OnePose: One-Shot Object Pose Estimation without CAD Models

Jiaming Sun[1,2*] Zihao Wang[1*] Siyu Zhang[2*] Xingyi He[1] Hongcheng Zhao[3]
Guofeng Zhang[1] Xiaowei Zhou[1†]

[1]Zhejiang University [2]SenseTime Research [3]TUM

Abstract

We propose a new method named OnePose for object pose estimation. Unlike existing instance-level or category-level methods, OnePose does not rely on CAD models and can handle objects in arbitrary categories without instance- or category-specific network training. OnePose draws the idea from visual localization and only requires a simple RGB video scan of the object to build a sparse SfM model of the object. Then, this model is registered to new query images with a generic feature matching network. To mitigate the slow runtime of existing visual localization methods, we propose a new graph attention network that directly matches 2D interest points in the query image with the 3D points in the SfM model, resulting in efficient and robust pose estimation. Combined with a feature-based pose tracker, OnePose is able to stably detect and track 6D poses of everyday household objects in real-time. We also collected a large-scale dataset that consists of 450 sequences of 150 objects. Code and data are available at the project page: https://zju3dv.github.io/onepose/.

1. Introduction

Object pose estimation plays an important role in augmented reality (AR). The ultimate goal of object pose estimation in AR is to use arbitrary objects as "virtual anchors" of AR effects, which demands the ability to estimate poses of surrounding objects in our daily life. Most established works in object pose estimation [16, 26, 46] assume that the CAD model of the object is known *a priori*. Since high-quality CAD models of everyday objects are often inaccessible, the research on object pose estimation for AR scenarios necessitates new problem settings.

To not rely on instance-level CAD models, many recent methods have been working on category-level pose estimation [4, 43]. By training a network on different instances in the same category, the network can learn a category-level representation of object appearances and shapes and thus be able to generalize to new instances in the same category. However, such approaches require a large number of training samples in the same category, which can be hard to obtain and annotate. Furthermore, the generalization capabilities of category-level methods are not guaranteed when a new instance has a significantly different appearance or shape. More importantly, training and deploying a network for each category are unaffordable in many real world applications, e.g., mobile AR, when the number of object categories to be handled is huge.

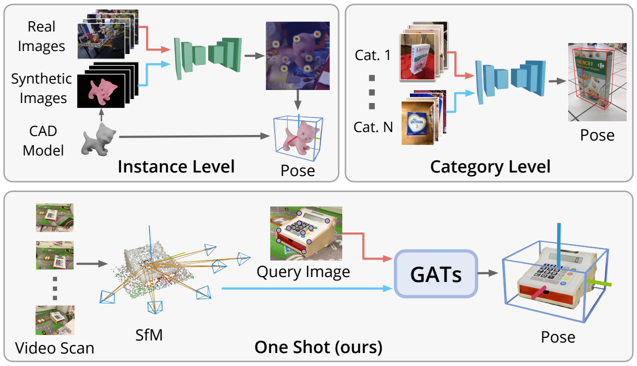

Figure 1. **Comparison of different problem settings** of instance/category-level object pose estimation and the one-shot pose estimation proposed in this work. Unlike previous works that rely on instance- or category-specific network training, the proposed approach only requires a simple video scan of the object to build a sparse SfM model of the object and uses a generic 3D-2D feature matching network (GATs) to estimate its pose, without CAD models or additional network training.

To alleviate the demand for CAD models or category-specific training, we go back to an "old" problem setting for object pose estimation, but renovate the entire pipeline with a new learning-based approach. Similar to the task of visual localization, which estimates the unknown camera pose given an SfM map of a scene, object pose estimation has long been formulated in the localization-based setting [21, 35]. Different from instance- or category-level

[*] Equal contribution. The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG and the ZJU-SenseTime Joint Lab of 3D Vision. [†]Corresponding author: Xiaowei Zhou.

methods, this setting assumes that a video sequence of the object is given, and a sparse point cloud model can be reconstructed from the sequence. Estimating the object pose is then equivalent to localizing the camera pose with respect to the reconstructed point cloud model. At test time, 2D local features are extracted from the query image and matched with the points in the SfM model to obtain 2D-3D correspondences, from which the object pose can be solved by PnP. Instead of learning instance- or category-specific representations by neural networks, this traditional pipeline leverages an explicit 3D model of the object that can be built on-the-fly for a new instance, which brings better generalization capabilities to arbitrary objects while making the system more explainable.

In this paper, we refer to this problem setting as *one-shot object pose estimation*, where the objective is being able to estimate 6D pose of an object in arbitrary category, given only a few pose-annotated images of the object for training. While this problem is similar to visual localization, directly migrating existing visual localization methods does not solve this problem. The modern visual localization pipeline [31] produces 2D-3D correspondences by first performing a 2D-2D matching between the query image and the retrieved database images. To ensure a high success rate of localization, matching to multiple image retrieval candidates is necessary, so that the 2D-2D matching can be expensive especially for learning-based matchers [32, 36]. As a result, the runtime of existing visual localization methods is often seconds and cannot satisfy the requirement to track moving objects in real-time.

For the reasons above, we propose to directly perform 2D-3D matching between the query image and the SfM point clouds. Our key idea is to use graph attention networks (GATs) [40] to aggregate the 2D features that correspond to the same 3D SfM point (i.e., a feature track) to form a 3D feature. The aggregated 3D features are later matched with 2D features in the query images with self- and cross-attention layers. Together with the self- and cross-attention layers, the GATs can capture the globally-consented and context-dependent matching priors exhibited in ground-truth 2D-3D correspondences, making the matching more accurate and robust.

To evaluate the proposed method, we collected a large-scale dataset for the one-shot pose estimation setting, which contains 450 sequences of 150 objects. Compared with previous instance-level method PVNet [27] and category-level method Objectron [4], OnePose achieves better precision without training for any object instances or categories in the validation set, while taking only 58 *ms* to process one frame on GPU. To the best of our knowledge, when combined with a feature-based pose tracker, OnePose is the first learning-based method that can stably detect and track poses of everyday household objects in real-time (refer to the project page).

Contributions.

- Renovating the visual localization pipeline for object pose estimation that can handle novel objects without CAD models or additional network training.
- A new architecture of graph attention networks for robust 2D-3D feature matching.
- A large-scale object dataset for one-shot object pose estimation with pose annotations.

2. Related works

CAD-Model-Based Object Pose Estimation. The state-of-the-art approaches for the object 6DoF pose estimation can be broadly categorized into regression and keypoint techniques. Given an image, the first type of methods [15,16,18,46] directly regress pose parameters with features within each Region of Interest (RoI). In contrast, the latter type of methods first find correspondences between image pixels and 3D object coordinates either by regression [22, 24, 25] or by voting [26, 27], and then compute the pose with Perspective-n-Points (PnP). These methods require high fidelity textured 3D models to generate auxiliary synthetic training data and for pose refinements [16,18] to achieve high accuracy on trained instances.

Unlike the abovementioned methods that train a single network for each instance, NOCS [43] proposes to establish correspondences between pixels on the image and Normalized Object Coordinates (NOCS) shared within each category. With this learned category-level shape prior, NOCS can eliminate the dependencies on CAD models during test time. Some later works [17, 38, 38, 42, 44] follow the trend of leveraging category-level prior to further recover a more accurate shape of the object with NOCS representation. A limitation of this line of work is that the shape and the appearance of some instances could vary significantly even they belong to the same category, thus the generalization capabilities of trained networks over these instances are questionable. Moreover, accurate CAD models are still required for ground-truth NOCS map generation during training, and different networks need to be trained for different categories. Our proposed method does not require CAD models both for training and test time and is category-agnostic.

CAD-Model-Free Object Pose Estimation. Recently, a few attempts have been made to achieve CAD-model-free object 6D pose estimation both at the training and test time. Both Neural Object Fitting [8] and LatentFusion [23] tackled the problem via analysis-by-synthesis approaches where differentiablly synthesized images are compared with target images to generate gradients for the object pose optimization. Neural Object Fitting [8] proposes to encode category-level appereance prior with a Variational Auto Encoder (VAE) trained with fully synthetic data, while LatentFusion [23] builds a 3D latent space based object representation with posed RGB-D images for each unseen object. However, the efficiency and accuracy of such methods are highly limited by image synthesizing networks and are not suitable for AR applications. RLLG [7] takes a different approach and learns correspondences from image pixels to object coordinates [5] without CAD models. Although RLLG can achieve comparable precision to its counterparts [24], it works only on the instance level and requires highly accurate instance masks to segment foreground pixels. Most recently, Objectron [4] proposes a data-driven approach that learns to regress pixel coordinates of projected box corners for each category with a tremendous amount of annotated training data. Such an approach is costly and only limited to a few categories as the learned model is category-specific. Moreover, it can only obtain up-to-scale poses without metric scales since it uses a single-view image as input. On the contrary, our method can leverage the visual-inertial odometry to recover metric scales during the mapping stage, thus being able to recover metric 6D poses at test time.

Feature-Matching-Based Pose Estimation. Visual localization pipelines based on feature-matching have long been studied. Traditionally, the localization problem is solved by finding 2D-3D correspondences between input RGB images and a 3D model from SfM with hand-crafted local features like SIFT [20], FAST [29] and ORB [30]. Recently, learning-based local feature detection, description [10–12, 39] and matching [32, 36] surpass these hand-crafted methods and have substituted the traditional counterparts in the localization pipeline. Notably, Hierarchical Localization (HLoc) [31] provides a complete toolbox for running SfM with COLMAP [33] and feature extraction and matching with SuperGlue [32]. Our method is inspired by SuperGlue in terms of using self- and cross-attention layers for feature matching. However, SuperGlue only focuses on 2D-2D matching between images and does not consider the graph structure of the SfM map. Our method uses graph attention networks [40] to process and aggregate 2D features that correspond to a 3D SfM point (i.e., a feature track), which preserves the graph structure of the SfM during 2D-3D matching.

Many traditional methods for object recognition and pose estimation also share the feature-based pipeline similar to visual localization. These methods first build object models by reconstructing sparse point clouds from matched keypoints across the views [9, 21, 35, 37], and localize with the sparse point cloud model given a query image. Some approaches [28, 45] propose to build a point cloud model online with a framework similar to Simultaneous Localization and Mapping (SLAM). Notably, BundleTrack [45] proposes an online pose tracking pipeline without instance- or category-level models, which resembles ours mostly. However, it uses 2D-2D feature matching instead of 2D-3D as in ours. To recover the 3D information, it also takes depth map as input which could limit its usage in AR.

3. Method

An overview of the proposed method is presented in Fig. 2. In the setting of one-shot object pose estimation introduced in Sec. 1, a video scan surrounding the object is captured with a mobile device (e.g. iPhone or iPad). Given the video scan and a test image sequence $\{\mathbf{I}_q\}$, the objective of one-shot object pose estimation is to estimate the object poses $\{\xi_q\} \in \mathbb{SE}(3)$ defined in the camera coordinate system, where q is the key-frame index in the video. We use bold letters (e.g. $\mathbf{I}$) to denote tensors, calligraphy letters (e.g. $\mathcal{G}$) to denote graphs and $\{\cdot\}$ to denote a set of these entities.

3.1. Preliminaries

Data Capture and Annotation. During the data capture, the object is assumed to be set on a flat surface and remains static during the capture. To define the canonical pose of the object, an object bounding box $\mathbf{B}$ is annotated in AR, with the camera poses $\{\xi_i\} \in \mathbb{SE}(3)$ tracked by off-the-self AR toolboxes like ARKit [2] or ARCore [1]. i is the frame index. The capture interface is shown in Fig. 4. $\mathbf{B}$ is parameterized by the center location, dimensions and rotation around the z-axis (yaw angle). After the data capture and annotation, the pipeline of OnePose can be separated into the offline mapping phase and the online localization phase.

Structure from Motion. In the mapping phase, given a set of images $\{\mathbf{I}_i\}$ extracted from the video scan, we use Structure from Motion (SfM) to reconstruct the sparse point cloud $\{\mathbf{P}_j\}$ of the object, where j is the point index. Since $\mathbf{B}$ is annotated, $\{\mathbf{P}_j\}$ can be defined in the object coordinate system. A visualization of all correspondence graphs of the object $\{\mathcal{G}_j\}$ can be found in Fig. 2 (**3**,**4**). Specifically, 2D keypoints and descriptors are first extracted from each image and matched between images to produce 2D-2D correspondences. Every reconstructed point $\mathbf{P}_j$ corresponds to a set of matched 2D keypoints and descriptors $\{\mathbf{F}_k^{2D}\} \in \mathbb{R}^d$ where k is the keypoint index and d is the dimension of the descriptor. The correspondence graphs $\{\mathcal{G}_j\}$, which are also called the feature tracks, are formed by keypoint indexes for $\{\mathbf{P}_j^{3D}\}$ as visualized in Fig. 2 (**3**,**4**).

Pose Estimation through Visual Localization. In the localization phase, a sequence of query images $\{\mathbf{I}_q\}$ are captured in real-time. Localizing the camera poses of the query images $\{\xi_q^{-1}\}$ with respect to $\{\mathbf{P}_j\}$ produces the object poses $\{\xi_q\}$ defined in the camera coordinate.

For each $\mathbf{I}_q$, 2D keypoints and descriptors $\{\mathbf{F}_q^{2D}\} \in \mathbb{R}^d$ are extracted and used for matching. In modern visual lo-

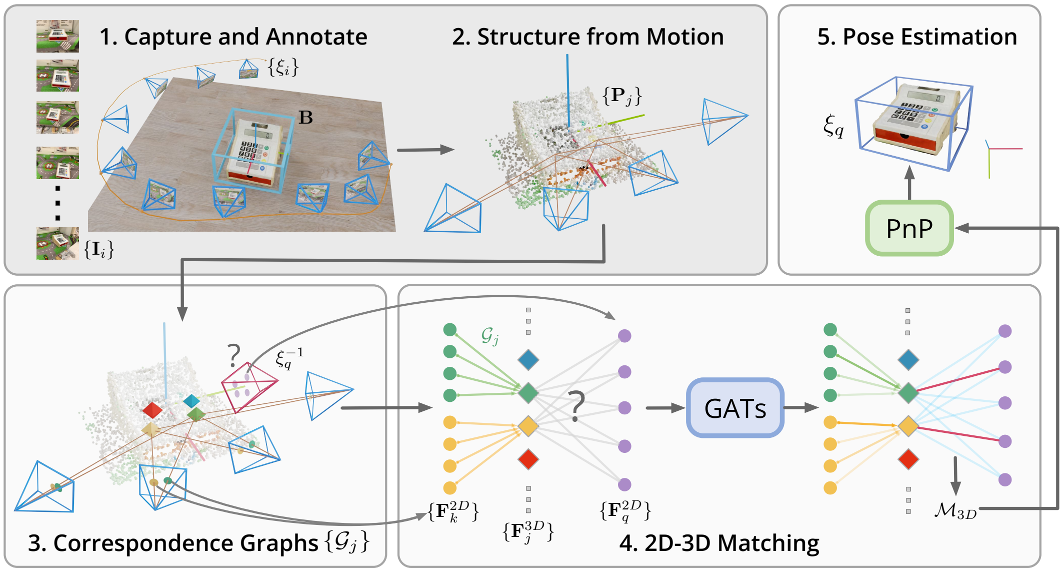

Figure 2. **Overview of OnePose. 1.** For each object, a video scan with RGB frames $\{\mathbf{I}_i\}$ and camera poses $\{\xi_i\}$ are collected together with the annotated 3D object bounding box $\mathbf{B}$. **2.** Structure from Motion (SfM) reconstructs a sparse point cloud $\{\mathbf{P}_j\}$ of the object. **3.** The correspondence graphs $\{\mathcal{G}_j\}$ (▷◆,▷◆) are built during SfM, which represent the 2D-3D correspondences in the SfM map. ◇ represents the camera to be localized in the object frame. **4.** 2D descriptors $\{\mathbf{F}_k^{2D}\}$ (●,●) are aggregated to 3D descriptors $\{\mathbf{F}_j^{3D}\}$ (◆◆◆) with the aggregration-attention layer. $\{\mathbf{F}_j^{3D}\}$ are later matched with 2D descriptors from the query image $\{\mathbf{F}_q^{2D}\}$ (●) to generate 2D-3D match predictions $\mathcal{M}_{3D}$. **5.** Finally, the object pose ξ_q is computed by solving the PnP problem with $\mathcal{M}_{3D}$. Grey background color denotes offline processes. Best viewed in color.

calization pipeline [31], an image retrieval network is used to extract image-level global features, which can be used to retrieve the image candidates from the SfM database for 2D-2D matching. Increasing the number of image pairs to be matched will significantly slow down the localization, especially for learning-based matchers like SuperGlue [32] or LoFTR [36]. Reducing the number of images retrieved can result in a low localization success rate and thus a trade-off has to be made between runtime and pose estimation accuracy.

To remedy this problem, we propose to directly perform 2D-3D matching between the query image and the SfM point clouds. Direct 2D-3D matching avoids the need of the image retrieval module, and thus can maintain localization accuracy while being fast. In the next section, we describe how to obtain the 2D-3D correspondences $\mathcal{M}_{3D}$.

3.2. OnePose

Graph Attention Networks (GATs) for 2D-3D Matching. Direct 2D-3D matching requires 3D feature descriptors. Since a 3D point $\mathbf{P}_j$ is associated with multiple $\mathbf{F}_{k'}^{2D}$ in $\mathcal{G}_j$, an aggregation operation is needed to update the 3D descriptors, defined as $\{\mathbf{F}_j^{3D}\} \in \mathbb{R}^d$ which are initialized by averaging the corresponding 2D descriptors. The aggregation operation could cause information loss since it reduces multiple descriptors to one. An ideal aggregation operation should be able to adaptively preserve the most informative 2D features in $\{\mathbf{F}_k^{2D}\}$ for the 2D-3D matching according to different $\mathbf{F}_q^{2D}$.

We propose to use the graph attention layer in [40] to achieve the adaptive aggregation. We name it the aggregation-attention layer. The aggregation-attention layer operates on each individual $\mathcal{G}_j$. For every $\mathcal{G}_j$, denoting the weight matrix as $\mathbf{W} \in \mathbb{R}^{d\times d}$, the aggregation-attention layer is defined as:

$$\mathrm{Aggr}(\{\mathbf{F}_{k'}^{2D}\}, \mathbf{F}_j^{3D}) = \mathbf{F}_j^{3D} + \sum_{\forall k' \in \mathcal{G}_j} \alpha_{k'} \mathbf{F}_{k'}^{2D},$$
$$\alpha_{k'} = \underset{\forall k' \in \mathcal{G}_j}{\mathrm{softmax}}(\mathrm{sim}(\mathbf{W}\cdot\mathbf{F}_{k'}^{2D}, \mathbf{W}\cdot\mathbf{F}_j^{3D}))$$

with $\mathrm{sim}(\cdot,\cdot) = \langle \mathbb{R}^d, \mathbb{R}^d \rangle \in \mathbb{R}$ computes the attention coefficient, which measures the importance of the descriptors in the aggregation operation.

Inspired by [32, 36], we further use self- and cross-attention layers following the aggregation-attention layers to process and transform the aggregated 3D descriptors and query 2D descriptors. A set of aggregation-, self- and cross-attention layers forms an *attention group*, specifically:

$$\begin{cases} \{\hat{\mathbf{F}}_j^{3D}\} = \mathrm{Aggr}(\{\mathbf{F}_k^{2D}\}, \{\mathbf{F}_j^{3D}\}), \\ \{\tilde{\mathbf{F}}_q^{2D}\} = \mathrm{Self}(\{\mathbf{F}_q^{2D}\}, \{\mathbf{F}_q^{2D}\}), \\ \{\tilde{\mathbf{F}}_j^{3D}\} = \mathrm{Self}(\{\hat{\mathbf{F}}_j^{3D}\}, \{\hat{\mathbf{F}}_j^{3D}\}), \\ \{\mathbf{F'}_q^{2D}\}, \{\mathbf{F'}_j^{3D}\} = \mathrm{Cross}(\{\tilde{\mathbf{F}}_q^{2D}\}, \{\tilde{\mathbf{F}}_j^{3D}\}). \end{cases}$$

The proposed architecture of graph attention networks (GATs) is composed of N stacked attention groups. Intuitively, the aggregation-attention layers will adaptively attend to different $\mathbf{F}_k^{2D}$ in $\mathcal{G}_j$ according to its relevance with

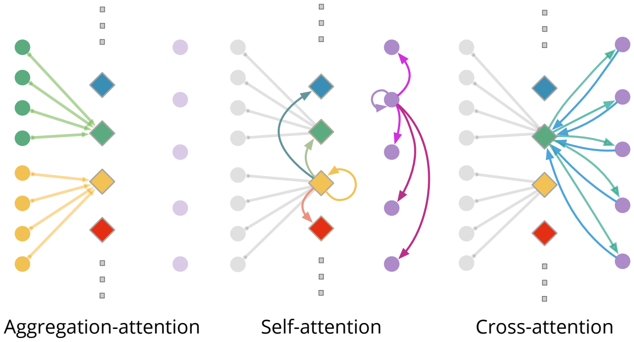

Figure 3. **Different types of attention layers in GATs.** $\{\mathbf{F}_j^{3D}\}$: ◆◆◆, $\{\mathbf{F}_q^{2D}\}$: ●, $\{\mathcal{G}_j\}$: (▷◆,▷◆) $\{\mathbf{F}_k^{2D}\}$: (●,●). For clarity, the complete relations of attentions in the figure are not shown.

$\mathbf{F}_q^{2D}$, thus preserving more descriminative information for 2D-3D matching. By interleaving the aggregation-attention layers with self- and cross-attention layers, $\{\mathbf{F}_k^{2D}\}$, $\{\mathbf{F}_j^{3D}\}$, $\{\mathbf{F}_q^{2D}\}$ can exchange information with each other, thus making the matching globally-consented and context-dependent.

Match Selection and Pose Calculation. We follow [36] to use the dual-softmax operator to differentiablly extract match confidence scores $\mathcal{P}_{3D}$. The score matrix $\mathbf{S}$ between the transformed features is first calculated by $\mathbf{S}\left(q,j\right) = \langle \mathbf{F'}_q^{2D}, \mathbf{F'}_j^{3D} \rangle$. Formally, the matching confidence $\mathbf{C}_{3D}$ is obtained by:

$$\mathbf{C}_{3D}(q,j) = \operatorname{softmax}\left(\mathbf{S}\left(q,\cdot\right)\right)_j \cdot \operatorname{softmax}\left(\mathbf{S}\left(\cdot,j\right)\right)_q .$$

After selecting a confidence threshold θ, $\mathbf{C}_{3D}$ becomes a permutation matrix $\mathcal{M}_{3D}$, which represents the 2D-3D match predictions. With $\mathcal{M}_{3D}$, the object pose in the camera coordinate ξ_q can be computed by the Perspective-n-Point (PnP) algorithm with RANSAC.

Supervision. The supervision signal $\mathcal{M}_{3D}^{gt}$ can be directly obtained from filtered 2D-3D correspondences in the SfM maps in the training set. The loss function L is the focal loss [19] over the confidence scores $\mathbf{C}_{3D}$ returned by the dual-softmax operator:

$$L = -\left(1 - \mathbf{C}'_{3D}\left(q,j\right)\right)^{\gamma} \log \mathbf{C}'_{3D}\left(q,j\right),$$

$$\begin{cases} \mathbf{C}'_{3D}(q,j) = \mathbf{C}_{3D}(q,j) & \text{if } \mathcal{M}_{3D}^{gt}(q,j) = 1 \\ \mathbf{C}'_{3D}(q,j) = 1 - \mathbf{C}_{3D}(q,j) & \text{if } \mathcal{M}_{3D}^{gt}(q,j) \neq 1. \end{cases}$$

Online Feature-based Pose Tracking. The above-mentioned pose estimation module takes only sparse key-frame images as input. To obtain stable object poses for AR applications, we further equip OnePose with a feature-based pose tracking module, which processes *every* frame in the test sequence. Similar to a SLAM system, the pose tracking module reconstructs a 3D map online and maintains its own key-frame pool. At each time-step, tracking adopts a tightly-coupled approach and relies on both the prebuilt SfM map and the online-built 3D map to find 2D-3D correspondences and solve for 6D poses. Since the pose tracking module preserves 2D and 3D information of the test sequence in the online-built map, it can be more stable than the single-frame-based pose estimation module. The pose estimation module helps to recover and re-initialize the tracking module when it fails. We provide more details about the pose tracking module in the supplementary material.

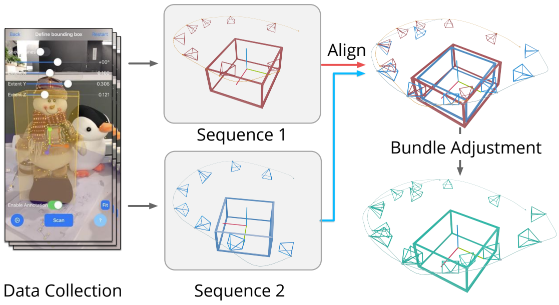

Figure 4. **Dataset collection and sequence registration by joint bundle adjustment.** The AR-based dataset collection and annotation interface are shown on the left. Multiple collected sequences of the same object are aligned according to the bounding box annotations. The final camera poses are optimized by bundle adjustment with the camera poses from ARKit as prior.

Remarks on the One-Shot Setting. Other than not using CAD models or additional network training, the one-shot setting of OnePose has many advantages compared with existing instance- or category-level pose estimation methods. During the mapping phase, OnePose takes as input a simple video scan of an object and builds an instance-specific 3D representation of the object geometry. Similar to the role of CAD models in instance-level methods, the 3D geometry of the object is crucial for recovering object poses with metric scales. In the localization phase, learned local feature matching in OnePose can handle large changes in viewpoint, lighting and scale, making the system more stable and robust compared to category-level methods. The local-feature-based pipeline also allows the pose estimation module to be naturally coupled with a feature-based tracking module to realize efficient and stable pose tracking.

3.3. OnePose Dataset

Since there is no existing large-scale dataset that can fit the setting of one-shot pose estimation, we collected a dataset with multiple video scans of the same object put in different locations. The OnePose dataset contains over 450 video sequences of 150 objects. For each object, multiple video recordings, accompanied camera poses and 3D bounding box annotations are provided. These sequences are collected under different background environments, and each has an average recording-length of 30 seconds covering all views of the object. The dataset is randomly divided into training and validation sets. For each object in the validation set, we assign one mapping sequence for building the SfM map, and use a test sequence for the evaluation.

To reduce the manual labor of data annotation, we propose a semi-automatic approach to simultaneously collect and annotate the data in AR. To be specific, an adjustable 3D bounding box is rendered onto the image in AR, as shown in Fig. 4. The only manual work is to adjust the rotation and rough dimensions of the 3D bounding box. Visualizations of the data capture interface and the post-processing process are shown in Fig. 4.

The objective of the post-processing is to reduce the pose drift error of ARKit for each sequence and ensure consistent pose annotations across sequences. To achieve this, we first align sequences with the annotated bounding boxes and perform bundle adjustment (BA) with COLMAP [31, 33]. Feature matches used in the BA are extracted with SuperGlue. As the backgrounds are different between sequences, we extract matches only in the foreground (i.e., within the 2D object bounding boxes) between all matchable pairs of images. For more details about our data collection and processing pipeline, please refer to our supplementary material.

4. Experiments

In this section, we first introduce our selection of baseline methods and evaluation protocols, as well as evaluation metrics on our proposed OnePose dataset in Sec. 4.1, followed by implementation details of our method in Sec. 4.2. Experimental results and ablation studies are detailed in Sec. 4.3 and Sec. 4.4, respectively.

4.1. Experiment Settings and Baselines

Baselines. We compare our method with the following baseline methods in three categories: 1) *Visual Localization* methods are most relevant to the proposed method in terms of estimating the pose based on local feature matching. To be specific, we compare our method with HLoc [31] using different keypoint descriptors (SIFT [20] and SuperPoint [10]), as well as matchers (Nearest Neighbour, SuperGlue [32]). 2) *Instance-level* method PVNet [26, 27]. 3) *Category-level* method Objectron [4]. To the best of our knowledge, Objectron is the only method for category-level object pose estimation with RGB image as input.

Evaluation Protocols. We apply per-frame pose estimation with the proposed method without the pose tracking module for a fair comparison in all the experiments. For our *Visual Localization* baselines and the proposed method, we use the same video scan to build the SfM map for the localization. Note that the original image retrieval module used for large scale scenes does not generalize well to objects, thus we equally sample a subset of five images with equal intervals from database images as retrieved images for feature matching. To train our *instance-level* baseline PVNet, we use 3D box corners instead of sampled semantic points from CAD models as keypoints to vote for, and further supply auxiliary mask supervision which is indispensable for training PVNet. Due to the data demanding nature of the *category-level* baseline Objectron [4], we directly use the models provided by the authors, which are trained on the original Objectron dataset.

Metrics. For evaluation metrics, we cannot directly adopt the commonly used ADD metric [13] and 2D projection metric [6] since CAD models are unavailable in our setting. Another commonly used metric for evaluating the quality of predicted object pose is the *5cm-5deg* metric proposed in [34] which deems a predicted pose as correct if the error is below 5*cm* and 5°. We further narrow down the criteria to *1cm-1deg* and *3cm-3deg* following a similar definition to set up more strict metrics for the pose estimation in augmented reality application. We divide the objects to three splits by their diameters with 40 *cm* and 25 *cm* as thresholds. When comparing with instance-level baseline and category-level baseline, we follow the metrics used in the original paper.

4.2. Implementation Details

During the mapping phase, to maintain a fast mapping speed, we reuse $\{\xi_i\}$ and use triangulation to reconstruct the point cloud, without further optimization on the camera poses by bundle adjustment. During the localization phase, we assume the 2D bounding box of the object is known, which can be easily obtained from an off-the-shelf 2D object detector (e.g. YOLOv5 [3]) in practice. To reduce possible mismatches in pose estimation, only the 3D points inside the annotated 3D bounding box are preserved during mapping, and only the 2D features inside the detected 2D bounding box are preserved during localization. For the network design, we use $N = 4$ attention groups in GATs. Linear Attention [14, 41] is used in all the attention layers following [36]. As the input of GATs, we randomly sample or pad a set of eight features from $\{\mathbf{F}_i^{2D}\}$ associated with each $\mathbf{F}_i^{3D}$ for all experiments in the paper. The $\{\mathbf{F}_i^{3D}\}$ are initialized by averaging all of the associated features $\{\mathbf{F}_i^{2D}\}$.

4.3. Evaluation Results

Comparison with Visual Localization Baselines. We compare our approach with visual localization baselines with different feature extractors and matchers, and present the results in Tab. 1. HLoc *(SPP + SPG)* is the baseline with learning-based feature extractor (SuperPoint) and matcher (SuperGlue), which mostly resembles our method among all the three variants. Our method performs on-par or slightly better compared with HLoc *(SPP + SPG)*,

	Large Objects			Medium Objects			Small Objects			Time (*ms*)
	1cm-1deg	3cm-3deg	5cm-5deg	1cm-1deg	3cm-3deg	5cm-5deg	1cm-1deg	3cm-3deg	5cm-5deg	
HLoc *(SIFT + NN)*	0.314	0.572	0.572	0.432	0.575	0.608	0.248	0.468	0.515	116.27
HLoc *(SPP + NN)*	0.357	0.675	0.675	0.508	0.659	0.706	0.342	0.612	0.687	136.98
HLoc *(SPP + SPG)*	0.435	0.813	0.813	**0.643**	0.793	0.831	**0.432**	**0.739**	**0.837**	618.29
Ours	**0.471**	**0.856**	**0.856**	0.629	**0.816**	**0.858**	0.405	0.729	0.832	**58.31**

Table 1. **Comparison with the *Visual Localization* baselines.** Our method are compared with HLoc [31] with different detectors including *SIFT* [20] and SuperPoint (*SPP*) [10], and matchers including Nearest Neighbor (*NN*) and SuperGlue (*SPG*) [32].

Obj. ID	0447	0450	0488	0493	0494	0524	0594
PVNet	0.253	0.127	0.042	0.094	0.192	0.119	0.077
Ours	**0.900**	**0.981**	**0.740**	**0.873**	**0.819**	**0.679**	**0.789**

Table 2. **Comparison with the *instance-level* baseline.** Our method is compared with PVNet [26] on selected objects from the OnePose dataset with the *5cm-5deg* metric.

while HLoc *(SPP + SPG)* takes ten times the runtime of our method. We believe the improvement comes from the ability of our method to selectively aggregate context from multiple images benefited from our GATs design, instead of only focusing on the two images being matched.

Comparison with the Instance-level Baseline PVNet. The proposed method is compared with PVNet [26] with *5cm-5deg* on selected objects from our OnePose dataset and the results are as presented in Tab. 2. To obtain segmentation masks for training PVNet, we need to additionally apply dense 3D reconstruction and render the reconstructed meshes to obtain masks on the data sequences. This process is time-consuming and greatly limits our choices for objects because of the quality of 3D reconstruction. Our method achieves much higher precision than PVNet, which demonstrates the superiority of our method. PVNet relies on memorizing the mapping from image patches to object-specific keypoints. Without pre-training on large-scale synthetic images (rendered with CAD models) that densely cover all possible views, the performance of PVNet drops drastically. Conversely, our method is able to leverage the learned local features that are relatively viewpoint-invariant and thus generalize to unseen views while maintaining the precision.

Comparison with the Category-level Baseline Objectron. We compare our method with Objectron [4] on all objects in the *Shoe* and *Cup* categories with the metrics used in the original paper and present the results in Tab. 3. For mean pixel error of 2D projection, the results of Objectron on our dataset are far from the reported results for the two categories on Objectron dataset. This is because of the deviations in ground-truth annotations between the Objectron dataset and our dataset. For a fair comparison, we further apply scaling and center alignment operations to the predictions of Objectron to alleviate this gap and provided results respectively as *Objectron (S)* and *Objectron (S+C)* in Tab. 3. Although the performances of Objectron do get boosted and are comparable with the reported results in the original paper, our method surpasses it by a large margin. Our method outperforms Objectron evidently in the average precision of azimuth error and elevation error, especially for the objects of *Cup* category where the shape and appearance may vary significantly between instances. These experiments illustrate the limited generalization ability of category-level methods to new object instances.

Obj. ID	0415	0475	0476	Cup	0592	0593	0594	0595	Shoe
Mean pixel error of 2D projection									
Objectron	0.269	0.474	0.483	0.054	0.189	0.183	0.118	0.124	0.039
Objectron (S)	0.170	0.340	0.347	-	0.123	0.115	0.092	0.089	-
Objectron (S+C)	0.158	0.331	0.342	-	0.103	0.098	0.084	0.079	-
Ours	**0.047**	**0.022**	**0.013**	-	**0.089**	**0.016**	**0.026**	**0.012**	-
Average precision at 15 ° Azimuth error									
Objectron	0.364	0.131	0.217	0.644	0.677	0.733	0.774	0.945	0.586
Ours	**1.0**	**1.0**	**1.0**	-	**0.855**	**0.998**	**0.984**	**1.0**	-
Average precision at 10 ° Elevation error									
Objectron	0.707	0.906	0.821	0.837	0.794	0.842	0.622	0.995	0.754
Ours	**1.0**	**1.0**	**1.0**	-	**0.831**	**0.996**	**0.973**	**1.0**	-

Table 3. **Comparison with the *category-level* baseline.** Our method is compared with Objectron [4] with auxiliary scale adjustment *(S)* and center alignment *(S+C)*. The category-level evaluation results reported in the original paper are provided in grey background below the name of the category.

Runtime Analysis. We report the runtimes of our visual localization baselines and our method in Tab. 1. The runtime consist of feature extraction for the query image with SuperPoint and the 2D-3D matching process without 2D detection and PnP. Our method runs ∼10× faster than HLoc (*SPP* + *SPG*). All the experiments are conducted on an NVIDIA TITAN RTX GPU.

4.4. Ablation Studies

In this section. we conduct several ablation experiments by substituting GATs with simpler counterparts of the feature aggregation and matching modules. All the results for our ablation studies are presented in Tab. 4.

Effectiveness of the Aggregation-Attention Layer. We validate the effectiveness of the proposed aggregation-attention layers by substituting the corresponding aggregation layers in GATs by the averaging operation and report the result in Tab. 4 as (i). Notice the 2D-3D matching is still based on a GNN with self- and cross-attention layers, which is similar to SuperGlue. Without the aggregation-attention layers, the results dropped significantly for large and medium objects, which indicates the effectiveness of aggregation-attention layers. The simple averaging op-

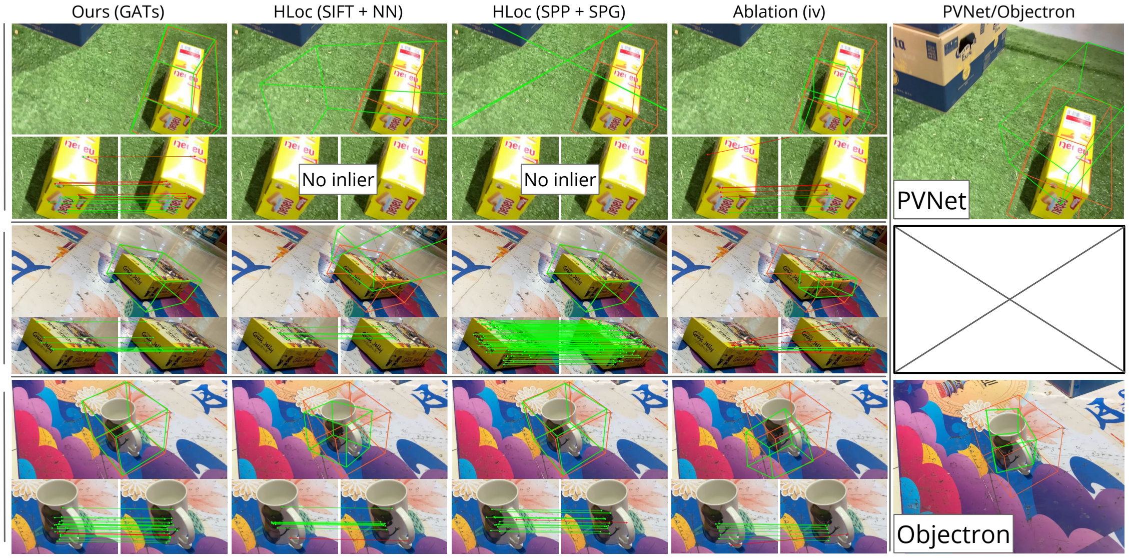

Figure 5. **Qualitative results.** Every two rows show the results on one test image. Green bounding boxes denote predictions and the red ones denote the ground truth. The 2D-3D matches are visualized by projecting the 3D matches of the detected 2D points (shown on the left image) onto the image plane (shown on the right image). Perfectly horizontal lines indicate correct 2D-3D matches. Reprojection error less than 10 px (in 512 × 512 images) is colored as green. RANSAC is applied to the matches to remove outliers. Our method is able to produce a larger quantity of matches compared to baseline HLoc (*SIFT + NN*). The matches from our method are also more accurate and less noisy compared to the baseline method Ablation (iv)). HLoc (*SPP + SPG*) achieves similar results with our method, but at a ∼10× run-time cost. PVNet and Objectron only obtained reasonable results in the first and last examples, respectively. Best viewed in color.

	Components			Large Objects	Medium Objects	Small Objects
	2D Feat.	Aggr.	Matching			
Ours	SPP	GATs	GNN	**0.471**	**0.629**	**0.405**
i	SPP	Avg.	GNN	0.449	0.602	0.390
ii	SPP	Avg.	NN	0.426	0.570	0.367
iii	SPP	K-Means	NN	0.431	0.595	0.379
iv	SIFT	Avg.	NN	0.355	0.470	0.256
v	SIFT	K-Means	NN	0.369	0.450	0.281

Table 4. **Ablation studies.** Different alternatives for components in the proposed method are compared with the *1cm-1deg* metric. SPP stands for SuperPoint and NN stands for Nearest Neighbor.

eration cannot adaptively select relevant information from different viewpoints according to different query features.

Other variants with 2D-3D NN Matching. To provide more comparisons with traditional pipelines [9, 35, 37] that estimate object pose with local features and 2D-3D matching, we also experimented with variants of our method based on different local features, feature aggregration methods and matchers for 2D-3D matching. The results are reported in Tab. 4 as (ii - v). (ii - v) are still unable to produce comparable results with our approach. Compared with (ii) and (iv) that use averaging for feature aggregation, our method consistently outperforms them by a significant margin. Similar to the analysis for (i), simply averaging the features from different viewpoints loses view-dependent information. For (iii) and (v), substituting averaging with K-Means clustering could provide richer 3D features but the results are still not comparable with ours.

Qualitative Comparisons. We provide some qualitative results to compare our method with baseline methods in Fig. 5. Please read the caption for details.

5. Conclusion

In this paper, we propose OnePose for one-shot object pose estimation. Unlike existing instance-level or category-level methods, OnePose does not rely on CAD models and can handle objects in arbitrary categories without instance- or category-specific network training. Compared with localization-based baseline methods, instance-level baseline method PVNet and category-level baseline method Objectron, OnePose achieves better pose estimation accuracy and faster inference speed. We also believe that our revisit to the localization-based setting (i.e., one-shot object pose estimation) is more practical for AR and valuable to the community.

Limitations. The limitations of our method come with the nature of relying on local feature matching for pose estimation. Our method may fail when applied to textureless objects. Although being enhanced by attention mechanisms, our method still has difficulty to handle extreme change of scales between images in the video scan and the testing sequences.

Acknowledgements. The authors would like to acknowledge the support from the National Key Research and Development Program of China (No. 2020AAA0108901), NSFC (No. 62172364), and the ZJU-SenseTime Joint Lab of 3D Vision.

References

[1] ARCore. https://developers.google.com/ar. 3

[2] ARKit. https://developer.apple.com/augmented-reality/. 3

[3] Ultralytics/yolov5. https://github.com/ultralytics/yolov5, 2021. 6

[4] Adel Ahmadyan, Liangkai Zhang, Jianing Wei, Artsiom Ablavatski, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *CVPR*, 2021. 1, 2, 3, 6, 7

[5] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV*. 2014. 3

[6] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, 2016. 6

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

[8] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. *ECCV*, 2020. 2

[9] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. *ICRA*, 2009. 3, 8

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. *CVPRW*, 2017. 3, 6, 7

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *arXiv:1707.07410 [cs]*, 2017. arXiv: 1707.07410. 3

[12] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable cnn for joint detection and description of local features. *CVPR*, 2019. 3

[13] 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. 6

[14] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, Proceedings of Machine Learning Research, 2020. 6

[15] 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. 2

[16] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. *ECCV*, 2020. 1, 2

[17] 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. 2

[18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6d pose estimation. In *ECCV*. 2018. 2

[19] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[20] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3, 6, 7

[21] Manuel Martinez, Alvaro Collet, and Siddhartha S Srinivasa. MOPED: A scalable and low latency object recognition and pose estimation system. In *ICRA*, 2010. 1, 3

[22] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3D object pose estimation. *ECCV*, 2018. 2

[23] 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. 2, 3

[24] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019. 2, 3

[25] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G. Derpanis, and Kostas Daniilidis. 6-DoF object pose from semantic keypoints. *ICRA*, 2017. 2

[26] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 2, 6, 7

[27] 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. 2, 6

[28] Kejie Qiu, Tong Qin, Wenliang Gao, and Shaojie Shen. Tracking 3-D motion of dynamic objects using monocular visual-inertial sensing. *IEEE Transactions on Robotics*, (4), 2019. 3

[29] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*. 2006. 3

[30] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *ICCV*, 2011. 3

[31] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 3, 4, 6, 7

[32] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *ICCV*, 2020. 2, 3, 4, 6, 7

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

[34] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew W. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 6

[35] I. Skrypnyk and D.G. Lowe. Scene modelling. In *ISMAR*. 1, 3, 8

[36] 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

[37] Jie Tang, Stephen Miller, Arjun Singh, and P. Abbeel. A textured object recognition pipeline for color and depth image data. *ICRA*, 2012. 3, 8

[38] Meng Tian, Marcelo H. Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6D object pose and size estimation. *ECCV*, 2020. 2

[39] Michal J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. In *NeurIPS*, 2020. 3

[40] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 2, 3, 4

[41] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. In *NeurIPS*, 2020. 6

[42] 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. 2

[43] 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, 2

[44] Jiaze Wang, Kai Chen, and Qi Dou. Category-level 6D object pose estimation via cascaded relation and recurrent reconstruction networks. *IROS*, 2021. 2

[45] B Wen and Kostas E Bekris. BundleTrack: 6d pose tracking for novel objects without instance or category-level 3d models. *ICRA*, 2021. 3

[46] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. *RSS*, 2018. 1, 2