Query6DoF: Learning Sparse Queries as Implicit Shape Prior for Category-Level 6DoF Pose Estimation

Ruiqi Wang[1], Xinggang Wang[1], Te Li[†2,3], Rong Yang[2,3], Minhong Wan[†2,3], Wenyu Liu[†1]

[1]Hubei Key Laboratory of Smart Internet Technology, School of Electronic Information and Communications, Huazhong University of Science and Technology
[2]Research Center for Intelligent Robotics, Research Institute of Interdisciplinary Innovation, Zhejiang Lab, China
[3]Zhejiang Engineering Research Center for Intelligent Robotics, China

Abstract

Category-level 6DoF object pose estimation intends to estimate the rotation, translation, and size of unseen objects. Many previous works use point clouds as a pre-learned shape prior to overcome intra-category variability. The shape prior is deformed to reconstruct instances' point clouds in canonical space and to build dense 3D-3D correspondences between the observed and reconstructed point clouds. However, the pre-learned shape prior is not jointly optimized with estimation networks, and they are trained with a surrogate objective. We propose a novel 6D pose estimation network, named Query6DoF, based on a series of category-specific sparse queries that represent the prior shape. Each query represents a shape component, and these queries are learnable embeddings that can be optimized together with the estimation network according to the point cloud reconstruction loss, the normalized object coordinate loss, and the 6d pose estimation loss. Query6DoF adopts a deformation-and-matching paradigm with attention, where the queries dynamically extract features from regions of interest using the attention mechanism and then directly regress results. Furthermore, Query6DoF reduces computation overhead through the sparseness of the queries and the incorporation of a lightweight global information injection block. With the aforementioned design, Query6DoF achieves state-of-the-art (SOTA) pose estimation performance on the NOCS datasets. The source code and models are available at https://github.com/hustvl/Query6DoF.

[†]Corresponding authors: Wenyu Liu (liuwy@hust.edu.cn), Te Li (lite@zhejianglab.com) and Minhong Wan (wanmh@zhejianglab.com).

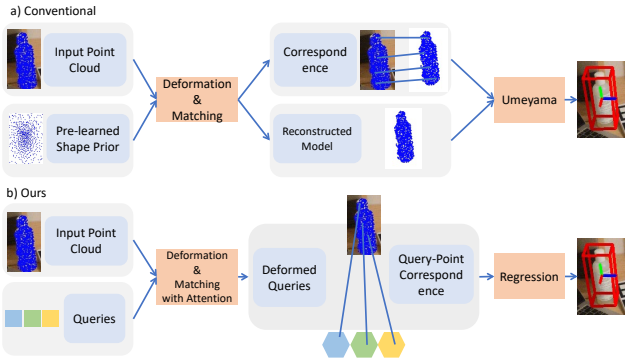

Figure 1: Comparisons with conventional works. a) Conventional methods adopt pre-learned shape prior and have a two-stage pipeline, *i.e.*, deforming the prior and building correspondences between points. b) Our method uses sparse learnable queries as the shape prior and conducts query/feature-based deformation and matching.

1. Introduction

Category-level object pose estimation is aimed at estimating an object's rotation, translation, and size from an RGB-D scene within a given set of categories. This task has attracted increasing attention due to its vital role in robotics [5], 3D understanding [4, 22], and augmented reality [29]. Compared with instance-level object pose estimation, category-level object pose estimation doesn't require obtaining the object's CAD model in advance, making it more generally applicable.

Many existing methods [2, 7, 16, 30, 41, 38, 20, 35] solve this problem as shown in Figure 1(a). First, they attempt to reconstruct the input object point cloud in the Normalized Object Coordinate Space [34] (NOCS). Then they estimate the 3D-3D correspondences between the input ob-

served point cloud and the reconstructed object point cloud. Finally, they determine object poses and sizes by applying the Umeyama algorithm [31] to these correspondences. To handle intra-category variability, a shape prior, which represents a basic shape for objects in the same category, is introduced in [30] to this task. This shape prior is a set of point clouds pre-learned in NOCS for each category. Objects are reconstructed by deforming the shape prior. However, this approach has its drawbacks: 1) the shape prior is a dense representation of objects in a specific category, which is not only redundant but also makes the network computationally expensive. 2) the shape prior is pre-learned in advance and remains static during model training. Additionally, these methods are trained to predict the deformation field D and a correspondence matrix A, which is a surrogate task. 3) Conventional model architectures [30, 3, 2] typically rely on either pooling operations in the PointNet [26] style or 3D graph convolution (3DGC) [19] to explore geometry features. However, the pooling operation can result in the loss of local geometry features, and these operations are not adaptive to extract the feature information.

To address these issues, we propose a novel method called Query6DoF (illustrated in Figure 1(b)). Instead of relying on dense and static point clouds as the shape prior, we introduce a series of sparse and learnable category-specific queries. Each query in this series is designed to encode a specific component of an object's shape with implicitly shared semantics and is optimized concurrently with the whole model parameters. Due to intra-category variability, the queries only need to capture the most representative components of objects and disregard unrepresentative shape details. Consequently, the queries can be sparse. Moreover, due to the sparseness of queries, the sparseness of the queries reduces computational overhead. Therefore, there is no need to use techniques such as Low-Rank transformer [36] mentioned in SGPA [2]. To eliminate the need for surrogate training objectives, we employ a direct prediction framework. However, unlike previous direct regression methods, *e.g.*, [3, 20], we choose to perform the core deformation-and-matching paradigm in feature space. The deformation process is accomplished based on an attention mechanism, which adaptively extracts information from instance features to queries. This transforms queries from category-specific into instance-specific shape representations. Then, correspondences between instance features and deformed queries are established by computing their similarity. Using these correspondences, the instance features are paired with deformed queries. Finally, these pairs are used to determine the pose and size through a neural network. Since the shape prior already consists of implicit queries, it is reasonable to also execute the entire pipeline implicitly in the feature space to better leverage these queries. Without pre-learned prior and surrogate training objectives, our model is trained entirely and directly. As a result, the prior can be more suitable for the model, enabling us to achieve state-of-the-art performance on the task.

Conventionally, the self-attention mechanism is used in transformers to inject global information. Unfortunately, adding self-attention layers significantly increases computational load. Therefore, we propose an efficient global enhancement layer to balance efficiency and performance.

The main contributions of this work can be summarized as:

- We propose to use sparse and learnable queries as shape prior rather than dense and static point clouds. In this way, the queries are optimized at the same time as the whole model to discover the optimal set of priors, and the sparseness design reduces the computation overhead.
- We adapt existing architectures to fit the novel shape prior representation with the help of attention mechanisms. We design a lightweight self-attention module in this to balance accuracy and speed.
- The overall 6DoF pose estimation accuracy of our method is better than previous state-of-the-art methods on the CAMERA25 and REAL275 datasets, especially, on the strict $5°2cm$ metric our results are significantly better than previous methods.

2. Related Work

2.1. Instance-Level 6D Object Pose Estimation

Based on the input data format, existing methods can be divided into two categories: RGB-based [15, 28, 25, 22, 37, 23] and RGB-D-based [14, 32, 10, 13] approaches. A common approach for using RGB as input to accomplish this task is to detect the object's keypoints in the image and match them to the CAD model. With the 2D-3D correspondences, the PnP algorithm is then implemented to obtain the object's pose. Additionally, some methods [12, 33, 6, 1, 39, 11] do not use PnP algorithm but instead employ neural networks to directly output object's pose.

2.2. Category-level 6D Object Pose and Size Estimation

[34] first introduced Normalized Object Coordinate Space to represent all the possible object instances in a unified space. DualPoseNet [18] introduced a network stacking an implicit decoder to impose complementary supervision on pose encoder. We absorb this practice and provide extra implicit supervision. To handle the intra-class shape variation, SPD [30] proposes a deep network to reconstruct the 3D object model by explicitly modeling the deformation from a pre-learning categorical shape prior. However,

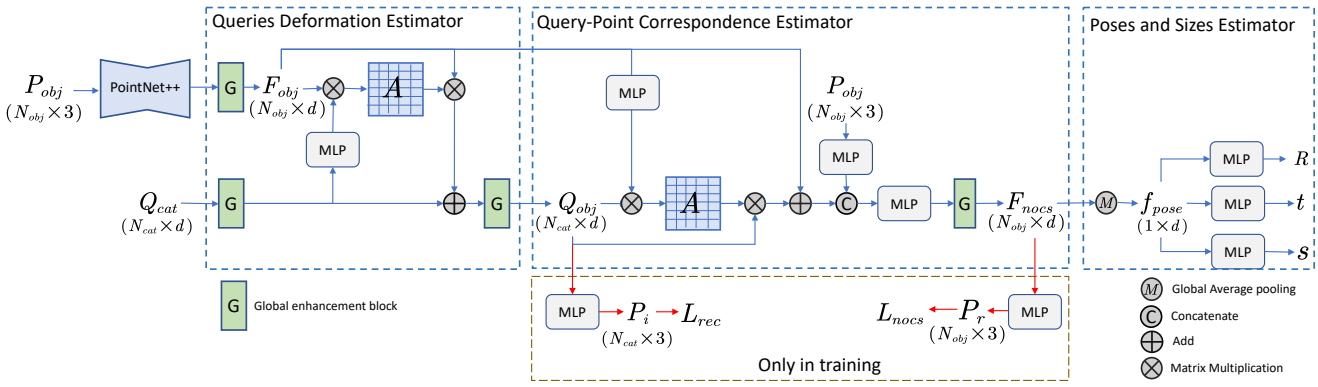

Figure 2: Entire schematic illustration of our proposed method. The target point cloud P_{obj} is fed to PointNet++ [27] as a backbone to extract the object's geometry features F_{obj}. A set of queries Q_{cat} is prepared to serve as the shape prior. Each of the queries represents a component of object shape. The queries pass through the queries deformation estimator to gain instance shape information from F_{obj} using an attention mechanism to adaptively select instance features, and then they are deformed to the specific representation of the target observed object Q_{obj}. Next, in the query-point correspondence estimator, the correspondence between Q_{obj} and F_{obj} is built. In the end, the poses and sizes estimator makes use of the previous output. This block simply adopts global average pooling to aggregate all the input features and output resulting poses and sizes through three parallel MLPs.

this prior is not flexible enough because it is obtained in advance and remains unchanged during training. Therefore, we propose to transform it into the learnable queries. DPDN [17] makes efforts to overcoming synthetic-to-real domain gap using self-supervised learning. We further develop its paradigm of deforming and matching in feature space with the attention mechanism to make it much more effective. SGPA [2] further leverages the advantage of category prior. It uses the transformer architecture and in order to reduce computational complexity SGPA utilized the low-rank transformer [36]. But in our work, our proposed queries do not require to this technique for its sparseness.

3. Method

3.1. Overview

Given an RGB-D image, the goal of the category-level object pose and size estimation is to predict the 6D object pose and size represented by rotation $R \in SO(3)$, translation $t \in \mathbb{R}^3$, and size $s \in \mathbb{R}^3$. We segment objects from the RGB image using an off-the-shelf object segmentation technique *e.g.*, Mask R-CNN [9]. The segmented object's depth data is transferred to a 3D point cloud. Then the 3D point cloud is uniformly sampled into a fixed number. The same as [3], noting that objects' appearance varies much more than their shape even in the same category, our method only uses point cloud data as input instead of together with RGB data.

Let's denote the observed target point cloud as $P_{obj} \in \mathbb{R}^{N_{obj} \times 3}$. Our method uses a feature extraction module to extract object geometry feature $F_{obj} \in \mathbb{R}^{N_{obj} \times d}$ from P_{obj}. Unlike conventional methods that use an extra set of point clouds as explicit shape priors, we prepare a set of queries $Q_{cat} \in \mathbb{R}^{N_{cat} \times d}$ for each category as implicit shape prior. Each query expresses a semantic component of the object and they are the general category-specific representation. The number of queries N_{cat} can be significantly fewer than the number of points in the input point cloud N_{obj}. Besides, the queries are learnable, which means that there is no pre-defined prior and the full model is trained entirely and directly.

A traditional approach to obtaining the final poses and sizes with shape prior can be summarized as follows. First, deform the prior point cloud to reconstruct the object shape. Then, the correspondence between the reconstructed point cloud and the observed object point cloud is estimated. Afterward, poses and sizes can be solved via Umeyama [31].

The entire schematic illustration of our proposed method is shown in Figure 2. We implement this process in feature space with an attention mechanism. Specifically, our method consists of three parts: queries deformation estimator, query-point correspondence estimator, and poses and sizes estimator. Furthermore, we propose a global enhancement block to reinforce the features with global information.

Queries deformation estimator is used to deform the queries Q_{cat} from the general category-specific representation to an object-specific one for the observed input object using the input instance geometry feature F_{obj}, which can

be formulated as:

$$Q_{obj} = QDE(Q_{cat}, F_{obj}) \tag{1}$$

where $Q_{obj} \in \mathbb{R}^{N_{cat} \times d}$ denotes the deformed queries. QDE denotes the queries deformation estimator. In this process, we adopt the attention mechanism to extract the information of regions of interest from F_{obj} adaptively. After deforming the queries, the canonical point cloud can be decoded from Q_{obj} denoted as:

$$P_i = \Phi_1(Q_{obj}) \tag{2}$$

where $P_i \in \mathbb{R}^{N_{cat} \times 3}$ denotes the object's canonical sparse point cloud. Φ_1 denotes the decoder implemented by an MLP (multi-layer perceptron). Supervision is applied to P_i to help better guide the learning of Q_{obj} via the loss function L_{rec}. Query-point correspondence estimator is used to estimate the correspondences between deformed queries Q_{obj} and object feature F_{obj}. The correspondence then is used to combine Q_{obj} with F_{obj} as matching feature pairs. These pairs are fed forward an MLP and a global enhancement block. The result is denoted as F_{nocs}. The corresponding NOCS coordinates of the observed object point cloud are obtained from F_{nocs} using an MLP.

$$P_r = \Phi_2(F_{nocs}) \tag{3}$$

where $P_r \in \mathbb{R}^{N_{obj} \times 3}$ denotes the corresponding NOCS coordinates. Φ_2 is an MLP. P_r does not take part in the calculation of objects' poses and sizes but only participates in the calculation of the loss function L_{nocs}. This loss function is an auxiliary loss function, which is used to apply supervision on P_r in order to guide the learning of F_{nocs}.

The pose and size estimator takes the previous output and gives the object's pose and size with three parallel MLPs.

3.2. Queries Deformation Estimator

In the queries deformation estimator, we aim to deform the queries Q_{cat} to a representation of the observed target object. This goal can be achieved by injecting the object geometry feature F_{obj} into Q_{cat}. Due to the unordered property of the point cloud, the PointNet [26] style operation, which is mainly composed of pooling, is adopted by many methods such as [30, 38]. However, this operation loses the detailed geometry structure and cannot adaptively pay attention to features of interest. Therefore, we use the cross-attention mechanism. Thanks to the permutation-invariance of the attention mechanism, it is feasible to implement it for point cloud processing. Moreover, we enhance the feature using the proposed global enhancement block.

The detailed structure of the queries deformation estimator is illustrated in Figure 2. **Cross-attention** is computed to determine which feature in F_{obj} is necessary to extract into Q_{cat}. Specifically, we take the Q_{cat} as the $query$ and

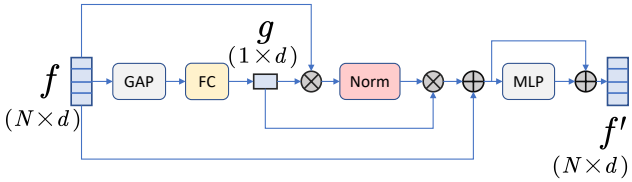

Figure 3: Overview of the global enhancement block. The input feature f pass through a global average pooling operation and a fully-connected layer to create a global feature g. Then, the input feature f serves as the $query$, and the global feature g serves as both key and $value$. A self-attention mechanism is then applied to them.

the F_{obj} as key and $value$ of the multi-head cross-attention module. The computation of attention map can be formulated as:

$$A^{(m)} = Norm(\Psi_1(Q_{cat})(F_{obj})^T) \tag{4}$$

where $A^{(m)} \in \mathbb{R}^{N_{cat} \times N_{obj}}$ is the attention map in the m-th head. $Norm$ is the normalization function. Instead of using learnable linear projections, we use Ψ_1, an MLP, applied to Q_{cat} in order to reduce the gap between Q_{cat} and F_{obj} and it experimentally works better. Vanilla attention uses softmax to normalize the attention map, which can be formulated as $norm = \frac{f(x_i)}{\Sigma_k f(x_k)}$ where $f(x) = e^x$ for softmax. Different from that, we use:

$$f(x) = \frac{\alpha}{1 + e^{-x}} \tag{5}$$

where α is a learnable parameter. This is a sigmoid function with a learnable parameter. This function's output is limited between 0 and α, so there is no need to scale the dot-product result between $query$ and key by dividing by $\sqrt{d}$. This function tends to saturate and thus activate larger regions than softmax does, and the learnable parameter allows the model to choose the range. Since each query corresponds to a certain semantic part of the object in a specific category, the attention map indicates which component of the observed object has the same semantics as the query by calculating the similarity between the queries Q_{cat} and the object feature F_{obj}.

Then, the multi-head cross-attention can be formulated as:

$$D = [A^{(1)}F_{obj}; ...; A^{(m)}F_{obj}]W_v \tag{6}$$

where $D \in \mathbb{R}^{N_{cat} \times d}$ is the multi-head cross-attention result and W_v is the learnable parameter. In this way, the related object feature is extracted from the F_{obj}. Then D can be used to deform Q_{cat} by adding it to Q_{cat}:

$$Q_{obj} = Q_{cat} + D \tag{7}$$

where $Q_{obj} \in \mathbb{R}^{N_{cat} \times d}$ is deformed queries.

Q_{obj} is expected to embed the object's canonical point cloud. It can be decoded into 3D points using a simple MLP.

$$P_i = Sigmoid(MLP(Q_{obj})) - 0.5 \tag{8}$$

where $P_i \in \mathbb{R}^{N_{cat} \times 3}$. A $Sigmoid$ function is used to limit it to $(-0.5, 0.5)$, so the output points are in canonical space. We apply supervision to P_i which could guide the learning of Q_{obj}. Therefore, loss function L_{rec} is applied to P_i to provide supervision to it.

Global enhancement block tends to enhance features by fusing their global feature. Normally, this can be achieved via the self-attention mechanism. However, to make the entire model lightweight, a more efficient block is needed. For this reason, we propose a global enhancement block.

The global enhancement block is illustrated in Figure 3. Suppose the global enhancement block takes feature $f \in \mathbb{R}^{N \times d}$ as input. We adopt the result of the global average pooling as the global feature and pass it through a fully connected layer. The outcome is denoted as g:

$$g = FC(AvgPool(f)) \tag{9}$$

where FC is a fully connected layer. Instead of simply concatenating it to f and then feeding it to an MLP, we borrow the idea from self-attention as shown in Figure 3. We use f as $query$ and g as key and $value$ to carry out self-attention, which can be denoted as:

$$f' = f + Norm(fg^T)g \tag{10}$$

After that, f' is fed into an MLP with residual. In this way, global information is injected into the feature. This block, unlike vanilla self-attention, reduces the complexity of the computation. Compared with the computation complexity of vanilla self-attention, $O(N^2 \times d)$, our efficient global enhancement block has a computation complexity of $O(N \times d)$.

3.3. Query-Point Correspondence Estimator

We obtain the deformed queries Q_{obj} by adaptively selecting feature from the observed object feature F_{obj}. In previous works, a correspondence matrix $A \in \mathbb{R}^{N_{obj} \times N_{cat}}$ is estimated, which models correspondences between points in observed point cloud P_{obj} and reconstructed point cloud P_i. This matrix then samples the points from P_i to be paired with P_{obj}, and it can be formulated as $P = A \times P_i$, where P denotes the sampled points. Finally, solving of Umeyama algorithm [31] to align the sample points with P_i gives out the target pose and size. Instead of using this conventional way, we move this process to the feature space. In the query-point correspondence estimator, the correspondences between F_{obj} and Q_{obj} are built as illustrated in Figure 2.

In detail, the correspondences are built by computing the similarity between F_{obj} and Q_{obj} using dot-product, which can be fomulated as:

$$A' = Norm(\Psi_2(F_{obj})(Q_{obj})^T) \tag{11}$$

where $A' \in \mathbb{R}^{N_{obj} \times N_{cat}}$ denotes the correspondence matrix between F_{obj} and Q_{obj}. $Norm$ is the same as that used in queries deformation estimator. Ψ_2 is an MLP. A' represents the soft matching between F_{obj} and Q_{obj}. Then we multiply A' by Q_{obj} to obtain the sampled features. Essentially, we find that this process is similar to cross-attention. For this reason, we add F_{obj} to the result for implementing a residual structure similar to attention mechanism. Also, we implement multi-head technique to the computation. Next, we need to pair the object feature with the sampled feature. Experimentally, it is better to use a more direct feature as the object feature here instead of the deep object feature F_{obj}. Thereby, we do not use F_{obj} but simply feed object point cloud P_{obj} forward an MLP. The output of the MLP is concatenated to the sampled feature. To further enhance the feature, we use an MLP and a global enhancement block after that. The result is denoted as F_{nocs}. To apply supervision on F_{nocs} and better guide its learning, we use a simple MLP to transform F_{nocs} into NOCS coordinates denoted as $P_r \in \mathbb{R}^{N_{obj} \times 3}$. P_r is not used to solve the Umeyama algorithm to obtain target pose and size, but only used to compute loss function L_{nocs}.

3.4. Poses and Sizes Estimator

The rest of the work is to obtain the final object's pose and size, which is accomplished in the pose and size estimator. The pose and size estimator applies global average pooling operation to F_{nocs}, which can be formulated as:

$$f_{pose} = AvgPool(F_{nocs}) \tag{12}$$

where $f_{pose} \in \mathbb{R}^{1 \times d}$. Then after it, three parallel MLPs predict the rotation R, translation t, and size s respectively.

$$R, t, s = MLP(f_{pose}), MLP(f_{pose}), MLP(f_{pose}) \tag{13}$$

The representation of R here is the Continuity of Rotation Representations [40]. For s, we follow [3] to estimate the residual between object's size and the mean category size. Similarly, for t, we predict the residual between the translation ground truth and the mean value of the point cloud.

3.5. Overall Loss Function

The overall loss function is as follows:

$$L = \lambda_1 L_{pose} + \lambda_2 L_{rec} + \lambda_3 L_{nocs} \tag{14}$$

L_{pose} is used to compute loss on predicted R, t and s. We simply use smooth L1 loss [8] for all of them with threshold of 1×10^{-3}, 5×10^{-3}, 5×10^{-3} respectively. L_{rec} and

L_{nocs} are the auxiliary losses function for P_i and P_r respectively. L_{rec} is applied to P_i to encourage it to reconstruct the instance point cloud in the NOCS space. Therefore, for L_{rec}, we use Chamfer distance between P_i and M_{gt} where $M_{gt} \in \mathbb{R}^{N_{obj} \times 3}$ represents the instance point cloud model in the NOCS space. L_{rec} can be formulated as follows:

$$L_{rec} = \frac{1}{N_{cat}} \sum_{x_i \in P_r} \min_{y_j \in M_{gt}} \|x_i - y_j\|_2^2 + \frac{1}{N_{obj}} \sum_{y_i \in M_{gt}} \min_{x_i \in P_r} \|x_i - y_j\|_2^2 \tag{15}$$

L_{nocs} is applied on P_r to encourage it in predicting the corresponding coordinates of P_{obj} in Normalized Object Coordinate Space. For L_{nocs}, we also use a smooth L1 loss [8] with a threshold of 0.1.

4. Experiments

Datasets. We conduct experiments using the benchmark CAMERA25 and REAL275 [34] datasets for category-level 6D object pose and size estimation which consists of six object categories. The CAMERA25 dataset is generated by rendering and compositing synthetic object instances under different views. The CAMERA25 dataset contains 300K synthetic images of 1,085 object instances, among which 25,000 images of 184 instances are used for evaluation. The REAL275 dataset is complementary to the CAMERA25 dataset. It includes 4,300 real-world images of 6 scenes with 3 unseen instances per category; its training set contains 4,300 images of 7 scenes, and the test set contains 2.750 images of 6 scenes. We train our model with a combined use of the two datasets, as done in [18].

Implement Details. We employ a Mask R-CNN [9] to obtain instance masks the same as [30]. The instances are then cropped based on the segmentation result. After that, the depth is converted to the instance point cloud using the camera's intrinsic parameters. 1024 points are randomly sampled from that, so N_{obj} is set to 1024. To extract instance geometry features, we choose PointNet++ [27] with four abstraction levels. The number of prior queries N_{cat} is set to 64. In this case, $N_{cat} \ll N_{obj}$. A detailed study on the parameter N_{cat} is given in our ablation study. The queries are initialized randomly at the beginning of training. The dimension of features and queries is set to 256. The cross-attention in queries deformation estimator and query-point correspondence estimator has four heads each. In the loss function, we set λ_1, λ_2, λ_3 to 1.0, 3.0, 1.0, respectively. We use AdamW [21] to train this model with an initial learning rate of 1×10^{-4} and weight decay 1×10^{-4}. All models are trained 42k iterations with batch size of 60 and cosine learning rate decay is conducted. We employ several strategies for depth augmentation, such as adding Gaussian noise, random scaling, random rotation, adding random points as well as linear shape augmentation proposed in [3] and non-linear shape augmentation proposed in [38]. All FPS(Frames Per Second) data of our method is evaluated on a single RTX 2080Ti, an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, PyTorch [24] 1.10, CUDA version 11.1.

Evaluation Metrics. For category-level pose estimation, we follow [34] and use two metrics to evaluate the quality.

- 3D IoU. The overlap of two 3D bounding boxes under the predicted pose and the ground truth pose is computed. The overlap ratios larger than the threshold are regarded as accepted. The mean Average Precision (mAP) of the overlapping is reported.
- $n^\circ m$ cm. The rotation and translation errors between the predicted pose and the ground truth pose are computed. The errors smaller than an angle threshold n° and a translation threshold m cm are accepted.

4.1. Comparisons with Existing Methods

NOCS-REAL275 In Table 1 we compare our method with the existing ones for category-level 6D object pose and size estimation. We divide these methods into two groups: those with and without shape prior. As it is shown in Table 1, our method outperforms other existing methods. In detail, we outperform NOCS [34] by 46.0 in IoU_{75}, 41.8 in $5^\circ 2cm$, 48.9 in $5^\circ 5cm$. Moreover, we achieve a better result than existing methods that using prior information. Specifically, we outperform SPD [30] by 22.9 in IoU_{75}, 29.7 in $5^\circ 2cm$, 37.5 in $5^\circ 5cm$. and 14.2, 13.1, and 19.3 higher than SGPA [2] which is a representative method based on attention. This outcome demonstrates that our sparse shape prior is superior to the traditional shape prior.

NOCS-CAMERA25 The results for CAMERA25 are shown Table 1. Our method outperforms others at least 4.9, 4.3 in the items of $5^\circ 2cm$, $5^\circ 5cm$, and it is on par with the best methods for IoU_{75}. Figure 4 presents a more detailed error evaluation result on two datasets.

4.2. Ablation Studies

Effect of the design in Query6DoF. We verify the efficacy of each component in Query6DoF in Table 2. By replacing queries with point cloud proposed in [30] and adding an extra PointNet++ [27] after the prior point cloud as a feature extractor, $5^\circ 2cm$ drops from 49.0 to 47.2, and computation increases. Removing the global enhancement block results in a 4.8 drop in $5^\circ 2cm$. Removing the queries deformation estimator results in a 2.5 drop in $5^\circ 2cm$. If the query-point correspondence estimator is detached, it renders the usage of queries and queries deformation estimator senseless. Therefore, without using them together, our method experiences a 13.6 drop in $5^\circ 2cm$. Moreover, we remove the poses and sizes estimator but use P_r and P_i

Table 1: Comparison with state-of-the-art methods on the CAMERA25 dataset and REAL275 dataset. * denotes results without linear shape augmentation and non-linear shape augmentation. The best results shown in bold.

Method	prior	CAMERA25						REAL275					
		IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$
NOCS [34]		83.9	69.5	32.3	40.9	48.2	64.6	78.0	30.1	7.2	10.0	13.8	25.2
DualPoseNet [18]		92.4	86.4	64.7	70.7	77.2	84.7	79.8	62.2	29.3	35.9	50.0	66.8
GPV-Pose [7]		93.4	88.3	72.1	79.1	-	89.0	83.0	64.4	32.0	42.9	-	73.3
SPD [30]	✓	93.2	83.1	54.3	59.0	73.3	81.5	77.3	53.2	19.3	21.4	43.2	54.1
CR-Net [35]	✓	**93.8**	88.0	72.0	76.4	81.0	87.7	79.3	55.9	27.8	34.3	47.2	60.8
SAR-Net [16]	✓	86.8	79.0	66.7	70.9	75.6	80.3	79.3	62.4	31.6	42.3	50.3	68.3
SGPA [2]	✓	93.2	88.1	70.7	74.5	82.7	88.4	80.1	61.9	35.9	39.6	61.3	70.7
RBP-Pose [38]	✓	93.1	**89.0**	73.5	79.6	82.1	89.5	-	67.8	38.2	48.1	63.1	79.2
Self-DPDN [17]	✓	-	-	-	-	-	-	**83.6**	76.0	46.0	50.7	**70.4**	78.4
Ours*	✓	92.3	88.6	**78.4**	**83.9**	**84.0**	**90.5**	82.9	76.0	46.8	54.7	67.9	81.6
Ours	✓	91.9	88.1	78.0	83.1	83.9	90.0	82.5	**76.1**	**49.0**	**58.9**	68.7	**83.0**

Table 2: Effect of the design on Query6DoF, evaluated on the REAL275 dataset. GlobEnhance denotes the global enhancement block, QueryDeform denotes the queries deformation estimator, Query2Point denotes the query-point correspondence estimator, and PoseSize denotes the poses and sizes estimator

	Queries	GlobEnhance	QueryDeform	Query2Point	PoseSize	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$
1	-	✓	✓	✓	✓	82.1	75.4	47.2	56.2	67.3	81.2
2	✓	-	✓	✓	✓	82.6	75.7	44.2	53.3	65.8	81.1
3	✓	✓	-	✓	✓	**83.0**	75.3	46.5	56.6	65.9	81.3
4	-	✓	-	-	✓	82.1	70.9	35.4	48.7	59.5	80.2
5	✓	✓	✓	✓	-	82.2	70.0	40.0	46.3	65.1	76.5
6	✓	✓	✓	✓	✓	82.5	**76.1**	**49.0**	**58.9**	**68.7**	**83.0**

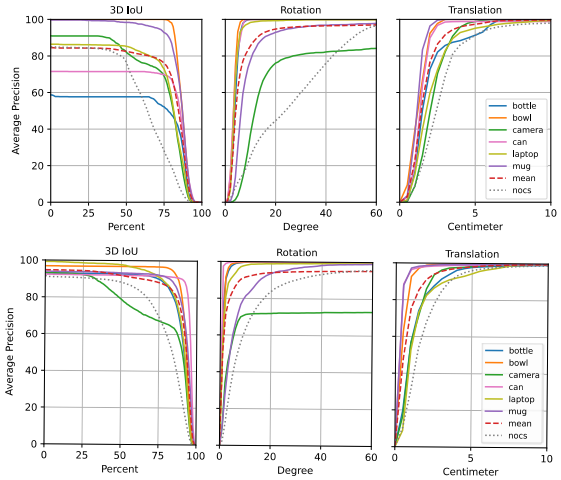

Figure 4: Mean Average Precision (mAP) in different thresholds. REAL275 (top rows) and CAMERA25 (bottom rows)

to obtain poses and sizes via Umeyama [31]. As a result, $5°2cm$ falls by 9.0 points.

Effect of the number of prior queries. In our method, we use sparse queries to represent the shape prior. In this experiment, we investigate the effect of the number of prior queries N_{cat} on the results of REAL275. Table 3 presents the result of the ablation of the effect of the number of prior queries. The results shows that reducing the number of prior queries does not significantly affect the performance of our method. This fact means that it is not necessary to use excessive points to express the object's shape prior. A small number of queries can effectively represent the object's shape. Specifically, when $N_{cat} = 64$, our method achieves the best performance in the $5°2cm$, $5°5cm$, $10°5cm$ metrics and when $N_{cat} = 1024$, our method achieves the best performance in the IoU_{50} metric. However, the smaller N_{cat} results in less computational overhead. Therefore, we adopt $N_{cat} = 64$ as the default setting in all other experiments.

Table 3: Effect of the number of queries on REAL275

N_{cat}	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$
16	82.5	74.6	46.0	55.4	66.6	81.0
32	82.5	74.9	46.7	55.9	67.2	81.5
64	82.6	76.1	**49.0**	**58.9**	68.7	**83.0**
128	82.8	**76.9**	46.1	55.9	68.8	82.8
256	82.8	76.6	47.6	57.3	67.8	81.6
512	82.6	75.6	46.8	56.2	66.9	81.4
1024	**83.1**	76.7	48.7	56.8	**70.4**	82.4

Table 4: Effect of different attention normalization evaluated on REAL275.

Norm	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$
Softmax	82.1	74.7	48.6	57.2	68.6	82.4
Ours	**82.5**	**76.1**	**49.0**	**58.9**	**68.7**	**83.0**

Effect of different attention normalization. While

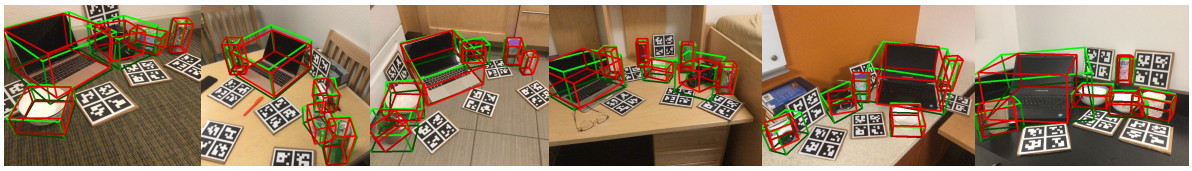

Figure 5: Qualitative results of our method (red line) and SGPA [2] (green line).

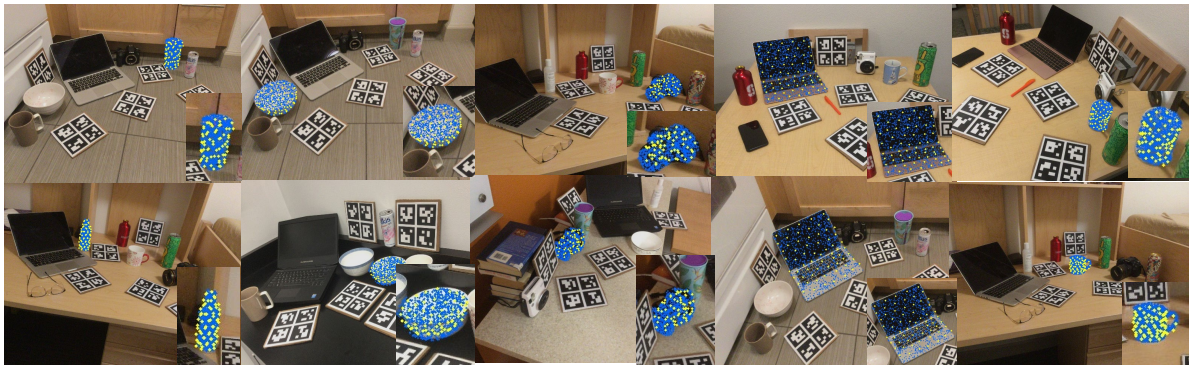

Figure 6: Sparse point cloud reconstruction in the REAL275 dataset. The yellow points represent the reconstruction points P_i, while the blue points are the observed point cloud.

Table 5: Model complexity and FPS w/o Mask R-CNN [9], evaluated on REAL275. GEB denotes the global enhancement block.

Method	Params(M)	FPS w/o Mask R-CNN [9]	$5°2cm$
$N_{cat} = 64$ (Default)	19.53	34.9	**49.0**
$N_{cat} = 1024$	21.18	23.3	48.7
w/o GEB	18.71	36.8	44.2
w/o Queries	22.24	24.0	47.2

common attention mechanisms adopt $softmax$ to normalize the attention map, in our method, we use a different normalization approach, namely $sigmoid$ with a learnable parameter mentioned earlier. In this experiment, we compare two different settings shown in Table 4. We conduct the evaluation on the REAL275 dataset. Using softmax to normalize results in a 1.4 drop in IoU_{75} and a 0.4 drop in $5°2cm$.

4.3. Runtime Analysis

As shown in Table 5, our method, in the default setting, has 19.43M parameters and runs at 34.9 FPS without Mask R-CNN [9]. Increasing N_{cat} to 1024 brings no gain but the FPS drops to 23.3, which shows that sparse queries are sufficient to represent the shape prior. Adding the global enhancement block results in a 4.8 improvement in $5°2cm$ with only a 1.9 FPS drop, demonstrating its efficiency. W/o Queries means that we replace the queries with the pre-learned shape prior proposed in [30] and use PointNet++ [27] as a feature extractor. In this way, due to the extra PointNet++ [27], the speed drops to 24.0FPS, but there is no gain in $5°2cm$. Therefore, the usage of queries is more efficient than using point cloud both in terms of accuracy and speed.

4.4. Visualization

Qualitative Results. Figure 5 shows the qualitative results of our method, demonstrating that our method can accurately estimate the objects' poses and sizes.

Sparse point cloud reconstruction. Our method outputs P_i as the sparse point cloud reconstruction in NOCS space. We align P_i to the object's location in RGB images for better comparison with the object's actual shape, as shown in Figure 6. P_i is the yellow points in it. The blue points in it represent the observed point cloud for comparison. It shows that our sparse point cloud can accurately cover different objects' shapes. This effect can be attributed to our queries deformation estimator, where category-specific queries transformed into instance-specific ones. Through this process, the queries are able to reconstruct the objects.

5. Conclusion

In conclusion, we present a novel network for category-level 6D object pose and size estimation, namely Query6DoF. In particular, we propose to use a series of sparse and learnable queries serving as a shape prior, which is a set of tokens that implicitly represent objects in a cer-

tain category. Furthermore, we transform the traditional deformation-and-matching paradigm to the feature space using attention. Specifically, the queries are deformed by a process where we dynamically use the attention mechanism to select object features and incorporate them into queries. This step transforms the queries into a representation of a certain target object. Next, we establish the correspondences between deformed queries and object features. Finally, the poses and sizes are regressed using the correspondences. Thanks to learnable queries and direct output results, the queries are trained directly alongside the entire network. This method achieves state-of-the-art performance compared to other existing methods and holds promise for applications in robotics and augmented reality.

Acknowledgement This work was supported by the Key Research Project of Zhejiang Lab (No. G2021NB0AL03) and Modern Agriculture Major-core Technology Innovation Project of Jiangsu Province (No. cx(22)2031).

References

[1] Tuo Cao, Fei Luo, Yanping Fu, Wenxiao Zhang, Shengjie Zheng, and Chunxia Xiao. Dgecn: A depth-guided edge convolutional network for end-to-end 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3783–3792, 2022.

[2] Kai Chen and Qi Dou. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2773–2782, 2021.

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

[4] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8648–8657, 2019.

[5] Xinke Deng, Yu Xiang, Arsalan Mousavian, Clemens Eppner, Timothy Bretl, and Dieter Fox. Self-supervised 6d object pose estimation for robot manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3665–3671. IEEE, 2020.

[6] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021.

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

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3003–3013, 2021.

[11] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020.

[12] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020.

[13] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 205–220. Springer, 2016.

[14] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the european conference on computer vision (eccv)*, pages 254–269, 2018.

[15] Zhigang Li, Gu Wang, and Xiangyang Ji. 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*, pages 7678–7687, 2019.

[16] Haitao Lin, Zichang Liu, Chilam Cheang, Yanwei Fu, Guodong Guo, and Xiangyang Xue. 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*, pages 6707–6717, 2022.

[17] Jiehong Lin, Zewei Wei, Changxing Ding, and Kui Jia. Category-level 6d object pose and size estimation using self-supervised deep prior deformation networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 19–34. Springer, 2022.

[18] Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia, and Yuanqing Li. 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*, pages 3560–3569, 2021.

[19] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d

graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1800–1809, 2020.

[20] Xingyu Liu, Gu Wang, Yi Li, and Xiangyang Ji. Catre: Iterative point clouds alignment for category-level object pose refinement. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 499–516. Springer, 2022.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[22] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020.

[23] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[25] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[28] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020.

[29] Yongzhi Su, Jason Rambach, Nareg Minaskan, Paul Lesur, Alain Pagani, and Didier Stricker. Deep multi-state object pose estimation for augmented reality assembly. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 222–227. IEEE, 2019.

[30] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 530–546. Springer, 2020.

[31] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.

[32] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[33] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.

[34] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. 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*, pages 2642–2651, 2019.

[35] Jiaze Wang, Kai Chen, and Qi Dou. Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4807–4814. IEEE, 2021.

[36] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[37] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: Dense 6d pose object detector in rgb images. *arXiv preprint arXiv:1902.11020*, 1(2), 2019.

[38] Ruida Zhang, Yan Di, Zhiqiang Lou, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Rbp-pose: Residual bounding box projection for category-level pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 655–672. Springer, 2022.

[39] Guangyuan Zhou, Huiqun Wang, Jiaxin Chen, and Di Huang. Pr-gcn: A deep graph convolutional network with point refinement for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2793–2802, 2021.

[40] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[41] Lu Zou, Zhangjin Huang, Naijie Gu, and Guoping Wang. 6d-vit: Category-level 6d object pose estimation via transformer-based instance representation learning. *IEEE Transactions on Image Processing*, 31:6907–6921, 2022.