly: Learning to complete assembly tasks has long been a goal for robotics. Deep learning based methods have been used to predict the relative position of a set of parts given a class [10], [11] or target image [12]. While this work is able to generate a variety of plausible, physically-valid assemblies, they are more concerned with the rough location of parts and not the exact pose required to fit them together under specific joining geometries. We see this type of work as complimentary to our own. The rough part location could be used to reduce the pose search space and define a task specific part-to-part correspondence, while our work would produce the exact mating pose required to join the parts. Many robotic assembly applications have simplified this task by focusing on relatively simple text-described insertion tasks [13], [14], learning only 2D insertion poses [15], [16], requiring annotated final configurations [17], [18], or by not generalizing to novel objects [19], [20], [21], [22], [23]. Our method generalizes to a wide variety of insertion shapes, requires no annotation of the final configuration, and predicts insertion poses from arbitrary 6D initial poses.

III. PROBLEM STATEMENT

Cross-pose Estimation. We follow the definition of cross-pose estimation problem in [1]. The goal is to identify the "goal pose" of object A relative to the pose of object B, given the point clouds $\mathbf{P}_A$ and $\mathbf{P}_B$, respectively. Specifically, we want to learn a function, $f(\cdot, \cdot)$, that outputs an SE(3) rigid transformation $\mathbf{T}_{AB} = f(\mathbf{P}_A, \mathbf{P}_B)$, such that if object A is transformed by $\mathcal{T}_{AB}$, then object A is in the goal configuration relative to object B.

However, object symmetries can make the prediction of this transform ambiguous, and the optimal $\mathbf{T}_{AB}$ might not be unique. In order to generalize this relationship to objects with symmetries, we denote the set of valid solutions as $\mathcal{T}_{AB} = \{\mathbf{T} \in SE(3) \mid \mathbf{T}\mathbf{P}_A \equiv \mathbf{T}_{AB}\mathbf{P}_A\}$, where $\equiv$ denotes equivalence between the infinitely dense sampling of the point clouds, agnostic of order. Phrased simply, two point clouds are equivalent if they could represent the same underlying geometry. For example, point clouds rotated about an axis of symmetry would all be considered equivalent. Consequently, the goal of f is to output any solution in the valid solution set $\mathcal{T}_{AB}$:

$$f(\mathbf{P}_A, \mathbf{P}_B) = \mathbf{T} \text{ such that } \mathbf{T} \in \mathcal{T}_{AB}. \tag{1}$$

Set Correspondences. We start by defining point correspondence. Point $\mathbf{p}_A \in \mathbf{P}_A$ corresponds to point $\mathbf{p}_B \in \mathbf{P}_B$ under cross-pose $\mathbf{T}_{AB}$ if $\mathbf{T}_{AB}\mathbf{p}_A = \mathbf{p}_B$. Note that the such correspondences only exist for points on the contact surfaces of two objects aligned by ground truth cross-pose and only if the sampling patterns on these surfaces match.

While differences in sampling patterns can be solved using distance thresholds, symmetries pose a more challenging problem. Since $\mathbf{T}_{AB}$ may not be the unique, there might exist multiple correspondences for $\mathbf{p}_A$, denoted as

$$\mathcal{M}(\mathbf{p}_A, \mathcal{T}_{AB}) = \{\mathbf{T}\mathbf{p}_A \mid \mathbf{T} \in \mathcal{T}_{AB}\}, \tag{2}$$

and vice versa for $\mathbf{p}_B$, where

$$\mathcal{M}(\mathbf{p}_B, \mathcal{T}_{BA}) = \{\mathbf{T}^{-1}\mathbf{p}_B \mid \mathbf{T} \in \mathcal{T}_{AB}\}. \tag{3}$$

If an object has symmetries, then the point-to-point correspondences will be many-to-many. However, if we view these correspondences from a set or object-region perspective as opposed to a point-to-point perspective, we can resolve this ambiguity. We say that a set of points $\mathcal{S}_A \subseteq \mathbf{P}_A$ correspond to set $\mathcal{S}_B \subseteq \mathbf{P}_B$ under the cross-poses $\mathcal{T}_{AB}$ if

$$\bigcup_{\mathbf{p} \in \mathcal{S}_A} \mathcal{M}(\mathbf{p}, \mathcal{T}_{AB}) \subseteq \mathcal{S}_B \wedge \bigcup_{\mathbf{p} \in \mathcal{S}_B} \mathcal{M}(\mathbf{p}, \mathcal{T}_{BA}) \subseteq \mathcal{S}_A, \tag{4}$$

where the containment metric $\mathbf{P} \subseteq \mathbf{P}'$ is defined up to point sampling, *e.g.,* that points $\mathbf{P}$ could be reasonably sampled from a portion of the surface represented by points $\mathbf{P}'$. For example, given points on two identical circles, it possible to define the correspondences as sets containing the contour of concentric circles with varying radius, or even a single set containing all points from each circle. The necessary property of these sets is that they are closed under all valid cross-poses, $\mathcal{T}_{AB}$.

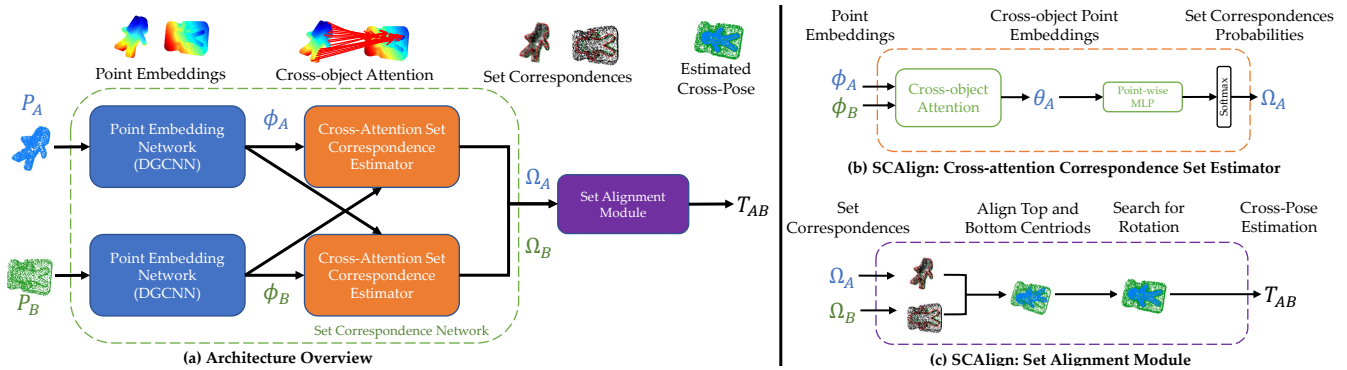


Fig. 2: Overview of our method. **(a)** Architecture overview of SCAlign, which takes as input point clouds $\mathbf{P}_A, \mathbf{P}_B$ of two objects, and outputs cross-pose $\mathbf{T}_{AB}$. SCAlign consists of a Set Correspondence Network (SCN), which extracts features from each point cloud and then estimate set correspondences via a cross-attention based set correspondence estimator, detailed in **(b)**. Our Set Alignment Module **(c)** computes the cross-pose $\mathbf{T}_{AB}$ from set correspondences.

IV. METHOD

A. Set Correspondence Network (SCN)

To estimate the cross-object set correspondences, we take inspirations from DCN [5] and TAX-Pose [1] for the design of our Set Correspondence Network (SCN). The SCN module, shown in Figure 2 (a), consists of a point embedding network and a cross-attention set correspondence estimator. The point embedding network extracts per point features using a Dynamic Graph CNN (DGCNN) [24]. The set correspondence estimator, shown in Figure 2 (b), then combines features from each object through cross-attention. These cross-object features are used to compute a per-point soft correspondence over the sets present in the other object. For each point in each object cloud, we compute the correspondence probability vector, $\mathbf{w_p} \in \mathbb{R}^{N+1}$, which we use to partition each cloud into $N+1$ correspondence sets (N matched sets and non-contact points):

$$\mathcal{S}_A^i = \{\mathbf{p} \mid \mathbf{p} \in \mathbf{P}_A \wedge \underset{j}{\operatorname{argmax}}\, \mathbf{w_p}(j) = i\} \tag{5}$$

$$\mathcal{S}_B^i = \{\mathbf{p} \mid \mathbf{p} \in \mathbf{P}_B \wedge \underset{j}{\operatorname{argmax}}\, \mathbf{w_p}(j) = i\}. \tag{6}$$

For $i \in [1, N+1]$, set $\mathcal{S}_A^i$ corresponds to set $\mathcal{S}_B^i$ and visa versa; $\mathcal{S}_A^0$ and $\mathcal{S}_B^0$ represent non-contact points in object A and object B that have no set correspondence with the other object.

B. Set Alignment Module

Given N set correspondences $(\mathcal{S}_A^i, \mathcal{S}_B^i)$ for $i \in [1, N+1]$, the set alignment module, shown in Figure 2 (c), aims to determine a transformation $\mathbf{T}_{AB}$ such that (4) is satisfied for all pairs of $(\mathcal{S}_A^i, \mathcal{S}_B^i)$. If we define $N \geq 3$ sets whose centroids, $\mathbf{c_A^i}$ and $\mathbf{c_B^i}$, are not co-linear, the transform $\mathbf{T}_{AB}$ can be computed as the least square solution to minimizing centroid distances in closed form [25]. For $N < 3$, solving from centroid correspondences is under-constrained, and supports many possible $\mathbf{T}_{AB}$ that aligns the set correspondences.

For width mate objects, we set $N = 2$, where the two set correspondences represent the top and bottom surfaces of the pegs and holes, and align the two pairs of corresponding set centroids, $(\mathbf{c_A^0}, \mathbf{c_B^0})$, and $(\mathbf{c_A^1}, \mathbf{c_B^1})$. Solving for an least square minimal transform $\hat{\mathbf{T}}_{AB}$ to align these centroids leaves us with an unconstrained rotation $\mathbf{R}_{\vec{\omega}}(\theta)$ of angle θ about the axis $\vec{\omega} = \mathbf{c_B^1} - \mathbf{c_B^0}$.

Hence, we can frame the problem in the specific case of width mates as solving the following optimization problem:

$$\theta^\star = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N+1} d(\mathbf{R}_{\vec{\omega}}(\theta)\hat{\mathbf{T}}_{AB}\mathcal{S}_A^i, \mathcal{S}_B^i), \tag{7}$$

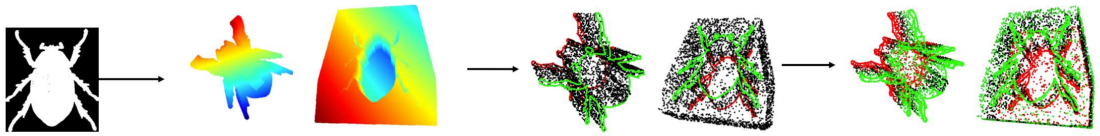
where d represents a general point cloud distance metric. This problem formulation also helps to reduce the effects of noises between set correspondences, potentially caused point cloud sampling or errors in set correspondence predictions.

To predict cross-pose from set correspondences between width mate objects, our method performs a two-step set alignment procedure: 1) align both objects according to their set correspondences, and 2) search for the best angle of rotation $\theta^\star$ according to (7). We compare different methods of optimizing the free rotation in step 2, including learning-based methods, Iterative Closest Point (ICP) [26], a search over the discretized 2d rotation space, and combining the search with ICP. For non-search based methods, we allow for an additional full 6D refinement transform to be estimated. Results of the ablation study can be found in Section V-A.

C. Matching Network

As the set correspondences retrieved from the previous task capture the potential contact surfaces of objects, we believe they provide crucial information regarding whether with-mate objects fit together. To explore this assumption, we propose SCMatcher for the task of relational inference between objects. Specifically, our goal is to learn a function $f_M(\mathbf{P}_A, \mathbf{P}_B)$ which outputs 1 if object A and object B are a positive pair that will fit together under the optimal cross-pose, and 0 otherwise.

SCMatcher shares the same architecture as the SCN, with the following modifications. First, a single feature from both objects is extracted by applying global max-pooling to the cross-attention features. These features are then concatenated and passed to a MLP with sigmoid activation to predict the



(a) MPEG7 (b) SynWidth (c) Hard Ground Truth Set Labels (d) Soft Ground Truth Set Labels (ε=0.3)

Fig. 3: Example from SynWidth. (a) Image from MPEG7[27] used to generate inner and outer intercepts for width mate generation. (b) Generated point clouds and meshes. (c) Downsampled point clouds at training and inference time, along with ground truth set correspondences denoted in red and green. (d) Soft set correspondences by relaxing the constraint of distance to ground truth correspondences. The tolerance ϵ is set to 0.3 for visualization purpose. Actual training procedure uses $\epsilon = 0.1$.

matching probability of the two objects. Second, the inputs to this matching network may optionally be transformed using a cross-pose alignment method. We explore the effects of this preprocessing step on matching accuracy of our method in Section V-B.

D. Training Details

Set Correspondence Network. Our set correspondence network is trained with cross-entropy loss between the predicted classification probability $\hat{\mathbf{w}}_{\mathbf{p}}$ and the ground truth set assignment $\mathbf{w}_{\mathbf{p}}$. The hard ground truth set assignment is defined as

$$\mathbf{w}_{\mathbf{p}}(i) = \begin{cases} 1 \text{ if } \mathbf{p} \in \mathcal{S}^i \\ 0 \text{ otherwise} \end{cases} \quad \forall i > 0, \tag{8}$$

and $\mathbf{w}_{\mathbf{p}}(0) = 1 - \max_{i>0} \mathbf{w}_{\mathbf{p}}(i)$, where $\mathbf{w}_{\mathbf{p}}(0)$ represents the probability that $\mathbf{p}$ does not belong to any set correspondence. In the hard assignment case, $\mathbf{w}_{\mathbf{p}}(i)$ is either zero or one with $\mathbf{w}_{\mathbf{p}}(0) = 1$ representing no set assignment > 0.

For training, we use a soft version of this ground truth set assignment, as a hard assignment can lead to the points near the boundaries being misclassified due to noise or point sampling ambiguity. This soft correspondence falls off smoothly, as the points move further from the set boundary, where

$$\sigma(\mathbf{p}, \mathcal{S}) = \max\left(0, 1 - \left(\frac{\min_{\mathbf{q}\in\mathcal{S}} ||\mathbf{q} - \mathbf{p}||_2}{\epsilon}\right)^3\right),$$

$$\mathbf{w}_{\mathbf{p}}(i) = \begin{cases} \sigma(\mathbf{p}, \mathcal{S}^i) \text{ if } \operatorname{argmax}_k \sigma(\mathbf{p}, \mathcal{S}^k) = i \\ 0 \text{ otherwise} \end{cases} \quad \forall i > 0,$$

with $\mathbf{w}_{\mathbf{p}}(0)$ being defined as before. An example illustrating the difference between the hard set assignment and the smoothed set assignment can be found in Figure 3(c) and (d). We supervised the training of SCAlign using cross-entropy loss with the soft ground truth set assignment, and updated the network parameters with the Adam optimizer [28] with a learning rate of $1e^{-4}$.

Matching Network. The matching network is trained with binary cross entropy loss against ground truth object relationships. The network parameters are similarly optimized with the Adam optimizer [28] with a learning rate of $1e^{-4}$.

V. Experiments

A. Cross-pose Evaluation on SynWidth

Dataset. We generate a synthetic dataset, SynWidth, to train and evaluate our method cross-pose method on width-mate objects with symmetries or near symmetries. This dataset contains a series of randomly generated pegs and holes, whose cross-sections are sampled from the binary images of the MPEG7 [27] and BinaryShape [29] datasets. The dataset is split into training, validation, and test sets, each containing 2k, 246, and 246 pairs of width-mate pairs. An example of a width mate pair from SynWidth can be found in Figure 3.

Evaluation Metrics. To evaluate the quality of our cross pose estimates, we use a variety of point cloud distance metrics. Given point clouds $\mathbf{P}_A$ and $\mathbf{P}_B$ of objects A and B, we measure the difference between predicted cross-pose $\mathbf{T}_{AB} = f(\mathbf{P}_A, \mathbf{P}_B)$ and the ground truth pose $\mathbf{T}_{AB}^{GT}$ as the distance between the point clouds transformed using the predicted pose versus the ground truth pose. Specifically, we measure this distance using the Chamfer Distance (CD) [30], and Earth Mover's Distance (EMD) [31]:

$$D_{\text{CD}} = \text{CD}(\mathbf{T}_{AB}\mathbf{P}_A,\ \mathbf{T}_{AB}^{GT}\mathbf{P}_A) + \text{CD}(\mathbf{T}_{AB}^{-1}\mathbf{P}_B,\ \mathbf{T}_{BA}^{GT}\mathbf{P}_B),$$
$$D_{\text{EMD}} = \text{EMD}(\mathbf{T}_{AB}\mathbf{P}_A,\ \mathbf{T}_{AB}^{GT}\mathbf{P}_A) + \text{EMD}(\mathbf{T}_{AB}^{-1}\mathbf{P}_B,\ \mathbf{T}_{BA}^{GT}\mathbf{P}_B),$$

In addition to CD and EMD, we propose another metric, Surface-point Penetration Distance (SPD), to quantify the violation of geometric constraint of the estimated cross-pose. This metric penalizes predicted cross-pose that result in the two objects penetrating each other. SPD is computed between points $\mathbf{P}$ and watertight mesh $\mathbf{M}$, as the negative sum of the signed distance measured over all transformed points that penetrate the other object's mesh:

$$D_{\text{SPD}} = \text{SPD}(\mathbf{T}_{AB}\mathbf{P}_A, \mathbf{M}_B) + \text{SPD}(\mathbf{T}_{AB}^{-1}\mathbf{P}_B, \mathbf{M}_A),$$
$$\text{where SPD}(\mathbf{P}, \mathbf{M}) = -\sum_{\mathbf{p}\in\mathbf{P}} \min(\text{sdf}_{\mathbf{M}}(\mathbf{p}), 0)$$

and $\text{sdf}_{\mathbf{M}}$ refers to the signed distance function of mesh $\mathbf{M}$, such that $\text{sdf}_{\mathbf{M}}(\mathbf{p}) < 0$ means point $\mathbf{p}$ is inside mesh $\mathbf{M}$.

Note that SPD alone is not sufficient for evaluating the predicted cross-pose. For example, the SPD would be zero when the objects are separated by a large distance, as they have no contact at all. Therefore, SPD should be examined in context of the other pose errors ($D_{\text{CD}}, D_{\text{EMD}}$).

Baselines. We compare our method against two cross-pose estimation baselines, TaxPose [1] and LNDF [2]. We trained TaxPose using two variations of point distance metrics, L2 loss and Chamfer Distance [30] loss, to explore the impact of using a symmetry-agnostic metric. LNDF is optimized to learn a neural descriptor field for occupancy reconstruction, and predicts cross-pose at inference time by optimizing alignment of querying points on training objects. The baselines are trained and evaluated on the same splits of SynWidth as our method.

Method	Chamfer Distance (mm)↓	Earth Mover's Distance (m) ↓	Surface Point Penetration Distance (m)↓	Inference Time (second per sample) ↓
Identity	70.99	387.633	20.997	**0.0062**
LNDF [3]	65.41	404.689	27.966	26.126
TaxPose [1]	23.31	260.092	27.156	0.0173
SCAlign	**2.868**	**22.111**	**15.242**	0.5490

TABLE I: Evaluation of Cross-pose Prediction on SynWidth.

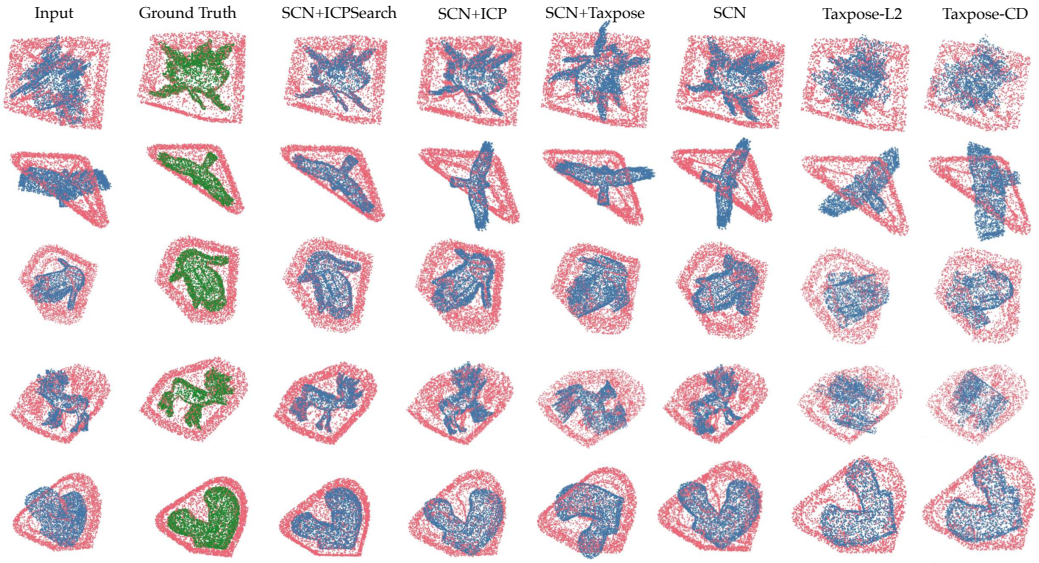


Fig. 4: Results on SynWidth. Green points represent $\mathbf{P}_A$ transformed by ground truth cross-pose $\mathbf{T}_{AB}^{GT}$. Blue points represent $\mathbf{P}_A$ transformed by predicted cross-pose $\mathbf{T}_{AB}$. Pink points represent $\mathbf{P}_B$.

Results. We summarize the results of our method and the baselines on SynWidth in Table I. On all evaluation metrics, our method significantly outperforms the baselines. We believe that the baselines fail to predict cross-pose accurately due to symmetries in object geometries. Although adopting a symmetry-agnostic Chamfer Distance [30] loss improves TaxPose's results compared to training with the with L2 loss, the optimization fails to correctly align contact surfaces with symmetries.

Additionally, we include an ablation study over different methods for solving for the unconstrained rotation in (7). For all of the methods, we use Chamfer Distance [30] as the distance metric d. **SCAlign** corresponds to only aligning the set centroids, and skipping any optimization of the free rotation. **SCAlign + TaxPose**, **SCAlign + ICP**, and **SCAlign + Search** correspond to centroid alignment followed by TaxPose [1], ICP [26], and search over the discretized rotation space respectively. **SCAlign + ICPSearch** corresponds to a descritized rotation search followed by an ICP refinement step, to account for error caused by discretization and potential noise in set correspondences. Among all methods, **SCAlign + ICPSearch** has the best performance in terms of CD, EMD, and SPD losses. Qualitative evaluations of each method can be found in Figure 4.

B. Matching Evaluation on SynWidth

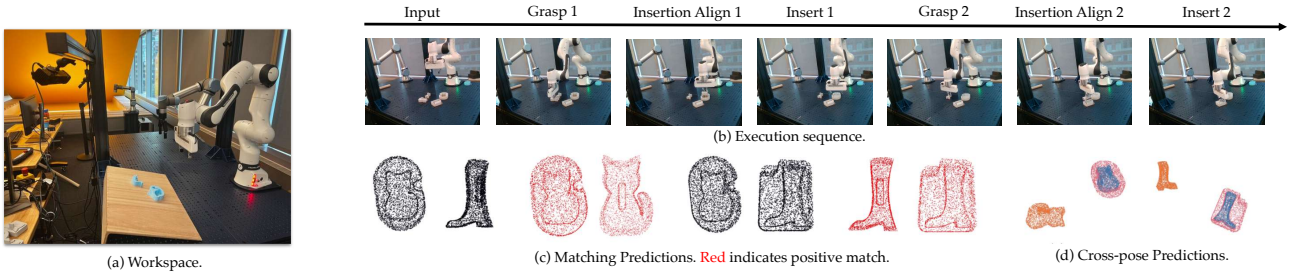
Dataset. To evaluate the object matching task, we leverage the same SynWidth dataset, but extend it with the equal number of negative object pairs. This matching dataset is split into training, validation, and test splits, each containing 4k, 492, and 492 pairs of objects. We evaluate the matching methods based on classification accuracy.

Method	Matching Accuracy ↑	Inference Time (second per sample) ↓
SCMatcher (allpoints)	0.6992	**0.1124**
TaxPose + SCMatcher (allpoints)	0.7989	0.1303
SCAlign + SCMatcher (allpoints)	0.8984	0.5694
SCAlign + SCMatcher (correspondences)	**0.9512**	0.5960

TABLE II: Evaluation of Matching on SynWidth.

C. Franka Insertion and Matching

Results. The results of matching task on SynWidth is summarized in Table II, along with an exploration of different preprocessing transformations to the inputs of SCMatcher. **SCMatcher (allpoints)** corresponds to using the full point clouds $\mathbf{P}_A$ and $\mathbf{P}_B$ in their arbitrary initial poses as input to the matching network. This input type requires the model to implicitly learn features that are robust to SE(3) transforms. We also evaluate various pre-alignment approaches. This should simplify the matching problem, but is subject to the errors present in each cross-pose method. **TaxPose + SCMatcher (allpoints)** corresponds to aligning the input point clouds based on the cross-pose predicted by TaxPose [1]. This method does improve matching, but the errors in TaxPose's alignment can affect the matching accuracy. Similarly, **SCAlign + SCMatcher (allpoints)** corresponds using the cross-pose estimated by SCAlign to align the input point clouds. While this method further improves performance, it includes many non-critical points to the object matching,



(a) Workspace.

(b) Execution sequence.

(c) Matching Predictions. Red indicates positive match.

(d) Cross-pose Predictions.

Fig. 5: Matching experiments on Franka arm using SCAlign and SCMatcher. (a) Key frames from video recording of the matching experiment. **(b)** Matching result from SCMatcher. Red indicates positive matches, and black indicates negative matches. **(c)** Cross-pose estimation between two width mate objects.

such as the points on the outer edges of the hole. **SCAlign + SCMatcher (correspondences)** further simplifies the problem by filtering the aligned input clouds to only include points in set correspondences. This amounts to only matching using aligned point on surface contact points and achieves the highest matching accuracy. We validate the utility of our method for robotic applications by conducting real world insertion and matching experiments using a Franka arm and a Photoneo camera. Note that our method requires models of objects of interests, which can be obtained from a scanning procedure. A picture of the workspace setup is provided in Figure 5(a). The pipeline of our Franka experiments consists of the following components:

1) Photoneo camera that captures RGB and depth images of our workspace.
2) GroundingDINO [32] to detect objects of interest in the workspace, and SegmentAnything [33] to extract segmentation masks for each detection.
3) Instance mesh registration module that identifies association of pre-scanned object meshes with each object segmentation. Object poses are estimated using MegaPose [34].
4) If there are more than two objects present in the scene, SCMatcher is used to identify the positive pairs.
5) For each positive pair, SCAlign is applied to predict the cross-pose between the objects.
6) Waypoints are planned for grasping and insertion. Position control of Franka arm is executed via Deoxys [35] interface.
7) A simple insertion behavior is added to account for control error from Deoxys [35], where the robot explores nearby area ($\pm$1.5cm) until reaching the desired position.

Insertion Experiments. We test our method on positive pairs of 3D printed peg and hole objects not found in our training dataset and achieve a success rate of 83% over 12 trials. Visualizations of part of the experiment trials are provided in Figure 6.

Matching Experiments. Additionally, we test our matching method with multiple pairs of peg and hole objects. Our method is able to correctly match and insert each of two pegs into their corresponding holes. Visualization of a matching trial is provided in Figure 5, along with the matching and cross-pose predictions from SCMatcher and SCAlign.

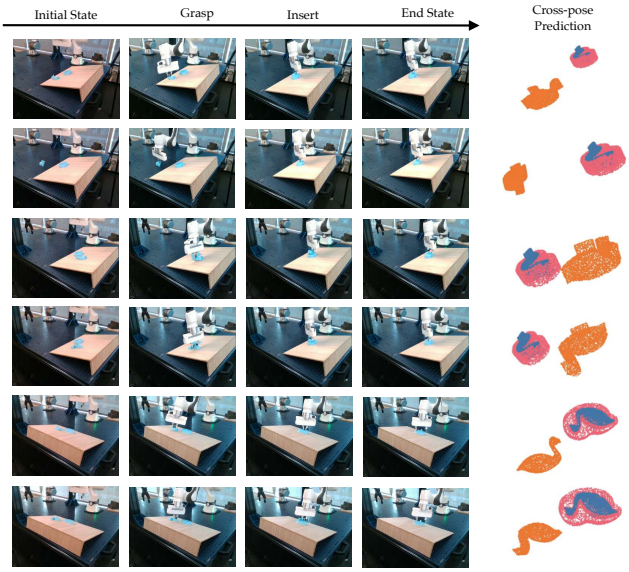


Fig. 6: Insertion experiments on Franka arm using SCAlign. Visualizations of cross-pose prediction indicate hole points with pink, peg points with orange, and peg points transformed by predicted cross-pose with blue.

VI. Conclusions and Future Work

In this paper, we propose SCAlign, a new method for predicting cross-pose from set correspondences. This method expands cross-pose estimation to symmetric and near-symmetric width-mate object interactions, and significantly outperforms other cross-pose estimation methods [1], [2] on SynWidth, a synthetic datasets with width-mate objects of symmetric or near-symmetric cross-sections. Additionally, we demonstrate that the set correspondences provide useful clue for relational inference between objects, where SCMatcher taking in aligned set correspondences as input outperforms the other baselines in the object matching task. Finally, we show that our method can be applied to real world insertion and matching experiments with a Franka arm. All this being said, we acknowledge our method is limited in that it requires object model, which could be potentially expensive to obtain in the real world scenarios. We leave applying our method to partially observed point cloud as a topic to explore in future research.

Acknowledgment

This work was done during an internship with the Boston Dynamics AI Institute.

REFERENCES

[1] Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning*, 2022.

[2] Ethan Chun, Yilun Du, Anthony Simeonov, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Local neural descriptor fields: Locally conditioned object representations for manipulation. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1830–1836, 2023.

[3] 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. *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400, 2021.

[4] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[5] 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*, pages 3523–3532, 2019.

[6] Anthony Simeonov, Yilun Du, Lin Yen-Chen, Alberto Rodriguez, Leslie Pack Kaelbling, Tomas Lozano-Perez, and Pulkit Agrawal. Se(3)-equivariant relational rearrangement with neural descriptor fields, 2022.

[7] Hyunwoo Ryu, Hong in Lee, Jeong-Hoon Lee, and Jongeun Choi. Equivariant descriptor fields: Se(3)-equivariant energy-based models for end-to-end visual robotic manipulation learning, 2023.

[8] Ondrej Biza, Skye Thompson, Kishore Reddy Pagidi, Abhinav Kumar, Elise van der Pol, Robin Walters, Thomas Kipf, Jan-Willem van de Meent, Lawson L. S. Wong, and Robert Platt. One-shot imitation learning via interaction warping, 2023.

[9] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration, 2022.

[10] R Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J Mitra, and Daniel Ritchie. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG)*, 39(6):1–20, 2020.

[11] 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.

[12] 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.

[13] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[14] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[15] 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.

[16] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[17] Fares J Abu-Dakka, Bojan Nemec, Jimmy A Jørgensen, Thiusius R Savarimuthu, Norbert Krüger, and Aleš Ude. Adaptation of manipulation skills in physical contact with the environment to reference force profiles. *Autonomous Robots*, 39:199–217, 2015.

[18] Yunsheng Tian, Jie Xu, Yichen Li, Jieliang Luo, Shinjiro Sueda, Hui Li, Karl DD Willis, and Wojciech Matusik. Assemble them all: Physics-based planning for generalizable assembly by disassembly. *ACM Transactions on Graphics (TOG)*, 41(6):1–11, 2022.

[19] Zhimin Hou, Jiajun Fei, Yuelin Deng, and Jing Xu. Data-efficient hierarchical reinforcement learning for robotic assembly control applications. *IEEE Transactions on Industrial Electronics*, 68(11):11565–11575, 2020.

[20] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[21] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5548–5555. IEEE, 2020.

[22] Lin Shao, Toki Migimatsu, and Jeannette Bohg. Learning to scaffold the development of robotic manipulation skills. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5671–5677. IEEE, 2020.

[23] Mel Vecerik, Oleg Sushkov, David Barker, Thomas Rothörl, Todd Hester, and Jon Scholz. A practical approach to insertion with variable socket position using deep reinforcement learning. In *2019 international conference on robotics and automation (ICRA)*, pages 754–760. IEEE, 2019.

[24] 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)*, 38:1 – 12, 2018.

[25] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.

[26] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:239–256, 1992.

[27] Longin Jan Latecki, Rolf Lakämper, and Ulrich Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, 1:424–429 vol.1, 2000.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[29] Binary shape databases. `https://vision.lems.brown.edu/content/available-software-and-databases`.

[30] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2016.

[31] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000.

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[34] Yann Labb'e, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Conference on Robot Learning*, 2022.

[35] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *arXiv preprint arXiv:2210.11339*, 2022.