

One View, Many Worlds: Single-Image to 3D Object Meets Generative Domain Randomization for One-Shot 6D Pose Estimation

Anonymous Author(s)

Affiliation

Address

email

1 Appendix

2 This appendix is organized as follows. In section A, we describe how carefully crafted text prompts
3 guide the generation of diverse 3D models using Trellis [1], emphasizing the impact of linguistic
4 variation on visual diversity. Section B details the construction of our synthetic training dataset,
5 including the scene setup and statistical distributions across object pose, visibility, and distance.
6 In section C, we report additional experimental results, including per-dataset performance analysis,
7 failure mode characterization, and discussions of robustness across challenging conditions. In sec-
8 tion D introduces a complete pipeline for constructing a ground-truth dataset for unseen objects in
9 real-world scenes, encompassing scene acquisition, 3D reconstruction, coordinate alignment, and
10 mask generation. Together, these components provide comprehensive support for evaluating our
11 method in both synthetic and real-world conditions.

12 A Text Prompts and Model Output Diversity

13 As described in Sec. 3.5 of the main text, we utilized Trellis [1] as a tool for generating diverse
14 textures using the following input prompt:

15 Prompt: Generate a series of realistic 3D models of thermometer guns, each unique yet plausible.
16 Designs should vary in style (minimalist, industrial, futuristic, ergonomic) while incorporating es-
17 sential features like a digital display, trigger, and sensor tip. Focus on realistic materials (matte
18 plastic, brushed metal) with subtle imperfections for authenticity. Each model should feature dis-
19 tinct color schemes to differentiate them, such as using vibrant colors for the body with contrasting
20 matte or glossy accents on buttons and grips. Ensure the models are practical and usable, employing
21 diverse color combinations not only for aesthetic appeal but also to enhance usability (e.g., color-
22 coded buttons for different functions).

23 We then input the models generated based on an anchor image into our pipeline, resulting in the
24 models shown in Fig. 1. It is evident that the generated models exhibit a rich diversity of textures,
25 which helps further narrow the domain gap between the training dataset and real-world objects in
26 the subsequent step of dataset generation. The experimental results validate the effectiveness of this
27 approach, as discussed in the main text.

28 B Statistical Distribution of the Training Dataset

29 **The method for constructing a training dataset.** First, we apply the method described in Sec.
30 3.5 to perform diversified texture generation on existing object models using text prompts, thereby
31 creating 100 differently textured models as shown in Fig. 1. These models are then fed into the
32 BlenderProc [2] rendering pipeline. For each synthetic scene, an initial model and three randomly
33 selected diversified texture models are chosen as training targets; simultaneously, ten objects are



Figure 1: **Diversified Models.** This figure showcases the results of two types of diversified texture generation. For each object, the original model is displayed on the right, while the model with diversified textures is shown on the left. It can be observed that our proposed method, which utilizes text prompts to promote texture diversity, is capable of generating rich and varied surface textures. This contributes to the enhancement and diversification of training dataset generation.



Figure 2: **Generated training dataset.** This figure illustrates the dataset generated using our diversified texture models. It is evident that our generated dataset encompasses a rich variety of backgrounds, object poses, occlusion relationships, and lighting conditions. This contributes to narrowing the domain gap between the training dataset and real-world scenarios.

34 randomly picked from the BOP dataset [3] to act as occluders, and a background environment is
 35 constructed by selecting a random texture map from the CCTextures. Subsequently, 100 camera
 36 positions are randomly determined with each camera oriented towards the geometric center of these
 37 14 objects (4 targets + 10 occluders), incorporating eccentric noise into the camera positions while
 38 also introducing random perturbations to the rotation angles around their Z-axis to simulate varia-
 39 tions in real-world shooting conditions. Through this process, we have constructed a large-scale,
 40 highly varied synthetic training dataset. Fig. 2 provides examples from this dataset, illustrating its
 41 rich variety of lighting conditions, degrees of occlusion, and object scales (distances from the cam-
 42 era), significantly enhancing the robustness and accuracy of the model in pose estimation tasks, as
 43 discussed in the main text.

44 **The object pose distribution of the training dataset.** We present the distributions of Azimuth,
 45 Elevation, Object Distance, and Visibility in the final generated dataset, as shown in Fig. 3. From
 46 these distributions, it is evident that our dataset exhibits a high degree of diversity in both azimuth
 47 and elevation angles, covering a wide range of horizontal and vertical orientations of objects relative
 48 to the camera. In terms of object distance, the dataset includes samples ranging from close-up to far-

field scenarios, simulating the variation in depth at which objects may appear in real-world settings. Regarding visibility, due to the inclusion of occluders and randomized scene layouts, the extent of visible object regions varies significantly across images, resulting in diverse levels of occlusion. Taken together, the multi-dimensional diversity in our dataset closely mimics the complexity of real-world camera-captured scenes. This design choice effectively enhances the generalization capability and robustness of model training.

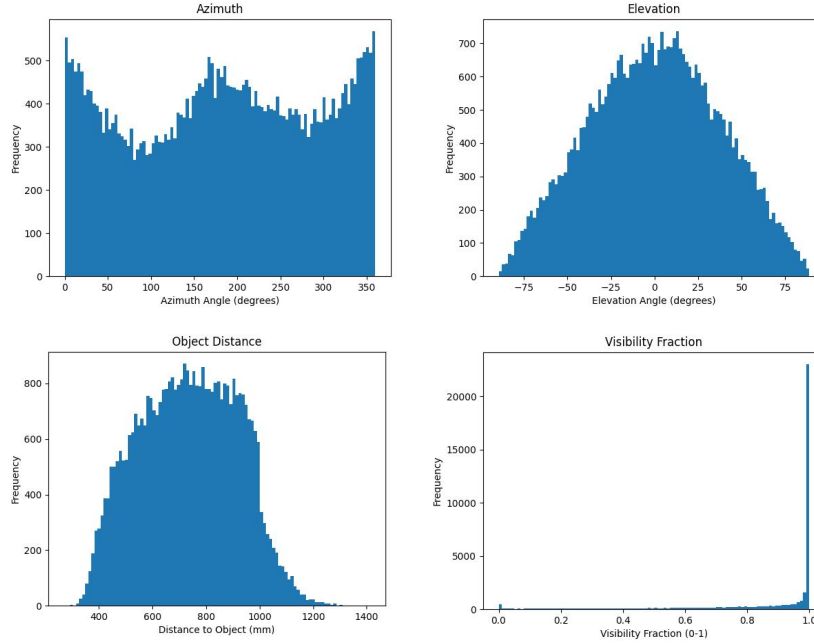


Figure 3: **Distribution Analysis of Training Dataset.** Top-left: Distribution of Azimuth angles for objects within the generated dataset; Top-right: Distribution of Elevation angles for objects; Bottom-left: Distribution of object distances from the camera; Bottom-right: Distribution of the proportion of object area visibility; These plots demonstrate that the constructed training dataset encompasses a rich diversity of poses and occlusion scenarios, underpinning its comprehensive variability and realism.

C Additional experiments

Metrics. We assess performance using the metrics defined by the BOP challenge, specifically focusing on Average Recall (AR) for Visual Surface Discrepancy (VSD), Maximum Symmetry-aware Surface Distance (MSSD), and Maximum Symmetry-aware Projection Distance (MSPD) [3]. These metrics offer complementary insights into pose accuracy by evaluating recall rates across various thresholds. This comprehensive evaluation approach ensures a thorough assessment of algorithmic performance in diverse scenarios, reflecting different aspects of precision and robustness in pose estimation.

Performance Analysis on the LM-O Dataset. The LM-O dataset [4] comprises 12 objects, predominantly distinguished by their lack of texture and frequent occlusion. Performance metrics for each object are summarized in Table 1, where τ denotes the misalignment tolerance. As shown in Fig. 4, performance degrades when the target object occupies a small region in the image. Nonetheless, the method still achieves relatively robust results under such challenging conditions. As observed in Table 1, the model exhibits stable pose estimation across most categories when the translational error threshold exceeds 0.15, indicating consistent precision and reliability in 6D pose estimation. However, the “ape” category shows notably lower accuracy, primarily due to its minimal textural information and ambiguous geometry. These characteristics result in a mismatch

between the reconstructed and ground-truth models, which in turn hampers accurate model alignment and pose estimation. Despite this limitation, as illustrated in Fig. 6, the proposed method demonstrates strong and consistent performance across the majority of object categories, highlighting its generalization capability under diverse shape and textural conditions in real-world 6D object pose estimation tasks.

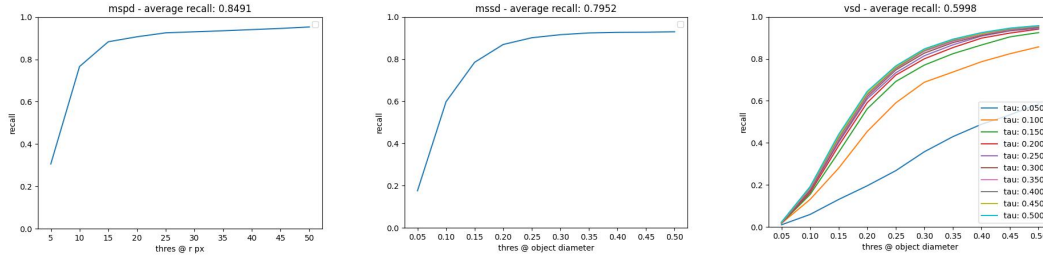


Figure 4: **Visualization of Metrics on the LM-O Dataset.** Left: Illustration of the MSPD metric variation with respect to the visible size of objects in pixels. Middle: Demonstration of the MSSD metric variation according to the object size in meters. Right: Presentation of the VSD metric variation with respect to τ (denoting the misalignment tolerance) and object size. These graphs effectively capture the metrics’ dependencies on object visibility, size, and alignment tolerance, providing insights into their varying influences under different conditions.

Table 1: Detailed Metrics on the LM-O Dataset.

Metrics	object								avg
	ape	can	cat	driller	duck	eggbox	glue	holepunch	
MSPD	79.6	93.7	80.8	82.1	84.4	86.0	77.9	91.5	84.9
MSSD	64.8	92.6	71.2	85.8	72.9	80.1	75.8	87.9	79.6
VSD	37.5	75.5	57.7	71.2	66.1	56.6	51.9	58.2	60.0
AR	60.6	87.3	69.9	79.7	74.5	74.2	68.5	79.2	74.8



Figure 5: **Performance in LM-O dataset.** In each image, the red, green, and blue lines represent the x, y, and z axes of the model, respectively, while the pink line shows the rendered contour under the estimated pose. By comparing the rendered contour with the ground-truth outline of the object, it is evident that our method is highly robust, performing well across various objects and under different occlusion scenarios.



Figure 6: **Comparison Between Original and Generated Models on LM-O dataset.** The first row displays the original object models, while the second row shows the generated models under the same pose. The third row presents a bottom-view comparison of the generated models. As can be seen, the generated models exhibit high quality and closely resemble the original objects in terms of texture and structure, demonstrating the effectiveness of our generation and scale recovery approach.

77 **Performance Analysis on the TOYL Dataset.** The TOYL [3] test dataset contains 21 objects,
78 primarily distinguished by complex lighting conditions and the fact that most objects are positioned
79 relatively far from the camera. Performance metrics for each object are summarized in Table 2. As
80 illustrated in Fig. 7, performance declines when the target occupies only a small portion of the image,
81 which is often due to the considerable distance from the camera. This makes texture the primary
82 discriminative cue for pose estimation, as the objects’ high symmetry reduces the effectiveness of
83 other features. Despite these challenges, the model exhibits relatively stable performance across
84 most object categories when the translational error threshold exceeds 0.15, indicating consistent
85 behavior under varying conditions. As shown in Table 2, the method faces notable difficulties with
86 objects numbered 04 and 18. Their high symmetry restricts reliable orientation estimation to surface
87 texture cues alone, often leading to mismatches when the opposite side of the object is visible.
88 Nevertheless, for the majority of other objects, the proposed method achieves stable and accurate
89 pose estimation, demonstrating strong robustness and adaptability even when objects are positioned
90 at significant distances from the camera.

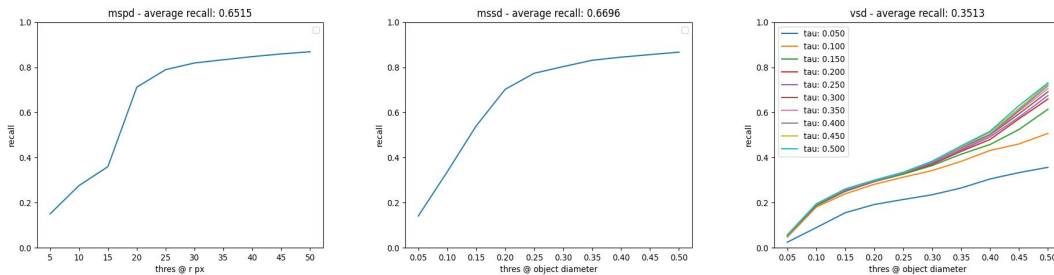


Figure 7: **Visualization of Metrics on the TYOL Dataset.** Left: Illustration of the MSPD metric variation with respect to the visible size of objects in pixels. Middle: Demonstration of the MSSD metric variation according to the object size in meters. Right: Presentation of the VSD metric variation with respect to τ (denoting the misalignment tolerance) and object size. These graphs effectively capture the metrics’ dependencies on object visibility, size, and alignment tolerance, providing insights into their varying influences under different conditions.

91 **Failure modes.** As shown in Fig. 10 and Fig. 11, the proposed method exhibits performance
92 limitations in scenarios involving severe occlusion, strong motion blur, as well as objects with low
93 texture and high symmetry.

Table 2: Detailed Metrics on the TOYL Dataset.

Metrics	object																					avg
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	
MSPD	80.1	48.5	50.5	36.8	72.1	46.9	74.2	42.2	85.1	82.2	77.8	84.6	83.1	68.9	80	72.5	59.5	31.9	74.4	65.3	54.5	55.7
MSSD	76.4	51.1	40.8	33.6	75.9	47.5	73	63.3	91.6	83.1	76.1	84.7	86.1	79.8	88.5	77.7	48.3	26	90.3	61.8	54.1	65.1
VSD	31.1	33.6	41.7	18.5	25.5	35	31.4	24.9	42.8	38.1	25.6	45.6	47.5	52.8	51.3	44.3	40.5	26.7	34.4	27.3	20.1	67.0
AR	62.5	44.4	44.3	29.6	57.8	43.1	59.5	43.5	73.2	67.8	59.8	71.6	72.2	67.2	73.3	64.8	49.4	28.2	66.4	51.5	42.9	35.2

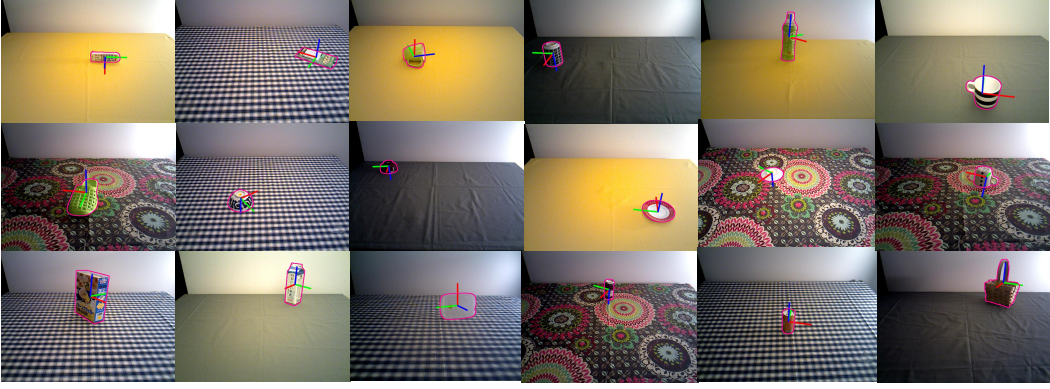


Figure 8: **Performance in TYOL dataset.** In each image, the red, green, and blue lines represent the x, y, and z axes of the model, respectively, while the pink line shows the rendered contour under the estimated pose. By comparing the rendered contour with the ground-truth outline of the object, it is evident that our method is highly robust, performing well across various objects and under different occlusion scenarios.

94 **More details about the real-world experiments.** The primary perception system for both tasks
95 relied on Intel RealSense cameras, providing RGB and depth information at a resolution of 640×480
96 and a frame rate of 30 Hz. For **Task 1** (ROKAE Pick-and-Place), the camera was fixed next to
97 the robotic arm, approximately 0.5 meter above the center of the manipulation area, angled at 15
98 degrees from the vertical. For **Task 2** (Dual-Arm AgileX PiPER Manipulation), an eye-in-hand
99 camera on each arm, offers close-up views beneficial for fine manipulation, whereas a fixed external
100 camera provides a broader view. All cameras were calibrated prior to the experiments using standard
101 procedures to ensure accurate spatial information.

102 **Task 1: Pick-and-Place with ROKAE Arm and XHAND1** This task evaluated the ability of a
103 ROKAE collaborative robot arm equipped with an XHAND1 dexterous hand to pick up a variety of
104 objects from a starting location and place them accurately at a designated target pose.

105 1. Hardware Configuration.

- 106 1. ROKAE Robot Arm ($\times 1$), is designed for collaborative applications, featuring a cabinet-
107 free design and integrated torque sensors in each joint, promoting deployment flexibility
108 and safety, with 6-DoF.
- 109 2. XHAND1 Dexterous Hand ($\times 1$), has 12 active degrees of freedom and supports force con-
110 trol with haptic sensor feedback, allowing us to adapt to object shapes and ensure stable
111 grasps. The hand is equipped with five 270° three-dimensional encircling tactile array sen-
112 sors on the fingertips, providing a tactile resolution of 12×10 per fingertip and sensing 3D
113 forces including tangential components (X and Y). Joint sensors provide position, velocity,
114 temperature, and current (torque) information.

115 The XHAND1 was mechanically attached to the ROKAE arm’s end-effector flange.

116 **2. Task Protocol.** A set of 15 diverse objects was used to evaluate the pick-and-place capabilities.
117 Objects were presented randomized within a $10 \times 10 \text{ cm}^2$ area with orientation randomized ± 15
118 degrees around the vertical axis. The target placement pose for each object was a plate into which
119 the object needed to be placed.



Figure 9: **Comparison Between Original and Generated Models on TOYL dataset.** The first row displays the original object models, while the second row shows the generated models under the same pose. The third row presents a bottom-view comparison of the generated models. As can be seen, the generated models exhibit high quality and closely resemble the original objects in terms of texture and structure, demonstrating the effectiveness of our generation and scale recovery approach in TOYL dataset.

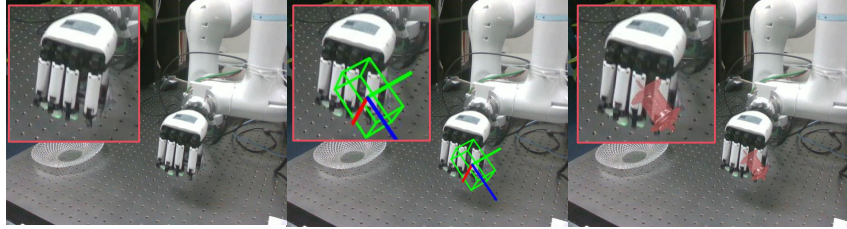


Figure 10: **Failure modes in real-world test.** Left: The original image. Middle: Detection results showing the bounding box and model’s XYZ axes. Right: Rendered model image based on the detected pose. This visualization demonstrates that our method struggles in scenarios characterized by high symmetry, significant occlusion, and severe motion blur. Despite these challenges, the method shows promise in more favorable conditions, highlighting areas for potential improvement.

120 For the 12-DoF XHAND1, grasp planning involved a pre-defined grasp synergies. Tactile and force
 121 feedback from the XHAND1’s sensors were used to confirm contact, and monitor grasp stability
 122 during lift and transport, leveraging its force-position control capabilities.

123 An initial 6D pose estimate of the object was obtained from the perception system (detailed in the
 124 main paper). This pose was used to determine the robot’s approach trajectory to the object and plan
 125 the motion to the target placement location. The system operated on the assumption that this initial

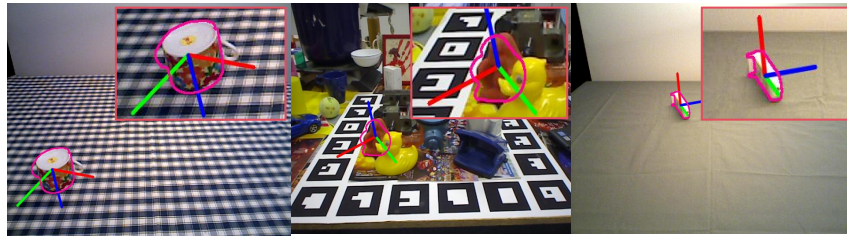


Figure 11: **Failure modes in LMO and TOYL dataset.** In each image, the red, green, and blue lines represent the x, y, and z axes of the model, respectively, while the pink line shows the rendered contour under the estimated pose. The top-right corner of each image displays an enlarged view of the corresponding failure region.

pose was sufficiently accurate for grasp execution. Then we use the pose tracking methods to guide the motion of the arm in real-time.

For each of the 15 objects, 30 pick-and-place trials were conducted. Between trials, the object was randomized within the defined start region, and the scene was reset.

3. Success Criteria.

1. Successful Grasp: object was securely held by the XHAND1, lifted at least 5 cm clear of the support surface, and maintained a stable grasp without slipping or being dropped during the initial lift.
2. Successful Transport: The object was moved from the pick location to the designated target area without any collisions with the environment or being dropped.
3. Successful Placement: The object was released at the target pose, the object had to remain stable in its placed configuration for at least 3 seconds after the XHAND1 retracted, without toppling or rolling away.
4. Overall Trial Success: A trial was deemed successful if and only if all three criteria (Grasp, Transport, and Placement) were met.

Task 2: Dual-Arm Manipulation (Pick, Handoff, Place) with AgileX PiPERs This task involved two AgileX PiPER robot arms collaboratively picking an object, handing it off from one arm to the other, and then placing it at a final target location. This sequence introduces challenges related to inter-arm coordination, synchronization, and stable object transfer.

1. Hardware Configuration.

1. AgileX PiPER Robot Arms ($\times 2$), the two PiPER arms were mounted on fixed bases, facing each other across a central workspace, 800 mm apart. Each PiPER arm was 6-DoF, equipped with an Agilix Pika Gripper.

2. Task Protocol. Similar to Task 1, objects were presented randomized within a $10 \times 10 \text{ cm}^2$ area with orientation randomized ± 15 degrees around the vertical axis. Arm 1 was set as Giver, while Arm 2 was Receiver.

A designated handoff procedure was followed. The giving arm (Arm 1) moved the object to a pre-defined handoff region (a specific area in the shared workspace). Arm 2 approached the object in Arm 1's gripper with a same grasp configuration, and then the receiving arm (Arm 2) placed the object at the table.

Arm 1 always performed the initial pick and acted as the "giver"; Arm 2 always acted as the "receiver" and performed the final place. Arm motions were synchronized using an event-based system where Arm 2's approach was triggered by Arm 1 reaching the handoff pose, and Arm 1's release was triggered by a confirmation signal from Arm 2 (successful grasp confirmed by force sensors). The timing of gripper actions (Arm 2 closing, Arm 1 opening) was critical to prevent drops.

An initial 6D pose estimate of the object was used by Arm 1 for the pick. During handoff, Arm 2 relied on the known pose of Arm 1's end-effector and visually track the object in Arm 1's gripper before Arm 2 grasped it. After handoff, the pose of the object was used by Arm 2 for planning the final placement.

For each of the 15 objects, 30 pick-handoff-place trials were conducted. Scene and object reset procedures were followed between trials.

3. Success Criteria.

1. Successful Pick (Arm 1): The object was securely grasped by Arm 1 and moved to the designated pre-handoff pose without slipping, dropping, or collision.
2. Successful Handoff: Arm 2 securely grasped the object from Arm 1, arm 1 released the object only after Arm 2's grasp was confirmed via threshold met on the gripper force sensor

- 172 (5 N). The object was not dropped during the transfer from Arm 1 to Arm 2 and the object
 173 was stable in Arm 2’s gripper after Arm 1 retracted.
- 174 3. **Successful Place (Arm 2):** The object was released by Arm 2 at the table, without slipping
 175 or being dropped.
- 176 4. **Overall Task Success:** A trial was considered successful if and only if all three stages (Pick,
 177 Handoff, and Place) were completed successfully.

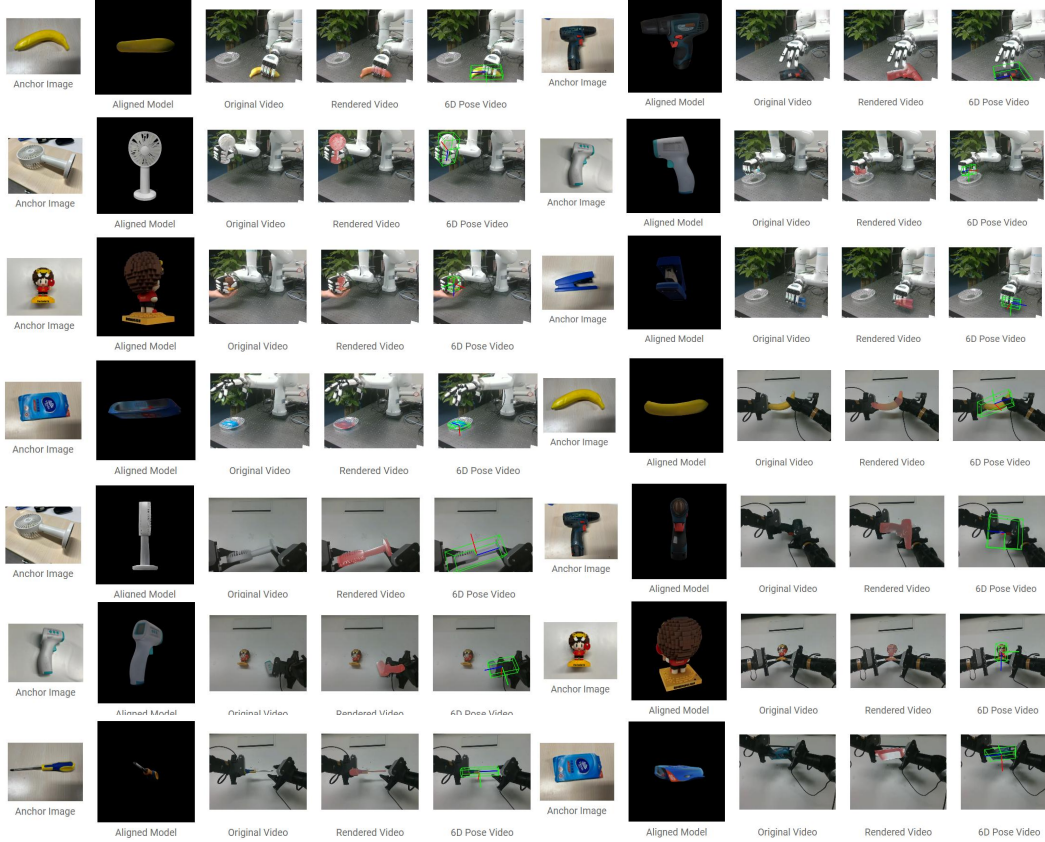


Figure 12: **Real-world Robot Manipulation Examples.** We recommend that readers visit the supplementary materials webpage to watch a dynamic demonstration video.

178 D Pipeline for Constructing a Ground Truth Real-world Dataset

179 **Scene data acquisition and preprocessing.** In this paragraph, we describe the complete pipeline
 180 for scene data acquisition and preprocessing, which includes four key stages:

- 181 1. **Deployment of AprilTags and Video Recording:** To facilitate accurate camera pose cal-
 182 culation and metric scale recovery, multiple high-contrast and easily detectable AprilT-
 183 ags [5] are strategically placed throughout the target scene. As shown in Fig. 13, these tags
 184 serve as known geometric reference points, significantly improving the robustness and ac-
 185 curacy of the reconstruction process. A monocular camera is then used to record a video
 186 sequence at a fixed frame rate, ensuring full coverage of the most of AprilTags from multi-
 187 ple viewpoints.
- 188 2. **Feature Matching and Sparse Reconstruction Using COLMAP:** Structure-from-
 189 Motion (SfM) [6] processing is performed using COLMAP. First, local feature descriptors
 190 are extracted from each frame. These features are then matched across images, followed

by geometric verification to remove outliers. Finally, an incremental SfM algorithm jointly optimizes both intrinsic and extrinsic camera parameters, resulting in a sparse 3D point cloud. This sparse reconstruction provides initial geometric priors for subsequent dense reconstruction.

3. **Distortion Correction:** Due to radial and tangential distortions inherent in standard camera lenses, which can degrade reconstruction quality, it is essential to perform distortion correction. The raw images are undistorted and mapped to a distortion-free coordinate system. Simultaneously, the corresponding camera parameter files are updated to reflect this transformation, providing more accurate input for the 3D reconstruction pipeline.

4. **Scale Estimation and Point Cloud Calibration Using XRSFM:** In purely vision-based SfM pipelines, the reconstructed scene is typically only defined up to a similarity transformation, lacking true metric scale. By leveraging the known physical dimensions of the AprilTags, the XRSFM [7] method is employed to compute the global scaling factor. This allows the sparse point cloud to be transformed from a similarity space into a Euclidean space with real-world units, enabling downstream tasks that require metric accuracy.

3D Reconstruction and Object Mesh Extraction. After metric calibration of the point cloud, we utilize Gaussian Opacity Fields (GOF) [8] to perform neural implicit field modeling of the scene. This method embeds sparse point cloud data into a Gaussian representation and models surface geometry through a learnable opacity field. Specifically, the point cloud is first initialized, and a set of spatially supported Gaussian attributes is constructed. The model is then trained using radiance optimization combined with geometry-aware training strategies, enabling the implicit representation of fine-grained surface details. Finally, a triangle mesh is extracted from the learned GOF representation via marching cubes or analogous voxelization techniques, yielding a high-quality surface reconstruction of the scene. The target object mesh is then manually segmented from the reconstructed scene mesh for further processing.

Coordinate alignment and camera/object pose estimation. This paragraph introduces the complete workflow for coordinate alignment and 6D pose estimation, including initial camera pose initialization, camera poses calculation, and per-frame object pose computation:

1. **Aligning Object and Scene Meshes for Initial Camera Pose Estimation:** The object mesh is rigidly aligned with the scene mesh using ICP (Iterative Closest Point) registration algorithms. This provides the object’s absolute position within the scene. Combined with AprilTag location information, this enables derivation of the initial camera pose $\mathbf{T}_{C_0}^W$ (rotation and translation matrices in the world coordinate system) for the first frame.

2. **Estimating Per-Frame Camera Poses Using COLMAP Features:** Based on the camera poses $\mathbf{T}_{C_i}^W$ output by COLMAP (relative to the sparse point cloud), and incorporating the previously estimated metric scale, all camera poses are transformed into the unified world coordinate system. This step establishes a reliable camera trajectory that supports accurate object position computation.

3. **Computing Object Poses Across Frames:** Assuming the object remains static in the scene, its pose in the first frame can be transformed into any other frame’s camera coordinate system through a rigid transformation chain. Specifically, given the object’s pose $\mathbf{T}_{O_1}^W$ in the world coordinate system (where subscript O_1 denotes the object in the first frame and superscript W denotes the world coordinate system), and the camera poses $\mathbf{T}_{C_i}^W$ for each frame i , we can compute the object’s pose $\mathbf{T}_{O_i}^{C_i}$ in the camera coordinate system of frame i using the following transformation:

$$\mathbf{T}_{O_i}^{C_i} = (\mathbf{T}_{C_i}^W)^{-1} \cdot \mathbf{T}_{O_1}^W \quad (1)$$

Here, $\mathbf{T}_{C_i}^W$ represents the transformation matrix from the world coordinate system to the camera coordinate system of frame i , and $(\mathbf{T}_{C_i}^W)^{-1}$ is its inverse, which transforms points from the camera coordinate system back to the world coordinate system.

239 The 6-DoF object pose in each image frame consists of three rotational parameters ($\mathbf{R}_{O_i}^{C_i}$)
 240 and three translational parameters ($\mathbf{t}_{O_i}^{C_i}$). These can be extracted from the transformation
 241 matrix $\mathbf{T}_{O_i}^{C_i}$ as follows:

$$\mathbf{T}_{O_i}^{C_i} = \begin{bmatrix} \mathbf{R}_{O_i}^{C_i} & \mathbf{t}_{O_i}^{C_i} \\ 0 & 1 \end{bmatrix} \quad (2)$$

242 **Object mask generation and visualization details.** This paragraph introduces the methodology
 243 for generating accurate object masks and visualizing key details, which includes two main steps:

- 244 1. **Rendering Object Masks Using 3D Models and Camera Poses:** Given the known 3D
 245 mesh model of the object and the current frame’s camera parameters (intrinsic + extrinsic),
 246 a binary object mask is rendered onto the image plane. This forward projection process
 247 generates a foreground-background segmentation that supports downstream tasks such as
 248 pose estimation and visibility analysis.
- 249 2. **Extracting Visible Region Masks Using Segmentation Methods (e.g., SAM 2):** To fur-
 250 ther improve mask accuracy—especially under occlusion, a pre-trained instance segmen-
 251 tation method such as SAM 2 [9] is applied to the input images. This generates a visible
 252 region mask highlighting only the observable parts of the object, which can be used during
 253 training or evaluation to exclude occluded or invisible regions and improve robustness.



Figure 13: **Testing dataset examples.** The dataset we have constructed is showcased from various views. It can be seen that our testing dataset includes a range of object poses, occlusion relationships, and distances from the camera, demonstrating its richness and diversity.

254 Finally, we constructed the following test dataset. The dataset visualization is shown in Fig. 13,
 255 and its distribution is illustrated in Fig. 14. These figures demonstrate that the dataset exhibits
 256 diverse object positions and viewing angles, making it well-suited for comprehensively evaluating
 257 the performance of the trained models.

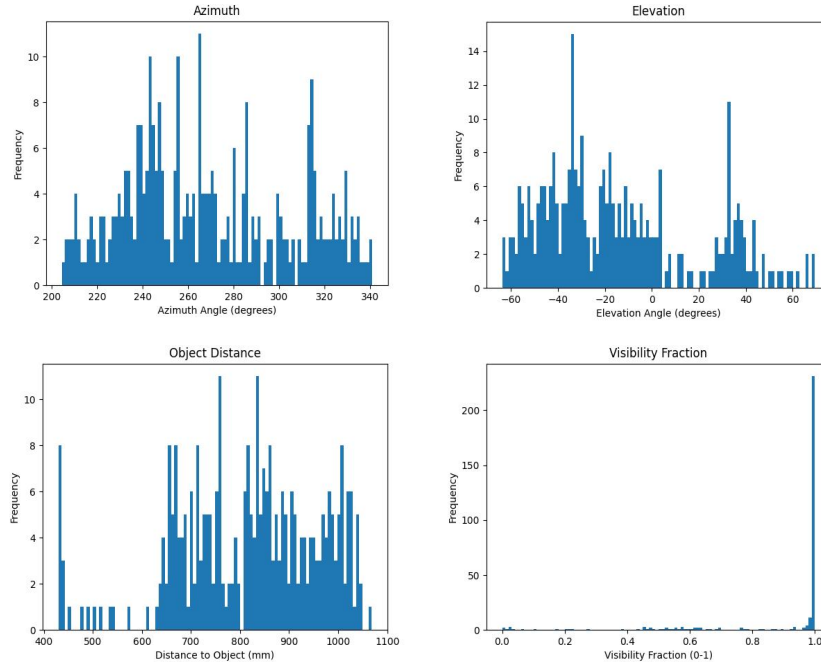


Figure 14: Distribution Analysis of Testing Dataset. Top-left: Distribution of Azimuth angles for objects within the generated dataset; Top-right: Distribution of Elevation angles for objects; Bottom-left: Distribution of object distances from the camera; Bottom-right: Distribution of the proportion of object area visibility; These plots demonstrate that the constructed testing dataset encompasses a rich diversity of poses and occlusion scenarios, underpinning its comprehensive variability and realism.

References

- [1] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.
- [2] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K. H. Strobl, M. Humt, and R. Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. doi:10.21105/joss.04901. URL <https://doi.org/10.21105/joss.04901>.
- [3] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.
- [4] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part II 13*, pages 536–551. Springer, 2014.
- [5] J. Wang and E. Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016. doi:10.1109/IROS.2016.7759617.
- [6] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi:10.1109/CVPR.2016.445.
- [7] X. Contributors. Openxrlab structure-from-motion toolbox and benchmark. <https://github.com/openxrlab/xrsfm>, 2022.
- [8] Z. Yu, T. Sattler, and A. Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.
- [9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.