Stable Object Reorientation using Contact Plane Registration

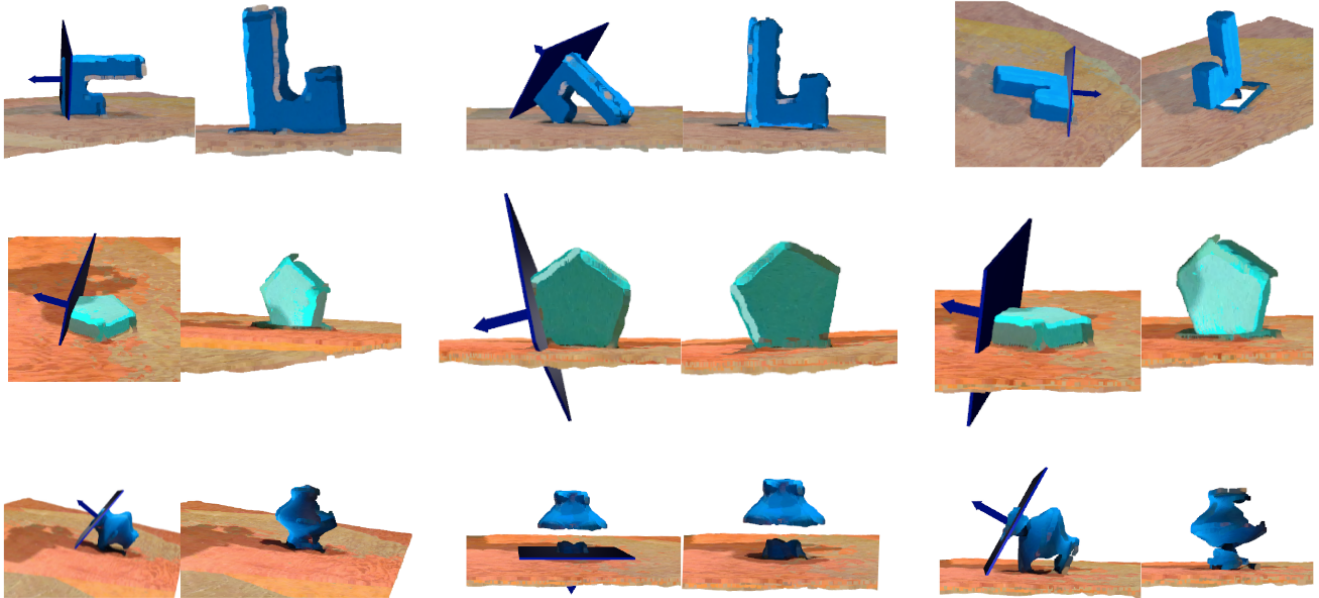Richard Li[1], Carlos Esteves[2], Ameesh Makadia[2] and Pulkit Agrawal[1]

Fig. 1: The image pairs visualize the prediction of our model in reorienting the object in its initial orientation (left) to the height-maximizing stable orientation (right). The results are shown for three previously *unseen*, *real world* test objects using point clouds captured from the RealSense depth cameras. Our method assumes that objects can be segmented from the scene. The desired transformation is obtained by predicting a contact plane (visualized in blue along with the normal vector) that needs to be aligned with the table to obtain a height-maximizing pose of that object.

Abstract— We present a system for accurately predicting stable orientations for diverse rigid objects. We propose to overcome the critical issue of modelling multimodality in the space of rotations by using a conditional generative model to accurately classify contact surfaces. Our system is capable of operating from noisy and partially-observed pointcloud observations captured by real world depth cameras. Our method substantially outperforms the current state-of-the-art systems on a simulated stacking task requiring highly accurate rotations, and demonstrates strong sim2real zero-shot transfer results across a variety of unseen objects on a real world reorientation task.

I. INTRODUCTION

Orienting objects is a fundamental operation in many robotic manipulation tasks such as pick-and-place, peg-in-hole, and tool use. Our goal is to develop a system for predicting rotations in the full $SO(3)$ space for pick-and-place from noisy sensory observations obtained using commodity depth cameras. We specifically focus on two issues central to rotation prediction: multimodality and generalization to unseen objects from a small number of training shapes.

The task of fitting a model where similar inputs correspond to very different outputs (with respect to the loss function) is known as the multimodal learning problem. Multimodality can arise from three sources in the rotation prediction task: 1) representational discontinuity, 2) object symmetry, and 3) equivalent optimal rewards.

1) Commonly used rotation representations such as Euler angles and quaternions, suffer from discontinuities in the mapping from observation to rotation representation [1], [2], [3], [4]. For example, in the 1D angle prediction task, similar observations of an object near the identity pose can have labels at 1 ° and 359 °.
2) For a symmetric object, multiple rotations of the underlying object model result in the same observation.
3) A reward function captures the functional utility of an action. If we consider the rotation predictions as actions given an observation, multiple rotations may correspond to the same optimal reward.

Prior works have addressed the multimodality problem by either implicitly modelling an unnormalized density function

[1]These authors are with the Improbable AI Lab at Massachusetts Institute of Technology, USA. Correspondence to: `rli4@mit.edu`

[2]These authors are with Google Research, New York.

[5], [6] or explicitly modelling the distribution of labels using conditional generative models [7], [8].

While multimodality and object generalization may be difficult problems to address in the general case of predicting arbitrary rotations, we consider rotations in a reduced space: namely, the rotations that align one planar contact surface to another. This assumption allows us to introduce specific algorithmic components that can address the two problems.

To predict the contact plane, we predict the probability that each point is in contact with the target surface when the object is in a correct pose. For example, in this work the correct poses are height-maximizing stable static equilibria. Then, we fit a contact plane to the contact points using random sampling and consensus (RANSAC) plane segmentation [9].

Our reformulation of rotation prediction to contact point probability prediction helps with object generalization. The reason object generalization improves is because contact point probabilities are primarily functions of the local geometry. An example of a local feature that our network might learn to determine contact point probability is the discrete Gaussian curvature computed from a point's Euclidean nearest neighbors. Specifically, points on contact planes would be distinguished with curvature 0. Given a new object, as long as the local regions around the contact surface are still similar to those seen in the training set, contact point probability prediction should succeed.

Our contact point probability formulation also helps with producing highly accurate predictions in presence of label multimodality. Our best model uses a CVAE to help with the contact point probability prediction. However, due to VAE-based models' tendency to produce "blurry" samples, the contact point probabilities will be a weighted average across different binary contact mask labels corresponding to a single input. The CVAE encourages this average to be weighted towards a single contact mask instead of being a uniform average over the contact masks, as would happen with a deterministic model predicting the contact probabilities. Under our formulation, the "averaged" contact point probabilities can still result in highly accurate predictions, as the RANSAC plane segmentation will find the contact plane with highest inlier support and ignore the outlier points corresponding to the less dense contact region(s).

We evaluate the proposed method on the object reorientation task shown in Figure 1. Given a segmented object point cloud in arbitrary configuration lying on the table, the goal is to predict the rotation that will transform the object into a height-maximizing and stable pose. We show that our method is capable of accurate sim2real transfer across a variety of novel real world objects, and it additionally outperforms state-of-the-art methods on a simulated stacking task that requires highly accurate rotation predictions.

II. METHOD

Let us assume we are provided a segmented object point-cloud, $\mathcal{X} : \{x_i \in \mathbb{R}^3\}_{i=1}^N$ with N points initialized in an arbitrary orientation on the table. Our goal is to learn a function $f : \mathcal{X} \mapsto \mathbf{R} \in \mathbf{SO}(3)$ that outputs a rotation, $\mathbf{R}$, that corresponds to placing the object in a height-maximizing and a stable orientation. For this, we first describe our formulation to predict a binary mask corresponding to the object surface that should lie on the table, when the object is in the height-maximizing orientation (see Fig. 3 and Sec. II-A).

Next, in Sec. II-B we describe a robust procedure for registering the contact plane with the placement surface. The outcome of the registration is the rotation $\mathbf{R}$. Finally, we describe neural network details to build the contact classifier in Sec. II-C, and real world and simulation results in Sec. IV. An overview of the rotation prediction pipeline is in Fig. 2.

A. Contact Plane Identification

We assume that for every pointcloud ($\mathcal{X}$) in the training set, we have access to a binary ground-truth mask, $\mathcal{Y} : Y_i \in [0, 1]$, indicating which points on the object are in the contact with the table (i.e., $Y_i = 1$) in the height-maximizing configuration. The procedure for obtaining such labels using minimal manual effort is described in Section III-B.1. Using this data we train a classifier for predicting contact points, $c : \mathcal{X} \mapsto \mathcal{Y}$, that takes object-segmented pointcloud ($N \times 3$) as input and outputs per-point probability ($N \times 1$) of a given point being a contact point. We threshold the points with predicted probability > 0.5 as the contact points. We now provide details of training this classifier.

1) Loss Function: We use binary cross entropy (BCE) loss. Because only a small fraction of pointcloud points are contact points, we up-weight the contribution of ground-truth contact points in the BCE loss as following:

$$\mathcal{L}(\hat{Y}, Y) = \sum_i -w_i[Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log(1 - \hat{Y}_i)]$$

where $\hat{Y}_i, w_i$ is the prediction and the weight of each sample. w_i is defined as:

$$w_i = Y_i \cdot r \cdot \frac{N - \sum_i Y_i}{\sum_i Y_i} + (1 - Y_i) \cdot$$

If $r = 1$, the above weighting would equalize the cardinality of positive and negative points with respect to the loss function. Changing r provides expressivity for controlling the precision and recall. We found $r = 2$ to work best.

B. Rotation Prediction via Contact Plane Registration

Given the predicted contact points, we use random sampling and consensus (RANSAC) [9] to fit a plane to these points [10]. This plane represents the predicted contact surface of the object when it is in in contact with the table in the height-maximizing pose. The choice of RANSAC ensures that plane fitting is robust to incorrect predictions by the classifier.

We find the rotation between the oriented surface normal of the contact plane and the gravity vector (which is the negative surface normal of the table plane) using Rodrigues' formula [11]. While we describe our method in the context of orienting the object to be placed on a horizontal surface, it
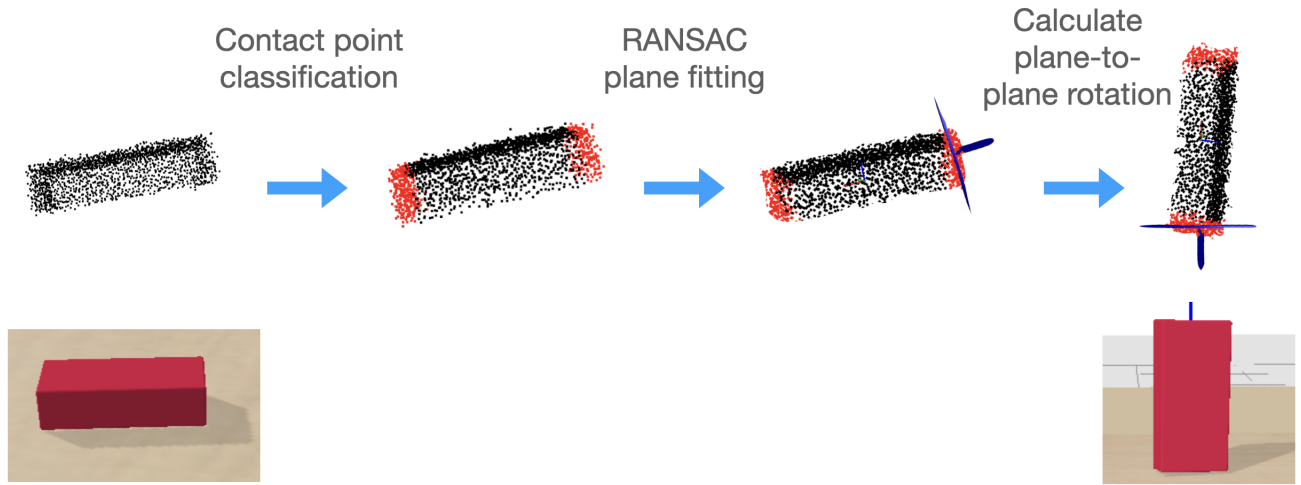
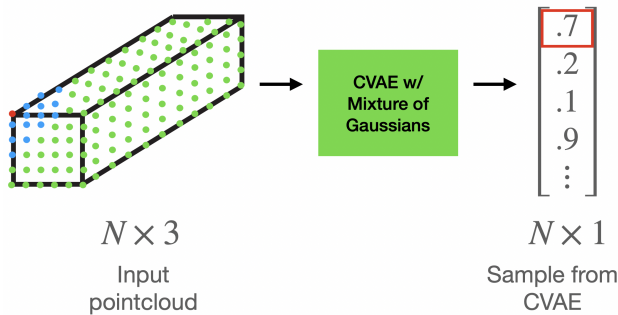Fig. 2: This figure depicts the three steps involved in generating a rotation from an input pointcloud.

Fig. 3: This figure depicts how our CVAE processes the input pointcloud. We highlight in red an input point and the associated predicted contact point probability. We highlight in blue some example points that may be processed by the DGCNN architecture in order to produce the red contact point probability prediction. The contact point probabilities are thresholded to produce a binary mask.

generalizes to placement on any oriented plane whose surface normal is known.

C. Contact Mask Classifier Details

Contact Mask Classifier Baseline (CMC) This baseline model predicts the probability that each point is a contact point, using a deterministic classification model. When the clusters of contact points are well-separated in Euclidean space, RANSAC plane segmentation can often find a planar fit through one of the clusters.

CVAE + CMC + MoG1 While the baseline model could potentially learn the correct probabilities, we found that combining the baseline model with a generative model improved the training fit and generalization of contact points. We label this model CVAE + CMC + MoG1 to denote its usage of a conditional VAE and a unimodal Gaussian prior. We found that the use of CVAE improved performance and the model more frequently generated samples with distinctive modes, in contrast to the CMC baseline.

CVAE + CMC + MoG5 It is known theoretically that more expressive priors allow VAE models to achieve lower evidence lower bound (ELBO) training objectives [12]. In our case, where the space of contact masks has discrete multimodality, it is natural to use a more expressive prior with discrete structure. We modify the CVAE + CMC model by replacing the unimodal Gaussian prior with a Mixture of Gaussians prior with five components. We found that not only did the CVAE + CMC + MoG5 model achieve a lower reconstruction error and KL regularization error, but it also improved accuracy on predicted rotations for unseen objects.

1) Data Augmentation: We apply four types of data augmentation: 1) random rotations in $SO(3)$, 2) random XY shear, 3) random XYZ scale, and 4) random noise in the pointcloud points sampled from the depth camera noise model (Section III-C). We apply these transformations in a way that ensures correctness of the labelled contact plane.

2) Neural Network Architecture: We used Dynamic Graph CNN (DGCNN) [13], a state-of-the-art pointcloud processing architecture with default hyperparameters.

D. Other Baseline Methods

1) PCA Oriented Bounding Box Baseline + Ground Truth Face Selection (OBB+GFTS): Oriented bounding boxes are commonly used as a simple pose representation that is consistent over many different object types. An oriented bounding box consists of eight points that describe the vertices of some cuboid enclosing an object's pointcloud. PCA is commonly used to compute oriented bounding boxes (OBB) for point sets in $\mathbb{R}^n$ [14]. Given an OBB, there are two choices for the height-maximizing contact face. They are the two faces along the longest axis of the OBB. We assume we have an oracle scoring function that tells us which of the two faces will lead to a stable static equilibrium (if one exists), when that face is parallel with the ground plane.

2) Direct Quaternion Prediction Baseline: We find that direct quaternion prediction, using the representation and "chordal loss" in [2] is challenging to optimize with multi-

modal rotation labels. To ensure that the challenge is caused by multimodality, we ran an ablation where we trained only on five objects in the training set that have no rotational symmetries. This training was successful indicating that multimodality in outputs induced by object symmetry impedes learning. We also experimented with the *ShapeMatchLoss* [15] that was specifically designed to handle rotationally symmetric objects. However similar to [16], we observed that ShapeMatchLoss is hard to optimize.

3) CVAE with Quaternions Baseline: We would like to understand the importance of modelling rotations with contact regions compared to an explicit representation. Thus, we consider a CVAE formulation whose decoder outputs quaternions. This baseline produces poor results because we are unable to find a good value for the KL-divegence term weight β that simultaneously minimizes the reconstruction and KL divergence loss.

4) Implicit PDF Baseline (IPDF): Implicit PDF [5] is a state-of-the-art method for learning multimodal distributions over rotation matrices. The main idea is to learn a function $f(x, R)$ that maps an observation x (pointcloud) and rotation R (rotation matrix) to the unnormalized joint density of the input random variables. Then, they acquire a conditional likelihood $p(R|x)$ training objective by approximately integrating this joint density over a discrete set of rotations.

III. EXPERIMENTAL SETUP

A. Object Datasets and Shape Generalization

We use objects from two different physical dexterity / stacking games: Bandu and Bausack. These games are played with similar objectives where players adversarially bid tokens (a finite game currency) to acquire easily stackable pieces and try to force opposing players to stack complicated, unstable pieces. The objects from these games substantially vary in shape and stacking affordances making them a good testbed for evaluating rotation prediction. We create a training, validation, and test set as follows: 15 objects from Bandu for training, 6 objects from Bandu for validation, and 20 objects from Bausack for testing. The train/val split was randomly selected given the constraint that the training set should include a significant number of multimodal objects. We train and validate on only Bandu objects and generalize to the significantly different shapes of the Bausack game.

Five out of fifteen objects in the training set and 11/20 objects in the test set have multimodality in the space of contact mask labels.

B. Training Data Generation and Labelling

We use the PyBullet simulator for our experiments [17]. The starting orientations are generated by randomly dropping objects from a height of 0.4 meters above the table, running forward simulation for 1000 timesteps until the objects have stabilized, then recording their orientation. We use the built-in PyBullet segmentation functionality to get the ground truth object-segmented pointclouds. Random dropping generates variations in starting orientations and occlusion.

A human expert labels one "target" orientation per object. This target orientation corresponds to a height-maximizing and stable pose of the object. However, even with only one target orientation per object, multimodality manifests in the label space of relative rotations due to the object symmetries. The relative rotation between the initial and target orientations is used as the ground truth rotation label.

1) Contact Mask Label Generation: Given the rotation labels available as quaternions, we follow an automated procedure for generating contact point segmentation masks. A contact mask label is a $N \times 1$ binary tensor that indicates whether each point in the $N \times 3$ pointcloud is a contact point or not. We rotate the observed pointcloud into the stacked position, choose the bottom $k\%$ of points as contact points, and designate the remaining points as non-contact points.

C. Simulating Noisy and Occluded Pointclouds

To emulate the complexity of real world pointcloud data, we generate training data and evaluate our learned model on a simulated stacking task using realistic noise and occlusion. Our pointcloud noise model is the axial noise model from [18], where researchers found that the Intel RealSense D435 depth camera noise was well-approximated by Gaussians with depth-conditional variance, which are centered at the true depth value. We use the same parameters for the noise model as their paper. Our pointclouds are collected by four depth cameras positioned around the table. The segmented pointclouds have occlusions, which are either table-object (object is partially covered by the main flat region of the table or table sides) or, less frequently, object-object (object is partially covered by another object) occlusions.

IV. RESULTS

We seek to answer the following questions:

1) How does the performance of our contact registration method compare with baselines?
2) What are the effects of various architectural choices on the performance of the proposed system?
3) Can our system produce accurate predictions for real world pointclouds?

A. Simulation Two-block Stacking Evaluation

We evaluate the performance of our method and various baselines for rotation prediction by measuring: (a) the accuracy of stacking objects into a tower; (b) and a more granular metric that we call the minimum angular error between normals (**MAEN**). Definitions of these metrics are:

Stacking Accuracy is computed using a $score$, which is initially set to 0. Given an ordered list of K objects to be stacked, we increment the $score$ by $+1$ if placing the object increases the height of the tower by the length of the object along the Z axis in the height-maximizing pose. To account for stability the score increment is only made if the object does not fall down and maintains the height-maximizing pose for 500 simulation timesteps after being placed. After all objects are placed, if $score = K$, then the stacking trial is

TABLE I: Stacking Success Rates and Minimum Angular Errors for Two Object Towers

	OBB+GTFS	IPDF	CMC	CVAE+CMC+MoG1	CVAE+CMC+MoG5 (Ours)
Barrell	0.81 (0.18±0.23)	0.97 (0.076±0.046)	0.05 (1.4±0.3)	0.9 (0.11±0.19)	0.99 (0.053±0.03)
Block	1.0 (0.051±0.027)	0.98 (0.056±0.028)	0.97 (0.043±0.022)	0.99 (0.039±0.021)	0.99 (0.038±0.022)
Bridge	0.94 (0.054±0.03)	0.9 (0.058±0.027)	0.92 (0.045±0.023)	0.95 (0.04±0.023)	0.97 (0.034±0.021)
Cone	0.76 (0.42±0.61)	1.0 (0.052±0.026)	0.98 (0.039±0.024)	1.0 (0.033±0.019)	1.0 (0.032±0.019)
Cone+Ball	0.99 (0.043±0.024)	0.97 (0.054±0.026)	0.9 (0.19±0.63)	0.89 (0.15±0.52)	0.97 (0.046±0.027)
Cross	0.67 (0.6±0.57)	0.69 (0.36±0.54)	0.98 (0.065±0.21)	0.93 (0.16±0.39)	0.94 (0.17±0.42)
Cross Round Top	0.73 (0.35±0.23)	0.74 (0.37±0.47)	0.39 (1.0±0.71)	0.65 (0.56±0.7)	0.92 (0.14±0.36)
Cylinder Long	0.96 (0.021±0.012)	0.75 (0.052±0.025)	0.33 (0.18±0.29)	0.4 (0.13±0.22)	0.38 (0.12±0.19)
Cylinder Parallelogram	0.01 (0.48±0.046)	0.86 (0.14±0.042)	0.93 (0.089±0.22)	0.96 (0.059±0.034)	1.0 (0.048±0.028)
Egg	0.77 (0.24±0.31)	0.67 (0.45±0.82)	0.21 (1.5±0.28)	0.46 (1.0±1.2)	0.69 (0.45±0.71)
Foundation	1.0 (0.051±0.03)	0.96 (0.05±0.029)	0.83 (0.06±0.038)	0.97 (0.05±0.028)	0.93 (0.052±0.032)
J Block	0.33 (0.48±0.076)	0.83 (0.25±0.39)	0.89 (0.12±0.34)	0.58 (0.66±0.91)	0.85 (0.21±0.48)
L Chamfered	0.97 (0.2±0.035)	1.0 (0.082±0.035)	0.96 (0.11±0.29)	0.96 (0.27±0.75)	1.0 (0.084±0.26)
Nut	0.32 (0.36±0.12)	0.4 (0.31±0.23)	0.81 (0.25±0.5)	0.95 (0.12±0.29)	0.81 (0.3±0.56)
Pentagram Stout	0.73 (0.16±0.089)	0.3 (0.33±0.16)	0.18 (1.2±0.54)	0.78 (0.44±0.62)	0.96 (0.3±0.54)
Pyramid w/ Eye	0.74 (0.47±0.2)	0.66 (0.55±0.26)	0.98 (0.043±0.026)	1.0 (0.037±0.023)	1.0 (0.051±0.033)
Rectangular Solid	0.69 (0.024±0.012)	0.48 (0.054±0.029)	0.37 (0.09±0.15)	0.4 (0.082±0.15)	0.34 (0.078±0.041)
Rounded Bobble	0.76 (0.21±0.27)	0.67 (0.61±1.0)	0.01 (1.6±0.2)	0.19 (1.5±1.2)	0.32 (1.0±1.0)
Vase w/ Hole	0.39 (0.86±0.56)	1.0 (0.066±0.03)	0.57 (0.52±0.49)	0.88 (0.16±0.27)	0.94 (0.087±0.17)
Wedge Long	0.3 (0.24±0.29)	0.81 (0.098±0.1)	0.92 (0.056±0.15)	0.92 (0.071±0.2)	0.93 (0.065±0.16)

considered to be a success. We measure accuracy over 100 trials.

MAEN quantifies the closeness of the predicted contact normal with the set of ground truth contact normals for an object. We label all optimal contact planes by specifying a list of oriented contact plane normals $(\vec{\eta}_{0:L})$ based on the *mesh* model (not observed pointcloud) of each object. When any of these normals is aligned with the gravity vector, the object enters into a height-maximizing and a stable pose. Let θ_l denote the rotation angle between the predicted surface normal and the $l^{th} \in [1, L]$ ground-truth normal $(\vec{\eta}_l)$. MAEN is defined as $\min_l \theta_l$.

1) Quantitative Evaluation: Our first experiment evaluates rotation prediction for the task of stacking two object towers, where in each trial the bottom object is a cuboid of fixed size and the top object is one of the test objects. We choose a cuboid as the bottom object because it has less surface area than the table and therefore stable placement of the object requires higher accuracy in rotation prediction. For each test object, accuracy evaluated over 100 trials is reported in Table I. The mean and standard deviation of the MAEN are included in the parentheses in Table I.

The results reveal that our method, CVAE+CMC+MoG5, outperforms all the baselines. Further quantification of which model performs best across all object types is presented in Fig. 4. The figure plots the fraction: $\frac{\text{\# objects whose failure rate} \leq \text{failure rate threshold}}{\text{total \# objects}}$ on the y-axis and incremental failure rate thresholds on the x-axis. [1] From this plot, we can see that for any failure threshold we select, our CVAE+CMC+MoG5 model's failure rates are below the threshold, for the highest number of objects, compared to other models. Our method stays under a 19% failure rate threshold for 16/20 objects, while the second best stays under that threshold for only 12/20 objects.

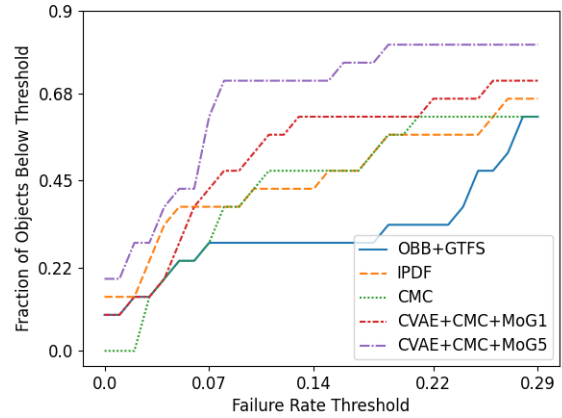[1]The object failure rate is is $1 - \texttt{success\_rate}$ (Table I).

Fig. 4: Curve showing (y-axis) fraction of objects types below a certain failure rate, associated with each model, (x-axis) for various failure rates (best viewed in color).

2) Simulation Qualitative Analysis: When the CMC method fails, it often produces large, arbitrary swathes of contact points that do not correspond to contact regions (Fig. 5). In comparison, CVAE+CMC+MoG5 produces thinner sets of contact points, which lead to more accurate contact plane fitting and higher accuracy. Both models tend to incorrectly predict multiple modes per sample for multimodal objects, and correctly predict one mode for unimodal objects. For the CVAE, this behavior of multiple mode prediction is likely caused by upweighting the r term in the loss function, which increases the rate of false positives. Since the RANSAC plane segmentation can exclude false positives at test time, we find that false positives are preferable to having many false negatives and too little contact points.

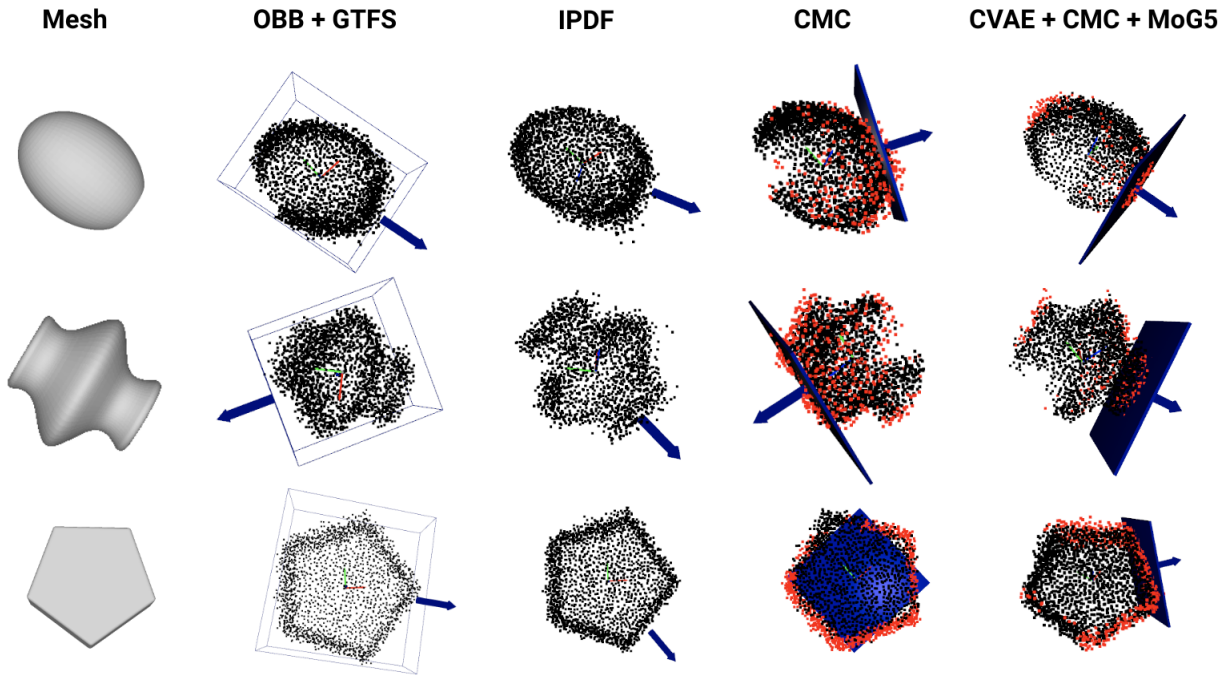| Mesh | OBB + GTFS | IPDF | CMC | CVAE + CMC + MoG5 |

Fig. 5: (Left to right: model name) Mesh, OBB + GTFS, IPDF, CMC, CVAE + CMC + MoG5. (Top to bottom: object name) Egg, Vase w/ Hole, Pentagram Stout. Although IPDF predicts rotations and does not directly generate normal vectors, we visualize here the gravity vector that has been rotated by the *inverse* of the rotation predicted by IPDF, to get a similar gauge on how well the IPDF predicted rotation is aligning the correct contact plane normal to the gravity vector.

TABLE II: Tabletop Reorientation Success Rates

	Simulation	Real world
Barrell	5/5	5/5
Bridge	5/5	5/5
Cone+Ball	5/5	5/5
Cross	5/5	3/5
J Block	5/5	5/5
Nut	4/5	5/5
Pentagram Stout	5/5	5/5
Pyramid w/ Eye	5/5	5/5
Vase w/ Hole	5/5	5/5

B. Sim2Real Real World Results

In this section, we characterize the performance of our trained model on 3D-printed, real world versions of the unseen test objects. Our evaluation metric is the success rate. Success is determined by a human evaluator as to whether the predicted contact plane closely (within 15 degrees) matches one of the height-maximizing contact planes of the true object. Because we are forced to use a relatively small number of trials in real world experiments, we match the sim and real world pose distributions by manually setting the object poses in pyBullet to the real world object poses. Additionally, we ensure our real world/simulation orientation distribution is realistic by approximately uniformly sampling within the various contact modes/stable static equilibria modes of the object. We do this because in the wild, objects are often set by humans in stable static equilibria, and the distribution

of these human-set equilibria may be significantly different from the distribution of static equilibria found by dropping the objects from the sky (which is the method we used to generate our training data in sim).

We show qualitative trials of our real world orientation predictions in Fig. 1. We see that these transformations are very accurate at reorienting the objects into an upright pose.

1) Analysis: We find that in all objects, the real world Sim2Real transfer is nearly perfect. There were two important details we implemented to get such results. The first is using the "High Accuracy" preset for the RealSense 415 cameras, which removes noise at the cost of also removing some points on the actual object itself. The second detail is appropriately scaling the real world pointclouds. Because we did not carefully calibrate the poses and intrinsic parameters of the cameras in the real world to match those in simulation, the real world pointclouds ended up being much smaller. In our first attempt without scaling, 7/9 objects were still nearly perfect (at least 4/5 successes), but the smallest Nut and Cross objects had 0/5 successes. To remedy this issue, we calculated the axis-aligned bounding boxes for the simulation pointclouds and similarly for the real world segmented pointclouds. We then took the ratio of the largest axis-aligned length for each dataset and divided the largest simulation length by the largest real world length to get our scaling factor for the real world pointclouds.

V. Related work

Multimodal Modelling Substantial progress has been made in modelling multimodal distributions using conditional generative models such as conditional variational autoencoders (CVAE) [8]. However, CVAEs still have difficulty producing crisp samples across all the data modes, in part due to difficulty optimizing the competing objectives of accurate reconstruction and KL divergence minimization [19]. Another way of modelling multimodal distributions that has worked well in robotics is classification into discretized bins of continuous quantities (such as rotations) [20], [21], [22], [4], [23]. This discretization can be challenging because if the bins are too coarse then resulting predictions will be inaccurate. Alternatively, if many small bins are constructed, it reduces the capability for information sharing across samples. Finally, a current state-of-the-art method is implicitly modelling the unnormalized density of (observation, rotation) pairs [5]. This method converts multimodal rotation prediction to unimodal density prediction.

Block Stacking While much prior work has investigated stacking towers of simple shapes, such as cuboids or triangular prisms, [24], [25], [26], [27], [28], [29], less work has demonstrated stacking complex, irregular 3D shapes. We describe works that do operate on complex shapes. [30] showed that a mesh reconstruction pipeline could be used to load natural limestones into a simulator and find stable poses for stacking. Errors in reconstruction may seriously affect downstream control based on the mesh.

[31] uses a learned, kernel-based scoring function to score translations of the pointcloud of the object-to-be-stacked. They assume the objects start off in the correct orientation, while our focus is on learning how to orient the object. [13] used a combination of learning from demonstration and reinforcement learning to learn to stack objects. Compared to their work, we consider more complex objects requiring large out-of-plane rotations and focus on introducing a new rotation representation to handle this complexity.

Pose Estimation for Symmetric Objects Many of the objects we consider in this work have discrete or continuous symmetries that introduce multimodality in the label space of rotations. Past works have investigated learning implicit function representations [5], reducing symmetry groups to canonical poses [32], and matching an input image to synthetically rendered views, in order to retrieve poses for symmetric objects [33].

VI. Acknowledgements

References

[1] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[2] V. Peretroukhin, M. Giamou, D. M. Rosen, W. N. Greene, N. Roy, and J. Kelly, "A smooth representation of belief over so (3) for deep rotation learning with uncertainty," *arXiv preprint arXiv:2006.01031*, 2020.

[3] J. Levinson, C. Esteves, K. Chen, N. Snavely, A. Kanazawa, A. Rostamizadeh, and A. Makadia, "An analysis of SVD for deep rotation estimation," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 22 554–22 565.

[4] D. Mohlin, G. Bianchi, and J. Sullivan, "Probabilistic orientation estimation with matrix fisher distributions," *arXiv preprint arXiv:2006.09740*, 2020.

[5] K. Murphy, C. Esteves, V. Jampani, S. Ramalingam, and A. Makadia, "Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold," *arXiv preprint arXiv:2106.05965*, 2021.

[6] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.

[7] C. M. Bishop, "Mixture density networks," *Aston University*, 1994.

[8] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[9] K. G. Derpanis, "Overview of the ransac algorithm," *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.

[10] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[11] T. Möller and J. F. Hughes, "Efficiently building a matrix to rotate one vector to another," *Journal of graphics tools*, vol. 4, no. 4, pp. 1–4, 1999.

[12] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," *arXiv preprint arXiv:1611.02731*, 2016.

[13] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[14] D. Dimitrov, C. Knauer, K. Kriegel, and G. Rote, "On the bounding boxes obtained by principal component analysis," in *22nd European Workshop on Computational Geometry*, 2006, pp. 193–196.

[15] 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.

[16] S. Devgon, J. Ichnowski, A. Balakrishna, H. Zhang, and K. Goldberg, "Orienting novel objects using self-supervised rotation estimation," in *2020 IEEE Conference on Automation Science and Engineering (CASE)*, 2020.

[17] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[18] M. S. Ahn, H. Chae, D. Noh, H. Nam, and D. Hong, "Analysis and noise modeling of the intel realsense d435 for mobile robots," in *2019 16th International Conference on Ubiquitous Robots (UR)*. IEEE, 2019, pp. 707–711.

[19] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[20] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," *ICRA*, 2016.

[21] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," in *Advances in Neural Information Processing Systems*, 2016, pp. 5074–5082.

[22] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2146–2153.

[23] S. Mahendran, H. Ali, and R. Vidal, "A mixed classification-regression framework for 3d pose estimation from 2d images," *arXiv preprint arXiv:1805.03225*, 2018.

[24] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," in *Robotics: Science and Systems VII*, vol. 7, 2011, pp. 57–64.

[25] Y. Duan, M. Andrychowicz, B. C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," *arXiv preprint arXiv:1703.07326*, 2017.

[26] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[27] M. Janner, S. Levine, W. T. Freeman, J. B. Tenenbaum, C. Finn, and J. Wu, "Reasoning about physical interactions with object-oriented prediction and planning," in *International Conference on Learning Representations*, 2019.

[28] O. OpenAI, M. Plappert, R. Sampedro, T. Xu, I. Akkaya, V. Kosaraju, P. Welinder, R. D'Sa, A. Petron, H. P. d. O. Pinto, *et al.*, "Asymmetric self-play for automatic goal discovery in robotic manipulation," *arXiv preprint arXiv:2101.04882*, 2021.

[29] R. Li, A. Jabri, T. Darrell, and P. Agrawal, "Towards practical multi-object manipulation using relational reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4051–4058.

[30] F. Furrer, M. Wermelinger, H. Yoshida, F. Gramazio, M. Kohler, R. Siegwart, and M. Hutter, "Autonomous robotic stone stacking with online next best object target pose planning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2350–2356.

[31] O. Kroemer, S. Leischnig, S. Luettgen, and J. Peters, "A kernel-based approach to learning contact distributions for robot manipulation tasks," *Autonomous Robots*, vol. 42, no. 3, pp. 581–600, 2018.

[32] G. Pitteri, M. Ramamonjisoa, S. Ilic, and V. Lepetit, "On object symmetries and 6d pose estimation from images," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 614–622.

[33] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3d orientation learning for 6d object detection," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 714–729, 2020.