Supplementary Material - One2Any: One-Reference 6D Pose Estimation for Any Object

Mengya Liu[1] Siyuan Li[1] Ajad Chhatkuli[2] Prune Truong[3]
Luc Van Gool[1,2] Federico Tombari[3,4]

[1] ETH Zurich [2] INSAIT, Sofia University "St. Kliment Ohridski" [3] Google [4] TUM

I Overview

In this document, we supplement the details left out from the main text. We first explain additional training details about our method in Section II. We then present more experimental results in Section III. In order to show that our method can be adapted to multi-view setting, we present results on multi-view data in Section IV.

II Training Details

We describe the training process of our model, including the preprocessing of the data, and the network architecture details.

II.1 Preprocessing of the Data

We begin by masking each object in the image, cropping it so the object is centered, and resizing it to a resolution of 192×192.

Scale matrix S. The scale matrix $\mathbf{S}$ is obtained from the reference image A. The computation of the scale matrix $\mathbf{S}$ is detailed here. Let d^{A_D} represent the depth values in A_D, and $\begin{bmatrix} u \\ v \end{bmatrix}^{A_D}$ denote the corresponding pixel positions of the object, pre-filtered using the mask A_M. The corresponding 3D point in the camera coordinate space for each pixel in the image A is given as $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3$. Using this, we construct a point cloud$P_A \in \mathbb{R}^{N\times 3}$ where N is the number of points, computed as $N = \sum_u \sum_v Q_M(u, v) = 1$.

To normalize the points into ROC space, we apply the scale matrix $\mathbf{S}$, as defined in Equation 1. For each instance in the reference image, we establish a Reference Object Coordinate (ROC).

$$w = \max(\max(\begin{bmatrix} x \\ y \\ z \end{bmatrix}^{P_A}) - \min(\begin{bmatrix} x \\ y \\ z \end{bmatrix}^{P_A})) \quad (1)$$

$$c = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix}^{P_A} = (\max(\begin{bmatrix} x \\ y \\ z \end{bmatrix}^{P_A}) + \min(\begin{bmatrix} x \\ y \\ z \end{bmatrix}^{P_A}))/2 \quad (2)$$

$$Y_A = (Y_A - c)/(w \times 1.1) \quad (3)$$

By determining the size scalar w and the center c of the point cloud P_A, we translate P_A to the origin with the offset c and scale it with $w \times 1.1$ to ensure all points lie within $[-0.5, 0.5]$. The result is the normalized 3D point cloud P_A in ROC space, which is then mapped into an RGB image Y_A, referred to as the ROC map, where the RGB channels encode the 3D ROC positions.

In this process, the scale and shift transformation matrix $\mathbf{S}$ is also computed and recorded for further alignment on the predicted ROC map from the test image.

$$s = 1/(w \times 1.1) \quad (4)$$

$$t = -1 \times c/s \quad (5)$$

$$\mathbf{S} = \begin{bmatrix} s & 0 & 0 & t_x \\ 0 & s & 0 & t_y \\ 0 & 0 & s & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

II.2 Network Architecture

Diffusion Model [20]. The text-to-image diffusion model is a generative framework designed to create high-quality images from textual descriptions. Its architecture consists of three key modules: an encoder and a decoder, based on VQ-VAE [5], which map features between the image and latent spaces, and a U-Net module that performs denoising within the latent space. The diffusion proves its strong capability in image generation, and it can also cooperate with other generation tasks like depth generation [17]. Here, we extend the

diffusion model to predict ROC maps. To improve time efficiency, we employ a feedforward stable diffusion model instead of the traditional iterative denoising process.

Following [2], we utilize the stable diffusion model [20], pre-trained on Laion-5b [21]. We freeze the VQVAE module to encode the input query image Q_I, mapping it to the latent space as the feature map $\mathcal{F}^Q$. The U-Net is fine-tuned using our data pairs. Reference RGBD images serve as conditional inputs, processed through the Reference Object Encoder (ROE) and passed into the U-Net as feature maps $\mathcal{F}^A$. The two feature embeddings $\mathcal{F}^Q$ and $\mathcal{F}^A$, are integrated via the cross-attention modules in U-Net, producing the semantic and spatially aligned feature map $\mathcal{F}^{Q2A}$. Since ROC maps differ significantly from typical RGB images, we train the decoder from scratch. The decoder progressively refines $\mathcal{F}^{Q2A}$ to reconstruct a high-resolution output aligned with the original image dimensions. The individual modules are detailed below.

VQVAE. We adopt the pre-trained VQVAE module from [20] and freeze it during training. Input images are normalized for better feature extraction.

U-Net. The U-Net is initialized with pre-trained weights from [20]. During the feedforward pass, no noise is added, allowing the U-Net to directly output meaningful feature embeddings. The ROE features serve as conditional inputs to the U-Net, interacting with the latent feature $\mathcal{F}^Q$ through cross-attention layers. The U-Net gradually learns to process the feature maps from ROPE and the test image.

ROE. The Reference Object Encoder (ROE), denoted as $f(A;\theta_A)$ extracts latent encodings from a channel-wise concatenation of the reference RGB image, ROC map, and object mask. This encoding captures both texture and geometric information. The ROE processes these inputs through three convolutional layers with batch normalization and ReLU activation. A residual connection block [9] further refines the features embeddings. The resulting feature maps are tokenized into patches with positional embeddings [1], effectively guiding ROC map generation and maintaining fidelity to reference data, even under occlusion.

Decoder. The decoder consists of five convolutional layers with residual connections [9]. Each layer includes a bilinear upsampling module to produce the final ROC map $\hat{Y}_Q$. The decoder is initialized using Kaiming Initialization [8].

III Experimental Results

In this section, we provide a detailed analysis of the experimental results and investigate the impact of training data scale on the performance of our model.

III.1 Full Test Results on the YCB-Video Dataset

We present the detailed test results on the YCB-Video [29] test set in Table A, including performance metrics for each object. For all one-view methods, only the first test image is used as the reference image. In contrast, multi-view-based methods such as FS6D [11], Predator [15], and LoFTR [23] are fine-tuned on the test set, following the protocol outlined in FS6D [11]. Specifically, FS6D divides the test objects into three groups, training on two groups while testing on the remaining one. Our method demonstrates superior performance compared to other one-view-based approaches, particularly for the ADD metric, where it significantly outperforms all competing methods. Our method even surpasses some multi-view approaches that employ fine-tuning on the test data.

III.2 Full Tracking Results on YCB-Video Sequences

We present the object tracking results on the full YCB-Video Sequences [29] , using the first frame as the reference, in Table B. We compare our approach with other CAD model-based novel object pose tracking methods, including [28], RGF [16], ICG [22], and FoundationPose [27]. Our method achieves the best performance among one-view-based tracking methods and performs comparably to CAD model-based approaches.

III.3 Robustness Analysis on the Selection of Reference Images

To further demonstrate the robustness of One2Any on the selection of different reference images, we additionally measure the standard deviation of the method when dealing with different reference images. We conducted experiments following Oryon [3], evaluating 2000 randomly selected reference-query image pairs from the Real275 [26] and Toyota-Light [14] datasets respectively. These datasets contain various combinations of the same objects with different reference-query pairs. As shown in Table 1 (main paper), our method significantly outperforms existing approaches when handling different reference images. To further analyze robustness, we computed the standard deviation (std) for each object and averaged them (see in Table C). Our method achieves a lower std, demonstrating its robustness to variations in reference images.

III.4 Performance on Occluded Scenes

For scenes with occlusions—whether in the reference image or the test image—our method demonstrates strong robustness in handling occluded scenarios, where other methods often fail. Detailed results are presented in Figure A.

Table A. Performance on occluded YCB-Video [29] dataset. We compare with point cloud registration methods, multi-view methods, and one-view methods. Predator [15], LoFTR [23] and FS6D [11] are fine-tuned on the YCB-Video dataset. We evaluate ADD-S AUC and ADD AUC metrics. Results of multi-view methods are adopted from [27]. For one-view methods, we provide the first image in the test set as the reference. The best performance among multi-view methods and one-view methods are highlighted in bold.

Methods Ref. Images	**PREDATOR** [15] 16		**LoFTR**[23] 16		**FS6D** [11] 16		**FoundationPose** [27] 16 - CAD		**FoundationPose**[27] 1 - CAD		**Oryon** [3] 1		**NOPE** [18] 1 + GT trans		**One2Any(Ours)** 1	
metrics	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD
002_master_chef_can	73.0	17.4	87.2	50.6	92.6	36.8	**96.9**	**91.3**	87.3	73.3	12.2	8.7	96.8	17.8	**94.9**	**84.3**
003_cracker_box	41.7	8.3	71.8	25.5	83.9	24.5	**97.5**	**96.2**	**92.0**	72.2	6.5	3.3	83.0	2.8	91.1	**83.3**
004_sugar_box	53.7	15.3	63.9	13.4	95.1	43.9	**97.5**	**87.2**	88.2	87.1	5.3	3.8	86.5	22.3	**95.3**	**88.0**
005_tomato_soup_can	81.2	44.4	77.1	52.9	93.0	54.2	**97.6**	**93.3**	95.2	**92.3**	10.9	5.9	**95.9**	48.4	93.6	80.9
006_mustard_bottle	35.5	5.0	84.5	59.0	97.0	71.1	**98.4**	**97.3**	88.4	76.6	5.5	4.7	91.3	42.7	**93.8**	**87.6**
007_tuna_fish_can	78.2	34.2	72.6	55.7	94.5	53.9	**97.7**	**73.7**	90.5	76.9	23.2	15.6	**97.0**	33.3	**95.9**	**90.0**
008_pudding_box	73.5	24.2	86.5	68.1	94.9	79.6	**98.5**	**97.0**	91.7	77.8	1.3	1.3	84.4	20.9	**96.1**	**93.3**
009_gelatin_box	81.4	37.5	71.6	45.2	98.3	32.1	**98.5**	**97.3**	92.7	87.7	1.3	1.3	87.3	35.3	**97.7**	**96.1**
010_potted_meat_can	62.0	20.9	67.4	45.1	87.6	54.9	**96.6**	**82.3**	90.3	**83.5**	38.9	19.2	**92.8**	31.9	86.3	72.5
011_banana	57.7	9.9	24.2	1.6	94.0	69.1	**98.1**	**95.4**	90.3	76.3	4.9	4.1	61.3	11.4	**95.0**	**85.7**
019_pitcher_base	83.7	18.1	58.7	22.3	91.1	40.4	**97.9**	**96.6**	92.1	86.9	41.2	14.9	88.9	6.1	**93.6**	**87.7**
021_bleach_cleanser	88.3	48.1	36.9	16.7	89.4	44.1	**97.4**	**93.3**	90.8	**85.5**	5.0	2.8	89.6	32.3	**93.0**	84.6
024_bowl	73.2	17.4	32.7	1.4	74.7	0.9	**94.9**	**89.7**	87.5	43.6	3.8	3.6	**93.2**	6.7	92.1	**65.1**
025_mug	84.8	29.5	47.3	23.6	86.5	39.2	**96.2**	**75.8**	91.0	74.1	2.6	2.6	92.5	31.6	**95.5**	**82.9**
035_power_drill	60.6	12.3	18.8	1.3	73.0	19.8	**98.0**	**96.3**	**97.0**	**96.8**	22.1	13.1	56.0	0.0	92.4	84.6
036_wood_block	70.5	10.0	49.9	1.4	94.7	27.9	**97.4**	**94.7**	67.1	19.9	26.3	10.5	77.1	0.0	**92.7**	**85.0**
037_scissors	75.5	25.0	32.3	14.6	74.2	27.7	**97.8**	**95.5**	**97.4**	**94.7**	9.5	5.8	75.5	0.0	92.6	84.7
040_large_marker	81.8	38.9	20.7	8.4	97.4	74.2	**98.6**	**96.5**	**92.7**	90.4	9.1	8.0	79.6	39.3	**96.5**	**91.0**
051_large_clamp	83.0	34.4	24.1	11.2	82.7	34.7	**96.9**	**92.7**	87.4	68.9	22.4	8.2	**100.0**	**100.0**	92.9	84.2
052_extra_large_clamp	72.9	24.1	15.0	1.8	65.7	10.1	**97.6**	**94.1**	**90.5**	43.7	16.9	8.1	82.6	0.0	89.5	**65.8**
061_foam_brick	79.2	35.5	59.4	31.4	95.7	45.8	**98.1**	**93.4**	**98.7**	90.9	11.3	9.1	95.2	43.5	97.6	**95.9**
mean	71.0	24.3	52.5	26.2	88.4	42.1	**97.4**	**91.5**	90.4	76.1	13.3	7.4	86.0	25.1	**93.7**	**84.4**

Table B. Performance on pose tracking task. We compared with CAD model based tracking methods on YCB-V full video sequences. For our method, we still only give the first frame as a reference, and we keep the reference during tracking.

Method Method	**Wuthrich**[28] CAD		**RGF**[16] CAD		**ICG**[22] CAD		**FoundationPose**[27] 16 frames-CAD		**FoundationPose**[27] 1^{st} frame-CAD		**One2Any(Ours)** 1^{st} frame	
metrics	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
002_master_chef_can	55.6	90.7	46.2	90.2	66.4	89.7	**91.2**	**96.9**	38.1	83.3	**83.8**	**94.8**
003_cracker_box	96.4	97.2	57.0	72.3	82.4	92.1	**96.2**	**97.5**	78.3	**94.0**	**83.0**	91.3
004_sugar_box	97.1	97.9	50.4	72.7	**96.1**	**98.4**	94.5	97.4	40.0	78.7	**88.7**	**95.3**
005_tomato_soup_can	64.7	89.5	72.4	91.6	73.2	97.3	**94.3**	**97.9**	14.0	49.3	**87.1**	**95.5**
006_mustard_bottle	97.1	98.0	87.7	98.2	96.2	98.4	**97.3**	**98.5**	24.8	58.8	**87.7**	**93.8**
007_tuna_fish_can	69.1	93.3	28.7	52.9	73.2	95.8	**84.0**	**97.8**	75.3	**97.4**	**89.5**	95.9
008_pudding_box	96.8	97.9	12.7	18.0	73.8	88.9	**96.9**	**98.5**	**96.9**	**98.3**	93.5	96.3
009_gelatin_box	97.5	98.4	49.1	70.7	97.2	**98.8**	**97.6**	98.5	**97.2**	**98.6**	96.1	97.7
010_potted_meat_can	83.7	86.7	44.1	45.6	93.3	97.3	**94.8**	**97.5**	5.5	52.6	**65.9**	**84.0**
011_banana	86.3	96.1	93.3	97.7	**95.6**	**98.4**	**95.6**	98.1	64.7	84.7	**83.6**	**95.1**
019_pitcher_base	97.3	97.7	97.9	98.2	**97.0**	**98.8**	96.8	98.0	**94.6**	**96.4**	87.0	93.4
021_bleach_cleanser	95.2	97.2	95.9	97.3	92.6	**97.5**	**94.7**	**97.5**	16.6	58.6	**84.8**	**93.2**
024_bowl	30.4	97.2	24.2	82.4	74.4	**98.4**	**90.5**	95.3	12.4	40.2	**71.8**	**91.8**
025_mug	83.2	93.3	60.0	71.2	**95.6**	**98.5**	91.5	96.1	54.4	91.3	**83.3**	**95.5**
035_power_drill	97.1	97.8	97.9	98.3	96.7	98.5	**96.3**	**97.9**	50.4	69.2	**85.5**	**92.8**
036_wood_block	95.5	96.9	45.7	62.5	**93.5**	**97.2**	92.9	97.0	**88.4**	**95.9**	85.5	92.8
037_scissors	4.2	16.2	20.9	38.6	93.5	97.3	**95.5**	**97.8**	**96.0**	**97.9**	81.0	91.7
040_large_marker	35.6	53.0	12.2	18.9	88.5	97.8	**96.6**	**98.6**	74.0	90.3	**90.3**	**96.2**
051_large_clamp	61.2	72.3	62.8	80.1	91.8	**96.9**	**92.5**	96.7	60.1	81.0	**84.5**	**93.2**
052_extra_large_clamp	93.7	96.6	67.5	69.7	85.9	94.3	**93.4**	**97.3**	44.4	85.1	**71.1**	**91.0**
061_foam_brick	96.8	98.1	70.0	86.5	96.2	**98.5**	**96.8**	98.3	89.8	**98.2**	**96.1**	97.7
mean	78.0	90.2	59.2	74.3	86.4	96.5	**93.7**	**97.5**	57.9	80.9	**84.8**	**93.8**

Table C. Robustness Analysis. We measure the standard deviation (std) of AR score and ADD-AUC metrics.

Methods	Real275 [26]		Toyota-Light [14]	
	AR	ADD-AUC	AR	ADD-AUC
Oryon [3] std	25.7	46.9	25.6	42.3
One2Any std	**15.6**	**27.2**	**22.5**	**30.5**

We observe that occlusions lead to incorrect correspondences in Oryon [3], causing significant translation errors. This results in a projected pose that is far from the ground truth, as illustrated in row 1, where the predicted pose is projected entirely out of the image. Similarly, NOPE [18] struggles with occlusions in the reference image. In such cases, suboptimal feature extraction leads to incorrect relative pose predictions in the latent space, as seen in rows 1–3. However, when the reference image provides clear, representative texture input (row 4), NOPE can handle pose estimation effectively, even if the test image is occluded. FoundationPose [27], on the other hand, heavily depends on the quality of its generated model. Figure A shows two views of the generated model: the upper view resembles the reference image, while the lower view corresponds to the test image. When the test view aligns well with the generated CAD model, FoundationPose performs reliably. However, when the test view deviates significantly from the generated model, as seen in rows 2 and 4, it fails to predict the correct pose.

In contrast, our method is highly robust to occlusions in both the reference and test images, consistently achieving accurate pose estimation even under challenging conditions.

III.5 Performance on Large View Variations

We show qualitative results for scenes with significant view variations in Figure B. These scenarios involve objects with markedly different perspectives in the reference and test images. Even in the absence of occlusion, low overlap between the object views presents a significant challenge for pose estimation when only a single reference view is provided.

Methods based on correspondences often fail in such cases due to the difficulty of establishing accurate matches. Similarly, the latent space searching approach used by NOPE [18] performs poorly when it is unable to conduct a comprehensive search in the latent space. For model-based methods, the effectiveness of the generated model can degrade under such conditions. As shown in the last two rows, only partial regions of the object are generated from the reference image, resulting in poor supervision for pose estimation when the test view has minimal overlap with the reference.

In contrast, One2Any performs robustly even with large view variations. This stability is attributed to the model's ability to predict ROC maps, effectively inferring missing parts of the object given the Reference Object Embedding (ROPE) and ensuring accurate pose predictions.

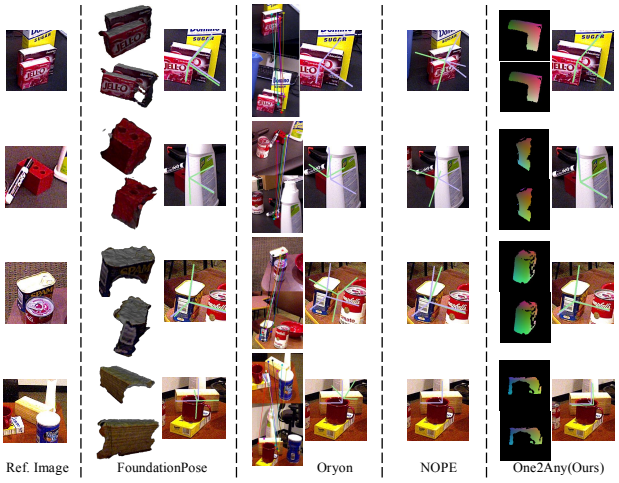

Figure A. Qualitative results on occluded scenes. We present results for scenarios where occlusion occurs in the reference image and the test image, respectively. The test image displays projected poses with axes, where the green color represents the predicted pose, and the pink color indicates the ground-truth pose. For FoundationPose [27], we also include the generated model derived from the reference image. The upper model view corresponds to the reference image, while the lower model view aligns with the test image. For Oryon [3], we visualize its predicted correspondences. For One2Any (Ours), we additionally compare the generated ROC map (below) with the ground-truth ROC map (above).

III.6 Performance on Textureless Objects

We further present additional results for textureless objects in Figure C. Textureless objects are known to introduce significant challenges for correspondence matching and template matching methods. As a result, these approaches often fail to predict accurate poses for textureless objects. For example, Oryon [3] struggles with large translation errors, leading to projections that fall outside the image frame.

In contrast, our method, which predicts poses through ROC map generation, demonstrates stability and robustness in handling textureless objects, maintaining accurate pose predictions even under these challenging conditions.

III.7 More Qualitative Results on Novel Objects

We present additional qualitative results of our method across various test datasets, including YCB-Video [29], Toyota-Light [14], T-LESS [13], IC-BIN [4], and TUD-L [14]. These datasets are popular in the BOP benchmark. The results are illustrated in Figure D, where we also display the predicted ROC map alongside the reference ROC map.

Our method demonstrates the ability to predict poses for a diverse range of unseen objects across different datasets, such as an unusual toy dinosaur and a charger. Additionally, our method is robust to challenging conditions like significant lighting changes. For instances with large view vari-

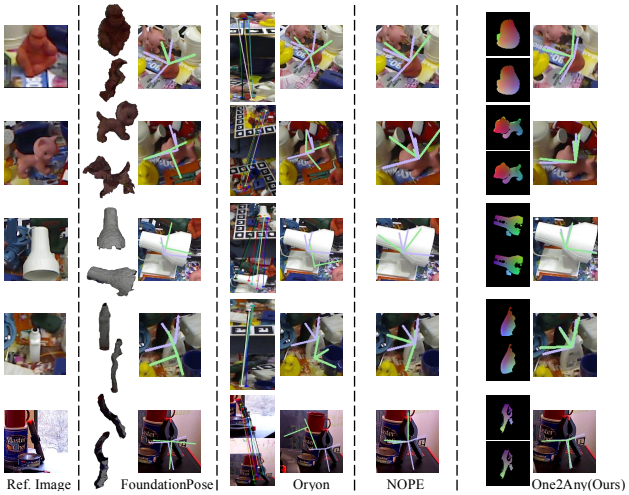

Figure B. Qualitative results on large view variations. We present results for scenarios with significant view variations, including cases with low or almost no overlap between the reference and test images. In the test images, we display the projected poses with axes, where the green color represents the predicted pose and the pink color indicates the ground-truth pose. For FoundationPose [27], we also include the generated model derived from the reference image. The upper view corresponds to the reference image, while the lower view aligns with the test image. For Oryon [3], we visualize its predicted correspondences. For One2Any (Ours), we additionally show the generated ROC map (below) alongside the ground-truth ROC map (above).

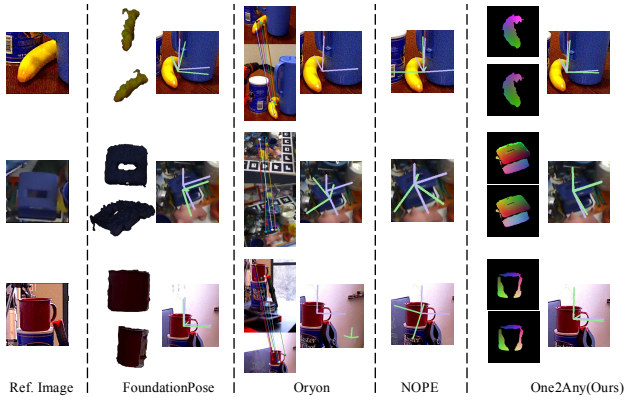

Figure C. Qualitative results on textureless objects. We evaluate the pose estimation performance for textureless objects, which is presented in Figure C. The test images display projected poses with axes, where the green color represents the predicted pose and the pink color indicates the ground-truth pose. For FoundationPose [27], we include the generated model derived from the reference image. The upper view corresponds to the reference image, while the lower view aligns with the test image. For Oryon [3], we visualize its predicted correspondences. For One2Any (Ours), we additionally compare the generated ROC map (below) with the ground-truth ROC map (above).

ations and minimal overlap, such as the cup example, our method accurately estimates the pose.

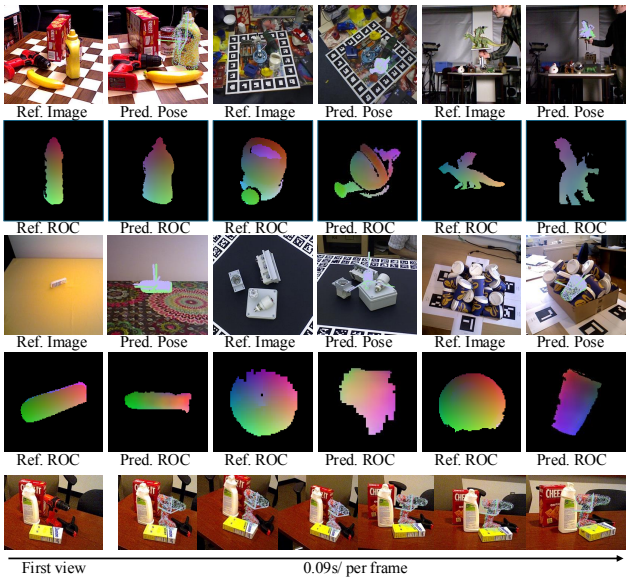

Figure D. More qualitative results. We present pose estimation results across additional test datasets, including YCB-Video [29], Toyota-Light [14], T-LESS [13], IC-BIN [4], and TUD-L [14]. For each dataset, we display the ROC maps for the reference images and the predicted ROC maps for the test images. Furthermore, we demonstrate the tracking performance on the YCB-Video dataset. The predicted pose is shown in green, while the ground-truth pose is shown in pink. Additionally, we present the projections of the object point cloud on the image, corresponding to both the ground-truth pose and the predicted pose.

We also showcase pose tracking performance with the driller object, where our method successfully follows the object's motion and achieves accurate pose tracking using only the first view as a reference. The processing time is approximately 0.09 seconds per frame with a single Nvidia RTX 4090 GPU.

III.8 Experiments with RGB-only Inputs

To further explore the ability of One2Any when dealing with RGB query images only (without depth inputs), we use RANSAC+PnP [7] for pose estimation and tested on YCB-Video [29]. Results are shown in Table D, we measure the average rotation error and translation error for detailed comparison. RGB-only input achieves competitive rotation estimation, even outperforming Umeyama [25]. However, estimating translation from a single view without depth is inherently challenging, as depth provides critical scale information. For instance, NOPE [18], a single view method which uses RGB-only input, does not estimate translation at all.

III.9 Training Data Scalability

We evaluate the impact of training data size on our method. Due to the simplicity of our data collection pro-

Table D. Performance of PnP method [7] (without depth input) compared with Umeyama algorithm [25] (with depth input). We test on YCB-V test set and measure the rotation error and translation error in detail.

Method	rotation err (°)	translation err (m)
Umeyama [25]	22.2	**0.02**
RANSAC+PnP [7]	**21.0**	0.04

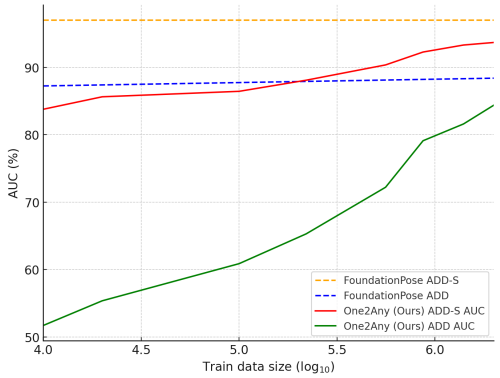

Figure E. Training data scalability. We analyze the effect of training data size on our method's performance and compare it with FoundationPose [27].

cess—requiring only image pairs of the same object—we can generate an unlimited amount of synthetic data for training. To analyze the effect of training data scale, we gradually increase the size of the training dataset and evaluate performance on the consistently used test set YCB-Video [29] test set as the evaluation benchmark. The metrics used are ADD-AUC and ADD-S AUC, with results presented in Figure E.

For comparison, we include performance from FoundationPose [27]. Our observations show that the performance of our method improves steadily with an increasing amount of training data. Notably, when the training data size increases to 1M, FoundationPose shows little to no improvement, whereas our method continues to demonstrate significant gains. This indicates that the performance of our approach can be further enhanced with even larger datasets. Due to computational resource limitations, we capped the training data size at approximately 2M samples.

IV Extension to Multi-view Setups

We further extend our method to multi-view setups. Owing to its time efficiency, our method can seamlessly handle multi-view configurations by estimating the relative pose between the test image and each reference image in the multi-view set. The best prediction is then selected based on these estimations.

IV.1 Simple Voting Strategy for Pose Selection

Assume we have N multi-view reference images, resulting in N predicted poses $[\mathbf{R}|\mathbf{t}]_{i\in N}$. We employ a simple voting strategy to identify the optimal prediction among these poses.

When utilizing multi-view reference images, we assume low overlap between the images. Consequently, for each test image, there exists an optimal prediction corresponding to one of the reference views. The objective is to use a voting strategy based on reprojection error to identify this best match.

First, we filter out incorrect predictions using reprojection error. Given a predicted relative pose between the test image and reference image A_{I_i}, we estimate the relative pose $[\hat{\mathbf{R}}|\hat{\mathbf{t}}]_{i2r}$, With the ground-truth pose of the reference view $[\mathbf{R}|\mathbf{t}]_i$. The predicted pose of the current test image is:

$$[\hat{\mathbf{R}}|\hat{\mathbf{t}}]_i = [\hat{\mathbf{R}}|\hat{\mathbf{t}}]_{i2r}[\mathbf{R}|\mathbf{t}]_i \tag{7}$$

Next, we compute the reprojection error from the test image to all reference images. Using the predicted pose, we obtain. We first get the point cloud W_Q in the world coordinate with the predicted pose.

$$W_Q = [\hat{\mathbf{R}}|\hat{\mathbf{t}}]_i^{-1}\mathbf{K}^{-1}Q_D[Q_M = 1] \tag{8}$$

$\mathbf{K} \in \mathbb{R}^{4\times4}$ is the camera intrinsic matrix, Q_D, Q_M are the depth image and the mask image respectively. Thus, for each reference image $A_j, j \in N$, we have

$$P_{A_j} = \mathbf{K}[\mathbf{R}|\mathbf{t}]_j W_Q \tag{9}$$

Then we get the projected pixels in the image frame

$$\begin{bmatrix} U \\ V \end{bmatrix} = \frac{P_{A_j}[:, :2]}{P_{A_j}[:, 2]} \tag{10}$$

And we get the projected mask as:

$$\hat{A}_{M_j}[U\ V] = 1 \tag{11}$$

To evaluate the prediction, we compute the mIoU [6] between the predicted mask $\hat{A}_{M_j}$ and the ground-truth mask A_{M_j}. Typically, for each test image, there exists at least one reference image where the reprojection results in a high overlap with the ground-truth mask, yielding a high mIoU. However, for other reference images, the predicted mask may result in a lower IoU. Consequently, we select the predicted pose corresponding to the maximum IoU across all reference images as the final predicted pose.

IV.2 Experiments with Multi-view Setups

We conduct experiments on the challenging LINEMOD dataset [12], which features significant view variation. Results are presented in Table E. In addition, we evaluated the

Table E. Performance on LINEMOD [12] dataset with multi-view setups. We report the recall of ADD-0.1d metric. Results of multi-view methods are taken from FS6D [27]. The best performance among multi-view methods are highlighted in bold. One2Any(Ours) 8-view, 16-view methods are evaluated with a pose selection strategy, while 8-best view and 16-best view results are approached with the best prediction.

Methods	**Modality**	**Ref. Images**	**ape**	**benchwise**	**cam**	**can**	**cat**	**driller**	**duck**	**eggbox**	**glue**	**holepuncher**	**iron**	**lamp**	**phone**	**mean**
OnePose [24]	RGB	200	11.8	92.6	88.1	77.2	47.9	74.5	34.2	71.3	37.5	54.9	89.2	87.6	60.6	63.6
OnePose++ [10]	RGB	200	31.2	97.3	88.0	89.8	70.4	92.5	42.3	99.7	48.0	69.7	**97.4**	**97.8**	76.0	76.9
LatentFusion [19]	RGBD	16	**88.0**	92.4	74.4	88.8	94.5	91.7	68.1	96.3	49.4	82.1	74.6	94.7	91.5	83.6
FS6D [11] + ICP	RGBD	16	78.0	88.5	91.0	89.5	**97.5**	92.0	75.5	99.5	99.5	**96.0**	87.5	97.0	**97.5**	91.5
One2Any (Ours)	RGBD	8	79.9	75.1	88.5	60.6	90.1	70.5	45.8	100.0	99.9	84.4	60.7	84.0	89.9	79.2
One2Any (Ours)	RGBD	16	82.1	85.5	92.8	75.9	94.1	80.4	65.9	100.0	99.9	70.7	61.7	91.5	84.1	83.7
One2Any (Ours)	RGBD	8-best view	85.0	93.7	97.8	84.9	94.9	90.1	73.2	100.0	99.9	88.8	89.8	95.4	85.2	90.7
One2Any (Ours)	RGBD	16-best view	84.8	**98.3**	**98.8**	**95.2**	95.9	**93.3**	**76.2**	**100.0**	**99.9**	92.9	95.1	94.4	93.9	**93.8**

performance of our method with 8 /16 reference views. Using the simple pose selection strategy, our method achieved a 50% improvement with 8 reference views and a 59% improvement with 16 reference views. Notably, the performance with 16 reference views surpasses LatentFusion [19], which learns an object latent space to render additional views. While FS6D [11] is specifically fine-tuned for the LINEMOD dataset, OnePose [24] and OnePose++ [10] require over 200 reference views to compensate for the lack of depth information but still perform poorly.

However, the simple selection strategy does not always guarantee the optimal predicted pose. To address this, we further analyze the best-case prediction results (last two rows of Table E),where the predictions most closely resemble the ground truth. These results show significant improvements, demonstrating that our method outperforms many existing multi-view approaches.

Furthermore, the experimental results demonstrate that our method has the potential to be further enhanced and explored in multi-view setups, offering even greater capabilities.

References

[1] Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020. 2

[2] Junhao Cai, Yisheng He, Weihao Yuan, Siyu Zhu, Zilong Dong, Liefeng Bo, and Qifeng Chen. Ov9d: Open-vocabulary category-level 9d object pose and size estimation. *arXiv preprint arXiv:2403.12396*, 2024. 2

[3] Jaime Corsetti, Davide Boscaini, Changjae Oh, Andrea Cavallaro, and Fabio Poiesi. Open-vocabulary object 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18071–18080, 2024. 2, 3, 4, 5

[4] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3583–3592, 2016. 4, 5

[5] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6

[7] MA FISCHLER AND. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 5, 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[10] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35:35103–35115, 2022. 7

[11] Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. Fs6d: Few-shot 6d pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6814–6824, 2022. 2, 3, 7

[12] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011. 6, 7

[13] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 4, 5

[14] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018. 2, 4, 5

[15] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4267–4276, 2021. 2, 3

[16] Jan Issac, Manuel Wüthrich, Cristina Garcia Cifuentes, Jeannette Bohg, Sebastian Trimpe, and Stefan Schaal. Depth-based object tracking using a robust gaussian filter. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 608–615. IEEE, 2016. 2, 3

[17] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 1

[18] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, and Vincent Lepetit. Nope: Novel object pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17923–17932, 2024. 3, 4, 5

[19] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10710–10719, 2020. 7

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2

[21] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2

[22] Manuel Stoiber, Martin Sundermeyer, and Rudolph Triebel. Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6855–6865, 2022. 2, 3

[23] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2, 3

[24] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022. 7

[25] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 5, 6

[26] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2, 4

[27] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024. 2, 3, 4, 5, 6, 7

[28] Manuel Wüthrich, Peter Pastor, Mrinal Kalakrishnan, Jeannette Bohg, and Stefan Schaal. Probabilistic object tracking using a range camera. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3195–3202. IEEE, 2013. 2, 3

[29] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 2, 3, 4, 5, 6