2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

SCARP: 3D Shape Completion in ARbitrary Poses for Improved Grasping

Bipasha Sen*[1], Aditya Agarwal*[1], Gaurav Singh*[1], Brojeshwar B.[2], Srinath Sridhar[3], and Madhava Krishna[1]

***Abstract*— Recovering full 3D shapes from partial observations is a challenging task that has been extensively addressed in the computer vision community. Many deep learning methods tackle this problem by training 3D shape generation networks to learn a prior over the full 3D shapes. In this training regime, the methods expect the inputs to be in a fixed canonical form, without which they fail to learn a valid prior over the 3D shapes. We propose SCARP, a model that performs Shape Completion in ARbitrary Poses. Given a partial pointcloud of an object, SCARP learns a disentangled feature representation of pose and shape by relying on rotationally equivariant pose features and geometric shape features trained using a multi-tasking objective. Unlike existing methods that depend on an external canonicalization method, SCARP performs canonicalization, pose estimation, and shape completion in a single network, improving the performance by 45% over the existing baselines. In this work, we use SCARP for improving grasp proposals on tabletop objects. By completing partial tabletop objects directly in their observed poses, SCARP enables a SOTA grasp proposal network improve their proposals by 71.2% on partial shapes. Project page: https://bipashasen.github.io/scarp**

I. INTRODUCTION

Given a partial observation of an object, 3D shape completion aims to recover the full 3D shape of the object. This has been widely addressed in computer vision [1]–[8] and has many diverse downstream applications in robotics including visual servoing [9], manipulation [10]–[13], visual inspection [14], autonomous driving [15]–[17].

Many existing methods tackle shape completion by incorporating a training scheme that learns a prior over the full 3D shapes. This is done by training an autoencoder [1], [6], [18], [19] or a GAN [20] over many different instances of full shapes. At inference, this learned prior space is conditionally queried on the partial observations. These methods however, suffer from a major limitation: they expect the partial input to be in a fixed canonical frame–a common frame of reference that is shared between instances in that category [21], [22]. A particular shape X in two different poses $\{R_1, T_1\}$ and $\{R_2, T_2\}$ will have very different geometry. As a result, X in different poses appear as novel instances for these methods inhibiting them from learning a valid prior over shapes.

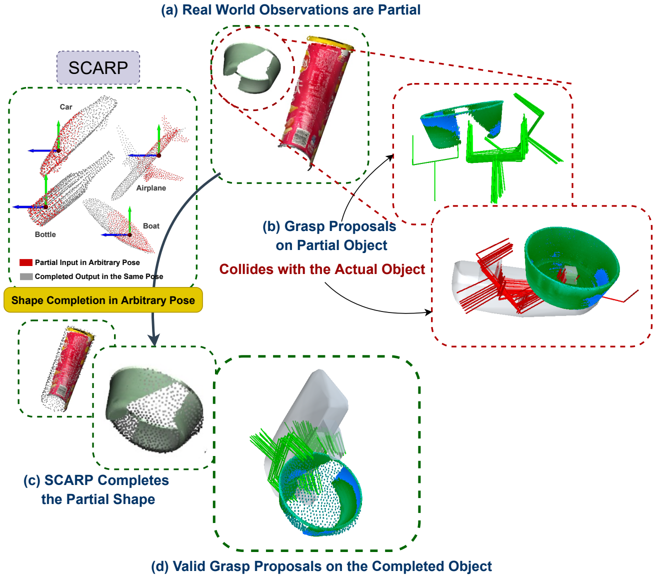
(a) Real World Observations are Partial
SCARP
Car
Airplane
Bottle
Boat
Partial Input in Arbitrary Pose
Completed Output in the Same Pose
Shape Completion in Arbitrary Pose
(b) Grasp Proposals
on Partial Object
Collides with the Actual Object
(c) SCARP Completes
the Partial Shape
(d) Valid Grasp Proposals on the Completed Object

Fig. 1: SCARP performs Shape Completion in ARbitrary Poses (top-left). We show an example of a real scene made of two tabletop objects. (a) The captured scene is partial leaving out a portion of the objects. (b) This results in grasp poses (in green) on the partial point cloud that directly collide (in red) with the actual object leading to a collision between the object and the Franka Panda's gripper (grey). (c) SCARP improves grasp proposal by accurately completing the partial pointcloud in the observed pose. (d) This enables the grasp proposal network to propose grasp poses (shown in green) on the completed pointcloud that do not collide with the actual object.

Existing datasets like ShapeNet [23] have shapes that are manually aligned to a canonical frame, but real shape observations (e.g., depth maps) do not contain this information. A naive approach to tackling this challenge is to *canonicalize*, i.e., map a 3D (full or partial) shape to a category-level canonical frame with [21] or without supervision [22], [24], [25]. A multi-stage pipeline can be built involving the sequential steps of (1) canonicalization, (2) shape completion, and (3) de-canonicalization (bringing the object back in the original pose). In such a pipeline however, the performance of a shape completion network directly depends on the output quality of the canonicalization module. This can lead to errors propagating between these modules leading to a sub-optimal completion.

We propose SCARP, a method that performs **S**hape **C**ompletion in **AR**bitrary **P**oses. Unlike existing methods that have to directly learn a prior over all possible poses and shapes, we first disentangle the pose from the shape of a partial pointcloud. We build a multi-task objective that: (1) generates a disentangled feature representation of pose and shape by canonicalizing an object to a fixed frame of reference, (2) estimates the exact pose of the object, and (3) completes the shape of the object using the disentangled

*Equal Contributions
[1] Robotics Research Center, IIIT-Hyderabad
[2] TCS Research, India
[3] Brown University

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

representation. This multi-task objective allows our network to jointly understand the pose and shape of the input. It does so by learning rotationally-equivariant and translationally-invariant pose features using Tensor Field Networks [26], and global geometric shape features using PointNet++ [27].

Application: Robotic grasp pose estimation [12], [13], [28], [29] is a challenging area of research that often expects a faithful reconstruction of the scene in 3D. As shown in Fig. 1 (b), under a partial observation, [13] generates grasp proposals that directly collide with the actual object in the scene (shown in red). As a result, the manipulator is likely to collide with the object as it attempts to grasp the objects using one of these predicted grasp poses. We use SCARP to complete these partial shapes directly in their observed poses and estimate grasp proposals on these completed shapes. We show that SCARP reduces such invalid grasps by 71.2% over predicting grasp poses directly on the partial observations. To summarize, our contributions are:

1) We propose SCARP, a novel architecture to perform shape completion from partial pointclouds in arbitrary poses.
2) We show for the first time how a multi-task objective can support: (1) canonicalization, (2) 6D pose estimation, and (3) shape completion on partial pointclouds.
3) We demonstrate that SCARP outperforms the existing shape completion baselines (with pre-canonicalization) by 45% and improves grasp pose estimation by reducing invalid grasp poses by 71%.

II. Related Work

Partial Pointcloud Completion has been extensively addressed over the years [1], [2], [6], [7], [18]–[20], [30]–[34]. Early 3D shape completion works relied on intermediate voxel representation for representing the 3D objects [35]–[37]. More recent works adopted an architecture similar to PointNet [38] that aggregated point-wise embeddings to achieve a global pointcloud feature. PCN [19] adopted such an architecture and pioneered learning-based methods for pointcloud completion. More recent works [1], [20] adopted a pointcloud generator network to learn a prior over the full pointcloud shapes. [20] applied a differential degradation layer on the output of a pointcloud generator [39] to obtain partial pointclouds from full outputs. Later, it relied on conditional GAN inversion [40] to obtain the latent code of a partial pointcloud's full output. [1] used a similar approach by training a quantized auto-encoder and adopting a second transformer based network to learn the shape prior over the quantized space. Other lines of work aim to generate high resolution pointclouds [6], [7], preserving high-details in the output. All of these existing methods expect the partial pointcloud to be in a fixed canonical frame. As a result, they depend on external canonicalization methods [21], [22], [24] to bring the partial inputs to their fixed frame. We explicitly disentangle the pointcloud's pose and shape and train our network to learn a prior over the pose and the shape separately. This allows us to perform partial pointcloud completion directly in any arbitrary pose. [2], [41] aim to complete pointclouds in arbitrary poses but need multiple observations from multiple views for completing the pointcloud. Capturing a particular object from multiple different views is not possible always, especially in a moving frame of reference like moving cars. [42] performs pointcloud completion with limited disturbance from the canonical pose and uses temporal information for completion. Unlike them, we do not assume any restriction in the pose and perform pointcloud completion from a single observation.

Pointcloud Canonicalization canonicalizes an input pointcloud of a given category to the category's fixed canonical frame. This canonical frame is implicitly defined by the network [22], [24]. Such a method can be clubbed with the existing shape completion networks in a multi-stage pipeline: (1) canonicalizing the partial pointcloud to a fixed implicit frame, (2) shape completion in the fixed implicit frame, and (3) de-canonicalization. However, the performance of the shape completion network is tightly coupled with the canonicalization method. Morever, there is an additional training overhead in such a method, where the existing dataset has to be first converted to the implicit canonical frame before the shape completion model is trained. Our model does not need an external canonicalization. We compare our model with the multi-stage pipeline by clubbing [22] with the existing shape completion networks [7], [20] and show substantial performance gain on shape completion metrics.

III. Background

Pointnet++ [27] is a 3D hierarchical network, P, that computes the geometric shape feature, p, for a given pointcloud $X \in \mathbb{R}^{3\times K}$, where K is the user-defined number of points in the pointcloud. It computes a global geometric feature by hierarchically aggregating local geometric features of the points in a pointcloud. For a more detailed understanding, please refer [27], [38]. These features are not rotation-aware. That is, for a point cloud X in any new orientation $\hat{R}(X)$, where $\hat{R} \in SO(3)$, a unique p is generated.

Tensor Field Networks [26], [43] is a 3D architecture, $\mathcal{X}$, that computes rotationally equivariant and translationally invariant feature matrix, $\hat{F}$. For a given pointcloud $X \in \mathbb{R}^{3\times K}$ and an integer (aka type) $\ell \in \mathbb{N}$, a TFN produces global (type ℓ) feature vectors of dimension $2\ell+1$ stacked in a matrix $\hat{F}^\ell \in \mathbb{R}^{(2\ell+1)\times N}$, where N is user-defined number of channel. $\hat{F}_{:,j}(X)$ satisfies the equivariance property $\hat{F}_{:,j}(RX) = D(R)\hat{F}_{:,j}(X)$, where $D : SO(3) \to SO(2\ell{+}1)$ is a Wigner matrix (of type ℓ) [44]–[46]. Additional details can be found in [26], [43], [47]–[49].

tree-GAN [39] is a GAN-based pointcloud generation network that adopts a graph convolutional generator, G, and a discrimintor similar to r-GAN [50]. G hierarchically upsamples a gaussian noise, $z \in \mathbb{R}^E$, sampled from a standard normal distribution to a pointcloud $X' \in \mathbb{R}^{3\times K}$.

Density Aware Chamfer's Distance [51] Given two pointclouds X and $Y \in \mathbb{R}^{3\times K}$ without known point-wise correspondences, Chamfer's Distance (CD)[1] can be used to

[1] https://pdal.io/en/stable/apps/chamfer.html

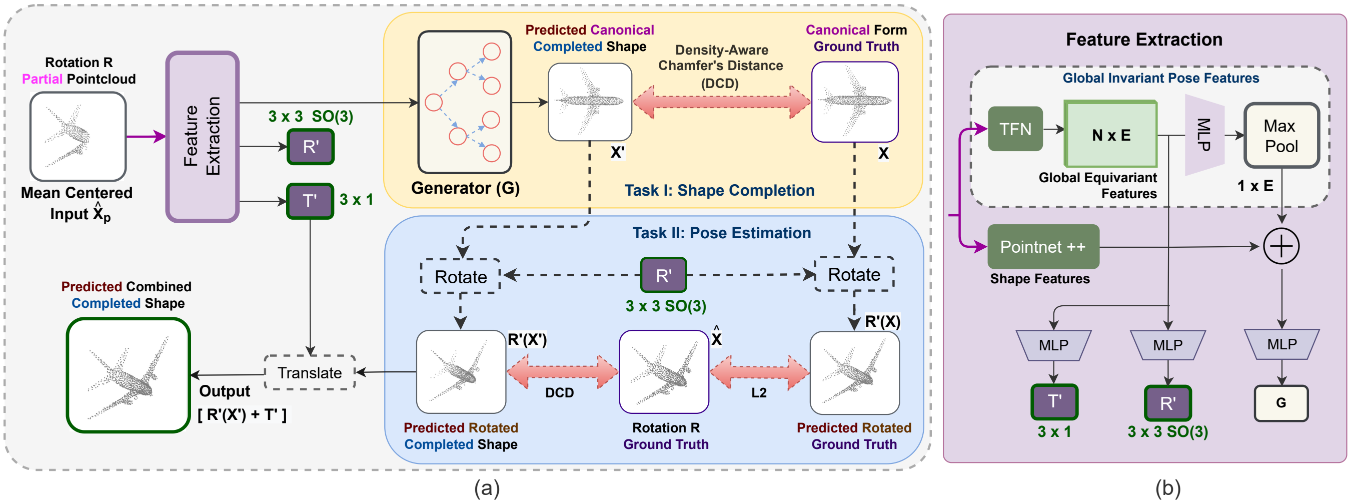

Fig. 2: **Overview of our proposed approach:** The input to SCARP is a mean-centered partial pointcloud $\hat{X}_p$ in an arbitrary orientation R. Our feature extraction module **(b)** disentangles the partial pointcloud's pose and shape and is trained in a multi-tasking objective **(a)**. In the first task, SCARP combines Pointnet++ [27] and TFN [26] features to generate a shape feature that is used by a pointcloud completion network, G, to generate X'. In the second task, the TFN pose feature is used to generate an equivariant frame $\{R', T'\}$. Our loss functions enable the overall network to learn a prior over the shape while understanding the pose of the partial input.

compute the distance $d_{CD}(X, Y)$ by considering the nearest neighbor of the points $\{x \in X, y \in Y\}$ in $\{Y, X\}$ as their correspondence. The distance is then given as:

$$\sum_{x\in X}\min_{y\in Y}||x-y||_2^2+\sum_{y\in Y}\min_{x\in X}||x-y||_2^2 \qquad (1)$$

However, CD does not guarantee uniformity in the output density. Density-Aware Chamfer's Distance (DCD) [51] overcomes this issue by modifying CD as:

$$\frac{1}{2}\left(\frac{1}{|X|}\sum_{x\in X}\left(1-\frac{e^{\mathcal{Z}_x}}{n_{\hat{y}}}\right)+\frac{1}{|Y|}\sum_{y\in Y}\left(1-\frac{e^{\mathcal{Z}_y}}{n_{\hat{x}}}\right)\right) \qquad (2)$$

Please refer to [51] for the exact notations.

IV. SCARP: Shape Completion in ARbitrary Poses

Given a partial object pointcloud $\hat{X}_p$ at an unknown pose $\{R, T\}$, we want to estimate this pose and the corresponding full object pointcloud $\hat{X}$ in the same pose.

This is a challenging task as for a neural network, a pointcloud X in two different poses $\{R_1, T_1\}$ and $\{R_2, T_2\}$ are two completely different pointclouds. Thus, we adopt a multi-tasking objective that disentangles the pose and the shape of the input partial pointcloud $\hat{X}_p$. The shape component allows us to understand that $\hat{X}_p$ is a partial observation of X which is $\hat{X}$ in its canonical form. The pose component is then used to estimate the pose transform $\{R, T\}$ between $\hat{X}$ and X.

A. Multi-tasking Pipeline for disentangling Shape and Pose

Let X_p and X be a partial and its corresponding full pointcloud in a fixed canonical frame. Then $\hat{X}_p$ and $\hat{X}$ are X_p and X in an unknown arbitrary pose $\{R, T\}$ such that $\hat{X}_p = R(X_p) + T$ and $\hat{X} = R(X) + T$. The input to our network is $\hat{X}_p$ which is mean centered at the origin and normalized to a unit bounding box. Our aim is to predict $\{R, T\}$ and the full pointcloud $\hat{X}$ which is posed as:

$$\{R, T, \hat{X}\} = \Phi(\hat{X}_p) \qquad (3)$$

where Φ denotes our proposed network, SCARP.

Our multi-tasking objective is formulated to (1) complete the partial pointcloud in a fixed canonical frame given by X and (2) estimate the pose transformation from the canonical frame to the original pose $\{R, T\}$. In this pipeline, the two components (1) pose and (2) shape are predicted separately using two different output heads as shown in Fig. 2.

1) Feature Extraction: To estimate the input's shape, we compute global geometric shape features, $p \in \mathbb{R}^E$, using Pointnet++ [27] as explained in Sec. III. To estimate the pose of the input, we adapt TFN [26] as explained in Sec. III. Our TFN computes a global equivariant feature, $F \in \mathbb{R}^{N\times E}$ by max pooling over the types $\{\ell\}_{\ell=0}^{\ell=\ell_{max}}$, where E is the dimension of the equivariant embeddings, N and ℓ_{max} are user-defined.

The input to our shape completion network is a non-linear combination of p and a global invariant embedding, $F_{\mathcal{X}} \in \mathbb{R}^E$, computed by max pooling F over the channel dimension, N. Additionally, F is used to estimate an equivariant frame of reference, $\{R' \in \mathbb{R}^{3\times3}, T' \in \mathbb{R}^3\}$ that transforms the invariant embeddings to X's original pose.

2) Task I: Shape Completion: Completing the shape of a partial input at any arbitrary orientation is difficult. Therefore, we aim to first complete the shape at a fixed canonical frame. To learn this canonical frame, the model needs to build an understanding of the full shape of the partial input. To achieve this, we train our model to predict a full canonicalized pointcloud $X' \in \mathbb{R}^{1024\times3}$ directly from $\hat{X}_p \in \mathbb{R}^{512\times3}$. Shape completion enables our model to learn a prior over the global shape of a category (a typical chair would have four legs and a backrest) enabling our network to directly canonicalize the partial inputs accurately.

We adopt G, as explained in Sec. III, as our shape completion network where (1) the input to G is a semantically meaningful embedding generated from a partial input $\hat{X}_p$ and (2) is trained using a distance loss against the full pointcloud X to learn a relationship between the partial input $\hat{X}_p$ and the predicted full canonical pointcloud X'. As shown in Fig. 2 (right), the input to G is a globally invariant feature vector $f \in \mathbb{R}^E$ computed by combining $p = P(\hat{X}_p)$ and $F_{\mathcal{X}} = \mathcal{X}(\hat{X}_p)$ non-linearly using a neural network ϕ_S given as:

$$X' = G(f) \quad \text{and} \quad f = \phi_S(\mathcal{X}(\hat{X}_p) \oplus P(\hat{X}_p))) \tag{4}$$

3) Task II: Pose Estimation: Once G predicts the full pointcloud X' in a canonical pose, it is important to estimate the correct rotation $SO(3)$ matrix $R \in \mathbb{R}^{3\times3}$ and translation $T \in \mathbb{R}^3$ to register X' back on $\hat{X}_p$. We predict R' and T' on the second head of our model using the rotationally equivariant TFN features F given as:

$$R' = \phi_R(F) \quad \text{and} \quad T' = \phi_T(F) \tag{5}$$

where ϕ_R and ϕ_T are multi-layered perceptrons.

B. Loss Functions for Multitask Training

1) Shape Completion in a fixed Canonical Frame: In the first task, we estimate the completed pointcloud in a fixed canonical frame given by X'. We use DCD (see Sec. III) to minimize the distance between the predicted pointcloud X' and the ground truth canonical pointcloud X given by:

$$\mathcal{L}_{shape} = d_{DCD}(X', X) \tag{6}$$

2) Estimating the pose of the object: To estimate the pose given by $\{R, T\}$, we use rotationally equivariant pose features F and pass it through $\{\phi_R, \phi_T\}$. We constrain this prediction against the canonical frame. To do so, we rotate the canonical output X' to obtain $R'(X')$ and compare it against the rotated ground truth $\hat{X}$. At this point however, the pointwise correspondences between X and X' are lost. Thus, a hard distance loss such as Euclidean distance cannot be directly used. To tackle this, we minimize permutation invariant CD objective as explained in Sec. III between X and X'. However, CD only minimizes the distance between the nearest neighbors of the points in the pointcloud. This results in local minimas where the loss is minimal even when the actual correspondences are far. As a result, the predicted pointcloud is often flipped about one of the axes. To tackle this issue, we rotate the canonical ground truth X using the predicted R' and compare against $\hat{X}$ using $L2$ loss. The overall loss is:

$$\mathcal{L}_{rot} = \delta d_{CD}(\hat{X}, R'(X')) + \gamma ||\hat{X}, R'(X)||_2 \tag{7}$$

$R'(X')$ is computed by detaching the forward computation graph at the output of G. The gradients from the loss does not backpropogate through G at the first head.

For symmetrical objects such as bowls and glasses, multiple R' predictions can be correct. A hard $L2$ loss penalizes the network for correct predictions even for correct R' if the correspondences do not exactly match. Thus, for symmetrical objects, we keep $\delta \sim 1.0$ and $\gamma \sim 0.0$ and for non-symmetrical objects we keep $\delta \sim 1.0$ and $\gamma \sim 1.0$.

The input to our network is a mean-centered partial pointcloud $\hat{X}_p$. At this point, we train our network to regress to $\hat{X}_p$'s centroid in the full pointcloud X given by T'. We directly supervise T' against the ground truth T given as:

$$\mathcal{L}_{trans} = ||T' - T||_2 \tag{8}$$

The final output is obtained by rotating and translating our predicted pointcloud X' by R' and T' respectively as:

$$X_o = R'(X') + T' \tag{9}$$

Orthonormality Loss: The rotation R' predicted by our network is a 3×3 matrix in the $SO(3)$ space. However, the matrix predicted by Eqn. 5 is not guaranteed to be a valid $SO(3)$ matrix. We therefore, enforce orthonormality on R' by minimizing its difference to its closest orthonormal matrix. To do so, we compute the SVD decomposition of $R = U\Sigma V^T$ and enforce unit eigenvalues as:

$$\mathcal{L}_{orth} = ||UV^T - R||_2 \tag{10}$$

3) Combined Loss: We train our network end-to-end by combining all the losses as:

$$\mathcal{L} = \mathcal{L}_{shape} + \mathcal{L}_{rot} + \mathcal{L}_{trans} + \mathcal{L}_{orth} \tag{11}$$

V. Experiments

In this section, we evaluate SCARP on two tasks: **(T1)** Shape completion in arbitrary poses and **(T2)** Improving grasp proposals by completing partial pointclouds.

Baselines: For comparison, we modify the existing shape completion networks by developing a multi-stage pipeline: (1) We use ConDor [22] to first canonicalize the input partial pointclouds to a fixed canonical frame defined implicitly by ConDor. (2) We train and test the existing shape completion methods on ConDor's canonical frame. (3) Bring the completed pointcloud to the original orientation using a pose transform predicted by ConDor. We compare against (1) ConDor+Pointr [7], a SOTA pointcloud completion network that generates high-resolution completed pointclouds and (2) ConDor+Shape Inversion (SInv.) [20] based on tree-GAN [39] that shares our generator G.

Metrics: We use Chamfer's Distance **(CD)** as explained in Sec. III to compute the distance between the ground truth pointcloud $\hat{X}$ and the predicted pointcloud given as $R'(X') + T'$ to evaluate the match in shape.

Earth Movers Distance-Maximum Mean Discrepancy **(MMD-EMD)** [39], [50] is used to evaluate for uniformity in the prediction by conducting bijective matching of points between two pointclouds. As we only want to measure the output's uniformity, we compute this metric between the canonical ground truth X and the canonical prediction X'.

We evaluate SCARP by measuring its impact in an important downstream task: grasp pose estimation. In this, we measure the a) the number of grasp proposals made on the

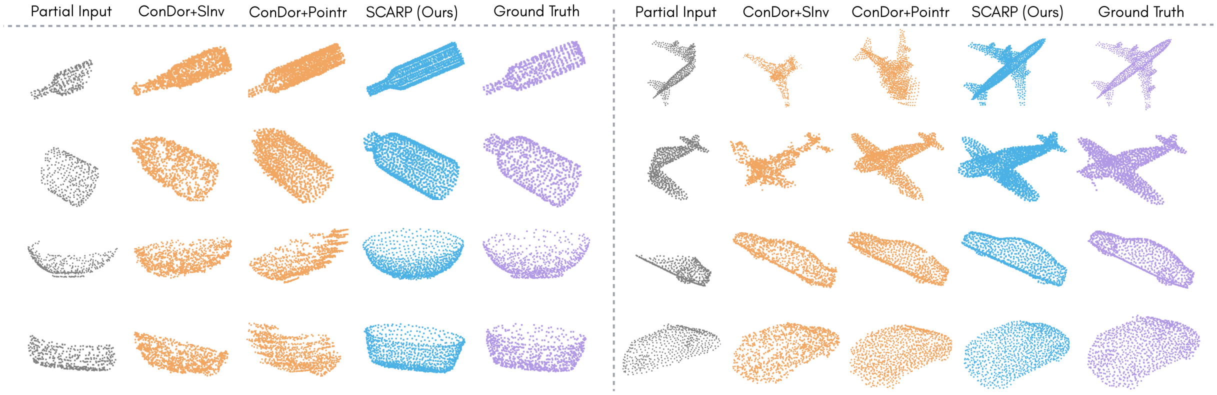

Fig. 3: Qualitative comparison of shape completion in arbitrary poses on SCARP and the existing multi-stage baselines: Canonicalization using ConDor, Shape Completion, and De-canonicalization. Pointr [7] is a SOTA pointcloud completion network that generates high-resolution completed pointclouds. Shape Inversion (SInv.) [20] is based on tree-GAN [39] that shares our generator G.

		Tabletop					Off-Table				
		Bowl	Bottle	Can	Mug	Basket	Plane	Car	Chair	Watercraft	Average
CD↓	ConDor+SInv.	82.7	27.4	45.4	41.5	85.3	34.2	14.7	59.4	39.9	47.8
	ConDor+Pointr	30.8	20.9	29.9	14.2	40.9	22.1	6.4	19.8	8.5	21.5
	SCARP (Ours)	**21.8**	**7.9**	**11.8**	**12.1**	**34.2**	**6.9**	**5.6**	**19.1**	**7.1**	**14.0**
		Bowl	Bottle	Can	Mug	Basket	Plane	Car	Chair	Watercraft	Average
MMD-EMD↓	ConDor+SInv.	27.3	17.2	20.1	19.9	29.2	19.6	11.3	22.2	18.9	20.6
	ConDor+Pointr	21.6	13.6	14.8	12.6	18.8	14.4	8.1	13.5	9.1	14.1
	SCARP (Ours)	**9.6**	**6.3**	**8.8**	**8.4**	**10.6**	**5.0**	**5.6**	**8.4**	**6.0**	**7.6**

TABLE I: Quantitative comparison of shape completion in arbitrary poses for tabletop and off-tabletop objects. Most tabletop objects are symmetrical whereas off-table objects have more variations in structure. Chamfer's Distance (CD) and Earth Movers Distance-Maximum Mean Discrepancy (MMD-EMD) are explained in Sec. V and are scaled by 10^3 and 10^2.

partial object that collide with the actual object on the table (shown in Fig. 1 and 4) denoted by C and b) number of invalid grasps that do not result in a valid grasping denoted by I. We then compute the Grasping Error **(GE)** as:

$$\frac{1}{\mathcal{D}} \sum_{i=1}^{\mathcal{D}} \frac{C+I}{\mathcal{N}} \tag{12}$$

where $\mathcal{N}$ is the number of top grasp proposals and $\mathcal{D}$ is the total number of pointcloud instances. In our case, $\mathcal{N} = 30$.

Dataset: Our dataset is a subset of [23] derived from [22] and [7] made of 5 tabletop (Bowl, Bottle, Can, Mug, Basket) and 4 non-tabletop (Plane, Car, Chair, Watercraft) categories. We evaluate GE only on the tabletop objects.

Results: As shown in Table II, SCARP outperforms the existing multi-stage baselines on all the categories on an average by 45%. The existing shape completion methods rely on the output of an external canonicalization model that suffer from their own inconsistencies as reported in their paper [22]. This results in an error propagation as the input to the shape completion networks are not always in the exact canonical forms. The errors in the input map to a larger error in the output of the networks. This is followed by an error in the transform from the canonical form to the original pose. The resulting output of the multi-stage pipeline suffer from high inconsitensies and sub-optimal outputs. Unlike these networks, our model is trained jointly on both tasks (canonicalization and shape-completion) using a multi-tasking objective. As we show in the ablations, this objective plays a crucial role in achieving a disentangled representation of shape and pose. Qualitative results are shown in Fig. 3 that vividly show the closeness of SCARP's output to the ground truth when compared with others.

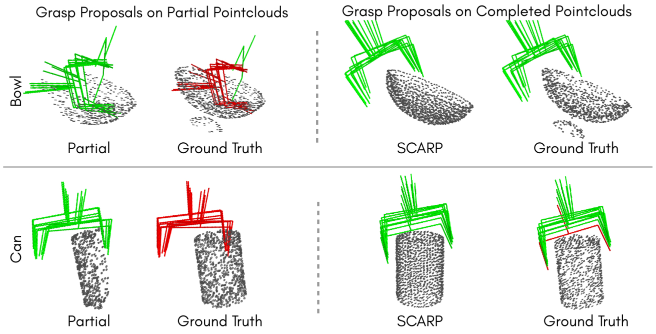

Fig. 4: **(left)**: Grasp proposals made by a SOTA grasp proposal network, [13], on partial observations lead to collisions with the actual object. *Partial* is a partial observation and *Ground truth* denotes the actual object. The proposals are made on *Partial* (shown in green) but collide with the actual object (shown in red). **(right)**: We use SCARP to complete the partial observations. Grasp proposals made on the completed objects align well with the actual object on the table reducing such collisions by a large margin.

Improvement in Grasp Proposals: Generating grasp proposals for partial pointclouds is a challenging task as a network may mistake a missing portion of an object as a potential area to grasp (see Fig. 1 and Fig. 4). We apply

		w/o SC	w/o $F_{\mathcal{X}}$	w/o p
Plane	Ours	**3.8**	**6.9**	**6.9**
	Modified	112.34	45.3	8.3
Bowl	Ours	**14.9**	**21.8**	**21.8**
	Modified	123.7	77.3	24.3
Mug	Ours	**10.79**	**12.1**	**12.1**
	Modified	47.6	45.5	20.1

TABLE II: Ablation Study: We show the affect of removing different components of our network. In w/o SC, we train SCARP as an auto-encoding network to verify if the model can still learns to canonicalize the input pointcloud. In w/o $F_{\mathcal{X}}$ and w/o p, we remove TFN and pointnet features, respectively, and evaluate the quality of shape completion. Each component plays a crucial role in achieving SCARP as is evident by the drop in metrics. Metrics are scaled by 10^3.

	Partial	SCARP	Ground Truth
Bowl	62.18	21.14	16.86
Bottle	46.5	7.35	6.32
Can	81.33	22.33	16.0
Mug	71.33	25.0	23.5
Basket	77.33	21.5	13.66
Average	67.73	19.46	15.27

TABLE III: Quantitative metrics on GE (explained in Sec. V): % of grasp proposals that are invalid or collide with the actual object on the table when the proposals are made on (1) Partial Observations, (2) Shape Completed objects by SCARP, and (3) Actual Objects on the table (Ground Truth). SCARP reduces invalid and colliding grasp proposals by 71.2% when only partial observations are available by accurately completing the object in the observed pose.

SCARP to complete these partial observations directly in the observed poses and predict grasp poses on these completed pointclouds using a SOTA grasp generation network Contact-Graspnet [13]. To evaluate the grasp proposals, we compute GE on (1) partial observations, (2) completed observations by SCARP, and (3) actual objects (ground truth). Actual objects are full pointclouds with no missing portion. As we show in Table III, SCARP shows a relative improvement of 71.2% and an absolute improvement of 48.27% over the grasp proposals on the partial pointclouds. Moreover, there is only an absolute degradation of 4.19% vis-a-vis the ground truth. The ground truth error in Table III is the datum error in the grasp proposals output by Contact-Graspnet. Qualitative results are shown in Fig. 4. Green and red proposals denote valid and colliding grasp proposals respectively. As can be seen, the grasps proposed on partial observations collide with the actual object (ground truth), whereas, the grasp proposals made on the completed object by SCARP are valid.

Ablation: SCARP is trained on a multi-tasking objective to achieve: (1) canonicalization, (2) 6D pose estimation, and (3) shape completion. We evaluate the contribution of the different components in our network in achieving these tasks.

(A) Canonicalization without Shape Completion: Canonicalization involves mapping an input X to its category's fixed canonical frame [21], [22], [25]. Learning a canonical frame for a partial input is challenging as a network may struggle to understand the overall structure of the partial shape. In our model, the structure of a category is correctly learned using the shape completion task. Thus, we analyze if SCARP can canonicalize the partial inputs without performing shape completion. To evaluate this, we modify SCARP by training to auto-encode the partial input $\hat{X}_p$ while simultaneously estimating its pose $\{R, T\}$. That is, our generator G generates X_p which is $\hat{X}_p$ in its canonical form and uses R' to rotate X_p back to $\hat{X}_p$. To measure the performance, we compute the CD between $G's$ canonical output and the canonical ground truth. In case of SCARP, this is given as $d_{CD}(X', X)$, and in case of ablation, this is given as $d_{CD}(X'_p, X_p)$. As shown in Table. II (w/o SC), on average d_{CD} on SCARP is 9.83 whereas when the shape completion aspect is removed, the average distance is 94.54. This indicates that the network does not learn anything meaningful if the task of shape completion is removed from the formulation.

(B) Shape Completion without pose and shape features: As shown in Fig. 2, G expects a disentangled feature embedding that is a non-linear combination of the pose $F_{\mathcal{X}}$ and shape embeddings p. We remove these features one by one and observe their impact on shape completion. We measure the performance of shape completion as $d_{CD}(\hat{X}, R'(X'))$. As shown in Table II (w/o $F_{\mathcal{X}}$ and w/o p), SCARP performs worse without either of the features. In both cases, the model learns a suboptimal transformation between the canonical and the original pose. Without pointnet (w/o p), the model does converge to predict the correct pose but misses out on per-instance shape details. Without TFN (w/o $F_{\mathcal{X}}$), completion in the canonical frame is more accurate (due to the shape features of pointnet), but fails to estimate the correct pose transform. In summary, as the shape completion in the canonical form suffers, the pose transform is also inaccurate thus indicating the importance of the multi-task objective.

VI. Conclusion

Existing shape completion works assume the partial inputs to be in a fixed canonical frame. This is difficult to achieve in a robotics setting where the objects are observed in arbitrary poses thus needing pre-canonicalization. This leads to an error propagation resulting in a sub-optimal shape completion. We propose SCARP, a novel architecture that performs Shape Completion in ARbitrary Poses. SCARP is trained using a multi-task objective to perform (1) canonicalization, (2) 6D pose estimation, and (3) shape completion. SCARP outperforms the existing multi-stage baselines by 45% and showcases its potential in improving grasp proposals on tabletop objects, reducing colliding grasps by more than 70%. SCARP has a huge potential in many more robotics applications like collision avoidance in trajectory planning or differential simulators for model-based RL planners.

Acknowledgement: We would like to thank Karthik Desingh, who is an Assistant Professor at the University of Minnesota, and Adrien Poulenard, who is a Postdoctoral Fellow at the Stanford University, for their valuable feedback.

References

[1] X. Yan, L. Lin, N. J. Mitra, D. Lischinski, D. Cohen-Or, and H. Huang, "Shapeformer: Transformer-based shape completion via sparse representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6239–6249.

[2] L. Pan, T. Wu, Z. Cai, Z. Liu, X. Yu, Y. Rao, J. Lu, J. Zhou, M. Xu, X. Luo *et al.*, "Multi-view partial (mvp) point cloud challenge 2021 on completion and registration: Methods and results," *arXiv preprint arXiv:2112.12053*, 2021.

[3] P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani, "Autosdf: Shape priors for 3d completion, reconstruction and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 306–315.

[4] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[5] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein, "Metasdf: Meta-learning signed distance functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 136–10 147, 2020.

[6] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," 2022.

[7] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 498–12 507.

[8] A. Avetisyan, A. Dai, and M. Nießner, "End-to-end cad model retrieval and 9dof alignment in 3d scans," in *Proceedings of the IEEE/CVF International Conference on computer vision*, 2019, pp. 2551–2560.

[9] G. Kumar, H. Pandya, A. Gaud, and K. M. Krishna, "Pose induction for visual servoing to a novel object instance." IEEE Press, 2017, p. 2953–2959. [Online]. Available: https://doi.org/10.1109/IROS.2017.8206130

[10] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.

[11] A. Agarwal, B. Sen, S. Narayanan, V. R. Mandadi, B. Bhowmick, and K. M. Krishna, "Approaches and challenges in robotic perception for table-top rearrangement and planning," 2022. [Online]. Available: https://arxiv.org/abs/2205.04090

[12] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910.

[13] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[14] H. Pandya, A. Gaud, G. Kumar, and M. Krishna, "Instance invariant visual servoing framework for part-aware autonomous vehicle inspection using mavs: Pandya et al." *Journal of Field Robotics*, vol. 36, 01 2019.

[15] M. Campbell, M. Egerstedt, J. How, and R. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 368, pp. 4649–72, 10 2010.

[16] W. Zeng, S. Wang, R. Liao, Y. Chen, B. Yang, and R. Urtasun, "Dsdnet: Deep structured self-driving network," in *European conference on computer vision*. Springer, 2020, pp. 156–172.

[17] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, and P. Mahoney, "Stanley: The robot that won the darpa grand challenge." *J. Field Robotics*, vol. 23, pp. 661–692, 01 2006.

[18] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 790–799.

[19] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[20] J. Zhang, X. Chen, Z. Cai, L. Pan, H. Zhao, S. Yi, C. K. Yeo, B. Dai, and C. C. Loy, "Unsupervised 3d shape completion through gan inversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1768–1777.

[21] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[22] R. Sajnani, A. Poulenard, J. Jain, R. Dua, L. J. Guibas, and S. Sridhar, "Condor: Self-supervised canonicalization of 3d pose for partial shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 969–16 979.

[23] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[24] W. Sun, A. Tagliasacchi, B. Deng, S. Sabour, S. Yazdani, G. E. Hinton, and K. M. Yi, "Canonical capsules: Self-supervised capsules in canonical pose," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 993–25 005, 2021.

[25] R. Spezialetti, F. Stella, M. Marcon, L. Silva, S. Salti, and L. Di Stefano, "Learning to orient surfaces by self-supervised spherical cnns," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5381–5392, 2020.

[26] A. Poulenard and L. J. Guibas, "A functional approach to rotation equivariant non-linearities for tensor field networks," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 169–13 178.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[28] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *Robotics: Science and Systems*, 2017.

[29] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta, "Same object, different grasps: Data and semantic knowledge for task-oriented grasping," *Conference on Robot Learning*, 2020.

[30] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: Gridding residual network for dense point cloud completion," in *European Conference on Computer Vision*. Springer, 2020, pp. 365–381.

[31] X. Wang, M. H. Ang, and G. H. Lee, "Point cloud completion by learning shape priors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 719–10 726.

[32] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Softpoolnet: Shape descriptor for point cloud completion and classification," in *European Conference on Computer Vision*. Springer, 2020, pp. 70–85.

[33] X. Wang, M. H. Ang, and G. Lee, "Cascaded refinement network for point cloud completion with self-supervision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[34] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," *arXiv preprint arXiv:1611.08974*, 2016.

[35] D. Stutz and A. Geiger, "Learning 3d shape completion from laser scan data with weak supervision," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1955–1964.

[36] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 85–93.

[37] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6545–6554.

[38] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[39] D. W. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3859–3868.

[40] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "Gan inversion: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[41] J. Gu, W.-C. Ma, S. Manivasagam, W. Zeng, Z. Wang, Y. Xiong, H. Su, and R. Urtasun, "Weakly-supervised 3d shape completion in the wild," in *European Conference on Computer Vision*. Springer, 2020, pp. 283–299.

[42] J. Shi, L. Xu, P. Li, X. Chen, and S. Shen, "Temporal point cloud completion with pose disturbance," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4165–4172, 2022.

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

[44] L. Lang and M. Weiler, "A wigner-eckart theorem for group equivariant convolution kernels," in *ICLR*, 2021. [Online]. Available: https://openreview.net/forum?id=ajOrOhQOsYx

[45] G. Chirikjian and A. Kyatkin, "Engineering applications of noncommutative harmonic analysis: With emphasis on rotation and motion groups (1st ed.)." in *CRC Press*, 2000. [Online]. Available: https://doi.org/10.1201/9781420041767

[46] A. W. Knapp, *Representation Theory of Semisimple Groups*. Princeton: Princeton University Press, 2016. [Online]. Available: https://doi.org/10.1515/9781400883974

[47] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, "3d steerable cnns: Learning rotationally equivariant features in volumetric data," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[48] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, 2011, pp. 1297–1304.

[49] B. Anderson, T. S. Hy, and R. Kondor, "Cormorant: Covariant molecular neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[50] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *International conference on machine learning*. PMLR, 2018, pp. 40–49.

[51] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," *Advances in Neural Information Processing Systems*, 2021.