Supplementary Materials of SecondPose

Yamei Chen[1,*], Yan Di[1,*], Guangyao Zhai[1,2,†], Fabian Manhardt[3], Chenyangguang Zhang[4], Ruida Zhang[4], Federico Tombari[1,3], Nassir Navab[1] and Benjamin Busam[1,2,5]

[1] Technical University of Munich [2] Munich Center for Machine Learning
[3] Google [4] Tsinghua University [5] 3dwe.ai

{yamei.chen,yan.di,guangyao.zhai}@tum.de

The supplementary material encompasses this PDF file, offering additional details about our work, a video showcasing SecondPose predictions frame by frame on the REAL275 and HouseCat6D test set, and a ZIP file containing the SecondPose code for reproducibility.

1. Implementation Details

Our network is implemented on Pytorch 1.13. The backbone is based on VI-Net [4]. To obtain DINOv2 features, we initially crop the object by its bounding boxes from the original image and subsequently resize it to a resolution of 210×210. The DINOv2 model version employed is 'dinov2_vits14', with a set stride of 14. Consequently, the resolution of the output DINOv2 feature is 15×15. We randomly select 100 points from the feature map as our sampled points with DINOv2 features.

For extracting geometric features, we initially randomly sample 300 points from the entire point cloud. These points serve as the basis for estimating point-wise normal vectors. To create the hierarchical panels, we then choose range parameters $(k_0, k_1, k_2, k_3, k_4, k_5, k_6) = (0, 10, 20, 40, 80, 160, 299)$.

See Tab 1 for an overview of the number of trainable parameters and frozen parameters of our method and VI-Net.

Number of Parameters	Trainable	Frozen
VI-Net	27,311,368	0
Ours	33,639,561	22,056,576

Table 1. Parameter Count

2. Further Explanations of the Pipeline

Invariance *vs* Equivariance. Following VI-Net [4], our backbone ensures that, when the input point-wise features are approximately SE(3)-invariant, the output feature map is approximately SE(3)-equivariant. We used the term "SE(3)-invariant" to emphasize that our input features are invariant.

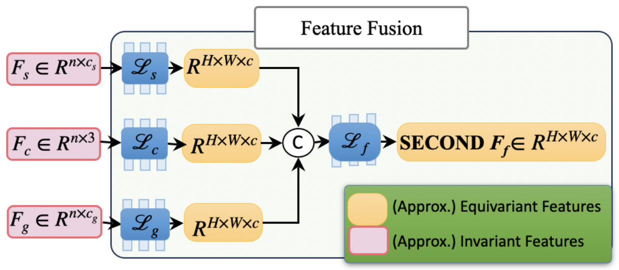

Figure 1. **Feature Fusion** We illustrate fusion process with annotated approximately equivalent and approximately invariant features.

The process of feature fusion is illustrated in the Figure 1 below, the RGB values F_c, the DINOv2 features F_s, and the HP-PPF features F_g are our invariant input features.
1) All input features are approximately invariant: the geometric features and RGB values are inherently invariant. The DINOv2 features are approximately invariant due to their training on a large-scale dataset, ensuring consistent semantic representation. This consistency in semantic meaning, regardless of rotation/translation, implies SE(3) consistency, thus leading to approximate SE(3) invariance.
2) The output is equivariant: similar to VI-Net, our backbone transforms point-wise features with the point cloud's 3D coordinates into a 2D feature map. These maps are then processed by ResNets that approximately maintain SE(3) equivariance (see section 3.4 and [4] for details). Consequently, when the input features are approximately SE(3) invariant, the output 2D feature map is approximately SE(3) equivariant.

Visualization of Feature Maps. In Fig 2, we visualize the features of same object in two frames, each with a different pose.

"RGB", "DINOv2", "HP-PPF" depict our input features, which are roughly SE(3) invariant. "Second(2D)" represents our post-fusion feature map, utilized for pose estima-

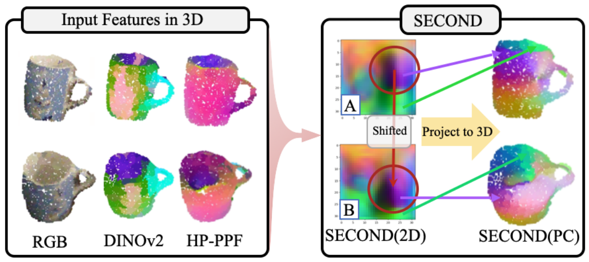

Figure 2. **Feature Maps** We fuse features of RGB, DINOv2 and HP-PPF intoo a 2D feature map that is approximately SE(3)-equivariant.

tion and approximately SE(3) equivariant; we observe slight shifts in the feature map pattern upon rotation and translation. For visualization, we also present "Second(PC)", a point-wise feature obtained by projecting "Second(2D)" back onto the point cloud, using the pixel-point correspondence, and it's also approximately invariant.

Shape of All Feature Maps and Other Intermediate Representation. Fig 1 illustrates our input features as: $F_c \in R^{n\times 3}$, $F_g \in R^{n\times c_g}$, and $F_s \in R^{n\times c_s}$, while after modules $\mathscr{L}_c$, $\mathscr{L}_g$, $\mathscr{L}_s$, our features have shape $R^{H\times W\times c}$, and after fusion module $\mathscr{L}_f$ the feature is of shape $R^{H\times W\times c}$.

3. More Experimental Results on HouseCat6D

We report more metrics on HouseCat6D [3] in Table 2. We note that our approach outperforms other methods by a significant margin across all metrics. Especially on the restricted metrics IoU$_{75}$ and 5° 2 cm, SecondPose outperforms VI-Net by 22.1% and 31.0% respectively.

We present the categorical results of our experiment on HouseCat6D in Fig. 5. Our method exhibits a substantial performance advantage over VI-Net in categories such as box, can, cup, remote, teapot, and shoe. However, in other categories, namely bottle, cutlery, glass, and tube, our method shows a slightly lower performance compared to VI-Net. We noted a shared characteristic among these categories—items within them typically display either high reflectivity or high transparency. Under optical conditions of this nature, DINOv2 tends to encounter difficulties in extracting meaningful semantic information.

4. Failure Cases and Limitations

We present typical failure cases in both REAL275 and HouseCat6D.

The failure cases of HouseCat6D are presented in Fig. 3. There are four common failure types. (A) highlights instances involving transparent items where DINOv2 struggles to extract meaningful semantic features, leading to poorer performance on transparent items. (B) illustrates a self-occlusion scenario, complicating pose prediction due to obscured essential features like the mug handle, crucial for object identification and orientation. In (C), the tube represents items with high reflectivity, a condition often associated with DINOv2 failures. (D) illustrates failures attributed to atypical shapes.

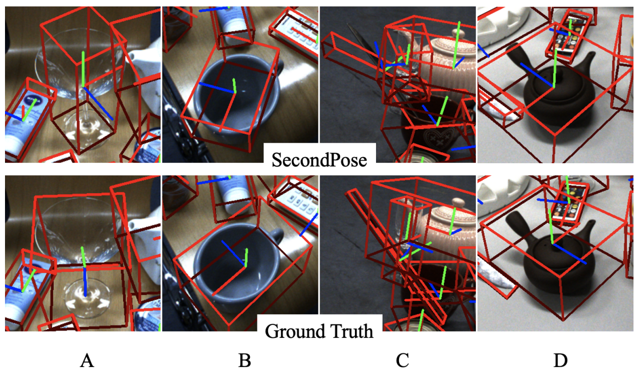

Figure 3. **Failure cases in HouseCat6D.** We illustrate common failure scenarios on HouseCat6D. (A) depicts instances of transparent items; (B) showcases items with pronounced self-occlusion; (C) the tube represents items with high reflectivity; (D) illustrates failures attributed to atypical shapes.

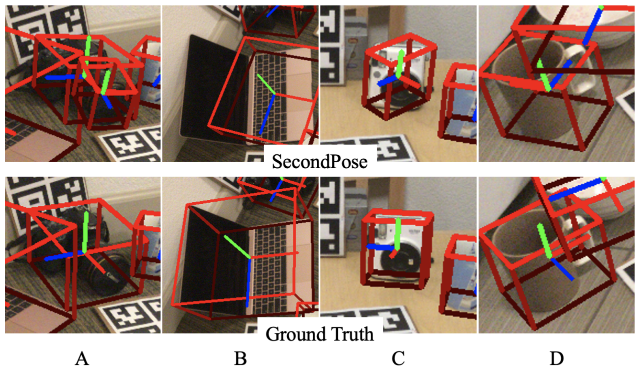

Figure 4. **Failure cases in REAL275.** We illustrate common failure scenarios on HouseCat6D. (A) represents failure due to wrong instance segmentation; (B)-(D) illustrates failures due to wrong prediction of the y-axis.

The failure cases of REAL275 are presented in Fig. 4. (A) signifies failures arising from false positive detection results. Meanwhile, (B)-(D) illustrate errors related to the wrong orientation prediction of the z-axis, where we observed that on REAL275, our model tends to predict the y-axis accurately.

In summary, there are four primary typical failure scenarios: Firstly, instances where DINOv2 fails to extract meaningful semantic information under specific optical conditions such as high reflectivity or high transparency. Secondly, when severe occlusions are present. Thirdly, when the item displays an atypical shape. Finally, errors are caused by the exclusive parts out of our pipeline, such as the detection frontend.

Method	HouseCat6D				
	IoU $_{75}$	5° 2 cm	5° 5 cm	10° 2 cm	10° 5 cm
FS-Net [1]	14.8	3.3	4.2	17.1	21.6
GPV-Pose [2]	15.2	3.5	4.6	17.8	22.7
VI-Net [4]	20.4	8.4	10.3	20.5	29.1
SecondPose (Ours)	**24.9**	**11.0**	**13.4**	**25.3**	**35.7**

Table 2. Quantitative comparisons of different methods for category-level 6D object pose estimation on HouseCat6D [3].

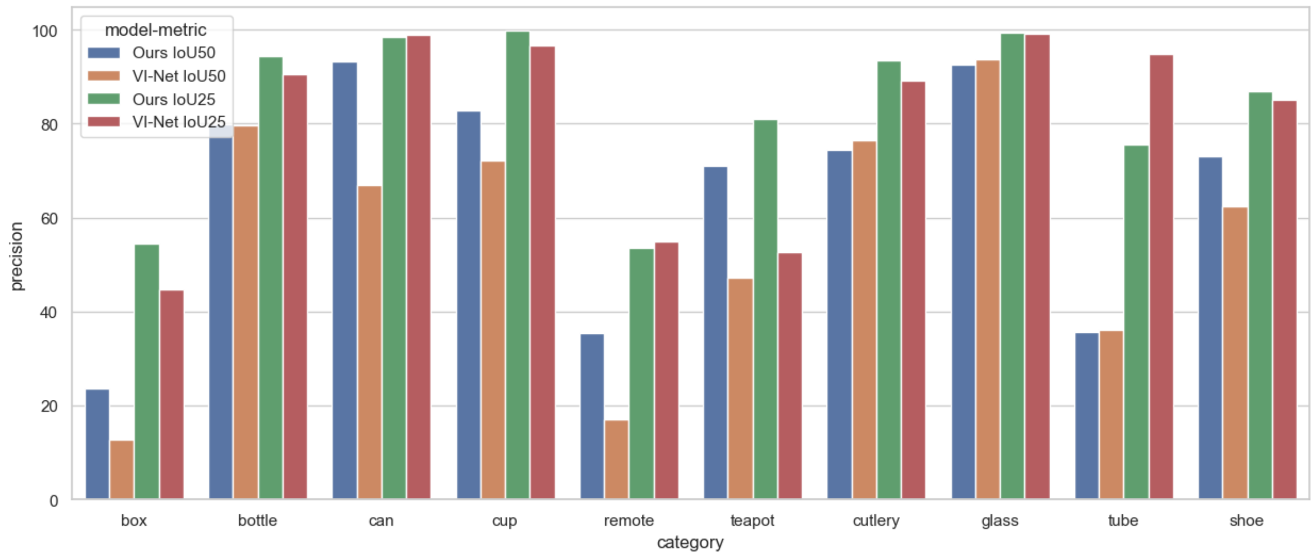

Figure 5. **Categorical results on HouseCat6D.** We visualize the comparison of our IoU$_{25}$ and IoU$_{50}$ results on HouseCat6D with those of VI-Net.

References

[1] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1581–1590, 2021. 3

[2] Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6781–6791, 2022. 3

[3] HyunJun Jung, Guangyao Zhai, Shun-Cheng Wu, Patrick Ruhkamp, Hannah Schieber, Pengyuan Wang, Giulia Rizzoli, Hongcheng Zhao, Sven Damian Meier, Daniel Roth, Nassir Navab, et al. Housecat6d–a large-scale multi-modal category level 6d object pose dataset with household objects in realistic scenarios. *arXiv preprint arXiv:2212.10428*, 2022. 2, 3

[4] Jiehong Lin, Zewei Wei, Yabin Zhang, and Kui Jia. Vi-net: Boosting category-level 6d object pose estimation via learning decoupled rotations on the spherical representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14001–14011, 2023. 1, 3