Learning to Estimate Multi-view Pose from Object Silhouettes: Supplementary Material

Yoni Kasten[1], True Price[2], David Geraghty[2], and Jan-Michael Frahm[2]

[1] NVIDIA Research
[2] Meta
ykasten@nvidia.com, {jtprice,dger,jmfrahm}@meta.com

In this supplementary material, we provide additional details about our datasets, and we show a number of additional results: a small experiment with real image masks collected using Mask R-CNN; 3D reconstruction results of IDR [9] using our estimated poses for initialization; a visualization of the relative epipolar geometries learned by our method, showing that our method predicts consistent pose estimates even when the actual pose prediction is incorrect; results from the new Glass Figurines Dataset; qualitative results on manually segmented images of a glass swan sculpture in different environments; visualizations of IOU results for instances in the RealScan dataset; and full quantitative results on the 3D Warehouse and RealScan datasets. We also present a proof of how to find the optimal globally-aligning rotation matrix in Section 3.2 of the main paper.

For convenience, here are links to the different sections of figures and tables in this document:

– Results using Mask-RCNN on images of a chair
– IDR [9] results using our initialization
– Visualization of the epipolar mapping constraint
– Images and results for the Glass Figurines dataset
– Qualitative results on a glass swan sculpture
– RealScan IOU visualizations
– Complete table of 3D warehouse results
– Complete table of RealScan results

1 Dataset Details

3D Warehouse dataset. Table 1 provides a breakdown of the number of training and testing objects used in our experiments. Overall, we adopted 15 object classes for training and testing, and 5 for only testing. For each training class, we took 600 objects or 70% of the total number of objects in the class, whichever is lower; for testing, we took 20% of the total number of objects in the respective class. Note that during training, we created a new set of 10 random views with a random global rotation for each batch element. For testing instances, we rendered one set of views per object with a single random global rotation of the

Table 1. Number of objects from each class in the 3D Warehouse dataset used for training and testing our network.

Object Class	# Training Objects	# Validation Objects
airplane	600	809
bench	600	362
bus	600	188
camera	80	22
faucet	521	149
guitar	558	160
knife	297	85
microphone	47	14
motorcycle/bike	236	68
piano	168	48
pistol/handgun	215	62
pot	422	120
rifle	600	474
oven/stove	153	44
vessel/watercraft	600	388
bathtub	0	171
car	0	703
chair	0	1356
mailbox	0	19
lamp	0	464
total	5697	5706

object. We rendered each image with a focal length equal to 1.4 times the image width in pixels, and with a principal point in the center of the image.

RealScan dataset. As mentioned in the paper, we also evaluated our method on a new set of 30 high-quality scanned 3D object models. Renderings of several such models can be seen in the top rows of Figures 8-15, and a full list of object names is provided in Table 4.

Glass Figurines dataset. This new dataset consists of 11 objects captured from 10 different views using April Tags [5] to determine ground-truth camera poses. The objects in this dataset present various challenges for reconstruction, including non-Lambertian reflectance, low texture, and semi-transparency (Figure 1). We manually extracted image masks for the objects and use these 10 masks as input to our method. Examples of the segmented images, along with complete results for the dataset, are shown in Table 2. We discuss the results in the next section.

2 Supplementary Results

2.1 Results with automatically extracted masks

One of our key motivations in this work is enabling object-centric camera pose estimation for unordered image collections of a low-texture object. To this end, we conducted a proof-of-concept experiment where we took pictures of a single chair in different settings, mimicking a scenario where multiple photos of an object are captured in different environments. We extracted an object mask of the chair in each image using the Detectron2 implementation of Mask R-CNN trained on the ImageNet dataset [2,8]. Figure 2 shows the result from running these mask images through our network. Qualitatively, we observe consistent left-right, top-bottom, and forward-backward ordering of the images when the poses are viewed as 3D frusta.

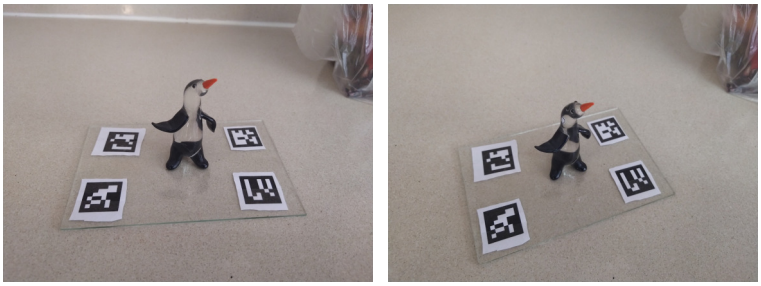

Fig. 1. Two example images from the Glass Figurines dataset, showing the same object from two different angles. See Table 2 for more images and results from this dataset.

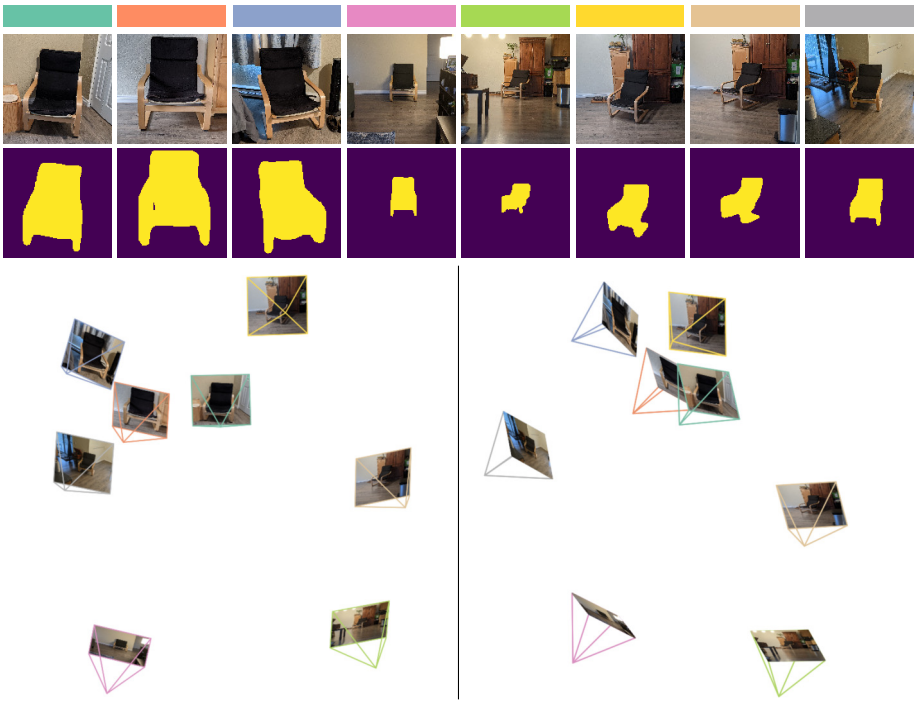

Fig. 2. Using Mask R-CNN [2] to extract masks for our method. Top: Input images with associated camera colors above. Middle: Extracted masks. Bottom: Camera frusta of our poses observed from two different views.

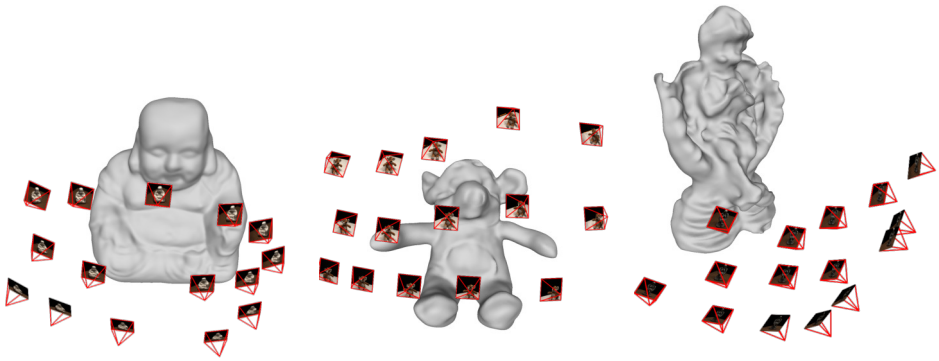

Fig. 3. 3D object reconstructions from DTU [3] using IDR [9] with our camera poses for initialization. The camera frusta show our poses before optimization by IDR.

This result is notable especially because the Mask R-CNN masks are much less accurate than the masks used in any of our previous experiments, which is quite promising for the generalization of our method. However, we do observe that our network output is somewhat sensitive to substantially inaccurate object masks. In particular, two additional object masks we captured (not shown) had segmentations that did not correctly capture the legs or arms of the chair. Including these in our inputs negatively affected the results. The major reason for this is due to the data used for training — if such categorical masking errors were included, then our network would have a better chance of learning to adjust confidence measures accordingly.

2.2 Initialization for IDR

The results presented in the main paper indicate that our method has fairly reliable accuracy, but the question remains whether this accuracy is low enough, in practice. To this end, we applied the implementation of IDR [9] provided by the authors, with poses initialized from our DTU results. Note that while IDR does optimize camera poses, it still requires reasonably accurate camera poses for initialization. As shown in Fig. 3, IDR recovers qualitatively accurate models using our initialization. This proof of concept supports our method's practicality, showing that our silhouette-based pose estimates provide adequate initialization for neural-network-based 3D reconstruction. Overall, we find the reconstruction quality quite promising, and the results appear to validate the quantitative performance of our method.

2.3 Epipolar mapping constraints

In the introduction to Section 3 of the main paper, we mentioned that our network is able to learn epipolar mapping constraints despite not directly training to respect them. Figure 5 shows a visualization of this quality; see the caption

and Figure 4 for a discussion on how to interpret the images. The epipolar mapping constraint states that, for a relative pose to be valid between two silhouette images, the silhouette epipolar lines should all intersect the silhouette in the other image.

In the top four rows of the figure, we show a "good" result of our method with low pose error. Naturally, the epipolar mapping constraint seems well-satisfied, here, and in each image, the general location of the epipole is consistent with the ground truth. The more interesting case is the second instance, where our network's pose accuracy was much lower. It is apparent from the epipole positions that the poses are incorrect (for example, the epipolar lines point in entirely the wrong direction in the first image pair). However, the epipolar mapping constraint is still well satisfied in our predictions.

2.4 Results on the Glass Figurines dataset

Table 2 shows complete results for the 11 objects in the Glass Figurines dataset. As with our results for the 3D Warehouse and RealScan datasets, we report the mean and median pose error error considering rotation, translation scale, and translation direction. We also evaluate our confidence measure by comparing the error of our confidence-ranked results against an oracle ranking. For this dataset, we observe generally similar results to those found in the other three datasets that were evaluated, with many results below 8° error in rotation, and low error in both translation metrics. We did observe one failure case (parrot) in this dataset, with a rotation error greater than 20°. We hypothesize that this failure is caused at least in part by the set of viewing angles from which the object was captured. Additional vantage points from the left and right of the object might be able to overcome silhouette ambiguities in the set of collected images, which mainly consist of side and top-down views of the object.

Also in Table 2, we provide a comparison to the camera results estimated by GNeRF [4] and by COLMAP's incremental SfM pipeline [6].

GNeRF. For GNeRF, we used the code provided by the authors[3] and conducted two experiments, one training the network on masked 256×256px color image inputs, and the other training with the same silhouette images used as inputs to our network. For each object, we trained the GNeRF model to 40k epochs with refinement starting at 12k epochs and progressive training at 20k epochs, as suggested by the authors. Since GNeRF requires a prior on the camera pose distribution, we tried two different settings per object: One with the distribution set to the azimuth range ($\pm 30°$), elevation range ($\pm 30°$), and distance-to-object range $[3.2r, 6r]$ used by our method during training, and the other with the same rotation ranges but fixed distance to the object of $4r$. Since GNeRF failed for more objects using the former settings, we report results using the latter settings, except for the color-image "dolphin" and "flamingo" datasets, which suceeded using the former settings but failed for the latter.

[3] https://github.com/MQ66/gnerf

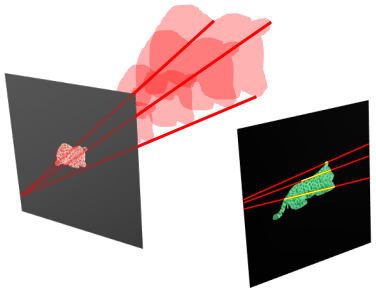

Fig. 4. An example two-view epipolar geometry. Each silhouette point in the first (left) image projects to a ray in 3D space (red lines), and all points together form a cone of rays. Each ray projects to a line in the second (right) image under the relative pose between the images, and the corresponding 2D point in the second image must lie somewhere along this line. The epipolar mapping constraint dictates that, for a given relative pose to be valid between the images, all such epipolar lines must intersect the object silhouette in the second image.

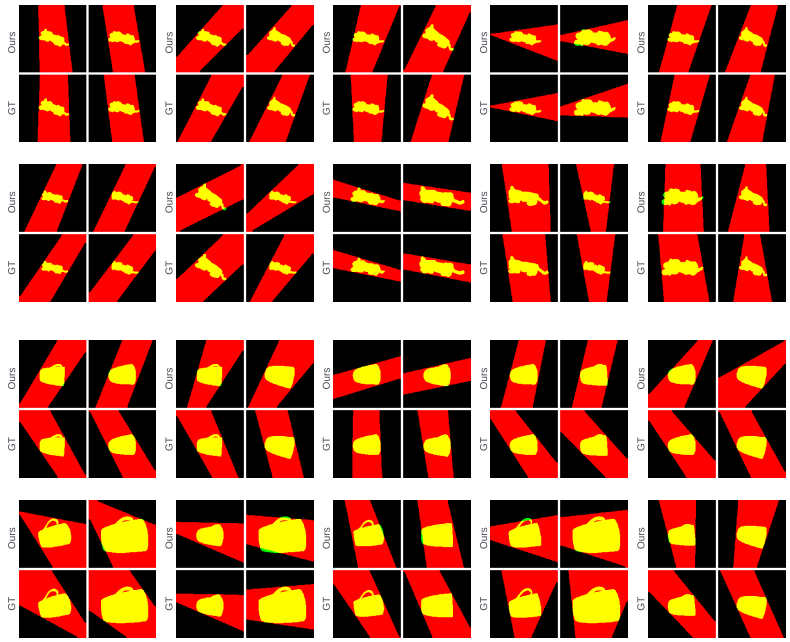

Fig. 5. A sample of two-view epipolar mappings (out of 45 total) for two objects from the RealScan dataset. Every odd row shows our result, and every even row shows the ground-truth result. The pencil of red lines in each image is a visualization of the epipolar lines from the left image's silhouette points into the right image, and *vice versa*. The input silhouette is shown in green, and the overlap of the silhouette and epipolar lines is shown in yellow. Top four rows: Visualization of a "good" result (Cheetah) in terms of pose error (4.5° rotation error). Bottom four rows: Visualization of a "bad" result (Plastic Cup) in terms of pose error (17.8° rotation error).

COLMAP. We provided COLMAP with the original 4000×3000px images for reconstruction, along with object masks at the same resolution. Masks are required here because the object is not stationary relative to the background scene. (Without masks, COLMAP will reconstruct the background only.) We extracted upright DSP-SIFT features [1] for each image and performed all-pairs matching prior to running COLMAP's SfM pipeline.

Results. All GNeRF and COLMAP statistics in Table 2 are reported as mean (median). GNeRF did not perform well on this dataset for either type of input (color images or silhouettes), with high pose rotation errors across all cases. The method tended to overfit to the training images and was unable to fully learn a correct radiance field representation; the estimated poses were therefore less informative to the final model. For several objects, GNeRF failed to populate the radiance field, at all, resulting in an empty scene with all estimated poses set to identity. This is not indicative of the general performance of GNeRF, but rather a result showing poor results of the method for our particular dataset. We suspect that GNeRF may have faired better on this dataset if it was provided with more images, and possibly with a wider range of input viewpoints. Nevertheless, compared to our approach, GNeRF ultimately requires more data to be available, and it uses a very large amount of compute time (on the order of hours or tens of hours) to process even a single image set. Our network runs in a single pass without additional training and produces more accurate pose estimates even on smaller image collections.

Concerning COLMAP, for 5 of the objects that had more texture and/or opacity, COLMAP performed very well, with much less than 1° error in rotation and translation. For the snake object, which has "medium" texture but very thin structure, COLMAP performed worse than for the other 5 successful objects, but its results for rotation error were still considerably better than those of our method on the same object. Interestingly, our method shows higher accuracy in the translation metrics for this object.

Finally, COLMAP failed to reconstruct 5 of the 11 objects (see rows marked as "failed" in the table), including the objects with the lowest texture ("dog (porcelain)" and "dolphin") and the objects with the most notable view-dependent effects ("flower", "'penguin", and "rabbit"). Our method's performance in these cases is a strong argument for incorporating silhouette cues into multi-view pose estimation, especially when view-dependent effects must be considered. Of course, compared to traditional point features, silhouette constraints provide a less precise signal for pose estimation. The trade-off is that SfM requires a sufficient number of high-quality point feature correspondences. One possible line of future work is to combine feature correspondence search with silhouette-based pose estimation to derive precise pose estimates when point feature correspondences are scarce and noisy.

2.5 Additional qualitative results

To further test the potential for our method to work on diverse photocollections of difficult-to-reconstruction objects, we collected 22 images of a glass swan

sculpture placed in different environments (Fig. 6, top) and manually extracted masks for each of the images (Fig. 7). We then provided these 22 object masks as input to our network and obtained an estimate of the relative image poses for the entire collection (Fig. 6, bottom). Fig. 7 shows a breakdown of the recovered camera poses in easier-to-digest groups of three and four images. Qualitatively, we observe a similar outcome to the chair example, with predicted relative poses visually matching the apparent camera motion w.r.t the object between different images. Similar viewpoints are also predicted to be closer together (*e.g.*., the gray and green images in row 4 of Fig. 7).

2.6 Additional IOU visualizations

In Figures 8-15, we provide additional visualizations of our IOU analysis for 8 representative test instances from the RealScan dataset. As in the reported results in our main paper, each row shows the predicted parameter being evaluated with all other values fixed to be the ground truth. Our prediction is in red, and the ground truth is in green. The number in the top left corner of each image shows the IOU score for that image.

As mentioned in the main paper, we note that we obtain accurate rotation- and translation-only scores in these evaluations, but that our combined pose ($R + t$, the last row of each figure) has much lower accuracy, on average. This is primarily due to our method for aligning our output to the ground truth, which is done by estimating a similarity transform to align the camera rotations and 3D centers of projection. After the similarity transform, the visual hull of our estimated cameras may have a much different center of mass than that of the actual object; this leads to instances where the 2D reprojected object after alignment is quite far from the center of the image, despite the overall pose prediction error being low. A better alignment for the particular measurement of $R+t$ IOU is possible if we were to instead optimize the transform to maximize IOU.

As mentioned in the main paper, we do observe lower IOU for thin structures (see: Wooden Spoon, Chair 2, and Eyeglasses). Nevertheless, our rotation and translation estimates for such objects are still qualitatively quite accurate. We observe lower accuracy for rotationally symmetric objects (see: Porcelain Cup and Coffee Pot). This is to be expected, as rotational symmetries cannot entirely be resolved from silhouettes alone. A large number of redundant poses may also negatively impact the network's performance, as too few constraints on pose may remain.

2.7 Complete results

Finally, Tables 3 and 4 show complete pose-accuracy and IOU results for the 3D Warehouse and RealScan datasets, respectively, with the same evaluation criteria in the main paper.

3 Solving for R_0

As described in Section 3.2, for log-likelihood l_i^j of camera j, we can compute R_0 as

$$R_0^* = \arg\max_{R_0} \sum_{j=1}^{N} \sum_{i=1}^{3} l_i^j(R_0), \tag{1}$$

where

$$l_i^j(R_0) = \log\left(C_3(\kappa_j)\right) + \kappa_j \mathbf{r}_{ij}^T (R_0 \bar{\mathbf{r}}_{ij}).$$

First, since the maximization does not depend on κ_j:

$$\begin{aligned} \arg\max_{R_0} \sum_{j=1}^{N} \sum_{i=1}^{3} l_i^j(R_0) &= \arg\max_{R_0} \sum_{j=1}^{N} \sum_{i=1}^{3} \kappa_j \mathbf{r}_{ij}^T (R_0 \bar{\mathbf{r}}_{ij}) \\ &= \arg\max_{R_0} \sum_{j=1}^{N} \sum_{i=1}^{3} \kappa_j \mathbf{r}_{ij}^T (R_0 \bar{\mathbf{r}}_{ij}) - \kappa_j. \end{aligned} \tag{2}$$

Since $\|\mathbf{r}_{ij}\| = \|R_0 \bar{\mathbf{r}}_{ij}\| = 1$,

$$\mathbf{r}_{ij}^T (R_0 \bar{\mathbf{r}}_{ij}) - 1 = -\frac{1}{2} \left(\|\mathbf{r}_{ij}\|^2 + \|R_0 \bar{\mathbf{r}}_{ij}\|^2 - 2\mathbf{r}_{ij}^T (R_0 \bar{\mathbf{r}}_{ij}) \right).$$

Therefore,

$$\kappa_j \mathbf{r}_{ij}^T (R_0 \bar{\mathbf{r}}_{ij}) - \kappa_j = -\frac{1}{2} \kappa_j \left\| \mathbf{r}_{ij} - R_0 \bar{\mathbf{r}}_{ij} \right\|^2. \tag{3}$$

Incorporating Eq. (3) into Eq. (2):

$$\begin{aligned} &\arg\max_{R_0} \sum_{j=1}^{N} \sum_{i=1}^{3} -\frac{1}{2} \kappa_j \left\| \mathbf{r}_{ij} - R_0 \bar{\mathbf{r}}_{ij} \right\|^2 \\ &= \arg\min_{R_0} \sum_{j=1}^{N} \sum_{i=1}^{3} \frac{1}{2} \kappa_j \left\| \mathbf{r}_{ij} - R_0 \bar{\mathbf{r}}_{ij} \right\|^2. \end{aligned} \tag{4}$$

Eq. (4) forms an instance of Wahba's problem [7], which has a closed form solution. First a matrix B is defined as follows:

$$B = \sum_{j=1}^{N} \sum_{i=1}^{3} \kappa_j \mathbf{r_{ij}} \bar{\mathbf{r}}_{ij}^T \tag{5}$$

Let $B = UDV^T$ be the singular value decomposition for B. Then, the solution is given by

$$R_0^* = U \operatorname{diag}\left[1 \; 1 \; \det(U) \det(V)\right] V^T. \tag{6}$$

We note that:

$$B = \sum_{j=1}^{N} \kappa_j \sum_{i=1}^{3} \mathbf{r_{ij}} \bar{\mathbf{r}}_{ij}^T = \sum_{j=1}^{N} \kappa_j R_j^T \bar{R}_j. \tag{7}$$

Therefore, the solution is an averaging of the relative rotations between the ground truth camera rotations to the predicted camera rotations, weighted by κ_j and followed by a SVD projection.

In practice, to prevent repetitive SVD computations during training, we perform the weighted averaging with quaternions to obtain and initial value of $\mathbf{q}_0^*$, followed by a normalization by $\|\mathbf{q}_0^*\|$ to finally have $\|\mathbf{q}_0^*\| = 1$.

Since quaternions are defined only up to a sign, in order to verify that the relative quaternions in the weighted sum do not cancel each other, we set all the signs of the relative quaternions before averaging such that the sign of the dot product between each relative quaternion with the first relative quaternion is positive.

References

1. Dong, J., Soatto, S.: Domain-size pooling in local descriptors: Dsp-sift. In: Computer Vision and Pattern Recognition (CVPR). pp. 5097–5106 (2015) 7
2. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: International Conference on Computer Vision (ICCV). pp. 2961–2969 (2017) 2, 3
3. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Computer Vision and Pattern Recognition (CVPR). pp. 406–413. IEEE (2014) 4
4. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: GNeRF: GAN-based Neural Radiance Field without Posed Camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 5
5. Olson, E.: AprilTag: A robust and flexible visual fiducial system. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 3400–3407. IEEE (May 2011) 2
6. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 5
7. Wahba, G.: A least squares estimate of satellite attitude. SIAM review **7**(3), 409–409 (1965) 9
8. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019) 2
9. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33 (2020) 1, 4

Table 2. Camera pose accuracy for the Glass Figurines dataset. For each object, the first three rows show our method, and COLMAP and GNeRF results are reported as mean (median); failures are marked with a dash. "GNeRF (masks)" is the result with silhouette inputs.

Object	Example Images		R [°]	t_s [ratio]	t_d [°]
brown squirrel		Mean (Median)	3.74 (2.87)	0.03 (0.02)	2.59 (2.82)
		Top 5 (Oracle)	2.36 (2.05)	0.02 (0.02)	2.72 (3.29)
		Bottom 5 (Oracle)	5.12 (5.43)	0.03 (0.03)	2.46 (1.88)
		GNeRF	18.71 (18.45)	0.04 (0.02)	7.12 (6.69)
		GNeRF (masks)		–	
		COLMAP	0.19 (0.19)	0.00 (0.00)	0.09 (0.09)
dog (chrome)		Mean (Median)	6.92 (5.97)	0.04 (0.02)	4.88 (4.56)
		Top 5 (Oracle)	4.76 (3.24)	0.05 (0.05)	4.09 (2.80)
		Bottom 5 (Oracle)	9.08 (10.60)	0.02 (0.02)	5.68 (6.96)
		GNeRF	14.00 (13.48)	0.13 (0.13)	3.02 (3.04)
		GNeRF (masks)	18.43 (16.52)	0.12 (0.11)	2.19 (2.15)
		COLMAP	0.24 (0.23)	0.00 (0.00)	0.10 (0.12)
dog (porcelain)		Mean (Median)	8.14 (5.80)	0.02 (0.01)	7.32 (5.58)
		Top 5 (Oracle)	6.35 (4.66)	0.02 (0.02)	5.80 (4.16)
		Bottom 5 (Oracle)	9.93 (11.61)	0.02 (0.02)	8.84 (10.48)
		GNeRF	17.56 (18.38)	0.05 (0.04)	9.36 (11.06)
		GNeRF (masks)		–	
		COLMAP		–	
dolphin		Mean (Median)	5.10 (3.66)	0.03 (0.02)	3.01 (2.64)
		Top 5 (Oracle)	3.82 (3.15)	0.02 (0.03)	2.29 (1.59)
		Bottom 5 (Oracle)	6.38 (7.05)	0.05 (0.04)	3.73 (4.43)
		GNeRF	24.70 (24.33)	0.03 (0.03)	7.52 (7.91)
		GNeRF (masks)		–	
		COLMAP		–	
flamingo		Mean (Median)	6.13 (3.72)	0.05 (0.04)	2.02 (2.01)
		Top 5 (Oracle)	5.09 (3.02)	0.03 (0.05)	1.82 (2.11)
		Bottom 5 (Oracle)	7.18 (9.25)	0.07 (0.04)	2.22 (1.93)
		GNeRF	21.05 (21.20)	0.10 (0.09)	2.97 (2.87)
		GNeRF (masks)		–	
		COLMAP	0.66 (0.61)	0.01 (0.01)	0.47 (0.49)
flower		Mean (Median)	3.94 (3.57)	0.02 (0.02)	3.33 (3.31)
		Top 5 (Oracle)	3.31 (2.74)	0.02 (0.03)	2.64 (2.82)
		Bottom 5 (Oracle)	4.56 (5.13)	0.03 (0.02)	4.02 (3.84)
		GNeRF		–	
		GNeRF (masks)	17.27 (17.12)	0.06 (0.05)	16.79 (17.68)
		COLMAP		–	
frog		Mean (Median)	3.59 (3.17)	0.02 (0.03)	0.91 (0.72)
		Top 5 (Oracle)	2.39 (2.34)	0.03 (0.03)	0.66 (0.78)
		Bottom 5 (Oracle)	4.79 (4.85)	0.02 (0.02)	1.16 (1.04)
		GNeRF	21.66 (17.85)	0.02 (0.02)	2.40 (1.94)
		GNeRF (masks)	15.48 (16.31)	0.07 (0.01)	11.41 (10.51)
		COLMAP	0.40 (0.37)	0.00 (0.00)	0.25 (0.25)
parrot		Mean (Median)	21.16 (22.36)	0.03 (0.03)	1.46 (1.40)
		Top 5 (Oracle)	12.52 (12.52)	0.03 (0.03)	0.68 (0.68)
		Bottom 5 (Oracle)	29.81 (29.81)	0.03 (0.03)	2.24 (2.24)
		GNeRF	27.40 (26.83)	0.05 (0.05)	7.24 (7.40)
		GNeRF (masks)	11.92 (12.26)	0.04 (0.03)	8.98 (10.08)
		COLMAP	0.38 (0.35)	0.00 (0.00)	0.16 (0.17)
penguin		Mean (Median)	9.09 (8.70)	0.04 (0.03)	1.97 (1.57)
		Top 5 (Oracle)	6.18 (6.18)	0.04 (0.04)	1.47 (1.47)
		Bottom 5 (Oracle)	12.00 (12.00)	0.05 (0.05)	2.47 (2.47)
		GNeRF	20.04 (18.66)	0.03 (0.04)	3.38 (2.98)
		GNeRF (masks)		–	
		COLMAP		–	
rabbit		Mean (Median)	5.38 (5.23)	0.02 (0.01)	3.21 (3.46)
		Top 5 (Oracle)	4.06 (3.68)	0.02 (0.01)	2.13 (2.12)
		Bottom 5 (Oracle)	6.69 (7.07)	0.02 (0.02)	4.29 (4.29)
		GNeRF	25.32 (30.02)	0.03 (0.03)	6.38 (6.73)
		GNeRF (masks)	20.75 (22.29)	0.02 (0.02)	1.96 (1.90)
		COLMAP		–	
snake		Mean (Median)	11.29 (10.86)	0.02 (0.02)	1.90 (2.03)
		Top 5 (Oracle)	8.16 (7.82)	0.01 (0.02)	1.61 (1.76)
		Bottom 5 (Oracle)	14.42 (14.76)	0.03 (0.02)	2.19 (2.04)
		GNeRF	29.85 (32.25)	0.05 (0.05)	7.85 (8.33)
		GNeRF (masks)	24.59 (27.22)	0.05 (0.05)	4.24 (3.71)
		COLMAP	4.61 (3.39)	0.03 (0.02)	3.79 (3.39)

Fig. 6. Original images for the swan dataset (top) and two views of the camera poses estimated by our network given all 22 images as input. The border colors for the images correspond to the line colors for the rendered cameras. See Fig. 7 for more visualizations.

Fig. 7. Subsets of the swan results for easier viewing. Each row corresponds to one column of Fig. 6, with the same image border coloring. Manually segmented object masks are also included in the middle column. The cameras on the right are a subset of those in Fig. 6, *i.e.*, they have the exact same relative poses.

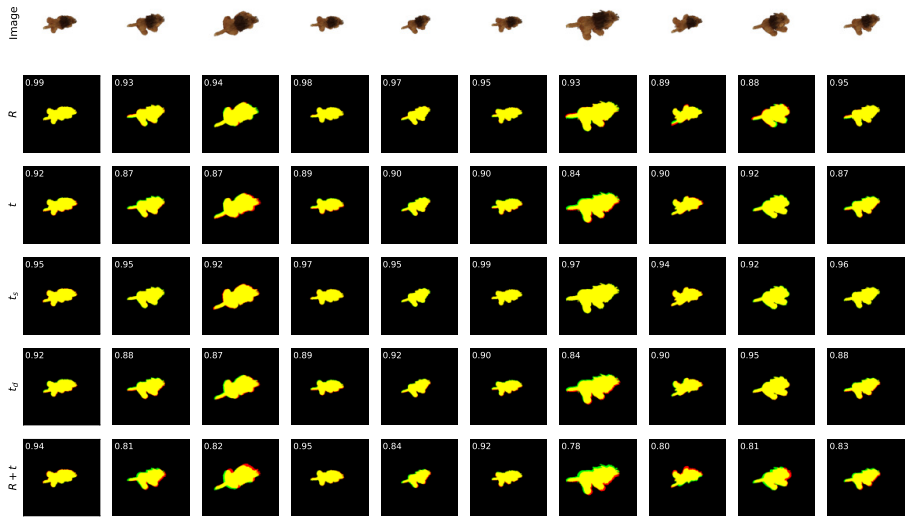

Fig. 8. IOU results for the RealScan Lion object.

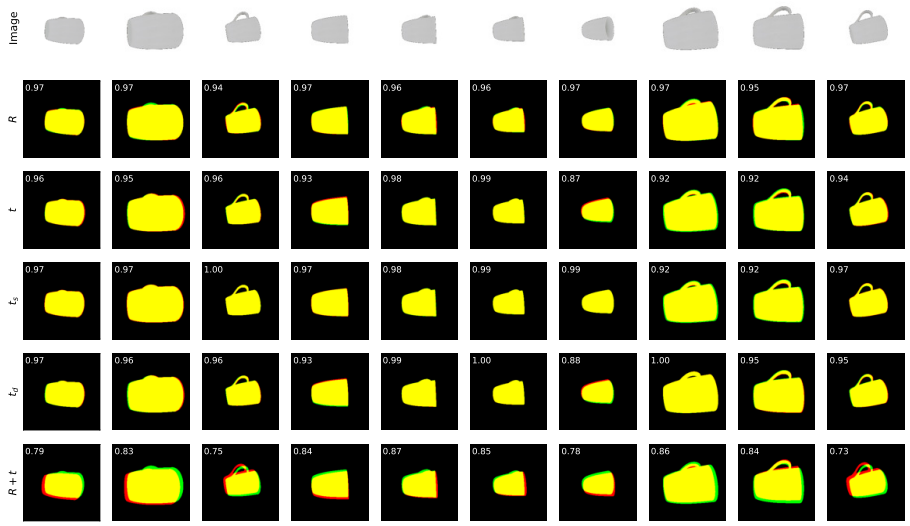

Fig. 9. IOU results for the RealScan Porcelain Cup object.

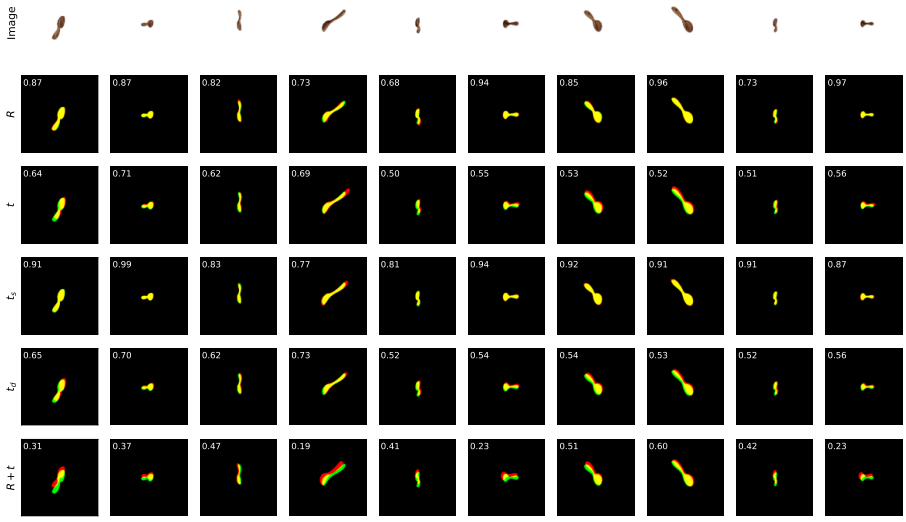

Fig. 10. IOU results for the RealScan Wooden Spoon object.

Fig. 11. IOU results for the RealScan Chessboard Knight object.

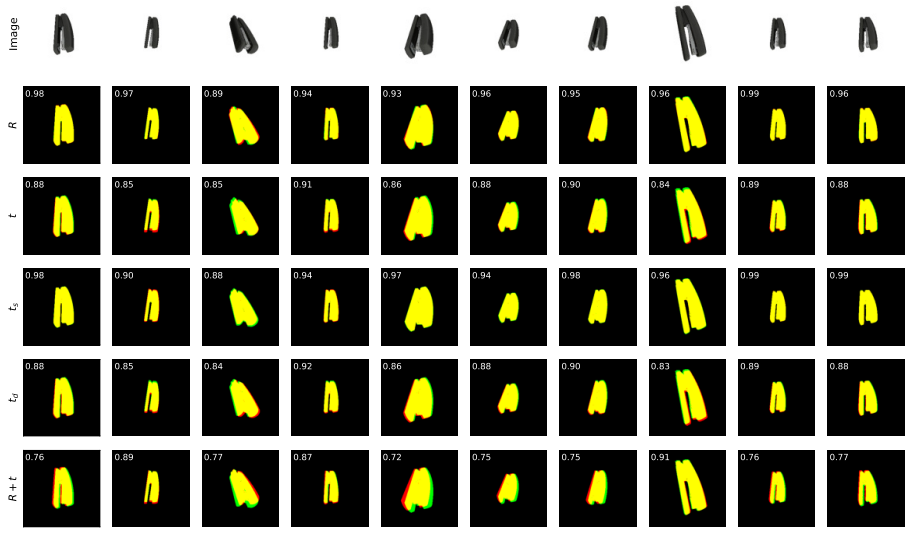

Fig. 12. IOU results for the RealScan Stapler object.

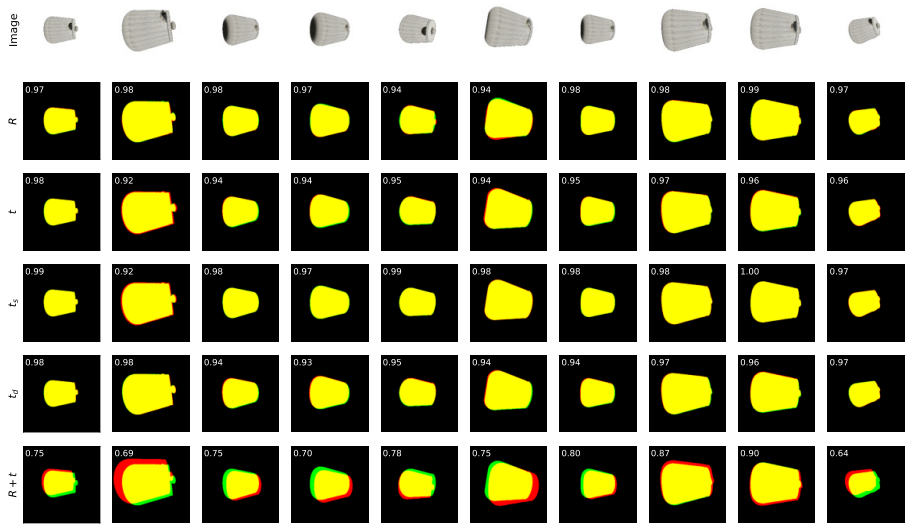

Fig. 13. IOU results for the RealScan Coffee Pot object.

Fig. 14. IOU results for the RealScan Chair 2 object.

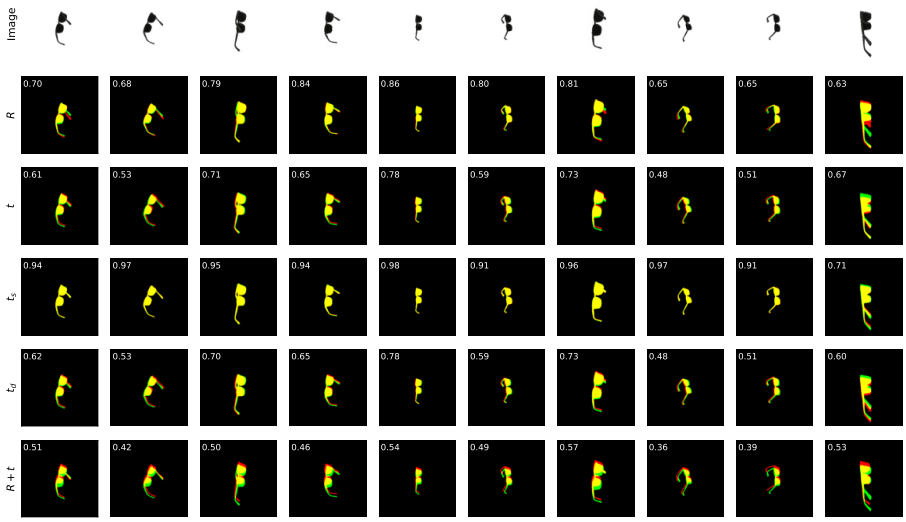

Fig. 15. IOU results for the RealScan Eyeglasses object.

Table 3. Camera pose accuracy and reprojection IOU for the 3D Warehouse dataset.

Class	Metrics	R [°]	t_s [ratio]	t_d [°]	R [IOU]	t [IOU]	t_s [IOU]	t_d [IOU]	$R+t$ [IOU]
Average over validation instances from training classes									
airplane	Mean (Med.)	4.58 (3.29)	0.03 (0.02)	1.80 (1.23)	0.83 (0.86)	0.66 (0.69)	0.92 (0.94)	0.67 (0.69)	0.44 (0.45)
	↑ 5 (Oracle)	3.47 (2.78)	0.03 (0.03)	1.57 (1.67)	0.85 (0.86)	0.67 (0.67)	0.93 (0.93)	0.68 (0.67)	0.48 (0.47)
	↓ 5 (Oracle)	5.68 (6.37)	0.03 (0.03)	2.03 (1.93)	0.80 (0.79)	0.65 (0.65)	0.91 (0.92)	0.66 (0.66)	0.41 (0.42)
bench	Mean (Med.)	4.58 (3.56)	0.03 (0.02)	1.76 (1.23)	0.80 (0.84)	0.67 (0.69)	0.88 (0.91)	0.68 (0.70)	0.48 (0.49)
	↑ 5 (Oracle)	3.58 (2.80)	0.02 (0.02)	1.54 (1.65)	0.82 (0.84)	0.68 (0.68)	0.89 (0.89)	0.69 (0.69)	0.52 (0.51)
	↓ 5 (Oracle)	5.58 (6.36)	0.03 (0.03)	1.98 (1.88)	0.78 (0.76)	0.66 (0.66)	0.86 (0.86)	0.67 (0.67)	0.45 (0.46)
bus	Mean (Med.)	4.41 (3.47)	0.03 (0.02)	1.65 (1.21)	0.93 (0.95)	0.82 (0.85)	0.95 (0.96)	0.83 (0.85)	0.62 (0.67)
	↑ 5 (Oracle)	3.33 (2.62)	0.02 (0.02)	1.44 (1.50)	0.94 (0.95)	0.83 (0.83)	0.96 (0.96)	0.84 (0.83)	0.66 (0.64)
	↓ 5 (Oracle)	5.49 (6.21)	0.03 (0.03)	1.85 (1.79)	0.92 (0.91)	0.81 (0.81)	0.95 (0.95)	0.82 (0.82)	0.58 (0.60)
camera	Mean (Med.)	8.60 (5.52)	0.04 (0.02)	3.69 (2.16)	0.89 (0.93)	0.83 (0.86)	0.94 (0.95)	0.84 (0.87)	0.52 (0.54)
	↑ 5 (Oracle)	6.81 (5.39)	0.03 (0.03)	2.95 (3.30)	0.90 (0.92)	0.84 (0.84)	0.94 (0.94)	0.85 (0.85)	0.57 (0.56)
	↓ 5 (Oracle)	10.40 (11.81)	0.04 (0.04)	4.44 (4.08)	0.88 (0.86)	0.83 (0.83)	0.93 (0.93)	0.84 (0.84)	0.48 (0.49)
faucet	Mean (Med.)	6.39 (4.68)	0.04 (0.02)	2.46 (1.40)	0.80 (0.85)	0.66 (0.71)	0.91 (0.94)	0.67 (0.72)	0.41 (0.40)
	↑ 5 (Oracle)	5.09 (3.96)	0.03 (0.03)	2.21 (2.36)	0.82 (0.84)	0.67 (0.67)	0.92 (0.92)	0.68 (0.68)	0.44 (0.43)
	↓ 5 (Oracle)	7.68 (8.82)	0.04 (0.04)	2.70 (2.56)	0.77 (0.75)	0.64 (0.65)	0.90 (0.90)	0.66 (0.66)	0.37 (0.38)
guitar	Mean (Med.)	5.05 (3.92)	0.04 (0.02)	1.95 (1.44)	0.81 (0.86)	0.55 (0.58)	0.92 (0.94)	0.56 (0.59)	0.33 (0.31)
	↑ 5 (Oracle)	3.91 (3.01)	0.03 (0.03)	1.78 (1.84)	0.83 (0.85)	0.57 (0.56)	0.93 (0.93)	0.58 (0.57)	0.37 (0.36)
	↓ 5 (Oracle)	6.20 (7.09)	0.04 (0.04)	2.12 (2.05)	0.79 (0.78)	0.53 (0.54)	0.91 (0.91)	0.53 (0.54)	0.30 (0.31)
knife	Mean (Med.)	7.92 (5.36)	0.05 (0.03)	2.72 (1.83)	0.75 (0.82)	0.49 (0.53)	0.92 (0.94)	0.49 (0.53)	0.25 (0.15)
	↑ 5 (Oracle)	6.34 (4.71)	0.04 (0.04)	2.75 (2.77)	0.77 (0.79)	0.51 (0.50)	0.93 (0.93)	0.52 (0.51)	0.27 (0.26)
	↓ 5 (Oracle)	9.50 (11.13)	0.05 (0.05)	2.70 (2.68)	0.73 (0.71)	0.47 (0.48)	0.91 (0.91)	0.47 (0.48)	0.24 (0.25)
microphone	Mean (Med.)	11.17 (7.97)	0.05 (0.03)	3.26 (1.48)	0.75 (0.81)	0.59 (0.68)	0.88 (0.94)	0.60 (0.70)	0.31 (0.20)
	↑ 5 (Oracle)	10.98 (6.96)	0.04 (0.04)	3.56 (2.97)	0.76 (0.76)	0.60 (0.60)	0.88 (0.89)	0.61 (0.61)	0.29 (0.30)
	↓ 5 (Oracle)	11.36 (15.39)	0.05 (0.05)	2.96 (3.54)	0.73 (0.73)	0.57 (0.58)	0.88 (0.87)	0.59 (0.59)	0.32 (0.31)
motorcycle/bike	Mean (Med.)	4.84 (3.96)	0.03 (0.02)	1.72 (1.29)	0.85 (0.88)	0.73 (0.76)	0.92 (0.94)	0.74 (0.76)	0.53 (0.56)
	↑ 5 (Oracle)	3.72 (2.89)	0.02 (0.02)	1.53 (1.63)	0.87 (0.89)	0.74 (0.74)	0.92 (0.92)	0.75 (0.75)	0.57 (0.56)
	↓ 5 (Oracle)	5.96 (6.79)	0.03 (0.03)	1.91 (1.81)	0.83 (0.82)	0.72 (0.73)	0.91 (0.91)	0.73 (0.74)	0.49 (0.51)
piano	Mean (Med.)	5.09 (3.71)	0.03 (0.02)	1.79 (1.09)	0.89 (0.93)	0.81 (0.84)	0.94 (0.96)	0.81 (0.85)	0.62 (0.70)
	↑ 5 (Oracle)	3.57 (2.93)	0.02 (0.02)	1.50 (1.71)	0.91 (0.92)	0.82 (0.82)	0.95 (0.95)	0.83 (0.83)	0.66 (0.65)
	↓ 5 (Oracle)	6.61 (7.24)	0.03 (0.03)	2.08 (1.87)	0.86 (0.85)	0.79 (0.79)	0.93 (0.93)	0.80 (0.80)	0.57 (0.59)
pistol/handgun	Mean (Med.)	5.95 (3.63)	0.04 (0.03)	2.18 (1.39)	0.85 (0.90)	0.70 (0.74)	0.94 (0.95)	0.71 (0.74)	0.45 (0.50)
	↑ 5 (Oracle)	4.73 (3.69)	0.03 (0.03)	1.97 (2.09)	0.88 (0.88)	0.71 (0.71)	0.95 (0.95)	0.71 (0.71)	0.50 (0.48)
	↓ 5 (Oracle)	7.16 (8.20)	0.04 (0.04)	2.39 (2.27)	0.83 (0.82)	0.69 (0.69)	0.93 (0.93)	0.70 (0.70)	0.40 (0.42)
pot	Mean (Med.)	8.68 (6.11)	0.04 (0.02)	3.09 (1.94)	0.87 (0.92)	0.81 (0.86)	0.92 (0.95)	0.82 (0.87)	0.56 (0.61)
	↑ 5 (Oracle)	7.15 (5.20)	0.03 (0.03)	2.86 (3.00)	0.89 (0.90)	0.82 (0.82)	0.93 (0.93)	0.83 (0.83)	0.58 (0.57)
	↓ 5 (Oracle)	10.21 (12.15)	0.04 (0.04)	3.32 (3.18)	0.86 (0.85)	0.81 (0.81)	0.92 (0.92)	0.81 (0.81)	0.54 (0.55)
rifle	Mean (Med.)	5.19 (3.90)	0.03 (0.02)	1.95 (1.34)	0.80 (0.84)	0.58 (0.61)	0.92 (0.94)	0.58 (0.61)	0.33 (0.32)
	↑ 5 (Oracle)	4.09 (3.16)	0.03 (0.03)	1.80 (1.81)	0.82 (0.84)	0.59 (0.59)	0.93 (0.93)	0.60 (0.59)	0.37 (0.36)
	↓ 5 (Oracle)	6.28 (7.21)	0.04 (0.04)	2.10 (2.09)	0.77 (0.76)	0.56 (0.56)	0.92 (0.91)	0.56 (0.57)	0.29 (0.30)
oven/stove	Mean (Med.)	6.15 (4.23)	0.03 (0.02)	2.46 (1.62)	0.90 (0.94)	0.82 (0.86)	0.94 (0.96)	0.82 (0.87)	0.57 (0.67)
	↑ 5 (Oracle)	4.81 (3.63)	0.03 (0.03)	2.43 (2.43)	0.92 (0.93)	0.83 (0.83)	0.95 (0.95)	0.84 (0.84)	0.60 (0.59)
	↓ 5 (Oracle)	7.48 (8.66)	0.03 (0.03)	2.49 (2.49)	0.89 (0.88)	0.80 (0.80)	0.94 (0.94)	0.80 (0.81)	0.53 (0.55)
vessel/watercraft	Mean (Med.)	7.42 (5.48)	0.04 (0.03)	2.86 (1.88)	0.85 (0.89)	0.72 (0.76)	0.93 (0.95)	0.73 (0.77)	0.42 (0.44)
	↑ 5 (Oracle)	6.03 (4.50)	0.04 (0.03)	2.58 (2.66)	0.87 (0.89)	0.74 (0.74)	0.93 (0.94)	0.75 (0.75)	0.46 (0.44)
	↓ 5 (Oracle)	8.81 (10.33)	0.04 (0.04)	3.13 (3.06)	0.84 (0.82)	0.71 (0.71)	0.92 (0.92)	0.72 (0.72)	0.38 (0.40)
Unseen test classes									
bathtub	Mean (Med.)	6.61 (4.69)	0.03 (0.02)	2.07 (1.33)	0.92 (0.95)	0.86 (0.88)	0.95 (0.96)	0.87 (0.89)	0.67 (0.74)
	↑ 5 (Oracle)	5.33 (4.13)	0.03 (0.03)	1.90 (1.92)	0.94 (0.94)	0.87 (0.87)	0.95 (0.96)	0.88 (0.88)	0.70 (0.69)
	↓ 5 (Oracle)	7.89 (9.09)	0.03 (0.04)	2.23 (2.22)	0.91 (0.90)	0.85 (0.85)	0.94 (0.94)	0.87 (0.87)	0.64 (0.65)
car	Mean (Med.)	6.12 (4.53)	0.03 (0.02)	2.27 (1.45)	0.92 (0.94)	0.84 (0.86)	0.94 (0.96)	0.85 (0.87)	0.62 (0.69)
	↑ 5 (Oracle)	4.92 (3.81)	0.03 (0.03)	1.99 (2.13)	0.93 (0.94)	0.85 (0.85)	0.95 (0.95)	0.86 (0.86)	0.66 (0.64)
	↓ 5 (Oracle)	7.32 (8.43)	0.03 (0.03)	2.55 (2.40)	0.90 (0.90)	0.83 (0.83)	0.94 (0.94)	0.84 (0.84)	0.58 (0.60)
chair	Mean (Med.)	6.79 (4.96)	0.03 (0.02)	2.67 (1.70)	0.83 (0.87)	0.74 (0.78)	0.91 (0.94)	0.75 (0.79)	0.49 (0.51)
	↑ 5 (Oracle)	5.38 (4.14)	0.03 (0.03)	2.38 (2.52)	0.85 (0.87)	0.76 (0.75)	0.92 (0.92)	0.76 (0.76)	0.53 (0.51)
	↓ 5 (Oracle)	8.19 (9.43)	0.04 (0.04)	2.97 (2.83)	0.80 (0.78)	0.73 (0.73)	0.90 (0.91)	0.74 (0.74)	0.45 (0.47)
lamp	Mean (Med.)	10.60 (7.26)	0.04 (0.03)	3.57 (2.24)	0.77 (0.83)	0.63 (0.68)	0.89 (0.93)	0.64 (0.69)	0.32 (0.27)
	↑ 5 (Oracle)	9.19 (6.57)	0.04 (0.04)	3.48 (3.51)	0.79 (0.80)	0.65 (0.65)	0.90 (0.90)	0.66 (0.66)	0.34 (0.33)
	↓ 5 (Oracle)	12.00 (14.62)	0.05 (0.05)	3.66 (3.63)	0.75 (0.73)	0.61 (0.62)	0.88 (0.88)	0.62 (0.63)	0.30 (0.31)
mailbox	Mean (Med.)	11.15 (5.13)	0.06 (0.03)	3.98 (2.12)	0.82 (0.88)	0.73 (0.78)	0.93 (0.95)	0.74 (0.78)	0.36 (0.30)
	↑ 5 (Oracle)	8.98 (7.56)	0.05 (0.05)	3.49 (3.42)	0.84 (0.86)	0.74 (0.75)	0.94 (0.93)	0.75 (0.76)	0.38 (0.38)
	↓ 5 (Oracle)	13.32 (14.73)	0.07 (0.07)	4.47 (4.54)	0.81 (0.79)	0.72 (0.71)	0.93 (0.93)	0.73 (0.71)	0.33 (0.33)

Table 4. Camera pose accuracy and reprojection IOU for the RealScan dataset.

Object	Metrics	R [°]	t_s [ratio]	t_d [°]	R [IOU]	t [IOU]	t_s [IOU]	t_d [IOU]	$R+t$ [IOU]
Lion	Mean (Med.)	10.00 (7.92)	0.04 (0.03)	4.34 (3.44)	0.87 (0.89)	0.87 (0.89)	0.95 (0.96)	0.88 (0.90)	0.41 (0.41)
	↑ 5 (Oracle)	8.06 (5.85)	0.04 (0.04)	3.79 (4.02)	0.88 (0.91)	0.88 (0.88)	0.96 (0.96)	0.89 (0.89)	0.44 (0.44)
	↓ 5 (Oracle)	11.94 (14.15)	0.04 (0.04)	4.89 (4.66)	0.86 (0.84)	0.87 (0.87)	0.94 (0.94)	0.88 (0.88)	0.38 (0.39)
Panther	Mean (Med.)	8.79 (6.91)	0.04 (0.03)	3.13 (2.42)	0.86 (0.89)	0.85 (0.87)	0.95 (0.96)	0.86 (0.88)	0.47 (0.45)
	↑ 5 (Oracle)	7.09 (5.20)	0.03 (0.04)	2.83 (3.16)	0.88 (0.90)	0.86 (0.86)	0.95 (0.95)	0.87 (0.87)	0.49 (0.47)
	↓ 5 (Oracle)	10.50 (12.39)	0.04 (0.04)	3.43 (3.10)	0.85 (0.83)	0.84 (0.84)	0.94 (0.94)	0.85 (0.85)	0.44 (0.46)
Cheetah	Mean (Med.)	8.78 (6.43)	0.04 (0.03)	3.37 (2.57)	0.87 (0.91)	0.85 (0.87)	0.95 (0.96)	0.86 (0.88)	0.45 (0.43)
	↑ 5 (Oracle)	7.14 (5.50)	0.03 (0.03)	2.80 (3.13)	0.89 (0.90)	0.87 (0.87)	0.95 (0.95)	0.87 (0.87)	0.51 (0.48)
	↓ 5 (Oracle)	10.42 (12.06)	0.04 (0.04)	3.94 (3.62)	0.86 (0.85)	0.84 (0.84)	0.94 (0.94)	0.85 (0.85)	0.40 (0.42)
Animal Doll	Mean (Med.)	5.69 (4.74)	0.03 (0.02)	2.36 (1.68)	0.85 (0.87)	0.85 (0.86)	0.94 (0.95)	0.86 (0.87)	0.50 (0.51)
	↑ 5 (Oracle)	4.92 (3.69)	0.02 (0.03)	2.08 (2.24)	0.86 (0.88)	0.85 (0.86)	0.95 (0.95)	0.86 (0.87)	0.54 (0.51)
	↓ 5 (Oracle)	6.45 (7.69)	0.03 (0.03)	2.65 (2.49)	0.84 (0.82)	0.85 (0.85)	0.94 (0.93)	0.87 (0.86)	0.45 (0.48)
Plastic Cup	Mean (Med.)	15.33 (12.96)	0.05 (0.04)	6.45 (5.41)	0.87 (0.90)	0.91 (0.92)	0.95 (0.96)	0.93 (0.94)	0.42 (0.42)
	↑ 5 (Oracle)	12.84 (9.70)	0.05 (0.05)	6.50 (6.75)	0.88 (0.90)	0.92 (0.92)	0.96 (0.95)	0.93 (0.93)	0.41 (0.42)
	↓ 5 (Oracle)	17.82 (20.96)	0.05 (0.05)	6.41 (6.16)	0.86 (0.84)	0.90 (0.90)	0.94 (0.94)	0.92 (0.92)	0.43 (0.41)
Porcelain Cup	Mean (Med.)	10.36 (7.78)	0.04 (0.03)	4.16 (3.28)	0.91 (0.93)	0.93 (0.94)	0.96 (0.97)	0.95 (0.96)	0.57 (0.58)
	↑ 5 (Oracle)	9.01 (6.33)	0.04 (0.04)	3.65 (4.35)	0.92 (0.94)	0.93 (0.94)	0.96 (0.96)	0.95 (0.95)	0.61 (0.57)
	↓ 5 (Oracle)	11.72 (14.40)	0.04 (0.04)	4.67 (3.97)	0.91 (0.89)	0.93 (0.92)	0.96 (0.96)	0.94 (0.94)	0.53 (0.57)
Wooden Spoon	Mean (Med.)	10.04 (8.18)	0.05 (0.03)	3.52 (2.57)	0.82 (0.86)	0.70 (0.74)	0.93 (0.95)	0.71 (0.75)	0.26 (0.19)
	↑ 5 (Oracle)	7.68 (5.76)	0.04 (0.04)	3.51 (3.65)	0.84 (0.85)	0.72 (0.73)	0.93 (0.94)	0.72 (0.73)	0.27 (0.27)
	↓ 5 (Oracle)	12.40 (14.32)	0.05 (0.05)	3.53 (3.40)	0.79 (0.79)	0.69 (0.68)	0.93 (0.92)	0.70 (0.69)	0.24 (0.24)
Wooden Fork	Mean (Med.)	12.04 (10.37)	0.06 (0.04)	4.29 (3.32)	0.79 (0.82)	0.62 (0.65)	0.91 (0.93)	0.63 (0.66)	0.19 (0.10)
	↑ 5 (Oracle)	10.37 (7.62)	0.06 (0.06)	4.46 (4.54)	0.80 (0.82)	0.63 (0.65)	0.91 (0.92)	0.63 (0.66)	0.19 (0.18)
	↓ 5 (Oracle)	13.72 (16.47)	0.06 (0.06)	4.12 (4.04)	0.78 (0.76)	0.61 (0.59)	0.91 (0.90)	0.62 (0.60)	0.19 (0.19)
Chess Knight	Mean (Med.)	10.29 (8.01)	0.04 (0.03)	4.84 (3.61)	0.90 (0.93)	0.88 (0.90)	0.94 (0.95)	0.89 (0.91)	0.44 (0.44)
	↑ 5 (Oracle)	8.12 (5.99)	0.04 (0.04)	4.97 (4.64)	0.92 (0.93)	0.89 (0.89)	0.95 (0.94)	0.90 (0.91)	0.46 (0.47)
	↓ 5 (Oracle)	12.47 (14.60)	0.05 (0.04)	4.70 (5.03)	0.88 (0.87)	0.87 (0.87)	0.94 (0.94)	0.89 (0.88)	0.42 (0.42)
Dry Erase Pen Clip	Mean (Med.)	6.72 (5.17)	0.03 (0.02)	1.96 (1.42)	0.85 (0.89)	0.86 (0.88)	0.94 (0.95)	0.87 (0.90)	0.62 (0.66)
	↑ 5 (Oracle)	5.39 (4.08)	0.03 (0.03)	1.96 (2.07)	0.88 (0.90)	0.87 (0.88)	0.94 (0.95)	0.89 (0.89)	0.64 (0.64)
	↓ 5 (Oracle)	8.05 (9.37)	0.03 (0.03)	1.96 (1.86)	0.82 (0.81)	0.85 (0.85)	0.93 (0.93)	0.86 (0.86)	0.60 (0.61)
Credit Card Wallet	Mean (Med.)	9.13 (7.11)	0.03 (0.02)	3.05 (2.36)	0.81 (0.86)	0.85 (0.88)	0.93 (0.94)	0.87 (0.90)	0.52 (0.51)
	↑ 5 (Oracle)	6.82 (5.43)	0.03 (0.03)	3.10 (3.11)	0.84 (0.86)	0.86 (0.86)	0.93 (0.93)	0.87 (0.88)	0.53 (0.53)
	↓ 5 (Oracle)	11.45 (12.84)	0.03 (0.03)	3.01 (2.99)	0.78 (0.76)	0.85 (0.84)	0.92 (0.92)	0.87 (0.86)	0.50 (0.51)
Gaming Keyboard	Mean (Med.)	8.96 (6.47)	0.04 (0.03)	3.16 (2.28)	0.81 (0.85)	0.82 (0.84)	0.92 (0.94)	0.83 (0.86)	0.46 (0.50)
	↑ 5 (Oracle)	6.98 (5.24)	0.03 (0.04)	3.28 (3.38)	0.84 (0.86)	0.84 (0.83)	0.93 (0.93)	0.85 (0.85)	0.48 (0.47)
	↓ 5 (Oracle)	10.94 (12.68)	0.04 (0.04)	3.03 (2.93)	0.78 (0.76)	0.80 (0.80)	0.92 (0.92)	0.81 (0.82)	0.43 (0.44)
Gaming Mouse	Mean (Med.)	13.70 (10.82)	0.05 (0.04)	5.95 (4.98)	0.89 (0.91)	0.89 (0.91)	0.94 (0.96)	0.90 (0.93)	0.39 (0.38)
	↑ 5 (Oracle)	10.53 (7.93)	0.05 (0.05)	6.17 (6.24)	0.90 (0.92)	0.89 (0.90)	0.94 (0.95)	0.91 (0.92)	0.39 (0.38)
	↓ 5 (Oracle)	16.87 (19.47)	0.05 (0.05)	5.72 (5.65)	0.87 (0.86)	0.88 (0.87)	0.94 (0.94)	0.89 (0.89)	0.40 (0.40)
Wireless Mouse	Mean (Med.)	17.86 (14.42)	0.06 (0.04)	6.99 (5.69)	0.87 (0.90)	0.89 (0.90)	0.94 (0.95)	0.91 (0.92)	0.36 (0.35)
	↑ 5 (Oracle)	16.08 (11.71)	0.06 (0.06)	7.12 (7.26)	0.88 (0.90)	0.90 (0.90)	0.94 (0.94)	0.92 (0.92)	0.35 (0.36)
	↓ 5 (Oracle)	19.64 (24.01)	0.06 (0.05)	6.86 (6.72)	0.87 (0.85)	0.89 (0.88)	0.94 (0.94)	0.90 (0.90)	0.37 (0.36)
Toy Cat	Mean (Med.)	10.90 (7.94)	0.05 (0.03)	4.47 (2.72)	0.92 (0.94)	0.90 (0.92)	0.95 (0.96)	0.91 (0.93)	0.54 (0.59)
	↑ 5 (Oracle)	8.58 (6.67)	0.04 (0.04)	4.06 (4.25)	0.93 (0.94)	0.91 (0.91)	0.95 (0.95)	0.92 (0.93)	0.55 (0.55)
	↓ 5 (Oracle)	13.23 (15.14)	0.05 (0.05)	4.88 (4.70)	0.91 (0.90)	0.89 (0.89)	0.95 (0.95)	0.90 (0.90)	0.52 (0.52)
Toy Bunny	Mean (Med.)	8.29 (5.92)	0.03 (0.02)	2.92 (1.95)	0.94 (0.95)	0.92 (0.93)	0.96 (0.97)	0.93 (0.94)	0.62 (0.68)
	↑ 5 (Oracle)	6.06 (4.51)	0.03 (0.03)	2.78 (2.80)	0.95 (0.96)	0.92 (0.93)	0.96 (0.96)	0.93 (0.94)	0.65 (0.65)
	↓ 5 (Oracle)	10.52 (12.08)	0.03 (0.03)	3.06 (3.04)	0.93 (0.92)	0.91 (0.91)	0.95 (0.95)	0.93 (0.92)	0.60 (0.60)
VR Headset	Mean (Med.)	13.43 (9.94)	0.05 (0.04)	4.95 (4.05)	0.78 (0.79)	0.84 (0.85)	0.92 (0.93)	0.86 (0.87)	0.33 (0.30)
	↑ 5 (Oracle)	10.74 (8.28)	0.05 (0.05)	4.70 (5.00)	0.81 (0.83)	0.85 (0.85)	0.92 (0.93)	0.88 (0.87)	0.35 (0.34)
	↓ 5 (Oracle)	16.13 (18.59)	0.04 (0.05)	5.20 (4.90)	0.75 (0.73)	0.82 (0.82)	0.92 (0.92)	0.83 (0.84)	0.31 (0.33)
VR Controller Right	Mean (Med.)	14.13 (10.24)	0.06 (0.03)	5.16 (4.11)	0.74 (0.83)	0.78 (0.81)	0.92 (0.94)	0.79 (0.82)	0.21 (0.17)
	↑ 5 (Oracle)	12.47 (9.24)	0.05 (0.06)	4.85 (5.08)	0.76 (0.78)	0.79 (0.80)	0.93 (0.92)	0.80 (0.81)	0.24 (0.22)
	↓ 5 (Oracle)	15.78 (19.01)	0.06 (0.05)	5.48 (5.24)	0.72 (0.70)	0.78 (0.76)	0.92 (0.92)	0.79 (0.77)	0.19 (0.21)
VR Controller Left	Mean (Med.)	13.14 (8.98)	0.05 (0.03)	4.27 (2.76)	0.78 (0.85)	0.78 (0.81)	0.91 (0.93)	0.79 (0.82)	0.29 (0.25)
	↑ 5 (Oracle)	11.06 (8.30)	0.05 (0.05)	3.90 (3.91)	0.81 (0.83)	0.79 (0.79)	0.92 (0.93)	0.80 (0.80)	0.32 (0.31)
	↓ 5 (Oracle)	15.22 (17.97)	0.06 (0.06)	4.64 (4.63)	0.76 (0.74)	0.77 (0.77)	0.91 (0.90)	0.78 (0.79)	0.26 (0.27)
Stapler	Mean (Med.)	8.88 (4.68)	0.03 (0.02)	2.77 (1.81)	0.90 (0.92)	0.89 (0.90)	0.96 (0.96)	0.89 (0.91)	0.55 (0.60)
	↑ 5 (Oracle)	7.44 (5.50)	0.03 (0.03)	2.43 (2.70)	0.91 (0.92)	0.89 (0.90)	0.96 (0.96)	0.90 (0.90)	0.57 (0.56)
	↓ 5 (Oracle)	10.31 (12.25)	0.03 (0.03)	3.10 (2.83)	0.89 (0.89)	0.88 (0.88)	0.95 (0.95)	0.89 (0.89)	0.53 (0.54)
Hole Punch 1	Mean (Med.)	11.79 (9.21)	0.05 (0.03)	4.45 (2.58)	0.86 (0.88)	0.89 (0.90)	0.94 (0.95)	0.90 (0.91)	0.53 (0.56)
	↑ 5 (Oracle)	9.35 (6.65)	0.05 (0.05)	4.40 (4.78)	0.88 (0.91)	0.89 (0.89)	0.94 (0.95)	0.90 (0.91)	0.55 (0.54)
	↓ 5 (Oracle)	14.23 (16.93)	0.05 (0.05)	4.50 (4.12)	0.84 (0.82)	0.88 (0.88)	0.93 (0.93)	0.90 (0.90)	0.51 (0.52)
Hole Punch 2	Mean (Med.)	8.94 (6.95)	0.03 (0.02)	2.69 (2.03)	0.87 (0.89)	0.84 (0.85)	0.94 (0.96)	0.85 (0.87)	0.54 (0.58)
	↑ 5 (Oracle)	7.40 (5.23)	0.03 (0.03)	2.58 (2.62)	0.89 (0.91)	0.85 (0.86)	0.94 (0.95)	0.87 (0.87)	0.56 (0.53)
	↓ 5 (Oracle)	10.48 (12.65)	0.03 (0.03)	2.80 (2.76)	0.85 (0.83)	0.82 (0.82)	0.94 (0.94)	0.84 (0.83)	0.53 (0.55)
Coffee Pot	Mean (Med.)	16.19 (13.73)	0.07 (0.05)	6.97 (5.92)	0.90 (0.93)	0.93 (0.94)	0.96 (0.97)	0.95 (0.95)	0.43 (0.42)
	↑ 5 (Oracle)	14.21 (10.37)	0.06 (0.07)	6.88 (6.98)	0.91 (0.92)	0.93 (0.93)	0.96 (0.96)	0.95 (0.95)	0.43 (0.43)
	↓ 5 (Oracle)	18.16 (22.00)	0.07 (0.07)	7.07 (6.97)	0.89 (0.88)	0.93 (0.93)	0.95 (0.96)	0.94 (0.94)	0.42 (0.42)
Chair 1	Mean (Med.)	14.17 (9.79)	0.06 (0.04)	5.05 (3.60)	0.71 (0.70)	0.77 (0.78)	0.88 (0.91)	0.79 (0.81)	0.37 (0.36)
	↑ 5 (Oracle)	11.72 (8.98)	0.06 (0.05)	4.70 (4.68)	0.73 (0.76)	0.78 (0.78)	0.89 (0.89)	0.80 (0.80)	0.40 (0.40)
	↓ 5 (Oracle)	16.62 (19.36)	0.06 (0.06)	5.40 (5.42)	0.68 (0.65)	0.76 (0.76)	0.88 (0.87)	0.78 (0.78)	0.35 (0.35)
Chair 2	Mean (Med.)	10.56 (7.63)	0.05 (0.04)	4.59 (2.98)	0.63 (0.65)	0.70 (0.73)	0.90 (0.92)	0.71 (0.74)	0.22 (0.18)
	↑ 5 (Oracle)	8.63 (6.51)	0.05 (0.05)	3.85 (4.07)	0.68 (0.71)	0.70 (0.71)	0.91 (0.90)	0.72 (0.73)	0.24 (0.23)
	↓ 5 (Oracle)	12.49 (14.61)	0.06 (0.06)	5.32 (5.11)	0.59 (0.56)	0.69 (0.68)	0.88 (0.89)	0.71 (0.70)	0.20 (0.20)
Chair 3	Mean (Med.)	9.49 (6.60)	0.05 (0.03)	3.97 (2.51)	0.71 (0.76)	0.73 (0.77)	0.90 (0.92)	0.74 (0.78)	0.35 (0.30)
	↑ 5 (Oracle)	7.43 (5.72)	0.04 (0.04)	3.47 (3.59)	0.74 (0.77)	0.74 (0.74)	0.92 (0.92)	0.75 (0.75)	0.38 (0.37)
	↓ 5 (Oracle)	11.55 (13.26)	0.05 (0.05)	4.48 (4.36)	0.67 (0.64)	0.72 (0.72)	0.89 (0.89)	0.73 (0.73)	0.31 (0.32)
Vase Tree	Mean (Med.)	14.60 (12.07)	0.06 (0.04)	5.39 (4.02)	0.90 (0.92)	0.91 (0.92)	0.95 (0.96)	0.93 (0.94)	0.47 (0.49)
	↑ 5 (Oracle)	12.00 (9.15)	0.06 (0.06)	4.86 (5.35)	0.92 (0.93)	0.92 (0.92)	0.95 (0.95)	0.93 (0.93)	0.50 (0.48)
	↓ 5 (Oracle)	17.21 (20.06)	0.07 (0.07)	5.92 (5.43)	0.89 (0.88)	0.90 (0.90)	0.94 (0.94)	0.92 (0.92)	0.45 (0.46)
Eyeglasses	Mean (Med.)	9.15 (7.57)	0.05 (0.03)	4.39 (3.00)	0.62 (0.64)	0.54 (0.56)	0.82 (0.86)	0.57 (0.58)	0.18 (0.11)
	↑ 5 (Oracle)	7.45 (5.95)	0.04 (0.04)	3.91 (4.23)	0.64 (0.69)	0.57 (0.57)	0.84 (0.84)	0.59 (0.59)	0.20 (0.20)
	↓ 5 (Oracle)	10.85 (12.35)	0.05 (0.06)	4.87 (4.56)	0.59 (0.55)	0.52 (0.52)	0.81 (0.81)	0.55 (0.54)	0.16 (0.17)
Dancer Sculpture	Mean (Med.)	17.51 (10.78)	0.04 (0.03)	3.90 (2.88)	0.78 (0.84)	0.85 (0.87)	0.95 (0.96)	0.86 (0.88)	0.35 (0.33)
	↑ 5 (Oracle)	14.31 (11.47)	0.04 (0.04)	3.72 (3.96)	0.80 (0.81)	0.86 (0.87)	0.95 (0.95)	0.87 (0.88)	0.37 (0.35)
	↓ 5 (Oracle)	20.71 (23.56)	0.04 (0.04)	4.07 (3.83)	0.76 (0.74)	0.84 (0.84)	0.94 (0.94)	0.86 (0.85)	0.33 (0.35)