

# Learning Positive-Incentive Point Sampling in Neural Implicit Fields for Object Pose Estimation

Yifei Shi, *Member, IEEE*, Boyan Wan, Xin Xu, *Senior Member, IEEE*, Min Liu,  
Kai Xu, *Senior Member, IEEE*

**Abstract**—Learning neural implicit fields of 3D shapes is a rapidly emerging field that enables shape representation at arbitrary resolutions. Due to the flexibility, neural implicit fields have succeeded in many research areas, including shape reconstruction, novel view image synthesis, and more recently, object pose estimation. Neural implicit fields enable learning dense correspondences between the camera space and the object’s canonical space – including unobserved regions in camera space – significantly boosting object pose estimation performance in challenging scenarios like highly occluded objects and novel shapes. Despite progress, predicting canonical coordinates for unobserved camera-space regions remains challenging due to the lack of direct observational signals. This necessitates heavy reliance on the model’s generalization ability, resulting in high uncertainty. Consequently, densely sampling points across the entire camera space may yield inaccurate estimations that hinder the learning process and compromise performance. To alleviate this problem, we propose a method combining an SO(3)-equivariant convolutional implicit network and a positive-incentive point sampling (PIPS) strategy. The SO(3)-equivariant convolutional implicit network estimates point-level attributes with SO(3)-equivariance at arbitrary query locations, demonstrating superior performance compared to most existing baselines. The PIPS strategy dynamically determines sampling locations based on the input, thereby boosting the network’s accuracy and training efficiency. The PIPS strategy is implemented with a PIPS estimation network which generates sparse sample points with distinctive features capable of determining all object pose DoFs with high certainty. To collect the training data of the PIPS estimation network, we propose to automatically generate the pseudo ground-truth with a teacher model. Our method outperforms the state-of-the-art on three pose estimation datasets. It achieves 0.63 in the  $5^\circ 2\text{cm}$  metric on NOCS-REAL275, 0.62 in the  $5^\circ 5\text{cm}$  metric on ShapeNet-C, and 77.3 in the AR metric on LineMOD-O. Notably, it demonstrates significant improvements in challenging scenarios, such as objects captured with unseen pose, high occlusion, novel geometry, and severe noise.

**Index Terms**—Neural Implicit Fields, Point Sampling Strategy, Object Pose Estimation.

## 1 INTRODUCTION

RECENT work has made significant progress in learning neural implicit fields of 3D shapes, enabling compact and expressive 3D representation for shape reconstruction [10], [44], [48] and image synthesis [27], [46]. By training to predict the SDFs on a set of sampled points, neural implicit fields are able to generate the continuous SDFs on untrained locations with good generality, thanks to the advances of coordinate-based networks and the strategy of training on densely sampled points which cover the whole space of interest.

Sharing a similar idea, neural implicit fields have emerged as a powerful approach for 6D object pose estimation. Specifically, these methods learn dense correspondences between the camera space and the object’s canonical space. This capability extends beyond the input points to infer correspondences even for unobserved regions in the camera space [1], [18], [52]. Consequently, compared to conventional methods that directly predict poses, neural implicit fields achieve significantly higher accuracy and robustness, particularly in challenging scenarios involving heavy occlusion or novel object shapes.

Despite the advantages, the whole space dense sampling is a non-optimal strategy for pose estimation. There are two main

reasons. First, the whole space dense sampling would incur hard training samples for which the network is difficult to learn, such as the sampling points from unobserved camera-space regions, due to their indistinctive features. This necessitates heavy reliance on the model’s generalization ability, resulting in high uncertainty.

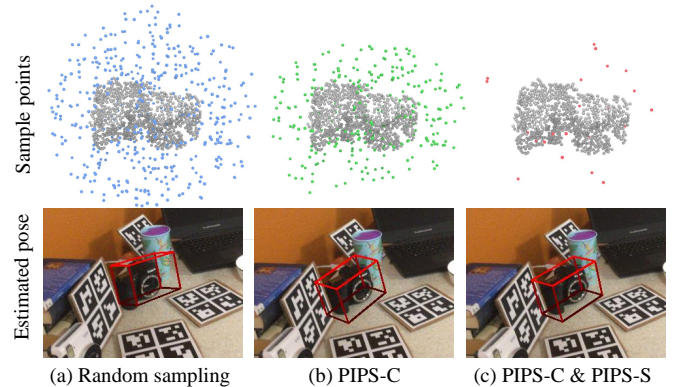


Fig. 1: We propose PIPS, a data-driven approach to dynamically determine where to sample to boost the network training, achieving better performance with training on fewer sampling points, compared to (a) the random sampling baseline. PIPS consists of two components: (b) positive-incentive point sampling with high estimation certainty (PIPS-C) and (c) positive-incentive point sampling with high geometric stability (PIPS-S).

- Yifei Shi and Xin Xu are with the College of Intelligence Science and Technology, National University of Defense Technology, China.
- Boyan Wan and Kai Xu are with the College of Computer Science, National University of Defense Technology, China.
- Min Liu is with the Academy of Military Science, China.
- Yifei Shi and Boyan Wan are joint first authors. Corresponding author: Kai Xu (kevin.kai.xu@gmail.com).

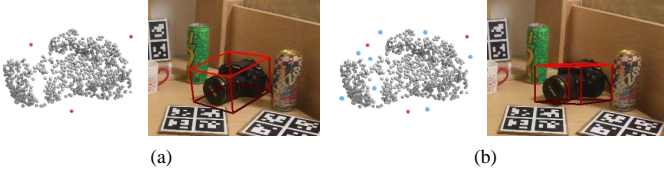


Fig. 2: (a) Point-wise canonical coordinate estimation on as few as three points (in red) is sufficient for determining all the 6-DoFs of object pose. (b) Extra voters with inaccurate point-level estimations (in blue) would degrade the performance.

As a result, employing data mining to select informative, distinguishing, and learnable training samples is necessary. Second, unlike shape reconstruction which requires all the point-wise estimations to be as accurate as possible, pose estimation only requires accurate point-wise estimation on a limited number of locations. As shown in Figure 2, canonical coordinate estimation on as few as three points is sufficient for determining all the DoFs of object pose. Extra voters with inaccurate point-level estimations might degrade the overall performance.

To tackle this problem, in this paper, we study the problem of how to generate sample points that would boost network training. We first provide the definition and the empirical analysis of the positive-incentive point sampling (PIPS) strategy. Specifically, we define PIPS as generating sparse sample points with distinctive features capable of determining all object pose DoFs with high certainty. To implement this idea and achieve robust performance, we propose to learn a neural implicit field using  $SO(3)$ -equivariant convolutions, trained with sample points generated by a PIPS estimation network.

The  $SO(3)$ -equivariant convolutional implicit network is a backbone that aggregates  $SO(3)$ -equivariant features from the input points and estimates point-level attributes at any query locations. To achieve this, the direction-independent point convolution kernels based on the vector neurons [11] are developed, making the operation of 3D convolution  $SO(3)$ -equivariant. By integrating the  $SO(3)$ -equivariant convolutions with a recent implicit neural network [65], the method outperforms most existing works of object pose estimation with implicit functions.

To generate sample points to train the above network, we propose a PIPS estimation network, a simple yet effective data-driven approach to dynamically determine where to sample to boost network training. The PIPS estimation network consists of two components: positive-incentive point sampling with high estimation certainty (PIPS-C) and positive-incentive point sampling with high geometric stability (PIPS-S). The PIPS-C estimation component contains a point cloud-based encoder and a volumetric grid-based decoder. It learns to generate sample points with high estimation certainty. As a result, training the  $SO(3)$ -equivariant convolutional implicit network on the PIPS-C sample points would bring sufficient information gain (Figure 1b). Using all the PIPS-C sample points is neither efficient nor necessary. To solve this problem, the PIPS-S estimation component further selects the sparse and geometrically stable subsets from the above sample points (Figure 1c). The PIPS-S estimation component is implemented with an attentional gating module trained by the Gumbel-Softmax trick. A stability loss function and a sparsity loss function are applied to optimize the sample points with high geometric stability while keeping the sample points sparse. As

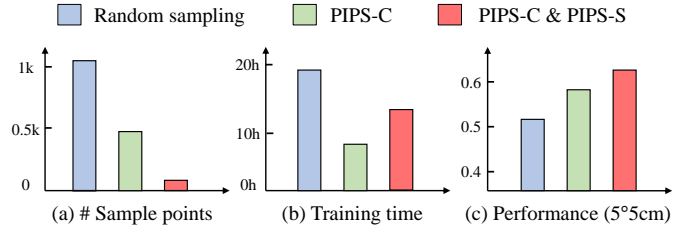


Fig. 3: The quantitative comparisons of the proposed PIPS-C and PIPS-S to the baseline of random sampling. We see our method reduce the number of sample points and the training time while achieving better performance in object pose estimation. The experiment is conducted on the NOCS-REAL275 dataset.

reported in Figure 3, the proposed PIPS greatly reduces the number of sample points and the training time while achieving better performance in pose estimation.

The PIPS estimation network is trained in a knowledge distillation manner, where a sophisticated teacher model is first trained to generate the pseudo ground-truth and a student model (i.e. the PIPS estimation network) is then optimized to mimic the teacher model. Interestingly, we found that using a small set of pseudo ground-truth is sufficient for the PIPS estimation network to generate meaningful and repeatable positive incentive points.

Experiments demonstrate that the proposed method outperforms the state-of-the-art on three datasets of pose estimation. It achieves 0.63 in the  $5^\circ 2\text{cm}$  metric on NOCS-REAL275, 0.62 in the  $5^\circ 5\text{cm}$  metric on ShapeNet-C, and 77.3 in the AR metric on LineMOD-O. It has better performances in various challenging scenarios, such as objects captured with unseen pose, high occlusion, novel geometry, and severe noise. Moreover, we provide in-depth qualitative analyses of the learned sampling strategy. We also demonstrate the cross-task generality of the learned sampling strategy by applying the trained PIPS estimation network to other tasks with neural implicit fields, such as implicit shape reconstruction.

In summary, we make the following contributions:

- We propose the idea of detecting positive-incentive sample points for neural implicit fields, that improve the accuracy and training efficiency.
- We propose an  $SO(3)$ -equivariant convolutional implicit network to estimate point-level attributes, achieving better performance compared to most existing implicit neural fields in pose estimation.
- We develop the PIPS estimation network, including a PIPS-C estimation component and a PIPS-S estimation component, to generate sample points with high estimation certainty and high geometric stability, respectively.
- Our method achieves state-of-the-art performance on three datasets of pose estimation. In particular, it performs well in various challenging scenarios.

## 2 RELATED WORK

**Neural Implicit Representation of 3D Shapes.** Many existing works have investigated implicitly representing 3D shapes with continuous and memory-efficient implicit fields that map  $(x, y, z)$  coordinates to signed distance fields [45], [48] or occupancy functions [10], [44], implemented by neural networks. These

neural implicit fields of 3D shapes allow high-quality surface representation with any resolution, facilitating the application of them in shape reconstruction. Combined with advanced network architectures [15], [51], [61] and training schemes [7], [22], neural implicit fields are not limited to single objects with simple geometry but also could scale to large-scale complicated scenes. In addition to representing geometry, recent works have explored encoding appearance with neural implicit fields [3], [4], [46], [76], achieving state-of-the-art visual quality of novel view image synthesis. Despite the progress, representing 3D shapes with neural implicit fields usually requires training on densely sampled points, some of which are less informative and would increase the computational cost and lead to less accurate estimation.

**Neural Implicit Fields for Pose Estimation.** While most of the research in neural implicit fields focuses on shape reconstruction and view synthesis, some other methods adopt implicit fields for object pose estimation [1], [52]. A conventional and straightforward solution is to reconstruct the object surface and estimate its pose simultaneously [6], [32], [49], [70] so that the two tasks could boost each other. For example, ShAPO [20] jointly predicts object shape, pose, and size in a single-shot network. DISP6D [70] disentangles the latent representation of shape and pose into two sub-spaces, improving the scalability and generality. With the mechanism of representing complex 3D geometry from a set of RGB images, Neural Radiance Fields (NeRF) is also applicable to optimize the object pose w.r.t a single-view image. For example, iNeRF [74] uses gradient descent to minimize the residual between pixels rendered from a pre-trained NeRF and pixels in an observed image. NeRF-Pose [31] first reconstructs the object from multiple views in the form of a neural implicit representation and then regresses the object pose by predicting pixel-wise 2D-3D correspondences between images and the reconstructed model. Unlike the traditional correspondence-based methods which predict 3D object coordinates merely at observed pixels in the input image, Huang et al. [18] predicts canonical coordinates at any sampled 3D in the camera frustum, generating continuous neural implicit fields of canonical coordinates for instance-level pose estimation. Wan et al. [18] extend the idea of dense per-point estimation to category-level pose estimation by proposing a semantically-aware canonical space and a transformer-based feature propagation module. Our method is inspired by the previous works of dense per-point estimation. However, it generates positive-incentive sample points with a learning-based method which could improve the training efficiency of the implicit neural networks.

**Point Sampling Strategy in Neural Implicit Fields.** Training neural implicit fields for 3D shapes is challenging and time-consuming as it requires large sample counts to cover the region both inside and outside the surface, especially for 3D shapes with complex geometry. Various sampling strategies have been adopted in neural implicit fields to achieve better training efficiency. For 3D reconstruction, uniform sampling and near-surface sampling that selects a certain number of points in the boundary space or near the underlying surface, are widely used in a large number of previous works. Some works adopt a combination of them to achieve balanced training [56]. Xu et al. [72] propose a farthest point sampling algorithm resulting in a fast network. Several adaptive point sampling strategies are developed to find the hard training points, allowing faster convergence and accurate representation of geometry details [23], [73], [75]. There are also a bunch of works that focus on developing sophisticated point sampling on NeRFs. Mildenhall et al. [46] utilize a hierarchical sampling procedure

by training an extra coarse network, allowing efficient network training. Li et al. [33] and Sun et al. [60] discretize the scene into voxels and compute the importance of each voxel to the rendered image. Therefore, the method can skip the invalid areas and avoid unnecessary computation. Training neural implicit fields to directly sample in an end-to-end manner is another direction [4], [29], [39], [53]. These methods require additional time to train the sampling network and would not generalize well in unseen scenarios. Our method is relevant to those methods. However, it is designed specifically for pose estimation from a novel prospect and can be generalized in other relevant tasks.

**Equivariant Network for Point Cloud Analysis.** The equivariance property is crucial for point cloud analysis. Various approaches were proposed to address this problem. A straightforward way is to estimate the orientation from the input so that the equivariance can be obtained. There are a bunch of existing works in this direction, including object orientation estimation [16], [50], [66] and local patch orientation estimation [54], [55], [71]. Recently, convolutions with steerable kernel bases have emerged [2], [43], [68], [69], using additional storage and specialized operations to guarantee equivariance for common network layers. For example, 3D Steerable CNNs adopt the convolution with steerable kernel bases [69]. This convolution results in a rotation of the features in the feature space, inheriting equivariance from input points to output features. Vector-based neural network is another direction [11], [24], [58]. For example, vector neurons [11] extend the 1D scalars to 3D vectors, enabling the mapping of  $SO(3)$  actions to the feature space. Our method adopts the idea of vector neurons. However, we make extensions to make it applicable for the 3D graph convolution layers, greatly broadening its application scope.

### 3 METHOD

#### 3.1 Definition

As discussed in the introduction, densely sampling points across the entire camera space often leads to inaccurate estimations that can hinder the learning process and compromise overall performance. This issue arises primarily due to the presence of points with indistinctive or less-informative features, which provide unreliable signals.

To tackle this challenge, we introduce positive-incentive point sampling (PIPS), a strategic sampling approach designed to selectively identify points that actively contribute to and facilitate effective network training. PIPS refers to a process that predicts sample points that would positively incentivize the learning process and the overall performance of implicit neural networks.

Specifically, the sample points of PIPS are required to satisfy three key criteria: 1) *Distinctive features*: Each sample point must exhibit distinctive and discriminative features that enable high-certainty estimation of its canonical coordinate; 2) *Sparsity*: The sample points should be sparse to maintain computational efficiency and avoid unnecessary redundancy; 3) *Inter-point complementarity*: The set of sample points should collectively provide sufficient information to constrain all DoFs of the object's pose.

#### 3.2 Overview

Our method consists of two crucial modules: the  $SO(3)$ -equivariant convolutional implicit network and the PIPS estimation network. The  $SO(3)$ -equivariant convolutional implicit network is a backbone that estimates point-wise object canonical

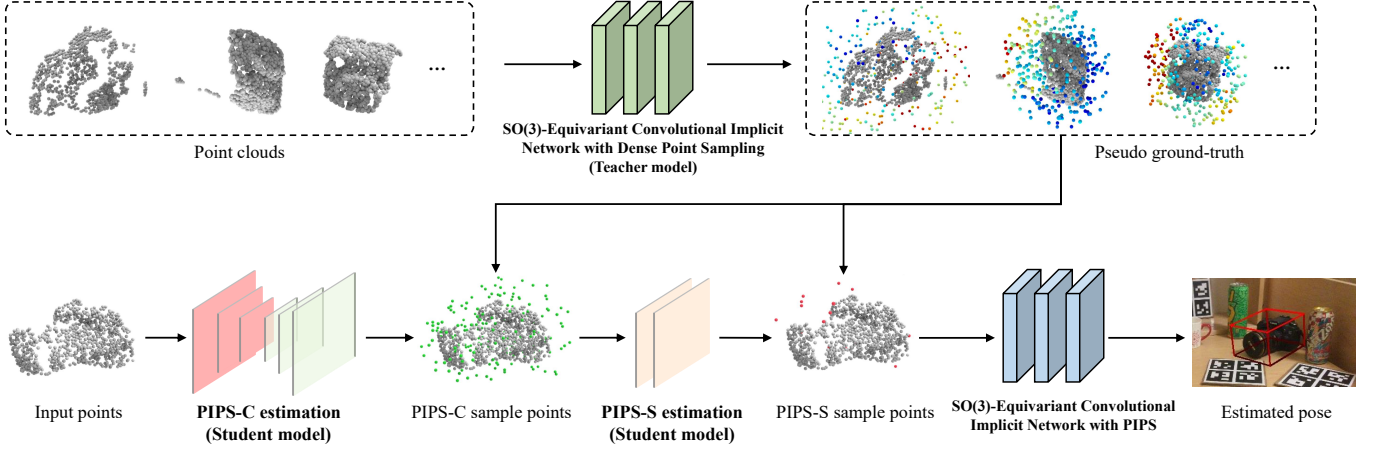


Fig. 4: Overview of the proposed method. First, an  $SO(3)$ -equivariant convolutional implicit network with dense point sampling (the teacher model) is optimized to generate the pseudo ground-truth. Second, the PIPS-C and PIPS-S estimation networks (the student model) are trained based on the generated pseudo ground-truth. Third, an  $SO(3)$ -equivariant convolutional implicit network is trained with the sample points estimated by the PIPS estimation network.

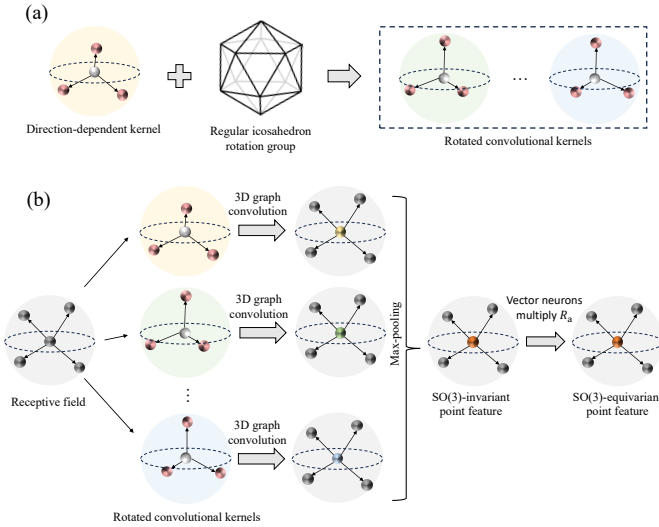


Fig. 5: (a) By rotating the 3D graph convolution kernel via a regular icosahedron rotation group, we generate a set of convolutional kernels. (b) Point cloud convolutions with the rotated convolutional kernels make the generated features  $SO(3)$ -invariant. By generating vector neurons by the  $SO(3)$ -invariant point feature and multiplying the vector neurons with the rotation matrix  $R_a$  corresponding to the  $q_a \in Q$  with the highest activation, the feature becomes  $SO(3)$ -equivariant.

coordinates at any query location based on the input points. The PIPS estimation network includes two main components: PIPS-C and PIPS-S, which are two successive modules that boost the training of the  $SO(3)$ -equivariant convolutional implicit network. Specifically, PIPS-C generates sample points with high estimation certainty. After that, PIPS-S further selects the sparse and geometrically stable subset from the sample points of PIPS-C, resulting in a more parsimonious pointset. An overview of the method is illustrated in Figure 4.

### 3.3 $SO(3)$ -Equivariant Convolutional Implicit Network

While previous non-equivariant neural networks could produce satisfactory point-level predictions, their training typically required data augmentation to ensure the training data represented a sufficiently diverse range of poses over the  $SO(3)$  group.  $SO(3)$ -equivariant neural networks reduce the model complexity, accelerating the training process and leading to more robust predictions. Recently, various methods have been developed to provide  $SO(3)$ -equivariance to basic neural network layers, but few can incorporate sophisticated layers, such as 3D convolution layers. To facilitate an effective feature extraction for 3D convolution layers, we propose an  $SO(3)$ -equivariant 3D graph convolution layer.

Implementing 3D convolution with  $SO(3)$ -equivariance presents several challenges. First, 3D convolutions aggregate information from local neighborhoods, as the relative positions and orientations of neighboring points change under rotation. The characteristic inherently breaks  $SO(3)$ -equivariance. Second, implementing 3D convolution with continuous  $SO(3)$ -equivariance requires substantial memory and computation cost. The proposed  $SO(3)$ -equivariant 3D graph convolution is a kernel-based approach that allows us to compute the convolution with a limited number of rotations, maintaining a balance between representation ability and computational cost.

#### 3.3.1 $SO(3)$ -equivariant 3D graph convolution layer

The  $SO(3)$ -equivariant 3D graph convolution layer takes a point cloud  $P$  as input and generates the per-point convolutional  $SO(3)$ -equivariant features. To achieve this, we adopt the idea of vector neurons [11] that extend each 1D scalar neuron in the network to a 3D vector. The vanilla vector neurons allow direct mapping of rotations on several basic layer types, including the linear layer, the pooling layers, and the normalization layers. To enforce  $SO(3)$ -equivariance in point cloud convolution while maintaining a balance between performance and computational overhead, we extend the vector neurons to make them applicable to the 3D graph convolution layers [38].

The process of the proposed  $SO(3)$ -equivariant 3D graph convolution is shown in Figure 5. Suppose the feature of each point  $p_n$  in  $P$  is  $f(p_n) \in \mathbb{R}^{C \times 3}$ . The receptive field of  $p_n$  is  $\Pi_n^M$ .  $M$  is the number of points in the receptive field. The learnable 3D



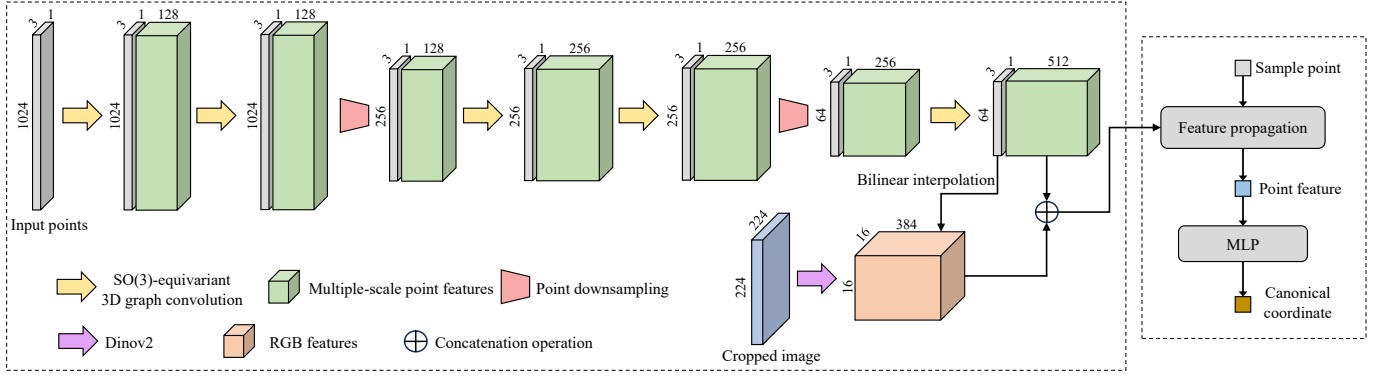


Fig. 6: Architecture of the SO(3)-equivariant convolutional implicit network. Left: Multiple SO(3)-equivariant 3D graph convolution layers and point downsampling layers are adopted to aggregate multiple-scale point features. The RGB features are extracted by DINOv2. Right: The aggregate features are propagated to any 3D sample point via feature propagation layers, estimating the point-wise object canonical coordinate.

convolution kernel  $K_U = \{k_0, k_1, k_2, \dots, k_U\}$  composes of one center point and  $U$  support points.  $k_0 = (0, 0, 0)$  is the center point.  $k_u$  ( $u \in \{1, 2, \dots, U\}$ ) are the support points in the kernel. The 3D graph convolution is:

$$\text{Conv}(\Pi_n^M, K_U) = \langle f(p_n), w(k_0) \rangle + \Theta, \quad (1)$$

where  $w(k_0) \in \mathbb{R}^{C \times 3}$  is the weight matrix of  $k_0$ .  $\langle \cdot, \cdot \rangle$  is the feature distance.  $\Theta$  measures the similarity between  $\Pi_n^M$  and the support points in  $K_U$ . Specifically,  $\langle \cdot, \cdot \rangle$  is the sum of the inner-product operation of each 1D neuron in  $f(p_n)$  and  $w(k_0)$ .  $\Theta$  is computed as:

$$\Theta = \max_{q \in Q} \sum_{u=1}^U \max_{m \in \{1, \dots, M\}} \text{sim}(p_m, k_u^q). \quad (2)$$

$Q$  is a rotation group. We implement it with the regular icosahedron rotation group.  $k_u^q$  is the support point rotated by  $q \in Q$ .  $p_m$  ( $m \in \{1, 2, \dots, M\}$ ) is point in the receptive field  $\Pi_n^M$ .  $\text{sim}(\cdot)$  is the similarity between the feature of  $p_m$  and the weight of  $k_u^q$ :

$$\text{sim}(p_m, k_u^q) = \langle f(p_m), w(k_u^q) \rangle \frac{\langle d_{m,n}, k_u^q \rangle}{\|d_{m,n}\| \cdot \|k_u^q\|}, \quad (3)$$

where  $d_{m,n}$  denotes the direction from point  $p_n$  to point  $p_m$ . Note that  $k_u$  ( $u \in \{1, 2, \dots, U\}$ ),  $w(k_0)$ , and  $w(k_u)$  ( $u \in \{1, 2, \dots, U\}$ ) are learnable elements which are optimized during network training.

As Equation 2 selects the rotation  $q_{\text{activation}} \in Q$  with the highest activation, the operation in Equation 1 is SO(3)-invariant. To further make the 3D graph convolution SO(3)-equivariant, we generate a 3D vector neuron by duplicating the output of Equation 1. We then multiply the 3D vector neuron with the rotation matrix  $R_{\text{activation}}$  corresponding to the rotation  $q_{\text{activation}}$ , which encodes the directional information into the output.

### 3.3.2 Network architecture for pose estimation

The architecture of the SO(3)-equivariant convolutional implicit network is shown in Figure 6. To train a network to output point-wise object canonical coordinates from which the object pose could be estimated, we apply multiple SO(3)-equivariant 3D graph convolution layers to aggregate multiple-scale point features. To be specific, the network contains five SO(3)-equivariant 3D graph convolution layers and two point downsampling layers. To leverage the information in the input RGB image, we crop the

image to make it only contain the target object and feed it into the DINOv2 [47] to extract the RGB features. The downsampled point cloud is projected onto the RGB feature maps to fetch its RGB features with a bilinear interpolation. The point features, as well as the RGB features, are concatenated to predict the point-wise object canonical coordinates of the input points. Moreover, to facilitate the object canonical coordinate estimation at any 3D sample points, we adopt the feature propagation layers in [65] to estimate features at query locations. Note that, all the 1D scalar neurons in the feature propagation layers are represented as 3D vectors to guarantee SO(3)-equivariance. The propagated features are then converted to SO(3)-invariant features with the invariant layer in [11] and used to predict the object's canonical coordinates with an MLP. The network is trained on the sample points with the mean squared error loss function. The object pose is then estimated with the method in [67].

### 3.4 PIPS Estimation Network

Having introduced the SO(3)-equivariant convolutional implicit network, we then describe the PIPS estimation network that generates sampling points to positively incentivize the training of it. The PIPS estimation network is trained in a knowledge distillation manner, where a sophisticated teacher model is first trained to generate the pseudo ground-truths and the PIPS estimation network is then optimized to learn from those. In the following, we first describe the process of generating the pseudo ground-truth and then elaborate on the network architecture of the PIPS estimation network.

#### 3.4.1 Generating pseudo ground-truth

Manually annotating the positive-incentive sampled points is infeasible due to the infinite potential sample locations and the lack of explicit labeling rules. Hence, we propose to achieve this by optimizing a neural network that automatically generates those points. The neural network (i.e. the teacher model) is implemented with the SO(3)-equivariant convolutional implicit network with an extra point-wise uncertainty estimation mechanism. Specifically, we adopt the dense sampling strategy which trains the teacher model on random locations  $\Phi$  spread near the input point cloud. For each  $\phi \in \mathbb{R}^3$  in  $\Phi$ , we not only estimate its object canonical coordinate but also add an output head to estimate a Gaussian distribution  $(x_\phi, \sigma_\phi)$  to represent the uncertainty, where  $x_\phi \in \mathbb{R}^3$

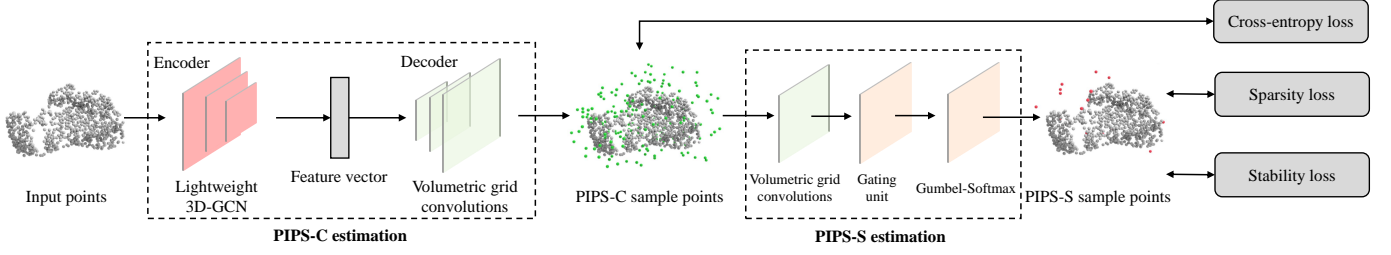


Fig. 7: The network architecture of the PIPS estimation network. The network takes a point cloud as input, embeds it into a high-dimensional feature space with an encoder, and generates the PIPS-C sample points with a decoder. A gating unit and a Gumbel-Softmax module are applied to generate the PIPS-S sample points. Multiple loss functions are adopted to train the network.

and  $\sigma_\phi \in \mathbb{R}$ . The network could be trained with a point-level adversarial loss function:

$$\mathcal{L}_{\text{ADV}} = \frac{1}{\sigma_\phi^2} \|x_\phi - \hat{x}_\phi\|^2 + \log \sigma_\phi^2, \quad (4)$$

where  $\hat{x}_\phi$  is the ground-truth object canonical coordinate. The intuition of the loss function is to encourage accurate object canonical coordinate estimation when the point feature is certain [26]. In such cases, the point is positive-incentive since it would bring sufficient information gain by training on it. Otherwise, it encourages high variance  $\sigma_\phi$ , implying that it is not positive-incentive.

Despite the ability of the above loss function to filter the less informative sample points, we found the point-level uncertainty estimation should be more fine-grained. As such, the variance  $\sigma_\phi$  should be anisotropic, i.e. contains uncertainties along each direction.

In our method, instead of estimating a scalar variance  $\sigma$ , the network predicts a scaling matrix  $S \in \mathbb{R}^{3 \times 3}$  and a quaternion  $q$  that represents a rotation  $R \in \mathbb{R}^{3 \times 3}$ . The rotated covariance matrix is represented as  $\Sigma_\phi = RSS^T R^T$ . Considering the anisotropic variance, the point-level adversarial loss function could be computed by the Kullback-Leibler divergence for multivariate Gaussian distributions [13]:

$$\mathcal{L}_{\text{ADV}} = (x_\phi - \hat{x}_\phi)^T \Sigma_\phi^{-1} (x_\phi - \hat{x}_\phi) + \ln |\Sigma_\phi| + \text{tr}(\Sigma_\phi^{-1}), \quad (5)$$

where  $\text{tr}(\cdot)$  is the trace of the matrix.

We train the teacher model until convergence. For each object, we label the points with  $\text{tr}(SS^T) < \omega$  as positive, and vice versa.  $\omega$  is the threshold. To facilitate the training of PIPS estimation network, the generated labels by the teacher model at random sample points  $\Phi$  are converted into the labels of the volumetric grids. The label of any center point in the voxels is computed by voting considering all the sample points in this voxel.

We dub the generated labels *pseudo ground-truth*, as they are generated by an auxiliary task, cannot be rigorously defined, and might be inaccurate. Nevertheless, we found the generated samples are not only geometrically meaningful but also can boost the performance. Please refer to the experiment section for the quantitative and qualitative comparisons.

### 3.4.2 Network architecture

As shown in Figure 7, the PIPS estimation network takes the point cloud  $P$  as well as the corresponding RGB image as input and outputs positive-incentive sample points. We divide the network into two sequential components PIPS-C and PIPS-S

that generate sample points with high estimation certainty and sample points with high geometric stability, respectively.

**PIPS-C estimation.** The PIPS-C estimation component takes  $P$  as input, embeds it into a high-dimensional feature space  $\mathbb{R}^d$  with a point cloud based encoder, and generates sample points with a volumetric grid based decoder. Each voxel in the output volumetric grid contains a label indicating whether it is a valid positive-incentive sample point. We adopt 3D-GCN [38] as the encoder and the convolutional occupancy networks as the decoder [51]. In the 3D-GCN, to leverage the color information in the input RGB image, we crop the image to make it only contain the target object and feed it into the DINOv2 [47] to extract the RGB features. The downsampled point cloud is projected onto the RGB feature maps to fetch its RGB features. The output volumetric grid  $V$  shares the same center as  $P$  and includes  $h^3$  voxels. We set the side length of  $V$  as double of the diagonal length of the target object, for instance-level pose estimation, and double of the diagonal length of the categorical mean shape [62], for category-level pose estimation, respectively.

The following loss function is applied to optimize the network:

$$\mathcal{L}_{\text{PIPS-C}} = \sum_{v \in V} \mathcal{L}_{\text{CE}}(o_v, \hat{o}_v), \quad (6)$$

where  $\mathcal{L}_{\text{CE}}(\cdot)$  is the cross-entropy loss.  $o_v \in \{0, 1\}$  is the estimated label of voxel  $v$ .  $o_v = 1$  indicates the center of the voxel is a positive-incentive sample point, and vice versa.  $\hat{o}_v$  is the pseudo ground-truth, which is pre-generated in Section 3.4.1. We denote the sample points generated by PIPS-C estimation component as  $P_{\text{PIPS-C}}$ .

**PIPS-S estimation.** The PIPS-C estimation component generates sample points with high estimation certainty and would boost the training of the SO(3)-equivariant convolutional implicit network. Nevertheless, it is unnecessary to use all of them. To further generate parsimonious sample points, we propose the PIPS-S estimation component.

We first mask the feature of the last layer in the PIPS-C estimation component by assigning 0 to the feature of voxels with  $o_v = 0$ . Taking the masked feature as input, the PIPS-S estimation component adopts two volumetric grid convolution layers, followed by a ReLU layer, to aggregate the feature  $F_{\text{PIPS-C}}$  of  $P_{\text{PIPS-C}}$ .

A softmax pooling layer is adopted to generate the global feature which is then concatenated with the feature vector in each voxel. A gating unit  $G$  is then applied, estimating a soft gating decisions  $G^{\text{soft}}$  which is a mask to indicate their activations:

$$G^{\text{soft}} = \tanh[G(F_{\text{PIPS-C}}) + \eta], \quad (7)$$

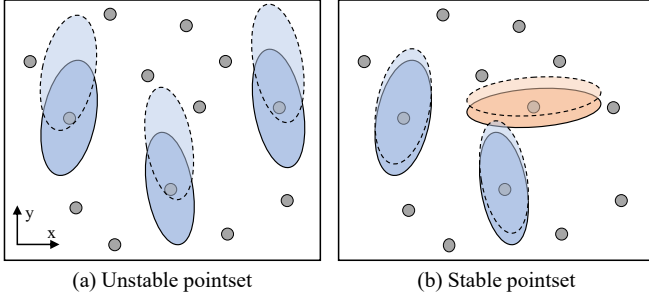


Fig. 8: A 2D example of PIPS-S selection. The dots are the input to the PIPS-S estimation component. The dots with an ellipse are the selected PIPS-S sample points, where the ellipse represents its anisotropic variances. (a) The pointset is unstable w.r.t. the alignment because it has a high variance along the y-axis direction, meaning that the locations along the y-axis are not well-constrained. (b) The pointset is stable w.r.t. the alignment since no DoF shows high variance for the points within the set, indicating the locations are well-constrained.

where  $\eta$  is the Gumbel noise, we implement the gating unit  $G$  with 1D MLPs.

Then, a Gumbel-Softmax module GSM [21] turns soft decisions  $G^{\text{soft}}$  into hard decisions  $G^{\text{hard}} \in \{0, 1\}^{\Theta}$  by replacing the softmax with an argmax during the forward pass and retaining the softmax during the backward pass [28], [64]:

$$G^{\text{hard}} = \text{GSM}(G^{\text{soft}}). \quad (8)$$

The hard decision  $G^{\text{hard}}$  is a binary mask that indicates which points in  $P_{\text{PIPS-C}}$  are activated. The Gumbel-Softmax module provides a mechanism that outputs a binary mask in the forward pass and allows the gradient to be back-propagated. Hence, the gating attentional unit is end-to-end trainable. We denote the activated points selected by the gating unit  $G^{\text{hard}}$  as  $P_{\text{PIPS-S}}$ . The PIPS-S estimation component is trained with the sparsity loss function  $\mathcal{L}_{\text{Sparsity}}$  and the stability loss function  $\mathcal{L}_{\text{Stability}}$ :

$$\mathcal{L}_{\text{PIPS-S}} = \mathcal{L}_{\text{Sparsity}} + \mathcal{L}_{\text{Stability}}. \quad (9)$$

The sparsity loss function considers the number of the activated points. Suppose  $\#P_{\text{PIPS-S}}$  and  $\#P_{\text{PIPS-C}}$  are the point number of  $P_{\text{PIPS-S}}$  and  $P_{\text{PIPS-C}}$ , respectively.  $\mathcal{L}_{\text{Sparsity}}$  is computed as the KL divergence between the pre-defined empirical sparsity  $g = \frac{\#P_{\text{PIPS-S}}}{\#P_{\text{PIPS-C}}}$  and a target sparsity  $\rho \in [0, 1]$ :

$$\mathcal{L}_{\text{Sparsity}} = KL(\rho \| g) = \rho \log\left(\frac{\rho}{g}\right) + (1 - \rho) \log\left(\frac{1 - \rho}{1 - g}\right). \quad (10)$$

The stability loss function optimizes the distribution of  $P_{\text{PIPS-S}}$  from the perspective of geometric stability during point cloud alignment [14], [57]. In the context of aligning two point clouds (here, the input point cloud and the object point cloud in canonical space), the input point cloud is considered geometrically stable w.r.t. the alignment if all DoFs are well-constrained, meaning that no DoF exhibits high variance. Conversely, the input point cloud is deemed geometrically unstable w.r.t. the alignment if any DoF shows high variance.

A 2D example is provided in Figure 8. The dots represent the input to the PIPS-S estimation module. Dots marked with ellipses indicate the selected PIPS-S sample points, where each

ellipse illustrates the anisotropic variance of the corresponding point. In Figure 8(a), the point set is geometrically unstable during alignment due to high variance along the y-axis. This implies that the vertical positions are not well-constrained, allowing the point cloud to slide along that direction. In contrast, the point set in Figure 8(b) is geometrically stable, as no DoF displays significant variance across the samples, indicating that all point locations are sufficiently constrained.

We then describe how to compute the stability in practice. Suppose  $p \in P_{\text{PIPS-S}}$ .  $x_p$  is the coordinate of  $p$ .  $\Sigma_p = R_p S_p S_p^T R_p^T$  is the rotated covariance matrix of  $p$  pre-computed in Section 3.4.1. We compute the stability of pointset  $P_{\text{PIPS-S}}$  under point-wise anisotropic uncertainties  $\{\Sigma_p\}$  as follows.

When a perturbation transformation  $[\Delta R, \Delta t]$  is applied to  $P_{\text{PIPS-S}}$ , the movement of the pointset  $\Delta E_{\text{PIPS-S}}$  considering the point-wise anisotropic uncertainties is defined as:

$$\Delta E_{\text{PIPS-S}} = \sum_{p \in P_{\text{PIPS-S}}} (\Delta E_p^T \Delta E_p), \quad (11)$$

where  $\Delta E_p$  is the per-point movement:

$$\Delta E_p = S_p^{-1} R_p^T (\Delta R x_p + \Delta t - x_p), \quad (12)$$

where  $S_p^{-1} R_p^T$  incorporates directional weighting from the anisotropic uncertainty:  $R_p^T$  rotates the displacement into the principal-axis frame of the covariance, and  $S_p^{-1}$  scales each principal direction by the reciprocal of its standard deviation, assigning a lower weight to directions with higher uncertainty.

Next, we approximate Equation (11) into the quadratic form:

$$\Delta E_{\text{PIPS-S}} = [\Delta r \ \Delta t]^T C [\Delta r \ \Delta t], \quad (13)$$

where  $\Delta r \in \mathbb{R}^3$  denotes the rotation vector associated with the rotation matrix  $\Delta R \in \mathbb{R}^{3 \times 3}$ .  $C$  is the covariance matrix accumulated from  $x_p$  and  $\Sigma_p$ . It encodes the increase in the alignment error when the transformation is moved away from its optimum [14].

The relation between the rotation vector  $\Delta r$  and the rotation matrix  $\Delta R$  can be expressed by expanding  $\Delta R$  via the Taylor series:

$$\Delta R = I + \Delta r^\wedge + \frac{(\Delta r^\wedge)^2}{2!} + \frac{(\Delta r^\wedge)^3}{3!} + \dots, \quad (14)$$

where  $\Delta r^\wedge$  is the skew-symmetric matrix representation of  $\Delta r$ . Retaining only the first-order term in (14) and substituting into (12) gives:

$$\begin{aligned} \Delta E_p &\approx S_p^{-1} R_p^T [(I + \Delta r^\wedge) x_p - x_p + \Delta t] \\ &= S_p^{-1} R_p^T [\Delta r^\wedge x_p + \Delta t] \\ &= S_p^{-1} R_p^T [-x_p^\wedge \Delta r + \Delta t] \\ &= \begin{bmatrix} -S_p^{-1} R_p^T x_p^\wedge & S_p^{-1} R_p^T \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta t \end{bmatrix}, \end{aligned} \quad (15)$$

where we used the property  $(\Delta r)^\wedge x_p = -x_p^\wedge \Delta r$ .

Substituting the above Equation (12) of  $\Delta E_p$  into Equation (11) yields the quadratic form in Equation (13), where the matrix  $C \in \mathbb{R}^{6 \times 6}$  is given by:

$$C = C_1 C_2, \quad (16)$$

$$C_1 = \begin{bmatrix} x_1^\wedge R_1 S_1^{-1} & \cdots & x_k^\wedge R_k S_k^{-1} \\ R_1 S_1^{-1} & \cdots & R_k S_k^{-1} \end{bmatrix}, \quad (17)$$

$$C_2 = \begin{bmatrix} -S_1^{-1} R_1^T x_1^\wedge & S_1^{-1} R_1^T \\ \vdots & \vdots \\ -S_k^{-1} R_k^T x_k^\wedge & S_k^{-1} R_k^T \end{bmatrix}, \quad (18)$$

where the sign change in  $C_1$  follows from the skew-symmetric property  $(x_p^\wedge)^T = -x_p^\wedge$ . Here,  $x_1, \dots, x_k$  are the coordinates of the points in  $P_{\text{PIPS-S}}$ , and  $x^\wedge$  denotes the skew-symmetric matrix associated with  $x$ .

The larger the pointset movement  $\Delta E_{\text{PIPS-S}}$  increase, the greater the stability, since the error landscape will have a deep, well-defined minimum. On the other hand, if there are perturbation transformations that only lead to a small increase in pointset movement, the pointset will be unstable concerning one or more DoFs of the perturbation transformation.

The eigenvectors of  $C$  could reflect the pointset stability. If all eigenvalues of  $C$  are large, any transformation away from the minimum will result in a relatively large increase in pointset movement. Let  $\lambda_1, \dots, \lambda_6$  be the eigenvalues of matrix  $C$ . To make the pointset stable, all the eigenvalues should be large. As a result, the stability loss function penalizes small eigenvalues:

$$\mathcal{L}_{\text{Stability}} = \sum_{i=1}^6 e^{-\lambda_i}. \quad (19)$$

An empirical explanation of the above two loss functions is to find sample points with high confidence that are sufficient to determine all the DoFs of the object pose while keeping the number of sample points small. It is worth mentioning while there might be multiple potential outputs that will lead to the two loss functions being small, we only require the network to output one of them. In practice, we found that using a small set of pseudo ground-truth is sufficient for the PIPS estimation network to generate meaningful and repeatable positively incentive points. The reason might be that different shapes share similar geometric patterns. The learned sampling strategy is generalizable across various objects.

### 3.5 Training and Inference

The training of the proposed method contains three stages, as shown in Figure 4. First, an SO(3)-equivariant convolutional implicit network with dense point sampling (i.e. the teacher model) is optimized to generate the pseudo ground-truth. Second, the PIPS estimation network (i.e. the student model) is trained based on the generated pseudo ground-truth. Third, another SO(3)-equivariant convolutional implicit network is trained with the sample points generated by the PIPS estimation network.

Despite the relatively high computational cost during the pseudo ground-truth generation procedure, this process is optimized before the neural implicit field with SO(3)-equivariant convolutions is trained. Moreover, an interesting phenomenon we observe is that a small number of pseudo ground-truth is sufficient for the PIPS estimation network, showing that the trained SO(3)-equivariant convolutional implicit network is effective and has

good cross-instance and cross-category generalities. Once the pseudo ground-truth is generated, the PIPS estimation network could be efficiently trained, facilitating the efficient training of downstream networks, e.g. the SO(3)-equivariant convolutional implicit network.

During inference, given an input point cloud, our method generates the positive-incentive sample points by the PIPS estimation network and predicts the per-point canonical coordinate on those points. The object pose is then computed from the estimated canonical coordinates by a modified Umeyama algorithm [63] considering the anisotropic uncertainty.

### 3.6 Implementation details

We implement our method with PyTorch. The networks are trained with the Adam optimizer on a single A100 GPU. The learning rate is  $10^{-3}$ . For category-level pose estimation, we train separate networks for each category. For the instance-level pose estimation, we train separate networks for each instance. The training of the generating pseudo ground-truth, PIPS-C estimation, and PIPS-S estimation components take about 6, 2, and 5 hours, respectively. The input point cloud includes 1,024 points. In the PIPS-C estimation component, we set the voxel size  $h$  as 8. In the PIPS-S estimation component, we set the target sparsity  $\rho$  as 0.1. When generating the pseudo ground-truth, we set the threshold  $\omega$  to 0.5. The dense sampling points used to train the teacher model are randomly sampled within a cube in the camera coordinate system. The cube is centered at the center of the input point cloud  $P$ . The cube's side length is twice the diagonal length of the bounding box of the instance or the categorical mean shape. A total of 4,096 points are sampled in the cube. In the SO(3)-equivariant convolutional implicit network, the farthest sampling algorithm was employed to downsample the input point cloud. Each kernel contains 13 support points; the receptive field includes 10 points. We add a small random noise to each of the PIPS-C and PIPS-S sample points when feeding them into the SO(3)-equivariant convolutional implicit network.

## 4 RESULTS AND EVALUATION

We conduct comprehensive experiments to evaluate the proposed method. First, we provide the experimental results on category-level pose estimation (Sec. 4.1) and instance-level pose estimation (Sec. 4.2). Second, ablation and parameter studies (Sec. 4.3) are conducted to analyze the crucial components and parameters of the proposed method. Third, we conduct a pressure test to quantitatively evaluate the robustness of the proposed method (Sec. 4.4). Last, we show the cross-task generality by demonstrating that the learned sampling strategy applies to other relevant tasks (Sec. 4.5).

### 4.1 Evaluation on Category-level Pose Estimation

#### 4.1.1 Experimental setting

Category-level pose estimation needs to consider novel objects that have not been trained, so the sample points should be carefully generated to cover all the underlying shapes. We compare our method with existing methods on category-level pose estimation to show the effects.

The experiments are conducted on two datasets: NOCS-REAL275 and ShapeNet-C. The NOCS-REAL275 dataset [67] is a dataset containing 4.3k training RGB-D images and 2.75k testing RGB-D images captured from multiple real-world scenes.



TABLE 1: The comparisons of ShapeNet-C to existing datasets.

Dataset	Avg. instances per category	Avg. Chamfer distance to mean shape (mm)	Avg. occlusion (%)
NOCS-REAL275 [67]	4	2.7	46
NOCS-CAMERA25 [67]	31	2.4	44
Wild6D [77]	360	2.2	42
HouseCat6D [25]	19	2.8	49
<b>ShapeNet-C</b>	<b>2000</b>	<b>3.1</b>	<b>53</b>

The objects in the dataset belong to 6 categories: bottle, bowl, can, camera, laptop, and mug. Note that, to better evaluate the learning ability on small sets of training data, we did not use the CAMERA training set in [67], which contains a large amount of synthetic data, during training.

To better evaluate the performance of our method in challenging scenarios. We propose a new dataset, i.e., the ShapeNet-C dataset. ShapeNet-C contains 60k training depth images and 6k testing depth images. Each RGB-D image contains one object, so no object detection method is needed, and the performance on pose estimation could be better reported. We set the diagonal length of the bounding box of each object as  $1m$ . The dataset contains object categories with large shape variations, such as airplane, chair, and sofa. More detailed statistical comparisons of ShapeNet-C to existing datasets are provided in Table 1. It shows that ShapeNet-C is more challenging in terms of shape diversity and occlusion. In particular, the test set of the ShapeNet-C dataset contains the following challenging data:

- **Holdout pose:** objects scanned from untrained camera views. We show the distribution of camera locations w.r.t. the object in Figure 9a, where the color of each point indicates the rotation of the camera along the roll axis. We see a large difference in the distribution of the training and testing samples.
- **Novel shape:** objects with large shape variation from those in the training data. We show the distribution of chamfer distance between each object to the categorical meaning shape [62] in Figure 9b. We see that 98% of the samples have a chamfer distance  $>9mm$  to the category’s mean shape.
- **High occlusion:** highly occluded objects with only a small proportion being observed. The distribution of occlusion percentage is given in Figure 9c. The statistics show that 91% of the samples have an occlusion rate exceeding 50%.
- **Severe noise:** objects with severe scan noise. We add random Gaussian noise to the point cloud. The distribution of the standard deviation is reported in Figure 9d. It tells that 79% of the samples have a noise magnitude exceeding 2mm.

We see that the data distributions of the four subsets are quite different from that of the training set. This makes pose estimation on ShapeNet-C extremely challenging.

We use standard metrics to evaluate the performance on the two datasets, respectively. For NOCS-REAL275, we adopt the intersection over union (IoU) under pre-given thresholds, and the average precision of detected instances for which the error is less than  $n^\circ$  for rotation and  $m$  for translation. These metrics are utilized to evaluate the performance of object detection and pose estimation, respectively. For ShapeNet-C, we report the rotational error, and the translational error in the form of mean, and median values. We also report the average precision of instances for which the error is less than  $5^\circ$  for rotation and 5cm for translation.

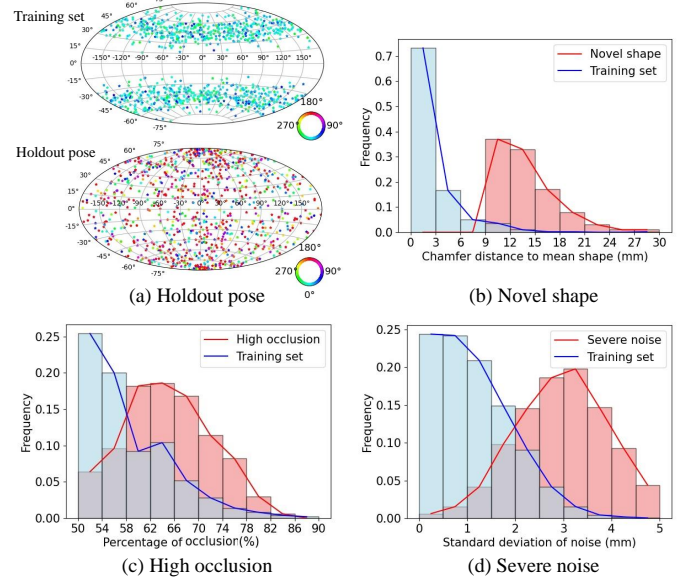


Fig. 9: The statistical comparisons between the training set and the test set of ShapeNet-C.

#### 4.1.2 Quantitative comparison

**Performance on NOCS-REAL275.** We first compare our method to the state-of-the-arts on NOCS-REAL275. The baselines include those with different input modalities and those trained on real data or real & synthetic data. The quantitative comparisons are given in Table 2. It shows that our method outperforms all the baselines on all evaluation metrics. The advantages come from our design of SO(3)-equivariant networks and novel point sampling strategy.

**Performance on ShapeNet-C.** We then compare our method to several representative baselines on ShapeNet-C. The quantitative comparisons are reported in Table 3. It shows that our method achieves state-of-the-art performance.

#### 4.1.3 Qualitative comparison

The qualitative comparisons of the two datasets are visualized in Figure 10 and Figure 11. We see that our method could estimate the pose for most objects accurately, while the baselines cannot. We also provide the visualization of the PIPS sample points in Figure 13a and Figure 13b. The results show some interesting phenomena. First, PIPS-C sample points are dense, distributed around the object surface, possibly around the unseen surface region, such as the occluded chair in Figure 13b. Second, instead of uniformly distributing around the object surface, the sampling rule varies on different shapes, see the laptop in Figure 13a. Third, PIPS-S sample points are sparse, covering the crucial regions of the input shape. Last, adding noise to the input points will not substantially change the distribution of PIPS-S sample points, such as the two cases in the last row of Figure 13b, where the sampled points are speared near the wings and tail of the airplane despite the noise.

## 4.2 Evaluation on Instance-level Pose Estimation

### 4.2.1 Experimental setting

We then evaluate our method on instance-level pose estimation on the public dataset LineMOD-O [5]. LineMOD-O is a widely used dataset for 6D object pose estimation of tabletop objects with

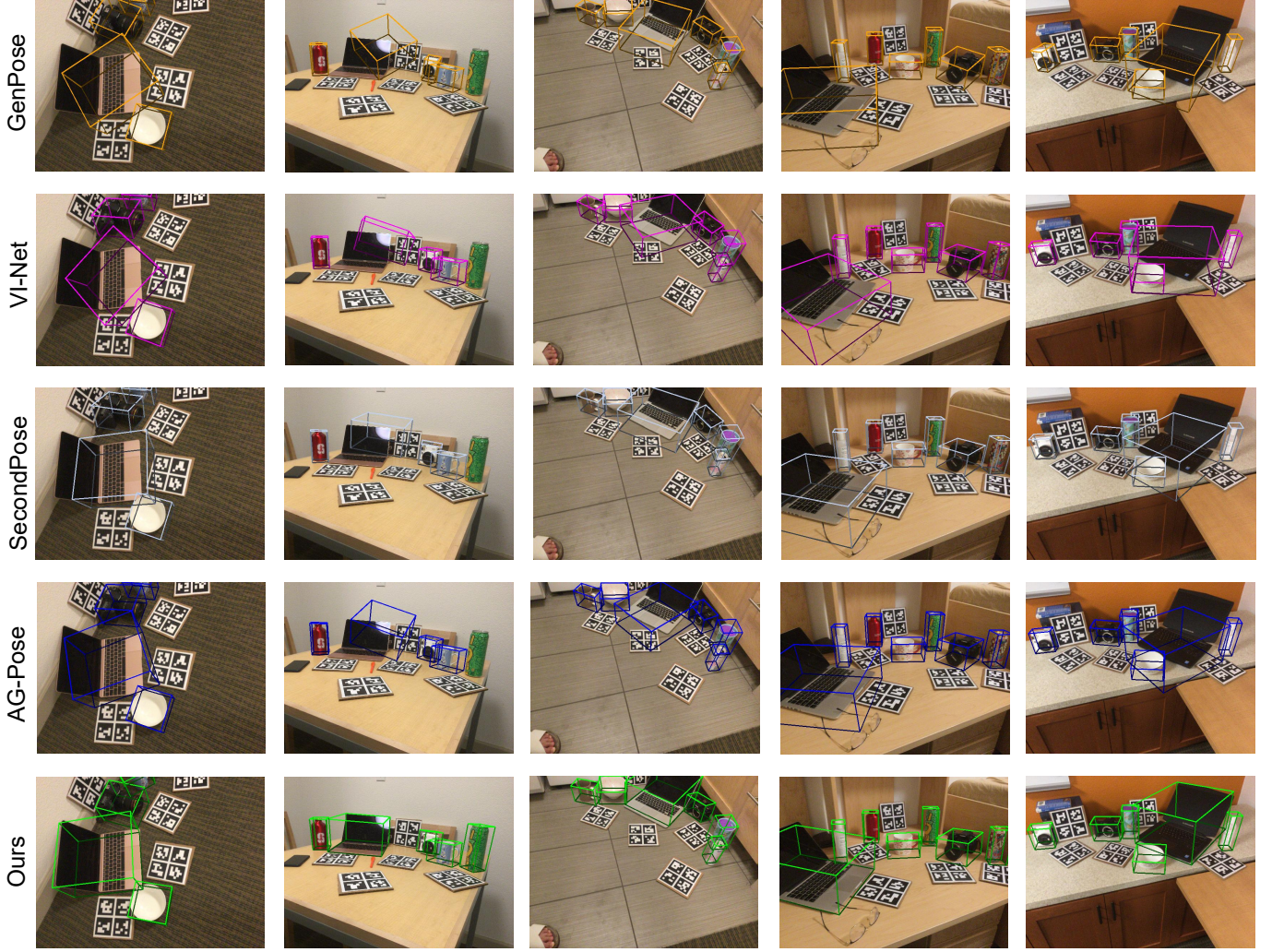


Fig. 10: The qualitative comparisons on the NOCS-REAL275 dataset. Our method could accurately estimate the pose on all the tested objects, while the baselines cannot.

TABLE 2: Quantitative comparisons on the NOCS-REAL275 dataset. ‘\*’ denotes the IoU metrics computed as in [40].

Methods	Input	Data source	Shape Prior	IoU75* $\uparrow$	5°2cm $\uparrow$	5°5cm $\uparrow$	10°2cm $\uparrow$	10°5cm $\uparrow$
SGPA [8]	RGB-D	Syn.+Real	✓	0.37	0.36	0.40	0.61	0.71
DPDN [35]	RGB-D	Syn.+Real	✓	0.54	0.46	0.51	0.70	0.78
HS-Pose [79]	D	Real	×	0.39	0.45	0.55	0.69	0.84
GenPose [78]	D	Real	×	0.50	0.52	0.61	0.72	0.84
VI-Net [36]	RGB-D	Real	×	0.48	0.50	0.58	0.71	0.82
SecondPose [9]	RGB-D	Syn.+Real	×	0.50	0.56	0.64	0.75	<b>0.86</b>
AG-Pose [37]	RGB-D	Syn.+Real	×	0.61	0.57	0.65	0.75	0.85
Ours	RGB-D	Real	×	<b>0.63</b>	<b>0.63</b>	<b>0.68</b>	<b>0.78</b>	<b>0.86</b>

heavy occlusion. We use the average recall (AR) as the evaluation metric [17]. It is computed as the arithmetic mean of the recall rates for three pose-error functions: the visible surface discrepancy (VSD), the maximum symmetry-aware surface distance (MSSD), and the maximum symmetry-aware projection distance (MSPD). The recall for each function is considered correct if the estimated error is less than a predefined threshold.

#### 4.2.2 Quantitative comparison

The quantitative comparison of our method to baselines is reported in Table 4. The baselines include state-of-the-art methods. In the table, AR is the average recall, i.e., the arithmetic mean of the recall rates for the pose-error functions. We see that our method achieves the best performance among all baselines without a refinement procedure. It is slightly inferior to GPose in the AR metric and has an obvious advantage in computational efficiency.



TABLE 3: Quantitative comparisons on the ShapeNet-C dataset.

Methods	Input	Data source	Rotation			Translation		
			Mean( $^{\circ}$ ) $\downarrow$	Median( $^{\circ}$ ) $\downarrow$	$5^{\circ}$ $\uparrow$	Mean(cm) $\downarrow$	Median(cm) $\downarrow$	$5^{\circ}$ 5cm $\uparrow$
NOCS [67]	RGB	Syn.	53.15	27.68	0.34	6.87	5.03	0.29
GPV-Pose [12]	D	Syn.	50.95	20.62	0.38	6.31	4.82	0.33
GenPose [78]	D	Syn.	48.29	11.81	0.41	5.76	4.29	0.37
VI-Net [36]	RGB-D	Syn.	46.45	9.81	0.47	5.55	3.89	0.42
SecondPose [9]	RGB-D	Syn.	45.77	7.87	0.49	5.42	3.91	0.45
AG-Pose [37]	RGB-D	Syn.	40.45	7.49	0.53	5.01	3.59	0.46
Ours	RGB-D	Syn.	<b>30.70</b>	<b>3.16</b>	<b>0.66</b>	<b>3.84</b>	<b>2.13</b>	<b>0.62</b>

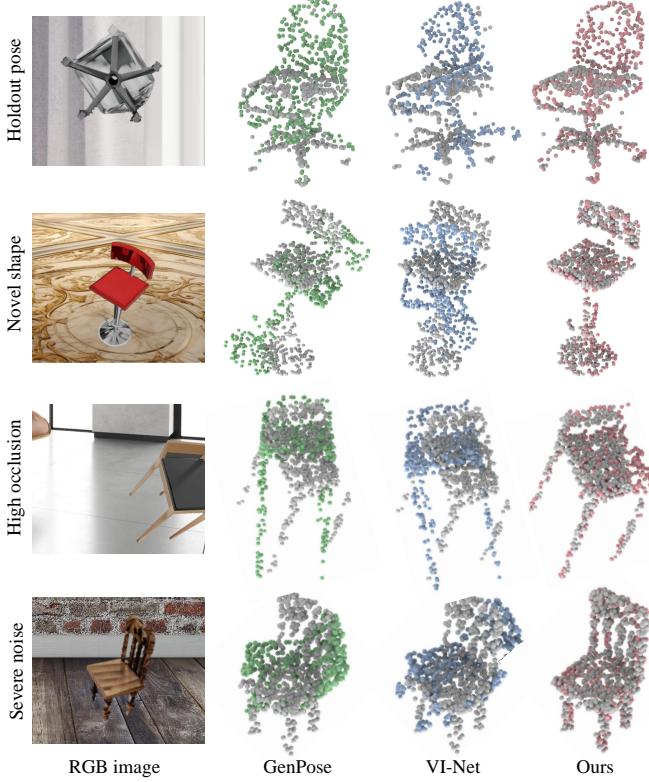


Fig. 11: The qualitative comparisons on the ShapeNet-C dataset. The grey points denote the input points. The colored points represent the canonical shapes transformed with the estimated poses.

The good performance of GPose stems from its mechanism of rendering and comparing for pose refinement, which incurs a significant computational cost.

#### 4.2.3 Qualitative comparison

We visualize the comparison of the estimated pose by our method on the LineMOD-O dataset in Figure 12. It is clear that our method would lead to more accurate pose estimation in challenging scenarios, as highlighted. We also provide visualizations of the sample points generated by PIPS in Figure 13c.

### 4.3 Ablation and Parameter Studies

In Table 5, we study the key components of our method to quantify their efficacy. In Table 6, Table 7, and Table 8, we study the impact

TABLE 4: Quantitative results on the LineMOD-O dataset. The best results are in bold, and the second-best results are underlined.

Methods	Data type	Refinement	AR	time
CDPNv2 [34]	RGB	×	62.4	0.98
NCF [19]	RGB	×	63.2	7.17
ZebraPose [59]	RGB	×	72.1	<b>0.25</b>
CosyPose [30]	RGB-D	✓	71.4	13.74
GDRNPP [41]	RGB-D	×	71.3	<u>0.28</u>
GDRNPP [41] (GPose)	RGB-D	✓	<b>80.5</b>	4.58
Ours	RGB-D	×	<u>77.3</u>	0.39

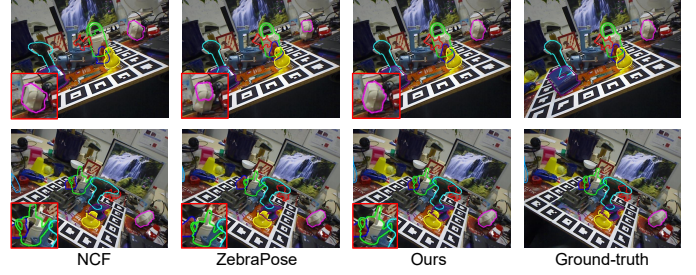


Fig. 12: The qualitative comparisons on the LineMOD-O dataset. The boundary of the shape transformed by the estimated pose is shown in the figure. Please pay attention to the highlighted objects.

of several parameter settings. The experiments are conducted on the ShapeNet-C dataset.

**No PIPS and its components.** The PIPS estimation network, including the PIPS-C and PIPS-S estimation components, is our core contribution. To evaluate the necessity, we turn off the components and retrain the networks. Several conclusions can be drawn from the results. First, the baseline of w/o PIPS uses near-surface sampling instead of PIPS. It is inferior to our full method, confirming the necessity of detecting positive-incentive sample points. However, this baseline achieves better performance when compared to state-of-the-art implicit neuron networks in pose estimation [65] (0.45 in  $5^{\circ}$ 5cm), demonstrating the effects of the proposed SO(3)-equivariant convolutional implicit network. Second, the baseline of w/o PIPS-C does not filter sample points with low estimation certainty, resulting in a substantial performance drop. Third, the baseline of w/o PIPS-S did not consider the geometric stability. Its performance is slightly inferior to our full method. The results demonstrate the need for the PIPS and its two components.

**No anisotropic variance.** To evaluate the effectiveness of the anisotropic variance in generating the pseudo ground-truth, we

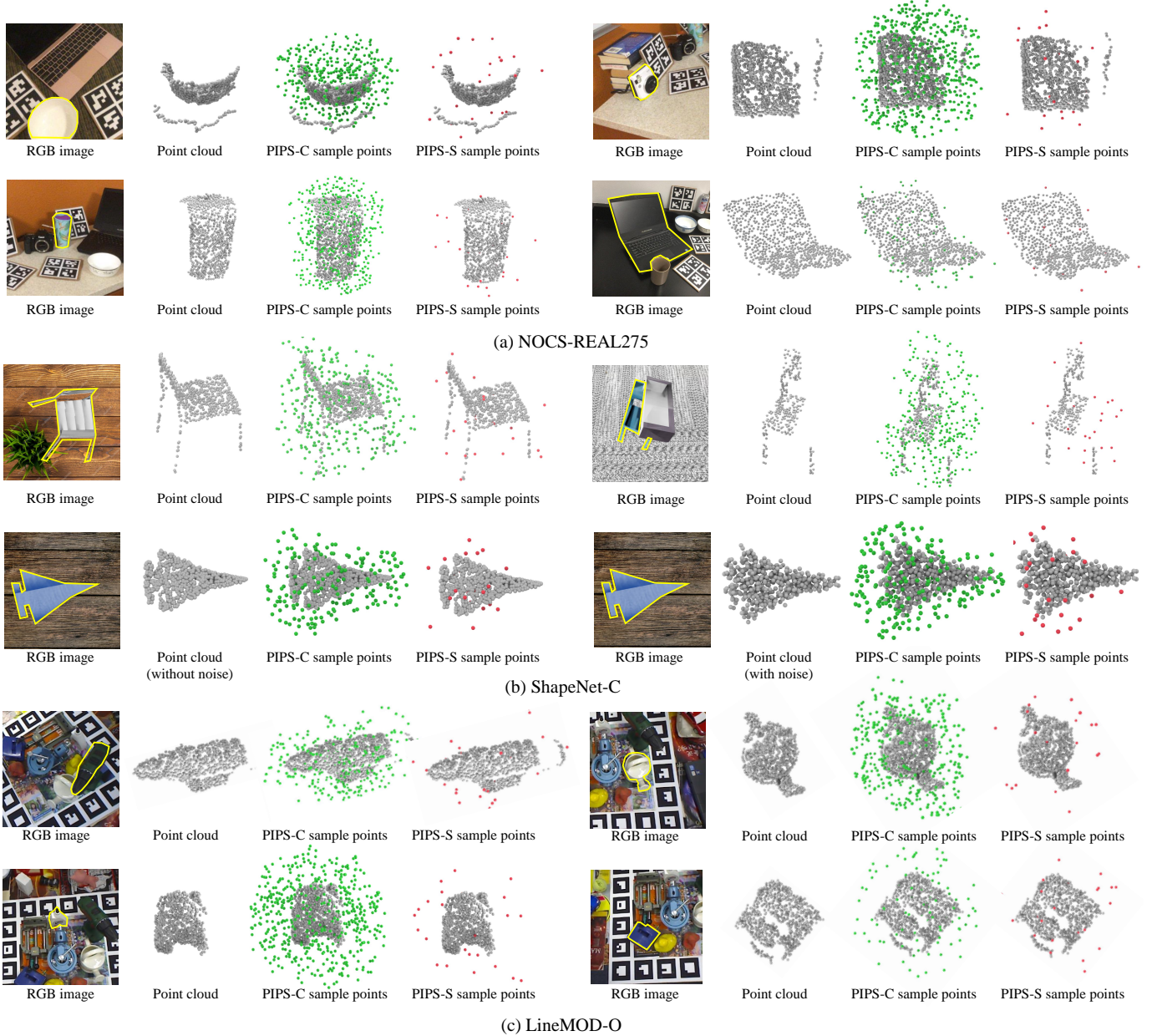


Fig. 13: Visualization of the PIPS sample points. The results show that: 1) PIPS-C sample points are dense, distributed around the object surface, possibly around the unseen surface region, such as the occluded chair in (b); 2) instead of uniformly distributing around the object surface, the sampling rule varies on different shapes, such as the laptop in (a); 3) PIPS-S sample points are sparse, covering the crucial regions of the input shape; 4) adding noise to the input points will not substantially change the distribution of PIPS-S sample points, such as the two cases in last row of (b), where the sampled points are speared near the wings and tail of the airplane despite the noise. The results are produced with only the point cloud as input, without the RGB image features.

replace it with a scalar variance and train the network in Section 3.4.1 with the loss function by formulation 4. Consequently, the PIPS-S estimation component is also turned off. The degraded performance of this baseline validates our design.

**No SO(3)-equivariance.** The SO(3)-equivariant convolutional implicit network is one of the main contributions. We replace it with an existing non-equivariant network, i.e., 3D-GCN [38]. We see that this baseline is less capable on the ShapeNet-C dataset. We also found that this baseline is significantly inferior to our full method in the holdout pose subset, where our full method achieves 0.61 in  $5^\circ 5\text{cm}$  and the baseline of w/o SO(3)-

equivariance achieves 0.38 in  $5^\circ 5\text{cm}$ .

**No RGB feature.** Adding RGB features seems straightforward, which could enhance the overall performance. To evaluate its importance, we ablate the RGB features in our method and retrain the networks. The performance drop shows that the RGB features are crucial in our method.

**Parameter settings.** Several key parameters are crucial to our method. Here, we evaluate other possibilities to study the rationality of the parameter settings. The parameters include: 1) the voxel size  $h$  of the network output in the PIPS-C estimation component; 2) the target sparsity  $\rho$  in the training loss function



TABLE 5: Ablation study of our method on ShapeNet-C.

Methods	Median( $^\circ$ ) $\downarrow$	Median(cm) $\downarrow$	$5^\circ 5\text{cm}\uparrow$
w/o PIPS	5.92	2.78	0.51
w/o PIPS-C	5.34	2.61	0.53
w/o PIPS-S	4.23	2.49	0.57
w/o anisotropic variance	4.51	2.89	0.55
w/o SO(3)-equivariance	5.18	3.02	0.49
w/o RGB	3.48	2.67	0.59
Full method	<b>3.16</b>	<b>2.13</b>	<b>0.62</b>

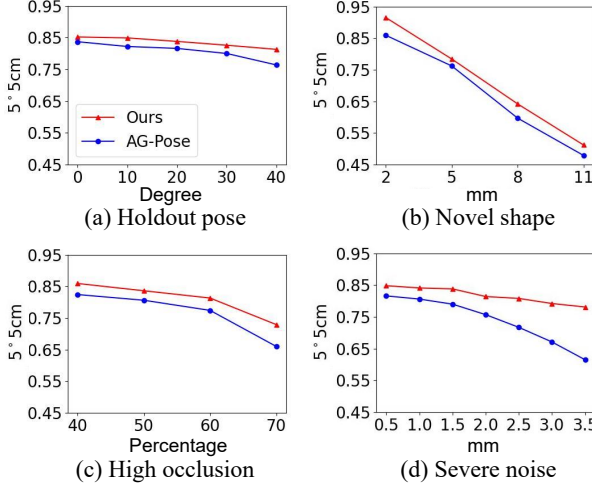


Fig. 14: Robustness evaluation. The quantitative comparisons of our method to a baseline method on the four challenging subsets of the NOCS-CAMERA25 dataset. Our method is more robust on unseen poses, novel shapes, high occlusion, and severe noise.

of the PIPS-S estimation component; 3) the threshold  $\omega$  to determine the positive samples in generating pseudo ground-truth for the PIPS estimation network. In general, the results show our method is not very sensitive to these parameter settings. Nevertheless, using a large  $h$  increases the training efforts and does not lead to significant improvements. Selecting a small  $\rho$  makes the generated sample points sparse, probably missing the detailed information. Adopting a small  $\omega$  incurs uncertain points that are unbeneficial to the network training.

#### 4.4 Robustness Evaluation

**Robustness in challenging scenarios.** To evaluate the robustness of our method, we conduct a pressure test on the NOCS-CAMERA25 dataset following a similar data division in Section 4.1. The quantitative comparisons to the baseline method AG-Pose [37] are shown in Figure 14. The plots show that our method is more stable and robust in scenarios with holdout pose, novel shape, high occlusion, and severe noise. There are several phenomena we can observe. First, our method achieves better performance on *holdout pose*, revealing the necessity of the SO(3)-equivariant convolutional implicit network. Second, the better performance of our method on *novel shape* demonstrates the advantage of balancing the exploitation and exploration of the proposed PIPS. Last, the results on *high occlusion* and *severe noise* imply the robustness of our method, thanks to the mechanism of sampling in the unobserved regions.

**Robustness to perturbation on sample points.** Another interesting problem is the method’s robustness under perturbations

TABLE 6: Effect of voxel size  $h$  in PIPS-C estimation.

Voxel size	Median( $^\circ$ ) $\downarrow$	Median(cm) $\downarrow$	$5^\circ \uparrow$	$5^\circ 5\text{cm}\uparrow$
4	5.42	3.15	0.56	0.49
8	<b>3.16</b>	2.13	<b>0.68</b>	<b>0.64</b>
16	3.28	<b>2.08</b>	<b>0.68</b>	0.62
32	3.71	2.34	0.63	0.58

TABLE 7: Effect of target sparsity  $\rho$  in PIPS-S estimation.

Target sparsity	Median( $^\circ$ ) $\downarrow$	Median(cm) $\downarrow$	$5^\circ \uparrow$	$5^\circ 5\text{cm}\uparrow$
0.3	3.42	2.65	0.63	0.56
0.2	3.28	2.38	<b>0.68</b>	0.61
0.1	<b>3.16</b>	<b>2.13</b>	0.66	<b>0.62</b>
0.05	3.71	2.84	0.59	0.52

TABLE 8: Effect of threshold  $\omega$  in positive sample selection of the PIPS estimation network.

Threshold	Median( $^\circ$ ) $\downarrow$	Median(cm) $\downarrow$	$5^\circ \uparrow$	$5^\circ 5\text{cm}\uparrow$
0.9	4.32	3.24	0.57	0.49
0.7	3.89	2.95	0.61	0.54
0.5	<b>3.16</b>	<b>2.13</b>	<b>0.66</b>	<b>0.62</b>
0.3	3.58	2.75	0.63	0.56

of the PIPS-S sample points. To investigate this problem, we randomly add Gaussian noise to the generated PIPS-S sample points during both the training and testing. We found the influence is not significant when the standard deviation of noise is less than 1 cm. Moreover, we conducted another experiment that randomly dropped out some PIPS-S sample points during the training of testing of the student model. Results show the performance is robust with about 30% of the sample points being turned off.

#### 4.5 Cross-task Generality of PIPS

The PIPS estimation network essentially learns to generate sample points whose features are informative and representative, allowing for a direction-aware quantification of prediction confidence for 3D coordinates. Consequently, the learned sampling strategy might be useful to other relevant tasks that involve per-point 3D coordinate regression and require uncertainty estimation for these outputs. To show the cross-task generality of PIPS, we apply the PIPS estimation network trained on category-level pose estimation (denoted as PIPSCAT) to two relevant tasks: instance-level pose estimation and shape reconstruction. Specifically, PIPSCAT is trained on the ShapeNet-C dataset.

**Generalizing to instance-level pose estimation.** We first apply PIPSCAT to the instance-level pose estimation task. To verify its generality, we train an alternative SO(3)-equivariant convolutional implicit networks to estimate object pose in the LineMOD-O dataset with the PIPS sample points generated by PIPSCAT. We then compare it with the original network trained on instance-level pose estimation (denoted as PIPSINS) as described in Section 4.2. Interestingly, we found PIPSCAT and PIPSINS produce similar distributions of sample points, resulting in comparable performances of the alternative method.

**Generalizing to shape reconstruction.** To validate the effects of PIPSCAT in the shape reconstruction task, we train two SO(3)-equivariant convolutional implicit networks to estimate the point-

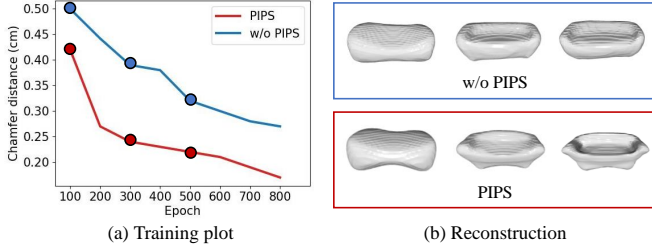


Fig. 15: (a) The training plot of the generalizing to shape reconstruction experiment. (b) Examples of the reconstructed shapes during the network training.

wise SDFs similar to DeepSDF [48]. The networks take single-view depth images as input and learn continuous signed distance functions to represent the complete shape. The object is then reconstructed by the Marching Cubes algorithm [42]. The two networks are trained with random sample points and the PIPS-C sample points by PIPSCAT, respectively. The training plots of the two networks are visualized in Figure 15a, which shows that PIPS-C sampled points lead to a faster error drop. The reconstructed shapes generated by the two methods are visualized in Figure 15b. The results indicate that the PIPS-C trained on pose estimation is indeed helpful to shape reconstruction, demonstrating the cross-task generality.

## 5 CONCLUSION AND DISCUSSION

In this work, we studied the problem of positive-incentive point sampling for neural implicit fields. To this end, we proposed a PIPS estimation network that generates sparse sample points that would gain sufficient information to determine all the DoFs of the object pose by training on them. Moreover, an SO(3)-equivariant convolutional implicit network that estimates point-level attributes with SO(3)-equivariance at any sample point is developed, outperforming most existing implicit neuron networks in pose estimation. Our method achieves state-of-the-art performance on three datasets. Several crucial conclusions can be drawn from this study. First, dense sampling is unnecessary for pose estimation with neural implicit fields. Second, positive-incentive point sampling can be estimated with a learning-based approach. Third, the learned sampling strategy is generalizable to other relevant tasks.

Our method estimates 3D anisotropic uncertainty from a pointset, making it applicable to tasks that involve per-point 3D coordinate regression and require uncertainty estimation for these outputs. Promising applications include: 1) Localization and Mapping: Our method can be used to select landmarks or points in a point cloud that provide the most reliable information for pose estimation, reducing drift and improving robustness in perceptually challenging environments; 2) Point Cloud Registration: Our method can prioritize point correspondences with high certainty and geometric stability, leading to faster convergence and higher accuracy in alignment algorithms.

Our method has the following limitations. First, the PIPS estimation network is trained with the pseudo ground-truth generated by a teacher model which requires additional training efforts. Integrating the two networks into a unified framework might make the method concise and further reduce the training costs. Second, our method cannot handle the problem of pose ambiguity caused by occlusions. An interesting direction is to opt for a diffusion

model-based network that could generate multiple outputs given an input point cloud. For future work, we expect to apply the proposed PIPS to Neural Radiance Fields [46] and 3D Gaussian Splatting [27], which might enable more efficient network training and improve the quality of synthesis images in large-scale scenes.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (62325211, 62132021, T2521006, 62302517), the Natural Science Foundation of Hunan Province (2023JJ20051), the Science and Technology Innovation Program of Hunan Province (2023RC3011), and the Cornerstone Foundation of NUDT (JS24-03).

## REFERENCES

- [1] Rohith Agaram, Shaurya Dewan, Rahul Sajani, Adrien Poulencard, Madhava Krishna, and Srinath Sridhar. Canonical fields: Self-supervised learning of pose-canonicalized neural fields. *arXiv preprint arXiv:2212.02493*, 2022.
- [2] Vincent Andrearczyk, Julien Fageot, Valentin Oreiller, Xavier Montet, and Adrien Depeursinge. Exploring local rotation invariance in 3d cnns with steerable filters. In *International Conference on Medical Imaging with Deep Learning*, pages 15–26. PMLR, 2019.
- [3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.
- [4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.
- [5] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part II 13*, pages 536–551. Springer, 2014.
- [6] Leonard Bruns and Patric Jensfelt. Sdfest: Categorical pose and shape estimation of objects from rgb-d using signed distance fields. *IEEE Robotics and Automation Letters*, 7(4):9597–9604, 2022.
- [7] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 608–625. Springer, 2020.
- [8] 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.
- [9] Yamei Chen, Yan Di, Guangyao Zhai, Fabian Manhardt, Chenyangguang Zhang, Ruida Zhang, Federico Tombari, Nassir Navab, and Benjamin Busam. Secondpose: Se(3)-consistent dual-stream feature fusion for category-level pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16–22, 2024*, pages 9959–9969. IEEE, 2024.
- [10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.
- [11] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulencard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.
- [12] 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.
- [13] John Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 3(1):2325–5870, 2007.

- [14] Natasha Gelfand, Leslie Ikemoto, Szymon Rusinkiewicz, and Marc Levoy. Geometrically stable sampling for the icp algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 260–267. IEEE, 2003.
- [15] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.
- [16] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.
- [17] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *European Conference on Computer Vision Workshops (ECCVW)*, 2020.
- [18] Lin Huang, Tomas Hodan, Lingni Ma, Linguang Zhang, Luan Tran, Christopher Twigg, Po-Chen Wu, Junsong Yuan, Cem Keskin, and Robert Wang. Neural correspondence field for object pose estimation. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 585–603. Springer, 2022.
- [19] Lin Huang, Tomas Hodan, Lingni Ma, Linguang Zhang, Luan Tran, Christopher D. Twigg, Po-Chen Wu, Junsong Yuan, Cem Keskin, and Robert Wang. Neural correspondence field for object pose estimation. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part X*, volume 13670 of *Lecture Notes in Computer Science*, pages 585–603. Springer, 2022.
- [20] Muhammad Zubair Irshad, Sergey Zakharov, Rares Ambrus, Thomas Kollar, Zsolt Kira, and Adrien Gaidon. Shapo: Implicit representations for multi-object shape, appearance, and pose optimization. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 275–292. Springer, 2022.
- [21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.
- [22] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.
- [23] Jiongchao Jin, Huanqiang Xu, and Biao Leng. Adaptive points sampling for implicit field reconstruction of industrial digital twin. *Sensors*, 22(17):6630, 2022.
- [24] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.
- [25] HyunJun Jung, Shun-Cheng Wu, Patrick Ruhkamp, Guangyao Zhai, Hannah Schieber, Giulia Rizzoli, Pengyuan Wang, Hongcheng Zhao, Lorenzo Garattoni, Daniel Roth, Sven Meier, Nassir Navab, and Benjamin Busam. Housecat6d - A large-scale multi-modal category level 6d object perception dataset with household objects in realistic scenarios. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 22498–22508. IEEE, 2024.
- [26] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.
- [27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [28] Shu Kong and Charles Fowlkes. Pixel-wise attentional gating for scene parsing. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1024–1033. IEEE, 2019.
- [29] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhöfer, and Markus Steinberger. Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. In *European Conference on Computer Vision*, pages 254–270. Springer, 2022.
- [30] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cospo: Consistent multi-view multi-object 6d pose estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII*, volume 12362 of *Lecture Notes in Computer Science*, pages 574–591. Springer, 2020.
- [31] Fu Li, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. *arXiv preprint arXiv:2203.04802*, 2022.
- [32] Guanglin Li, Yifeng Li, Zhichao Ye, Qihang Zhang, Tao Kong, Zhaopeng Cui, and Guofeng Zhang. Generative category-level shape and pose estimation with semantic primitives. *arXiv preprint arXiv:2210.01112*, 2022.
- [33] Ruilong Li, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: A general nerf acceleration toolbox. *arXiv preprint arXiv:2210.04847*, 2022.
- [34] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7677–7686. IEEE, 2019.
- [35] 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.
- [36] Jiehong Lin, Zewei Wei, Yabin Zhang, and Kui Jia. Vi-net: Boosting category-level 6d object pose estimation via learning decoupled rotations on the spherical representations. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 13955–13965. IEEE, 2023.
- [37] Xiao Lin, Wenfei Yang, Yuan Gao, and Tianzhu Zhang. Instance-adaptive and geometric-aware keypoint learning for category-level 6d object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 21040–21049. IEEE, 2024.
- [38] 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.
- [39] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14556–14565, 2021.
- [40] Xingyu Liu, Gu Wang, Yi Li, and Xiangyang Ji. CATRE: iterative point clouds alignment for category-level object pose refinement. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part II*, volume 13662 of *Lecture Notes in Computer Science*, pages 499–516. Springer, 2022.
- [41] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Gu Wang, Jiwen Tang, Zhigang Li, and Xiangyang Ji. GDRNPP: A geometry-guided and fully learning-based object pose estimator. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(7):5742–5759, 2025.
- [42] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.
- [43] Pavlo Melnyk, Michael Felsberg, and Marten Wadenback. Steerable 3d spherical neurons. In *International Conference on Machine Learning*, pages 15330–15339. PMLR, 2022.
- [44] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.
- [45] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019.
- [46] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [47] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [48] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.
- [49] Dario Pavlo, David Joseph Tan, Marie-Julie Rakotosaona, and Federico Tombari. Shape, pose, and appearance from a single image via bootstrapped radiance field inversion. *arXiv preprint arXiv:2211.11674*, 2022.
- [50] 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.
- [51] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020.
- [52] Wanli Peng, Jianhang Yan, Hongtao Wen, and Yi Sun. Self-supervised category-level 6d object pose estimation with deep implicit shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2082–2090, 2022.
- [53] Martin Píala and Ronald Clark. Terminerf: Ray termination prediction for efficient neural rendering. In *2021 International Conference on 3D Vision (3DV)*, pages 1106–1114. IEEE, 2021.
- [54] 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.
- [55] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.
- [56] Xuelin Qian, Li Wang, Yi Zhu, Li Zhang, Yanwei Fu, and Xiangyang Xue. Impdet: exploring implicit fields for 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4260–4270, 2023.
- [57] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.
- [58] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- [59] Yongzhi Su, Mahdi Saleh, Torben Fetzner, Jason R. Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to fine surface encoding for 6dof object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6728–6738. IEEE, 2022.
- [60] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, XiaoWei Zhou, and Noah Snavely. Neural 3d reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.
- [61] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021.
- [62] 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.
- [63] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.
- [64] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2320–2329, 2020.
- [65] Boyan Wan, Yifei Shi, and Kai Xu. SoCs: Semantically-aware object coordinate space for category-level 6d object pose estimation under large shape variations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [66] 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.
- [67] 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.
- [68] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.
- [69] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.
- [70] Yilin Wen, Xiangyu Li, Hao Pan, Lei Yang, Zheng Wang, Taku Komura, and Wenping Wang. Disp6d: Disentangled implicit shape and pose learning for scalable 6d pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 404–421. Springer, 2022.
- [71] Yilin Wen, Hao Pan, Lei Yang, and Wenping Wang. Edge enhanced implicit orientation learning with geometric prior for 6d pose estimation. *IEEE Robotics and Automation Letters*, 5(3):4931–4938, 2020.
- [72] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In *European Conference on Computer Vision*, pages 248–263. Springer, 2020.
- [73] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.
- [74] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.
- [75] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–383, 2021.
- [76] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.
- [77] Yanjie Ze and Xiaolong Wang. Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and A new dataset. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [78] Jiyao Zhang, Mingdong Wu, and Hao Dong. Genpose: Generative category-level object pose estimation via diffusion models. *CoRR*, abs/2306.10531, 2023.
- [79] Linfang Zheng, Chen Wang, Yinghan Sun, Esha Dasgupta, Hua Chen, Ales Leonardis, Wei Zhang, and Hyung Jin Chang. Hs-pose: Hybrid scope feature extraction for category-level object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 17163–17173. IEEE, 2023.



**Yifei Shi** is a Professor at the College of Intelligence Science and Technology, National University of Defense Technology (NUDT). He received his Ph.D. degree in computer science from NUDT in 2019. During 2017-2018, he was a visiting student research collaborator at Princeton University, advised by Thomas Funkhouser and Szymon Rusinkiewicz. His research interests mainly include computer vision, computer graphics, especially on object/scene analysis and manipulation by machine learning and geometric processing techniques. He has published 20+ papers in top-tier conferences and journals, including CVPR, ECCV, ICCV, SIGGRAPH Asia, IEEE Transactions on Pattern Analysis and Machine Intelligence, and ACM Transactions on Graphics.



**Boyan Wan** is pursuing his Ph.D. degree in College of Computer Science, NUDT. His research interests include pose estimation, 3D reconstruction and generative model. He has published several papers in international journals and conferences such as the International Conference on Computer Vision, IEEE TASLP. etc.





**Xin Xu** received the B.S. degree in electrical engineering from the Department of Automatic Control, NUDT, and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, NUDT. He has been a visiting scientist for cooperation research in the Hong Kong Polytechnic University, University of Alberta, and the University of Strathclyde, respectively. Currently, he is a full Professor and the director of the Department of Intelligent Science and Technology at NUDT. His main re-

search areas include: reinforcement learning and intelligent vehicles, learning control, robotics and machine learning. He has coauthored four books and published more than 150 papers in international journals and conferences. He is an associate editor of Information Sciences, CAAI Transactions on Intelligence Technology, Acta Automatica Sinica, Intelligent Automation and Soft Computing. He has also been a Guest Editor of IEEE Transactions on System, Man and Cybernetics: Systems, International Journal of Adaptive Control and Signal Processing. Prof. Xu is one of the recipients received the 2nd class National Natural Science Award of China in 2012, the 1st class Natural Science Award from Hunan Province, P. R. China, in 2009 and the Fork Ying Tong Youth Teacher Fund of China in 2008. He is a Senior Member of IEEE, a Committee Member of the IEEE Technical Committee on Approximate Dynamic Programming and Reinforcement Learning and the IEEE Technical Committee on Robot Learning. He has served as a PC member or Session Chair in many international conferences.



**Min Liu** received the B.Eng. degree from Wuhan University, Wuhan, China, in 2013, and the M.Sc. and Ph.D. degrees from National University of Defense Technology, Changsha, China, in 2015 and 2020, respectively. He was a joint Ph.D. student with University of Maryland, College Park, USA, from 2018 to 2020. He is currently an Associate Professor with the Academy of Military Science, Beijing, China. His research interests include Intelligent Perception, Robotics, and Gaming Systems.



**Kai Xu** is a Professor at the College of Computer, NUDT, where he received his Ph.D. in 2011. He conducted visiting research at Simon Fraser University and Princeton University. His research interests include geometric modeling and shape analysis, especially on data-driven approaches to the problems in those directions, as well as 3D vision and its robotic applications. He has published over 80 research papers, including 20+ SIGGRAPH/TOG papers. He has co-organized several SIGGRAPH Asia courses

and Eurographics STAR tutorials. He serves on the editorial board of ACM Transactions on Graphics, Computer Graphics Forum, Computers & Graphics, and The Visual Computer. He also served as program co-chair of CAD/Graphics 2017, ICVRV 2017 and ISVC 2018, as well as PC member for several prestigious conferences including SIGGRAPH, SIGGRAPH Asia, Eurographics, SGP, PG, etc. His research work can be found in his personal website: [www.kevinkaixu.net](http://www.kevinkaixu.net).