Supplementary Material

A Object Instance Distribution and Snapshots

The distribution of object instances in both PACE and PACE-Sim is illustrated in Figure 1. A snapshot of all the objects can be found in Figure 2. All models have a clean exported 3D triangular mesh from the professional software provided by EinScan Pro.

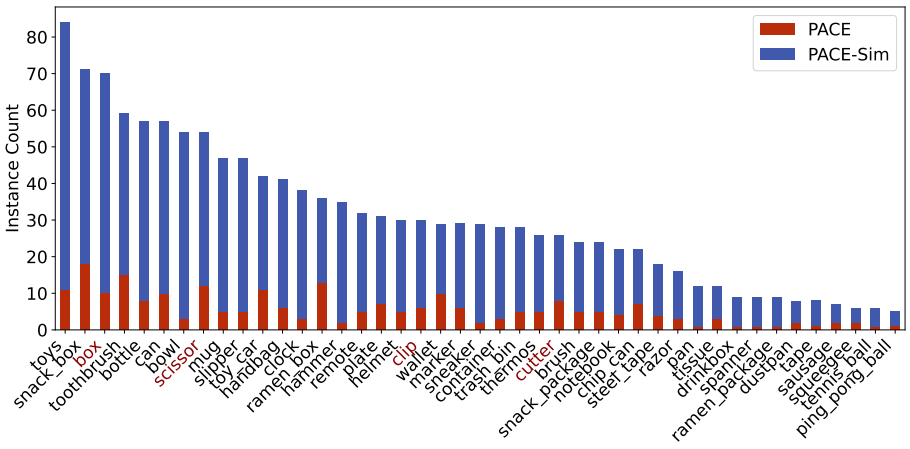

Fig. 1: Distribution of object instance counts across different categories.

B PACE-Sim visualization

Figure 3 gives example images from PACE-Sim. These images are photo-realistic and simulate the real-world clutters.

C Annotation Software and Pipeline Details

Our annotation interface features a primary display that simultaneously presents views from a 3-camera setup. Adjacently, a sidebar is integrated, showcasing the current annotations within the scene, along with a catalog of potential 3D models from our database. Users can initiate the pose estimation by aligning corresponding points on the 3D models with their 2D image counterparts, utilizing a RANSAC-PnP algorithm. Subsequent refinements to the pose are achieved through incremental rotations and translations, facilitated by keyboard inputs. Specifically, the keys ‘Q’, ‘E’, ‘D’, ‘A’, ‘W’, and ‘S’ are assigned to control

translations along the positive x-axis, negative x-axis, positive y-axis, negative y-axis, positive z-axis, and negative z-axis, respectively. Each key press results in a translation of the object by $\delta_{trans} = 1\,\text{mm}$. Additionally, the keys 'Z', 'X', 'C', 'V', 'B', and 'N' are designated for controlling clockwise and counter-clockwise rotations about the three axes, with a rotation increment of $\delta_{rot} = 0.5°$.

For enhanced user experience and annotation accuracy/efficiency, we have designed a tripartite panel system. This allows users to seamlessly toggle between 2D annotation, 3D annotation, and 2D segmentation workflows. In the 2D pose annotation panel, annotated object models are rendered and superimposed onto the original RGB image; while in the 3D pose annotation panel, object models are also placed together with the fused point cloud from all 3 cameras. As illustrated in Figure 4, this configuration enables real-time previews of annotations in both 2D and 3D, ensuring an intuitive and accurate annotation process.

D Synthetic Experiments on Annotation Quality

To assess the precision of our annotation methodology, in addition to real-world experiments, we randomly selected objects from our synthetic dataset to generate 100 video sequences of moving objects, each containing 100 frames. Initially, objects were positioned at a height above the ground and allowed to fall, guided by simulated physics. Sample sequences are illustrated in Figure. For every video sequence, we arbitrarily chose 10 objects to annotate their poses throughout the entire sequence, culminating in annotations for a total of 1000 objects.

We evaluated our annotation approach against fully automated methods that utilize the output from BundleTrack [10] as annotations. Qualitative outcomes are illustrated in Figure 5, and quantitative results are presented in Table 1. Both mean error and accuracy are reported on frames with $> 10\%$ visible parts. It is evident that our meticulous manual verification yields superior results compared to BundleTrack, which is prone to potential drift and fails to recover upon tracking loss. When annotating real-world data, we also observed occasional tracking losses with BundleTrack, necessitating manual re-annotation and tracking from the point of loss.

	Mean Error		Accuracy %	
	Rot. (°)↓	Trans. (mm)↓	Rot. ($< 2°$)↑	Trans. ($< 2mm$)↑
BundleTrack [10]	17.5	38.7	16.5	25.8
Ours Annotation	**0.9**	**1.2**	**95.9**	**93.1**

Table 1: Quantitative comparisons of annotation quality between BundleTrack and ours.

E Object Detection Benchmark

Though the primary focus of this dataset is the pose estimation, we are interested in how current state-of-the-art detection models perform on our dataset. In practice, the best method can be served as the default detection method when evaluating the pose estimation result.

E.1 Instance-Level Object Detection

Evaluation Metrics: Consistent with established literature, we employ Average Precision (AP), Average Precision at IoU thresholds of 50% (AP_{50}) and 75% (AP_{75}), along with Average Recall (AR) as the metrics for evaluation.

Baseline Methods: The task is categorized into bounding box (BBox) detection and instance segmentation (Mask). We benchmark the performance against three models: Mask R-CNN [3], YOLO-X [1], and MaskDINO [5]. MaskDINO is notable for its claim of providing precise bounding box and instance segmentation results. Models are trained on a provided PBR dataset, treating each instance as an individual "category", culminating in 576 unique categories. Results are averaged across all instances in the test set.

Results and Analysis: Quantitative outcomes are presented in Table 2. We see that YOLO-X is the best in terms of the bounding box detection while MaskDINO is the best in instance segmentation.

	Type	AP↑	AP_{50}↑	AP_{75}↑	AR↑
YOLO-X [1]	BBox	**45.4**	**60.0**	**51.5**	**61.5**
MaskDINO (BBox) [5]	BBox	31.5	42.5	36.1	48.8
Mask R-CNN [3]	Mask	26.1	41.7	30.2	33.6
MaskDINO (Mask) [5]	Mask	**29.5**	**42.0**	**33.2**	**44.1**

Table 2: Instance-level object detection results.

E.2 Category-Level Object Detection

Evaluation Metrics: The metrics for category-level object detection are identical to those used for instance-level detection.

Baseline Methods: This task is bifurcated into bounding box (BBox) detection and instance segmentation (Mask). We evaluate the performance of YOLO-X [1], MaskDINO [5], and the zero-shot detector GLIP [6]. GLIP is capable of inferring bounding boxes based on textual descriptions of target categories. For instance segmentation, we compare Mask R-CNN [3], MaskDINO [5], and Grounded SAM [7], which can generate instance masks from text prompts in a zero-shot fashion. Non-zero-shot methods are trained on images from the provided PBR training set.

Results and Analysis: Quantitative findings are detailed in Table 3. YOLO-X again is the winner is detecting bounding boxes and MaskDINO is the winner in instance segmentation. Though zero-shot methods can achieve some results but they are still far behind the supervised methods, partially due to the text ambiguity in describing the objects.

	Type	ZS	AP↑	AP_{50}↑	AP_{75}↑	AR↑
YOLO-X [1]	BBox	✗	**50.9**	**66.4**	**57.9**	**64.3**
MaskDINO (BBox)	BBox	✗	46.8	65.3	53.3	59.7
GLIP [6]	BBox	✓	22.2	30.3	26.1	59.8
Mask R-CNN [3]	Mask	✗	40.9	60.3	**47.8**	51.4
MaskDINO (Mask)	Mask	✗	**42.9**	**65.1**	47.4	**54.6**
Grounded-SAM [7]	Mask	✓	8.2	11.7	9.6	15.0

Table 3: Category-level object detection results. This table will reflect the comparative effectiveness of both zero-shot and traditional detection methods in category-level detection tasks.

F Baseline Adaptations

This section delineates the adaptations made to the baseline methodologies enabling their evaluation on our dataset. Unless otherwise stated, the configurations adhere to the defaults specified in their respective originating papers.

F.1 Handling Object Symmetry

For baselines tasked with rotational regression, we address the ambiguity presented by object symmetry, where multiple rotations correspond to a single input and causing ambiguity. We transform these rotations into distinct, unambiguous targets, following the method outlined in [8]. When evaluated against ground-truths, we consider all possible equivalent rotations following NOCS [9].

F.2 Instance-Level Pose Estimation Baselines

CosyPose [4] To enforce a fair comparison with other baselines, we exclude the random background pasting augmentation from CosyPose. Moreover, to accommodate our hardware constraints, we reduce the image resolution by half.

SurfEmb [2] With SurfEmb, images are decoded into a shared continuous embedding space alongside 3D point embeddings. Due to the extensive variety of objects in our dataset (931 in total), a separate image decoder for each would consume more than 100G GPU memory which is not feasible. Consequently, we implement a unified decoder across all objects.

F.3 Category-Level Pose Estimation Baselines

NOCS [9] We adjusted the learning rate of NOCS from 1×10^{-3} to 3×10^{-4} and transitioned the optimizer from Stochastic Gradient Descent (SGD) to Adam. These modifications resulted in a more stable loss and an expedited reduction in training loss.

F.4 Pose Tracking Baselines

CAPTRA [11] We treat all objects as non-symmetric for training purposes, given that CAPTRA's loss is designed to handle only asymmetry or continuous symmetry along the y-axis. We also limit our scope to articulated objects comprising two parts, aligning with CAPTRA's fixed-part count assumption. To manage the higher image resolution and variable object size in our dataset compared to NOCS, we decrease the *radius* parameter when computing the axis-aligned bounding box *aabb* and downsample the back-projected point clouds prior to ball filtering.

G Qualitative Comparison on Pose Estimation/Tracking

In this section, we present additional illustrations that further elucidate the performance of various baseline methodologies. Specifically, Figure 6 provides a visualization of the efficacy of instance-level pose estimation methods. Similarly, Figure 7 demonstrates the performance of category-level pose estimation techniques. Additionally, Figure 8 and Figure 9 offer visual representations of the performance metrics for model-based and model-free tracking methods, respectively. These visualizations serve to complement the quantitative analyses provided earlier, offering a more comprehensive understanding of each method's effectiveness.

Fig. 2: Snapshots from all the collected objects.

Fig. 3: Example images generated with the physically based renderer.

Fig. 4: From top to bottom: 2D pose annotation panel, 3D pose annotation panel with integrated point clouds from all three views, 2D segmentation annotation panel.

Fig. 5: Qualitative comparisons between BundleTrack and ours annotation. We see though BundleTrack can give accurate poses in some cases (first row), it is prone to drift (second and third row), and fails to recover when objects are lost (last row). Therefore, to get an accurate pose annotation, manual check and refinement are needed.

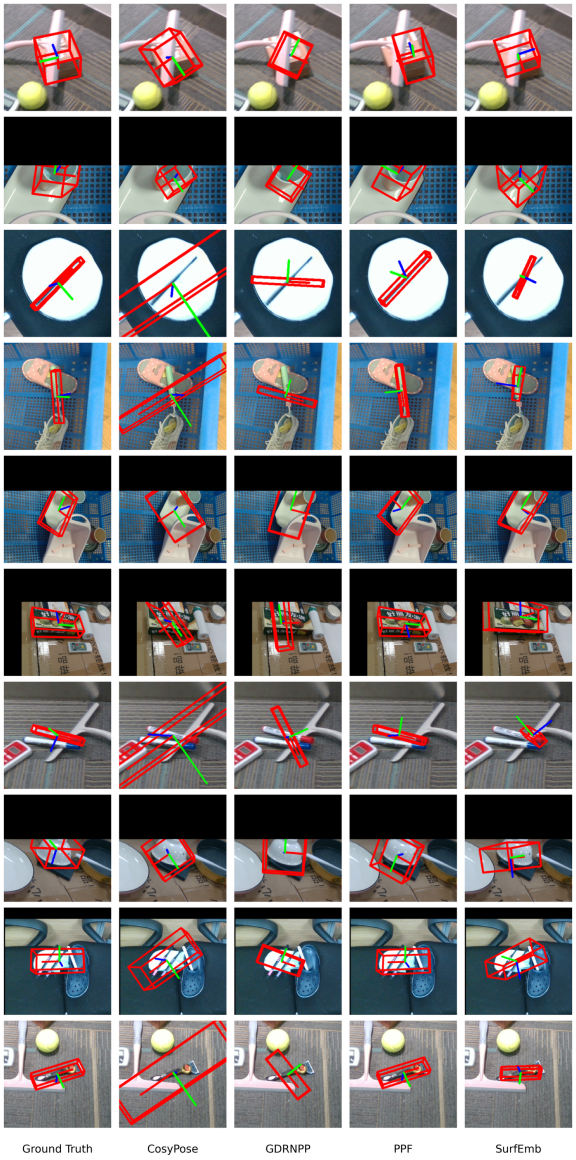

Fig. 6: Qualitative comparisons of instance-level pose estimation methods highlighting the robust performance of the PPF method across various instances.

Fig. 7: Qualitative comparisons of category-level pose estimation methods.

Fig. 8: Qualitative comparisons of model-based tracking methods.

Fig. 9: Qualitative comparisons of model-free tracking methods.

References

1. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
2. Haugaard, R.L., Buch, A.G.: Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6749–6758 (2022)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
4. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16. pp. 574–591. Springer (2020)
5. Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y.: Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3041–3050 (2023)
6. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10965–10975 (2022)
7. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
8. Pitteri, G., Ramamonjisoa, M., Ilic, S., Lepetit, V.: On object symmetries and 6d pose estimation from images. In: 2019 International conference on 3D vision (3DV). pp. 614–622. IEEE (2019)
9. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: 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. pp. 2642–2651 (2019)
10. Wen, B., Bekris, K.: Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8067–8074. IEEE (2021)
11. Weng, Y., Wang, H., Zhou, Q., Qin, Y., Duan, Y., Fan, Q., Chen, B., Su, H., Guibas, L.J.: Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13209–13218 (2021)