information

MDPI

Article

TGNF-Net: Two-Stage Geometric Neighborhood Fusion Network for Category-Level 6D Pose Estimation

Xiaolong Zhao [1,2], Feihu Yan [1,2], Guangzhe Zhao [1,2,*] and Caiyong Wang [1,2]

[1] School of Intelligence Science and Technology, Beijing University of Civil Engineering and Architecture, Beijing 100044, China; 2108550022061@stu.bucea.edu.cn (X.Z.); yanfeihu@bucea.edu.cn (F.Y.); wangcaiyong@bucea.edu.cn (C.W.)
[2] Beijing Key Laboratory of Robot Bionics and Function Research, Beijing 100044, China
* Correspondence: zhaoguangzhe@bucea.edu.cn

Abstract: The main goal of six-dimensional pose estimation is to accurately ascertain the location and orientation of an object in three-dimensional space, which has a wide range of applications in the field of artificial intelligence. Due to the relative sparseness of the point cloud data captured by the depth camera, the ability of models to fully understand the shape, structure, and other features of the object is hindered. Consequently, the model exhibits weak generalization when faced with objects with significant shape differences in the new scene. The deep integration of feature levels and the mining of local and global information can effectively alleviate the influence of the above factors. To solve these problems, we propose a new Two-Stage Geometric Neighborhood Fusion Network for category-level 6D pose estimation (TGNF-Net) to estimate objects that have not appeared in the training phase, which strengthens the fusion capacity of feature points within a specific range of neighborhoods, enabling the feature points to be more sensitive to both local and global geometric information. Our approach includes a neighborhood information fusion module, which can effectively utilize neighborhood information to enrich the feature set of different modal data and overcome the problem of heterogeneity between image and point cloud data. In addition to this, we design a two-stage geometric information embedding module, which can effectively fuse geometric information of the multi-scale range into keypoint features. This way enhances the robustness of the model and enables the model to exhibit stronger generalization capabilities when faced with unknown or complex scenes. These two strategies enhance the expression of features and make NOCS coordinate predictions more accurate. Many experiments show that our approach is superior to other classical methods on the CAMERA25, REAL275, HouseCat6D, and Omni6DPose datasets.

Keywords: six-dimensional pose estimation; feature fusion; neighborhood information; geometric information

check for updates

Academic Editor: Francesco Fontanella

Received: 4 December 2024
Revised: 30 January 2025
Accepted: 1 February 2025
Published: 6 February 2025

Citation: Zhao, X.; Yan, F.; Zhao, G.; Wang, C. TGNF-Net: Two-Stage Geometric Neighborhood Fusion Network for Category-Level 6D Pose Estimation. *Information* **2025**, *16*, 113. https://doi.org/10.3390/info16020113

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Six-dimensional pose estimation is one of the core tasks for objects to perceive and understand their positions in complex environments. In many practical applications, especially in embodied artificial intelligence [1], robot manipulation [2], and augmented reality (AR) [3], it is essential to precisely identify the location of an object. Instance-level methods [4,5] mostly rely on instance-level Computer Aided Design (CAD) [6] models, which do not generalize to new instances and greatly limit their practical application in life. To overcome this difficulty, category-level 6D pose estimation has emerged, which

has attracted great attention from researchers because it can estimate the poses of unseen instances without using instance-specific CAD models.

Previous methods for category-level 6D pose estimation [7–9] were based on a single data source type; however, such methods exhibit limitations due to factors such as illumination variations, occlusion, weak textures, and instability at different camera viewpoints. Ref. [10] proposed integrating angle and pose markers to solve the problem of instability at different camera views. However, the lack of information richness of a single data source type limits the information flow and complexity of the data representation. When confronted with new scenarios, generalization ability may be limited. Therefore, most of the later methods [11–13] shifted their research focus to the fusion of multi-modal data to improve the feature richness, but these methods (Figure 1a) simply fused the data of two modalities, which may lead to inaccurate pose estimation results due to the limited information expressed by the fused feature points. To overcome these issues, as displayed in Figure 1b, we devise a new neighborhood information fusion strategy. We use the neighborhood information of the feature points to embed them into the corresponding feature points of different modalities, which can enhance the feature information richness and stability.

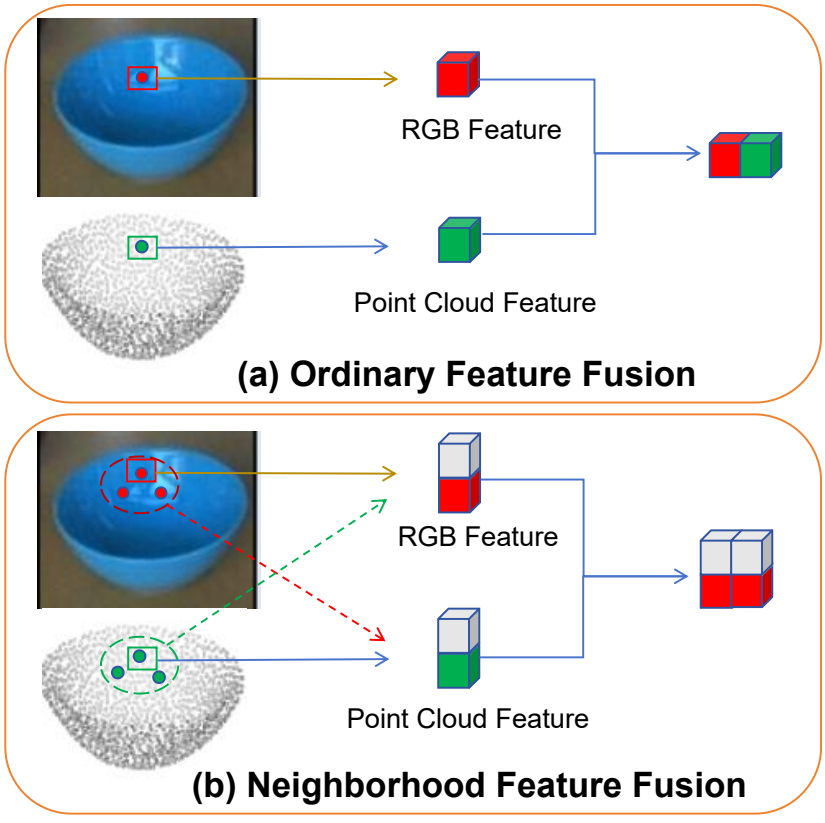


Figure 1. Comparison between different networks. Unlike ordinary feature fusion (**a**), our method (**b**) fuses the neighborhood information of different modalities into the feature points. The red/green/gray cubes represent RGB/point cloud/neighborhood features respectively.

In the category-level 6D pose estimation method, we need models of different instances to train the network and require them to belong to the same category. Thus, significant category variation becomes a great challenge. Some previous work [14,15] aimed at establishing dense correspondences between images and eventually regressing the object pose through point-level dense correspondences. However, massively incorrect correspondence diagrams are often generated when the instances differ significantly. To address this crucial issue, we introduce a two-stage geometry information embedding module. The module is

able to capture geometric features in localized areas of images and point clouds that are essential for understanding the details and local structure of objects. Based on the captured local information, the module further integrates this information into a global context. A more comprehensive comprehension of the overall shape can be achieved by taking the relative positions and relationships among feature points into account.

In order to address the above problem, we devise a Two-Stage Geometric Neighborhood Fusion Network for category-level 6D position estimation (TGNF-Net). Firstly, we design a neighborhood information fusion (NIF) module and for data of different modalities, different methods are used to extract their feature points. The feature similarity is calculated for the two types of feature points. For each feature point, the N most similar feature points of another modality are selected, and these N most similar feature points are weighted and summed to be integrated into the current feature points. This approach fuses different modal data to enrich the feature information. Meanwhile, we design a two-stage geometric information embedding (TGIE) module, which can effectively extract the geometry of the object. For local information capture, we select the K nearest neighbors of each feature point in the whole point cloud space as the local information features and incorporate the local information features into each feature point through the attention mechanism. For global information capture, the K nearest neighbor points in the set of extracted keypoints are selected as features of global information, and the global information features are embedded in each keypoint. Each feature point is embedded with both global and local geometric information, thus establishing a stable keypoint set correspondence for 6D pose estimation.

In summary, the principal contributions can be summarized as follows:

- We introduce a new category-level 6D pose estimation TGNF-Net architecture, which can estimate instances that have not appeared in the training phase without using shape priors of the object and improve generalization of the model for different types and shapes of objects.
- We propose neighborhood information fusion and two-stage geometric information embedding modules, which uses the neighborhood information as well as the geometric features to represent the structure of the object, enhancing the spatial comprehension ability of the model and the stability of pose estimation in complex scenes.

The subsequent sections are organized as follows. Section 2 recall the related work. Section 3 describes the structure of our model, including an explanation of each module and its functions. Section 4 focuses on the experimental setup, including comparative experiments and ablation studies. Section 5 is our conclusion of the entire paper.

2. Related Work

This portion offers a concise summary of the latest advancement in 6D pose estimation research, organized into two aspects for review and discussion.

2.1. Instance-Level 6D Object Pose Estimation

The task of instance-level 6D object pose estimation focuses on determining the 6D pose parameters of an object with a known 3D model in a real scene image. Extant approaches can be classified into four distinct groups: dense correspondence methods, template matching methods, voting-based methods, and direct regression methods.

Dense correspondence methods aim to build dense 2D–3D or 3D–3D dense correspondence and determine the position of object based on the dense correspondences using the Perspective-n-Point (PnP) [16] algorithm or the least squares method. CDPN [17] proposes a coordinate-based disentangled pose network that divides different rotations and translations into independent branches for object pose estimation. DPOD [18] proposes a

new 2D–3D multi-class object detection module based on dense correspondences and a personalized optimization method based on deep learning. On this basis, DPODv2 [19] proposes a unified deep network capable of centralizing the processing of multiple modal data and incorporates a rendering-based differentiable object pose optimization method. Some methods [20–23] recover the target object pose by establishing a 3D–3D correspondence and usually rely on depth sensors or reconstructing a 3D point cloud to acquire the 3D data of the object and establish the correspondence through a matching algorithm. Overall the above methods utilize local correspondences to forecast the pose of targets, but the above algorithms face great difficulties when dealing with objects that lack distinct shapes and textures.

Template matching [24–26] methods consist in discovering the closest template from a set of templates with the real posture of the object. The template they obtain contain the 2D projection obtained from the CAD model of the object, but in the case of low light, it is difficult to obtain the object information, resulting in poor pose estimation performance. Ref. [27] uses the point cloud as a template to propose a match normalization strategy to deal with the difference in feature distribution between the point cloud and the CAD model. In order to obtain higher accuracy, template matching methods need to occupy substantial template storage capacity, which increases the computational cost of the algorithm.

Most of the voting-based methods are aimed at ascertaining the location of the object through pixel-level or point-level voting schemes. MV6D [28] takes multi-view RGB-D images as input, extracts image features and geometric features, respectively, and predicts the target pose in a cluttered scene by means of feature fusion. SC6D [29] proposes a three-dimensional rotational representation that avoids the dependence on more prior knowledge of symmetrical objects. These methods usually rely on the quality of pixel points, which may lead to poor robustness and a relatively time-consuming training process and require a large number of computing resources.

The goal of the regression-based approaches is to directly obtain the pose of an object from the learned features. GCCN [30] and HFT6D [31] take advantage of the complementarity of the two data sources, and the combination of the two improves the accuracy of the regression method. SMOC-Net [32] puts forward a self-supervised network, which consists of a teacher module and a student module; the teacher module performs preliminary training, and the student module predicts the camera pose from the real image. These methods are often sensitive to the shape and texture of objects, and regression methods do not work well for occlusion and other complex scenes.

In addition to these, most instance-level methods require a precise target CAD model and have limited generalization capabilities for objects that cannot be seen. However, our model can adaptively examine the keypoints of the object according to different instances, extract local-to-global information, and effectively perform object pose estimation for instances without accurate CAD models.

2.2. Category-Level 6D Object Pose Estimation

The category-level 6D object pose estimation method can estimate objects that have not been seen in a category. Existing methods can be roughly classified into two groups: methods based on shape priors and methods without shape priors.

The methods based on shape priors use the instance CAD models that have been detected in the category to learn, derive the shape priors, and then use their prior information to guide the objects in the classes that have not detected the instance CAD models for pose estimation. Some methods [33–35] obtain the object pose by reconstructing the NOCS [36] shape and aligning the NOCS [36] shape and the target point cloud. SSP-Pose [37] proposes a new symmetrical perception shape prior method and directly introduces it into the

pose estimation network. CATRE [38] designed a new method based on shape priors and target point cloud alignment, which creates a new optimization of the collected target pose. GeoReF [39] combines the learnable affine transformation on the basis of CATRE [38] to realize the processing of complex geometries. Recently, MH6D [40] proposed a consistent learning method based on multiple assumptions, which effectively overcame the uncertainties in the model by extracting and fusing multiple features. On the whole, although the above research can significantly raise the precision of pose estimation, it is time-consuming and laborious to establish a CAD model library and then train neural networks to obtain morphological prior information.

The shapeless prior methods do not rely on the shape prior and have a good generalization ability. FS-Net [41] uses 3D graph convolution to design a new framework, which consists of three parts: object detection network, shape-based network, and residual-based network. GPV-Pose [42] proposes a point cloud reconstruction method based on symmetry geometric perception and uses the edge box voting algorithm in the point direction to raise the geometric guidance capacity. On the basis of GPV-Pose [42], HS-Pose [43] proposes a mixed-region feature extraction model that can identify both global and local geometric structures and realizes the effective encoding of size and translation information. Based on a pre-trained visual language model, CLIPose [44] fully excavates the rich semantic information of images, point clouds, and text, and uses a multi-modal contrast learning method to realize the expression of aligned images, point clouds, and text. AG-Pose [45] designs an example detection method to characterize the geometric relationship between samples through a series of sparse keypoints.

However, the above methods do not take into account the characteristics of different modal data but simply splice them, and some neighborhood feature information is not fully mined; we solve this problem perfectly by embedding the neighborhood information into the feature points and improve the accuracy of object pose estimation.

3. Method

In this section, we first introduce the architecture of the two-stage geometric neighborhood fusion network, as shown in Figure 2. Then, we describe a new strategy for cross-modal neighborhood information fusion. Next, we introduce a two-stage geometric information embedding module. Finally, we describe how to estimate object poses and how to supervise the model. We provide detailed explanations for each component.

3.1. Overview

First of all, our network framework was based on RGB-D images as input, and we used MaskRCNN [46] to segment each object, obtaining the category label and segmentation mask of each object. We inputted each segmented object into the neighborhood information fusion module and obtained a new fusion feature F_{rgbd} after the personalized fusion of the point cloud and RGB features. We set up a shared category detector Q to adaptively detect keypoints according to the different instances features. Next, we inputted the fusion feature F_{agg} into the two-stage geometric information embedding module, and we obtained features $F_{stageII}$ that contained both local and global structures. All feature points went through the above operations. Finally, all feature point were inputted into the pose predictor module, and an Multilayer Perceptron (MLP) was used to forecast the NOCS [36] coordinates and return the final pose.

3.2. Neighborhood Information Aggregation Module

If we want to estimate the pose of different instances, we need to extract the sparse keypoints of different instances to represent their shape. As we all know, the ability of

sparse keypoints to express information is limited, so how to mine the feature information in sparse keypoints has become a difficult problem. Our method solves this problem by incorporating the neighborhood information of different modalities into feature points.

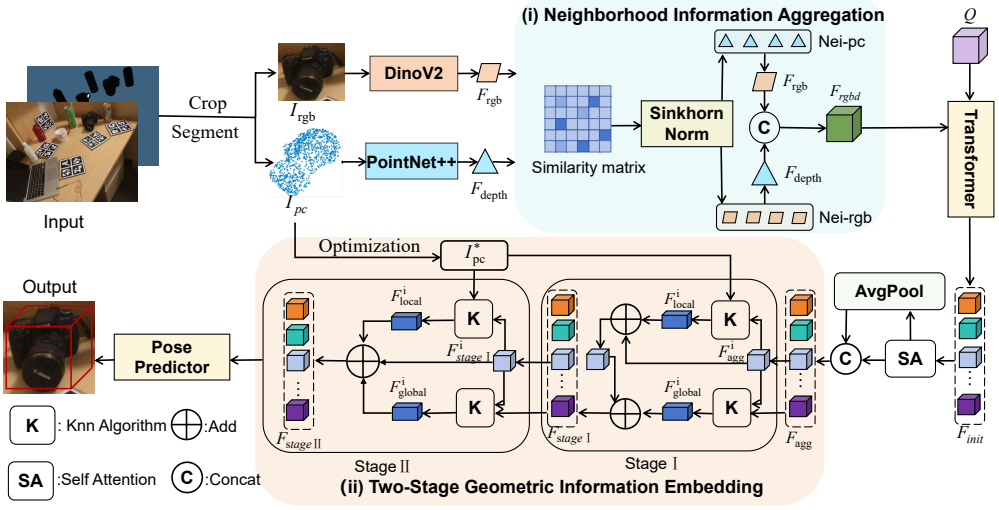


Figure 2. Framework of TGNF-Net. We input RGB-D images for cropping and segmentation and obtain F_{rgb} and F_{depth} through the feature extraction network. (i) Illustration of the Neighborhood Information Aggregation module. We calculate the similarity of F_{rgb} and F_{depth}. For a certain RGB feature point, F_{rgb} obtains the neighborhood point cloud feature points *Nei-pc* with the highest similarity. For a certain feature point of point cloud, F_{depth} obtains the neighborhood RGB feature points *Nei-rgb* with the highest similarity. *Nei-pc* and *Nei-rgb* are fused into F_{rgb} and F_{depth} respectively to obtain F_{rgbd}. We initialize a set of category-shared learnable queries Q and represent the keypoint detector. (ii) Illustration of Two-Stage Geometric Information Embedding module. In the *stageI*, for a certain feature point, we extract its local features in the point cloud and its global features inside the feature points. In the *stageII*, we use the NNS algorithm to expand the number of detected point clouds and internal feature points to double the previous number.

For each segmented object, we used different feature extractors to extract image features and point cloud features. For the inputted point cloud I_{pc}, we used PointNet++ [47] to extract the point cloud feature F_{depth}. For the inputted RGB image I_{rgb}, we used DINOv2 [48] to extract the image features and then select the pixels corresponding to the point cloud features to obtain the RGB feature F_{rgb}. After obtaining the features of these two modalities, we did not stitch them together directly. We applied a similarity measure to both of them. Specifically, for each feature pixel point of the RGB feature, we performed a similarity product with each feature point in the point cloud, and the similarity score was normalized. In the experiments, we combined the performance of multiple objects to set a threshold for the similarity score and take into account points with similarity scores above that threshold. We conducted multiple sets of experiments, and the detailed results are presented in Section 4.5 Ablation Studies. At the same time, we set no more than N feature points to participate in the fusion; if there were more than N neighborhood feature points included in the consideration range, we took the first N feature points with high similarity to be fused; if there were less than N neighborhood feature points included in the consideration range, we chose the nearest neighborhood points based on the current feature points to be supplemented. We named the N points with the highest similarity or the closest distance in the neighborhood as topN. Finally, we weighted these points based on the similarity scores to be fused with the feature pixel points. After each feature pixel

point was operated in this way, we obtained the RGB feature embedded in the point cloud neighborhood information. The same was performed for point cloud features. In detail:

$$\begin{cases} S_{r,j} = sim(F_{rgb}^{j}, F_{depth}), \\ F_{rgb}^{j} = MLP(softmax(S_{r,j}^{topN}) \cdot F_{depth}^{topN} + F_{rgb}^{j}), \end{cases} \tag{1}$$

$$\begin{cases} S_{d,i} = sim(F_{depth}^{i}, F_{rgb}), \\ F_{depth}^{i} = MLP(softmax(S_{d,i}^{topN}) \cdot F_{rgb}^{topN} + F_{depth}^{i}). \end{cases} \tag{2}$$

where F_{depth}^{i},F_{rgb}^{j} separately represent the ith point in the point cloud feature and the jth point in the image feature. $S_{d,i}$ represents the similarity score between the ith point in the point cloud feature and all the points in the image feature. $S_{d,i}^{topN}$ represents the topN similarity score in $S_{d,i}$, F_{rgb}^{topN} represents the N feature points corresponding to $S_{d,i}^{topN}$ in the image feature. $S_{r,j}$ represents the similarity score between the jth point in the image feature and all the points in the point cloud feature. $S_{r,j}^{topN}$ represents the topN similarity score in $S_{r,j}$, and F_{depth}^{topN} represents the N feature points corresponding to $S_{r,j}^{topN}$ in the point cloud feature.

In this way, for RGB features and point cloud features, each feature point contained the neighborhood information of other modal data, which improved the richness and comprehensiveness of features and captured finer local structures.

$$F_{rgbd} = concat(F_{rgb}, F_{depth}). \tag{3}$$

where F_{rgbd} is stitched from RGB features F_{rgb} and point cloud features F_{depth}. Each feature point in F_{rgb} and F_{depth} was obtained by embedding the neighborhood information of another modality. For objects with sparse feature points, this method provided rich neighborhood feature information for each feature point, and the ability of feature transmission information was enhanced.

3.3. Two-Stage Geometry Information Embedding Module

For category-level 6D pose estimation methods, we tend to estimate object that we have never seen within the class, but there is no instance CAD model, which leads to very inaccurate estimation results. Inspired by AG-Pose [45], we initialized a shared learnable instance detector Q, and we used transformer layers to combine the feature F_{rgbd} with the initial instance detector Q to generate a personalized instance detector Q_P. Then, the cosine similarity between the feature F_{rgbd} and Q_P was calculated to generate a keypoint heat map M. The keypoint KP and the feature F_{init} were obtained by a weighted summation in camera space.

$$KP = softmax(M) \cdot I_{pc}, \tag{4}$$

$$F_{init} = softmax(M) \cdot F_{rgbd}. \tag{5}$$

where KP and F_{init}, respectively, represent new keypoints and feature points. To enable the capability of perceiving contextual information and global information, we designed a self-global module. Inspired by the Graph Neural Network model (GNN) [49], we performed self-attention layers on F_{init}, which made the keypoints more closely integrated. After that, we performed an avgpool operation on F_{init} to extract its global features to obtain a new feature F_{agg} that could perceive the context information, which established the groundwork for the next integration of geometric information.

$$F_{agg} = concat(avgpool(self(F_{init})), self(F_{init})). \tag{6}$$

where F_{agg} is a new fusion feature that passes through the self-attention layer and average pooling layer.

To incorporate the geometric structure into the keypoint, we designed two stages to embed the geometric information. We used the ground-truth pose to convert point cloud I_{pc} into NOCS coordinates and removed outlier points in the mesh model to form a new valid point cloud I^*_{pc}. To ensure that point cloud I^*_{pc} was representative of the geometric structure of the object, we used a loss function L_{div} to keep the clustered keypoints away from each other. In addition, constraining the keypoints with a loss function L_{ocd} allowed it to be closer to the point cloud I^*_{pc} (see Section 3.4 Pose Calculation and Supervision, for details). For the *stageI*, firstly, we found feature point F^i_{agg} and the corresponding points in the point cloud I^*_{pc}. In the point cloud I^*_{pc}, we used the K-Nearest Neighbors (KNN) algorithm to find the K neighbors as F^i_l that were closest to the feature points F^i_{agg}, and we assigned corresponding weights to the K points according to the distance, recorded them as local geometric feature F^i_{local}, and finally embedded F^i_{local} into the corresponding features point F^i_{agg}. The specific formula was as follows:

$$F^i_l = KNN(I^*_{pc}) \quad F^i_{local} = softmax(F^i_l) \cdot F^i_l, \tag{7}$$

$$F^i_{agg} = MLP(F^i_{local} + F^i_{agg}). \tag{8}$$

where F^i_{agg} represents the *i-th* feature point of F_{agg}, and F^i_{local} represents the local feature of F^i_{agg}.

Once all feature points were processed as above, the next step was to extract the global geometry information. Unlike the above operations, we did not extract geometric information on the point cloud I^*_{pc} but between feature points F_{agg}. We found the closest K neighbors F^i_g to each feature point in F_{agg}. The remaining steps were similar to extracting local geometric information:

$$F^i_g = KNN(F_{agg}) \quad F^i_{global} = softmax(F^i_g) \cdot F^i_g, \tag{9}$$

$$F^i_{stageI} = MLP(F^i_{global} + F^i_{agg}). \tag{10}$$

where F^i_{global} represents the global feature of F^i_{agg}, and F^i_{stageI} is the *i-th* feature obtained by *stageI*. After all feature points were processed with the above steps, each feature point contained both global and local geometric embeddings.

$$F^i_l = concat(NNS(F^i_l), F^i_l) \quad F^i_{local} = softmax(F^i_l) \cdot F^i_l, \tag{11}$$

$$F^i_g = concat(NNS(F^i_g), F^i_{lg}) \;\; F^i_{global} = softmax(F^i_g) \cdot F^i_g, \tag{12}$$

$$F^i_{stageII} = MLP(F^i_{global} + F^i_{local} + F^i_{stageI}). \tag{13}$$

where $F^i_{stageII}$ is the *i-th* feature obtained by *stageII*. For the *stageII*, as shown in Figure 3, we used the Nearest Neighbor Search (NNS) algorithm to double the number of points and extract local feature F^i_{local} and global feature F^i_{global} based on these points. Finally, we used an MLP to fuse F^i_{global}, F^i_{local} and F^i_{stageI} together to form a new feature $F^i_{stageII}$.

In this way, for each feature point, the geometric structure was embedded into the feature point with the help of the local and global geometric position relationship. The geometric information could help establish the point-level correspondence and provided strong support for the subsequent pose estimation.

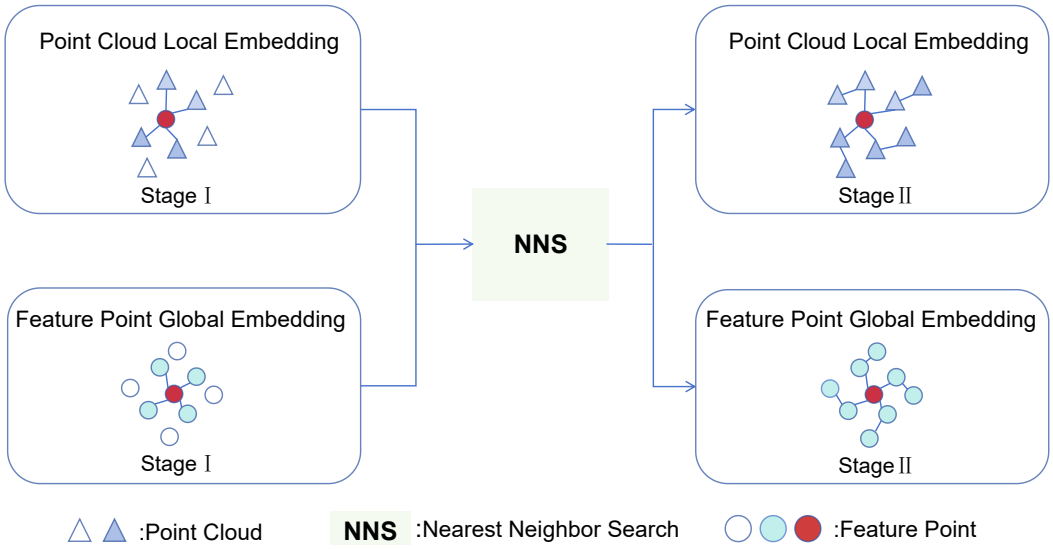


Figure 3. Two-stage geometric information embedding process. We used the NNS algorithm to double the number of points.

3.4. Pose Calculation and Supervision

In order to ensure that the extracted geometric features accurately described the geometric structure of the target, we needed to ensure that keypoints were uniformly scattered across the surface of the object. We used loss L_{div} to evenly fill the surface of the object by moving points that were clustered inside the object away from each other. The formula is shown below:

$$L_{div} = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} d(KP^i, KP^j), \tag{14}$$

$$d(KP^i, KP^j) = max\{th_1 - ||KP^i - KP^j||_2, 0\}. \tag{15}$$

where th_1 is a hyperparameter, and KP^i corresponds to the *i-th* keypoint. In the process of adaptive detection of object keypoints, we used the ground-truth pose to convert point cloud to NOCS coordinates and then compared with the mesh model to generate a valid point cloud I^*_{pc} within the mesh model. To guarantee the correctness of the detected keypoints, we removed the transformed outlier points. Finally, the keypoints were constrained by L_{ocd} to be close to I^*_{pc}. The details are as follows:

$$I^*_{pc} = \{X_i | X_i \in I_{pc} \quad and \quad \min_{Y_j \in Mesh} || \frac{1}{||S_{gt}||_2} R_{gt}(X_i - T_{gt}) - Y_j||_2 < th_2\}, \tag{16}$$

$$L_{ocd} = \frac{1}{KP} \sum_{x_i \in KP} \min_{y_j \in I^*_{pc}} ||x_i - y_j||_2. \tag{17}$$

where I^*_{pc} represents the converted point cloud, and th_2 is a hyperparameter. According to prior research [50], an *MLP* was used to forecast the NOCS [36] coordinates from Q_p. Through this set of keypoint level correspondence, the final pose and size R, T, S was returned. It is shown below:

$$KP^{nocs} = MLP(Q_p), \tag{18}$$

$$F_{pose} = concat(KP, F_{init}, KP^{nocs}, Q_p), \tag{19}$$

$$R, T, S = MLP_R(F_{pose}), MLP_T(F_{pose}), MLP_S(F_{pose}). \tag{20}$$

For rotation R, we refer to [51] for representation; for translation T, we refer to [43] to forecast the direct residual translation between point cloud and ground truth. The overall loss function is shown below:

$$LOSS = \alpha_1 L_{pose} + \alpha_2 L_{nocs} + \alpha_3 L_{div} + \alpha_4 L_{ocd}. \quad (21)$$

where α_1, α_2, α_3 and α_4 are hyperparameters used to balance the contribution of each item. For L_{pose}, we used the L1 loss. For L_{nocs}, we used R_{gt}, T_{gt}, and S_{gt} to project the coordinates of the keypoints within the camera space onto the NOCS to generate the ground-truth NOCS coordinates of keypoints KP^{gt}, and we used the SmoothL1 loss [50]. The details are as follows:

$$L_{pose} = ||R_{gt} - R|| + ||T_{gt} - T|| + ||S_{gt} - S||, \quad (22)$$

$$KP^{gt} = \frac{1}{||S_{gt}||_2} R_{gt}(KP - T_{gt}), \quad (23)$$

$$L_{nocs} = SommthL1(KP^{gt}, KP^{nocs}). \quad (24)$$

4. Experiments

4.1. Experiment Settings

Dataset. We undertook comprehensive experimental assessments to verify the effectiveness of our method across four datasets: REAL275 [36], CAMERA25 [36], HouseCat6D [52] and Omni6DPose [53]. The CAMERA25 datasets are synthetic datasets that were generated using mixed-reality methods. The CAMERA25 dataset contains six everyday objects, specifically 275K training objects and 25K test objects, which consist of a realistic background image and a compositely rendered foreground object. The REAL275 dataset is the actual dataset, which uses the same six object classes as the CAMERA25 dataset, including three unique instances in each category in the training set and test set, and each scene containing more than five objects. The HouseCat6D dataset is a multi-modal category-level 6D object perception dataset, which contains 194 family object categories with different luminosity complexities, covering a wide range of viewing angle distributions. The dataset contains RGB, depth, and RGB+P images for scenes with no texture, strong reflections, or translucent objects. Omni6DPose is a large-scale dataset with multiple object classes. It consists of three parts: ROPE (the real dataset for evaluation), SOPE (the simulated dataset for training), and PAM (the pose-aligned 3D models used in both ROPE and SOPE). The dataset used in the experiment included a variety of objects of different shapes, which fully verified the effectiveness of the method.

Metrics. We used the 3D IoU [36,42,54] and cm-deg [36,42,54] to assess the capability of the network. The IoU is the result of dividing the overlapping portion of two regions by the union portion of the two regions. For the IoU metric, we respectively calculated the average accuracy of two thresholds 50% and 75%. For the cm-deg metric, if the translation was less than cm and the rotation was less than deg, the prediction was deemed accurate. We measured the errors in rotation and translation below 5 cm–10 deg and 5 cm–5 deg, respectively, and then further narrowed the error to 2 cm–10 deg and 2 cm–5 deg, respectively.

4.2. Implementation Details

In terms of network design, we used MaskRCNN to split objects to obtain a split mask. In the neighborhood information fusion module, we took four point features in other modal data as neighborhood features for embedding, and the similarity threshold was set at 80%. For the two-stage geometric information embedding module, for the *stageI* geometric feature points, we chose K = 16, for the *stageII* geometric feature points, we

chose K = 32. The image was pre-processed and resized to 192 × 224, the point cloud was configured with 1024 points. In the training phase, we adjusted the batch size to 36 for synthetic data and 12 for real data, and we used two RTX3090Ti GPUs to train the network for about 20 h. During the testing phase, we tested two datasets separately with a single RTX3090Ti GPU.

4.3. Quantitative Experiments

Results on the REAL275 dataset. We compared our method with the current leading pose estimation method on different metrics and display the results in Table 1. Experimental results show that our method had superior performance on these two metrics.

Table 1. Comparison of our method with other advanced methods on REAL275 datasets. The best results are indicated in bold, and the second-best results are underlined.

Method	Year	IoU_{50}	IoU_{75}	2 cm–5 deg	5 cm–5 deg	2 cm–10 deg	5 cm–10 deg
NOCS [36]	2019	78	30.1	7.2	10.0	13.8	25.2
SPD [33]	2020	77.3	53.2	19.3	21.4	43.2	54.1
DualPoseNet [55]	2021	79.8	62.2	29.3	35.9	50.0	66.8
SGPA [56]	2021	80.1	61.9	35.9	39.6	61.3	70.7
GPV-Pose [42]	2022	-	64.4	32	42.9	-	73.3
SAR-Net [57]	2022	79.3	62.4	31.6	42.3	50.3	68.3
RBP-Pose [58]	2022	-	67.8	38.2	48.1	63.1	79.2
DPDN [50]	2022	83.4	76.0	46.0	50.7	70.4	78.4
IST-Net [59]	2023	82.9	76.6	47.5	53.4	72.1	80.5
Query6DoF [60]	2023	82.5	76.1	49.0	58.9	68.7	83.0
HoPENet [54]	2024	82.7	76.2	50.5	57.9	70.6	82.5
AG-Pose [45]	2024	**84.1**	<u>80.1</u>	<u>57.0</u>	<u>64.6</u>	<u>75.1</u>	**84.7**
Ours	-	<u>84.0</u>	**80.6**	**58.7**	**66.1**	**75.6**	<u>84.5</u>

In addition, our approach did not require the use of shape priors, which means that our model could handle the targets of a variety of shapes, not just predefined shapes. Compared with the AG-Pose [45], which achieved the best result in the current REAL175 dataset, it was evident that the precision of our method still outperformed AG-Pose by 0.5%, 1.7%, 1.5%, and 0.5% at the more stringent metrics of IOU_{75}, 2 cm–5 deg, 5 cm–5 deg, and 2 cm–10 deg respectively. Comparable performance on IOU_{50} and 5 cm–10 deg (only 0.1% and 0.2% lower). In summary, our approach achieved more leading results. This is because our method took into account more detailed neighborhood information, and the information provided by the local details of the neighborhood helped our pose estimation to be better on more stringent metrics.

Results on the CAMERA25 dataset. The results of our method compared with other methods on the second dataset CAMERA25 are shown in Table 2. It can be seen that for both the IOU metric and the cm-deg metric, our method was leading in performance. In detail, other methods did not take into account the importance of local information when fusing the two modal data features, instead simply stitch them, and there was no way to capture local information when extracting features from sparse point clouds. As can be seen from Table 2, compared to AG-Pose [45], which is currently the most effective method, our method improve by 0.9%, 0.3%, 1.2%, and 0.5% on the four metrics of 2 cm–5 deg, 5 cm–5 deg, 2 cm–10 deg, and 5 cm–10 deg respectively. This was due to our full attention to neighborhood information, which also proved the efficacy of our approach. In addition, our two-stage geometric information embedding also provided strong support for neighborhood information, complementing each other to fully explore the geometric position relationship between neighborhoods and making the sparse keypoints have more abundant information.

Table 2. Comparison of our method with other advanced methods on the CAMERA25 dataset. The best results are indicated in bold, and the second-best results are underlined.

Method	Year	IoU_{50}	IoU_{75}	2 cm–5 deg	5 cm–5 deg	2 cm–10 deg	5 cm–10 deg
NOCS [36]	2019	83.9	69.5	32.3	40.9	48.2	64.4
SPD [33]	2020	93.2	83.1	54.3	59.0	73.3	81.5
DualPoseNet [55]	2021	92.4	86.4	64.7	70.7	77.2	84.7
SGPA [56]	2021	93.2	88.1	70.7	74.5	82.7	88.4
SAR-Net [57]	2022	86.8	79.0	66.7	70.9	75.3	80.3
RBP-Pose [58]	2022	93.1	89.0	73.5	79.6	82.1	89.5
GPV-Pose [42]	2022	93.4	88.3	72.1	79.1	-	89.0
Query6DoF [60]	2023	91.9	88.1	78.0	83.1	83.9	90.0
HoPENet [54]	2024	92.2	88.4	78.4	**84.0**	83.9	90.4
AG-Pose [45]	2024	<u>94.2</u>	<u>92.5</u>	<u>79.5</u>	<u>83.7</u>	<u>87.1</u>	<u>92.6</u>
Ours	-	**94.3**	**92.6**	**80.4**	**84.0**	**88.3**	**93.1**

We performed a set of experiments, as shown in Table 3, comparing our method to other methods on six objects (bottle, bowl, camera, can, laptop, mug). The experiment was performed on the most stringent 2 cm–5 deg metric based on the CAMERA25 dataset. The results showed that our method was significantly better than other methods in the most stringent metric, which was due to our attention to neighborhood information and geometric features, and the capture of these details with a strict metric was beneficial to obtain accurate results, which proved the efficacy of our approach.

Table 3. Comparison of our method with other methods on different objects. The experiment was performed on the CAMERA25 dataset, and the evaluation metric was 2 cm–5 deg. The best results are in bold.

Method	Object Name						Mean
	Bottle	Bowl	Camera	Can	Laptop	Mug	
GPV-Pose [42]	79.5	89.6	53.8	90.6	68.3	50.9	72.1
Query6DoF [60]	85.7	95.4	59.5	96.6	73.9	57.0	78.0
AG-Pose [45]	87.0	97.1	61.2	98.1	75.3	58.3	79.5
Ours	**88.1**	**97.8**	**62.0**	**99.1**	**76.3**	**59.2**	**80.4**

4.4. Qualitative Experiments

We visually compared our method with other methods, and the comparison results are shown in Figure 4. The points in the above row represent the identified points of NOCS prediction errors, where green represents small errors, red represents large errors, and the gradual deepening of the color indicates that the error rate is gradually increasing. The bottom row represents the results of the ground truth and the predicted final pose. As illustrated in Figure 4, the result of our method on smooth objects was as good as that of other methods, because the geometric features extracted from smooth objects cannot fully exert their functions due to the limitation of the recognizable geometry structure. However, on large and textured objects such as laptops, our method could more accurately capture the contours, edges, and other features through geometric information embedding and could more accurately determine the position and posture during prediction. In addition, our method worked better than the other methods both in terms of predicting the error rate of NOCS coordinates. This is because when the extracted keypoints were sparse, our method could extract the key information of local details. The embedding of cross-modal neighborhood information and geometric information also provided a good basis for describing the position relationship of objects in space. Therefore, we had fewer error points in predicting NOCS coordinates.

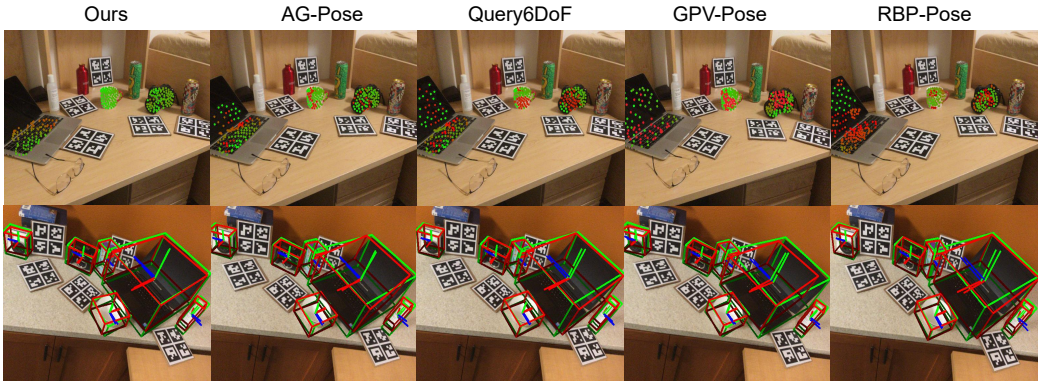


Figure 4. Visual comparison of our method with other methods on the REAL275 dataset. In the visualization, the green bounding boxes represent the ground truth, and the red bounding boxes represent the prediction. Green points represent small errors, and red points represent large errors.

To assess the generalization of our model, we performed experiments on HouseCat6D and Omni6DPose datasets. In both datasets, the objects used in the testing phase did not appear in the training phase, they merely belonged to similar categories as the training data. The results of the quantitative experiments are displayed in Table 4, and the results of the qualitative experiments are displayed in Figures 5 and 6. For the HouseCat6D dataset, we compared the two sequences from the test set with other methods in Figure 5 and can see that what our predictions were much closer to the ground truth. In 10 categories of objects, our method obtained results that were ahead of the other methods on almost every object. In addition, compared to other classical methods in Table 4, we had leading results for nine categories of objects. For the Omni6DPose dataset, we compared with other methods on multiple objects in Figure 6, and it is easy to see that our test results outperformed other methods for different shapes and types of objects. This shows that the generalization of our method was stronger than that of the other methods, which was due to our method focusing on the geometrical information, which made it less affected by categories, thus obtaining good results.

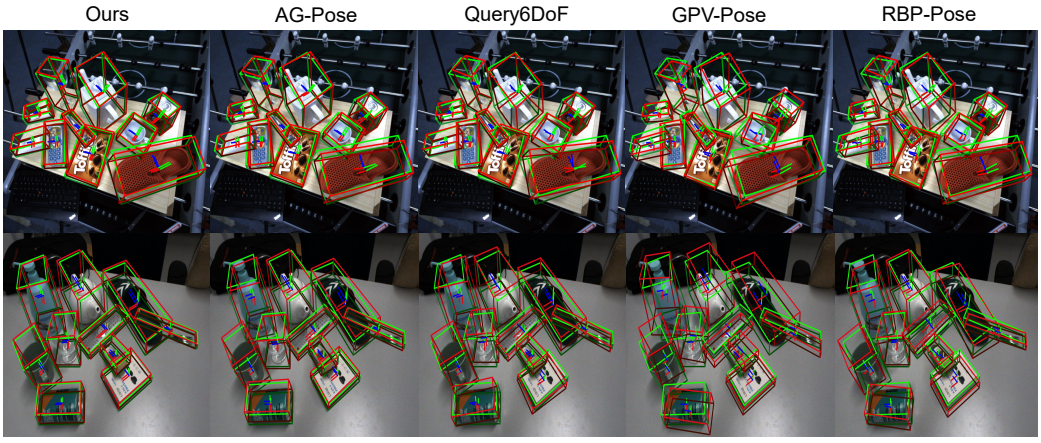


Figure 5. Visual comparison of our method with other methods on the HouseCat6D dataset. In the visualization, the green bounding boxes represent the ground truth, and the red bounding boxes represent the prediction.

Table 4. Class-wise evaluation of 3D IoU_{25} on the HouseCat6D dataset. The best results are in bold.

Method	Object Name										Mean
	Bottle	Box	Can	Cup	Remote	Teapot	Cutlery	Glass	Tube	Shoe	
NOCS [36]	41.9	43.3	81.9	68.8	81.8	24.3	14.7	95.4	21.0	26.4	50.0
GPV-Pose [42]	66.3	37.3	96.5	95.3	66.2	75.4	70.9	97.2	76.9	67.4	74.9
AG-Pose [45]	85.3	68.7	99.3	98.2	77.4	88.9	83.6	**98.7**	93.2	87.4	88.1
Ours	**86.1**	**74.4**	**99.6**	**99.1**	**81.6**	**90.2**	**80.4**	98.1	**95.5**	**88.1**	**89.3**

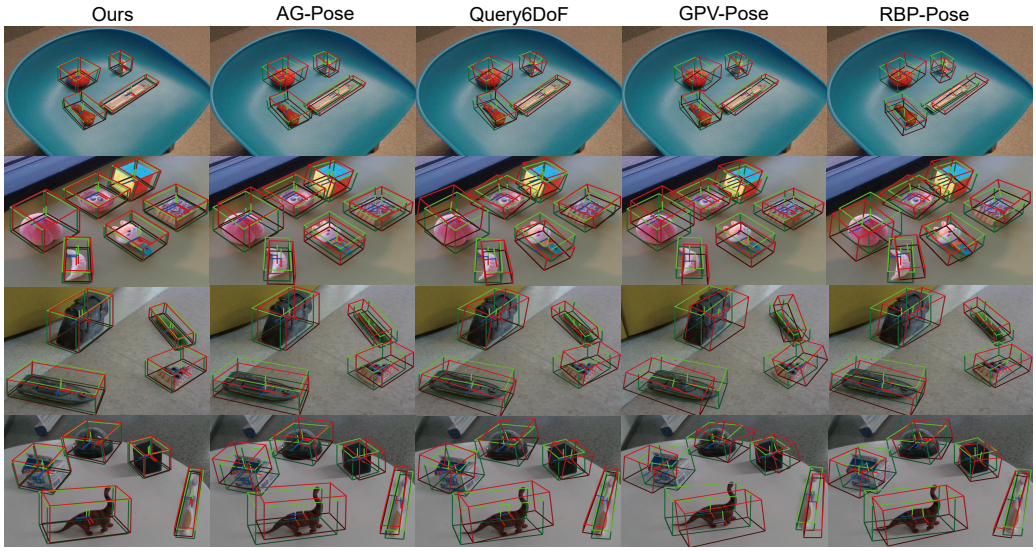


Figure 6. Visual comparison of our method with other methods on the Omni6DPose dataset. In the visualization, the green bounding boxes represent the ground truth, and the red bounding boxes represent the prediction.

We show a set of cm-deg metric from strict to loose trend graphs in Figure 7. We compared our method to AG-Pose [45] and Query6DoF [60] on six object classes (bottle, bowl, camera, can, laptop, mug), and all experiments were performed on the CAMERA25 dataset. The precision of our network was always at the leading level, which verified the effectiveness of our proposed method and showed that our method still had high stability even for different objects.

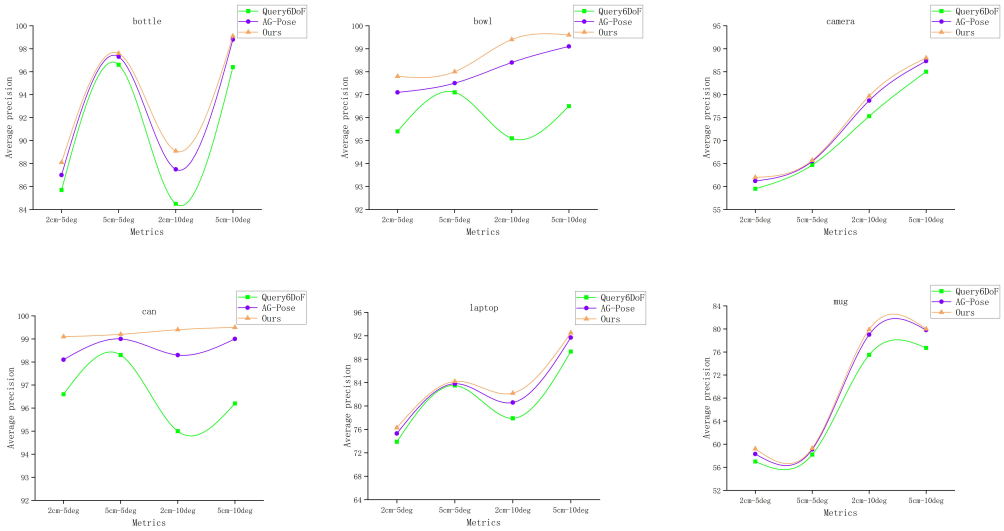


Figure 7. Comparison of our method with AG-Pose and Query6DoF on different objects. We show the trend plot of the precision obtained by different cm-deg thresholds on the CAMERA25 dataset.

4.5. Ablation Studies

In this section, we performed tests to verify the efficacy of the two-stage geometric information embedding module (TGIE) and the neighborhood information fusion module (NIF). In addition, we performed several ablation experiments by controlling the number of neighborhood feature points as well as the number of two-stage geometric feature points. Finally, we validated the effect of the similarity threshold on our model. The outcomes of the test are presented in Tables 5–8.

Table 5. We quantitatively validated the effectiveness of the neighborhood information fusion module and the two-stage geometric information embedding module on the CAMERA25 dataset. The best results are in bold.

Module	2 cm–5 deg	5 cm–5 deg	2 cm–10 deg
W/O NIF	79.5	83.6	87.3
W/O TGIE	78.8	82.7	87.0
All modules	**80.4**	**84.0**	**88.3**

Table 6. We validated the influence of neighborhood feature points in the neighborhood information fusion module on the CAMERA25 dataset. The best results are in bold.

NIF-Point	2 cm–5 deg	5 cm–5 deg	2 cm–10 deg
NP = 8	79.9	83.3	87.5
NP = 16	80.0	83.6	87.3
Ours (NP = 4)	**80.4**	**84.0**	**88.3**

Table 7. We used the 2 cm–5 deg metric to evaluate the influence of the similarity threshold in the neighborhood information fusion module on the CAMERA25 dataset. The best results are in bold.

Threshold	Object Name						Mean
	Bottle	Bowl	Camera	Can	Laptop	Mug	
90%	86.9	95.8	**62.7**	98.4	74.9	55.8	79.1
85%	88.0	94.6	61.1	98.2	75.1	**60.1**	79.5
80%	**88.1**	**97.8**	62.0	**99.1**	**76.3**	59.2	**80.4**
70%	87.1	92.4	57.6	97.8	75.7	58.9	78.3

Table 8. We validated the influence of two-stage geometry feature points in the two-stage geometric information embedding module on the CAMERA25 dataset. S1 represents *stageI*, and S2 represents *stageII*. The best results are in bold.

TGIE-Point		2 cm–5 deg	5 cm–5 deg	2 cm–10 deg
S1 = 16	S2 = 16	80.0	83.5	87.8
S1 = 32	S2 = 32	79.9	83.8	87.7
S1 = 16	S2 = 32	**80.4**	**84.0**	**88.3**

Effect of the neighborhood information fusion module (NIF) and two-stage geometry information embedding module (TGIE). We validated the neighborhood information fusion module and the two-stage geometric information embedding module in the model, and the outcome demonstrated the efficacy of these two modules. As shown in Table 5, we performed different experiments with the removal of the two modules. In the first set of experiments, we removed the neighborhood information fusion module, and the results showed that the accuracy of three metrics (2 cm–5 deg, 5 cm–5 deg, and 2 cm–10 deg) decreased by about 0.9%, 0.4%, and 1%, which indicated that our neighborhood information fusion module was effective. In the second set of experiments, we removed the two-stage

geometric information embedding module, and the results showed a decrease of 1.6% on the most stringent 2 cm–5 deg metric and 1.3% on the other metrics, which was a good verification of the effectiveness of our two-stage geometric information embedding module. In Figure 8, we can see the impact of the two modules more intuitively.

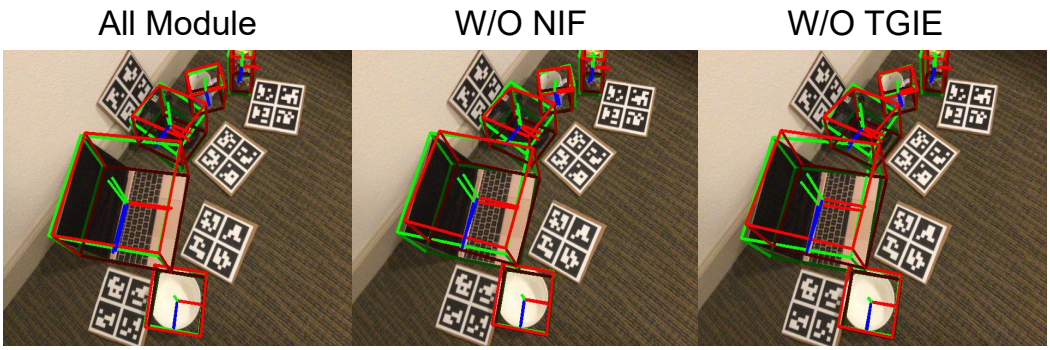


Figure 8. Visual comparison of our method with the other two modules. In the visualization, the green bounding boxes represent the ground truth, and the red bounding boxes represent the prediction.

Effect of neighborhood feature points in the neighborhood information fusion module. We conducted three sets of experiments based on the number of neighborhood feature points, and the results of the experiments are shown in Table 6. In our experiments, we set the number of points in the point cloud to 1024. As can be seen from Table 6, our method showed very good performance when fused with very few neighborhood feature points. With the increase in the number of neighborhood feature points, we speculated that the redundancy and noise information would gradually increase, and the performance of the model would decrease. However, the performance was stronger with fewer neighborhood feature points, which was enough to prove the superiority of our model.

Effect of the similarity threshold in the neighborhood information fusion module. We set the similarity threshold to gradually decrease and conducted four sets of experiments. As shown in Table 7, similarity thresholds that were too high or too low were not beneficial to our model. This is due to the fact that the threshold was too low to take into account the keypoints that contained too much noise. Conversely, when the threshold was too high, it resulted in the exclusion of some keypoints that contained important features, which was not conducive to the fusion of neighborhood information by the model. Taking into account the synthesis of the performance of various objects, we set a threshold of 80%, which was the most beneficial setting for our model.

Effect of the geometry feature points in the two-stage geometry information embedding module. In order to verify the influence of the two-stage geometric feature points on the model, we conducted three different sets of experiments, the outcomes of the test are presented in Table 8. In the first set of experiments, we set the number of geometric keypoint to 16 for the two stages, and the 2 cm–5 deg, 5 cm–5 deg, and 2 cm–10 deg metrics separately decreased by 0.4%, 0.5%, and 0.5% compared to our setup. This was due to the fact that few keypoints resulted in the geometric information not being fully expressed. In the second set of experiments, we set the number of geometric keypoint to 32 for the two stages, and the three metrics separately decreased by 0.5%, 0.2%, and 0.6% compared to our setup. This was due to the fact that there were too many geometric keypoints that contained noisy information, which affected the performance of the model. In the third set of experiments, the number of geometric keypoints for *stageI* and *stageII* was set to 16 and 32 respectively, which was obtained the best performance. This setting was also used in our model.

5. Conclusions

In this paper, we proposed a new two-stage geometric neighborhood fusion network for category-level 6D pose estimation. This paper solved the problems of insufficient feature fusion and poor generalization of the model. The neighborhood information fusion module allowed the neglected neighborhood feature to be embedded into the keypoints, making the feature information contained in the keypoints richer and improving the accuracy of pose estimation. In addition, we designed a two-stage geometric information embedding module, and we used geometric features to represent the structure of objects, showing strong generalization for objects with different shapes. Our approach did not require precise object CAD model, and it could handle objects that have not appeared in the training phase. Experiments showed that our method obtained better performance than other category-level object pose estimation methods.

Future work. We will consider the research and optimization of object-adaptive similarity thresholds in the neighborhood information fusion module, which will make the fusion target of each object more explicit.

Author Contributions: Conceptualization, G.Z., F.Y. and C.W.; methodology, X.Z.; software, X.Z.; validation, X.Z.; formal analysis, F.Y. and G.Z.; investigation, X.Z.; resources, G.Z.; data curation, X.Z.; writing—original draft preparation, X.Z.; visualization, X.Z.; supervision, G.Z., F.Y. and C.W.; project administration, G.Z.; funding acquisition, G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Basic Resources Investigation Program of China (No. 2019FY100900), the National Natural Science Foundation of China (No. 62176018) and the Young Elite Scientist Sponsorship Program by BAST (No. BYESS2023130).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hughes, J.; Abdulali, A.; Hashem, R.; Iida, F. Embodied artificial intelligence: Enabling the next intelligence revolution. *IOP Conf. Ser. Mater. Sci. Eng.* **2022**, *1261*, 012001. [CrossRef]
2. Deng, X.; Xiang, Y.; Mousavian, A.; Eppner, C.; Bretl, T.; Fox, D. Self-supervised 6d object pose estimation for robot manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation, Paris, France, 31 May–31 August 2020; pp. 3665–3671.
3. Su, Y.; Rambach, J.; Minaskan, N.; Lesur, P.; Pagani, A.; Stricker, D. Deep multi-state object pose estimation for augmented reality assembly. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality Adjunct, Beijing, China, 10–18 October 2019; pp. 222–227.
4. Tekin, B.; Sinha, S.N.; Fua, P. Real-time seamless single shot 6d object pose prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 292–301.
5. Doosti, B.; Naha, S.; Mirbagheri, M.; Crandall, D.J. Hope-net: A graph-based model for hand-object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6608–6617.
6. Encarnacao, J.L.; Lindner, R.; Schlechtendahl, E.G. *Computer Aided Design: Fundamentals and System Architectures*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
7. Lin, Y.; Tremblay, J.; Tyree, S.; Vela, P.A.; Birchfield, S. Single-stage keypoint-based category-level object pose estimation from an RGB image. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 1547–1553.
8. Lee, T.; Lee, B.U.; Kim, M.; Kweon, I.S. Category-level metric scale object shape and pose estimation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8575–8582. [CrossRef]

9. Fu, Y.; Wang, X. Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 27469–27483.
10. Chen, H.; Zendehdel, N.; Leu, M.C.; Moniruzzaman, M.; Yin, Z.; Hajmohammadi, S. Repetitive Action Counting Through Joint Angle Analysis and Video Transformer Techniques. In Proceedings of the International Symposium on Flexible Automation. American Society of Mechanical Engineers, Seattle, WA, USA, 21–24 July 2024; Volume 87882, p. V001T08A003.
11. Wang, P.; Jung, H.; Li, Y.; Shen, S.; Srikanth, R.P.; Garattoni, L.; Meier, S.; Navab, N.; Busam, B. Phocal: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 21222–21231.
12. Huang, J.; Yu, H.; Yu, K.T.; Navab, N.; Ilic, S.; Busam, B. Matchu: Matching unseen objects for 6d pose estimation from rgb-d images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 10095–10105.
13. Wang, P.; Ikeda, T.; Lee, R.; Nishiwaki, K. GS-Pose: Category-Level Object Pose Estimation via Geometric and Semantic Correspondence. *arXiv* **2023**, arXiv:2311.13777.
14. Liu, P.; Zhang, Q.; Zhang, J.; Wang, F.; Cheng, J. MFPN-6D: Real-time one-stage pose estimation of objects on RGB images. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 12939–12945.
15. Hu, Y.; Speierer, S.; Jakob, W.; Fua, P.; Salzmann, M. Wide-depth-range 6d object pose estimation in space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15870–15879.
16. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An accurate O (n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [CrossRef]
17. Li, Z.; Wang, G.; Ji, X. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 7678–7687.
18. Zakharov, S.; Shugurov, I.; Ilic, S. DPOD: 6D Pose Object Detector and Refiner. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 1941–1950.
19. Shugurov, I.; Zakharov, S.; Ilic, S. DPODv2: Dense Correspondence-Based 6 DoF Pose Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7417–7435. [CrossRef] [PubMed]
20. Huang, L.; Hodan, T.; Ma, L.; Zhang, L.; Tran, L.; Twigg, C.; Wu, P.C.; Yuan, J.; Keskin, C.; Wang, R. Neural correspondence field for object pose estimation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 585–603.
21. Jiang, H.; Dang, Z.; Gu, S.; Xie, J.; Salzmann, M.; Yang, J. Center-Based Decoupled Point-cloud Registration for 6D Object Pose Estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Vancouver, BC, Canada, 17–24 June 2023; pp. 3427–3437.
22. Besl, P.J.; McKay, N.D. Method for registration of 3D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*; SPIE: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
23. Lin, Y.; Su, Y.; Nathan, P.; Inuganti, S.; Di, Y.; Sundermeyer, M.; Manhardt, F.; Stricker, D.; Rambach, J.; Zhang, Y. Hipose: Hierarchical binary surface encoding and correspondence pruning for rgb-d 6dof object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 10148–10158.
24. Papaioannidis, C.; Mygdalis, V.; Pitas, I. Domain-translated 3D object pose estimation. *IEEE Trans. Image Process.* **2020**, *29*, 9279–9291. [CrossRef] [PubMed]
25. Li, Z.; Ji, X. Pose-guided auto-encoder and feature-based refinement for 6-dof object pose regression. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 8397–8403.
26. Deng, X.; Mousavian, A.; Xiang, Y.; Xia, F.; Bretl, T.; Fox, D. PoseRBPF: A Rao–Blackwellized particle filter for 6-D object pose tracking. *IEEE Trans. Robot.* **2021**, *37*, 1328–1342. [CrossRef]
27. Dang, Z.; Wang, L.; Guo, Y.; Salzmann, M. Match normalization: Learning-based point cloud registration for 6d object pose estimation in the real world. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46* , 4489–4503. [CrossRef] [PubMed]
28. Duffhauss, F.; Demmler, T.; Neumann, G. Mv6d: Multi-view 6d pose estimation on rgb-d frames using a deep point-wise voting network. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 3568–3575.
29. Cai, D.; Heikkilä, J.; Rahtu, E. SC6D: Symmetry-agnostic and correspondence-free 6d object pose estimation. In Proceedings of the 2022 International Conference on 3D Vision (3DV), Prague, Czech Republic, 12–16 September 2022; pp. 536–546.
30. Wen, Y.; Fang, Y.; Cai, J.; Tung, K.; Cheng, H. GCCN: Geometric constraint co-attention network for 6d object pose estimation. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, 20–24 October 2021; pp. 2671–2679.
31. An, Y.; Yang, D.; Song, M. HFT6D: Multimodal 6D object pose estimation based on hierarchical feature transformer. *Measurement* **2024**, *224*, 113848. [CrossRef]

32. Tan, T.; Dong, Q. SMOC-Net: Leveraging Camera Pose for Self-Supervised Monocular Object Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 21307–21316.
33. Tian, M.; Ang, M.H.; Lee, G.H. Shape prior deformation for categorical 6d object pose and size estimation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXI 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 530–546.
34. Wang, J.; Chen, K.; Dou, Q. Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4807–4814.
35. Fan, Z.; Song, Z.; Wang, Z.; Xu, J.; Wu, K.; Liu, H.; He, J. Acr-pose: Adversarial canonical representation reconstruction network for category level 6d object pose estimation. In Proceedings of the 2024 International Conference on Multimedia Retrieval, Phuket, Thailand, 10–14 June 2024; pp. 55–63.
36. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 2642–2651.
37. Zhang, R.; Di, Y.; Manhardt, F.; Tombari, F.; Ji, X. SSP-Pose: Symmetry-Aware Shape Prior Deformation for Direct Category-Level Object Pose Estimation. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 7452–7459.
38. Liu, X.; Wang, G.; Li, Y.; Ji, X. CATRE: Iterative Point Clouds Alignment for Category-level Object Pose Refinement. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 499–516.
39. Zheng, L.; Tse, T.H.E.; Wang, C.; Sun, Y.; Chen, H.; Leonardis, A.; Zhang, W.; Chang, H.J. GeoReF: Geometric Alignment Across Shape Variation for Category-level Object Pose Refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 10693–10703.
40. Liu, J.; Sun, W.; Liu, C.; Yang, H.; Zhang, X.; Mian, A. MH6D: Multi-Hypothesis Consistency Learning for Category-Level 6D Object Pose Estimation. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, 1–14. [CrossRef] [PubMed]
41. Chen, W.; Jia, X.; Chang, H.J.; Duan, J.; Shen, L.; Leonardis, A. 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, Nashville, TN, USA, 20–25 June 2021; pp. 1581–1590.
42. Di, Y.; Zhang, R.; Lou, Z.; Manhardt, F.; Ji, X.; Navab, N.; Tombari, F. 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, New Orleans, LA, USA, 18–24 June 2022; pp. 6781–6791.
43. Zheng, L.; Wang, C.; Sun, Y.; Dasgupta, E.; Chen, H.; Leonardis, A.; Zhang, W.; Chang, H.J. HS-Pose: Hybrid Scope Feature Extraction for Category-level Object Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 17163–17173.
44. Lin, X.; Zhu, M.; Dang, R.; Zhou, G.; Shu, S.; Lin, F.; Liu, C.; Chen, Q. CLIPose: Category-Level Object Pose Estimation with Pre-trained Vision-Language Knowledge. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 9125–9138. [CrossRef]
45. Lin, X.; Yang, W.; Gao, Y.; Zhang, T. Instance-adaptive and geometric-aware keypoint learning for category-level 6d object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 21040–21049.
46. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
47. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5105–5114.
48. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. Dinov2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
49. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
50. Lin, J.; Wei, Z.; Ding, C.; Jia, K. Category-level 6d object pose and size estimation using self-supervised deep prior deformation networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 19–34.
51. Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; Li, H. On the continuity of rotation representations in neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5745–5753.
52. Jung, H.; Wu, S.C.; Ruhkamp, P.; Zhai, G.; Schieber, H.; Rizzoli, G.; Wang, P.; Zhao, H.; Garattoni, L.; Meier, S.; et al. HouseCat6D-A Large-Scale Multi-Modal Category Level 6D Object Perception Dataset with Household Objects in Realistic Scenarios. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 22498–22508.

53. Zhang, J.; Huang, W.; Peng, B.; Wu, M.; Hu, F.; Chen, Z.; Zhao, B.; Dong, H. Omni6dpose: A benchmark and model for universal 6d object pose estimation and tracking. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2025; pp. 199–216.
54. Jiang, C.; Mu, X.; Zhang, B.; Xie, M.; Liang, C. Category Level Object Pose Estimation via Global High-Order Pooling. *Electronics* **2024**, *13*, 1720. [CrossRef]
55. Lin, J.; Wei, Z.; Li, Z.; Xu, S.; Jia, K.; Li, Y. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 3560–3569.
56. Chen, K.; Dou, Q. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 2773–2782.
57. Lin, H.; Liu, Z.; Cheang, C.; Fu, Y.; Guo, G.; Xue, X. Sar-net: Shape alignment and recovery network for category-level 6d object pose and size estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6707–6717.
58. Zhang, R.; Di, Y.; Lou, Z.; Manhardt, F.; Tombari, F.; Ji, X. Rbp-pose: Residual bounding box projection for category-level pose estimation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 655–672.
59. Liu, J.; Chen, Y.; Ye, X.; Qi, X. IST-Net: Prior-free category-level pose estimation with implicit space transformation. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 13932–13942.
60. Wang, R.; Wang, X.; Li, T.; Yang, R.; Wan, M.; Liu, W. Query6dof: Learning sparse queries as implicit shape prior for category-level 6dof pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 14055–14064.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.