7 Appendix

7.1 Data Capture and Annotation Pipeline

We use an eye-in-hand Franka Emika Panda with Intel RealSense RGBD D435i camera to collect the dataset. Intel D435i camera has a resolution of 1280x720 with field of view of 87 degrees horizontally and 58 degrees vertically [4]. All RGB and depth images are captured under 640x480 resolution . To fully automate the annotation, AprilTags in the 36h11 family, where each tag is a square with 40mm side length, is used. Objects are placed to a fixed location and pose with respect to the tags, which can be used later generate the ground truth with the object mesh. The Franka Panda robot eye-in-hand system is built, and controllers for the robot as well as the RealSense camera are programmed using FrankaPy [42]. In terms of automating annotation, we use the AprilTag2 [41] approach, which has more convenient API support and enables easier 6DoF Pose calculations. The detailed dataset collection process is performed as follows:

- An array of AprilTag2 are placed around a transparent object, each tag is from the same family with a different index. And the offset from the tag and object is known using a printed template.
- Multiple positions and viewing angles are selected. For every viewpoint, the robot aims the camera towards objects and captures the RGB-D images.
- For automatic annotation, we first detect the AprilTag pose using the color input. The object pose can be extracted via the tag's pose. Using the 3D model of the object, the segmentation mask, ground truth depth map can be computed.
- Every viewpoint in the dataset will consist of the color image, raw depth map, ground truth depth, segmentation mask as well as each object's 6DoF pose.

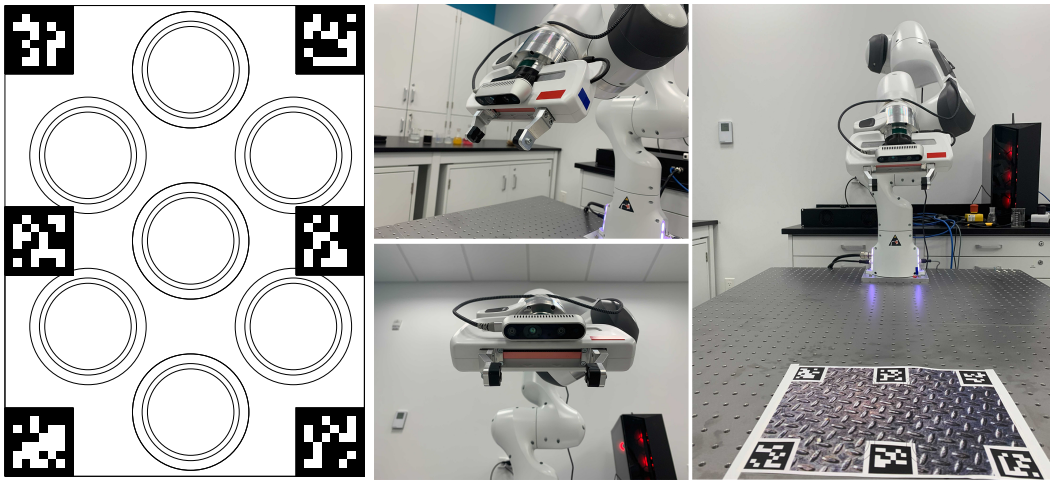

Figure 7: Dataset Collection Setup. Left: Template with April Tags and alignment marker for object placement. The relative positions for markers and tags are fixed so transparent object pose can be calculated. Right: Franka robot and eye-in-hand RGB-D camera setup for dataset collection.

7.2 Implementation Details

The Point Cloud Completion module is trained with Adam optimizer[50] with initial parameters $\alpha = 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$. The learning rate decays by 2 after 50 epochs and training is stopped after 300 epochs. For training the decoder modulation depth completion module, we use the Adam optimizer [50] with initial learning rate set to 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use the EfficientNet-b4 [51] pretrained on ImageNet [52] as the backbone. Experiments conducted on the ClearGrasp [1] dataset are trained with a resolution of 240×320. Experiments conducted on

[4]Intel RealSense D435i's specifications are listed in https://ark.intel.com/content/www/us/en/ark/products/190004/intel-realsense-depth-camera-d435i.html

Figure 8: Four different color backgrounds used for April Tag templates in TODD dataset

Figure 9: Six objects and corresponding 3D models used in TODD , from left to right is Beaker 1, Beaker 2, Beaker 3, Flask 1, Flask 2 and Flask 3. From top to bottom is empty objects, objects filled with five different liquids and 3D models.

our proposed dataset, TODD , are trained with a resolution of 480×640. Training is performed end-to-end for 100 epochs, with early stopping.

7.3 Evaluating the Point Cloud Completion Module Separately

In Table 4, we present an evaluation of the quality of our Point Cloud Completion model. $only-valid$ and all refer to different considerations when computing the metrics. $only - valid$ refers to only calculating the metrics using points within the transparent object mask which are valid. This means that the non-valid depths as produced by the sparse Point Cloud Completion module is disregarded during $only - valid$ version of metrics calculation. all refers to calculating the metrics using the entirety of the object mask, by considering non-valid depths of the Point Cloud Completion module as zero values. As a result, due to the sparsity of the Point Cloud, we can see that the all version, which considers the invalid depths in the metric calculation, has significantly worse values than the $only - valid$ version. We can also see from the results of Table 4 that the $only - valid$ version achieved relatively high scores across the board for the metrics shown. This means that the sparse

Table 4: Evaluation of PCC Module Performance. Validation and Test set both contain non-overlapping novel scenarios, placements, and objects. * Metrics used are an adapted version that excludes empty depths.

	RMSE (↓)	MAE (↓)	$\delta_{1.05}$ (↑)	$\delta_{1.10}$ (↑)	$\delta_{1.25}$ (↑)	Rel (↓)
	Validation					
TanspareNet-PCC Only$^{*}_{\text{only-valid}}$	0.0115	0.0074	0.8949	0.9587	0.9969	0.0150
TanspareNet-PCC Only$_{\text{all}}$	0.2549	0.1820	0.4521	0.4785	0.4938	0.5146
	Test					
TanspareNet-PCC Only$^{*}_{\text{only-valid}}$	0.0118	0.0076	0.8874	0.9588	0.9953	0.0158
TanspareNet-PCC Only$_{\text{all}}$	0.2601	0.1887	0.4270	0.4547	0.4681	0.5400

depths that are output from the Point Cloud Completion module is indeed representative of the depth within the region of the transparent object, although relatively sparse. However, we note that due to the computation load required to increase the density of the point cloud output, we can only produce a relatively sparse point cloud from the Point Cloud Completion module, which cannot be directly used for downstream manipulation tasks. Hence, we add the Depth Completion (DC) module discussed in the main text.

7.4 Additional Qualitative Results

In Figure 10, we present some additional qualitative examples of the performance of TranspareNet on TODD . We note that TranspareNet is generally more capable at producing accurate depths along edges of glass vessels (Row 1). We can also see that in cases where the Depth Completion (DC) is inaccurate, along sides of vessels, TranspareNet is able to provide a more complete depth (Row 2). For cases where there are depth artifacts along depth discontinuities, TranspareNet is able to filter out the artifacts and generate a more realistic depth map where the outline of the respective vessels are artifact-free (Row 3, Row 6).

In Figure 11, we show some failure cases of TranspareNet. Through analysis of the failure cases, we see that failures often occur when the point-cloud generated by the Point Cloud Completion (PCC) module is too sparse in nature, or if there is heavy occlusion of an object by another transparent object, such that the prominent edges of the object is non-visible.

7.5 Pose Estimation

RGB-D information is vital for downstream robotic manipulation tasks, especially for 3D object 6DoF pose estimation. We demonstrate the importance of correct and complete depth information for pose estimation by comparing results on sensor raw depth, ground truth depth, and predicted depths from TranspareNet. For the comparison, MaskedFusion[53], a state-of-the-art 6DoF pose estimation network, is trained on TODD with instance segmentation masks, depth and RGB images. MaskedFusion is based on DenseFusion[54], but includes additional mask branch in their PoseNet[53] and crops out corresponding image and depth based on object's mask.

Metrics: Following MaskedFusion [53], the Average Distance of Model Points (ADD) is used to evaluate distance between ground truth model and predicted model. We denote the ground truth rotation R and translation t and the estimated rotation $\hat{R}$ and translation $\hat{t}$. The average distance

Table 5: Pose Estimation: Quantitative evaluation of MaskedFusion [53] 6DoF pose using the ADD metric on the TODD dataset using raw, ground truth and TranspareNet predicted depth.

	Beaker 1	Beaker 2	Beaker 3	Flask 1	Flask 2	Flask 3	Average
	ADD (↓)						
RGB + Raw Depth	0.03251	0.01067	0.008682	0.01001	0.01016	0.007643	0.01578
RGB + GT Depth	0.01913	0.006411	0.004913	0.01097	0.009505	0.005148	0.01187
RGB + TranspareNet Depth	0.01925	0.006553	0.005231	0.01091	0.009764	0.005406	0.01209
	<2cm (↑)						
RGB + Raw Depth	51.79	96.65	98.93	95.61	97.66	99.16	85.09
RGB + GT Depth	74.74	98.61	98.91	91.67	98.31	99.08	88.24
RGB + TranspareNet Depth	74.78	98.71	99.24	91.61	98.02	99.75	88.14

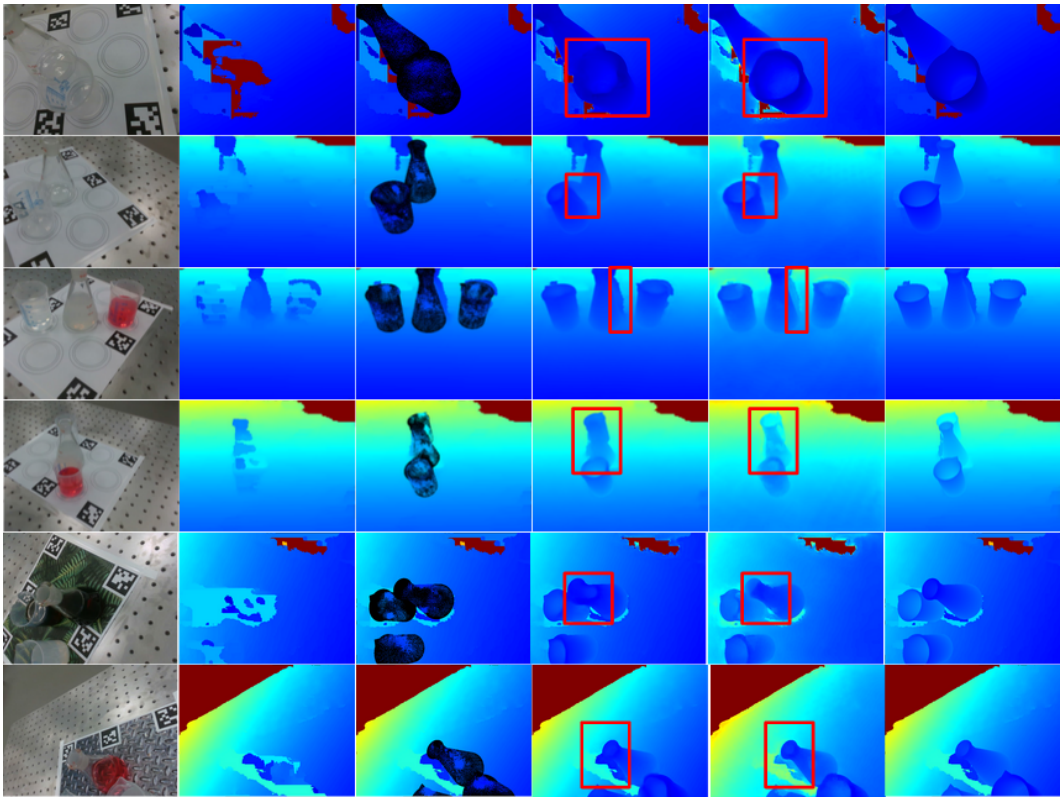

Figure 10: Visualization of performance on TODD dataset. From left to right, (a) RGB image (b) Raw depth from RGB-D sensor (c) PCC predicted depth (d) DC predicted depth (e) TranspareNet predicted depth (f) Ground Truth.

is calculated using the mean of the pairwise distances between the 3D model points of the ground truth pose and the estimated pose. The percentage of objects whose ADD is less than 2cm is used as another metric.

$$ADD = \frac{1}{m} \sum_{x \in M} |(Rx + t) - (\hat{R}x + \hat{t})| \tag{4}$$

We trained three versions of MaskedFusion on TODD with raw, ground truth and TranspareNet depth respectively, each model is trained for 50 epochs and trained with Adam optimizer[50] with initial parameters $\alpha = 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$. Table 5 demonstrates the effectiveness of TranspareNet's, with the performance of MaskedFusion trained on TranspareNet predicted depth closely matches with the version using ground truth depth as input. Comparing both models with the raw depth one, the importance of correct depth estimation is highlighted, as distorted and incomplete raw depth reported by RGB-D sensor significantly deteriorates MaskedFusion's accuracy.

Our work is directly relevant to perception-based closed-loop manipulation. Most manipulation techniques use point clouds as the perceptual input which is useful for object pose estimation and sim-to-real transfer [55, 56, 57, 58]. Point clouds are deprojected from raw depth maps taken from sensors, so the quality of the depth map directly affects the input for manipulation. In previous works, [59] uses higher quality cameras and [60] observer scenes from multi-viewpoints to deal with sensor noise and occlusion. Recently, [56, 57] take partial point clouds as input and train a multi-stage network on synthetic data. To simulate sensor noise, they employ various data-jittering methods on simulated point clouds. However, these methods assume objects in constructed scenes are opaque and overall geometrical information is preserved in point clouds. For transparent objects, ToF / stereo-based depth sensors cannot accurately capture the depth of transparent and specular objects. Figure 12 illustrates the quality of the depth estimation by an Intel Realsense Camera where the depth information of the glass beaker is mostly missed or incorrectly captured.

According to ClearGrasp report, the effect of an inaccurate depth map on downstream manipulation tasks has been quantified to 12% grasping success rate before any depth estimation methodologies for

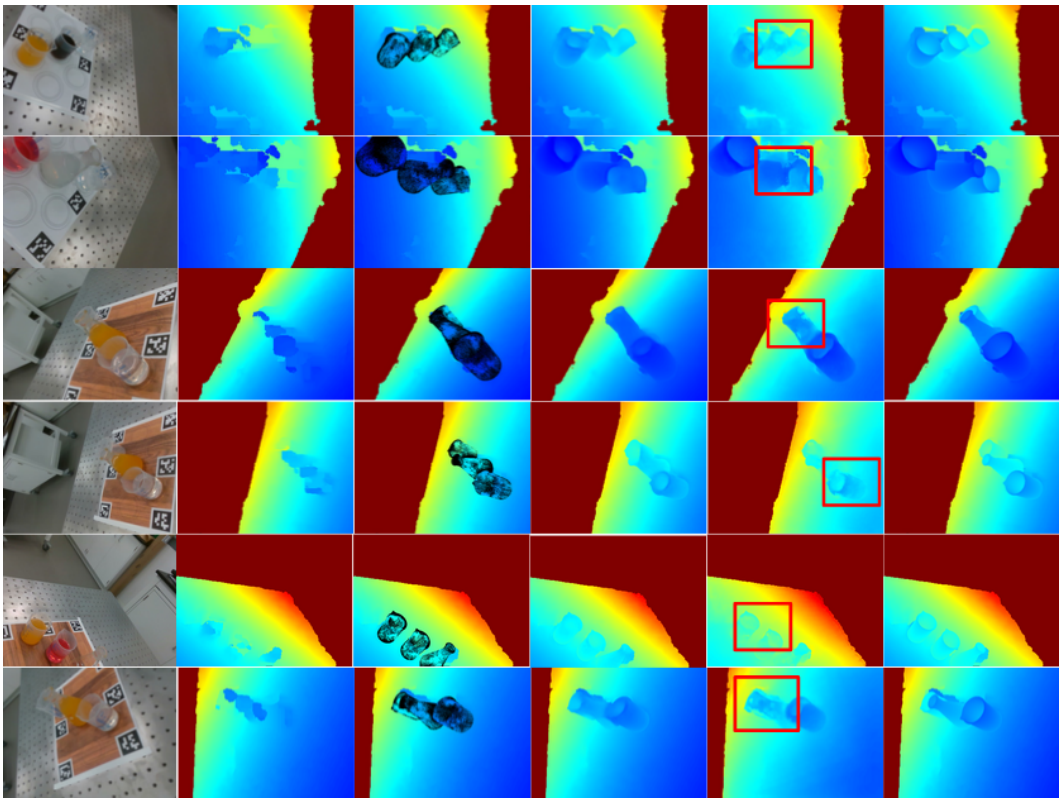

Figure 11: Visualization of failure cases on TODD dataset. From left to right, (a) RGB image (b) Raw depth from RGB-D sensor (c) PCC predicted depth (d) DC predicted depth (e) TranspareNet predicted depth (f) Ground Truth.

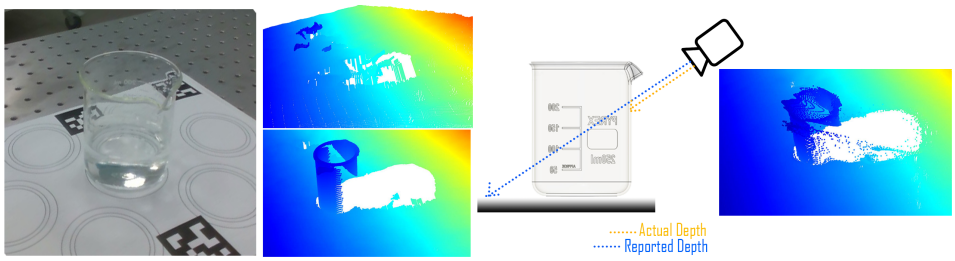

Figure 12: Distortion in Transparent Object Depth. From left to right, (a) RGB image, (b) Raw Depth and Ground Truth Depth, (c) Incorrect depth measurement due to reporting background depth as transparent object depth, (d) Predicted Depth from TranspareNet

a parallel jaw gripper, and 64% for suction. This shows the importance of accurate depth estimation for handling transparent objects.

7.6 Evaluation of Surface Normals

In evaluating surface normals, we use the same metrics as that of ClearGrasp [1]. We compute the mean and median errors of the predicted vectors, as compared to ground truth over all pixels within the image. We also report percentages of the predicted normals which are within $11.25°$, $22.5°$, and $30°$ of that of the ground truth normals. Note that we mask the pixels which include transparent vessels for metric computation.

Because we do not directly predict normals within our model, we compute surface normals from depth estimations. We consider the depth image to be a function of $z(x, y)$, where x is along the horizontal axis, y is along the vertical axis, and z denotes the depth (in metres).

Table 6: Evaluation of Surface Normals. We present an assessment of normals estimated from the predicted depth from TranspareNet.

	Mean (↓)	Median (↓)	$\delta_{11.25^\circ}$ (↑)	$\delta_{22.5^\circ}$ (↑)	δ_{30° (↑)
Novel 1 Object Scene	17.32	14.47	44.35	71.99	82.81
Novel 2 Object Scene	17.01	13.92	46.14	72.12	82.62
Novel 3 Object Scene	19.45	15.75	39.14	65.74	77.81
Novel Objects Combined	18.57	15.14	41.54	68.17	79.70

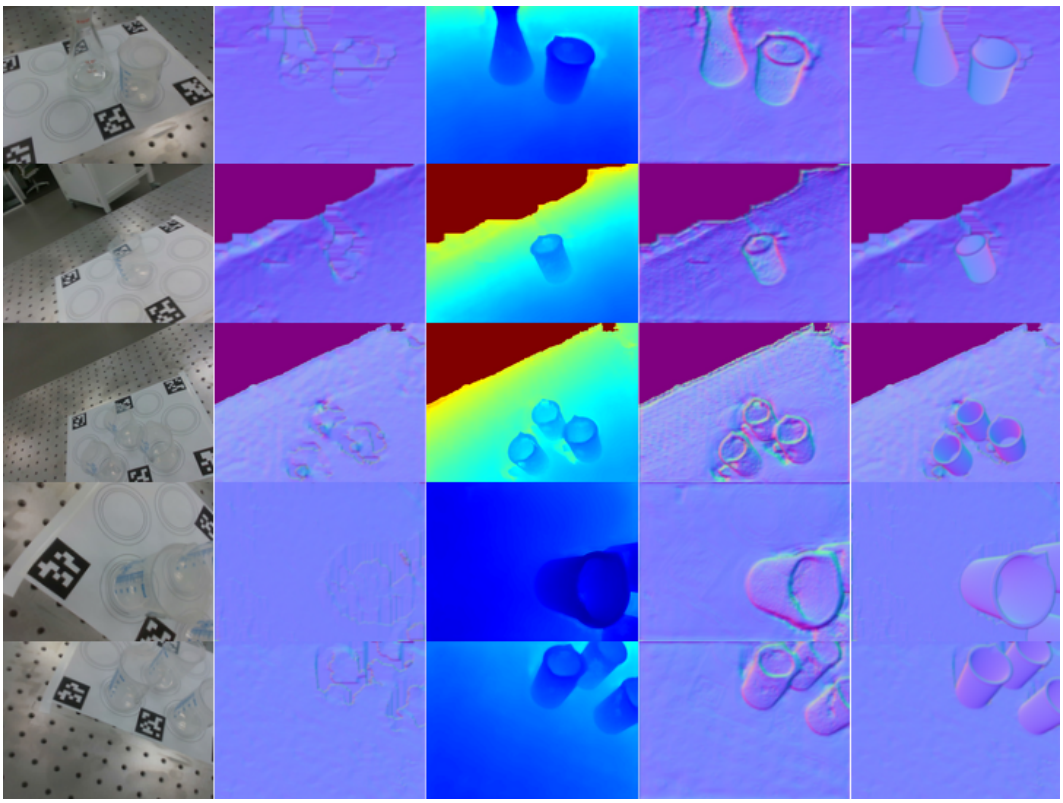

Figure 13: Visualization of normals estimated from depth on TODD dataset. From left to right, (a) RGB image (b) Normal estimated from raw depth from RGB-D sensor (c) Depth predicted by TranspareNet (d) Normals estimated from predicted depth (e) Normals estimated from ground truth depth.

The orthogonal vectors tangent to the plane parallel to the x and y axis can then be represented as $(1, 0, dz/dx)$ and $(0, 1, dz/dy)$, respectively. Taking the cross product of these vectors, we arrive at the vector which represents the surface normal $(-dz/dx, -dz/dy, 1)$.

We present an evaluation of the quality of estimated normals based on angular difference with the estimated ground truth normals in Table 6. We also provide a visualization of estimated normals in Figure 13.

7.7 Inference using non-perfect object masks

In a real life deployment scenario, it is often the case where there is no segmentation mask to select the objects of interest for depth completion. Therefore, we present an evaluation for the inference ability of our model when the mask is predicted using the pre-trained transparent object segmentation model from [3]. Using the pre-trained model, we find that the predicted masks have an IoU of 0.484225 with the ground truth masks. Using the predicted masks, we evaluate the performance of (1) TranspareNet Depth Completion Only, and (2) TranspareNet which involves Point Cloud Completion and Depth Completion, results are presented in Table 7. We also show a visualization of the differences in estimated depths (when the mask is predicted), as compared to when the mask is ground truth in Figure 14.

Table 7: Inference using predicted object masks. We assess the performance of various models with Novel 1 Object images when the given mask input is predicted by the pre-trained model in [3]. The arrows beside the metrics denote whether lower or higher values are more desired.

	RMSE (↓)	MAE (↓)	$\delta_{1.05}$ (↑)	$\delta_{1.10}$ (↑)	$\delta_{1.25}$ (↑)	REL (↓)
	Novel 1 Object Scene					
TranspareNet-DC Only	0.0446	0.0362	0.4064	0.6056	0.8511	0.0914
TranspareNet (ours)	**0.0341**	**0.0279**	**0.4740**	**0.7056**	**0.9102**	**0.0598**

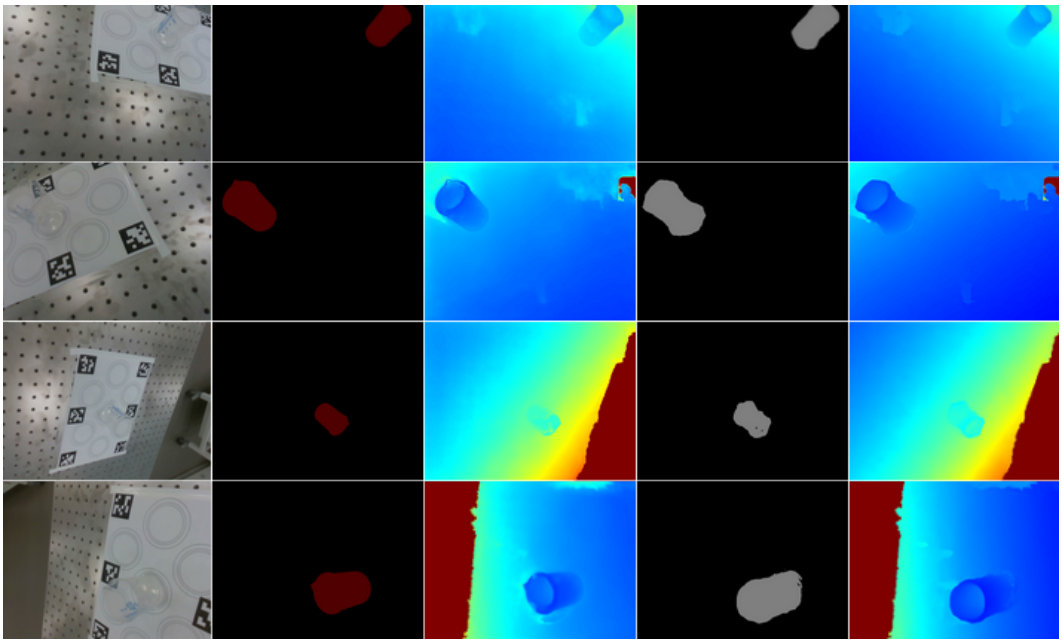

Figure 14: Visualization of differences in predicted depth when the given mask is predicted vs. ground truth. From left to right, (a) RGB image (b) Ground Truth Mask (c) Depth predicted by TranspareNet with GT Mask (d) Predicted Mask (e) Depth predicted by TranspareNet with Predicted Mask.

Note that this model was only trained using Trans10k. There is no fine-tuning on our dataset. We anticipate that fine-tuning the pre-trained Trans10k [3] on our dataset will yield improved predicted segmentation IoU, and hence improved depth completion by TranspareNet.

7.8 Assessment of the Importance of April Tags in Model Performance

Because our method automates object pose extraction using April Tags [41], we hope to identify whether April Tags play a role in the performance of our depth completion module. To evaluate whether such a connection exists, we mask out all April Tags from the RGB-input during inference, and compare with the TranspareNet performance when the tag is not masked.

We present an evaluation in Table 8. We also show a visualization of the differences in predicted depths with and without Tags removed in Figure 15, as can be seen the visualization, the predicted depths with and without April Tag masked are similar, which shows that our model did not overfit to the positioning of the April Tags, and is learning useful insight based on the geometry the vessels.

7.9 Extended Test Set

We introduce an extended test set, consisting of a glass bowl and cup, as shown in Figure 16. The test set consists of 4k images of these objects arranged in different settings. We provide an evaluation of TranspareNet on the extended test set. Results are shown in Table 9. We further provide visualizations for a representative sample of the extended test set, and the quality of the predicted depth in Figure 17.

Additionally, our automated dataset creation pipeline allows users to customize the dataset for their intended application scenario by rapidly collecting and annotating images for the transparent

Table 8: Depth completion results with and without April Tags on TODD Dataset. We assess the performance of various models with Novel 1 Object images, as well as novel cluttered images with 2 or 3 objects. The arrows beside the metrics denote whether lower or higher values are more desired.

	RMSE (↓)	MAE (↓)	$\delta_{1.05}$ (↑)	$\delta_{1.10}$ (↑)	$\delta_{1.25}$ (↑)	REL (↓)
	Novel 1 Object Scene					
Tags Removed	0.0260	0.0230	0.3735	0.7451	0.9855	0.0614
Tags Not Removed	0.0166	0.0140	0.7133	0.9299	0.9945	0.0398
	Novel 2 Object Scene					
Tags Removed	0.0288	0.0233	0.3611	0.6649	0.9665	0.0687
Tags Not Removed	0.0194	0.0159	0.6475	0.8693	0.9876	0.0496
	Novel 3 Object Scene					
Tags Removed	0.0329	0.0282	0.3189	0.6332	0.9704	0.0739
Tags Not Removed	0.0232	0.0190	0.5817	0.8408	0.9904	0.0546
	Novel Combined					
Tags Removed	0.0309	0.0271	0.3293	0.6452	0.9706	0.0707
Tags Not Removed	0.0213	0.0175	0.6180	0.8619	0.9905	0.0510

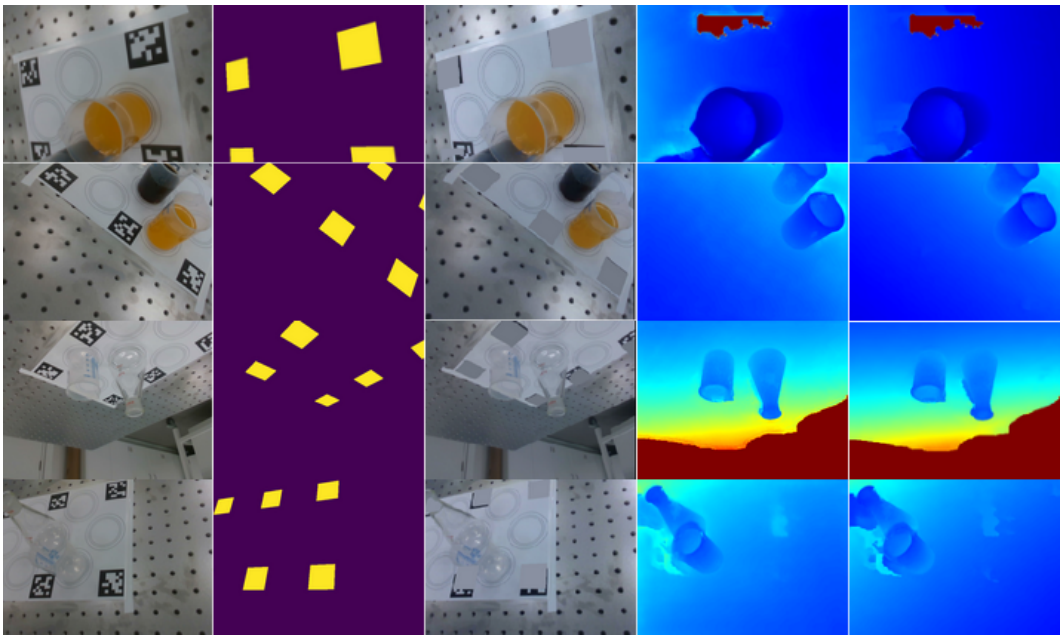

Figure 15: Visualization of differences in predicted depth when the April Tags is Masked out vs. Not Masked out. From left to right, (a) RGB image (With Tag) (b) April Tag Mask (c) RGB image (Tags Removed) (d) Depth predicted by TranspareNet with Tag in RGB (e) Depth predicted by TranspareNet without Tag in RGB.

Table 9: Inference on Extended Test Set. We assess the performance of various models using the extended Test set. The arrows beside the metrics denote whether lower or higher values are more desired.

	RMSE (↓)	MAE (↓)	$\delta_{1.05}$ (↑)	$\delta_{1.10}$ (↑)	$\delta_{1.25}$ (↑)	REL (↓)
TranspareNet-DC Only	**0.0349**	**0.0308**	0.3428	0.5580	**0.9228**	**0.0808**
TranspareNet (ours)	0.0363	0.0335	**0.4236**	**0.5702**	0.8417	0.0868

objects in the scene. For example, the additional images only took 8 hours to collect and annotate automatically, in contrast with ClearGrasp [1], which involved manual placement, capture, and annotation of the small (200) real-life test set. Although the existing TODD dataset is limited in object types and scenes, we think the pipeline provides a feasible approach to tailor the dataset for specific applications.

Figure 16: New transparent objects (bowl and cup) with different shapes for TODD extended test set.

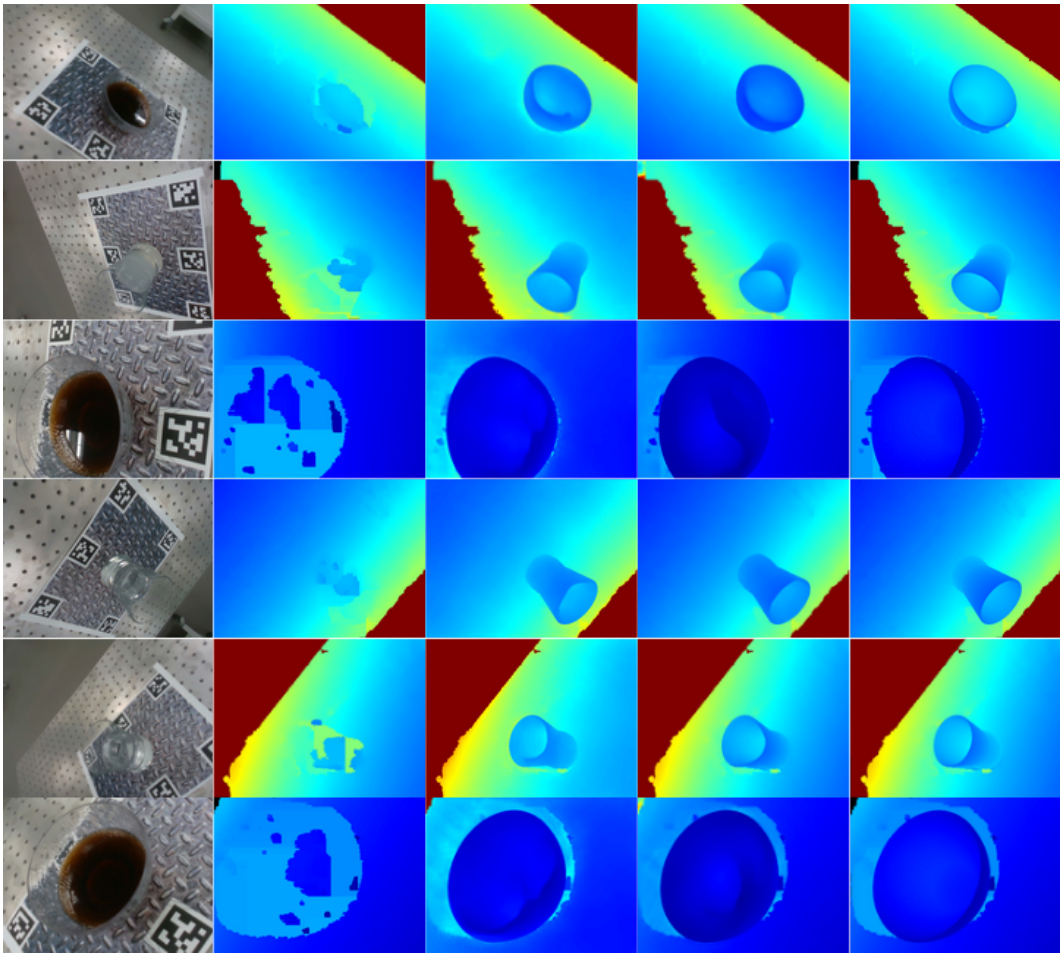

Figure 17: Visualization of Extended Test Set. The extended test set contains new vessels, the shapes of which have not been exposed during the training process. From left to right, (a) RGB image (b) Raw Depth (c) Depth predicted by TranspareNet-DC Only (d) Depth predicted by TranspareNet (e) Ground-truth Depth