# iG-6DoF: Model-free 6DoF Pose Estimation for Unseen Object via Iterative 3D Gaussian Splatting


## Abstract

*In this supplementary material, we first elaborate the details about our detector architecture in Section A. Then, preliminary of group convolution is given in Section B, and detailed description of the training loss function can be found in Section C. We explain the differences between our work and camera pose estimation methods in Section D. Finally, more visual results are given in Section E. Note that we did not include all the material in the main paper due to the space limit.*


## 1. A. Detial of detector

Similar to Gen6D [3] and TGID [1], we apply a correlation-based instance object detector. We employ a VGG-11 [6] network to extract feature maps from both reference and query images. Subsequently, reference image feature maps serve as convolution kernels to generate score maps by convolving with the query image feature map. From the resulting multi-scale score maps, we regress a heatmap and a scale map. Unlike Gen6D, which outputs 2D bounding boxes from the object detection module and includes unnecessary background information, we output segmentation masks to ensure the accuracy of the subsequent refiner. Specifically, we set a per-pixel confidence score, and pixels are considered part of the target object when their confidence exceeds a certain threshold.

## 2. B. Preliminary of group convolution

we introduce some backgrounds about $\mathcal{SO}(3)$ space in this section. For more detail please refer to [4, 8].

**Icosahedral Group**: We define feature maps on the largest discrete finite subgroup of $SO(3)$, the icosahedral group $G$. This group consists of 60 rotations that preserve the symmetry of a regular icosahedron. Due to the group's closure property, for any $g, h \in G$, the composition of rotations $gh$ is also in $G$.

**Group Action**: An element $g \in G$ can act on other objects. We use two types of group actions: $T_g \circ \mathcal{P}$ represents rotating a set of points $\mathcal{P}$ by the rotation $g \in G$, and $P_g \circ f$ represents permuting the matrix $f$ according to $g \in G$.

**Neighborhood Set**: To define the convolution layer on the icosahedral group $G$, we introduce a neighborhood set $H$, which is analogous to the $3 \times 3$ or $5 \times 5$ neighborhoods used in standard image convolution.

## 3. C. Loss of heat map

Given a heat map $H$, we treat all values as logits and employ a binary classification loss to guide the prediction of the heat map as in Gen6D [3]. To do this, we first project the object center onto the image using the ground-truth object pose. A pixel on the heat map is considered correct if its distance from the object center projection is less than 1.5 pixels; otherwise, it is incorrect. The loss function is formulated as:

$$\begin{aligned}\ell_{heat} = \sum_{p} & -\mathbb{1}(\|p - c_{prj}\|_2 < 1.5) \log \sigma(H(p)) \\ & - (1 - \mathbb{1}(\|p - c_{prj}\|_2 < 1.5)) \log(1 - \sigma(H(p))), \quad (1)\end{aligned}$$

where $\mathbb{1}$ is an indicator function, $c_{prj} \in \mathbb{R}^2$ is the 2D projection of the object center, $p$ represents a pixel on the heat map, $\sigma$ is the Sigmoid function, $H(p)$ denotes the heat value at pixel $p$.

## 4. D. Compared with NeRF-based Camera Pose Estimation.

Currently, some methods have successfully applied NeRF [5] or 3DGS [2] to camera pose estimation tasks, achieving impressive results. However, our approach differs significantly in that object pose estimation involves additional challenges, such as object detection and separating the object of interest from the background. Furthermore, objects often occupy only a small portion of the entire image, resulting in insufficient keypoints and a higher likelihood of weakly textured regions. Additionally, while methods like iNeRF [9] and iComMa [7] optimize the camera pose $R$ and $t$ jointly, we decouple $R$ and $t$ and process them asynchronously, which leads to better results.

## 5. E. More visual results

Here, we present additional visual results. Figure 1 shows results on the LM dataset, including models of an ape, cat, and duck. Figure 2 displays results from real-world scenes captured by us. For more details, please refer to the demo video in the supplementary materials.

## References


[1] Phil Ammirato, Cheng-Yang Fu, Mykhailo Shvets, Jana Kosecka, and Alexander C Berg. Target driven instance detection. *arXiv preprint arXiv:1803.04610*, 2018. 1

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[3] Yuan Liu and Yilin Wen. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In *ECCV*, 2022. 1

[4] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[5] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[7] Yuan Sun, Xuan Wang, Yunfan Zhang, Jie Zhang, Caigui Jiang, Yu Guo, and Fei Wang. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint arXiv:2312.09031*, 2023. 1

[8] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10376–10393, 2023. 1

[9] Lin Yen-Chen and Pete Florence. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 1

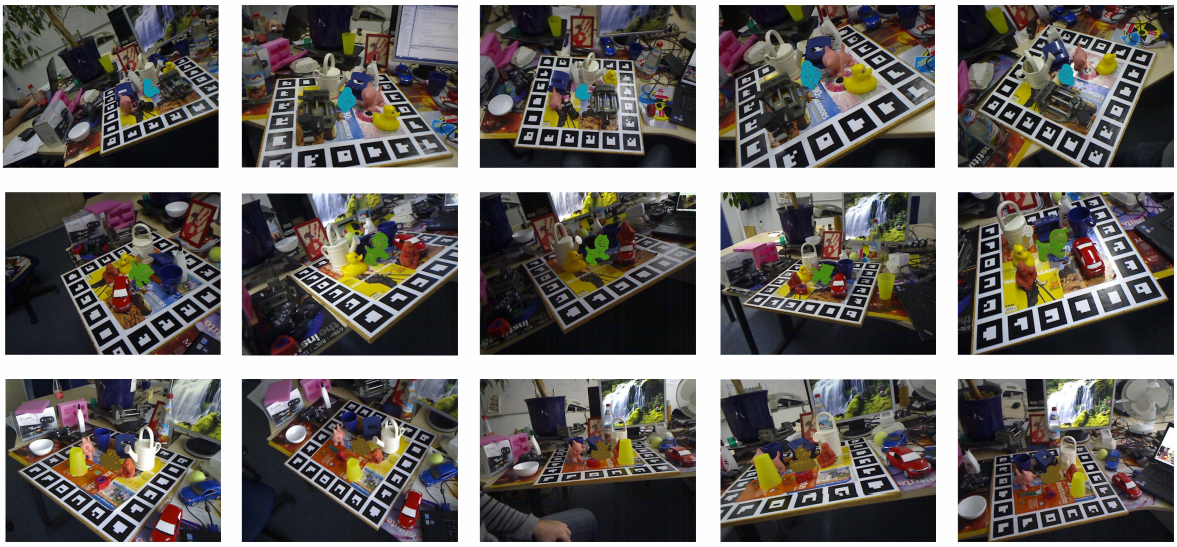

Figure 1. Qualitative results on LM dataset. Here we show visualizations of results on LM dataset. Points on different meshes in the same scene are in different colors which projected back to the image after being transformed by the predicted pose.

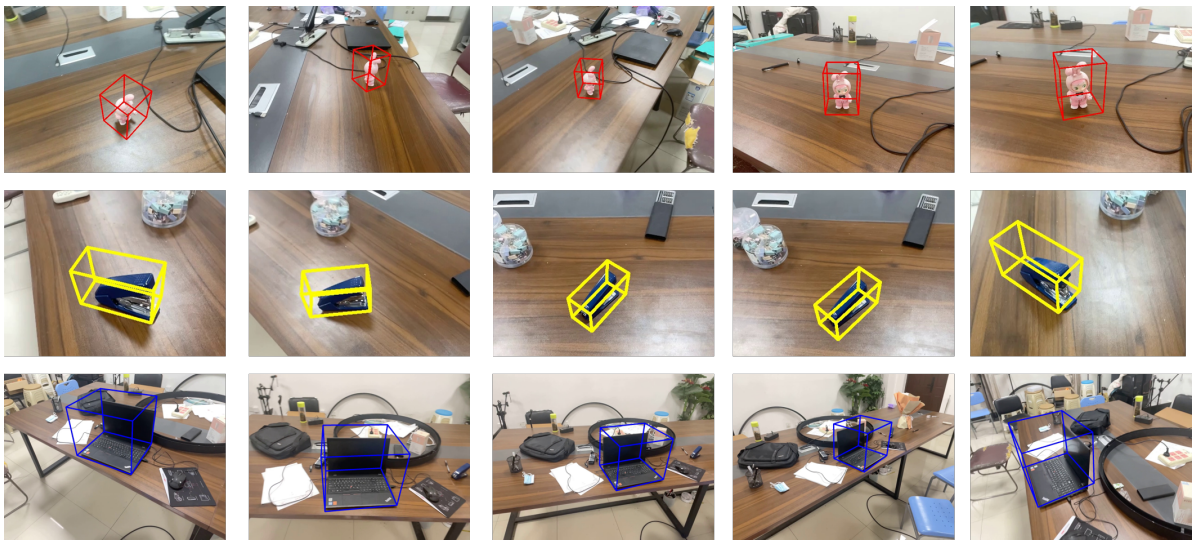

Figure 2. Visual results on real scene.