OnePose++: Keypoint-Free One-Shot Object Pose Estimation without CAD Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a new method for object pose estimation without CAD models. The previous feature-matching-based method OnePose has shown promising results under a one-shot setting, eliminating the object-specific training and aiming for the pose estimation of arbitrary unseen objects. However, OnePose relies on detecting repeatable image keypoints and is thus prone to fail on low-textured objects. We propose to remove the need for keypoint detection with a keypoint-free SfM and pose estimation pipeline. Built upon the keypoint-free feature matching method LoFTR, our novel keypoint-free SfM reconstructs object semi-dense point clouds, which provides complete 3D models for pose estimation of low-textured objects. During pose estimation, our novel keypoint-free 2D-3D matching network matches the reconstructed point cloud with the query image to build 2D-3D correspondences for object pose estimation. The keypoint-free SfM and pose estimation pipeline enables pose estimation of low-textured objects under the one-shot setting, making it more applicable in the real world. We demonstrate that the proposed method outperforms existing one-shot CAD-model-free methods by a large margin and even achieves comparable results with CAD-model-based methods on LINEMOD. We also collect a new dataset composed of 80 sequences of 40 low-textured objects to boost future research on one-shot object pose estimation. Code and data will be made available upon the publication of this paper.

1 Introduction

Object pose estimation is crucial for immersive human-object interactions in augmented reality (AR). The AR scenario demands the pose estimation of arbitrary household objects in our daily lives. However, most existing methods [24, 16, 23, 34, 2, 3, 22] either rely on high-fidelity object CAD models or require training a separate network for each object category. The instance- or category-specific nature of these methods limits the applications of object pose estimation in the real world.

To alleviate the need for CAD models or category-specific training, OnePose [30] proposes a new setting of *one-shot object pose estimation*. It assumes that only a video sequence with annotated poses is available for each object, and aims for its pose estimation in arbitrary environments. This setting eliminates the requirements for CAD models and the separated training for each object, thus is applicable to arbitrary objects and more friendly for applications in AR. OnePose adopts the feature-matching-based visual localization pipeline for this setting. It reconstructs sparse object point clouds with SfM [27] and establishes 2D-3D correspondences between keypoints in the query image and the point cloud model to estimate the object pose. Being dependent on detecting repeatable keypoints, OnePose struggles with low-textured objects whose complete point clouds are difficult to construct with keypoint-based SfM. Without complete point clouds, pose estimation is prone to fail for many household low-textured objects.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

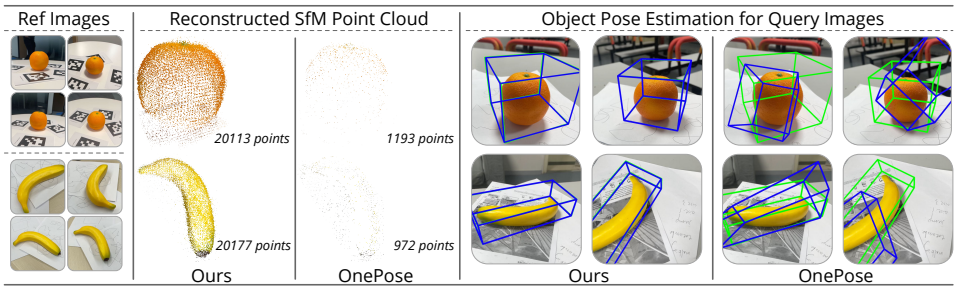

Figure 1: **Comparsion between our method and the keypoint-based baseline OnePose [30].** For low-textured objects challenging for OnePose, our method can reconstruct their semi-dense point clouds with complete geometry and thus achieves accurate object pose estimation. Green boxes represent ground truth poses, and the blue ones for predictions.

We propose to use a keypoint-free feature matching pipeline on top of OnePose to handle low-textured objects. The keypoint-free semi-dense feature matching method LoFTR [29] achieves outstanding performance on matching image pairs and shows strong capabilities for finding correspondences in low-textured regions. It uses centers of regular grids on a left image as "keypoints", and extracts sub-pixel accuracy matches on the right image in a coarse-to-fine manner. However, this two-view-dependent nature leads to inconsistent "keypoints" and fragmentary feature tracks, which go against the preference of modern SfM systems. Therefore, keypoint-free feature matching cannot be directly applied to the OnePose framework for object pose estimation. We will further discuss this problem in Sec. 3.1.

To get the best of both worlds, we design a dedicated system to adapt keypoint-free matching for one-shot object pose estimation. We propose a two-stage pipeline for reconstructing a 3D structure, striving for both accuracy and completeness. Then, for testing, our sparse-to-dense 2D-3D matching network efficiently establish accurate 2D-3D correspondences for pose estimation, taking full advantage of our keypoint-free design.

More specifically, to better adapt LoFTR [29] for SfM, we design a coarse-to-fine framework for accurate and complete semi-dense object reconstruction. We disassemble the coarse-to-fine structure of LoFTR and integrate them into our reconstruction pipeline compactly. In the coarse reconstruction phase, we use less accurate yet repeatable LoFTR coarse correspondences for SfM to construct consistent feature tracks and yield an inaccurate but complete semi-dense point cloud. Then, our novel refinement phase optimizes the initial point cloud by refining "keypoints" in coarse feature tracks to sub-pixel accuracy. As shown in Fig. 1, our framework can reconstruct accurate and complete semi-dense point clouds, even for low-textured objects, which lays the foundation for building high-quality 2D-3D correspondences for pose estimation.

At test time, we draw inspiration from the sparse-to-dense matching strategy in visual localization [6], and further adapt it to direct 2D-3D matching in a coarse-to-fine manner for efficiency. Additionally, we use self- and cross-attention to model long-range dependencies required for robust 2D-3D matching and pose estimation of complex real-world objects, usually containing repetitive patterns or low-textured regions.

We evaluate our framework on the OnePose [30] dataset and the LINEMOD [8] dataset. The experiments show that our method outperforms all existing model-free generalizable methods [30, 19] by a large margin and even achieves comparable results with instance-level methods [24, 16], which are trained for each object instance with its corresponding CAD model. To further evaluate and demonstrate the capability of our method in real-world scenarios, we collect a new dataset named OnePose-HARD , which comprises 80 sequences of 40 low-textured objects.

Contributions.

- A keypoint-free SfM framework for accurate and complete semi-dense reconstruction of both texture-full and low-textured objects.
- A sparse-to-dense 2D-3D matching framework for accurate object pose estimation.
- A challenging real-world dataset OnePose-HARD composed of 40 low-textured objects with ground-truth object pose annotations.

2 Related works

CAD-model-based Object Pose Estimation. Many previous methods leverage known object CAD models for pose estimation, which can be further categorized into instance-level, category-level, and generalizable methods by their generalizability. Instance-level methods estimate object poses either by directly regressing poses from images [36, 12, 16] or construct 2D-3D correspondences and then solve poses with PnP [24, 37]. The primary deficiency is that these methods need to train a separate network for each object. Category-level methods, such as [34, 31, 13, 33, 35, 3], learn the shape prior shared within a category and eliminate the need for CAD models in the same category at test time. However, these methods cannot handle objects in unseen categories. Some recent methods leverage the generalization power of 2D feature matching for the pose estimation of unseen objects. Reference images are rendered with CAD models, then matched with the query image, using either sparse keypoints matching [38] or dense optical flow [28]. All methods mentioned above depend on high-fidelity textured CAD models for training or rendering, which are not easily accessible in real-world applications. Our framework, instead, reconstructs a 3D object model from pose-annotated images for object pose estimation.

CAD-model-free Object Pose Estimation. Some recent methods get rid of the CAD model completely. RLLG [2] uses correspondences between image pairs as supervision for training, without a known object model. However, it still requires accurate object masks as supervision, which are not easily accessible without a CAD model. NeRF-Pose [14] reconstructs a neural representation NeRF [21] for an object first and train an object coordinate regression network for pose estimation. These methods are not capable of generalizing to unseen objects. The recently proposed Gen6D [19] and OnePose [30] only require a set of reference images with annotated poses to estimate object poses and can generalize to unseen objects. Gen6D uses detection and retrieval to initialize the pose of a query image and then refine it by regressing the pose residual. However, Gen6D requires an accurately detected 2D bounding box for pose initialization and struggles with occlusion scenarios. OnePose reconstructs objects' sparse point cloud and then extracts 2D-3D correspondences for solving poses. It performs poorly on low-textured objects because of its reliance on repeatably detected keypoints . Our work is inspired by OnePose but eliminates the need for keypoints in the entire framework, which leads to better performance on both textured and low-textured objects.

Structure from Motion and Visual Localization. Visual localization estimates camera poses of query images relative to a known scene structure. The scene structure is usually reconstructed with Structure-from-Motion(SfM) [27], relying on correspondences of image keypoints [20, 4]. The localization problem is then solved by finding 2D-3D correspondences between the 3D scene model and a query image. HLoc [25] scales up visual localization in a coarse-to-fine manner. It first retrieves n candidate images attached to the 3D scene for a query image. Then, it establishes 2D-3D correspondences with 2D-2D matchings between the query image and all its candidate images. HLoc cannot be used for the pose estimation of low-textured objects because it relies on repeatable keypoints for both SfM and localization. Moreover, HLoc is slow during pose estimation because it depends on multiple 2D-2D image matchings as the proxy for building 2D-3D correspondences. In contrast, our keypoint-free SfM framework performs well on low-textured objects. We efficiently establish 2D-3D correspondences between a point cloud and a 2D query image with only a single forward pass. Note that our keypoint-free SfM framework is also related to PixSfM [18] but comes with different motivations. PixSfM improves keypoint-based SfM for more accurate 3D reconstructions by refining inaccurately-detected sparse local features with dense feature maps. Different from PixSfM, we aim to adapt the keypoint-free method LoFTR [29] to SfM for object pose estimation.

3 Method

An overview of our method is shown in Fig. 2. Given a reference image sequence with known object poses $\{\mathbf{I}_i, \boldsymbol{\xi}_i\}$, our objective is to estimate the object poses $\{\boldsymbol{\xi}_q\}$ for the test images, where i, q denotes the index of the reference images and test images, respectively. To achieve this goal, we propose a novel two-stage pipeline, which first reconstructs the accurate semi-dense object point cloud from reference images (Section 3.2), then solves the object pose by building 2D-3D correspondences in a coarse-to-fine manner for test images (Section 3.3). Since our method is highly related to the keypoint-free matching method LoFTR [29], we give it a short overview in Section 3.1.

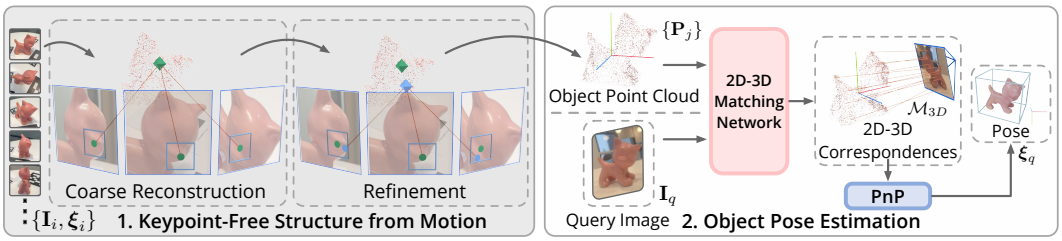

Figure 2: **Overview. 1.** For each object, given a reference image sequences $\{\mathbf{I}_i\}$ with known object poses $\{\boldsymbol{\xi}_i\}$, our keypoint-free SfM framework reconstructs the semi-dense object point cloud in a coarse-to-fine manner. The coarse reconstruction yields the initial point cloud (◆) which are then optimized to obtain an accurate point cloud (◆) in the refinement phase. **2.** At test time, our 2D-3D matching network directly matches a reconstructed object point cloud with a query image $\mathbf{I}_q$ to build 2D-3D correspondences $\mathcal{M}_{3D}$, then the object pose $\boldsymbol{\xi}_q$ is estimated by solving PnP with $\mathcal{M}_{3D}$.

3.1 Background

Keypoint-free feature matching method LoFTR. Without a keypoint detector, LoFTR [29] build semi-dense matches between image pairs (noted as left and right images) in a coarse-to-fine pipeline. First, dense matches between two coarse-level feature maps (1/8 resolution in LoFTR) are built and upsampled, yielding coarse semi-dense matches in the original resolution. With the locations of all left matches fixed, the right matches are refined to a sub-pixel level using fine-level feature maps. Thanks to the keypoint-free design and the global receptive field of Transformers, LoFTR is capable of building correspondences in low-textured regions.

Problem of using LoFTR for keypoint-based SfM. Directly combining LoFTR with modern keypoint-based SfM systems such as COLMAP [27] is not applicable since they rely on fixed keypoints detected on each image to construct feature tracks for estimating 3D structures. However, for LoFTR, its matching locations on a right image depend on its pairing left images. Therefore, the right matching locations are not consistent when paired with multiple left images. Due to this reason, keypoint-free feature matching cannot establish feature tracks across multiple views for effective 3D structure optimization in SfM and is thus not directly applicable in OnePose.

3.2 Keypoint-Free Structure from Motion

To better adapt LoFTR for SfM, we design a coarse-to-fine SfM framework leveraging the properties of LoFTR's coarse and fine stages separately. Our framework constructs the coarse structure of the feature tracks $\{\mathcal{T}_c^j\}$ and point cloud $\{\mathbf{P}_c^j\}$ in the coarse reconstruction phase. Then in the refinement phase, the coarse structures are refined to obtain the accurate point cloud $\{\mathbf{P}^j\}$. For clarity, in this part, we use $\tilde{\cdot}$ to denote the coarse matching results and use $\hat{\cdot}$ to denote fine matching results. We consider the feature track $\mathcal{T}^j = \{\mathbf{u}^k \in \mathbb{R}^2 | k = 1...N_j\}$ as a set of matched 2D points observing a 3D point $\mathbf{P}^j \in \mathbb{R}^3$. j denotes the index of the feature track and its corresponding 3D point.

Coarse Reconstruction. We first strive for the completeness of the initially constructed 3D structure. We propose to use the inaccurate yet repeatable coarse correspondences of LoFTR for COLMAP [27] to construct the coarse 3D structure. The coarse correspondences, as shown in Fig. 3(**1**), can be seen as pixel-wise dense correspondences on downsampled image pairs. Every pixel in the downsampled image can be regarded as a "keypoint" in the original image. Therefore, performing coarse matching can provide repeatable semi-dense correspondences for COLMAP to construct coarse feature tracks $\{\mathcal{T}_c^j\}$ and semi-dense point cloud $\{\mathbf{P}_c^j\}$, as shown in Fig. 3(**2**).

Refinement. Due to the limited accuracy of performing matching on downsampled images, the point cloud from the coarse reconstruction is inaccurate and thus insufficient for the object pose estimation. Therefore, we further refine the object point cloud $\{\mathbf{P}_c^j\}$ with sub-pixel correspondences. To achieve this, we first fix the position of one node for each feature track $\mathcal{T}_c^j$ and refine other nodes within the track. Then we use the refined tracks $\{\mathcal{T}_f^j\}$ to optimize the $\{\mathbf{P}_c^j\}$.

For the refinement of $\{\mathcal{T}_c^j\}$, we draw the idea from the LoFTR fine module and adapt it to the multi-view scenario. As shown in Fig.3(**3**), we first select and fix one node in each $\mathcal{T}_c^j$ as the reference node $\mathbf{u}_r$, and then perform fine matching with each of the remaining source node $\tilde{\mathbf{u}}_s^k$. The fine

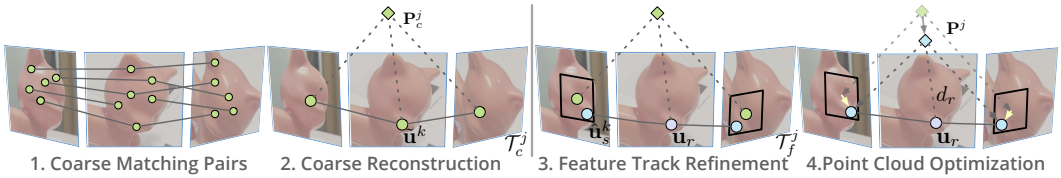

Figure 3: **Keypoint-Free SfM. 1.** We firstly build repeatable coarse semi-dense 2D matches between image pairs. **2.** Then we feed coarse matches to COLMAP [27] to build a coarse feature track $\mathcal{T}_c^j$ and a coarse 3D point $\mathbf{P}_c^j$ (◇). **3.** To refine $\mathcal{T}_c^j$, we fix a reference node $\mathbf{u}_r$ (○) and search around the local window (□) of each source node $\tilde{\mathbf{u}}_s^k$ (○) for sub-pixel correspondences $\hat{\mathbf{u}}_s^k$ (○). **4.** Then, we optimize the depth d_r of $\mathbf{u}_r$ by minimizing reprojection errors. We back-project $\mathbf{u}_r$ with its refined d_r to the object coordinate to obtain a optimized accurate object point cloud $\mathbf{P}^j$ (◇).

matching searches within a local region around each $\tilde{\mathbf{u}}_s^k$ for a sub-pixel correspondence $\hat{\mathbf{u}}_s^k$, the nodes' locations in the coarse feature track are thus refined. We denote the refined feature tracks as $\{\mathcal{T}_f^j\}$. Details about the selection of reference nodes $\mathbf{u}_r$ are provided in the supplementary.

We now treat the refined feature tracks $\{\mathcal{T}_f^j\}$ as fixed measurements, and optimize the 3D locations of the coarse point cloud $\{\mathbf{P}_c^j\}$ using reprojection errors as shown in Fig. 3(**4**). To accelerate the convergence, inspired by SVO [5], we further decrease the DoF of each $\mathbf{P}_c^j$ by optimizing the depth d_r of each reference node $\mathbf{u}_r$ only. Specifically, we transform each point $\mathbf{P}_c^j$ to the frame of $\mathbf{u}_r$ and use its coordinate of z-axis to initialize the d_r. Then we optimize each reference node depth d_r by minimizing the distance between each reprojected location and the refined feature location $\hat{\mathbf{u}}_s^k$:

$$d_r^* = \underset{d_r}{\operatorname{argmin}} \frac{1}{2} \sum_k \| \hat{\mathbf{u}}_s^k - \boldsymbol{\pi} \left(\boldsymbol{\xi}_{r \to s_k} \cdot \boldsymbol{\pi}^{-1} \left(\mathbf{u}_r, d_r \right) \right) \|^2, \tag{1}$$

where $\boldsymbol{\pi}$ is the projection determined by intrinsic camera parameters, and $\boldsymbol{\xi}_{r \to s_k} = \boldsymbol{\xi}_{s_k} \cdot \boldsymbol{\xi}_r^{-1}$ is the relative pose between the frame of the reference node and kth source node.

Finally, the optimized depth d_r^* of each reference node is transformed to the canonical object coordinate to get the refined 3D point $\mathbf{P}^j$. Notably, when applying the proposed system in practical AR applications, we can optimize inaccurate camera poses obtained from ARKit along with the 3D points, i.e., solve a bundle adjustment problem. For the later 2D-3D matching at test time, we calculate and store each 3D point feature by averaging the 2D features of its associated 2D points. Note that we store coarse and fine 3D features separately, which are extracted from multi-resolution feature maps of LoFTR's feature backbone.

3.3 Object Pose Estimation

At test time, we establish 2D-3D matches between the object point cloud $\{\mathbf{P}^j\}$ and the query image $\mathbf{I}_q$ to estimate object pose $\boldsymbol{\xi}_q$. Inspired by [29], we first extract hierarchical feature maps of $\mathbf{I}_q$ and then perform matching in a coarse-to-fine manner for efficiency, as illustrated in Fig.4.

Coarse 2D-3D matching. We first perform dense matching between the pre-calculated coarse 3D point features $\tilde{\mathbf{F}}_{3D} \in \mathbb{R}^{N \times C_c}$ and the extracted coarse image feature map $\tilde{\mathbf{F}}_{2D} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_c}$. This phase globally searches for a rough correspondence of each 3D object point on the query image, which determines whether the 3D point is observable by $\mathbf{I}_q$.

We augment 3D and 2D features $\{\tilde{\mathbf{F}}_{3D}, \tilde{\mathbf{F}}_{2D}\}$ with positional encodings, to make them position-dependent, thus facilitates their matching. Please refer to the supplementary material for more details. We then flatten the 2D feature map and perform self- and cross-attention layers by N_c times to yield the transformed features $\{\tilde{\mathbf{F}}_{3D}^t, \tilde{\mathbf{F}}_{2D}^t\}$. Linear Attention [11] is used in our model to reduce the computational complexity, following [29]. A score matrix S is calculated as the similarity between two sets of features $\tilde{\mathbf{F}}_{3D}^t$ and $\tilde{\mathbf{F}}_{2D}^t$. We then apply dual-softmax operation [32] on S to get the matching probability matrix $\mathcal{P}^c$:

$$\mathcal{P}^c(j,q) = \operatorname{softmax}\left(\mathrm{S}\left(j,\cdot\right)\right)_q \cdot \operatorname{softmax}\left(\mathrm{S}\left(\cdot,q\right)\right)_j, \text{where } \mathrm{S}\left(j,q\right) = \frac{1}{\tau} \cdot \langle \tilde{\mathbf{F}}_{3D}^t(j), \tilde{\mathbf{F}}_{2D}^t(q) \rangle. \tag{2}$$

$\langle \cdot, \cdot \rangle$ is the inner product, τ is a scale factor, and j, q denotes the indices of a 3D point and a pixel in the flattened query image, respectively. The coarse 2D-3D correspondences $\mathcal{M}_{3D}^c$ are established from $\mathcal{P}^c$ by selecting correspondences above a confidence threshold θ:

$$\mathcal{M}_{3D}^c = \{(j,q) \mid \forall (j,q) \in \mathrm{MNN}\left(\mathcal{P}^c\right),\ \mathcal{P}^c\left(j,q\right) \geq \theta\}. \tag{3}$$

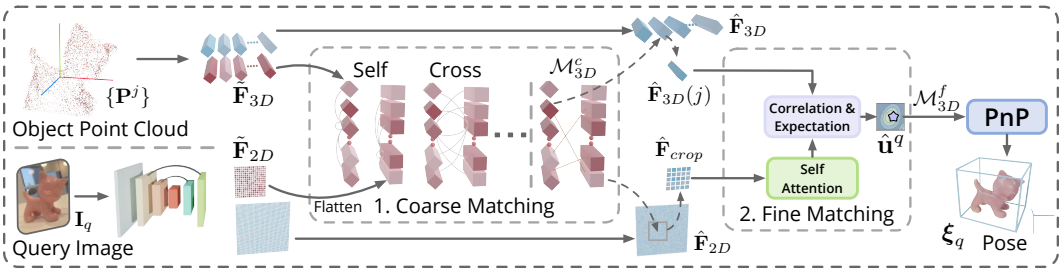

Figure 4: **Object Pose Estimation.** At test time, we first extract multi-scale query image features $\{\tilde{\mathbf{F}}_{2D}, \hat{\mathbf{F}}_{2D}\}$. **Coarse matching** module transforms coarse 2D and 3D features N_c times with self- and cross-attention module and then build their coarse 2D-3D correspondences $\mathcal{M}^c_{3D}$. Next, we crop the local window $\hat{\mathbf{F}}_{crop}$ on the fine feature map around each coarse 2D match. **Fine matching** module transforms the cropped 2D features and calculates each 2D fine match location $\hat{\mathbf{u}}^q$ with feature correlation and expectation. The object pose $\boldsymbol{\xi}_q$ is then solved using PnP with $\mathcal{M}^f_{3D}$.

MNN refers to the mutual nearest neighbor. We use this strict criterion to suppress potential false matches.

Fine matching. For a visible 3D point $\mathbf{P}^j$ determined by $\mathcal{M}^c_{3D}$, our fine matching module searches for its sub-pixel 2D correspondence $\hat{\mathbf{u}}^q$ within the local region of its coarse correspondence $\tilde{\mathbf{u}}^q$. Similar to [29], we crop a local window W with a size of $w \times w$ around $\tilde{\mathbf{u}}^q$ in the fine feature map $\hat{\mathbf{F}}_{2D} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C_f}$, and transform the cropped feature $\hat{\mathbf{F}}_{crop} \in \mathbb{R}^{w \times w \times C_f}$ with N_f self-attention layers. Then we correlate the 3D fine feature vector $\hat{\mathbf{F}}_{3D}(j) \in \mathbb{R}^{C_f}$ with all elements in the transformed 2D feature $\hat{\mathbf{F}}^t_{crop}$ and apply a softmax to get the probability distribution of its 2D correspondence in the cropped local window:

$$p(\mathbf{u}|j, \hat{\mathbf{F}}_{3D}, \hat{\mathbf{F}}_{crop}) = \frac{\exp\left(\hat{\mathbf{F}}_{3D}(j)^{\mathrm{T}} \cdot \hat{\mathbf{F}}_{crop}(\mathbf{u})\right)}{\sum_{\mathbf{u} \in W} \exp\left(\hat{\mathbf{F}}_{3D}(j)^{\mathrm{T}} \cdot \hat{\mathbf{F}}_{crop}(\mathbf{u})\right)}, \tag{4}$$

The fine correspondence $\hat{\mathbf{u}}^q$ of $\mathbf{P}^j$ on the query image is then obtained with an expectation:

$$\hat{\mathbf{u}}^q = \tilde{\mathbf{u}}^q + \sum_{\mathbf{u} \in W} \mathbf{u} \cdot p(\mathbf{u}|j, \hat{\mathbf{F}}_{3D}, \hat{\mathbf{F}}_{crop}), \tag{5}$$

After building the 2D-3D correspondences $\mathcal{M}^f_{3D}$ between the query image and the object point cloud, we solve the object pose $\boldsymbol{\xi}_q$ with the Perspective-n-Point (PnP) and RANSAC.

Supervision. We jointly train the coarse and fine modules of our 2D-3D matching framework with different supervisions, following [29]. The ground-truth 2D-3D correspondences $\mathcal{M}^{gt}_{3D}$ and the coarse matching probability matrix $\mathcal{P}^c_{gt}$ are obtained by projecting observable SfM points to the 2D frame with the ground-truth object pose. We optimize the coarse matching module by minimizing the focal loss [17] between the predicted $\mathcal{P}^c$ and $\mathcal{P}^c_{gt}$. For the fine module, we minimize the ℓ_2 loss between the predicted 2D coordinate $\hat{\mathbf{u}}^q$ and the ground truth $\hat{\mathbf{u}}^q_{gt}$. The total loss is the weighted sum of coarse and fine losses. More details are provided in the supplementary material.

3.4 Implementation Details

Keypoint-Free SfM. The COLMAP [27] triangulation is used to construct the coarse 3D structure. In the refinement phase, the local window with a size of 9×9 is searched for the sub-pixel correspondence of each reference node. We use the Levenberg-Marquardt algorithm implemented in DeepLM [10] to optimize the coarse object point cloud. We use the LoFTR [29] outdoor model pre-trained on MegaDepth [15]. Running time analyses are illustrated in supplementary.

Object pose estimation. We use ResNet-18 [7] as the image backbone and set $N_c = 3, N_f = 1$ for the 2D-3D attention module. The scale factor τ is 0.08, the cropped window size w in the fine level is 5, and the confidence threshold θ is set to 0.4. As for training, the backbone of our model is initialized with the LoFTR outdoor model, and the remaining parts of our model use randomly initialized weights. The entire model is trained on the OnePose training set, and we randomly sample or pad the reconstructed point cloud to 7000 points for training. We use the AdamW optimizer with an initial learning rate of 4×10^{-3}. The network training takes about 20 hours with a batch size of 32 on 8 NVIDIA-V100 GPUs. During testing, we use all reconstructed 3D points($\sim$ 15000) for building 2D-3D correspondences, and our 2D-3D matching module runs at 87ms for a 512×512 query image on a single V100 GPU.

Table 1: **Comparison with *generalizable* baselines.** Our method is compared with HLoc[25] combined with different feature matching methods and OnePose [30].

	OnePose dataset			OnePose-HARD		
	1cm-1deg	3cm-3deg	5cm-5deg	1cm-1deg	3cm-3deg	5cm-5deg
HLoc *(SPP + SPG)*	**51.1**	75.9	82.0	13.8	36.1	42.2
HLoc *(LoFTR*)*	39.2	72.3	80.4	13.2	41.3	52.3
OnePose	49.7	77.5	84.1	12.4	35.7	45.4
Ours	50.7	**80.0**	**87.0**	**16.3**	**55.4**	**70.3**

Table 2: **Comparison with the *instance-level* baseline** Our method is compared with PVNet[24] on objects with CAD models in the OnePose-HARD dataset on the *ADD(S)-0.1d* metric.

Obj. ID	0600	0601	0604	0606	0607	0623	0627	0639	0640	0641	Mean
PVNet	12.1	90.0	68.1	68.1	9.20	67.6	95.6	57.3	49.6	**61.3**	57.9
Ours	**89.5**	**99.1**	**97.2**	**86.5**	**86.3**	**92.6**	**98.5**	**79.5**	**97.2**	57.6	**88.4**

4 Experiments

4.1 Dataset

OnePose and LINEMOD dataset. We validate our method on the OnePose [30] and LINEMOD [8] dataset. The OnePose dataset is newly proposed for object pose estimation, which contains around 450 real-world video sequences of 150 objects. LINEMOD is a broadly used dataset for object pose estimation. For both datasets, we follow the train-test split in previous methods [30, 16] to conduct experiments.

OnePose-HARD dataset. Since the original OnePose evaluation set mainly comprises textured objects and is not challenging enough, we collected an additional evaluation set, named OnePose-HARD, to supplement the original OnePose dataset. The OnePose-HARD dataset is composed of 40 household low-textured objects. Each object of OnePose-HARD has two corresponding videos captured with different backgrounds, one as the reference video, and the other is used for the test. Besides, to evaluate and compare our method with CAD-model-based methods, we further obtain high-fidelity 3D models for ten randomly selected objects with a commercial 3D scanner. Please refer to the supplementary for more details.

4.2 Experiment Settings and Baselines

Baselines. We compare the proposed method with the following baselines in two categories: 1) *Generalizable baselines* [30, 19, 25] that hold the same setting as ours. OnePose [30] and HLoc [25] are most relevant to our method in leveraging feature matching for reconstruction and pose estimation. To be specific, we compare with HLoc combined with different feature matching methods including SuperGlue [26] and LoFTR [29]. 2) *Instance-level baselines* [24, 16] that need to be trained separately for each object. These methods achieve high accuracy through training on many rendered images with extensive augmentations. We compare our method with them to demonstrate that our method achieves competitive results while not relying on CAD models and generalizes well to unseen objects.

Evaluation Protocols. We compare our method with OnePose and HLoc using the same set of reference images. Since HLoc's original retrieval module is designed for the outdoor scene, we use uniformly sampled 10 reference views for 2D-2D matching for pose estimation, following [30]. For the comparison with PVNet [24], we follow its original training setting, which first samples 8 keypoints on the object surface and then trains a network using 5000 synthetic images for each object. In contrast, our method only uses around 200 reference images to reconstruct the object point cloud. We evaluate our method and PVNet on the same real-world test sequence, while our matching model has never seen this object before. As for the experiments on LINEMOD, our method and OnePose share the same 2D bounding boxes from an off-the-shelf object detector YOLOv5 [1]. We use the training images (~ 180) for object reconstruction and all test images for evaluation. Note that the object detector is trained on real-world images only to provide rough bounding boxes. The results of other baselines on LINEMOD are from the original papers.

Table 3: **Results on LINEMOD.** Our method is compared with *instance-level* and *generalizable* baselines. Note that Gen6D is fine-tuned on a selected subset of objects and uses the rest for testing. Gen6D† is the version without fine-tuning on LINEMOD. Symmetric objects are indicated by *.

Type	Name	Object Name													Avg.
		ape	benchwise	cam	can	cat	driller	duck	eggbox*	glue*	holepuncher	iron	lamp	phone	
		ADD(S)-0.1d													
Ins.	CDPN	67.3	98.8	92.8	96.6	86.6	95.1	75.2	99.6	99.6	89.7	97.9	97.8	80.7	91.4
	PVNet	43.6	99.9	86.9	95.5	79.3	96.4	52.6	99.2	95.7	81.9	98.9	99.3	92.4	86.3
Gen.	Gen6D†	-	62.1	45.6	-	40.9	48.8	16.2	-	-	-	-	-	-	-
	Gen6D	-	77.0	66.1	-	60.7	67.4	40.5	95.7	**87.2**	-	-	-	-	-
	OnePose	11.0	90.4	85.4	71.9	46.7	71.0	31.2	65.1	32.7	48.5	84.9	86.0	59.2	60.3
	Ours	**31.2**	**97.3**	**88.0**	**89.8**	**70.4**	**92.5**	**42.3**	**99.7**	48.0	**69.7**	**97.4**	**97.8**	**76.0**	**76.9**
		Proj2D													
Ins.	CDPN	97.5	98.8	98.6	99.6	99.3	94.9	98.4	99.1	98.4	99.5	97.9	95.7	96.8	98.0
	PVNet	99.2	99.8	99.2	99.9	99.3	96.9	98.0	99.3	98.5	100.0	99.2	98.3	99.4	99.0
Gen.	OnePose	33.9	95.3	96.4	86.2	74.5	74.7	72.6	85.6	**53.5**	76.5	89.5	88.7	69.2	76.7
	Ours	**97.3**	**99.6**	**99.6**	**99.2**	**98.7**	**93.1**	**97.7**	**98.7**	51.8	**98.6**	**98.9**	**98.8**	**94.5**	**94.3**

Metrics. We use metrics including the *cm-degree* pose error, the *ADD(S)-0.1d* average distance with a 10% object diameter threshold, and the 2D projection error *Proj2D* with a threshold of 5 pixels. The definitions of these metrics are detailed in the supplementary.

4.3 Results on OnePose and OnePose-HARD dataset

Comparison with Generalizable baselines. The *cm-deg* metric with different thresholds are used for evaluation. As shown in Tab. 1, our method substantially outperforms OnePose [30] and HLoc [25]. Objects in the OnePose dataset have rich textures, benefiting the keypoint detection. Therefore, keypoint-based methods OnePose and HLoc(*SPP+SPG*) perform reasonably well. Our method achieves even higher accuracy thanks to the keypoint-free design, effectively utilizing both texture-full and low-textured object regions for pose estimation. On the OnePose-HARD dataset, composed of low-textured objects, our method surpasses OnePose and HLoc by a large margin. This further demonstrates the capability of our keypoint-free design for the object reconstruction and the sparse-to-dense 2D-3D matching for the object pose estimation. HLoc(*LoFTR**) uses LoFTR coarse matches for SfM and uses full LoFTR to match the query and its retrieved images for pose estimation. It does not rely on keypoints, similar to our design. Our method significantly outperforms it on accuracy and runs $\sim 10\times$ faster. The improved accuracy and speed come from the accurate point cloud reconstructed by our novel SfM framework and the efficient 2D-3D matching module.

Comparison with Instance-Level baseline PVNet. On the OnePose-HARD dataset, the proposed method is compared with PVNet [24] on the subset objects with scanned models. The *ADD(S)-0.1d* results are presented in Tab.2. Even though PVNet is trained on a large number(~ 5000) of rendered images covering almost all possible views, our method still outperforms it on most objects without additional training. We attribute this to PVNet's susceptibility to domain gaps and our method's robustness and generalizability, thanks to its design and the large-scale pre-training.

4.4 Results on LINEMOD

We compare the proposed method with generalizable methods OnePose [30], Gen6D [19], and instance-level methods PVNet [24] and CDPN [16] on *ADD(S)-0.1d* and *Proj2D* metrics. As shown in Tab. 3, our method outperforms existing generalizable methods significantly and achieves comparable performance with instance-level methods. Notably, our method and OnePose are only trained on the OnePose training set and tested on LINEMOD without additional training.

Since LINEMOD is mainly composed of low-textured objects, our method outperforms OnePose significantly thanks to the independence of image keypoints . Gen6D [19] is CAD-model-free and can generalize to unseen objects similar to our method. However, it relies on accurate and compact object detection for pose initialization, which is hard on LINEMOD because of the poor image quality and object occlusion. In contrast, our method only needs rough object detection to reduce possible mismatches. Moreover, the performance of Gen6D drops significantly without training on a subset of LINEMOD, while our method requires no extra training and achieves much higher accuracy than Gen6D. The experiment demonstrates the superiority of our method over existing generalizable CAD-model-free methods.

Our method has lower or comparable performance with instance-level methods [24, 16], which are trained to fit for each object instance, thus perform well naturally, at the expense of tedious generat-

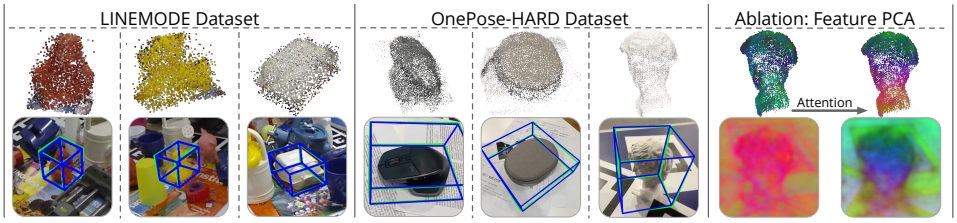

Figure 5: **Qualitative results** showing the reconstructed semi-dense object point clouds and the estimated object poses. The ablation part visualizes the 2D and 3D features before and after our 2D-3D attention module. Features become more discriminative as shown by the color contrast.

Table 4: **Ablation studies.** We quantitatively validate the effectiveness of our point cloud refinement in the keypoint-free SfM and the attention module in the 2D-3D matching network.

	Point cloud accuracy			Pose error on OnePose dataset				Pose error on OnePose-HARD		
	1mm	3mm	5mm	1cm-1deg	3mm-3deg	5mm-5deg		1cm-1deg	3cm-3deg	5cm-5deg
w/o refine.	25.6	68.9	84.1	45.5	78.6	86.0	w/o attention.	12.2	40.7	55.3
w refine.	**29.5**	**73.6**	**85.8**	**50.4**	**80.0**	**87.0**	w attention.	**16.3**	**55.4**	**70.3**

ing of synthetic data and separated training for each object. In contrast, our method is grounded in highly generalizable local features and generalizes to unseen objects with comparable performances.

4.5 Ablation Studies

We conduct several experiments on the OnePose dataset and the OnePose-HARD dataset to validate the efficacy of point cloud refinement and the attention module in our framework. More ablation studies are detailed in the supplementary.

Point cloud refinement in the keypoint-free SfM. We validate the effectiveness of our point cloud refinement from two perspectives, as shown in Tab.4(Left). One is evaluating the accuracy of the reconstructed point cloud with ground truth object mesh in OnePose-HARD following evaluations in [18], and the other is quantifying the impact of reconstruction accuracy on the final pose estimation results. Compared with the coarse point cloud reconstructed with coarse-level LoFTR only, the accuracy of our refined point cloud increased significantly, especially under a strict threshold (*1mm*). Moreover, using our refined point cloud for object pose estimation brings around 5% improvement on the strict *1cm-1deg* metric. These experiments demonstrate that our point cloud refinement improves the reconstructed point cloud's precision, thus benefiting the pose estimation.

Attention module in the pose estimation network. We validate the attention design in our 2D-3D matching network quantitatively and qualitatively. It is shown in Tab. 4(Right) that compared with directly matching the backbone features, using the transformed features for 2D-3D correspondences obtains 13% improvement on the *5cm-5deg* metric on OnePose-HARD . The visualization of features in Fig.5 shows that the transformed 2D and 3D features become more discriminative for establishing 2D-3D correspondences. The ablation demonstrates that the attention module provides the global receptive field and plays a critical role in the pose estimation of low-textured objects.

5 Conclusion

We propose a keypoint-free SfM and pose estimation pipeline that enables pose estimation of both texture-full and low-textured objects under the one-shot CAD-model-free setting. Our method can efficiently reconstruct accurate and complete 3D structures of low-textured objects, and build robust 2D-3D correspondences with the test image for accurate object pose estimation. The experiments show that our method achieves significantly better pose estimation accuracy compared with existing CAD-model-free methods, and even achieves comparable results with CAD-model-based instance-level methods. Lastly, although we do not see the immediate negative societal impact of our work, we do note that accurate object pose estimation can be potentially used for malicious purposes.

Limitations. Being dependent on local feature matching, our method inherently suffers from very low-resolution images and extreme scale and viewpoint changes. In the current pipeline, we still need a separated object detector to provide rough regions of interest. In the future, we envision a more tight integration with the object detector, where object detection can also be carried out through local feature matching.

References

[1] Ultralytics/yolov5. https://github.com/ultralytics/yolov5, 2021. 7

[2] Ming Cai and Ian Reid. Reconstruct locally, localize globally: A model free method for object pose estimation. In *CVPR*, 2020. 1, 3

[3] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. *ECCV*, 2020. 1, 3

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. *CVPRW*, 2017. 3

[5] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014. 5

[6] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. *2019 International Conference on 3D Vision (3DV)*, pages 513–523, 2019. 2

[7] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[8] Stefan Hinterstoißer, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary R. Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, 2012. 2, 7

[9] Stefan Hinterstoißer, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary R. Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*. 2012. 11

[10] Jingwei Huang, Shan Huang, and Mingwei Sun. Deeplm: Large-scale nonlinear least squares on deep learning frameworks using stochastic domain decomposition. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10303–10312, 2021. 6

[11] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Franccois Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. *ArXiv*, abs/2006.16236, 2020. 5

[12] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017. 3

[13] Taeyeop Lee, Byeong-Uk Lee, Myungchul Kim, and In So Kweon. Category-Level metric scale object shape and pose estimation. *IEEE Robotics and Automation Letters*, 2021. 3

[14] 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. 3

[15] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 6

[16] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7677–7686, 2019. 1, 2, 3, 7, 8

[17] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 6

[18] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5967–5977, 2021. 3, 9

[19] Yuan Liu, Yilin Wen, Sida Peng, Chu-Hsing Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. *ArXiv*, abs/2204.10776, 2022. 2, 3, 7, 8

[20] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3

[21] 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. In *ECCV*, 2020. 3

[22] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *CVPR*, 2020. 1

[23] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019. 1

[24] Sida Peng, Xiaowei Zhou, Yuan Liu, Haotong Lin, Qixing Huang, and Hujun Bao. PVNet: pixel-wise voting network for 6dof object pose estimation. *T-PAMI*, 2020. 1, 2, 3, 7, 8

[25] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 3, 7, 8

[26] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *ICCV*, 2020. 7

[27] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3, 4, 5, 6

[28] Ivan S. Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. Osop: A multi-stage one shot object pose estimation framework. *ArXiv*, abs/2203.15533, 2022. 3

[29] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 2, 3, 4, 5, 6, 7

[30] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. OnePose: One-shot object pose estimation without CAD models. *CVPR*, 2022. 1, 2, 3, 7, 8, 11

[31] Meng Tian, Marcelo H. Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6D object pose and size estimation. *ECCV*, 2020. 3
[32] Michal J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. In *NeurIPS*, 2020. 5
[33] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In *CVPR*, 2021. 3
[34] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 1, 3
[35] Jiaze Wang, Kai Chen, and Qi Dou. Category-level 6D object pose estimation via cascaded relation and recurrent reconstruction networks. *IROS*, 2021. 3
[36] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. *RSS*, 2018. 3
[37] Sergey Zakharov, Ivan S. Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1941–1950, 2019. 3
[38] Chengliang Zhong, Chao Yang, Ji Qi, Fuchun Sun, Huaping Liu, Xiaodong Mu, and Wenbing Huang. Sim2real object-centric keypoint detection and description. *ArXiv*, abs/2202.00448, 2022. 3

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The OnePose [30] dataset and the LINEMOD [9] dataset used in the paper are public; our code and the collected OnePose-HARD dataset will be published.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec.3.4 and Sec.4.1
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Unfortunately, we did not provide the error bars due to resource constraints. We will make this up in the future.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Sec.3.4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]