IST-Net: Prior-free Category-level Pose Estimation with Implicit Space Transformation

Jianhui Liu, Yukang Chen, Xiaoqing Ye, Xiaojuan Qi
1The University of Hong Kong, 2The Chinese University of Hong Kong, 3Baidu
jhliu0212@gmail.com, yukangchen@cse.cuhk.edu.hk, yexiaoqing@baidu.com, xjqi@eee.hku.hk

Abstract

Category-level 6D pose estimation aims to predict the poses and sizes of unseen objects from a specific category. Thanks to prior deformation, which explicitly adapts a category-specific 3D prior (i.e., a 3D template) to a given object instance, prior-based methods attained great success and have become a major research stream. However, obtaining category-specific priors requires collecting a large amount of 3D models, which is labor-consuming and often not accessible in practice. This motivates us to investigate whether priors are necessary to make prior-based methods effective. Our empirical study shows that the 3D prior itself is not the credit to the high performance. The keypoint actually is the explicit deformation process, which aligns camera and world coordinates supervised by world-space 3D models (also called canonical space). Inspired by these observations, we introduce a simple prior-free implicit space transformation network, namely IST-Net, to transform camera-space features to world-space counterparts and build correspondences between them in an implicit manner without relying on 3D priors. Besides, we design camera- and world-space enhancers to enrich the features with pose-sensitive information and geometrical constraints, respectively. Albeit simple, IST-Net achieves state-of-the-art performance based on prior-free design, with top inference speed on 3D75 with the best inference speed.

Figure 1. Comparison with competitive methods on REAL275 dataset. DPN refers to the DualPoseNet [26]. All speeds are measured on a single RTX3090Ti GPU. We use blue/red to distinguish prior-based/free methods. IST-Net achieves top performance on 3D75 with the best inference speed.

1. Introduction

Category-level pose estimation [39] draws great attention and plays an important role in practical applications, including robotic manipulation [6, 28, 22], augmented reality [34], and scene understanding [29, 43, 19, 20, 18, 14, 46]. Unlike instance-level pose estimation [15, 17, 23, 30, 13, 42], which requires a 3D CAD model for each object instance, this task aims at exploiting category-specific information and thus can further generalize to unseen objects within given categories.

Recently, many methods [25, 3, 35, 40, 4, 2, 5] have been proposed for category-level pose estimation, which can be categorized into two groups: prior-free methods and prior-based methods. Prior-free methods [4, 26, 39, 38] mainly focus on designing network structures to fit the training data better. These methods are relatively simple but struggle to generalize to novel objects and suffer from poor performance.

To address this issue, prior-based methods [25, 3, 35, 7, 21, 10] leverage category-specific 3D priors (templates) to guide pose estimation. They adopt a prior-driven deformation module [35] to deform the prior for synthesizing the target object in world-space. And then, they formulate the pose estimation problem as a camera- and world-space correspondence learning problem which explicitly aligns the coordinates [35]. Although considerable progress has been attained with prior-based methods [3, 45, 40], the requirements of collecting a large amount of ground-truth 3D models of target objects for obtaining the 3D prior and super-
vising training the prior deformation module hinders their practical applicability.

This motivates us to investigate the mechanism that makes prior-based methods effective. We experiment with the shape deformation module which is used to deform the given shape prior to the desired instance (Fig. 3) by replacing the shape priors with random noise and fixed shape prior from another category (see Fig. 2). We observe that the deformation module can adapt any inputs (noise or fixed prior) into a target world-space object (see Fig. 2 (b)). Besides, the model performance remains high regardless of the 3D priors (see Fig. 2 (a)). The above suggests: the shape prior itself is not necessary for the high performance of prior-based methods, but the deformation module that learns to synthesize world-space target objects and explicitly builds the correspondence between camera- and world-space is the key as the performance degrades dramatically without prior deformation. This promotes us to investigate new ways to build camera-to-world correspondence without requiring 3D priors and models.

In this paper, we propose a simple yet effective prior-free model, named Implicit Space Transformation Network (IST-Net), which implicitly sets up feature correspondence between camera-space and world-space without requiring 3D priors or ground-truth 3D models of target objects. Specifically, given the camera-space features, the network transforms them into world-space features which together with the camera-space features are further used for estimating camera poses. For learning the transformation, we propose a world-space enhancer that distills standard world-space features to supervise the transformed features. Note that the standard world-space features are obtained by transforming the input target object into its world-space with the ground-truth pose and feeding them into a feature extractor. Besides, given camera-space inputs, the backbone network’s feature extraction capabilities are boosted by introducing an auxiliary pose estimation loss, namely the camera-space enhancer. Notably, both enhancers are training only, which brings considerable performance improvements without introducing computational overhead.

Our main contributions are summarized as follows:

- We investigate prior-based methods and find that shape priors are not necessary for obtaining high performance while building the camera and world-space correspondence with prior deformation is a key factor.

- We propose a simple yet effective Implicit Space Transformation Network (IST-Net) that implicitly builds the correspondence between camera- and world-space on feature-level without requiring 3D models or priors.

- We introduce two different space enhancers to facilitate learning the transformation and enhance their representation capability for pose estimation.

- We conduct a series of experiments on REAL275 [39] and Wild6D [10] datasets to demonstrate the effectiveness of the proposed method. Notably, IST-Net is currently the only prior-free method that achieves state-of-the-art performance on the REAL275 benchmark and attains notable gains over the prior-based method in terms of efficiency and accuracy (see Fig. 1).

2. Related Works

2.1. Prior-free Methods

Prior-free methods focus on designing architectures for predicting the object pose in a concise manner. Sahin et al. [33] propose an “Intrinsic Structure Adaptor” to adapt the distribution shifts arising from shape discrepancies. Wang et al. [39] first introduce a new category-level benchmark by normalizing all object instances into a shared canonical representation named Normalized Object Coordinate Space-(NOCS) and try to recover the angle of view in NOCS for pose estimation. Chen et al. [2] introduce a learned canonical shape space to handle intra-class variation. Chen et al. [5] attempt to synthesize image matches upon neural rendering in order to verify the probability of each possible pose candidate for pose estimation. Wang et al. [37] propose 6D-PACK which learns to compactly represent an object by a handful of 3D key points based on the motion information and compute the pose by tracking. In pursuit of more efficient and direct pose estimation, a few methods [4, 7, 26] work on designing the network in an end-to-end manner. Chen et al. [4] decouple the rotation into two mutually orthogonal vectors to fully decode the orientation information which allows the network to naturally handle the circle symmetry object. Di et al. [7] embody the geometric insights with bounding box projection to enhance the learning of category-level pose-sensitive features. Lin et al. [26] introduce DualPoseNet which is composed of two parallel pose decoders on top of a shared pose encoder. The two decoders work in an implicit and explicit manner with the restriction of the predicted pose consistency.

2.2. Prior-based Methods

Since the severe intra-class variation, the generalization of the prior-free models is greatly suppressed. To alleviate this issue, some literature [10, 35, 25, 44, 41] turn to focus on prior-based methods. Tian et al. [35] present a general solution. They set up shape prior for each category upon an autoencoder and then use these priors as the standard template to reconstruct the canonical model for each instance. Chen et al. [2] use a variational autoencoder (VAE) [16] for reconstructing standard object shape, followed by a fully sparse convolution network for pose regression. Wang et
al. [40] propose a cascaded relation network to capture the underlying relations of multi-source inputs. Kai et al. [3] utilize a transformer network [8] to model the global structure similarity prior and target object, based on which the object semantic information is injected into the prior feature to dynamically adapt the category-level prior to each particular object. Fan et al. [9] adopt a shape prior guided reconstruction network and a discriminator network to learn high-quality canonical representations. Zhang et al. [45] use the shape priors as the indicator to predict pose and zero-mean residual vectors which encapsulate the spatial cues of the pose and enable geometry-guided consistency terms. Zhang et al. [44] learn dense correspondences between input images and the canonical shape prior via surface embedding. Lin et al. [25] establish deep correspondence in the feature space between shape prior and canonical model, which yields a surprising performance boost.

3. Analysis of Shape Priors

3.1. Preliminary

To overcome intra-class variation, the prior deformation, as a practical module, has been widely adopted by recent works [9, 3, 25]. The vanilla version of prior deformation can be divided into two parts: 1) generating shape priors and 2) leveraging shape priors to develop prior deformation techniques.

For the former one, the common solution is to train an autoencoder with various object models sampled from ShapeNet [1], then acquire the category-level shape embedding by averaging the latent vectors output by the encoder. These shape embeddings will be fed into the decoder to get the shape priors. It is worth mentioning that this process needs to rely on a large number of 3D models to obtain a general prior of a category.

For the latter one, we use Fig. 3 to illustrate the process. Given the shape prior, image patch, and observed points, the network first learns a deformation field that deforms the shape prior to the desired object instance, which is supervised by a ground-truth 3D model. Furthermore, the network outputs a matching matrix that indicates the point-to-point correspondences between the observed target points and the reconstructed models. These correspondences transform the models to the viewpoint in the world coordinate system. With the information from camera-space (depth images) and world-space (matched priors), pose parameters can be easily solved via Umeyama algorithm [36] or pose regression by neural networks.

3.2. Is Shape Prior Necessary?

We conduct extensive experiments to verify whether the shape prior is necessary to address the intra-class variation problem. Specifically, we choose a competitive candidate from prior-based methods, DPDN [25]. (For other methods please refer to the Appendix). We set up the following settings:

- **Case-1**: Official baseline.
- **Case-2**: All the categories share the same prior (can) in replace of the class-specific priors.
• **Case-3:** Using random noise restricted to the unit cube instead of standard shape priors.

• **Case-4:** Removing the prior deformation from original framework.

From Fig. 2, we can conclude that category-specific priors are not necessary as the model can learn to deform a shape to match the target from even random noise. The reason stems from the explicit supervision of ground-truth 3D models during training, which enables the model to learn to deform any given prior (e.g., random noise) to reconstruct a target object. Nevertheless, the prior deformation module contributes to the prior-based model as the performance drops dramatically without this module (see Fig. 2 (a) w/o deformation). The above suggests that the key to the success of prior-based methods is the deformation module that aligns objects in the camera and world-space and facilitates building correspondences, but not the prior itself.

Hence, we explore new ways to transform camera space inputs to world-space and implicitly builds the correspondence between them without relying on priors or 3D ground-truth models in training. We present a prior-free implicit feature transformation network (IST-Net) with details unfolded in Sec. 4.2. On one hand, our method gets rid of the dependence on a large number of 3D models required by prior-based methods. On the other hand, we make the model aware of the information in world-space with a simple design which further allows us to develop an efficient prior-free pose estimator without sacrificing model performance.

### 4. Method

Before delving into the details, we first show some mathematical notations in the category-level 6D pose estimation task for clarity. \( P_o \in \mathbb{R}^{N_o \times 3} \) and \( I_o \in \mathbb{R}^{h \times w \times 3} \) refer to the observed point cloud and corresponding RGB image, where \((h, w)\) denotes the size of the image and \(N_o\) is the number of object points. Given these inputs, the objective is to estimate the pose of the input instance, including rotation \( R \in SO(3) \), translation \( t \in \mathbb{R}^3 \) and size \( s \in \mathbb{R}^3 \).

#### 4.1. Overview

In this section, we will introduce the detailed architecture of the proposed IST-Net, as shown in Fig. 4. Given camera-space RGB image \( I_o \) and point cloud \( P_o \) as inputs, a CNN backbone network, and PointNet++ are used to extract image feature \( F_{I_o} \) and point feature \( F_{P_o} \) and \( F_r \) respectively. Then, the learned features are fed into the implicit space transformation module (Sec. 4.2) which performs implicit camera-space to world-space feature \( F_{Q_o} \) transformation. We also provide a detailed analysis of our implicit design and compare it with its explicit counterparts in Sec. 4.2.

Further, given point cloud \( P_o \), point-wise aligned world-space feature \( F_{Q_o} \) and camera-space feature \( F_{I_o} \) and \( F_{P_o} \), the pose estimator directly regress the poses \( \{ R, t, s \} \).

We adopt the pose estimator in DPDN [25]. Details are included in the Appendix. Finally, we introduce two auxiliary modules, namely camera-space enhancer and world-space enhancer (Sec. 4.3 and Sec. 4.4), to boost feature representations and facilitate learning the implicit transformation network.

### 4.2. Implicit Space Transformation

Since we have shown in Sec. 3.2, shape prior itself is not necessary, the important factor is how to transform camera-space inputs into world-space, align them and build their correspondences. To address these issues, we propose an implicit space transformation module, which transforms camera-space features to world-space in an implicit manner without resorting to ground-truth 3D models during training.

As depicted in Fig. 4, given an RGB image \( I_o \) and corresponding object point cloud \( P_o \), we first use the feature extractor to acquire semantic feature which sampled as \( F_{I_o} \in \mathbb{R}^{N_o \times d} \) and geometry feature \( F_{P_o} \in \mathbb{R}^{N_o \times d} \). Notably, before entering PointNet++, the point cloud will be pre-processed, which means subtracting the average value of all coordinates. In other words, the feature extractor can focus on the relative geometric relationship, avoiding the interference of spatial locations. However, this will make the network insensitive to location information, which is indispensable for the estimation of the translation and rotation. Hence, we utilize a Multi-Layer Perceptron (MLP) to encode the accurate position information into latent space as \( F_r \). Then we concatenate three of them as the input of the implicit space transformation module. Sequentially, an MLP is employed to fuse these input features followed by a global average pooling layer. Then the local and global features are concatenated and fed into an MLP to get the world-space feature \( F_{Q_o} \). This process can be formulated as follows:

\[
F_L = \text{MLP} ([F_{P_o}, F_{I_o}, F_r]), \\
F_G = G (F_L), \\
F_{Q_o} = \text{MLP} ([F_L, F_G]),
\]

where \([\ldots]\) refers to concatenation and \(G\) denotes the global average pooling. \(F_L\) and \(F_G\) refers to the local and global feature.

It should be noted that the implicit transformation network may not certainly transform the camera space feature to world space without meaningful supervision. We thus introduce a reconstruction-based loss \(L_{rec}\) to regularize the learning. Specifically, given \(F_{Q_o}\) as inputs, we use another MLP to predict reconstruct per-point coordinate in the
world-space \( \tilde{Q}_o \) and supervise it using ground-truth world-space coordinate \( Q_o \). Note that \( Q_o \) can be obtained by transforming input point cloud \( P_o \) using ground-truth poses without resorting to 3D models. Following [35], we adopt smooth-L1 loss as:

\[
L_{\text{rec}} = L_{\text{SL1}} \left( \tilde{Q}_o, Q_o \right), Q_o = \Gamma(P_o, R, t, s),
\]

where \( L_{\text{SL1}} \) denotes the Smooth-L1 loss [27] and \( \Gamma \) indicates the 3D geometric transformation operation according to pose. This supervision will encourage the transformed features to be in the world-space.

**Explicit Counterpart** To better showcase the merits of our implicit space transformation, we set up a comparison with its explicit counterpart. As shown in Fig. 5, we firstly use an extra pose estimator to predict a group of pose parameters and then use them to explicitly transform the camera coordinate to the world coordinate as \( \tilde{Q}_o \). And then we utilize another PointNet++ to extract world-space features \( F_{\tilde{Q}_o} \). Finally, we feed \( F_{\tilde{Q}_o}, \tilde{Q}_o \), and camera space items into the estimator for pose regression. Explicit transformation is more intuitive but struggles in model redundancy. In contrast, our method is quite concise during inference, reaching a better balance between efficiency and performance. More analysis are presented in Sec. 5.5.

**4.3. Camera-space Enhancer**

Since the transformed feature \( F_{\tilde{Q}_o} \) is derived from camera-space features, the representative power of \( F_{P_0} \) and \( F_{L_0} \) becomes crucial. Therefore, we propose a camera-space enhancement strategy to strengthen the camera-space features supervised by the auxiliary pose estimation task. The camera-space enhancer can easily be an auxiliary pose estimator whose training doesn’t rely on the world-space 3D model. This encourages the feature extractor to learn representations that bridge camera and world-space and implicitly establish a correspondence with world-space features, which is helpful for subsequent transformations and final predictions.

**4.4. World-space Enhancer**

In the implicit space transformation module, we predict the world-space coordinates, which encourage the network to transform camera-space features into world-space feature \( F_{\tilde{Q}_o} \), and regularize the learning indirectly. Here, we propose world-space enhancer which directly enforces \( F_{\tilde{Q}_o} \) to be similar to world-space counterparts.

As shown in Fig. 4, the world-space enhancer takes the world-space point cloud \( Q_o \) derived using Eq. (2) as input and produces (ground-truth) world-space features \( F_{\tilde{Q}_o} \). \( F_{Q_o} \) is used to directly supervise \( F_{\tilde{Q}_o} \) as

\[
L_{\text{feat}} = L_{\text{MSE}} \left( F_{\tilde{Q}_o}, F_{Q_o} \right),
\]

\[
L_{\text{MSE}} \left( F_{\tilde{Q}_o}, F_{Q_o} \right) = \frac{1}{N_o \cdot d} \sum_{i=1}^{N_o} \sum_{j=1}^{d} \left| f_{\tilde{Q}_o}^{ij} - f_{Q_o}^{ij} \right|^2.
\]

To train the feature extractor (i.e. PointNet++) and obtain high-quality features \( F_{\tilde{Q}_o} \), we use another auxiliary pose estimator with the same architecture as the one that generates
pose estimates to produce supervisory signals. Notably, the only difference from the main pose estimator (red boxes in Fig. 4) is the world-space inputs, fully encoded ground-truth world-space coordinates \( Q_o \). In this way, we believe that \( F_{Q_o} \) shares the same feature space with \( F_{Q_{\text{pred}}} \) but is more accurate, compact, and friendly for pose estimation.

In conclusion, this module generates a standard latent feature that further provides restriction from a high-level aspect, regularizing \( F_{Q_o} \) with implicit geometric information. In conjunction with constraint \( L_{\text{rec}} \) in Sec. 4.2, the space transformation is guided to learn to transform camera-space features to world-space.

\subsection{4.5. Overall Loss Function}

In summary, the overall loss function is as below:

\[
L = L_{\text{main}} + L_{\text{aux1}} + L_{\text{aux2}} + \lambda_f L_{\text{feat}} + \lambda_t L_{\text{rec}},
\]

where \( \lambda_f \) and \( \lambda_t \) are hyper-parameters which used to balance the individual loss contributions. \( L_{\text{main}} \), \( L_{\text{aux1}} \), and \( L_{\text{aux2}} \) refer to the supervision for the outputs of main pose estimator and two feature enhancers which share the same loss format as:

\[
L_{\text{pose}} = \| R_{\text{pred}} - R_{\text{gt}} \|_2 + \| t_{\text{pred}} - t_{\text{gt}} \|_2 + \| s_{\text{pred}} - s_{\text{gt}} \|_2,
\]

where subscripts “pred” and “gt” denote the predicted and ground truth pose parameters, respectively.

\section{5. Experiments}

\subsection{5.1. Datasets}

\textbf{REAL275 & CAMERA25:} Our model is trained on both the virtual dataset, CAMERA25, and the real dataset, REAL275 [39], and conducted an evaluation on REAL275 test split. CAMERA25 contains 300k synthetic RGB-G images, which are generated by rendering 1,085 synthetic objects with real-world backgrounds. REAL275 includes 8k RGB-D images, where 4300 images are split for training, 950 images for validation, and 2750 images for testing. In both datasets, there are 6 categories, including bottle, bowl, camera, can, laptop, and mug.

\textbf{Wild6D:} Wild6D [10] contains 5,166 videos with 1722 object instances and 5 categories (bottle, bowl, camera, can, laptop, and mug). Among this data, 486 videos are split for model evaluation.

\subsection{5.2. Implementation Details}

Following previous work [25], we use the off-the-shelf MaskRCNN [11] with a backbone of ResNet101 [12] for generating high-quality instance masks. We adopt a PSP Network [47] based on ResNet-18 [12] for semantic feature extraction and PointNet++ [32] for point-level feature extraction. The number of object point \( N_o \) is set as 1024 and the size of the RGB image is resized to \( 192 \times 192 \). We adopt several commonly used data augmentation, including random uniform noise, random rotational and translational perturbations, and bounding box-based adjustment, which is proposed by FS-Net [4]. For the issue of symmetry, we follow previous work [31], mapping ambiguous rotations in training images with the same object appearance to a canonical rotation. Given an input image, the network predicts the canonical rotation, which is the best that can be done in the presence of \( e \) of symmetries. The hyper-parameters of the loss weights \( \lambda_f \) and \( \lambda_t \) are set to 10 and 1, respectively. All the experiments are conducted on 2 RTX3090Ti GPUs with a batch size of 24, and the ratio of real data to synthetic data is 1:3. For a fair comparison, the total training epoch is fixed to 30 epochs, and all the modules are trained in an end-

<table>
<thead>
<tr>
<th>Method</th>
<th>Prior</th>
<th>3D_{25}</th>
<th>3D_{50}</th>
<th>3D_{75}</th>
<th>5^22cm</th>
<th>5^5cm</th>
<th>10^2cm</th>
<th>10^5cm</th>
<th>10^10cm</th>
<th>Speed (FPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSC [39]</td>
<td>x</td>
<td>84.9</td>
<td>80.5</td>
<td>30.1</td>
<td>7.2</td>
<td>10</td>
<td>13.8</td>
<td>25.2</td>
<td>26.7</td>
<td>4.8</td>
</tr>
<tr>
<td>CASS [2]</td>
<td>x</td>
<td>84.2</td>
<td>77.7</td>
<td>-</td>
<td>-</td>
<td>23.5</td>
<td>-</td>
<td>58.0</td>
<td>58.3</td>
<td>-</td>
</tr>
<tr>
<td>FS-Net [4]</td>
<td>x</td>
<td>84.2</td>
<td>77.7</td>
<td>-</td>
<td>-</td>
<td>28.2</td>
<td>-</td>
<td>60.8</td>
<td>-</td>
<td>20.8</td>
</tr>
<tr>
<td>DualPoseNet [26]</td>
<td>x</td>
<td>-</td>
<td>79.8</td>
<td>62.2</td>
<td>29.3</td>
<td>35.9</td>
<td>66.8</td>
<td>-</td>
<td>-</td>
<td>2.3</td>
</tr>
<tr>
<td>GPV-Pose [7]</td>
<td>x</td>
<td>84.2</td>
<td>83.0</td>
<td>64.4</td>
<td>32.0</td>
<td>42.9</td>
<td>-</td>
<td>73.3</td>
<td>74.6</td>
<td>23.6</td>
</tr>
<tr>
<td>SPD [35]</td>
<td>✓</td>
<td>83.4</td>
<td>77.3</td>
<td>53.2</td>
<td>19.3</td>
<td>21.4</td>
<td>43.2</td>
<td>54.1</td>
<td>-</td>
<td>6.3</td>
</tr>
<tr>
<td>CR-Net [40]</td>
<td>✓</td>
<td>-</td>
<td>79.3</td>
<td>55.9</td>
<td>27.8</td>
<td>34.3</td>
<td>47.2</td>
<td>60.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SGPA [3]</td>
<td>✓</td>
<td>-</td>
<td>80.1</td>
<td>61.9</td>
<td>35.9</td>
<td>39.6</td>
<td>61.3</td>
<td>70.7</td>
<td>-</td>
<td>5.2</td>
</tr>
<tr>
<td>DO-Net [24]</td>
<td>✓</td>
<td>80.4</td>
<td>63.7</td>
<td>-</td>
<td>24.1</td>
<td>34.8</td>
<td>45.3</td>
<td>67.4</td>
<td>-</td>
<td>11.3</td>
</tr>
<tr>
<td>RBP-Pose [45]</td>
<td>✓</td>
<td>-</td>
<td>67.8</td>
<td>-</td>
<td>38.2</td>
<td>48.1</td>
<td>63.1</td>
<td>79.2</td>
<td>-</td>
<td>26.8</td>
</tr>
<tr>
<td>DPN [25]</td>
<td>✓</td>
<td>84.3</td>
<td>83.4</td>
<td>76.0</td>
<td>46.0</td>
<td>50.7</td>
<td>70.4</td>
<td>78.4</td>
<td>80.4</td>
<td>25.2</td>
</tr>
<tr>
<td>IST-Net (Ours)</td>
<td>x</td>
<td>84.3</td>
<td>82.5</td>
<td>76.6</td>
<td>47.5</td>
<td>53.4</td>
<td>72.1</td>
<td>80.5</td>
<td>82.6</td>
<td>34.6</td>
</tr>
</tbody>
</table>

Table 1. Comparisons with state-of-art methods on REAL275 dataset. We summarize the pose estimation results reported in the original papers. Prior refers to whether the method builds upon shape priors. '-' denotes no results are reported under this metric. All the experiments for inference speed are conducted with a single RTX3090Ti GPU.
to-end manner. During inference, only the feature extractor and implicit space transformation module are preserved.

### 5.3. Evaluation Metrics

We follow [39, 7] and utilize the widely adopted metrics for evaluation, including the mean precision of 3D intersection over union (IoU) to jointly evaluate rotation, translation, and size. Besides, the 5°2cm, 5°5cm, 10°2cm, and 10°5cm are used to evaluate the rotation and translation error directly, specifically, only the prediction error under both thresholds can be considered correct.

### 5.4. Comparison with State of the Arts

We present the results with IST-Net with state-of-the-art methods on REAL275 [39], as shown in Tab. 1. For comparison with prior-free methods, we surpass others with a large gap on all evaluation metrics, e.g., we reach 47.5 and 76.6 on 5°2cm and D75, which outperform GPV-Pose by 19.4% and 41%. As for prior-based methods, compared with the current most powerful method DPDN [25], we still perform significant improvements in most of the metrics, e.g., 76.6 vs 76 on D75, 47.5 vs 46.0 on 5°2cm, 53.4 vs 50.7 on 5°5cm, 80.5 vs 78.4 on 10°5cm and 82.6 vs 80.4. Notably, it is the first time for the prior-free methods to achieve comparable or even higher performance compared with prior-based methods on REAL275. In addition, we present a per-class comparison between IST-Net and DPDN, as shown in Fig. 7. Notably, our method performs better on the prediction of rotation, the error curve is steeper on the geometrically complex object, e.g., camera, which clearly proves the effectiveness of our contributions. Apart from the performance, model efficiency is also worthy of attention, we list the inference time in the last column of Tab. 1. IST-Net reaches top inference speed which far exceeds other methods by more than 25% acceleration.

We conduct experiments on a larger dataset, Wild6D [10], to further verify the effectiveness of the proposed method. We directly test our model which are trained on REAL275 and CAMERA25 datasets without extra fine-tuning. The results are reported in Tab 2. It can be observed that IST-Net is much better than those designed for the REAL275 dataset and has an obvious improvement in various matrices. Compared with RePoNet [10] and Self-Pose [44] both of which are trained upon Wild6D, our method shows good generalization without fine-tuning on the target dataset. We attribute this to the fact that our model is designed to learn the correspondence between camera and world space. We achieve the highest performance of 93.4 and 79.6 on 3D25 and 3D50. As for other metrics, IST-Net can also reach a similar results to Self-Pose. These analysis and results verify the potential of our method.

### 5.5. Ablation Study

**Effects of Proposed Modules.** We ablate the combination of different modules of the proposed method, the results are shown in Tab 3. Firstly we present the effectiveness of the implicit space transformation module (IST). By adding this module, we can easily observe that the baseline is greatly lifted, suggesting that transforming camera-space features to world-space counterparts and building the correspondence between them in an implicit manner indeed can benefit the pose estimation. Besides this, after adding the camera-space enhancer (CE), the precision on 5°5cm increases from 48.5 to 52.5, the reason is that with this auxiliary module, the feature extractors are enriched with more pose-sensitive information, which is beneficial to the quality of feature transformation and to improve the accuracy of final pose estimation. In addition, we show the advantage of world-space enhancer (WE), by combining it with IST. The results (E4) show that WE can further extend the performance, especially on 10°2cm and 10°5cm, which indicates that high-level supervision provides additional information different from low-level constraint. Finally, by combining all modules, we reach relatively competitive performance.

**Effects of Position Encoding Term.** In this part, we verify the effect of the positional encoding (PE) term, the results are shown in Tab. 4. Obviously, without the PE module, the performance drops significantly on 5°2cm and 5°5cm. This
Table 5. Comparison with explicit space transformation. Param. refers to the number of parameters.

<table>
<thead>
<tr>
<th>Method</th>
<th>3D_{D0}</th>
<th>3D_{D75}</th>
<th>5°/2cm</th>
<th>5°/5cm</th>
<th>10°/2cm</th>
<th>10°/5cm</th>
<th>Param.</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>73.9</td>
<td>44.2</td>
<td>52.5</td>
<td>68.5</td>
<td>79.2</td>
<td>21M</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Explicit</td>
<td>75.1</td>
<td>45.0</td>
<td>50.7</td>
<td>69.6</td>
<td>80.0</td>
<td>24M</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Ablation on the predicted world coordinate. M is our matching-based variant, using the Umeyama algorithm for testing.

<table>
<thead>
<tr>
<th>Method</th>
<th>3D_{D0}</th>
<th>3D_{D75}</th>
<th>5°/2cm</th>
<th>5°/5cm</th>
<th>10°/2cm</th>
<th>10°/5cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPDN</td>
<td>78.8</td>
<td>48.6</td>
<td>11.1</td>
<td>13.1</td>
<td>31.6</td>
<td>45.3</td>
</tr>
<tr>
<td>IST-Net</td>
<td>81.0</td>
<td>53.9</td>
<td>25.5</td>
<td>29.7</td>
<td>44.9</td>
<td>53.9</td>
</tr>
</tbody>
</table>

Table 7. Performance with limited training data (REAL275 only).

<table>
<thead>
<tr>
<th>Method</th>
<th>3D_{D0}</th>
<th>3D_{D75}</th>
<th>5°/2cm</th>
<th>5°/5cm</th>
<th>10°/2cm</th>
<th>10°/5cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPDN</td>
<td>83.0 (0.4)</td>
<td>75.4 (1.6)</td>
<td>41.5 (3.7)</td>
<td>44.9 (3.8)</td>
<td>69.2 (1.2)</td>
<td>76.5 (3.9)</td>
</tr>
<tr>
<td>IST-Net</td>
<td>82.2 (0.2)</td>
<td>75.9 (1.0)</td>
<td>41.0 (2.2)</td>
<td>46.7 (5.2)</td>
<td>69.6 (1.3)</td>
<td>79.0 (4.9)</td>
</tr>
</tbody>
</table>

Table 8. Robustness test on REAL275 dataset with Gaussian noise.

<table>
<thead>
<tr>
<th>Unseen class</th>
<th>3D_{D0}</th>
<th>3D_{D75}</th>
<th>5°/2cm</th>
<th>5°/5cm</th>
<th>10°/2cm</th>
<th>10°/5cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowl</td>
<td>10.9 (100)</td>
<td>99.3 (1.2)</td>
<td>56.1 (0.0)</td>
<td>49.1 (0.0)</td>
<td>59.7 (0.0)</td>
<td>97.6 (0.0)</td>
</tr>
<tr>
<td>Laptop</td>
<td>17.7 (18.2)</td>
<td>0.4 (0.3)</td>
<td>0.0 (0.0)</td>
<td>0.0 (0.0)</td>
<td>0.0 (0.0)</td>
<td>0.0 (0.0)</td>
</tr>
</tbody>
</table>

Table 9. Results on unseen class. Color map: Black for IST-Net, Gray for DPDN (prior-based).

<table>
<thead>
<tr>
<th>λ</th>
<th>3D_{D0}</th>
<th>3D_{D75}</th>
<th>5°/2cm</th>
<th>5°/5cm</th>
<th>10°/2cm</th>
<th>10°/5cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.6</td>
<td>72.6</td>
<td>40.8</td>
<td>47.0</td>
<td>68.0</td>
<td>77.7</td>
</tr>
<tr>
<td>3</td>
<td>82.7</td>
<td>76.1</td>
<td>42.9</td>
<td>48.9</td>
<td>70.1</td>
<td>80.1</td>
</tr>
<tr>
<td>5</td>
<td>83.2</td>
<td>76.1</td>
<td>43.6</td>
<td>50.6</td>
<td>69.1</td>
<td>79.7</td>
</tr>
<tr>
<td>10</td>
<td>82.5</td>
<td>76.6</td>
<td>47.4</td>
<td>53.4</td>
<td>72.1</td>
<td>80.5</td>
</tr>
<tr>
<td>20</td>
<td>82.0</td>
<td>75.2</td>
<td>45.0</td>
<td>51.5</td>
<td>68.0</td>
<td>77.9</td>
</tr>
<tr>
<td>50</td>
<td>82.8</td>
<td>75.3</td>
<td>41.5</td>
<td>47.2</td>
<td>68.8</td>
<td>77.9</td>
</tr>
<tr>
<td>100</td>
<td>83.1</td>
<td>76.3</td>
<td>45.3</td>
<td>50.4</td>
<td>70.4</td>
<td>78.9</td>
</tr>
</tbody>
</table>

Table 10. Ablation on the loss weight λ_f.

in turn proves that the PE module makes up for the position lost by the feature extractor, which benefits pose regression.

Comparison with Explicit Space Transformation. To further verify the effectiveness of the proposed implicit space transformation, we set up an experiment with its explicit counterpart. For reaching a fair comparison, only WE and IST are included in the implicit candidate. From the results shown in Tab. 5, we can easily find the results of the two methods are very close. However, our method yields obvious superiority in speed (34Hz vs 22Hz) and parameter quantity (21M vs 24M) which attributes to the succinct feature space transformation instead of introducing repetitive modules for extracting features from coordinates. This further indicates the potential of the proposed modules.

Ablations on Predicted World Coordinate. Considering that we predict the coordinate of observed points in world-space for supervising implicit space transformation from a low-level perspective. Therefore, the quality of the gener-

Table 11. Ablation on the loss type of L_{feat}

<table>
<thead>
<tr>
<th>Type</th>
<th>3D_{D0}</th>
<th>3D_{D75}</th>
<th>5°/2cm</th>
<th>5°/5cm</th>
<th>10°/2cm</th>
<th>10°/5cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>82.5</td>
<td>76.6</td>
<td>47.5</td>
<td>53.4</td>
<td>72.1</td>
<td>80.5</td>
</tr>
<tr>
<td>L1</td>
<td>82.8</td>
<td>74.6</td>
<td>44.6</td>
<td>50.4</td>
<td>69.0</td>
<td>78.8</td>
</tr>
</tbody>
</table>

6. Qualitative Analysis

In Fig 6, we visually compare our methods and DPDN on the REAL275 dataset. It clearly shows the superiority of our method. As highlighted in blue boxes, DPDN...
7. Conclusion

In this paper, we analyze the overlooked issues in prior-based pose estimation methods and empirically find that shape prior does not contribute to performance boosts. The keypoint is actually the deformation process, which builds correspondence between camera and world coordinates by reconstructing the object shape in the world space. Inspired by this, we design an implicit space transformation network (IST-Net) to transform the camera-space features to world space in an implicit manner. It builds the space correspondence without requiring 3D priors or ground-truth 3D models of target objects. Besides, we design two independent feature enhancers to further enhance the features from both camera- and world-space, which enriches them with more pose-sensitive information and geometrical constraints. Extensive experiments on the challenging benchmark show the effectiveness of our method in both efficiency and accuracy. We hope our investigation can provide new insights for future research in the community.

8. Acknowledgements

This work has been supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant no. 17202422), and RGC matching fund scheme (RMGS). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.
References


Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. CoRR, abs/1910.10750, 2019. 2

Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose estimation using dual pose network with refined learning of pose consistency. CoRR, abs/2103.06526, 2021. 1, 2, 6, 7


Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2901–2910, 2019. 1


Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. CoRR, abs/1910.10750, 2019. 2


