PicoPose: Progressive Pixel-to-Pixel Correspondence Learning for Novel Object Pose Estimation

Liuhua Liu[1] **Jiehong Lin**[2]* **Zhenxin Liu**[1] **Kui Jia**[3]
[1]South China University of Technology [2]The University of Hong Kong
[3]The Chinese University of Hong Kong, Shenzhen

Abstract: RGB-based novel object pose estimation is critical for rapid deployment in robotic applications, yet zero-shot generalization remains a key challenge. In this paper, we introduce **PicoPose**, a novel framework designed to tackle this task using a three-stage pixel-to-pixel correspondence learning process. Firstly, PicoPose matches features from the RGB observation with those from rendered object templates, identifying the best-matched template and establishing coarse correspondences. Secondly, PicoPose smooths the correspondences by globally regressing a 2D affine transformation, including in-plane rotation, scale, and 2D translation, from the coarse correspondence map. Thirdly, PicoPose applies the affine transformation to the feature map of the best-matched template and learns correspondence offsets within local regions to achieve fine-grained correspondences. By progressively refining the correspondences, PicoPose significantly improves the accuracy of object poses computed via PnP/RANSAC. PicoPose achieves state-of-the-art performance on the seven core datasets of the BOP benchmark, demonstrating exceptional generalization to novel objects. Code and trained models are available at `https://github.com/foollh/PicoPose`.

Keywords: Novel Object Pose Estimation, Robotic Manipulation

1 Introduction

Object poses are typically represented by six degrees of freedom (DoFs) parameters, including 3D rotation and translation, to define the transformation from a canonical object space to the camera space. Estimating object poses is highly sought after in real-world applications, particularly in the context of robotics, where it enables precise manipulation, grasping, and interaction with various objects [1, 2, 3, 4, 5] (see Fig. 1), and is therefore extensively explored in research.

Early research [6, 7, 8, 9] primarily focused on pose estimation with the same object CAD models for both training and testing phases, but lacked flexibility for the objects unseen during training. Later studies [10, 11, 12, 13, 14] addressed unseen objects within known categories by defining a normalized object coordinate space, but struggled with novel categories. With the advancement of foundation models [15, 16, 17], recent research [18, 19, 20] has increasingly focused on handling new objects to achieve zero-shot pose estimation. While this capability enables rapid deployment of robotic systems, it presents significant generalization challenges that remain to be addressed.

For the zero-shot task of novel object pose estimation, recent methods using RGB-D images have achieved remarkable performance through techniques such as template matching with pose updating [18, 20] or point registration for pose computation [19]. The success of these approaches is largely attributed to the essential geometric support provided by depth maps, which supply crucial features for matching and offer geometric priors that enhance object localization in 3D space. However, the high cost of depth sensors often limits their practicality in real-world applications, making methods

*Corresponding author: mortimer.jh.lin@gmail.com.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

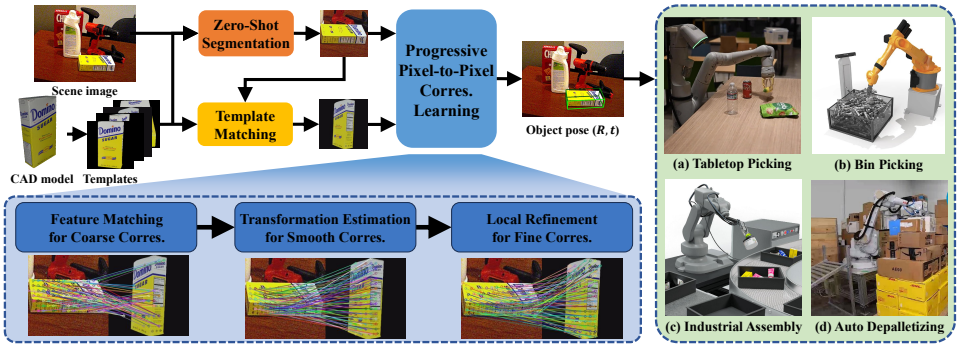

Figure 1: An overview of our proposed PicoPose with a three-stage pixel-to-pixel correspondence learning process for novel object pose estimation from RGB images. By progressively refining the correspondences, PicoPose significantly improves the accuracy of object poses computed via PnP/RANSAC. With zero-shot capability, PicoPose enables rapid deployment across various robotic manipulation systems for unseen objects.

based solely on RGB images a more appealing option. Despite this, RGB-only approaches remain underexplored and generally fail to achieve competitive performance. Representative methods like GigaPose [21] and FoundPose [22], which rely on establishing correspondences between observed scenes and rendered templates via simply feature matching, often suffer from noisy correspondences prone to outliers, leading to imprecise pose predictions.

To this end, we introduce a novel framework for progressive **pi**xel-to-pixel **co**rrespondence learning, termed as **PicoPose**, to enable precise pose estimation of novel objects from RGB images. As illustrated in Fig. 1, PicoPose progressively refines the correspondences between RGB observations and templates across three stages, significantly enhancing the accuracy of object poses. With zero-shot capability, PicoPose enables rapid deployment across various robotic manipulation systems.

The architecture of PicoPose is illustrated in Fig. 2. More specifically, given an RGB image of a cluttered scene and a CAD model of an object that was not seen during training, PicoPose begins by rendering object templates from various viewpoints of the CAD model. These templates are then used in conjunction with zero-shot segmentation techniques (e.g., CNOS [23]) to detect the target object within the RGB scene. PicoPose then uses a three-stage correspondence learning process to identify the best-matched template for the detected object and to learn fine-grained pixel-to-pixel correspondences between them. Since each pixel in the template corresponds to a 3D surface point on the CAD model, we establish pairs of 2D positions on the observation and the corresponding 3D points on the template, which are subsequently used to compute the 6D pose through PnP/RANSAC. For the process of correspondence learning, the three stages are described as follows:

- **Stage 1: Feature Matching for Coarse Correspondences.** In this stage, PicoPose utilizes visual transformers to capture features for matching the RGB observation and the rendered templates, identifying the best-matched one and obtaining coarse correspondences.
- **Stage 2: Global Transformation Estimation for Smooth Correspondences.** In this stage, PicoPose represents the coarse correspondences as a correspondence map, from which a 2D affine transformation, including in-plane rotation, scale, and 2D translation, is regressed to smooth the coarse correspondences and filter outliers.
- **Stage 3: Local Refinement for Fine Correspondences.** In this stage, PicoPose applies the affine transformation to the feature map of the best-matched template and employs several offset regression blocks to learn correspondence offsets within local regions.

We train PicoPose on the synthetic datasets of ShapeNet-Objects [24] and Google-Scanned-Objects [25], and test it on seven BOP datasets [26]. The quantitative results on these datasets outperform other existing methods, demonstrating the zero-shot capability of PicoPose. We also apply PicoPose in scenarios where object reference images are used to represent novel objects.

In this paper, our key contributions are: (1) the introduction of PicoPose, a novel framework that leverages progressive pixel-to-pixel correspondence learning for pose estimation of novel objects from RGB images; (2) the development of three designed stages of correspondence learning to improve pose accuracy within PicoPose; and (3) the achievement of state-of-the-art results on the seven core datasets of the BOP benchmark for the RGB-based task, especially without refinement.

2 Related Work

Methods Based on Image Matching. To address the challenge of generalization, some approaches [27, 28, 18, 21, 20, 29] simplify the task of novel object pose estimation by using an image matching strategy, which involves rendering object templates in various poses and then retrieving the best-matched template to determine the corresponding pose. This strategy is often followed by downstream refinements, as in MegaPose [18] and GenFlow [29]. In contrast, FoundationPose [20] first updates the poses of templates before selecting the best-matched one.

Methods Based on Pixel/Point Matching. This group of methods estimates object poses by establishing correspondences, including 2D-3D correspondences for RGB inputs and 3D-3D correspondences for RGB-D inputs. For instance, OnePose [30] matches the pixel descriptors in object proposals with the point descriptors obtained from Structure from Motion (SfM) to construct the 2D-3D correspondences, and OnePose++ [31] further proposes coarse to fine matching to obtain more accurate correspondences. SAM-6D [19] learns 3D-3D correspondences through a two-stage point matching process incorporating background tokens. FoundPose [22] leverages the generalization capabilities of foundation models to extract pixel features and establish 2D-3D correspondences.

3 Method

3.1 Overview of PicoPose

The goal of novel object pose estimation from RGB images is to determine the 6D transformation between an RGB observation and a CAD model of an object unseen during training. To address this task, we introduce a novel framework for generalizable **pi**xel-to-pixel **co**rrespondence learning, termed as **PicoPose**. The architecture of PicoPose is illustrated in Fig. 2. For a given RGB image of a clustered scene and an object CAD model, we begin by rendering object templates from various viewpoints of the CAD model and employing zero-shot segmentation [23] to identify and crop the region containing the target object from the RGB scene. We then resize both the detected crop and the object templates to a fixed size of $H \times W$, representing them as $\mathcal{I}$ and $\{\mathcal{T}_i\}_{i=1}^{K}$, respectively, where K is the number of templates. PicoPose utilizes these inputs to search for the best-matched template, denoted as $\mathcal{T}$, and progressively learns the pixel-to-pixel correspondences between $\mathcal{I}$ and $\mathcal{T}$ across three stages, as detailed in Sec. 3.2. Each foreground pixel on $\mathcal{T}$ corresponds to a 3D surface point on the CAD model, enabling us to establish pairs of 2D positions on $\mathcal{I}$ and their corresponding 3D points on $\mathcal{T}$, which are utilized for computing the 6D object pose via PnP/RANSAC.

3.2 Progressive Pixel-to-Pixel Correspondence Learning of PicoPose

Given the resized detected crop $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ and the object templates $\{\mathcal{T}_i \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{K}$, PicoPose employs the ViT-L backbone [32], pretranined by DINOv2 [16], to extract their patch features $\mathcal{F}_{\mathcal{I}} \in \mathbb{R}^{N\times D}$ and $\{\mathcal{F}_{\mathcal{T}_i} \in \mathbb{R}^{N\times D}\}_{i=1}^{K}$, respectively, where D is the feature dimension and N is the number of patches. PicoPose then identifies the best-matched templates $\mathcal{T}$ and performs three learning stages to build pixel-to-pixel correspondences between $\mathcal{I}$ and $\mathcal{T}$.

Stage 1: Feature Matching for Coarse Correspondences

Feature similarities between patches of the RGB observation $\mathcal{I}$ and the object templates $\{\mathcal{T}_i\}_{i=1}^{K}$, particularly with the best-matched template to $\mathcal{I}$, could establish coarse correspondences. To make it more effective, we initially retrieve the best-matched template $\mathcal{T}$ by scoring the degree of similarity between each template $\mathcal{T}_i$ and the observation $\mathcal{I}$. For each template $\mathcal{T}_i$, we obtain this template

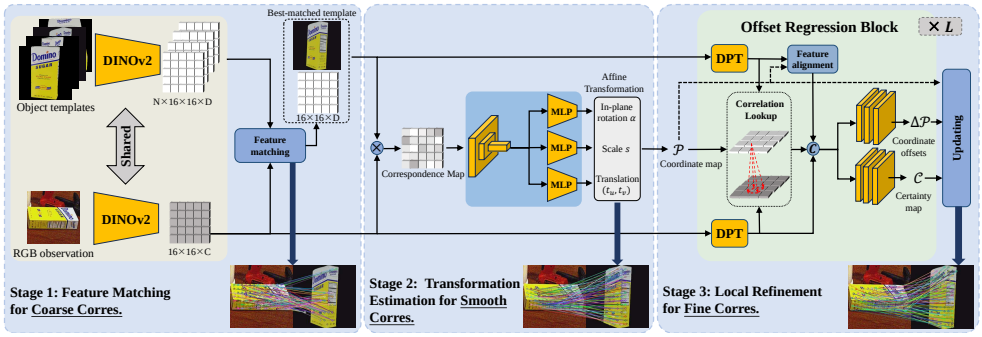

Figure 2: An illustration of our proposed **PicoPose**.

matching score c_i by averaging the maximum feature cosine similarities of the foreground patches in $\mathcal{I}$ (identified by a prior-step zero-shot segmentation) with the patches in $\mathcal{T}_i$ as follows:

$$c_i = \frac{1}{N'} \sum_{j \in \texttt{FG}(\mathcal{F}_\mathcal{I})} \max_{k=1,\dots,N} \frac{< \mathbf{f}_{\mathcal{I},j}, \mathbf{f}_{\mathcal{T}_i,k} >}{|\mathbf{f}_{\mathcal{I},j}| \cdot |\mathbf{f}_{\mathcal{T}_i,k}|}, \tag{1}$$

where $\mathbf{f}_{\mathcal{I},j} \in \mathcal{F}_\mathcal{I}$ and $\mathbf{f}_{\mathcal{T}_i,k} \in \mathcal{F}_{\mathcal{T}_i}$ are the j^{th} patch features in $\mathcal{I}$ and the k^{th} patch features in $\mathcal{T}_i$, respectively, with $< \cdot, \cdot >$ denoting an inner product. $\texttt{FG}(\mathcal{F}_\mathcal{I})$ represents the indices of foreground patches in $\mathcal{I}$, and N' is the count of foreground patches. From $\{\mathcal{T}_i\}_{i=1}^K$, the template $\mathcal{T}$ with the highest score is chosen as the best match. Furthermore, the feature similarities between $\mathcal{I}$ and $\mathcal{T}$ give coarse correspondences by determining the most similar patch in $\mathcal{T}$ for each patch in $\mathcal{I}$.

Stage 2: Global Transformation Estimation for Smooth Correspondences

In Stage 1, we exploit feature matching to obtain the coarse and sparse correspondences, which, however, often exhibit cluttered distributions with noise and many outliers. Therefore, the objective of this stage is to improve the smoothness of these correspondences and filter out the outliers.

To achieve the objective, we adopt a global approach to estimate the 2D affine transformation $\mathcal{M}$ between $\mathcal{I}$ and $\mathcal{T}$, which can be parameterized with 4 degrees of freedom (DoFs) [21] as follows:

$$\mathcal{M} = \begin{bmatrix} s\cos(\alpha) & -s\sin(\alpha) & t_u \\ s\sin(\alpha) & s\cos(\alpha) & t_v \end{bmatrix}, \tag{2}$$

where α denotes the in-plane rotation angle, s denotes the relative scale between $\mathcal{I}$ and $\mathcal{T}$, and (t_u, t_v) represents the 2D translation of the object centroid in these two images. Applying $\mathcal{M}$ to transform the template $\mathcal{T}$ facilitates pixel alignments with $\mathcal{I}$, thus enabling smooth correspondences.

As highlighted in Fig. 3, the correspondence map $\mathcal{A}$ between $\mathcal{I}$ and $\mathcal{T}$ can effectively capture the variations in α, s, and (t_u, t_v), thereby encapsulating the essential patterns for learning $\mathcal{M}$. Therefore, instead of directly concatenating the patch features $\mathcal{F}_\mathcal{I}$ and $\mathcal{F}_\mathcal{T}$ for regressing $\mathcal{M}$, we propose a more effective approach by utilizing the coarse correspondences obtained in Stage 1, represented as the correspondence map $\mathcal{A}$ at this stage, to realize the target. More specifically, we first normalize the feature vectors of $\mathcal{F}_\mathcal{I}$ and $\mathcal{F}_\mathcal{T}$ to $\overline{\mathcal{F}}_\mathcal{I}$ and $\overline{\mathcal{F}}_\mathcal{T}$, respectively, and compute the correspondence map $\mathcal{A}$ as $\mathcal{A} = \overline{\mathcal{F}}_\mathcal{I}(\overline{\mathcal{F}}_\mathcal{T})^T \in \mathbb{R}^{N \times N}$, where $N = HW/196$. Subsequently, $\mathcal{A}$ is reshaped to the size of $(H/14) \times (W/14) \times N$ and passed through several stacked convolutions to reduce the spatial dimensions to a global pose vector. Finally, three parallel Multi-layer Perceptrons (MLPs) are applied to the pose vector to learn $(cos(\alpha), sin(\alpha))$, s and (t_u, t_v) of $\mathcal{M}$, respectively.

Stage 3: Local Refinement for Fine Correspondences

With the transformation $\mathcal{M}$ predicted in Stage 2, we can align the feature map of the best-matched template $\mathcal{T}$ with the RGB observation $\mathcal{I}$, thereby achieving the smooth correspondences. For any position (u, v) on $\mathcal{I}$, the corresponding position (u', v') on $\mathcal{T}$ could be obtained using $\mathcal{M}$ as follows:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \mathcal{M} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{3}$$

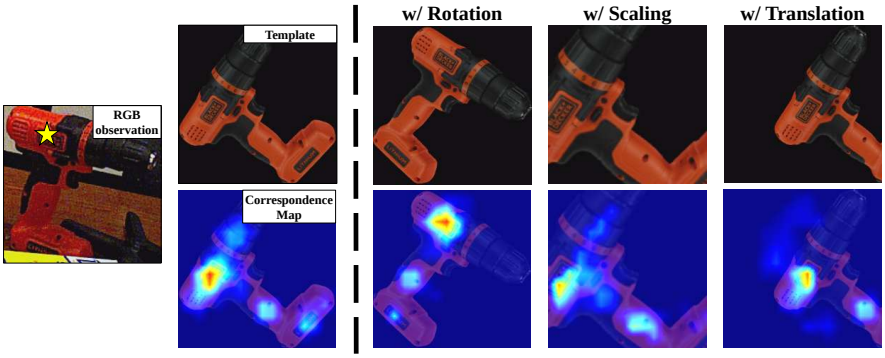

Figure 3: Visualization of correspondence maps between the feature of a point on the RGB observation (marked by a yellow star) and the features of templates with various affine transformations.

The deviation between (u, v) and (u', v') can be interpreted as the commonly known "optical flow" [33]. For all pixels in $\mathcal{I}$, we compute their corresponding positions in $\mathcal{T}$, denoted as $\mathcal{P} \in \mathbb{R}^{H \times W \times 2}$, via Eq. (3) to represent the smooth correspondences. The affine transformation applied to the feature map of $\mathcal{T}$ is then achieved by using $\mathcal{P}$ as indices to gather features from $\mathcal{T}$, producing a transformed feature map aligned with $\mathcal{I}$. At this stage, we further learn the offsets $\Delta\mathcal{P} \in \mathbb{R}^{H \times W \times 2}$ to update $\mathcal{P}$ to $\mathcal{P} + \Delta\mathcal{P}$, enabling fine-grained correspondence adjustments within local regions.

We realize local refinement of correspondences in a progressive learning manner. Specifically, we first apply Dense Prediction Transformer (DPT) [34] to our backbone, generating L hierarchical feature maps $\{\mathcal{F}_{\mathcal{I}l} \in \mathbb{R}^{H_l \times W_l \times D_l}\}_{l=1}^{L}$ and $\{\mathcal{F}_{\mathcal{T}l} \in \mathbb{R}^{H_l \times W_l \times D_l}\}_{l=1}^{L}$ of $\mathcal{I}$ and $\mathcal{T}$, respectively, where $H_l \times W_l$ denotes the spatial size and D_l is the number of channels for the l^{th} feature map. We then use L offset regression blocks to iteratively update $\mathcal{P}$.

For the l^{th} offset regression block, the current $\mathcal{P}$ is resized to $H_l \times W_l \times 2$ and scaled by dividing each 2D position within it by $[H/H_l, W/W_l]$ to ensure spatial consistency. We denote this resized and scaled version as $\mathcal{P}_l$, which we use as indices to gather features from $\mathcal{F}_{\mathcal{T}l}$, resulting in the transformed feature map $\mathcal{F}'_{\mathcal{T}l}$ to align with $\mathcal{F}_{\mathcal{I}l}$. Additionally, we introduce a third correlation feature map $\mathcal{F}_{\mathcal{C}l}$ via a Correlation Lookup module, introduced in RAFT [35], to explicitly provide correlation degrees and facilitate easier learning of offsets; more details on this module can be found in the RAFT paper [35]. We then concatenate $\mathcal{F}_{\mathcal{I}l}$, $\mathcal{F}'_{\mathcal{T}l}$, and $\mathcal{F}_{\mathcal{C}l}$ to form the input for two sequences of stacked convolutions, which are used to regress offsets $\Delta\mathcal{P}_l \in \mathbb{R}^{H_l \times W_l \times 2}$ and certainty map $\mathcal{S}_l \in \mathbb{R}^{H_l \times W_l}$. $\Delta\mathcal{P}_l$ is then interpolated to the size of $H \times W \times 2$, scaled by multiplying the 2D coordinates within it by $[H/H_l, W/W_l]$, and added to $\mathcal{P}$ for updating. The certainty map $\Delta\mathcal{S}_l$ represents the confidence of the regressed offsets and is upsampled to generate $\mathcal{C}'_l \in \mathbb{R}^{H \times W}$.

With L offset regression blocks, we have the fine-grained $\mathcal{P}$, with a certainty map $\frac{1}{L}\sum_{i=1}^{L} \mathcal{C}'_l$. For each foreground pixel in $\mathcal{I}$, if its correspondence certainty exceeds 0.5, we use the position in $\mathcal{P}$ to find the corresponding pixel in $\mathcal{T}$, which is linked to a 3D surface point. Therefore, all the pixel-to-pixel correspondences generate the associated 2D-3D pairs to compute the final object pose.

3.3 Training of PicoPose

We perform end-to-end training of the three-stage correspondence learning process in PicoPose by optimizing the following objective:

$$\min \mathcal{L} = \mathcal{L}_{coarse} + \mathcal{L}_{smooth} + \mathcal{L}_{fine}, \tag{4}$$

where $\mathcal{L}_{coarse}$, $\mathcal{L}_{smooth}$, and $\mathcal{L}_{fine}$ are the loss terms associated with each of the three stages.

In Stage 1, we adopt the InfoNCE loss [36], as used in GigaPose [21], as the training objective $\mathcal{L}_{coarse}$ to learn feature matching. In Stage 2, we predict the 2D affine transformation $\mathcal{M}$, including in-plane rotation angle α, scale s, and 2D translation (t_u, t_v), to generate smooth correspondences between $\mathcal{I}$ and $\mathcal{T}$. Letting $\hat{\alpha}$, $\hat{s}$, and $(\hat{t}_u, \hat{t}_v)$ represent the respective ground truths of the predicted

Method	#Hypothesis	BOP Dataset							Mean
		LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	
w/o Iterative Refinement									
MegaPose [18]	-	22.9	17.7	25.8	15.2	10.8	25.1	28.1	20.8
GenFlow [29]	-	25.0	21.5	30.0	16.8	15.4	28.3	27.7	23.5
GigaPose [21]	1	27.8	26.3	27.8	21.4	16.9	31.2	27.6	25.6
FoundPose [22]	1	39.5	39.6	**56.7**	28.3	26.2	58.5	49.7	42.6
PicoPose (Ours)	1	**46.3**	**39.7**	53.6	**36.4**	**31.0**	**66.5**	**58.7**	**47.5**
GigaPose [21]	5	29.6	26.4	30.0	22.3	17.5	34.1	27.8	26.8
FoundPose w/o FM [22]	5	39.6	33.8	46.7	23.9	20.4	50.8	45.2	37.2
FoundPose [22]	5	42.0	**43.6**	**60.2**	30.5	27.3	53.7	51.3	44.1
PicoPose (Ours)	5	**49.2**	41.3	58.4	**37.8**	**32.7**	**67.6**	**57.6**	**49.2**
w/ Refiner of MegaPose [18]									
MegaPose [18]	1	49.9	47.7	**65.3**	36.7	31.5	65.4	60.1	50.9
GigaPose [21]	1	55.7	54.1	58.0	45.0	37.6	69.3	63.2	54.7
FoundPose w/o FM [22]	1	55.4	51.0	63.3	43.0	34.6	69.5	66.1	54.7
FoundPose [22]	1	55.7	51.0	63.3	43.3	35.7	69.7	66.1	55.0
PicoPose (Ours)	1	**60.5**	**56.6**	63.6	**46.5**	**40.1**	**75.9**	**68.7**	**58.8**
MegaPose [18]	5	56.0	50.7	68.4	41.4	33.8	70.4	62.1	54.7
GigaPose [21]	5	59.8	56.5	63.1	47.3	39.7	72.2	66.1	57.8
FoundPose w/o FM [22]	5	58.6	54.9	65.7	44.4	36.1	70.3	67.3	56.8
FoundPose [22]	5	61.0	57.0	**69.4**	47.9	40.7	72.3	69.0	59.6
PicoPose (Ours)	5	**61.1**	**57.1**	65.0	**48.2**	**42.1**	**76.3**	**69.6**	**59.9**

Table 1: Quantitative results of different methods on BOP datasets [26]. We report the mean Average Recall (AR) among VSD, MSSD and MSPD. 'FM' denotes featuremetric pose refinement [22].

parameters, we define the training objective $\mathcal{L}_{smooth}$ for this stage as follows:

$$\mathcal{L}_{smooth} = \mathcal{L}_{geo}(\alpha, \hat{\alpha}) + |\ln(s) - \ln(\hat{s})| + |t_u - \hat{t}_u| + |t_v - \hat{t}_v|, \tag{5}$$

where $\mathcal{L}_{geo}(\alpha, \hat{\alpha})$ is the geodesic distance between two angles α and $\hat{\alpha}$, defined as follows:

$$\mathcal{L}_{geo}(\alpha, \hat{\alpha}) = \text{acos}\big(\cos(\alpha)\cos(\hat{\alpha}) + \sin(\alpha)\sin(\hat{\alpha})\big)\ . \tag{6}$$

In Stage 3, we employ the L_1 distance and the binary cross entropy objective to guide the learning of coordinate offsets $\{\Delta\mathcal{P}_l\}_{l=1}^{L}$ and certainty maps $\{\mathcal{C}_l\}_{l=1}^{L}$ across all L offset regression blocks:

$$\mathcal{L}_{fine} = \sum_{l=1}^{L} \lambda||\hat{\mathcal{C}}_l \cdot (\Delta\mathcal{P}_l - \Delta\hat{\mathcal{P}}_l)|| + \mu\mathcal{L}_{bce}(\mathcal{C}_l, \hat{\mathcal{C}}_l), \tag{7}$$

where $\{\Delta\hat{\mathcal{P}}_l\}_{l=1}^{L}$ and $\{\hat{\mathcal{C}}_l\}_{l=1}^{L}$ represent the corresponding ground truths of $\{\Delta\mathcal{P}_l\}_{l=1}^{L}$ and $\{\mathcal{C}_l\}_{l=1}^{L}$, while λ and μ are the weights to balance the loss terms. $\mathcal{L}_{bce}$ denotes the binary cross entropy objective. For the supervision of $\Delta\mathcal{P}_l$, we use $\hat{\mathcal{C}}_l$ to mask and exclude invalid correspondences.

4 Experiments

Datasets. we train PicoPose on synthetic datasets of ShapeNet-Objects [24] and Google-Scanned-Objects [25] provided by [18], using a total of 2 million training images. Evaluation is conducted on seven BOP datasets [26], including LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, and YCB-V.

Evaluation Metrics. We report the mean Average Recall (AR) w.r.t three error functions, i.e., Visible Surface Discrepancy (VSD), Maximum Symmetry-Aware Surface Distance (MSSD), and Maximum Symmetry-Aware Projection Distance (MSPD) [26] . We also report the end-point-error (EPE), a widely used metric in flow estimation [35], to assess the quality of the correspondences.

4.1 Comparisons with Existing Methods

We evaluate PicoPose against existing methods on the seven core datasets of the BOP benchmark [26]. Inspired by GigaPose [21], we enhance the robustness of PicoPose by using the top 5 templates in Stage 2 and Stage 3 to learn fine correspondences and selecting the poses that best match these correspondences. The quantitative results are shown in Table 1, where PicoPose significantly outperforms other methods, highlighting its superior zero-shot capability for novel object pose estimation through progressive correspondence learning. For example, a single model of PicoPose using

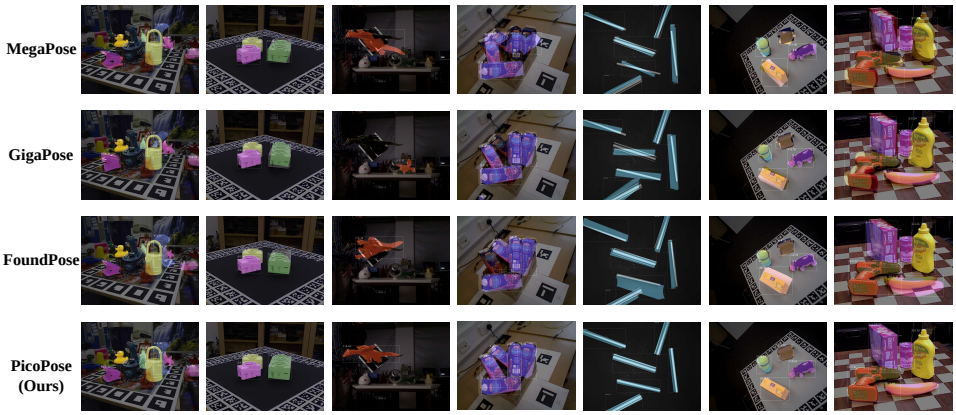

Figure 4: Qualitative results of different methods without iterative refinement on BOP datasets [26], including LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, and YCB-V, arranged from left to right.

Method	#Hypothesis	AR
GigaPose [21]	1	17.9
PicoPose (Ours)	1	**20.4**
GigaPose [21]	5	18.3
PicoPose (Ours)	5	**22.0**

Table 2: Quantitative results of single reference image on LM-O.

Method	#Hyp	Server	Time (s)
GigaPose [21]	5	NVIDIA V100	0.640
FoundPose [22]	5	Tesla P100	3.360
GigaPose [21]	1	GeForce RTX 3090	0.631
GigaPose [21]	5		1.209
PicoPose (Ours)	1		0.659
PicoPose (Ours)	5		1.562

Table 3: Per-image runtime on LM-O.

top 5 templates outperforms the single models of GigaPose [21] and FoundPose [22] by 22.4% and 5.1% AR, respectively. In Table 1, we also report results with the iterative refinement proposed by MegaPose [18], where PicoPose consistently outperforms the others, whether using 1 or 5 pose hypotheses for refinement. The visualizations in Fig. 4 further validate the advantages of PicoPose.

Results with Single Object Reference Images. Since obtaining perfect object CAD models is not always practical, object images are sometimes used as references. Following GigaPose [21], we evaluate PicoPose in the most extreme scenario, i.e., with only one reference image, by employing Wonder3D [37] to reconstruct the object CAD model from this image. As shown in Table 2, PicoPose can successfully handle this extreme setting, achieving results comparable to GigaPose.

Runtime Analysis. We report the per-image processing time, including segmentation and pose estimation, of different methods without iterative refinement in Table 3. For a fair comparison, both GigaPose [21] and PicoPose are tested on the same servers; as shown in Table 3, PicoPose achieves comparable speeds while delivering more impressive results, demonstrating its accuracy and efficiency. Notably, while FoundPose [22] employs more advanced servers, it still incurs significantly higher computational costs when matching its extensive template library (800 templates per object). In contrast, both PicoPose and GigaPose use only 162 templates per object.

4.2 Ablation Studies and Analyses

We conduct ablation studies to evaluate the efficacy of designs in PicoPose. Except for specific cases, the results are achieved using only the best-matched templates in Stage 2 and Stage 3.

Efficacy of Progressive Correspondence Learning. The key to the success of PicoPose lies in its design of progressive pixel-to-pixel correspondence learning. To evaluate this, we first analyze the quality improvements of correspondences and examine their impact on pose estimation. For Stage 2, the predicted affine transformations are applied to obtain correspondences. As shown in Table 4, pose precision improves with finer correspondences, supporting the core claim of this paper. Stage 2 smooths coarse correspondences and filters outliers (Fig. 5), significantly improving translation accuracy (errors < 5 cm), while its overall 6D pose enhancement over Stage 1 is marginal, as Stage 2

Stage	LM-O	T-LESS	YCB-V	MEAN
6D Pose Estimation (AR ↑)				
1	28.6	27.3	41.2	32.4
2	31.2	25.6	45.5	34.1
3	**46.3**	**39.7**	**58.7**	**48.2**
Translation Estimation (Accuracy ↑)				
1	40.3	43.1	60.2	47.9
2	43.2	48.4	69.6	53.7
3	**62.6**	**56.0**	**78.7**	**65.8**
Correspondence Estimation (EPE ↓)				
1	3.6	4.4	4.5	4.2
2	3.0	4.2	2.1	3.1
3	**2.1**	**3.8**	**1.2**	**2.4**

Table 4: Quantitative comparisons among different stages of correspondence learning.

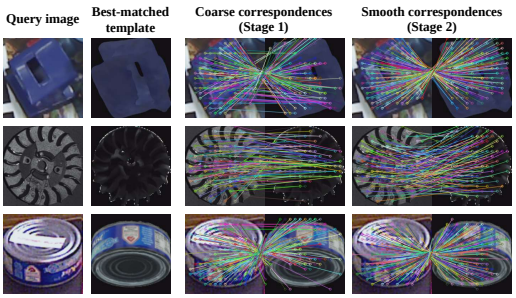

Figure 5: Visualization comparisons between the coarse correspondences from Stage 1 and the smooth ones from Stage 2.

Method	Pose Estimation (AR ↑)				Processing Time on LM-O (s ↓)		
	LM-O	T-LESS	YCB-V	MEAN	Model Forward	Post-Processing	ALL
$\mathbf{F}_{ist}$ in GigaPose [21]	26.0	22.5	25.3	24.9	0.465	0.166	0.631
Stage 2 w/ concatenated features	27.0	21.7	35.0	27.9	0.329	-	0.329
Stage 2 w/ correspondence map	**31.2**	**25.6**	**45.5**	**34.1**	0.329	-	0.329

Table 5: Quantitative comparisons among different variants of Stage 2.

does not include viewpoint rotation updates like Stage 1 with PnP. Stage 3 further enhances precision with a 14.1% AR improvement by locally refining the correspondences from Stage 2.

Efficacy of Stage 2. We first assess the effectiveness of Stage 2 by presenting the results without it in Table 6, where we initialize the input correspondences for Stage 3 in two ways: 1) by directly using the coarse correspondences from Stage 1, and 2) by using the predicted poses from Stage 1 via PnP/RANSAC to obtain smoother correspondences. The first approach yields less precise results due to the high noise in the correspondences, while the second is less efficient because of the additional PnP/RANSAC processing. Next, we conduct experimental comparisons with regression based on direct feature concatenation of the observation and the template. As shown in Table 5, learning from correspondence maps proves to be more effective, as it explicitly models coarse correspondences and effectively captures the variations in affine transformations. Additionally, we replace our Stage 2 with $\mathbf{F}_{ist}$ from GigaPose [21], which is less efficient, as discussed in Sec. 3.2.

Efficacy of Stacked Offset Regression Blocks in Stage 3. In Stage 3, we use L offset regression blocks, specifically $L = 3$ in our experiments with feature spatial sizes from DPT [34] set to 16×16, 32×32, and 64×64, to refine correspondences within local regions. The results from different offset regression blocks are reported in Table 7, where performance progressively improves as more blocks are used, indicating that finer correspondences are achieved through progressive local refinement.

Initial Correspondence	LM-O		T-LESS	
	AR	Time (s)	AR	Time (s)
Matching in Stage 1	34.5	0.655	25.0	0.612
Pose from Stage 1	45.7	0.936	33.9	0.823
Pose from Stage 2	**46.3**	0.659	**39.7**	0.617

Table 6: Quantitative results with different initial correspondences inputted to Stage 3.

OR Block	Size	LM-O	T-LESS	YCB-V	MEAN
1	16×16	33.4	27.2	42.3	34.3
2	32×32	42.9	36.6	53.9	44.5
3	64×64	**46.3**	**39.7**	**58.7**	**48.2**

Table 7: Quantitative results of different offset regression blocks (denoted as "OR Block") in Stage 3.

5 Conclusion

In this paper, we propose PicoPose, a novel framework for object pose estimation from RGB images that uses progressive pixel-to-pixel correspondence learning across three carefully designed stages. We demonstrate the zero-shot capabilities of PicoPose on seven core datasets of the BOP benchmark, enabling practical deployment in robotic applications. In future work, we aim to further increase the speed of PicoPose to achieve real-time performance and explore ways to reduce its reliance on templates.

6 Limitations

While PicoPose demonstrates strong performance on 6D object pose estimation benchmarks with real-world cluttered scenes and shows promising potential for rapid deployment in robotic applications through simulated grasping experiments, several areas remain for future improvement.

First, the reliance on multiple templates for 3D object representation means that its performance is inherently tied to the number of templates used. Although our approach achieves comparable results with fewer templates than existing methods (as detailed in Supplementary Section 2), future work could explore more efficient template utilization strategies or alternative 3D representations.

Second, while PicoPose benefits from the iterative refinement of MegaPose [18], the performance gains are relatively modest compared to other methods, probably because PicoPose's initial estimates already approach the refiner's performance upper bound. This motivates the development of specialized refinement approaches for high-quality initial predictions.

Third, in case of a single reference image of the target object, where we follow GigaPose [21] to use Wonder3D for reconstructing the object in 3D space, the quality of reconstruction remains a limiting factor that affects final pose estimation accuracy. We believe integrating emerging neural reconstruction techniques could significantly improve performance in this challenging scenario.

References

[1] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis. Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943, 2014. doi:10.1109/ICRA.2014.6907430.

[2] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning (CoRL)*, 2018. URL https://arxiv.org/abs/1809.10790.

[3] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[4] H. Mnyusiwalla, P. Triantafyllou, P. Sotiropoulos, M. A. Roa, W. Friedl, A. M. Sundaram, D. Russell, and G. Deacon. A bin-picking benchmark for systematic evaluation of robotic pick-and-place systems. *IEEE Robotics and Automation Letters*, 5(2):1389–1396, 2020. doi:10.1109/LRA.2020.2965076.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[6] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[7] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. 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.

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

[9] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022.

[10] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. 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.

[11] J. Lin, Z. Wei, Z. Li, S. Xu, K. Jia, and Y. Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3560–3569, 2021.

[12] J. Lin, Z. Wei, C. Ding, and K. Jia. Category-level 6d object pose and size estimation using self-supervised deep prior deformation networks. In *European Conference on Computer Vision*, pages 19–34. Springer, 2022.

[13] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, and F. 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.

[14] J. Lin, Z. Wei, Y. Zhang, and K. Jia. Vi-net: Boosting category-level 6d object pose estimation via learning decoupled rotations on the spherical representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14001–14011, 2023.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[16] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[18] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022.

[19] J. Lin, L. Liu, D. Lu, and K. Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27906–27916, 2024.

[20] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[21] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9903–9913, 2024.

[22] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan. Foundpose: Unseen object pose estimation with foundation features. In *European Conference on Computer Vision*, pages 163–182. Springer, 2025.

[23] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2134–2140, 2023.

[24] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[25] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[26] T. Hodan, M. Sundermeyer, Y. Labbe, V. N. Nguyen, G. Wang, E. Brachmann, B. Drost, V. Lepetit, C. Rother, and J. Matas. Bop challenge 2023 on detection segmentation and pose estimation of seen and unseen rigid objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2024.

[27] N. Gao, V. A. Ngo, H. Ziesche, and G. Neumann. SA6d: Self-adaptive few-shot 6d pose estimator for novel and occluded objects. In *7th Annual Conference on Robot Learning*, 2023.

[28] D. Cai, J. Heikkilä, and E. Rahtu. Ove6d: Object viewpoint encoding for depth-based 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6803–6813, 2022.

[29] S. Moon, H. Son, D. Hur, and S. Kim. Genflow: Generalizable recurrent flow for 6d pose refinement of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2024.

[30] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[31] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35:35103–35115, 2022.

[32] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[33] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[34] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[35] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[36] A. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[37] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024.

[38] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. *http://pybullet. org*, 2016–2019.

[39] V. Lepetit, F. Moreno-Noguer, and P. Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009.

[40] X. Liu, R. Zhang, C. Zhang, B. Fu, J. Tang, X. Liang, J. Tang, X. Cheng, Y. Zhang, G. Wang, and X. Ji. Gdrnpp. https://github.com/shanice-l/gdrnpp_bop2022, 2022.

[41] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. 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.

[42] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[43] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[44] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3583–3592, 2016.

[45] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2200–2208, 2017.

[46] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

Appendix

This appendix is structured as follows. In **Section A**, we provide more details on the network architecture, training setup, equipment, pose solving configuration, and data augmentation methods used in our research. In **Section B**, we perform more ablation studies on the effects of correlation lookup in Stage 3 and the influence of the number of templates on pose precision. **Section C** shows more qualitative results of different stages in PicoPose and more qualitative comparisons of different methods on real-world benchmarks. In **Section D**, we conduct robotic grasping experiments in a simulation environment using the PyBullet physics engine [38] to verify the application of our method in robotic grasping.

A More Experimental Details

Network architecture details. In Stage 1, we build on previous work [21] by using dinov2_vitl14 as our backbone. The feature dimension of this backbone is $1,024$ for each token. In Stage 2, we employ two conventional layers with group normalization and a ReLU activation function to reduce the spatial size of the input correspondence map $\mathcal{A}$ to 8×8, after which we flatten the feature map to obtain the global pose vector. In Stage 3, in order to reduce network parameters, we set the number of channels D_l of the l^{th} feature map generated from DPT [34] to 256. Additionally, we utilize the standard lookup operation in RAFT [35] for L blocks. Since the feature sizes in each block vary, the hyperparameter settings of each block need to be adjusted individually. For the l^{th} block, we establish the layer of the correlation pyramid as $l + 1$ and set the radius of the correlation lookup to 4. We also list the model sizes of different stages in Table 8.

Stage	#Param
1	304 M
2	18 M
3	58 M
Total	380 M

Table 8: The model sizes of different stages in our network.

Details of training settings. We report the detailed hyperparameter settings to train our network in Table 9.

Hyperparameters	Settings
Optimizer	AdamW
AdamW β	(0.5, 0.99)
AdamW eps	1e-6
Learning rate scheduler	Cosine decay
Training iterations	$400,000$
Warmup iterations	$1,000$
Learning rate	1e-5
Weight decay	5e-4
Batch size	32

Table 9: Detailed hyperparameters in training our network.

More details of devices. We conducted all experiments with GPU in GeForce RTX 3090 24G, and CPU in Intel (R) Xeon (R) CPU E5-2678 v3 @ 2.50 GHz under the Linux operating system.

Details of PnP/RANSAC. We utilize the EPnP algorithm [39] along with the RANSAC scheme in the fine correspondence to solve the object pose in Stage 3. The RANSAC iterations are configured to 150, and the reprojection error threshold is set to 2.

Details of augmentations. In Stage 1, to better adapt to the input images in real-world scenarios, we conduct data augmentation on the query image of the training data in a manner similar to GDRNPP

[40]. In Stage 3, we apply random 2D translation, in-plane rotation, and scale noise to the ground truth affine transformation $\mathcal{M}$ to generate the initial coordinate map $\mathcal{P}$.

B More Ablation Studies

Effects of correlation lookup in Stage 3. In Stage 3, we use the correlation lookup operation in RAFT [35] to obtain flow features, and combine them with the features of the input image and the best-matched template to predict coordinate offsets and the certainty map. To verify the effectiveness, we conducted experiments without using the correlation lookup operation. As shown in Table 10, the features obtained by the correlation lookup operation can improve the results significantly.

Correlation Lookup	LM-O	T-LESS	YCB-V	MEAN
×	35.0	25.9	46.8	35.9
✓	**46.3**	**39.7**	**58.7**	**48.2**

Table 10: Quantitative results of correlation lookup operation in Stage 3. We report the mean Average Recall (AR) among VSD, MSSD and MSPD.

Influence of the number of templates. We follow the setup of GigaPose [21] by using $K = 162$ templates per object in our evaluation experiments. In Table 11, we present additional quantitative results with different numbers of templates for both GigaPose and our proposed PicoPose. The results show improvement as more templates are used, since both methods rely on template matching to select the best-matched template for the target object. However, the rate of improvement slows as the number of templates increases. Notably, PicoPose is more effective than GigaPose when using fewer templates, further highlighting the advantages of PicoPose.

Method	#Temp	LM-O	T-LESS	YCB-V	MEAN
GigaPose [21]	2	4.8	4.4	2.0	3.7
PicoPose		**10.5**	**12.8**	**14.1**	**12.5**
GigaPose [21]	6	11.5	9.2	5.9	8.9
PicoPose		**27.5**	**25.0**	**39.5**	**30.7**
GigaPose [21]	42	25.0	23.3	23.4	23.9
PicoPose		**43.9**	**37.9**	**57.4**	**46.4**
GigaPose [21]	162	29.6	26.4	27.8	27.9
PicoPose		**46.3**	**39.7**	**58.7**	**48.2**

Table 11: Quantitative comparison with GigaPose [21] on the number of templates. We report the mean Average Recall (AR) among VSD, MSSD and MSPD.

C Additional Qualitative Results

More qualitative results of different methods. We present more qualitative results of different methods on the seven core datasets (LM-O[41], T-LESS[42], TUD-L[43], IC-BIN[44], ITODD[45], HB[46], and YCB-V[6]) in the BOP benchmark [26], shown in Fig. 6. We illustrate the estimated 6D pose by rendering the 3D model on the input image and using the overlap rate as a basis, where a higher overlap rate indicates a more accurate estimated 6D pose. Specifically, all methods use the same zero-shot segmentation method, i.e., CNOS [23].

More qualitative results of different stages. We visualize the correspondences between the query image $\mathcal{I}$ and the best-matched template $\mathcal{T}$ in Stage 1 and Stage 2. As shown in Fig. 7, the coarse correspondences generated in Stage 1 contain numerous outliers and inconsistencies, many of which are effectively resolved by Stage 2 to produce smooth correspondences. In Stage 3, we enhance the display of fine correspondences by visualizing the coordinate map $\mathcal{P}$ as optical flow with the certainty map on YCB-V dataset [6], shown in Fig. 8.

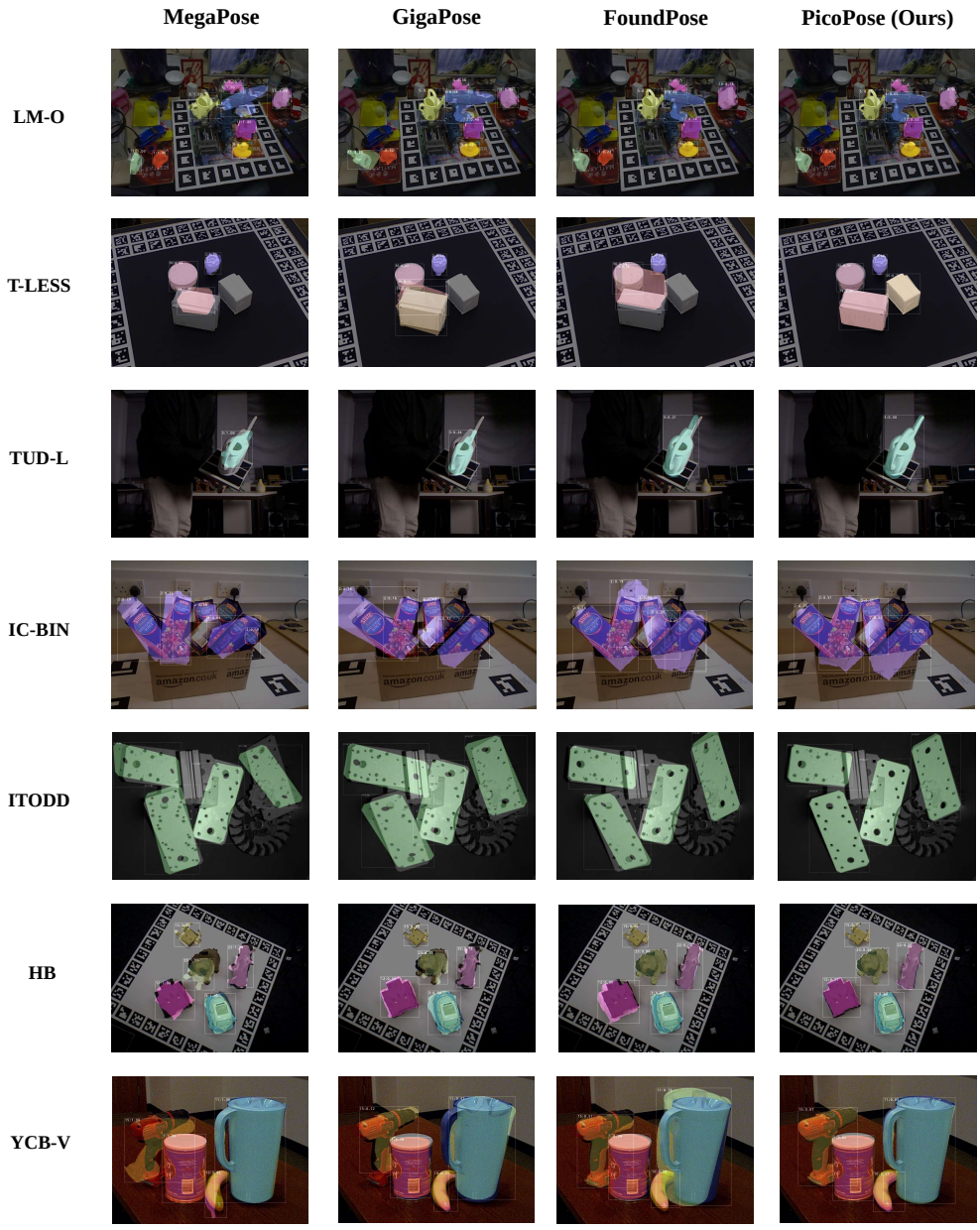

Figure 6: More qualitative results of different methods on the seven core datasets of BOP benchmark [26], including LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, and YCB-V, arranged from top to bottom.

D Application: Robotic Grasping in Simulated Environments

In this section, our proposed PicoPose demonstrates seamless integration for robotic grasping applications using PyBullet [38] with the setup shown in Fig. 9. Our experimental scene comprises (1) a Franka Emika Panda robotic arm, (2) distractor objects, including the target, randomly arranged on the workspace, and (3) a placement tray. A fixed virtual camera captures single RGB images of the cluttered scene as input to our system.

The processing pipeline consists of three key stages. First, CNOS [23] segments the target object in the RGB scene. Second, our proposed PicoPose estimates the 6D pose of the target object in camera

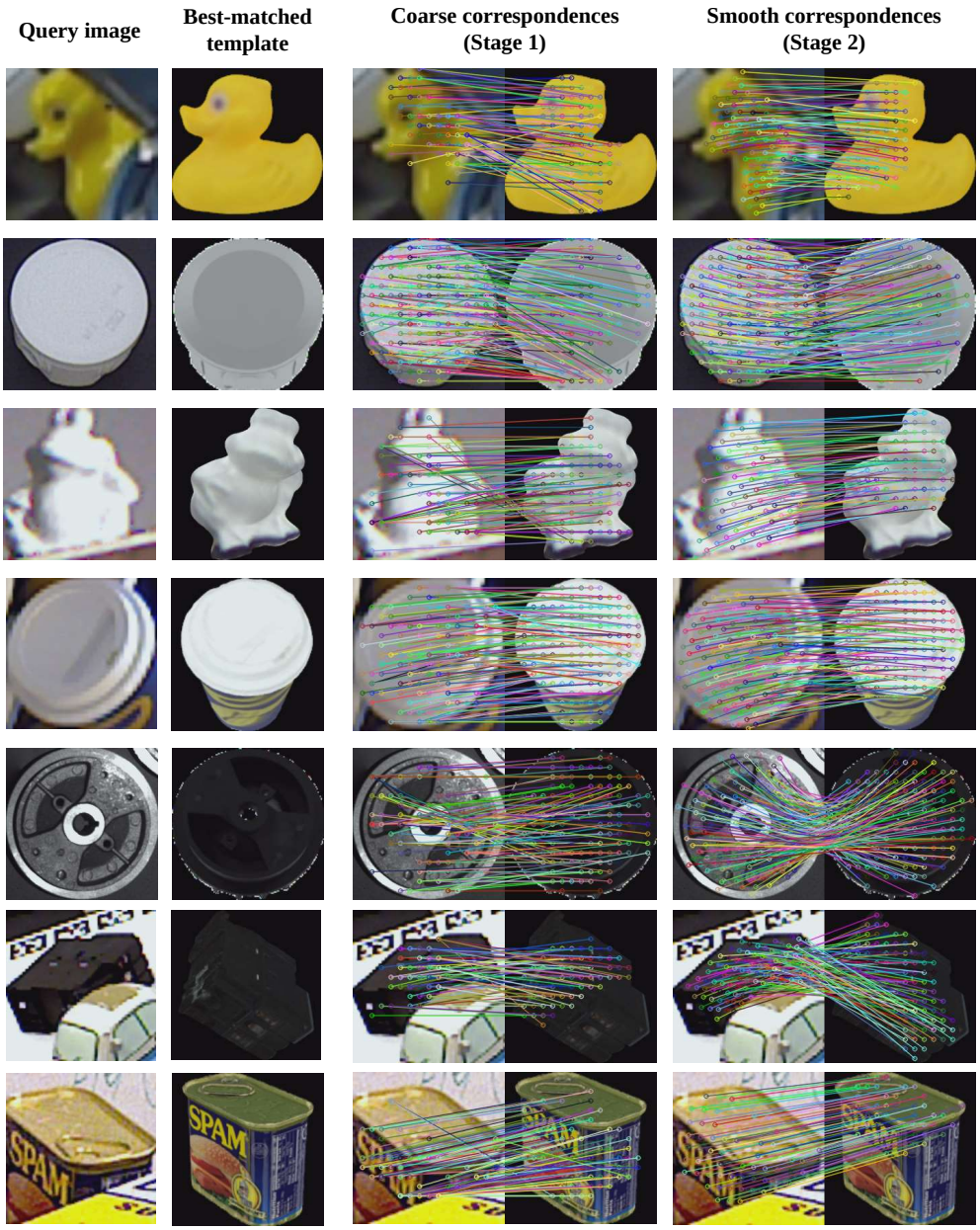

Figure 7: Qualitative results of coarse correspondences in Stage 1 and smooth correspondences in Stage 2 on the seven core datasets of BOP benchmark [26], including LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, and YCB-V, arranged from top to bottom.

Figure 8: Qualitative results of fine correspondences in Stage 3 on YCB-V dataset [6].

coordinates. Third, we use the known camera-to-robot coordinate transformation to convert this pose into a 6D grasping pose and use inverse kinematics to generate robot motions to successfully grasp the target object.

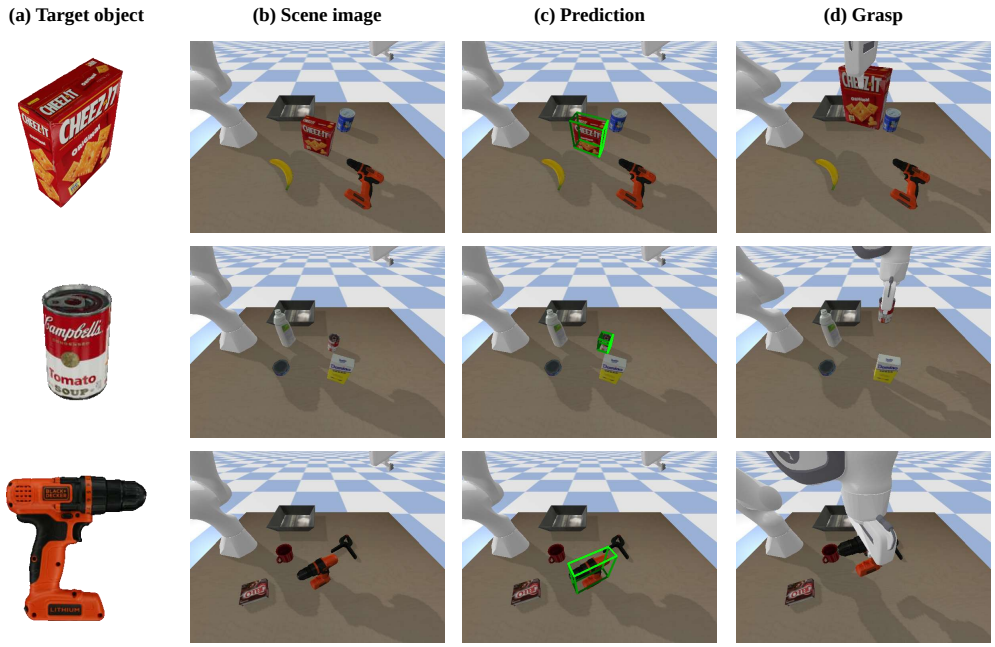

Figure 9: Robotic grasping application of PicoPose in simulated environment.