Kit-Net: Self-Supervised Learning to Kit Novel 3D Objects into Novel 3D Cavities

Shivin Devgon[1], Jeffrey Ichnowski[1], Michael Danielczuk[1], Daniel S. Brown[1], Ashwin Balakrishna[1], Shirin Joshi[2], Eduardo M. C. Rocha[2], Eugen Solowjow[2], Ken Goldberg[1]

***Abstract*— In industrial part kitting, 3D objects are inserted into cavities for transportation or subsequent assembly. Kitting is a critical step in industrial transportation and assembly, as kitting can decrease downstream processing and handling times and enable lower storage and shipping costs. We present Kit-Net, a framework for kitting previously unseen 3D objects into cavities given depth images of both the target cavity and an object held by a gripper in an unknown initial orientation. Kit-Net uses self-supervised deep learning and data-augmentation to train a CNN to robustly estimate 3D rotations between objects and matching concave or convex cavities using a large dataset of simulated depth images pairs. Kit-Net then uses the trained CNN to implement a controller to orient and position novel objects for insertion into novel prismatic and conformal 3D cavities. Experiments in simulation suggest that Kit-Net can orient objects to have a 98.9 % average intersection volume between the object mesh and that of the target cavity. Physical experiments with 3 industrial objects suggest that Kit-Net can successfully insert objects into cavities with a 63 % success rate while a baseline which restricts itself to 2D rotations succeeds only 18 % of the time.**

I. Introduction

Kitting is a critical aspect of industrial automation that involves organizing and placing 3D parts into complementary cavities. This process saves time on the manufacturing line and frees up space to reduce shipping and storage cost. Automating kitting requires picking and rotating a part to a desired position and orientation, then inserting it into a cavity that loosely conforms to the object geometry. However, this process is a great challenge, and in industry, most kitting is performed manually.

Given a 3D CAD model of the object to be inserted and the desired object pose, one approach is to directly estimate the object pose and the transformation to the desired pose [2, 22]. However, CAD models may not be available for all objects to be kitted and are time consuming to create for every object, motivating an algorithm that can kit previously unseen objects without requiring such models. Prior work has considered kitting objects without models, but has focused on $SE(2)$ transforms (1D rotation and 2D translation) for extruded 2D polygonal objects [24, 25].

We formalize the problem of rotating and translating a novel 3D object to insert it into a novel kitting cavity and present Kit-Net, a framework for inserting previously unseen 3D objects with unknown geometry into a novel target cavity given depth images of the object in its current orientation and a depth image of either a flipped (convex) or standard (concave) target cavity. Kit-Net extends prior work from Devgon *et al.* [7], which used simulation and self-supervision to train a deep neural network to directly estimate 3D transformations between the two depth images. Given the trained deep neural network, a depth image of a previously unseen insertion cavity, and a depth image of a previously unseen object, Kit-Net iteratively estimates the $SE(3)$ transform to reorient and insert the object, without requiring detailed knowledge of its geometry. Kit-Net improves on prior work by (a) introducing dataset augmentations that make the controller more robust, (b) using a suction cup gripper to minimize object occlusion during rotation, (c) incorporating 3D translations, and (d) applying the resulting

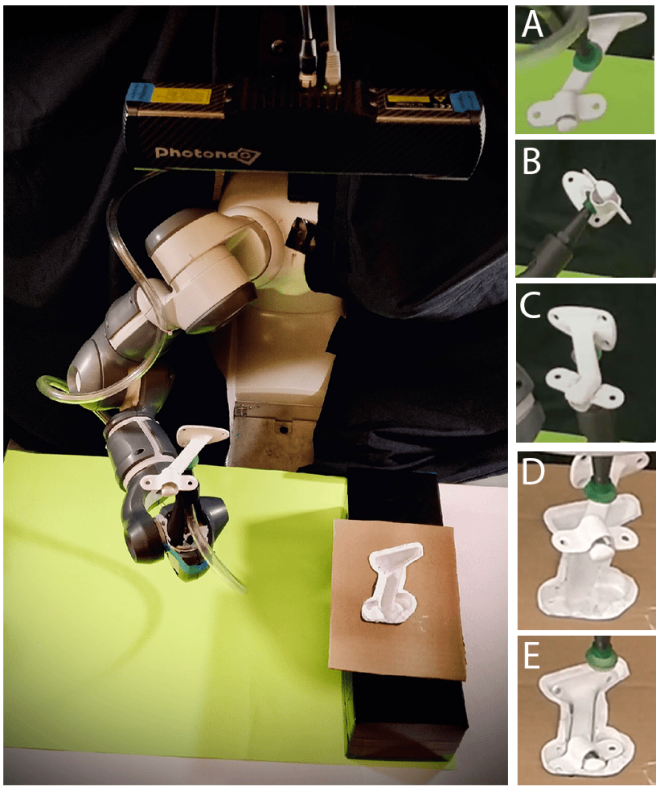

Fig. 1: Physical experiments using the ABB YuMi and a Photoneo depth camera. (Left) A suction gripper holds the handrail bracket, an object unseen during training time, near the kitting cavity. Kit-Net orients the handrail bracket for insertion into the cavity through 5 steps. A) Starting state. B) Rotate the object by 180° to face the 3D camera and minimize occlusion from the gripper. C) Iteratively orient the object into a goal configuration. D) Rotate by 180 degrees and align centroids of the object and cavity to prepare for insertion. E) Insert and release.

[1]The AUTOLAB at the University of California, Berkeley. [2]Siemens Research Lab, Berkeley, CA. {shivin302, jeffi, mdanielczuk, dsbrown, ashwin_balakrishna}@berkeley.edu, {shirin.joshi, eduardo.moura_cirilo_rocha, eugen.solowjow}@siemens.com, goldberg@berkeley.edu

controller to kit novel objects into novel cavities on a physical robot. We evaluate Kit-Net both in simulation and in physical experiments on an ABB YuMi robot with a suction gripper and overhead depth camera. Experiments in simulation suggest that Kit-Net can orient objects to have a 98.9,% average intersection volume between the object mesh and that of the target cavity. Physical experiments with 3 industrial objects and cavities suggest that Kit-Net can kit objects at a 63 % success rate from a diverse set of initial orientations.

This paper makes the following contributions:

1) Formulating the problem of iteratively kitting a novel 3D object into a novel 3D cavity.
2) Kit-Net: a self-supervised deep-learning framework for this problem
3) Simulation experiments suggesting that Kit-Net can reliably orient novel objects for insertion into prismatic cavities.
4) Physical experiments suggesting that Kit-Net can significantly increase the success rate of 3D kitting into conformal 3D cavities from 18 % to 63 % over a baseline inspired by Form2Fit [24] which only considers 2D transformations when kitting.

II. Related Work

There has been significant prior work on reorienting objects using geometric algorithms. Goldberg [9] proposes a geometric algorithm that orients polygonal parts with known geometry without requiring sensors. Akella *et al.* [1] extend the work of Goldberg with sensor-based and sensor-less algorithms for orienting objects with known geometry and shape variation. Kumbla *et al.* [11] propose a method for estimating object pose via computer vision and then reorient the object using active probing. Leveroni *et al.* [12] optimize robot finger motions to reorient a known convex object while maintaining grasp stability. In contrast to the above works, which require prior knowledge of object geometry, Kit-Net can reorient objects without 3D object models.

Melekhov *et al.* [16], Suwajanakorn *et al.* [19], Wen *et al.* [21], and Devgon *et al.* [7] use data-driven approaches to estimate the relative pose difference between images of an object in different configurations. Melekhov *et al.* [16] use a Siamese network to estimate the relative pose between two cameras given an RGB image from each camera. Suwajanakorn *et al.* [19] propose KeypointNet, a deep learning approach that learns 3D keypoints by estimating the relative pose between two different RGB images of an object of unknown geometry, but known category. Wen *et al.* [21] considers an object tracking task by estimating a change in pose between an RGBD image of the object at the current timestep and a rendering of the object at the previous timestep, but require a known 3D object model. We use the network architecture from Wen *et al.* [21] to train Kit-Net, and extend the self-supervised training method and controller from Devgon *et al.* [7] to kit novel objects into previously unseen cavities. We find that by extending Devgon *et al.* [7] to be more robust to object translations and using a suction gripper to reduce occlusions, Kit-Net is able to learn more accurate reorientation controllers.

There has also been significant interest in leveraging ideas in pose estimation for core tasks in industrial automation. Litvak *et al.* [13] leverage CAD models and assemble gear like mechanisms using depth images taken from a camera on a robotic arm's end effector. Stevic *et al.* [18] estimate a goal object's pose to perform a shape assembly task involving inserting objects which conform to a specific shape template into a prismatic cavity. Zachares *et al.* [23] combines vision and tactile sensorimotor traces for an object fitting task involving known holes and object types. Huang *et al.* [10] consider the problem of assembling a 3D shape composed of several different parts. This method assumes known part geometry and develops an algorithm to generate the 6-DOF poses that will rearrange the parts to assemble the desired 3D shape. In contrast to the above work, we focus on the problem of designing a controller which can reorient and place a novel object within a previously unseen cavity for industrial kitting tasks.

Object kitting has also seen recent interest from the robotics community. Zakka *et al.* [24] introduce Form2Fit, an algorithm which learns $SE(2)$ transforms to perform pick-and-place for kitting planar objects. In contrast, we consider 6DOF transforms of 3D objects. Zeng *et al.* [25] propose a network for selecting suction grasps and grasp-conditioned placement, which can generalize to multiple robotic manipulation tasks, including pick-and-place for novel flat objects. Zeng *et al.* focuses on $SE(2)$ rotations and translations for pick-and-place tasks involving novel flat, 2D extruded objects. Zeng *et al.* also presents an algorithm for $SE(3)$ pick-and-place tasks, but only evaluate the algorithm on 2D extruded objects. In contrast, we use Kit-Net to kit novel 3D objects with a wide range of complex geometries.

III. Problem Statement

Let $T^s \in SE(3)$ be the initial 6D pose of a unknown 3D rigid object O in the world coordinate frame, consisting of a rotation $R^s \in SO(3)$ and a translation $t^s \in \mathbb{R}^3$. Given O with starting pose T^s and a kitting cavity K, let $\mathcal{G} \subset SE(3)$ be the set of goal 6-DOF poses of object O that result in successful kitting. The goal is to orient object O to $T^g \in \mathcal{G}$, where T^g consists of rotation R^g and translation t^g. Figure 2 shows a simulated example where a 3D object O is successfully kitted into a concave cavity K.

A. Assumptions

We assume access to depth images of a rigid object O and a kitting cavity K. The cavity image may be taken with the cavity either in its standard, concave orientation (i.e., open to object insertion), or flipped, convex orientation (i.e., mirroring the shape of the object to be inserted). We also assume that orienting O to a pose in $T^g \in G$ and releasing the gripper results in a successful kitting action.

B. Input

Let $I^s \in \mathbb{R}^{H \times W}$ be a depth image observation of the object in initial pose T^s, and $I^k \in \mathbb{R}^{H \times W}$ be the depth image

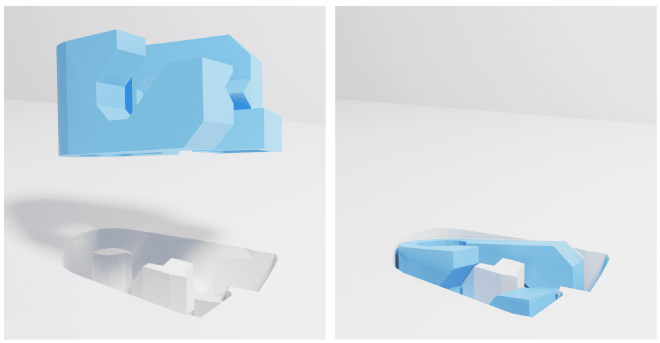

Fig. 2: **Successful Kitting:** Visualization of successfully kitting a 3D object into a concave cavity.

observation of a kitting cavity, K. See Figure 8 for physical examples of objects and kitting cavities.

C. Output

The goal is to successfully kit an unknown 3D object O into a novel 3D cavity K (Fig. 1, Fig. 2). Thus, we aim to transform the initial pose T^s into a goal pose that fits into the cavity (i.e., $T^g \in \mathcal{G}$). For objects with symmetries, the objective is to estimate and orient objects relative to a (symmetric) orientation that results in successful insertion into the cavity K.

IV. Kit-Net Framework

We present Kit-Net, a framework that first reorients the object into a pose that can be successful kit in a desired cavity, and then translates and inserts the object into the cavity. We do this by learning to estimate ${}_sT^g \in SO(3)$, a relative transformation consisting of rotation ${}_sR^g$ and translation ${}_s\hat{t}^g$, which transforms the object from T^s to T^g for some $T^g \in \mathcal{G}$. The overall approach is to (1) compute an estimate of ${}_sR^g$, denoted ${}_s\hat{R}^g$, given only image observations I^s and I^k, (2) iteratively reorient the object according to ${}_s\hat{R}^g$ until $\hat{R}^g \in \mathcal{G}$, and (3) translate the object by ${}_s\hat{t}^g$, an estimate of the translation between the start and goal object translations, such that it lies in the kitting cavity.

We first discuss preliminaries (Section IV-A) and then describe the key new ideas in training Kit-Net (Section IV-B) which make it possible to design a controller to rotate and translate an object to fit it in a cavity (Section IV-D).

A. Preliminaries: Estimating Quaternion Rotations in 3D

Devgon *et al.* [7] presented a self-supervised deep-learning method to align two 3D objects. The method takes two depth images as input: I^s, an image of the object in its current orientation R^s and I^g, an image of the same object in its desired goal orientation R^g. It trains a deep neural network $f_\theta : \mathbb{R}^{H\times W}, \mathbb{R}^{H\times W} \rightarrow SO(3)$ to estimate the rotation (parametrized by a quaternion) between the pair of images (I^s, I^g). Then, using a proportional controller, it iteratively rotates the object to minimize the estimated rotational difference. This controller applies $\eta\,{}_s\hat{R}^g$ until the network predicts that the current object rotation R^s is within $\delta = 0.5^\circ$ of R^g, or until the controller reaches an iteration limit. The tunable constant η is the Spherical-Linear intERPolation (slerp) factor describing the proportion of $\hat{R}^g$ that the controller will apply to O. Devgon *et al.* use a small slerp value of $\eta = 0.2$ and a maximum iteration limit of 50 rotations.

For training, Devgon *et al.* generate a dataset consisting 200 pairs of synthetic depth images for each of the 698 training objects with random relative rotations, for a total of 139,600 pairs. To account for parallax effects, each pair of images were generated from a fixed translation relative to the camera. Devgon *et al.* propose three loss functions to train f_θ: a cosine loss, a symmetry-resilient loss, and a hybrid of the two, with the hybrid loss outperforming the first two. Note that Devgon *et al.* do not consider cavity insertion tasks, which is complicated by the need to reason about the translations and required alignment with a cavity.

Kit-Net improves on Devgon *et al.* by (1) introducing dataset augmentations to make the controller more robust, (2) using a suction cup gripper to minimize object occlusion during rotation, and (3) incorporating 3D translations into the controller to enable kitting. We discuss these contributions in the following sections.

B. Kit-Net Dataset Generation, Augmentation, and Training

Kit-Net trains a neural network f_θ with a self-supervised objective by taking as input pairs of depth images (I^s, I^g) and estimating ${}_s\hat{R}^g$ from image pair (I^s, I^g). As in Devgon *et al.*, f_θ encodes each depth image into a length 1024 embedding, concatenates the embeddings, and passes the result through two fully connected layers to estimate a quaternion representation of the rotational difference between the object poses.

1) Initial Dataset Generation: In contrast to Devgon *et al.*, we are interested in kitting, rather than just reorienting an object in the robot gripper. Thus, in this paper we focus on two types of kitting cavities: prismatic cavities (Fig. 3) and conformal cavities (Fig. 8). We generate a separate dataset for each type of cavity and train a separate network for each dataset. To generate both datasets, we use the set of 698 meshes from Mahler *et al.* [15]. For each mesh, we generate 512 depth image pairs, for a total of 357,376 pairs. To do this we first generate a pair of rotations (R^s, R^g), where R^s is generated by applying one rotation sampled uniformly at random from $SO(3)$ to O, and R^g is generated by applying a random rotation with rotation angle less than 30 degrees onto R^s. To generate the conformal cavity dataset, we then obtain a pair of depth images (I^s, I^g) by rendering the object in rotations R^s and R^g from an overhead view. The pair is labeled with the ground truth rotation difference between the images. To generate the prismatic cavity dataset we follow the same process as above, except we fit and render a prismatic box around the rotated object (Fig. 5) and render the depth image pairs (Fig. 3). This process results in two datasets, each containing 357,376 total labeled image pairs (I^s, I^g) with ground truth rotation labels ${}_sR^g$.

2) Data Augmentation: We found that simply training a network directly on the datasets described above results in poor generalization to depth images from the physical system, which contain sensor noise, object occlusions from

both the object itself and the arm or gripper, and 3D object translations within the image. To address these three points, we introduce a set of dataset augmentations to ease network transfer from simulated to real depth images. To simulate noise and occlusion in training, we randomly zero out 1 % of the pixels in each depth image and add rectangular cuts of width 30 % of zero pixels to the image, respectively. To simulate translations in training, we translate the object across a range of 10 cm in the x, y, and z axes with respect to the camera in the simulated images. We also crop the images at sizes from 5 % to 25 % greater than the object size with center points offset from the object's centroid by 5 pixels to simulate I^s, I^g pairs generated from objects and cavities outside the direct overhead view for the kitting task.

3) Training: We adopt the network architecture from Wen *et al.* [21], as it is designed to be trained in simulation and demonstrates state-of-the-art performance on object tracking [6] by regressing the relative pose between two images. We use the hybrid quaternion loss proposed by Devgon *et al.* [7]. The network is trained with the Adam optimizer with learning rate 0.002, decaying by a factor of 0.9 every 5 epochs with an L2 regularization penalty of 10^{-9}.

C. Kit-Net Suction Gripper

Kit-Net uses an industrial unicontact suction gripper from Mahler *et al.* [14] to grasp the object for kitting. In contrast, Devgon *et al.* used a parallel jaw gripper. We find the suction gripper to be better suited for kitting because it reduces gripper occlusions and enables the robot to position the object directly inside the kitting cavity.

D. Kit-Net Controller

The Kit-Net controller consists of two stages: rotation and translation.

1) Rotation: Kit-Net first re-orients an object using the depth image of the current object pose I^s and a depth image of the goal cavity pose I^g. In preliminary experiments, we found the rotation parameters used by Devgon *et al.* [7] (Section IV-A) to be overly conservative. Thus, to speed up the alignment process, we use a larger slerp factor of $\eta = 0.8$. If the network predicts a rotation difference of less than $\delta = 5^\circ$, then we assume that the object is close enough to the required pose for kitting into the cavity and terminate the rotation controller. Because of the larger slerp value, Kit-Net is able to quickly reorient the object, thus we terminate the rotation controller after a maximum of 8 sequential rotations (8 iterations).

2) Translation: Once the rotational alignment is computed, Kit-Net computes a 2D translation to move the object directly over the target cavity, and then lowers the object and releases it into the cavity. To calculate the 2D translation, we perform centroid matching between the final depth image I^s of the object after rotation and the depth image of the cavity I^g.

V. Simulation Experiments

We first discuss metrics to evaluate performance in Section V-B. We then introduce a baseline algorithm (Section V-A) with which to compare Kit-Net and present experimental results in Section V-C. In experiments, we first evaluate Kit-Net on re-orienting novel objects with unknown geometry into a target prismatic box in simulation (Section V-C.1).

A. Baselines

1) Random Baseline: We also compare Kit-Net with a baseline that applies a randomly sampled rotation but with the correct rotation angle to evaluate how important precise reorientation is for successful kitting.

2) 2D Baseline: To evaluate the importance of estimating 3D rotations for successful kitting, we compare Kit-Net to a baseline inspired by Form2Fit [24], which only considers 2D rotations when orienting objects for kitting. The baseline (1) aligns the centroids of the point clouds of the object and the cavity and (2) searches over all possible z-axis rotations at a 1° discretization to find the rotation that minimizes Chamfer distance between the centroid-aligned point clouds.

B. Metrics

1) Object Eccentricity: We categorize test objects by their *eccentricity*, which provides a measure of kitting difficulty. This categorization follows the intuition that objects that are more elongated along certain dimensions than others have a smaller set of acceptable orientations in which they can be successfully kit into a cavity. Let the eccentricity ε of a 3D object be $\varepsilon = A - 1$, where A is the aspect ratio (ratio of longest side to shortest side) of the minimum volume bounding box of the object. This definition generalizes the 2D definition of eccentricity from Goldberg *et al.* [8] to 3D. Under this definition, a sphere has $\varepsilon = 0$, and if one axis is elongated by a factor p, then the resulting ellipsoid has eccentricity $p - 1$. This definition is also consistent with the intuition provided earlier, as a sphere is entirely rotationally symmetric, and thus does not require any reorientation for kitting. By contrast, the ellipsoid will require reorientation to ensure that its longer side is aligned to a region with sufficient space in the cavity. Thus, we use objects with high eccentricity in evaluating both Kit-Net and the baselines, as these objects pose the greatest challenge for kitting in practice.

C. Results

When evaluating Kit-Net in simulation, we have access to ground-truth object and cavity geometry. Thus, we evaluate kitting performance using the following percent fit metric:

$$\hat{f}(I^s, I^g, {}_s\hat{T}^g) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{(p_i \in K)}, \tag{V.1}$$

where we sample N points within the object volume at configuration ${}_s\hat{R}^g R^s$ (after the target object has been rotated for insertion) and count the proportion of sampled points that also lie within cavity. This metric can be efficiently computed using ray tracing and effectively estimates how much of the object fits inside the target mesh after the predicted rotation. In experiments, we use $N = 1000$ sampled points to evaluate $\hat{f}$. Assuming the true percent fit metric is f, a 95 % confidence

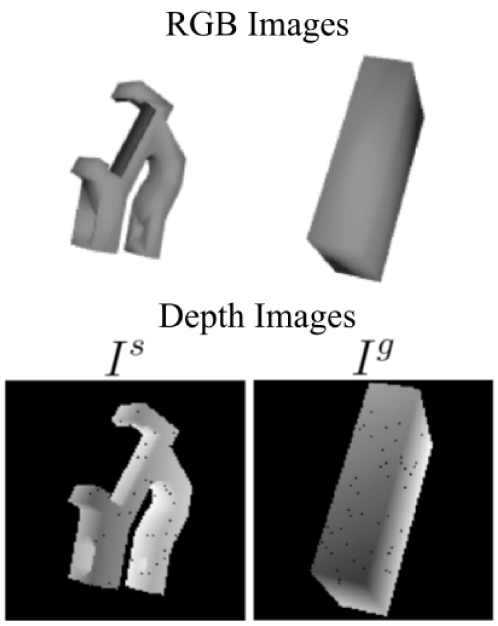

Fig. 3: **Endstop Holder and Target Prismatic Cavity in Simulation:** Given I^s, an image of an object in some configuration (top left) and I^g, an image of a target prismatic box to which the object must be aligned (top right), the objective is to find a 3D rotation ${}_s\hat{R}^g$ that would allow the object to fit within the box. In simulation experiments, R^g is a $X°$ rotation from R^s, where $X \in (0, 30)$. The image in the figure shows a 30°rotation, meaning the object must be rotated by 30°to perfectly fit it inside the prism. 3D models corresponding to I^s and I^g are shown in the bottom row for clarity.

interval for f is $\hat{f} \pm 1.96\sqrt{\frac{\hat{f}(1-\hat{f})}{1000}}$. For example, if $\hat{f} = 0.99$, then f lies between $(0.984, 0.996)$ with 95 % confidence.

1) Simulated Kitting into a Prismatic Target: We first study whether Kit-Net can orient objects into alignment with a prismatic cavity that loosely conforms to their 3D geometry in simulation. Precisely, we first generate the prismatic cavity for the target by creating a mesh with faces corresponding to its minimum volume bounding box. We then rotate both the prismatic cavity and target to random orientations within 30 degrees of each other. The objective is to apply a rotation ${}_s\hat{R}^g$ that will allow the object to fit into the cavity. An example image pair of an object and an associated prismatic cavity is shown in Figure 3.

Fig. 4 shows the percent fit across 174 unseen test objects. We use the eccentricity ε of the objects to sort them into 5 bins of increasing difficulty (increasing ε). We find that Kit-Net is able to reliably kit novel objects, significantly outperforming the 2D rotation baseline. When averaged across all eccentricities, Kit-Net achieves an average fit of 98.9 % compared to an average fit of 93.6 % for the 2D baseline and 83.1 % when applying a random 30° quaternion. These results demonstrate the need for 3D rotations to solve complex kitting problems. Figure 4 demonstrates that Kit-Net is robust to highly eccentric objects which require the most precision for kitting. Kit-Net achieves an average fit of 89.9 % for objects with eccentricity greater than 8. The 2D rotation baseline performs especially poorly for these difficult objects and achieves an average fit of only 72.7 % while applying a random 30° quaternion results in an average fit of just 37.4 %.

As described in Section IV-D, Kit-Net iteratively orients

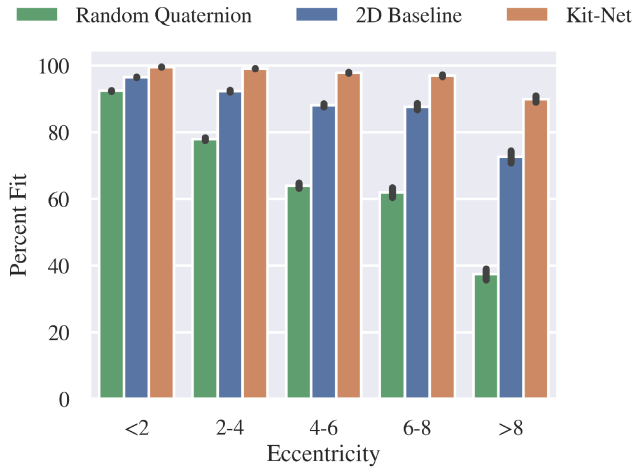

Fig. 4: **Aligning Objects to Prismatic Cavities in Simulation:** We evaluate Kit-Nets ability to align objects with prismatic cavities under the percent fit metric introduced in Section V-B across 512 depth image pairs for each of 174 objects not seen during training. Given (I^s, I^g), the network predicts ${}_s\hat{R}^g$ that will allow it to fit inside the cavity. We bin results by object eccentricity and observe that the mean percent fit decreases for objects of higher eccentricity. Kit-Net outperforms both the 2D and random baselines by a greater amount as object eccentricity increases.

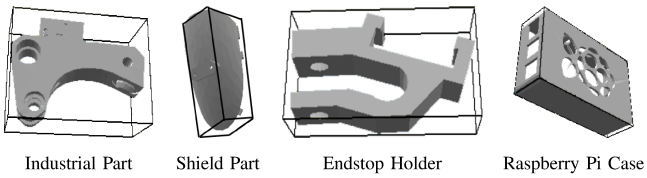

Fig. 5: **Examples of Novel Objects for Kit-Net Simulation Experiments:** The four test objects are unseen during training and have eccentricity greater than 2, meaning their minimum volume bounding boxes are narrow and long. An outline of the corresponding minimum volume bounding box is shown around each part.

each object using the controller until ${}_s\hat{R}^g \leq 5°$ or until we hit the stopping condition of 8 rotations. Our previous results in Fig. 4 suggests that Kit-Net can consistently align objects within 5 controller steps. To better visualize the ability of Kit-Net to rapidly reorient an object for kitting, we plotted the per-iteration performance of Kit-Net for 4 test objects unseen during training time with high eccentricity ($\varepsilon \geq 2$). Fig. 5 shows renderings of these objects along with outlines of the corresponding prismatic kitting cavities. Fig. 6 shows the average per-iteration percent fit across 100 controller rollouts of randomly sampled (I^s, I^g) pairs for each object. We find that Kit-Net is able to consistently align objects with their target prismatic cavities, and achieves a median fit percentage of 99.4 % after only 3 successive reorientations. By contrast, the 2D baseline is not able to surpass an average fit of 90 % for any of the objects. The results in Fig. 6 validate the importance of iteratively reorienting parts and demonstrates that applying multiple iterations of the rotation output by the trained network can greatly help to reduce the error between ${}_s\hat{R}^g R^s$ and R^g as compared to a single iteration.

VI. Physical Experiments

Our previous experiments studied the effectiveness of Kit-Net for insertion tasks involving prismatic cavities. However,

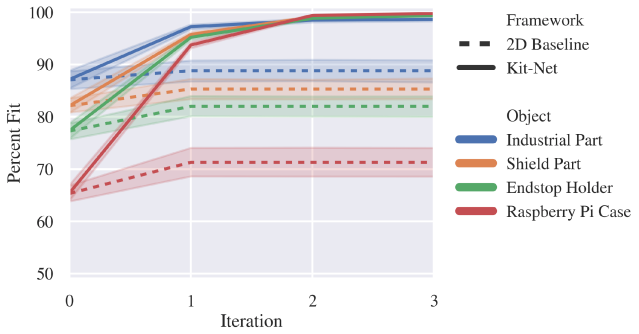

Fig. 6: **Kit-Net Simulation Results:** We visualize data from 100 runs on each of the 4 objects shown in Figure 5. All objects require a 30° rotation to be in alignment with the prismatic target at iteration 0, but their initial percent fits differ due to different eccentricities. Results suggest that Kit-Net is able to successfully align all 4 objects with their respective prismatic cavities while the baseline, which restricts itself to 2D rotations, performs significantly worse on all 4 objects.

Fig. 7: **Objects for Kit-Net Physical Experiments:** We use 3 packaged industrial objects that can be commonly found in a hardware store. These objects were selected for their complex geometries, making precise orientation critical for effective kitting.

as shown in Fig. 7 and Fig. 8, many physical kitting tasks involve non-prismatic cavities. In this section, we study how Kit-Net can be used to kit objects in physical trials using depth images of the types of cavities shown in Fig. 8. We call these *conformal cavities*, as they "conform" to some degree to the object shape.

In these experiments we use a quaternion prediction network trained to predict the quaternion that will rotate a simulated depth image of an object to another simulated depth image of the same object in a different pose. We propose two possible methods for applying this trained network to kitting. Our first method is designed to work well with the clamshell cavities shown in Fig. 8. Rather than image the hole of the cavity, we define a *convex conformal cavity* to be the depth image of the inverted cavity. To obtain these depth images, we flip the cavity so the hole is pointing down and take a depth image of the positive mass of the cavity. The left image in Fig. 8 shows examples of these convex conformal cavities. Our second method works with a *concave conformal cavity*, like that shown in Fig. 2 and the right image in Fig. 8, that are formed as impressions into a surface. These types of cavities cannot simply be flipped upside down to obtain a depth image of their shape. Instead, we take a depth image of the actual cavity (where the cavity has negative mass) and rotate it 180° about its principal axis.

We discuss the results for applying Kit-Net to novel convex conformal cavities in Section VI-A and to novel concave conformal cavities Section VI-B. For the physical kitting experiments we measure success using a binary success metric for insertion by visually inspecting whether or not the object is completely contained in the target cavity.

A. Physical Kitting into Convex Conformal Cavities

We also evaluate Kit-Net in physical kitting trials on an ABB-YuMi robot with a Photoneo depth camera, shown in Fig. 1, using 4 packaged objects widely available in hardware stores and which are unseen during training (Fig. 7). To prepare objects for kitting, we carefully extract each tool and kitting shell from its packaging and spray paint the shell to facilitate depth sensing as shown in Fig. 8. We then place the

Fig. 8: **Examples of Physical Kitting Cavities:** The handrail bracket (bottom) and the ornamental handrail bracket (top), next to the corresponding convex cavity (left) and concave cavity (right).

kitting cavity open end down and image the cavity to generate I^g before flipping it to expose its opening for the insertion task. For each trial, we insert the object into the cavity by hand, grasp it using the robot's suction gripper, translate it to be directly under the camera, and apply a random rotation of either 30° or 60°, uniformly sampled from $SO(3)$ to simulate grasping the object from a bin in a non-uniform pose. Then, we flip the object such that the object faces the overhead depth camera and the suction cup grasp is occluded from the camera by the object. This process is illustrated in Fig. 1.

Kit-Net then orients the object using the learned controller, and matches centroids between the object and cavity for insertion before flipping it again and attempting to kit it. Table I shows the number of successful kitting trials (out of 10 per object) of Kit-Net and the 2D baseline across 3 objects. We report a kitting trial as successful if the object is fully contained within the cavity from visual inspection. We observe that Kit-Net outperforms the baseline for 30° initial rotations on 2 of the 3 objects, performing similarly to the baseline on the sink handle. We find that Kit-Net significantly outperforms the baseline on all objects for 60° initial rotations.

Kit-Net's main failure modes are due to errors in the

centroid matching procedure, as illustrated in Fig. 9. On the 30 degree sink handle task, Kit-Net aligned it correctly every time, but the centroid matching had it about 0.5 cm off, and there is no slack at the top of the cavity.

Object	Angle	2D Baseline	Kit-Net
Handrail bracket	30°	3/10	**10/10**
Ornamental handrail bracket	30°	8/10	**10/10**
Sink handle	30°	**4/10**	3/10
Handrail bracket	60°	1/10	**9/10**
Ornamental handrail bracket	60°	2/10	**7/10**
Sink handle	60°	0/10	**7/10**

TABLE I: **Physical Experiments Results for Convex Cavities:** We report the number of successful kitting trials for Kit-Net and the 2D baseline over 10 trials for 3 previously unseen objects with initial rotations of 30°and 60°. Results suggest that Kit-Net significantly outperforms the 2D baseline for initial rotations of 60°and outperforms the baseline for two out of three objects for initial rotations of 30°.

B. Physical Kitting into Concave Conformal Cavities

Here, we perform the same experiment as in Section VI-A, but instead generate I^g directly from an image of the cavity without flipping. Precisely, we segment out the cavity from an overhead depth image, deproject the depth image into its point cloud representation, and rotate the point cloud 180° around its center of mass. Then, we project the rotated point cloud to the depth image I^g.

Object	Angle	2D Baseline	Kit-Net
Handrail bracket	30°	0/10	**9/10**
Ornamental handrail bracket	30°	0/10	**7/10**
Sink handle	30°	1/10	**3/10**
Handrail bracket	60°	0/10	**7/10**
Ornamental handrail bracket	60°	0/10	0/10
Sink handle	60°	1/10	**4/10**

TABLE II: **Physical Experiments Results for Concave Cavities:** We report the number of successful kitting trials for Kit-Net and the 2D baseline over 10 trials for 3 previously unseen objects with initial rotations of 30°and 60°. Results suggest that Kit-Net significantly outperforms the baseline in all settings except for the handrail bracket with an initial rotation of 60°, for which neither Kit-Net nor the baseline can successfully kit the object.

Table II shows results from experiments with 3 novel objects from Fig. 7 across 10 controller rollouts. We observe that Kit-Net outperforms the baseline for initial rotations of both 30 and 60 degrees on the handrail bracket and sink handle, and for an initial rotation of 30 °for the ornamental handrail bracket. For the ornamental handrail bracket, the depth image from the concave cavity is low quality as shown in Fig. 9 (center image), causing Kit-Net to fail when the object is 60 degrees away the correct insertion orientation. We examined this failure and found that it occurs because the cavity for the neck of the bracket is very thin, making it difficult to obtain a good depth image. Kit-Net also has low performance on the sink handle due to small errors in centroid matching, as discussed in the prior section. Fig. 9 (bottom left) shows an example failure case where the sink handle is correctly oriented but the translation is slightly off. There were also occasional cases (Fig. 9 (top left)) where

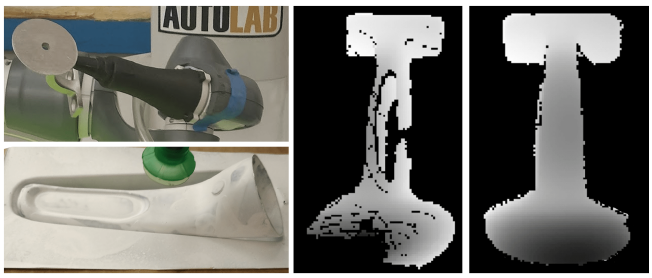

Fig. 9: **Kit-Net Failure Cases:** The top-left image shows a configuration of the ornamental handrail bracket where the suction gripper occludes the handle below the base. The bottom-left image shows the sink handle. Although Kit-Net was able to orient the handle correctly for insertion, the centroid matching had a small error in estimating translation and the cavity does not have enough slack to be properly inserted. The center and right images show depth images for the ornamental handrail bracket for the concave conformal cavity and convex conformal cavity, respectively. The inside of the concave cavity is very thin and the angle of the camera makes it hard to perfectly image it, resulting in a poor depth image (center image). This leads to 0 successes for both the baseline and for Kit-Net when the initial rotation is 60° away from the desired rotation for kitting.

the suction gripper occludes the handle of the ornamental handrail bracket. In these cases, the robot can only see the base, resulting in failure.

VII. Discussion and Future Work

We present Kit-Net, a framework that uses self-supervised deep learning in simulation to kit novel 3D objects into novel 3D cavities. Results in simulation experiments suggest that Kit-Net can kit unseen objects with unknown geometries into a prismatic target in less than 5 controller steps with a median percent fit of 99 %. In physical experiments kitting novel 3D objects into novel 3D cavities, Kit-Net is able to successfully kit novel objects 63 % of the time while a 2D baseline which only considers $SE(2)$ transforms only succeeds 18 % of the time. In future work, we will work to improve performance by using the predicted error from Kit-Net to regrasp the object in a new stable pose [3, 5, 20] before reattempting the kitting task, study Kit-Net's performance with other depth sensors, and apply Kit-Net to kit objects that are initially grasped from a heap [4, 17].

VIII. Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. Authors were also supported by the Scalable Collaborative Human-Robot Learning (SCHooL) Project, a NSF National Robotics Initiative Award 1734633, and in part by donations from Siemens, Google, Toyota Research Institute, Autodesk, Honda, Intel, Hewlett-Packard and by equipment grants from PhotoNeo and Nvidia. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1752814. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] S. Akella and M. T. Mason, "Parts orienting with shape uncertainty", in *International Conference on Robotics and Automation (ICRA)*, 1998.

[2] C. Choy, W. Dong, and V. Koltun, "Deep global registration", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2511–2520.

[3] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces", in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 1578–1585.

[4] M. Danielczuk, A. Angelova, V. Vanhoucke, and K. Goldberg, "X-ray: Mechanical search for an occluded object by minimizing support of learned occupancy distributions", *arXiv preprint arXiv:2004.09039*, 2020.

[5] M. Danielczuk, A. Balakrishna, D. S. Brown, S. Devgon, and K. Goldberg, "Exploratory grasping: Asymptotically optimal algorithms for grasping challenging polyhedral objects", in *Conference on Robot Learning (CoRL)*, 2020.

[6] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao–Blackwellized particle filter for 6-D object pose tracking", *IEEE Transactions on Robotics*, pp. 1–15, 2021.

[7] S. Devgon, J. Ichnowski, A. Balakrishna, H. Zhang, and K. Goldberg, "Orienting novel 3D objects using self-supervised learning of rotation transforms", in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1453–1460.

[8] K. Goldberg, M. Overmars, and A. F. van der Stappen, "Geometric eccentricity and the complexity of manipulation plans", *Algorithmica*, vol. 26, no. 3-4, pp. 494–514, Mar. 2000.

[9] K. Y. Goldberg, "Orienting polygonal parts without sensors", *Algorithmica*, vol. 10, no. 2-4, pp. 201–225, 1993.

[10] J. Huang, G. Zhan, Q. Fan, K. Mo, L. Shao, B. Chen, L. Guibas, and H. Dong, "Generative 3D part assembly via dynamic graph learning", *ArXiv*, vol. abs/2006.07793, 2020.

[11] N. B. Kumbla, J. A. Marvel, and S. K. Gupta, "Enabling fixtureless assemblies in human-robot collaborative workcells by reducing uncertainty in the part pose estimate", in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2018, pp. 1336–1342.

[12] S. Leveroni and K. Salisbury, "Reorienting objects with a robot hand using grasp gaits", in *Int. S. Robotics Research (ISRR)*, 1996.

[13] Y. Litvak, A. Biess, and A. Bar-Hillel, "Learning pose estimation for high-precision robotic assembly using simulated depth images", *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3521–3527, 2019.

[14] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning", *arXiv preprint arXiv:1709.06670*, 2017.

[15] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies", *Science Robotics*, vol. 4, no. 26, eaau4984, 2019.

[16] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks", in *Advanced Concepts for Intelligent Vision Systems - 18th International Conference, ACIVS 2017, Proceedings*, Springer Verlag, 2017, pp. 675–687.

[17] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 6232–6238.

[18] S. Stevic, S. Christen, and O. Hilliges, "Learning to assemble: Estimating 6D poses for robotic object-object manipulation", *IEEE Robotics and Automation Letters*, vol. 5, pp. 1159–1166, 2020.

[19] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi, "Discovery of latent 3D keypoints via end-to-end geometric reasoning", in *Proc. Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., Curran Associates, Inc., 2018, pp. 2059–2070.

[20] P. Tournassoud, T. Lozano-Pérez, and E. Mazer, "Regrasping", in *Proceedings. 1987 IEEE international conference on robotics and automation*, IEEE, vol. 4, 1987, pp. 1924–1928.

[21] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "SE(3)-TrackNet: Data-driven 6D pose tracking by calibrating image residuals in synthetic domains", *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10 367–10 373, 2020.

[22] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes", *arXiv preprint arXiv:1711.00199*, 2017.

[23] P. A. Zachares, M. Lee, W. Lian, and J. Bohg, "Interpreting contact interactions to overcome failure in robot assembly tasks", *ArXiv*, vol. abs/2101.02725, 2021.

[24] K. Zakka, A. Zeng, J. Lee, and S. Song, "Form2Fit: Learning shape priors for generalizable assembly from disassembly", *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9404–9410, 2020.

[25] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation", *Conference on Robot Learning (CoRL)*, 2020.