Robotic Grasp Detection With 6-D Pose Estimation Based on Graph Convolution and Refinement

Sheng Yu, Di-Hua Zhai, Yuanqing Xia, *Fellow, IEEE*, Wei Wang, Chengyu Zhang, and Shiqi Zhao

***Abstract*—Six-dimensional (6-D) object pose estimation plays a critical role in robotic grasp, which performs extensive usage in manufacturing. The current state-of-the-art pose estimation techniques primarily depend on matching keypoints. Typically, these methods establish a correspondence between 2-D keypoints in an image and the corresponding ones in a 3-D object model. And then they use the PnP-RANSAC algorithm to determine the 6-D pose of the object. However, this approach is not end-to-end trainable and may encounter difficulties when applied to scenarios necessitating differentiable poses. When employing a direct end-to-end regression method, the outcomes are often inferior. To tackle the mentioned problems, we present GR6D, which is a keypoint- and graph-convolution-based neural network for differentiable pose estimation based on RGB-D data. First, we propose a multiscale fusion method that utilizes convolution and graph convolution to exploit information contained in RGB and depth images. Additionally, we propose a transformer-based pose refinement module to further adjust features from RGB images and point clouds. We evaluate GR6D on three datasets: 1) LINEMOD; 2) occlusion LINEMOD; and 3) YCB-Video dataset, and it outperforms most state-of-the-art methods. Finally, we apply GR6D to pose estimation and the robotic grasping task in the real world, manifesting superior performance.**

***Index Terms*—Convolution network, grasp detection, pose estimation, robot, transformer.**

I. Introduction

ROBOTIC grasp is a vital component of the manufacturing process [1]. To achieve accurate robotic grasp, several deep learning methods have been developed [2], [3], [4], [5], [6]. Among these methods, six-dimensional (6-D) object pose estimation is one of the most vital approaches [2], [7], [8]. The primary objective of the 6-D pose estimation task is to determine the rotation and translation components of an object located in three-dimensional (3-D) space. Classical 6-D pose estimation methods, such as those described in [9] and [10], usually rely on handcrafted features. These methods detect two-dimensional (2-D) keypoints in RGB images based on these features and then correspond these keypoints to predetermined 3-D keypoints. They use perspective-n-point (PnP) to ultimately estimate the 6-D pose. Unfortunately, when dealing with textureless objects or heavily occluded scenes, these methods often suffer from low accuracy [11].

In the past few years, there has been a significant exploration of deep learning-based methods for 6-D pose estimation. Various approaches, such as those presented in [2], [11], [12], [13], and [14], have been extensively researched. These methods can be categorized into two groups based on the inputs they use: 1) RGB image-based 6-D pose estimation [2], [12], [15], [16] and 2) RGB-D image-based 6-D pose estimation [11], [14], [17]. Each group employs either direct regression or keypoint matching techniques to estimate the 6-D pose.

In the work of Xiang et al. [18], they proposed PoseCNN which extracts features from RGB images and directly regresses the translational and rotational components of the object's pose. While direct regression-based methods offer an end-to-end differentiable solution, they often have lower precision compared to keypoint-matching methods. As a result, current state-of-the-art techniques still rely on keypoint matching, particularly using RANSAC-based PnP [19] to match 2-D object keypoints with their corresponding 3-D keypoints in object models. This approach has been widely adopted by recent methods, such as [7], [17], [20], and [21].

Keypoint-based approaches for 6-D pose estimation have demonstrated higher accuracy compared to direct regression-based methods [22]. For example, in the work of Tekin et al. [7], they proposed YOLO6D, which utilizes Darknet as the backbone and segments the image into multiple cells. This network predicts the centroid and eight corners of the 3-D bounding box within each cell and calculates the 6-D pose using PnP. Similarly, Peng et al. [20] proposed PVNet, where the position of 2-D keypoints is determined by voting at each pixel location in the image. These keypoints are then matched to 3-D counterparts and used to estimate the object's 6-D pose through PnP.

Manuscript received 20 May 2023; revised 14 October 2023; accepted 26 February 2024. Date of publication 19 March 2024; date of current version 17 May 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62173035, Grant 61803033, and Grant 61836001; in part by the "Xiaomi Young Scholars" from Xiaomi Foundation; in part by the China Unicom Innovation Ecological Cooperation Plan; and in part by the BIT Research and Innovation Promoting Project under Grant 2023YCXY035. This article was recommended by Associate Editor X. Wu. *(Corresponding author: Di-Hua Zhai.)*

Sheng Yu is with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: yusheng@bit.edu.cn).

Di-Hua Zhai is with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with the Yangtze Delta Region Academy Beijing Institute of Technology, Jiaxing 314001, China (e-mail: zhaidih@bit.edu.cn).

Yuanqing Xia is with Zhongyuan University of Technology, Zhengzhou 450007, Henan, China, and also with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: xia_yuanqing@bit.edu.cn).

Wei Wang, Chengyu Zhang, and Shiqi Zhao are with the Research Institute of China United Network Communications Corporation Limited, Beijing 100176, China (e-mail: wangw558@chinaunicom.cn; zhangcy365@chinaunicom.cn; zhaosq82@chinaunicom.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2024.3371580.

Digital Object Identifier 10.1109/TSMC.2024.3371580

However, keypoint-based methods also have some drawbacks. First, the 6-D poses estimated using these methods are nondifferentiable, which limits their learning ability. These methods first estimate the 2-D keypoints of the object and then determine the 6-D pose based on matching these 2-D keypoints with their corresponding 3-D keypoints using the RANSAC-based PnP algorithm. Since the PnP algorithm is nondifferentiable and there is no backpropagation of pose loss during the process of determining 6-D poses, this leads to the nondifferentiability of poses [12], [22]. Second, direct estimation of keypoints may introduce deviations between the directly estimated keypoints and the actual keypoints, resulting in errors in the final estimates of object poses. Third, the PnP-RANSAC iteration process can be time-consuming when dealing with multiple objects and multiple keypoints [12], [22]. Finally, in cases where RGB-D-based methods are used, most methods do not take into account the geometric and topological relationships between points in the point cloud.

To tackle the challenges associated with existing approaches, this article introduces GR6D, a novel 6-D pose estimation network that leverages graph convolution and refinement modules. The proposed GR6D network takes RGB images and point clouds as inputs, considering the geometric relationships between points. In order to extract and integrate information from both modalities, we introduce a cross-fusion technique using a graph convolutional neural network. Additionally, we employ a transformer-based feature refinement module to enhance the extracted features. In contrast to conventional methods, GR6D predicts the offset of each point in the scene relative to the keypoint, instead of directly determining the keypoints' locations. This strategy helps mitigate potential inaccuracies in location estimation. Furthermore, to address issues, such as time consumption and nondifferentiability, associated with the RANSAC-based PnP iterative process, we introduce an singular value decomposition (SVD)-based 6-D pose estimation method. This approach significantly reduces processing time and ensures differentiability during the pose estimation process.

In summary, the contribution of this article mainly consists of

1) We propose GR6D, a network for 6-D pose estimation, which incorporates a fusion method based on graph convolutional neural networks (GCN) and convolution. This fusion method leverages both RGB and point cloud data to improve the accuracy of pose estimation.
2) We propose a feature refinement module based on transformers, which can refine the fused features by adjusting attention while taking into account the information from both RGB images and point clouds.
3) We also propose a keypoint-based differentiable 6-D pose estimation method, which can realize an end-to-end pose estimation.
4) The experimental results on public datasets show that the GR6D gets better performance than most related methods. We also successfully apply the GR6D to the robotic grasping task in the real world, and get better performance than existing methods.

II. Related Works

A. Direct Methods

With the recent advancements in deep learning, several 6-D object pose estimation approaches have been proposed based on deep learning, including methods like [14], [15], [21], and [20]. Initially, regression-based methods were introduced, such as those in [14], [15], [18], and [23]. Kehl et al. proposed SSD-6D in [15], which is produced based on the SSD approach [24]. This method predicts 2-D bounding boxes and estimates the rotation component in one step, followed by predicting the translation component of 6-D pose based on the bounding box, which is then refined using a pose refinement technique. Xiang et al. [18] proposed PoseCNN, a 6-D estimation network that utilizes RGB images as input and VGG [25] as the backbone for feature extraction. To estimate the rotation part, regression is used, while voting is employed to estimate the translation part of the 6-D pose. Finally, Wang et al. [14] proposed DenseFusion, which offers a new perspective to estimate the 6-D pose based on RGB-D images. DenseFusion extracts features from both RGB and depth images, using a dense fusion approach to combine their features. The 6-D pose is estimated using direct regression based on the fused features.

B. Indirect Methods

For 6-D pose estimation, indirect methods establish 2-D–3-D correspondences of keypoints and utilize a variant of the RANSAC-based PnP method to predict the pose. These methods have gained popularity, with several works, such as [2], [17], [20], [21], [22], and [26]. Some methods use the eight vertexes of the 3-D bounding box as keypoints, such as those in [2], [7], and [27]. Rad and Lepetit proposed BB8 in [27], which takes RGB images as input and estimates eight keypoints of the 3-D bounding box based on a convolutional neural network (CNN). Tekin et al. [7] proposed YOLO-6D based on the concept of YOLO [28], [29], and estimate the position of eight keypoints. However, since the keypoints are located outside the surface of the object, the estimated positions may be inaccurate, which can lead to errors in pose estimation.

To address this issue, many researchers have explored defining keypoints on the surface of the 3-D model. For instance, He et al. [17] introduced PVN3D, which estimates keypoints through voting and utilizes semantic segmentation to improve prediction accuracy. Peng et al. [20] proposed PVNet, which takes RGB images as input and predicts keypoints based on voting. Subsequently, the 6-D pose is predicted using PnP with the 2-D–3-D keypoint correspondence. Additionally, Cao et al. [22] presented DGECN, a depth-guided 6-D pose estimation network that leverages edge convolution to extract geometric and topological information. Through dynamically learning the graph PnP using the correspondences of 2-D–3-D keypoints, DGECN predicts the 6-D pose. Another approach is presented in [30] by Zakharov et al., where DPOD predicts dense 2-D–3-D correspondences between pixels in the image and the 3-D model, enabling the calculation of the object's 6-D pose based on these correspondences.

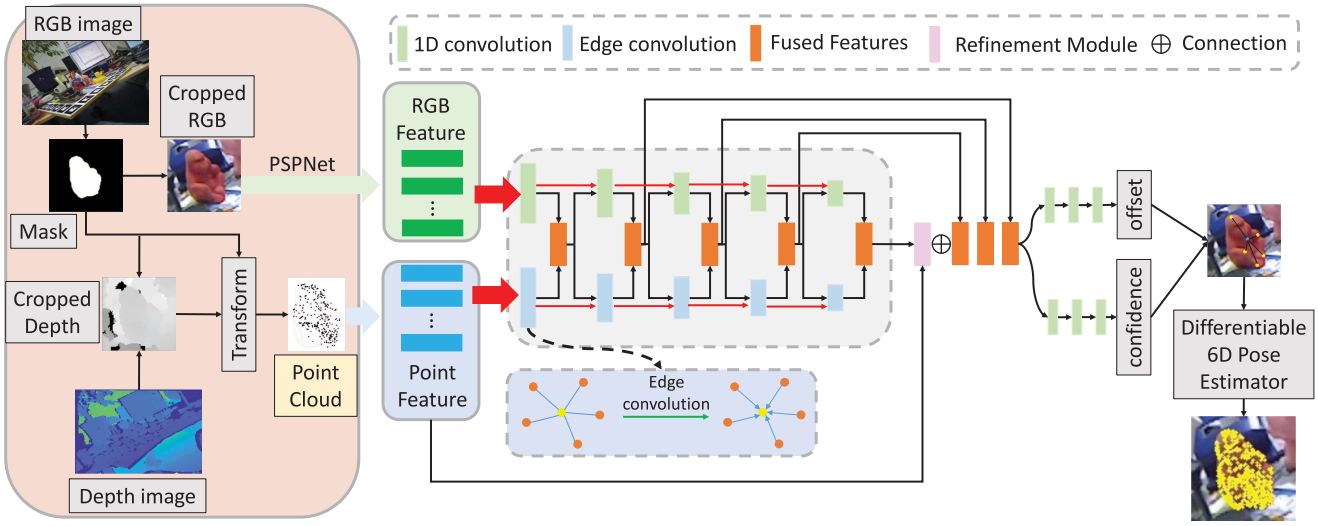


Fig. 1. Overview of the structure of the GR6D.

Nevertheless, due to the nondifferentiable nature and computational cost of the PnP-RANSAC algorithm, several enhanced methods have been proposed in recent studies, including [12], [13], [26], [31], [32], and [16]. Hodan et al. [31] introduced EPOS, which predicts 2-D–3-D correspondences of keypoints and employs a novel PnP-RANSAC method for 6-D pose estimation. Similarly, Hu et al. [16] proposed a PointNet-like [33] architecture that utilizes 2-D–3-D keypoint correspondences to estimate the object's 6-D pose. Furthermore, Wang et al. present GDR-Net in [12], which predicts object surface regions and employs Patch-PnP to calculate the 6-D pose. Moreover, Hua et al. propose REDE in [32], where confidence is utilized to refine the object's pose and perform end-to-end pose calculation.

C. Robotic Grasp Detection Based on 6-D Pose Estimation

Researchers have also explored the application of 6-D pose estimation in robotic grasp detection. A notable example is the "deep object pose estimation (DOPE)" proposed by Tremblay et al. [2], which predicts the eight corners and a center point of the object from RGB images and estimates the 6-D pose using 2-D–3-D keypoint correspondences. Additionally, Wang et al. introduced Densefusion in [14], which takes RGB-D images as input, extracts features from both RGB and point cloud using PSPNet [34] and PointNet [33], respectively, and predicts the 6-D pose directly using fused features from both networks. Both methods, DOPE and DenseFusion, have shown promising results in real-world robotic grasping tasks. Zhang et al. proposed FastNet in [8] inspired by DOPE and create a fast pose estimation network capable of approximating object poses in the real world. Finally, they apply FastNet to the robotic grasp task in the real world. Recognizing grasping objects based on 6-D pose estimation is critical for robotic grasp detection. Achieving accurate and rapid object pose estimation is an important issue, and this article aims to develop an accurate and rapid object pose estimation network that can be deployed for real-world robotic grasp tasks.

III. Method

Given an RGB-D image, the network's objective is to predict a transformation matrix that will translate the object from its coordinate system to the camera coordinate system. This transformation matrix can be separated into two components: 1) the rotation matrix $R \in SO(3)$ and 2) the translation vector $t \in \mathbb{R}^3$. By leveraging information from RGB-D images effectively, the network aims to predict the accurate 6-D pose of the object.

A. Overview

In this article, we introduce GR6D, a novel differentiable network for estimating object 6-D pose. The architecture of GR6D is presented in Fig. 1. We first use the Mask-RCNN [35] to perform the object detection and segmentation on the RGB image. Then we crop the RGB-D image with the generated segmentation mask of the target object. To extract features from RGB-D images, we first use a CNN to extract features from the RGB image $I \in \mathbb{R}^{H\times W\times 3}$. For the point cloud $P_o \in \mathbb{R}^{N_o\times 3}$ generated from the depth image, we incorporate geometric information and topology cues between points in the cloud by using a GCN, with dynamic graph CNN (DGCNN) [36] serving as the backbone. The features extracted from both RGB and point cloud data are then fused and processed by a transformer-based refinement module. The network subsequently predicts confidence levels and offsets from scene points to key points. Unlike most keypoint-based methods, our network predicted 6-D pose using a differentiable method based on confidence levels and offset.

B. RGB-Point Cloud Feature Fusion

First, we take the RGB-D image and use a segmentation network to identify objects of interest in the scene. In this article, we employ PSPNet [34] to produce RGB feature embeddings $F_{rgb} \in \mathbb{R}^{N_o\times c}$. Then, we crop the original depth image according to the segmentation boundaries, ultimately

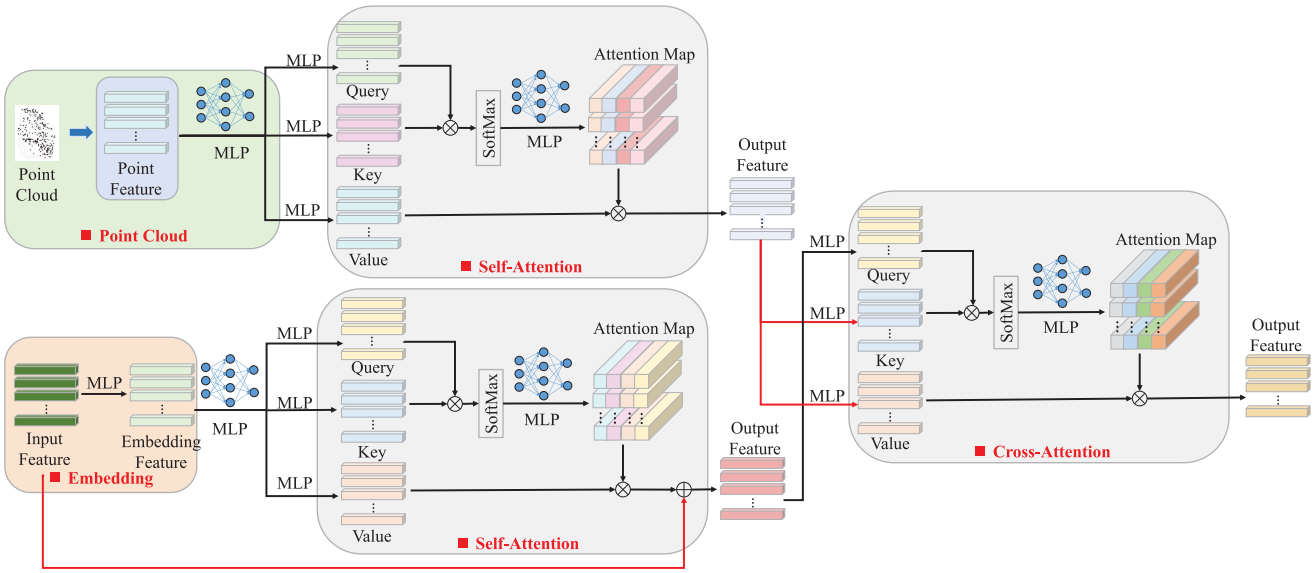


Fig. 2. Structure of the refinement module.

generating a new cropped depth image. To obtain geometric characteristics of the object, we utilize the camera intrinsic matrix M_i to transform the resulting image into a point cloud $P_o = [p_0, p_1, \ldots, p_{N_o}] \in \mathbb{R}^{N_o \times 3}$ consisting only of selected points, where N_o is the total number of points. Both the RGB features and point cloud data are then combined and utilized by GR6D as input.

We aim to predict offsets between scene points and key points based on the point cloud data, thus we must consider geometric and topological relationships between different cloud points. To realize this target, we utilize GCN for feature processing and extraction in the point cloud. GCN can effectively represent point clouds as graphs, and handle interpoint relationships well. We take P_o as the input, and use DGCNN [36] to perform the feature extraction. DGCNN extracts the features of the point cloud through GCN, and can demonstrate the point cloud as a graph. What is more, DGCNN considers the geometrical information and the topology information of the point cloud, and establishes the relationship between different points. So we use the DGCNN to extract the feature, and get the point cloud feature $F_o \in \mathbb{R}^{N_o \times c}$.

RGB images and depth measurements provide complementary information for object pose estimation. While depth measurements offer reliable geometric data, they can be affected by reflective surfaces or other factors, leading to missing information. Conversely, RGB images may capture objects that are not visible in depth images. Relying solely on RGB data poses challenges in achieving accurate 6-D pose predictions. To overcome this limitation, we propose a feature fusion approach that extracts features from both RGB and point cloud data. By leveraging the combined features from appearance and geometry, our network can effectively utilize both types of data to enhance the accuracy of object pose estimation.

In previous feature fusion frameworks like DenseFusion, the fusion of features only occurs at the final output of the backbone network, which is then used for pose estimation. However, CNNs possess distinct receptive fields at each layer, potentially causing the loss of local information from shallow layers if fusion is exclusively performed at the final layer. To overcome this limitation, we propose a novel cross-fusion method that efficiently combines features in complementary ways. This approach allows us to capture both local information from shallow layers and global information from deeper layers, resulting in enhanced pose estimation performance.

To combine features from both the RGB branch F^i_{rgb} and point cloud branch F^i_p, we concatenate them along their channel dimensions and use an multilayer perceptron (MLP) layer to refine the connected features, producing fused features

$$F^i_{\text{fuse}} = \text{MLP}\left(F^i_{rgb} \oplus F^i_p\right) \tag{1}$$

where $\oplus$ indicates channel dimension connection and i indicates the ith layer of the cross-fusion module.

Then, we generate the features at the next layer by the fused features with the original RGB and point cloud features, these operations can be calculated by

$$F^{i+1}_{rgb} = \text{MLP}\left(F^i_{rgb} \oplus F^i_{\text{fuse}}\right) \tag{2}$$

$$F^{i+1}_p = DG\left(F^i_p \oplus F^i_{\text{fuse}}\right) \tag{3}$$

where DG indicates the DGCNN.

In contrast to certain recent RGB-D-based methods, such as DenseFusion [14] and PVN3D [17], which perform feature fusion solely at the final layer of their networks, our GR6D incorporates this fusion process within every block throughout the network. By integrating feature fusion at multiple stages, our methodology enables richer feature fusion and multiscale receptive fields across all parts of the network. As a result, the fusion features generated by our approach exhibit increased complementarity, thereby facilitating more accurate predictions of 6-D object poses.

C. Refinement Module

By employing the cross-fusion module, we are able to effectively combine features extracted from both RGB images and point clouds. However, this process solely focuses on feature fusion. Since these fused features span across multiple layers, capturing multiscale object characteristics observed at varying receptive fields, it becomes crucial to refine these features by preserving the essential information needed for accurate pose estimation while discarding irrelevant details. In recent years, transformers have shown remarkable performance superiority over CNNs in computer vision tasks. This is attributed to their exceptional adaptability in adjusting network attention and capturing global features, resulting in significant improvements in network performance.

Therefore, to enhance the accuracy of refining the previously fused data, we present a novel refinement module that leverages transformers. As illustrated in Fig. 2, this module operates exclusively within the last layer of the network, where the fused features from this layer encompass both local-wise and global-wise features. The transformer demonstrates proficiency in extracting valuable information from feature maps and adjusting network attention. Hence, we employ the transformer to establish connections between the local-wise and global-wise features. Additionally, we incorporate point cloud features F_p into the refinement module to prevent the loss of crucial geometric feature information and facilitate feature fusion across both local-wise and global-wise features.

Our refinement module takes fused input features F_{fuse}, along with original point cloud features F_p, and combines them through two self-attention modules and a cross-module. First, we apply a series of MLP layers to generate query, key, and value vectors for our self-attention components

$$q_f, k_f, v_f = \text{MLP}(F_{\text{fuse}}) \tag{4}$$

$$q_p, k_p, v_p = \text{MLP}(F_p) \tag{5}$$

where q_*, k_*, and v_* indicate query, key, and value, respectively, f and p indicate the parameters of the fused feature and the point features.

Then we apply the self-attention module to refine and adjust the attention of features

$$F_{\text{fuse}} = \text{SoftMax}\left(\frac{q_f \times k_f^T}{\sqrt{d_f}}\right)v_f + F_{\text{fuse}} \tag{6}$$

$$F_p = \text{SoftMax}\left(\frac{q_p \times k_p^T}{\sqrt{d_p}}\right)v_p \tag{7}$$

where d_* is the dimension of k_*.

Our objective in using self-attention modules is to enhance and refine our previously obtained fused features without degrading their quality, as we connect these modules with a short connection. With this in mind, we implement cross-attention on the fused data via MLP layers. First we generate and process query parameters for F_{fuse}, and then subsequently extract key and value parameters from the original point cloud data through two additional MLP layers

$$q = \text{MLP}(F_{\text{fuse}}) \tag{8}$$

$$k, v = \text{MLP}(F_p). \tag{9}$$

After that, we use the cross-attention module to adjust and refine F_p, which can be calculated by

$$F = \text{SoftMax}\left(\frac{q \times k^T}{\sqrt{d}}\right)v \tag{10}$$

where F is the final output of the refinement module, and it will be connected with the previous fused features to perform the final 6-D pose estimation.

D. Differentiable 6-D Pose Estimator

Using the refined features, we predict the offsets from the scene points to each keypoint and determine their corresponding confidence levels. We then employ a differentiable estimation approach to calculate the 6-D pose.

1) Keypoints Selection: In this article, we apply a keypoint-based method for object 6-D pose estimation, which demonstrates more effectiveness and robustness when compared to direct regression methods. By incorporating depth information, this approach is also capable of capturing key geometric characteristics that bolster prediction accuracy. Before we can begin making predictions, it is necessary to select a series of typical keypoints. In our work, we follow the methodology presented by [20], which involves sampling keypoints $K_i, i \in \{1, \ldots, M\}$ from the 3-D model of our objects via the farthest point sampling (FPS) technique.

2) Offset Prediction: In previous keypoint-based methods, such as [17], [20], and [21], the training process lacks differentiability since they calculate the 6-D pose through 2-D–3-D keypoint matching without incorporating back-propagation for the predicted pose. Consequently, the network is unable to optimize the desired pose, leading to a potential decrease in detection accuracy. To tackle this problem, we introduce a novel approach for predicting the 6-D pose in this article.

Rather than directly predicting the position of our keypoints, we opt to predict their respective offsets from scene points. Suppose we are dealing with a given keypoint k_i within our scene, where $i \in \{1, \ldots, M\}$, we can define the offset between this point and a fixed scene point s_n through

$$o_{n,i} = s_n - k_i \tag{11}$$

where $o_{n,i}$ signifies the offset value between the fixed point s_n and keypoint k_i.

Due to real-world environmental constraints, we cannot always observe all keypoints within our scene, and some may even prove challenging to estimate accurately. To address this issue, we refine our previously determined offsets via predicted confidence values $C_{n,i}$. Specifically, these indicators represent the offset confidence between scene points s_n and keypoint k_i.

First, we normalize the confidence, and generate the normalized confidence by

$$c_{n,i} = \frac{C_{n,i}}{\sum_{n=1}^{N} C_{n,i}} \tag{12}$$

where $c_{n,i}$ is the normalized confidence.

Then, we can get the refined keypoint by

$$\hat{k}_i = \sum_{n=1}^{N} c_{n,i}(s_n + \hat{o}_{n,i}) \tag{13}$$

where $\hat{o}_{n,i}$ is the predicted offset from the scene point s_n to the keypoint k_i, $\hat{k}_i$ is the predicted keypoint.

3) Differentiable Pose Estimator: Previous methods calculate an object's 6-D pose using RANSAC-based PnP, which may prove time-consuming in instances where dense 2-D–3-D correspondences are necessary. To combat this drawback, we instead approach our pose estimation as an optimization problem. Given predicted 3-D keypoints $\hat{k}_i$ and the existing keypoint K_i on our 3-D model, our goal is to minimize the distance between $\hat{k}_i$ and transformed 3-D model keypoints. This target can be expressed mathematically as

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_{i=1}^{M} ||(R \cdot K_i + t) - \hat{k}_i||^2 \tag{14}$$

where R and t are the rotation component and translation component of the 6-D pose. We adopt SVD as our primary solution methodology. With this optimization challenge resolved, we can seamlessly integrate pose estimation processes into back-propagation routines for end-to-end training.

When processing RGB-D images, obtaining only one perspective often results in occlusions that hinder the accurate observation of certain object keypoints. Consequently, the network may struggle to predict these keypoints precisely, leading to persistently low-confidence levels. Nevertheless, we have noticed that confidence regarding visible keypoints tends to be more reliable. Building on this insight, we propose a novel method for pose estimation and refinement in this article.

Given a scene keypoint k_i, we can calculate its complete confidence by cumulating all the corresponding confidences from scene points to this keypoint, which can be calculated by

$$C_i = \sum_{n=1}^{N} C_{n,i}. \tag{15}$$

We set a certain threshold τ for C_i. When C_i surpasses τ, we then select the associated refined keypoint $\hat{k}_i$ to represent a candidate. In this manner, we develop a series of candidate keypoints denoted as $\hat{k}_j, j \in \{1, \ldots, J\}, J \leq M$. From there, given (14), we randomly choose three keypoints that fulfill SVD's minimum requirements, leading us to obtain $\mathcal{C}_J^3$ outputs, where $\mathcal{C}_*^*$ denotes the combination operator.

Our process yields a series of 6-D pose candidates, each comprising $\hat{R}_a$ and $\hat{t}_a$ for $a \in \{1, \ldots, \mathcal{C}_J^3\}$. Nevertheless, we must fine-tune these candidates to yield an optimal pose. Furthermore, we recognize that symmetrical objects tend to have multiple correct 6-D poses. To address this issue, we evaluate 6-D pose fitness based on the shortest distance between a scene point s_n and its nearest transformed neighbor p_n. Defining such distance as

$$d_n = \sum_{n=1}^{N} ||\hat{p}_n - s_n|| \tag{16}$$

where $\hat{p}_n = \hat{R} \cdot p_n + \hat{t}$, d_n is the distance between s_n and p_n.

Further, we generate the weight of each 6-D pose based on the distance. In this article, we use the softmax to calculate

$$w_a = \frac{e^{-\frac{d_a}{\lambda} - m}}{e^{-\frac{d_a}{\lambda} - m} + \sum_{j \neq a}^{\mathcal{C}_J^3} e^{-\frac{d_j}{\lambda}}} \tag{17}$$

where λ is a coefficient and m is a non-negative real number. Based on the weight, we can refine the object 6-D pose. For the translation part, the translation can be calculated by

$$\hat{t} = \sum_{a=1}^{\mathcal{C}_J^3} w_a \cdot \hat{t}_a. \tag{18}$$

Given that rotation space is a nonlinear domain, directly averaging calculated rotation matrices may violate underlying constraints of SO(3). To circumvent this issue, we convert our rotation matrix $\hat{R}_a$ into quaternion format $\hat{q}_a$. From there, we can derive refined rotations by using the following equation:

$$\tilde{q} = \sum_{a=1}^{\mathcal{C}_J^3} w_a \cdot \hat{q}_a \tag{19}$$

where $\tilde{q}$ is the refined rotation in the quaternion format.

After that, we normalize the $\tilde{q}$, which can be calculated by

$$\hat{q} = \frac{\tilde{q}}{||\tilde{q}||}. \tag{20}$$

Finally, we can get the estimated 6-D pose $\{\hat{q}, \hat{t}\}$, and transform the pose from the quaternion format $\hat{q}$ into the rotation matrices $\hat{R}$.

E. Loss Function

The loss function mainly contains two parts: 1) the offset loss and 2) the pose loss, which makes the 6-D pose differentiable.

1) Offset Loss: The offset loss contains the loss in three directions: x, y, and z. We use the smooth $L1$ loss as the loss function, which can be calculated by

$$L_{\text{offset}} = \sum_{i=1}^{M} \sum_{n=1}^{N} L_1\left(\Delta o_{n,i}^{(x)}\right) + L_1\left(\Delta o_{n,i}^{(y)}\right) + L_1\left(\Delta o_{n,i}^{(z)}\right) \tag{21}$$

where $\Delta o_{n,i}^{(*)} = \hat{o}_{n,i}^{(*)} - o_{n,i}^{(*)}$, $\hat{o}_{n,i}^{(*)}$ indicates the predicted offset, $o_{n,i}^{(*)}$ is the ground truth offset, $* \in \{x, y, z\}$ indicates x-axis, y-axis and z-axis, and L_1 is the smooth L1 loss.

2) Pose Loss: Since the pose estimator is differentiable, we can use a loss function to calculate and back propagate the error of the 6-D pose. The pose loss is defined as

$$L_{\text{pose}} = ||t - \hat{t}||_2 + \lambda ||R \cdot \hat{R}^T - I||_F \tag{22}$$

where $|| \cdot ||_2$ is the L_2 norm, $|| \cdot ||_F$ is the Frobenius norm, t and R are the ground truth for translational and rotational components, respectively, $\hat{t}$ and $\hat{R}$ are the prediction of translational and rotational components, respectively, and λ is the coefficient of rotational components.

3) Total Loss: The total loss of the network is defined as

$$L = L_{\text{offset}} + \alpha L_{\text{pose}} \tag{23}$$

where α is the coefficient of pose loss.

IV. Experiments

A. Dataset

We evaluate our network on three benchmark datasets.

1) LINEMOD Dataset: The LINEMOD dataset [10] is widely recognized as a prominent benchmark dataset for evaluating instance-level capabilities in object pose estimation tasks. The dataset encompasses thirteen low-textured objects depicted over 13 videos, with challenging scenes exhibiting cluttered environments and fluctuating lighting conditions. Each object in the dataset is labeled with both a 6-D pose and semantic mask. We follow the established procedures of prior research [14], [17], dividing the dataset into training and testing sets. During the training phase, we leverage roughly 180 real-world RGB-D images per category, while testing involves about one thousand images. To address the relative paucity of the training set, we generate several synthesized RGB-D images detailed in Section IV-B.

2) Occlusion LINEMOD Dataset: To expand the scope of the original LINEMOD dataset, Brachmann et al. [37] created the LINEMOD-occ dataset. Unlike its predecessor, the LINEMOD-occ dataset includes multiple annotated objects per scene, with intense occlusions present throughout. This heightened complexity renders object recognition more challenging overall.

3) YCB-Video Dataset: The YCB-Video dataset [18] consists of 21 objects that differ in size and shape. Across 92 recorded videos captured by an RGB-D camera in realistic settings, every RGB-D image is annotated with both a 6-D pose and instance semantic mask. Moreover, the YCB-Video dataset includes several challenging scenes replete with obstacles, such as image noise, occlusions, and variable lighting conditions. In this article, we follow the data split established in prior scholarship [14], [17], [18], dividing the dataset into designated sections for training and testing purposes.

B. Additional Dataset Generation

Despite being widely used, the three datasets mentioned above have relatively small sample sizes. This limitation, especially evident in the LINEMOD and LINEMOD-occ datasets, can result in overfitting when training network models directly. Accurately predicting an object's 6-D pose is crucial for effective robotic manipulation. To address this issue, we employ our 3-D object module to generate synthetic data by randomly selecting background images and objects and then rendering the latter onto the former. Our approach creates additional synthetic data by randomly selecting background images from SUN397 [38] and rendering all objects from both the LINEMOD and YCB-Video datasets onto them. The augmented dataset, which includes these rendered scenes, is shown in Fig. 3.

C. Evaluation Metrics

We follow the evaluation metrics used in most 6-D pose estimation methods, such as [11], [14], [18], and [21]. There are mainly two metrics: 1) the average distance (ADD) and 2) the ADD symmetric (ADD-S), which are proposed in [18].

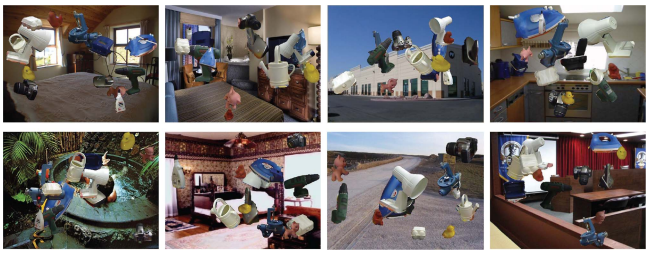


Fig. 3. Some examples of synthetic images on the LINEMOD dataset. We render the 3-D model of the object on the randomly selected images in SUN397.

1) ADD: The ADD calculates the distance between the points of the transformed 3-D model and the ground truth point. If the ADD is less than the predefined threshold, the prediction can be considered correct. The ADD is defined as

$$\text{ADD} = \frac{1}{m}\sum_{x\in M} ||(Rx+t) - (\hat{R}x+\hat{t})|| \tag{24}$$

where M is the set of points in the 3-D model, m is the number of points, R and t are the ground truths, and $\hat{R}$ and $\hat{t}$ are the predicted poses.

2) ADD-S: Since the symmetric objects contain more than one suitable pose, the ADD-S is also used in this article to evaluate the pose of symmetric objects, which is defined as

$$\text{ADD-}S = \frac{1}{m}\sum_{x_1\in M}\min_{x_2\in M} ||(Rx_1+t) - (\hat{R}x_2+\hat{t})|| \tag{25}$$

where x_1 and x_2 are the points on the 3-D module, R and t are the ground truths, and $\hat{R}$ and $\hat{t}$ are the predicted poses.

D. Training Details

In this article, we train the GR6D on Ubuntu 18.04, and use a single NVIDIA TITAN RTX GPU with 24G memory size. The deep learning architecture used in the experiment is PyTorch 1.5, and the version of CUDA is 10.2. The optimizer is Adam, and the learning rate is set as 1×10^{-4}. In the training process, the batch size is set as 8, and set as 1 in the evaluating process.

E. Evaluation on the LINEMOD Dataset and LINEMOD-Occ Dataset

Initially, we evaluate the performance of GR6D using both the LINEMOD dataset and LINEMOD-occ dataset.

The LINEMOD dataset consists of numerous scenes, each containing multiple objects. Our primary objective is to estimate the pose of the object situated at the center of each scene, as illustrated in Fig. 4. Notably, we solely label the target object with a 6-D pose, while other objects remain unlabeled. As the target object is minimally occluded, estimating its pose is relatively straightforward.

Real-world objects are often obscured, making it challenging to detect and accurately determine their pose. The LINEMOD dataset includes images where other objects are occluded, allowing researchers to assess network performance in cluttered scenes. To evaluate the performance of the

TABLE I
COMPARISON EXPERIMENTS ON LINEMOD DATASET

Object	RGB based methods					RGB-D based methods					
	PoseCNN[18]	PVNet[20]	CDPN[13]	DPOD[30]	DPVL[39]	SSD-6D[15]	DenseFusion[14]	PoseNet[40]	REDE[32]	PVN3D[17]	Ours
ape	77.0	43.6	64.4	87.7	69.1	65.0	92.3	85.0	95.6	97.3	**98.7**
benchvise	97.5	99.9	97.8	98.5	**100.0**	80.0	93.2	95.5	99.4	99.7	99.9
camera	93.5	86.9	91.7	96.1	94.1	78.0	94.4	91.3	99.6	99.6	**99.9**
can	96.5	95.5	95.9	99.7	98.5	86.0	93.1	95.2	99.5	99.5	**99.8**
cat	82.1	79.3	83.8	94.7	83.1	70.0	96.5	93.6	99.5	99.8	**100.0**
driller	95.0	96.4	96.2	98.8	99.0	73.0	87.0	82.6	99.3	99.3	**99.7**
duck	77.7	52.6	66.8	86.3	63.5	66.0	92.3	88.1	97.0	**98.2**	98.1
eggbox	97.1	99.2	99.7	99.9	**100.0**	**100.0**	99.8	99.9	100.0	99.8	**100.0**
glue	99.4	95.7	99.6	96.8	98.0	**100.0**	**100.0**	99.6	99.9	**100.0**	**100.0**
holepuncher	52.8	82.0	85.8	86.9	88.2	49.0	92.1	92.6	98.6	**99.9**	99.1
iron	98.3	98.9	97.9	**100.0**	99.9	78.0	97.0	95.9	99.3	99.7	99.1
lamp	97.5	99.3	97.9	96.8	99.8	73.0	95.3	94.4	99.3	99.8	**100.0**
phone	87.7	92.4	90.8	94.7	96.4	79.0	92.8	93.6	99.3	99.5	**99.7**
MEAN	88.6	86.3	89.9	95.2	91.5	79.0	94.3	92.9	98.9	99.4	**99.5**

TABLE II
COMPARISON EXPERIMENTS ON LINEMOD-OCC DATASET

Object	PoseCNN[18]	PVNet[20]	DPOD[30]	DPVL[39]	GDR-Net[12]	HybridPose[41]	REDE[32]	PVN3D[17]	Ours
ape	9.6	15.8	-	19.2	41.3	20.9	53.1	33.9	**59.8**
can	45.2	63.3	-	69.8	71.1	75.3	88.5	**88.6**	85.8
cat	0.9	16.7	-	21.1	23.5	24.9	35.9	39.1	**39.8**
driller	41.4	65.7	-	71.6	54.6	70.2	77.8	78.4	**82.6**
duck	19.6	25.2	-	34.3	41.7	27.9	46.2	41.9	**46.5**
eggbox	22.0	50.2	-	47.3	40.2	52.4	71.8	**80.9**	69.8
glue	38.5	49.6	-	39.7	59.5	53.8	75.0	68.1	**75.6**
holepuncher	22.1	39.7	-	45.3	52.6	54.2	**75.5**	74.7	73.4
MEAN	24.9	40.8	47.3	43.5	47.4	47.5	65.4	63.2	**66.7**

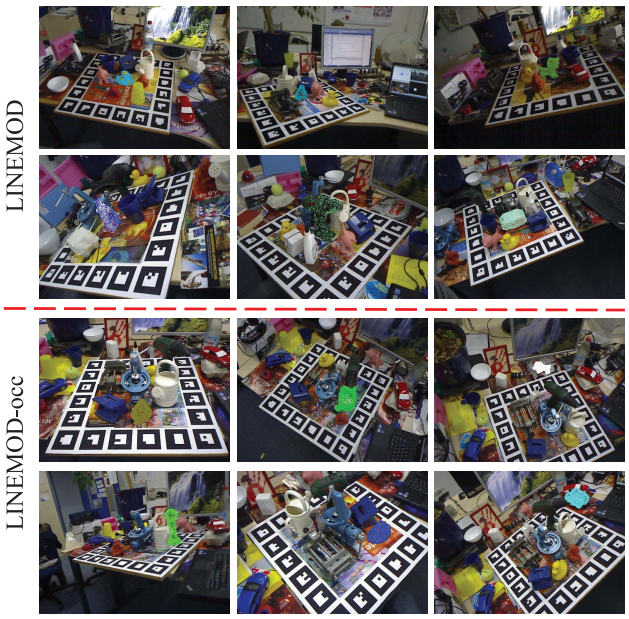


Fig. 4. Experimental results on the LINEMOD and LINEMOD-occ dataset.

network, researchers mark all the objects with a 6-D pose in a scene from the LINEMOD dataset. Fig. 4 depicts some detection results on the LINEMOD-occ dataset. To make the results more clear, only the pose of a single object is displayed on the image. The original image contains many occluded or partially occluded objects, simulating the process of object pose estimation in unknown scenarios.

Using a 3-D object model, we randomly sample 500 points and render them according to the predicted 6-D poses. If these rendered points align well with the object surfaces, we consider the network-generated 6-D pose to be accurate. As shown in Fig. 4, the majority of the points closely match their corresponding object surfaces, providing strong evidence for the effectiveness of our proposed 6-D pose estimation methodology.

In our evaluation, we compare GR6D with other relevant methodologies on both the LINEMOD dataset and LINEMOD-occ dataset. We present the results of this comparison, including the accuracy percentages for 6-D pose detection, in Tables I and II, respectively. As shown in these tables, GR6D surpasses the baseline and state-of-the-art models by a significant margin, demonstrating its superior performance.

Remark 1: In the evaluation process, we utilize the estimated object pose and the 3-D model to compute the transformed points. To assess the performance of the network, we measure the distance between these points. This article adopts the commonly used evaluation metrics of ADD and ADD symmetric (ADD-S) to evaluate the network's performance. If the calculated distance falls below a specified threshold, it indicates accurate predictions of the poses. This approach effectively facilitates the evaluation of the network's predictions.

To visualize the estimated pose, we render the object model onto the image. If the model fits well with the target object, we consider the prediction to be reasonably accurate. Although this method serves as a qualitative evaluation rather than a completely precise way of evaluating prediction results, it

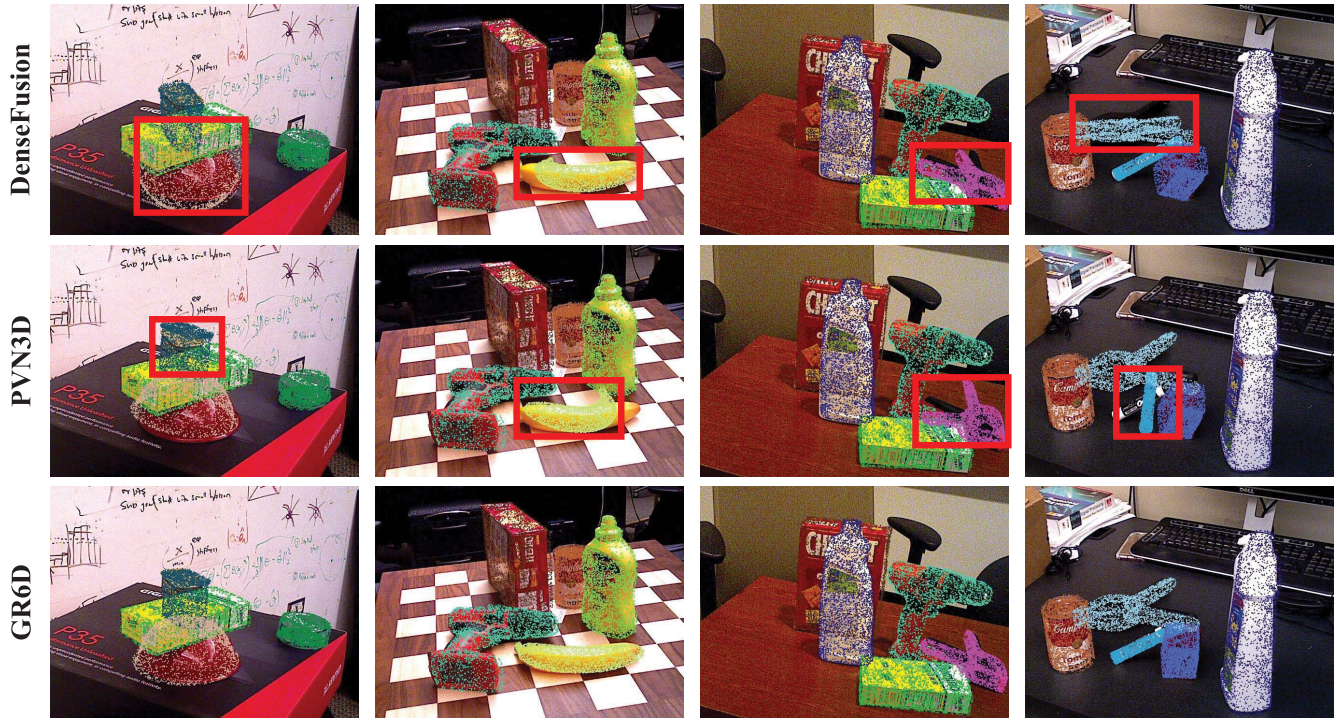


Fig. 5. Comparison experiments on YCB-Video dataset.

allows us to visualize how the prediction affects the image. If there are slight differences between the predicted and ground truth positions, we can easily discern these discrepancies from the image.

F. Evaluation on the YCB-Video Dataset

In our evaluation, we also compare GR6D with alternative RGB-D image-based methodologies. Specifically, we examine the performance of DenseFusion [14] and PVN3D [17], both of which have shown impressive results on the YCB-Video dataset. This dataset consists of a wide range of objects often occluded by other items in captured scenes. We showcase selected detection outcomes for each model in Fig. 5. Additionally, we render 1000 randomly chosen points from each object model under the estimated 6-D poses. Any incorrect projections, indicating misplaced detections, are highlighted within the red rectangle in Fig. 5. Notably, our proposed GR6D significantly outperforms DenseFusion and PVN3D in terms of accuracy across the diverse set of objects included in the YCB-Video dataset.

Subsequently, we compare the effectiveness of GR6D with other relevant methods using the ADD-S<2-cm metric previously employed by PoseCNN [18] and DenseFusion [14]. In this evaluation, a successful detection is defined as any instance where the predicted ADD-S is below 2 cm. The performance results are summarized in Table III, clearly demonstrating that our proposed methodology outperforms state-of-the-art models by a significant margin.

TABLE III
COMPARISON RESULTS ON YCB-VIDEO DATASET

Authors	ADD-S<2cm
PoseCNN[18]	93.2
DOPE[2]	93.1
DenseFusion[14]	96.8
REDE[32]	97.8
PVN3D[17]	97.6
Our	**98.9**

G. Experimental Analysis

DenseFusion utilizes PointNet [33] for feature extraction from point clouds, but it does not consider the topological and geometric relationships between points. Furthermore, this approach combines RGB and point cloud features without employing complementary fusion techniques. Importantly, it directly regresses 6-D pose estimations solely based on fused feature inputs, without considering salient keypoints on object models. As a result, the prediction accuracy is reduced.

Similar to DenseFusion, PVN3D also neglects the use of GCN in analyzing the geometric and topological relationships between points in the point cloud. Although PVN3D utilizes keypoints in its 6-D pose estimation methodology, this process is not differentiable, requiring the adoption of RANSAC-based PnP for calculating pose predictions. This approach can be time-consuming and may result in incorrect estimation outputs.

When compared with related methodologies, GR6D excels in its ability to handle 6-D pose estimation tasks within complex scenes.

First, to enhance the accuracy of pose estimation, we propose a novel cross-fusion module that effectively combines RGB image and point cloud features in a complementary manner. Unlike DenseFusion and PVN3D, which only perform fusion at the final output layer, our cross-fusion module enables information fusion at different receptive

TABLE IV
RESULTS OF ABLATION EXPERIMENTS ON THE LINEMOD DATASET

GCN	Complementary Fusion	Refinement Module	ADD-S$< 2\ cm(\%)$
	✓	✓	97.2
✓		✓	96.8
✓	✓		96.5
✓	✓	✓	**99.5**

fields. This allows the network to capture both local and global information, resulting in improved overall accuracy. Additionally, we utilize DGCNN to extract geometric and topological information from the point cloud, facilitating the establishment of relationships between different points. This relationship is crucial for accurately estimating the offset between keypoints and scene points.

Second, to further refine the fusion feature and improve the accuracy of pose estimation, we introduce a novel transformer-based refinement module as our second contribution. Even though the network has already fused features from the RGB image and point cloud, there remains a wealth of information to be extracted about the observed object. Hence, we devised a transformer-based refinement module to assist the network in discerning relevant information for accurate pose estimation. This module enables the network to adjust its attention, prioritizing important information while disregarding irrelevant details. Furthermore, we employ a cross-attention module to fuse the refined feature with the point cloud feature, ensuring no loss of information occurs during the fusion process.

Lastly, we present a novel differentiable 6-D pose estimation method that leverages keypoints. Traditional regression-based methods like DenseFusion offer differentiable training but suffer from low accuracy. On the other hand, keypoint-based methods like PVN3D rely on nondifferentiable PnP solutions for 6-D pose estimation. To resolve this limitation, we propose a unique approach where we convert the PnP problem into an optimization problem, making it differentiable. This enables object pose estimation during the training process while improving the network's ability to handle object pose based on keypoints.

H. Ablation Experiments

In order to assess the efficacy of our proposed GCN, complementary fusion, and refinement module, we perform ablation experiments on the LINEMOD dataset. In these experiments, we substitute GCN with PointNet and replace our complementary fusion approach with the simpler fusion protocol employed by DenseFusion. The performance results obtained from these experiments are documented in Table IV.

In our initial ablation experiment, we remove GCN and employ PointNet [33] for feature extraction. By removing GCN from the GR6D framework, there is a noticeable reduction in the extraction of geometric information from the point cloud. As a result, the network receives diminished aggregated input, leading to a decline in performance.

After removing our complementary fusion instance, we adopt the feature fusion method described in [14]. However, this change leads to a slight decline in network accuracy. The reason behind this is that the RGB and point cloud features fail to achieve the required complementarity and fusion across different feature scales, which is only accomplished in the later layers. Since this approach lacks comprehensiveness, it overlooks vital information from other scales, resulting in performance challenges within the network.

In the our latest experiment, we aim to eliminate the refinement module from our GR6D framework. The GR6D model takes RGB-D images as input and performs object pose estimation by leveraging information from these images. The RGB image offers texture information, while the depth image provides geometric details. By combining these two sources of information, the network improves its ability to handle object pose. To accomplish this, we propose a novel cross-fusion technique that integrates information from RGB and depth images at different layers throughout the network.

By employing the aforementioned fusion technique, we effectively combine features from RGB and depth images. Nonetheless, it is crucial to identify and prioritize the most effective features within the fused representation. To address this, we introduce a novel refinement module designed to refine and select the salient features within the fusion output. This module plays a crucial role in enhancing the overall performance of the system.

To evaluate the efficacy of the refinement module, we conducted ablation studies on our network. Upon removing the refinement module, the experimental results demonstrate a noticeable decline in network performance. While the GR6D framework incorporates feature fusion at earlier layers, the fused features remain in a preliminary state and require further refinement. Neglecting this refinement step can have detrimental effects on pose estimation. Without the inclusion of the refinement module, the network fails to effectively utilize these features, leading to a decrease in overall performance.

The addition of the refinement module to the network yields a significant enhancement in its performance. In this study, we employ a transformer for the refinement process. The transformer proves to be highly effective in capturing global information and adjusting the attention of the network. The incorporation of global information allows the network to better handle object pose estimation at a higher level. The attention mechanism enables the network to selectively focus on critical features while disregarding less relevant elements, such as background details.

The decrease in performance can be attributed to the absence of GCN or complementary fusion, which results in the fused features lacking complete complementation and refinement. The refinement module addresses this task by improving the quality of the fused features. However, with the removal of this module, there is no further opportunity for fusion, leading to a degradation in the accuracy of the network's output.

V. EXPERIMENTS IN THE REAL WORLD

A. Pose Estimation in the Real World

To assess the practical applicability of GR6D, we conduct 6-D pose estimation experiments. As obtaining LINEMOD

Fig. 6. Object pose estimation in the real world.

dataset objects for real-world testing can be challenging, we instead utilize readily accessible objects from the YCB-Video dataset. Specifically, we evaluate our model's performance on four carefully selected YCB-Video objects: 1) cracker box; 2) tomato soup can; 3) potted meat can; and 4) mustard bottle. Throughout our experimentation, we employ the Intel Realsense D435i camera to capture RGB-D information for input.

To enhance the interpretability of predicted 6-D poses, we adopt 3-D bounding boxes to represent them. If these boxes properly encompass the target objects, it indicates the corresponding network prediction is successful. To achieve a more comprehensive evaluation of performance, we conduct comparative experiments in real-world environments between DenseFusion, PVN3D, and GR6D. The outcomes of these trials are presented in Fig. 6.

To evaluate the network's performance in real-world scenarios, we strategically position objects to create more complex and challenging 6-D pose estimation tasks, deliberately hiding certain items from view. As illustrated in the first two columns of Fig. 6, our model demonstrates superior performance compared to both DenseFusion and PVN3D in terms of pose estimation. Moreover, the results in the last column of Fig. 6 indicate that even when dealing with cluttered scenes containing unrelated objects, GR6D exhibits greater proficiency over DenseFusion and PVN3D. This is attributed to GR6D's systematic approach in considering point-to-point relationships and object 6-D poses.

B. Robotic Grasp Experiments

In this article, we employ GR6D for object pose estimation and utilize a real Baxter robot to perform robotic grasping tasks. By leveraging the estimated 6-D pose and the 3-D model of objects, we can determine the central location of an object within the workspace. Our objective is to position the center point of the robot's gripper in close proximity to the center of the object, taking into account the gripper's rotation angle. To achieve this, we utilize the estimated 6-D pose to calculate the gripper's rotation angle, ensuring that it aligns vertically with the contact surface of the object's enclosing 3-D frame. Throughout the entire robotic grasping task, the gripper maintains a vertical orientation relative to the table.

Fig. 7. Robotic grasp experiment in the real world.

TABLE V
AVERAGE GRASP SUCCESS RATES IN THE ROBOTIC GRASPING EXPERIMENTS

Authors	Success Rates (%)
DOPE[2]	81.7
FastNet[8]	77.3
DenseFusion[14]	73.0
Our	83.5

We deploy the Intel Realsense D435i RGB-D camera for our experiments. All objects used in the experimental results are placed randomly, with corresponding detection results and grasping processes depicted within Fig. 7.

We define successful grasping within our framework as being able to lift and place an object to a predefined target position using our Baxter robot. The success rate associated with the GR6D model and related works is presented within Table V. Within this article, we conduct 50 robotic grasping experiments, several of which are unsuccessful.

Here, are some of the reasons that result in failed grasp.

1) Initially, when utilizing GR6D, it relies on RGB-D data captured by a camera. However, the depth information obtained from the camera can be subject to inaccuracies caused by environmental factors and camera noise, thereby affecting the accuracy of pose estimation. In turn, inaccurate pose estimation can have a detrimental impact on the robot's grasping precision, potentially leading to failed grasp attempts.
2) Second, this article primarily emphasizes object pose estimation and grasp detection, without placing significant emphasis on robust motion planning. Consequently, in cluttered scenes, there is a possibility of the robotic gripper colliding with other objects, ultimately leading to failed grasp attempts.
3) Finally, deviations between the actual grasping position and the calculated position may occur due to errors in both hand-eye calibration and robot motion positioning.

C. Future Work

In future work, we will try to solve these issues.

1) To begin with, our primary focus will be the development of an additional depth completion network. The purpose of this network is to refine the depth information, aiming to provide the primary network with accurate and reliable depth data.
2) Next, we will implement reliable and effective motion planning methods within the context of robotic grasping. By incorporating object pose information with suitable algorithms, our objective is to generate efficient paths that ensure the avoidance of any collisions between objects and grippers.
3) Finally, our goal is to further improve the accuracy of the robot grasping experiment by taking measures to reduce errors. To achieve this, we plan to perform multiple calibrations during the hand-eye calibration process and select the most accurate option as our calibration result. Moreover, if there are any motion position inaccuracies in the robot, we will apply position compensation techniques to adjust its movement or even consider replacing the robot with a more precise model entirely for the grasping task. Overall, our aim is to minimize the number of errors that may arise during the course of the robot grasping experiment.

VI. Conclusion

We propose a novel differentiable 6-D pose estimation method, GR6D. This state-of-the-art technique operates upon RGB-D images through unique methods to fuse features in an innovative complementary manner. Furthermore, the introduction of a new refinement module serves to enhance feature quality. Finally, we enable 6-D object pose estimation through our differentiable estimation model. Our analysis reveals that in comparison to competitive techniques, GR6D proves to be more accurate and robust when assessing occluded scenes. In addition, we apply our model toward real-world grasping tasks with a Baxter robot, where empirical results indicate that compared to alternative models, GR6D achieves significantly higher success rates.

References

[1] Y. Hu, Z. Li, G. Li, P. Yuan, C. Yang, and R. Song, "Development of sensory-motor fusion-based manipulation and grasping control for a robotic hand-eye system," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1169–1180, Jul. 2017.

[2] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Proc. 2nd Conf. Robot Learn.*, 2018, pp. 306–316.

[3] Y. Yu, Z. Cao, Z. Liu, W. Geng, J. Yu, and W. Zhang, "A two-stream CNN with simultaneous detection and segmentation for robotic grasping," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 2, pp. 1167–1181, Feb. 2022.

[4] S. Yu, D.-H. Zhai, Y. Xia, H. Wu, and J. Liao, "SE-ResUNet: A novel robotic grasp detection method," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5238–5245, Apr. 2022.

[5] A. Mousavian, C. Eppner, and D. Fox, "6-DOF GraspNet: Variational grasp generation for object manipulation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2901–2910.

[6] H. Zhang, X. Zhou, X. Lan, J. Li, Z. Tian, and N. Zheng, "A real-time robotic grasping approach with oriented anchor box," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 5, pp. 3014–3025, May 2021.

[7] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 292–301.

[8] H. Zhang et al., "A practical robotic grasping method by using 6-D pose estimation with protective correction," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3876–3886, Apr. 2022.

[9] S. Hinterstoisser et al., "Model based training, detection and pose estimation of texture-less 3-D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 548–562.

[10] S. Hinterstoisser et al., "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 858–865.

[11] G. Zhou, H. Wang, J. Chen, and D. Huang, "PR-GCN: A deep graph convolutional network with point refinement for 6D pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2793–2802.

[12] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16611–16621.

[13] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time rgb-based 6-DOF object pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7678–7687.

[14] C. Wang et al., "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3343–3352.

[15] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1521–1529.

[16] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6D object pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2930–2939.

[17] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3-D keypoints voting network for 6DoF pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11632–11641.

[18] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot. Sci. Syst.*, 2018, pp. 1–9.

[19] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate o (n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.

[20] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DOF pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4561–4570.

[21] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3003–3012.

[22] T. Cao, F. Luo, Y. Fu, W. Zhang, S. Zheng, and C. Xiao, "DGECN: A depth-guided edge convolutional network for end-to-end 6D pose estimation," 2022, *arXiv:2204.09983*.

[23] C. Li, J. Bai, and G. D. Hager, "A unified framework for multi-view multi-class object pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 254–269.

[24] W. Liu et al., "SSD: Single shot MultiBox detector," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.

[26] S. Lin, Z. Wang, Y. Ling, Y. Tao, and C. Yang, "E2EK: End-to-end regression network based on Keypoint for 6D pose estimation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6526–6533, Jul. 2022.

[27] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3828–3836.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

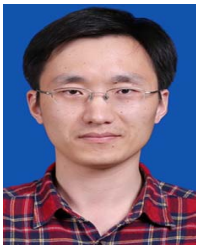
[29] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.

[30] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1941–1950.

[31] T. Hodan, D. Barath, and J. Matas, "EPOS: Estimating 6D pose of objects with symmetries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11703–11712.
[32] W. Hua, Z. Zhou, J. Wu, H. Huang, Y. Wang, and R. Xiong, "REDE: End-to-end object 6D pose robust estimation using differentiable outliers elimination," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2886–2893, Apr. 2021.
[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
[34] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.
[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[36] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[37] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 536–551.
[38] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.
[39] X. Yu, Z. Zhuang, P. Koniusz, and H. Li, "6DoF object pose estimation via differentiable proxy voting regularizer," in *Proc. Brit. Mach. Vis. Conf.*, 2020, pp. 1–13.
[40] M. Tian, L. Pan, M. H. Ang, and G. H. Lee, "Robust 6D object pose estimation by learning RGB-D features," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 6218–6224.
[41] C. Song, J. Song, and Q. Huang, "HybridPose: 6D object pose estimation under hybrid representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 431–440.

Sheng Yu received the B.Eng. degree in automation from the Hebei University of Technology, Tianjin, China, in 2018, and the M.Eng. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2021. He is currently pursuing the Doctoral degree in control science and engineering with the School of Automation, Beijing Institute of Technology.

His current research interests include robotic grasping, computer vision, and deep learning.



Di-Hua Zhai received the B.Eng. degree in automation from Anhui University, Hefei, China, in 2010, the M.Eng. degree in control science and engineering from the University of Science and Technology of China, Hefei, in 2013, and the Dr.Eng. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2017.

Since 2017, he has been with the School of Automation, Beijing Institute of Technology, where he is currently an Associate Professor. His research interests include intelligent robot, networked robots, computer vision in robotics, artificial intelligence in medicine, switched control, and optimal control.

Yuanqing Xia (Fellow, IEEE) received the M.S. degree in fundamental mathematics from Anhui University, Hefei, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001.

From January 2002 to November 2003, he was a Postdoctoral Research Associate with the Institute of Systems Science, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing. From November 2003 to February 2004, he was a Research Fellow with the National University of Singapore, where he worked on variable structure control. From February 2004 to February 2006, he was a Research Fellow with the University of Glamorgan, Pontypridd, U.K. From February 2007 to June 2008, he was a Guest Professor with Innsbruck Medical University, Innsbruck, Austria. Since 2004, he has been with the Department of Automatic Control, Beijing Institute of Technology, Beijing, first as an Associate Professor, then a Professor, since 2008. In October 2023, he was also hired as the President of the Zhongyuan University of Technology, Zhengzhou, China. He is currently a Professor with the Beijing Institute of Technology, Beijing, and the President of the Zhongyuan University of Technology. His research interests are in the fields of cloud control systems, networked control systems, robust control and signal processing, active disturbance rejection control, unmanned system control, and flight control.

Wei Wang received the Master of Science degree in system theory from Beijing Jiaotong University, Beijing, China, in 2017.

He is currently a Middle-Level Engineer with China Unicom Research Institute, Beijing. His current research interests include big data, cloud computing, blockchain, industrial Internet, computer vision, and deep learning.

Chengyu Zhang received the M.Sc. degree in communication engineering and networks from the University of Birmingham, Birmingham, U.K., in 2012.

He is currently a Senior Engineer with China Unicom Research Institute, Beijing, China. His current research interests include big data, cloud computing, artificial intelligence, and industrial Internet.

Shiqi Zhao received the B.Eng. degree in electronics engineering from Beihang University, Beijing, China, in 2014, and the M.Eng. degree in electronics engineering from the University of Chinese Academy of Sciences, Beijing, in 2017.

He is currently a Senior Researcher of Artificial Intelligence with China Unicom, Beijing. His current research interests include computer vision and multimodal learning.