Unsupervised Learning of Category-Level 3D Pose from Object-Centric Videos

Leonhard Sommer [1] Artur Jesslen [1] Eddy Ilg [2] Adam Kortylewski [1]

[1]University of Freiburg [2]Saarland University

Abstract

Category-level 3D pose estimation is a fundamentally important problem in computer vision and robotics, e.g. for embodied agents or to train 3D generative models. However, so far methods that estimate the category-level object pose require either large amounts of human annotations, CAD models or input from RGB-D sensors. In contrast, we tackle the problem of learning to estimate the category-level 3D pose only from casually taken object-centric videos without human supervision. We propose a two-step pipeline: First, we introduce a multi-view alignment procedure that determines canonical camera poses across videos with a novel and robust cyclic distance formulation for geometric and appearance matching using reconstructed coarse meshes and DINOv2 features. In a second step, the canonical poses and reconstructed meshes enable us to train a model for 3D pose estimation from a single image. In particular, our model learns to estimate dense correspondences between images and a prototypical 3D template by predicting, for each pixel in a 2D image, a feature vector of the corresponding vertex in the template mesh. We demonstrate that our method outperforms all baselines at the unsupervised alignment of object-centric videos by a large margin and provides faithful and robust predictions in-the-wild on the Pascal3D+ and ObjectNet3D datasets.

1. Introduction

Category-level object pose estimation is a fundamentally important task in computer vision and robotics with a multitude of real-world applications, e.g. for training 3D generative models on real data and for robots that need to grasp and manipulate objects. However, defining and determining the pose of an object is a task that is far from easy. Current approaches achieve high performance, but they require large amounts of annotated training data to generalize successfully [1, 26, 28, 33, 38], or additional inputs during inference, such as CAD Models [7, 14], 3D Shapes [3, 32, 37] or RGB-D [8]. However, all of these are either time-consuming to obtain or not available in practice at all. This motivates the development of methods for learning category-level 3D pose estimators in a fully unsupervised fashion. While doing so from images in the wild seems infeasible, object-centric video data [19] offers a more accessible alternative. Such videos can be easily captured using consumer-grade cameras and makes it possible to leverage coarse 3D reconstructions during training, providing a practical and cost-effective method for collecting data. Therefore, we propose the new task of learning a single-image category-level 3D pose estimator from casually captured object-centric videos without any human labels or other supervision. In practice, we leverage CO3D [19] as training data and show that our proposed model is able to generalize and predict accurate poses in the wild for Pascal3D+ [29] and ObjectNet3D [31].

We address the challenging task of learning category-level 3D pose in an unsupervised fashion from casually captured object-centric videos. In particular, we propose a two-step pipeline (Figure 1). The first step extracts DINOv2 [18] features from the images and reconstructs a coarse 3D mesh from the video with off-the-shelf methods [4, 11]. Building on this input, we introduce a novel 3D alignment procedure, where a key contribution is a novel 3D cyclical distance in terms of geometry and appearance that enables the robust alignment of shape reconstructions even under severe noise and variations in the object topology. As a result, we can align all objects from the object-centric training videos into a canonical coordinate frame without supervision.

In a second step, we leverage the canonical poses and 3D meshes obtained from the first step to train a category-level neural mesh [9, 16, 26, 36] in an unsupervised manner. In particular, we represent objects using a prototypical 3D mesh with surface features to capture the geometry and neural appearance of an object category and train a neural network backbone to predict, for each pixel in a 2D image, a feature vector of the corresponding vertex in the template mesh. Finally, the object 3D pose is solved using a pose fitting algorithm based on the estimated correspondence pairs.

We demonstrate that our method outperforms all baselines by a large margin at the unsupervised alignment of object-centric videos on the CO3D [19] dataset. Moreover,

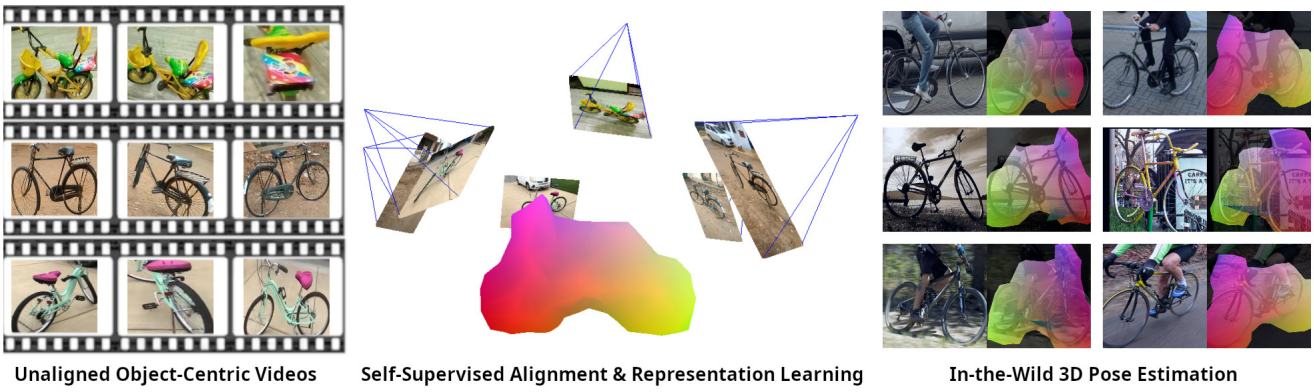

Figure 1. Illustration of our approach for the unsupervised learning of category-level 3D pose. Our method starts from unaligned object-centric videos of an object category (left) and aligns these into a canonical coordinate frame in a self-supervised manner using a prototypical 3D mesh and self-supervised transformer features (center). Using the aligned videos, we train a neural network backbone to predict 2D-3D correspondences from a single image to enable 3D object pose estimation in the wild (right).

our model provides faithful and robust predictions in-the-wild on Pascal3D+ [29] and ObjectNet3D [31] despite being trained from raw object-centric videos only.

2. Related Work

Supervised Category-Level 3D Pose Estimation. Traditional methods to determine object poses were to label keypoints in images and train supervised methods to predict them [23, 38], or to use only pose labels and directly predict them by casting the pose estimation problem as bin classification by discretizing the pose space [21]. More recent methods utilize 3D meshes of objects or object categories. NeMo [26] uses 3D meshes with neural features that are rendered-and-compared to feature maps from a CNN to obtain pose estimates. [13] predicts a single embedding vector per image and shows that superior performance can be obtained by simply retrieving the closest training sample. In contrast to the above methods, our approach is fully unsupervised.

Few-Shot and Zero-Shot Pose Prediction. [33] proposes to train a supervised pose estimator across many categories and shows that their approach can generalize well to similar but unseen objects. Many works implement zero shot pose estimation by conditioning the model on the 3D shape of the unseen object [3, 32, 34, 37] or by leveraging renderings of CAD models [7, 14]. Other methods use few-shot learning [22] or zero-shot learning with DINOv2 [18]. In contrast to the above, our method does not require any annotated dataset, 3D shapes or CAD models.

Pose Alignment. The work from Goodwin et al. [5] (ZSP) is most close to the first step of our method, as it aligns the poses of two object-centric videos in a fully unsupervised fashion. Similar to our work, they use DINO [2] to obtain semantic correspondences. They perform first a coarse alignment by matching one image from the source video to one of many images from the reference video. Then they leverage cyclical distances to select few promising correspondences in the two images, and finally leverage respective depth maps to align both images using least squares. Goodwin et al. extend their work in [6] (UCD+) to match many images from the source video to many of the reference video. By finding a consensus over these many to many alignments with a single transformation they demonstrate improved performance. The first step of our work is similar to these works by that it also leverages DINO features and cyclical distances. However, our work adds a geometric distance to perform the alignment directly in 3D and introduces weighted correspondences to enable the necessary robust regression of the SE3 transformations from noisy and inaccurate geometries, which leads to significant improvements in the alignment accuracy. Note also that these previous approaches used RGB-D inputs, while our method works on images directly.

Surface Embeddings and Neural Mesh Models. Recent work uses known poses and approximate object geometries to learn features in 3D space to uniquely identify parts of objects. [17] first used known mesh templates of deformable objects to train a network that predicts surface embeddings from the images. NeMo [26] presents a generative model trained with contrastive learning. [27] presents an extension through replacing vertices by Gaussian ellipsoids and using volume rendering. Similar to our work, many recent works leverage pre-trained vision tranformers DINO [2] and DINOv2 [18] to unproject image features onto depth maps [5, 6, 37]. In contrast to [17, 26, 27], our approach does not require any pose annotation, while also going beyond [5, 6] by enabling 3D pose estimation in the wild from a single image, and not requiring RGB-D images and CAD models as input [37].

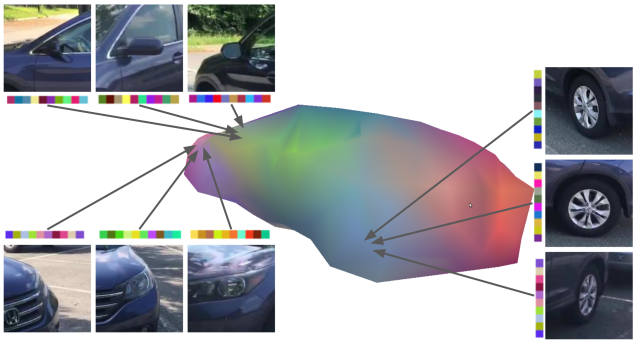

Figure 2. Illustration of a neural mesh. It consists of a 3D mesh with one or several neural features per vertex. These features encode patch-level features from a feature extractor. In our unsupervised alignment method, we capture the viewpoint-dependent features from DINOv2. For pose estimation, we learn a category-level neural mesh with viewpoint-invariant features.

3. Method

In this section, we describe our approach for learning category-level 3D pose estimation without supervision from object-centric videos. Our method proceeds in a two-step approach. First, we align object instances across videos in an unsupervised manner to bring them into a canonical reference frame (Section 3.2). Given the aligned videos, our model learns to establish dense correspondences between images and a reconstructed 3D template mesh by predicting a feature vector for each pixel in a 2D image that corresponds to a visible vertex in the template mesh (Section 3.3). Finally, we describe how our model can efficiently estimate object poses from in-the-wild data using the predicted correspondences via render-and-compare.

3.1. Meshes with Surface Features

In both steps of our approach, the video alignment and the representation learning, we represent objects as *neural meshes*, i.e. meshes with surface features to capture the geometry and appearance of an object instance or category [9, 16, 26, 36]. In particular, the geometric representation is a triangular mesh, where we denote the set of vertices as $\mathrm{V} = \{\mathrm{v}_i \in \mathbb{R}^3\}_{i=1}^{|V|}$. The appearance is represented by storing one or multiple appearance features $\mathrm{F} = \{\{\mathrm{f}_i^k \in \mathbb{R}^D\}_{k=1}^{|F|}\}_{i=1}^{|V|}$ at each mesh vertex. Together, the geometry and appearance define a neural mesh as $S = \{V, F\}$.

3.2. Self-supervised Alignment of Objects

Our goal is to align the camera poses of multiple object-centric videos into a common coordinate frame. To achieve this, we represent each object-centric video as a neural mesh with self-supervised surface features. In particular, we utilize off-the-shelf structure-from motion [20] to obtain a coarse object shape reconstruction for each video. Note that the reconstructed shapes cover the whole object, as the object-centric videos move in a full circle around the object. We post-process the reconstructed point cloud to clean it and generate a watertight mesh, for which we provide details in the supplementary material. Subsequently, we project the reconstructed coarse meshes into the feature map $\Psi(I)$ that is obtained from a self-supervised transformer backbone [18]. We collect from every video a set of feature vectors for every vertex v_i that describe the appearance of a local patch (Figure 2). Thus the number of features per vertex depends on the number of images in which the vertex is visible. As feature extractor Ψ, we use a self-supervised vision transformer [18] which has shown emerging correspondence matching abilities.

Finding geometric and appearance correspondences. Given the mesh vertices of the source object instance V and the corresponding aggregated features F, we aim to align them to the reference counterparts $\bar{\mathrm{V}}$ and $\bar{\mathrm{F}}$. In practice, we select a reference video at random from the set of all available videos and align the remaining videos to the reference. More precisely, we aim to optimize the transformation T, which is composed of rotation, translation and scale, under which the transformed source vertices and corresponding features yield the minimal distance to the reference counterparts with respect to geometry as well as appearance. Formally, our optimization problem optimizes

$$\min_{\mathrm{T}} \mathcal{D}(S, \bar{S}, \mathrm{T}) = \mathcal{D}_{\text{geo}}(\mathrm{V}, \bar{\mathrm{V}}) + \mathcal{D}_{\text{app}}(S, \bar{S}), \tag{1}$$

where $\mathcal{D}$ is the similarity between two videos given a transformation T, which combines a geometric distance between the mesh geometries $\mathcal{D}_{\text{geo}}(\mathrm{V}, \bar{\mathrm{V}})$ and an appearance distance $\mathcal{D}_{\text{app}}(S, \bar{S})$ between surface features.

Assuming that the object instances shapes contain a negligible variance and no symmetries, a suitable geometric distance is the Chamfer Distance defined as

$$\mathcal{D}_{\text{geo}}(\mathrm{V}, \bar{\mathrm{V}}) = \sum_{\mathrm{v}_i \in \mathrm{V} \cup \bar{\mathrm{V}}} ||\mathrm{v}_i - \mathrm{v}_{\chi(\mathrm{v}_i)}||_2, \tag{2}$$

where $\chi(\mathrm{v}_i)$ is the vertex index of the Euclidean nearest neighbor of vertex v_i in the respective other set of vertices

$$\chi(\mathrm{v}_i) = \begin{cases} \underset{j \in 1 \ldots |\bar{\mathrm{V}}|}{\operatorname{argmin}} ||\mathrm{T}\mathrm{v}_i - \mathrm{v}_j||_2 \,, & \text{for } \mathrm{v}_i \in \mathrm{V} \\ \underset{j \in 1 \ldots |\mathrm{V}|}{\operatorname{argmin}} ||\mathrm{T}\mathrm{v}_j - \mathrm{v}_i||_2 \,, & \text{for } \mathrm{v}_i \in \bar{\mathrm{V}}. \end{cases} \tag{3}$$

However, as the 3D object is rather coarse and noisy, and the alignment can be ambiguous due to symmetries or shape differences among objects, we optimize each vertex to also be geometrically close to its nearest neighbor in feature space using the appearance distance

$$\mathcal{D}_{\text{app}}(S, \bar{S}) = \sum_{\mathrm{v}_i \in \mathrm{V} \cup \bar{\mathrm{V}}} ||\mathrm{v}_i - \mathrm{v}_{\psi(\mathrm{v}_i, \mathrm{f}_i)}||_2, \tag{4}$$

with the nearest neighbor in feature space $\psi(\mathrm{v})$ defined as

$$\psi(\mathrm{v}_i, \mathrm{f}_i) = \begin{cases} \underset{j \in 1 \ldots |\bar{\mathrm{V}}|}{\operatorname{argmin}} \min_{k,l} ||\mathrm{f}_j^k - \mathrm{f}_i^l||\,, & \text{for } \mathrm{v}_i \in \mathrm{V} \\ \underset{j \in 1 \ldots |\mathrm{V}|}{\operatorname{argmin}} \min_{k,l} ||\mathrm{f}_j^k - \mathrm{f}_i^l||\,, & \text{for } \mathrm{v}_i \in \bar{\mathrm{V}}. \end{cases} \tag{5}$$

As the self-supervised vertex features are view-dependent, the appearance distance computes the minimum feature distance across all views to select the nearest neighbor.

Weighting Correspondences. An open challenge for the alignment of casually captured object-centric videos is that the estimated correspondence pairs between videos can be unreliable. For example, due to errors in the shape reconstruction or significant topology changes among different object instances, such as one bicycle having support wheels whereas the other does not. In these cases the correspondences in the geometry and feature space are ill-defined which leads to unreliable correspondence estimates. To account for such unreliable correspondences, we introduce a weight factor for each correspondence pair that estimates its quality. At the core of the correspondence weighting, we introduce a 3D cyclical distance among the vertices of two neural meshes that is inspired by 2D cyclical distances [5] for correspondence estimation, and is defined as

$$\mathrm{d}_{\text{cycle}}(\mathrm{v}_i, \mathrm{f}_i) = ||\mathrm{v}_i - \mathrm{v}_{\psi(\mathrm{v}_j, \mathrm{f}_j)}||_2, \text{ with } j = \psi(\mathrm{v}_i, \mathrm{f}_i). \tag{6}$$

The nested structure of our 3D cyclical distance first computes j as the index of the nearest neighbor of vertex v_i in the feature space, and in turn computes the nearest neighbor of v_j as $\mathrm{v}_{\psi(\mathrm{v}_j, \mathrm{f}_j)}$. Notably, $\mathrm{d}_{\text{cycle}}(\mathrm{v}_i, \mathrm{f}_i) = 0$ if the nearest neighbour maps back to the original vertex $\mathrm{v}_{\psi(\mathrm{v}_j, \mathrm{f}_j)} = \mathrm{v}_i$ and hence the correspondence is reliable. Building on this 3D cyclical distance, we define the validity criteria for each pair of vertices as the sum of cyclical distances of the correspondence pair

$$\rho(\mathrm{v}_i\,, \mathrm{f}_i\,, \mathrm{v}_j\,, \mathrm{f}_j) = -\frac{\mathrm{d}_{\text{cycle}}(\mathrm{v}_i, \mathrm{f}_i) + \mathrm{d}_{\text{cycle}}(\mathrm{v}_j, \mathrm{f}_j)}{2\tau(\mathrm{D}(\mathrm{V}) + \mathrm{D}(\bar{\mathrm{V}}))}, \tag{7}$$

where $\mathrm{D}(\cdot)$ is the diameter of a neural mesh given as $\mathrm{D}(\mathrm{V}) = \max_{\mathrm{v}_i, \mathrm{v}_j \in \mathrm{V}} ||\mathrm{v}_i - \mathrm{v}_j||_2$.

To obtain the final weight factor for a correspondence pair we use the softmax normalization $\sigma_\tau(i, j) = \texttt{Softmax}_\tau(\rho(\mathrm{v}_i\,, \mathrm{f}_i\,, \mathrm{v}_j\,, \mathrm{f}_j))$, across all feature and gemoetric correspondences. For the softmax normalization, we introduce the temperature τ, which enables us to steer between taking into account fewer high quality correspondences or more low quality ones (see Section 4.4). Together with the weighting we formulate the weighted geometric distance as

$$\mathcal{D}^*_{\text{geo}}(S, \bar{S}) = \sum_{\mathrm{v}_i \in \mathrm{V} \cup \bar{\mathrm{V}}} \sigma(i, \chi(\mathrm{v}_i))\; ||\mathrm{v}_i - \mathrm{v}_{\chi(\mathrm{v}_i)}||_2, \tag{8}$$

and likewise the weighted appearance distance as

$$\mathcal{D}^*_{\text{app}}(S, \bar{S}) = \sum_{\mathrm{v}_i \in \mathrm{V} \cup \bar{\mathrm{V}}} \sigma(i, \psi(\mathrm{v}_i))\; ||\mathrm{v}_i - \mathrm{v}_{\psi(\mathrm{v}_i)}||_2. \tag{9}$$

Our final distance measure to compare two neural meshes is computed as

$$\mathcal{L}(S, \bar{S}) = (1 - \alpha)\mathcal{D}^*_{\text{geo}}(S, \bar{S}) + \alpha\mathcal{D}^*_{\text{app}}(S, \bar{S}). \tag{10}$$

To find an approximately optimal solution, we use a RANSAC strategy, where we randomly choose four vertices on the source surface mesh. Together with their nearest neighbors in the feature space on the reference surface mesh, we estimate a single transformation using the Umeyama method [24].

3.3. 3D Pose Estimation In-the-Wild

Our goal is to perform 3D pose estimation in in-the-wild images. To achieve this, we generalize our approach from the multi-view setting used to align object-centric videos, towards 3D pose inference from a single image. Our model uses a feature extractor $\Psi_w(I) = F \in \mathbb{R}^{D \times H \times W}$ to obtain image features from input image I, where w denotes the parameters of the backbone. The backbone output is a feature map F with feature vectors $f_i \in \mathbb{R}^D$ at positions i on a 2D lattice. For training, we use the aligned object-centric videos (Section 3.2) to train the weights w of the feature extractor Ψ_w such that it predicts dense correspondences between image pixels and the 3D neural mesh template. Specifically, we relate the features of an image $\Psi_w(I)$ extracted by a backbone feature extractor to the vertex and background features by Von-Mises-Fisher (vMF) probability distributions [12]. In particular, we model the likelihood of generating the feature at an image pixel f_i from corresponding vertex feature f_r as $P(f_i|f_r) = c_p(\kappa)e^{\kappa f_i \cdot f_r}$, where f_r is the mean of each vMF kernel, κ is the corresponding concentration parameter, and c_p is the normalization constant ($\|f_i\| = 1, \|f_r\| = 1$). We also model the likelihood of generating the feature f_i from background feature as $P(f_i|\beta) = c_p(\kappa)e^{\kappa f_i \cdot \beta}$ for $\beta \in \mathbb{R}^D$.

When learning the models, as described next, we will learn the vertex features $\{f_r\}$, the background feature $\beta \in \mathbb{R}^D$, and the parameters w of the neural network backbone Ψ_w. We emphasize that our model requires that the backbone must be able to extract features that are invariant to the viewpoint of the object to ensure that $f_i \cdot f_r$ is large irrespective of the viewpoint.

Learning viewpoint-invariant vertex features. For training our model, we use the visible vertex features and their corresponding image features $\mathcal{P} = \{(f_r, f_i)\}$. Further, we use image features randomly sampled from the background $\mathcal{B} = \{f_i\}$. As optimization objective, we use the

cross-entropy loss

$$\mathcal{L}_{train} = -\sum_{(f_r,f_i)\in\mathcal{P}} \log\left(\frac{P(f_i|f_r)}{\sum P(f_i|f_r)+\sum P(f_i|\beta)}\right) - \sum_{f_i\in\mathcal{B}} \log\left(\frac{P(f_i|\beta)}{\sum P(f_i|f_r)+\sum P(f_i|\beta)}\right). \tag{11}$$

3D pose inference. We use the mesh with the vertex features $\{f_r\}$, the background feature β and the trained backbone Ψ_w to estimate the camera pose α via render-and-compare. At each optimization step, we render a feature map $\{\bar{f}_i(\alpha)\}$ under pose α and compare it with the encoder's feature map $F = \{f_i\}$. Determined by the rendering, each feature map consists of foreground features F_{front} and background features $F_{back} = F \backslash F_{front}$. Thereupon, we maximize the joint likelihood for all image features under the assumption of independence, given as

$$P(F|\alpha,\{f_r\},\beta) = \prod_{f_i\in F_{front}} \max_{f'\in\{\bar{f}_i(\alpha),\beta\}} P(f_i|f') \prod_{f_i\in F_{back}} P(f_i|\beta). \tag{12}$$

Note, by allowing foreground image features to be generated by the background feature, we also account for clutter.

We estimate the pose by first finding the best initialization of the object pose α by computing the joint likelihood (Eq.12) for a set of pre-defined poses via template matching and choosing the one with the highest likelihood. Subsequently, we iteratively update our initial pose using a differentiable renderer to obtain the final pose prediction $\hat{\alpha}$.

4. Experiments

In this section, we discuss our experimental setup (Section 4.1), present baselines and results for unsupervised alignment of object-centric videos (Section 4.2) and 3D pose estimation in-the-wild (Section 4.3). Additionally, we perform ablations of key model components in Section 4.4.

4.1. Experimental Setup

Dataset for alignment. To evaluate the unsupervised alignment of object-centric videos, we use the recently released Common Objects in 3D (CO3D) dataset [19] that provides images of multiple object categories, with a large amount of intra-category instance variation, and with varied object viewpoints. It contains 1.5 million frames, capturing objects from 50 categories, across nearly 19k scenes. For each object instance, CO3D provides approximately $100-200$ frames promising a $360°$ viewpoint sweep with handheld cameras. CO3D supplements these videos with relative camera poses and estimated object point clouds using Structure-from-Motion [20].

We find that the unfiltered videos of CO3D are not ideal for our purpose. In particular, we find that videos with little viewpoint variation lead to inferior structure-from-motion results. Also, videos that are not focusing on the object's center in 3D or are taken too close to it, contain little information for correspondence learning. Therefore, we filter the videos accordingly, targeting 50 videos per category. For multiple categories we end up with less than 50 videos namely, "remote" 17, "mouse" 15, "tv" 16, "toilet" 7, "toybus" 41, "hairdryer" 28, "couch" 49, and "cellphone" 23. With our simple filters, we end up aiming for 50 videos per category. More precise details for the filtering procedure are appended in the supplementary. As labels, we use the ground truth pose annotations provided by ZSP [5], that cover ten object instances of twenty different categories.

Datasets for 3D pose estimation in-the-wild. We evaluate on two common datasets PASCAL3D+ [30] and ObjectNet3D [31]. While PASCAL3D+ provides poses for the 12 rigid classes of PASCAL VOC 2012, ObjectNet3D covers pose annotations for over 100 categories. The object-centric video dataset CO3D covers 50 categories from the MS-COCO [15] dataset. We find 23 common categories across ObjectNet3D and CO3D, even tolerating the gap between a toybus in CO3D and a real one in PASCAL3D+ and ObjectNet3D. We believe that this non-neglegible gap could be bridged by exploiting the multiple viewpoint knowledge of the same object instance. Overall we validate on PASCAL3D+ with 6233 images, using the same validation set as [25], and on ObjectNet3D on 12039 images. Following [38], we center all objects.

Implementation details. In our alignment step, we use $\tau = 100$ and $\alpha = 0.2$. Further, we leverage as self-supervised ViT the publicly available small version of DINOv2 [18] with 21M parameters and a patch size of 14. At the input we use a resolution 448x448 ending up with a 32x32 feature map, where each feature yields 384 dimensions. In our second step, we use the same ViT as backbone and freeze its parameters. Further, we add on top three ResNet blocks with an upsampling step preceding the final block. Ending up with a 64x64 feature map, where each feature has 128 dimensions. We optimize the cross-entropy loss for 10 epochs with Adam [10]. In one epoch, we make use of all filtered videos. The training for each category-level representation takes less than an hour on a single NVIDIA GeForce RTX 2080.

We note that the quality of our alignment method and the subsequent representation learning can vary depending on the chosen reference video. Therefore, we randomly choose five reference videos per category and report the mean performance and the standard deviation across all results.

We report the $30°$ accuracy for pose estimation where the angle error for an estimated rotation R_{pred} and a ground

	Per Category						All Categories (20)	
	backpack	car	chair	keyboard	laptop	motorcycle	Acc. 30°	Acc. 15°
TEASER++ [35]	1.0	5.0	9.0	8.0	6.0	1.0	3.8	1.1
ZSP [5]	44.0	65.0	47.0	69.0	85.0	85.0	49.4	28.4
UCD+ [6]	67.0	86.0	76.0	76.0	**100.0**	**100.0**	69.8	54.6
Ours	**85.6**	**100.0**	**98.9**	**82.2**	**100.0**	**100.0**	**77.0**	**61.6**
	± 10.5	± 0.0	± 3.5	± 10.7	± 0.0	± 0.0	± 11.9	± 15.9

Table 1. Unsupervised alignment evaluation on the CO3D dataset across 20 categories. The reported metric is the 30° accuracy if not stated otherwise. The mean is computed across all 20 categories. We see that our method substantially outperforms the state of the art.

Figure 3. Qualitative comparison of two unsupervised alignment methods. The first row shows the alignment of our proposed method. The second row shows the alignment using ZSP [5]. For both methods we use the 5th object instance from left as reference. We see that our proposed method is more accurate compared with ZSP. Especially for cars ZSP often confuses back and front.

truth rotation R_{pred} is given as

$$\Delta(\mathrm{R}_{pred}, \mathrm{R}_{gt}) = \arccos\left(\frac{1}{2}\mathrm{tr}\left(\mathrm{R}_{pred}^{\mathrm{T}}\mathrm{R}_{gt}\right) - 1\right). \quad (13)$$

We note that for the current state of unsupervised pose estimation, achieving 30° precision remains unsolved.

4.2. Unsupervised Alignment

We follow the evaluation protocol of ZSP [5] and measure the alignment of one object instance to the nine remaining ones of the same category that are labelled. Additionally, we report the standard deviation across the chosen reference object instances. The quantitative results in Table 1 show that our proposed method significantly improves the state of the art by 7.2% from 69.8% to 77.0%. Our alignment algorithm can more efficiently use the video frames compared to ZSP [5], which only compares a single RGB-D frame from the source video with many RGB-D frames of the reference video. We note that ZSP uses DINOv1 features in contrast to our method, which uses DINOv2. Therefore, we provide an ablation of our method with respect to different feature extractors in the supplementary. One reason for our model to outperform UCD+ [6], an extension of ZSP, is likely that our optimization does exploit the object geometry extensively, whereas others are using it only for refinement. A qualitative comparison of our method against ZSP is depicted in Figure 3. It shows that our alignments are highly accurate despite a large variability in the object instances. We note that at the time of writing, there is no source code publicly available to compare with UCD+.

4.3. In-the-Wild 3D Pose Estimation

As we are not aware of any unsupervised method learning pose estimation from videos, we compare our pose estimation method against two supervised methods [27, 38] and ZSP. We provide ZSP with ten uniformly-distributed images of the same reference video that our method uses. Further, we provide ZSP with depth annotations using the category-level CAD models and pose annotations in the PASCAL3D+ and ObjectNet3D data. Despite our method not requiring any depth information, it outperforms ZSP by a large-margin on both PASCAL3D+, see Table 2, and on ObjectNet3D, see Table 3. Qualitative results are depicted in Figure 4. We find that ZSP is highly compute intensive, requiring 10.92 seconds per sample on average, while our proposed method takes only 0.22 seconds on average.

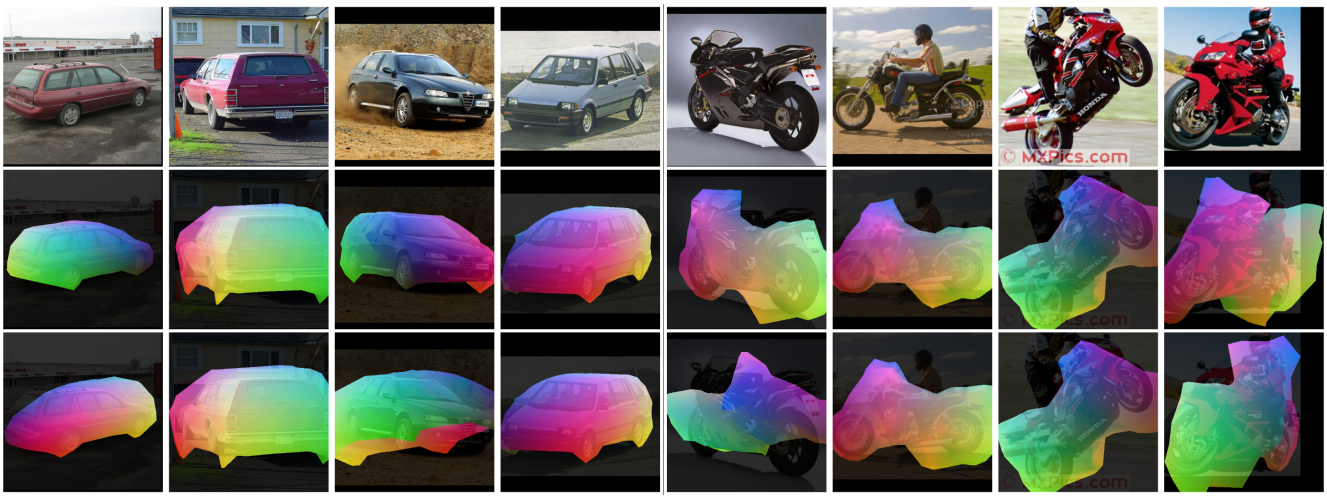

Figure 4. Qualitative comparison of our method (top) and ZSP (bottom) at category-level 3D pose prediction in the wild on samples from PASCAL3D+ and ObjectNet3D (we randomly selected the samples to demonstrate the diversity of the results). For both methods, we overlay our coarse mesh reconstruction in the predicted 3D pose.

	Method	bicycle	bus	car	chair	motorcycle	couch	tv	Mean
Supervised	StarMap [38]	83.2	94.4	90.0	75.4	68.8	79.8	85.8	82.49
	VoGE [27]	82.6	98.1	99.0	90.5	87.5	94.9	83.9	90.93
Unsupervised	ZSP	**61.7**	21.4	61.6	42.6	43.1	52.9	39.0	46.0
		± 14.7	± 8.1	± 11.4	± 11.3	± 22.1	± 16.0	± 36.2	± 17.1
	Ours	58.4	**79.3**	**98.2**	**51.9**	**67.0**	**76.6**	**53.1**	**69.2**
		± 5.0	± 11.8	± 1.0	± 10.5	± 8.9	± 13.7	± 20.4	± 10.2

Table 2. 3D Pose Estimation in-the-wild on 7 categories of PASCAL3D+. Top two rows show supervised methods (as upper bound) while bottom two rows show unsupervised methods. The reported metric is 30° accuracy. The mean is averaged over all 7 categories. Our method shows superior performance over ZSP, which requires depth annotations.

Categorical discussion. We observe that our method performs better for categories with only small topology changes and deformations, (e.g. car, microwave, couch) compared to categories with large intra-class variability (e.g. chair). Further, we recognize, that our method even generalizes well from a toybus to a real bus. Besides that, we analyze, that categories with less available videos (e.g. remote, TV, toilet) on average achieve lower performance.

4.4. Ablation

Unsupervised alignment. Using the ground-truth annotations of our five references, we measure the effect of both parameters introduced in the alignment method. Namely, the appearance distance weight α and the cyclical distance temperature τ. We remark that for the distance between two meshes with surface features, the appearance weight trades-off feature correspondences versus Euclidean correspondences. Where an appearance weight of $\alpha = 0$ means that the distance depends solely on the Euclidean correspondences. Contrarily, an appearance weight of $\alpha = 1$ results in solely depending on feature correspondences. Further, our cyclical distance temperature weights each correspondence, implicitly trading-off many low-quality correspondences versus few high-quality ones. Intuitively, increas-

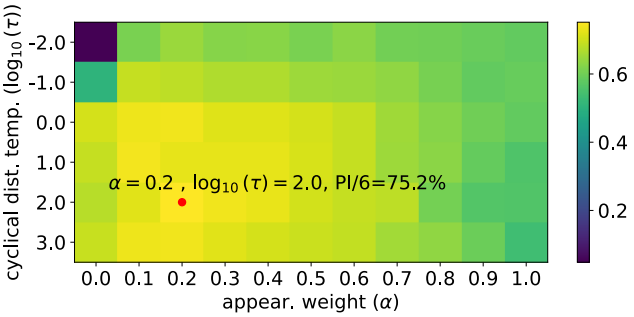

Figure 5. We report the 30° accuracy of our alignment method for different choices of our appearance weight α and our cyclical distance temperature τ resulting in different distances between two meshes with surface features. We see that the maximum accuracy of 75.2% is reached for $\alpha = 0.2$ and $\tau = 100$.

	phone	m'wave	b'pack	bench	cup	h'dryer	laptop	mouse	remote	toaster	
ZSP	46.4	50.5	**23.1**	50.8	33.0	**21.7**	**60.5**	28.8	41.6	28.8	
	± 8.4	± 44.3	± 14.9	± 17.0	± 18.7	± 13.3	± 8.9	± 10.1	± 18.1	± 8.16	
Ours	**54.6**	**80.3**	18.0	**62.1**	**38.2**	14.1	53.3	**44.7**	**54.4**	**60.6**	
	± 3.4	± 21.7	± 12.0	± 7.1	± 21.6	± 5.8	± 9.7	± 9.9	± 4.4	± 2.2	
	toilet	b'cycle	bus	car	chair	couch	k'board	m'cycle	suitcase	tv	Mean
ZSP	**56.3**	**58.6**	30.5	60.3	36.8	55.5	**46.8**	50.3	**25.8**	37.9	42.2
	± 13.9	± 10.4	± 10.6	± 9.1	± 10.6	± 16.1	± 14.5	± 20.7	± 14.3	± 35.1	± 15.9
Ours	39.6	57.8	**78.3**	**98.1**	**52.2**	**76.6**	26.9	**69.0**	15.5	**53.2**	**52.4**
	± 12.7	± 5.1	± 12.1	± 0.9	± 9.6	± 12.2	± 8.4	± 9.1	± 5.5	± 22.0	± 9.8

Table 3. 3D Pose Evaluation on 10 categories of ObjectNet3D. The mean is averaged over 20 categories. Metric is the 30° accuracy. Despite not requiring any depth information, our method significantly outperforms ZSP.

	Method	Acc. 30°	Acc. 15°
PASCAL3D+	Regression	66.0	25.8
	Ours	**69.2**	**41.3**
ObjectNet3D	Regression	44.5	16.4
	Ours	**52.4**	**25.5**

Table 4. Average 30° and 15° accuracies on PASCAL3D+ and ObjectNet3D for using directly neural network regression.

ing the value of τ results in averaging over more correspondences, while decreasing τ results in taking only the correspondences with high validity into account. In Figure 5, we see that both parameters yield a significant impact on the 30° accuracy. With an optimum for $\alpha = 0.2$ and $\tau = 100$. Intuitively, this means that taking many correspondences into account is more beneficial. Additionally, the Euclidean correspondences are weighted four times as much as the feature correspondences. Besides that, the ablation shows that while many correspondences are essential for using solely Euclidean correspondences, the opposite is true when using solely feature correspondences.

3D pose estimation in-the-wild. Following the alignment, the in-the-wild 3D pose estimation task can also be solved using neural network regression with the 6D rotation representation proposed in [39]. However, we observe that the results are worse than our 3D template learning method combined with render-and-compare, see Table 4.

4.5. Limitations

We have proposed a model which substantially outperforms existing applicable baselines for the task of unsupervised category-level 3D pose estimation in-the-wild. However, our proposed method does not yet reach the performance of fully supervised baselines. One advancement we aspire is to relax the rigidity constraint of our shape model. Therefore, we plan to leverage the aligned reconstructions and introduce a parameterized model for the shape. A deformable shape would yield the potential to improve the correspondence learning as well as the subsequent matching of features at inference. Moreover, we see a future research direction in enabling the model to learn from a continuous stream of data, instead of building on a set of pre-recorded videos. This would even better reflect the complex real-world scenarios of embodied agents.

5. Conclusion

In this paper, we have proposed a highly challenging (but realistic) task: unsupervised category-level 3D pose estimation from object-centric videos. In our proposed task, a model is required to align object-centric videos of instances of an object category without having any pose-labelled data. Subsequently, the model learns a 3D representation from the aligned videos to perform 3D category-level pose estimation in the wild. Our task defines a complex real-world problem which requires both semantic and geometric understanding of objects, and we demonstrate that existing baselines cannot solve the task. We further proposed a novel method for unsupervised learning of category-level 3D pose estimation that follows a two-step process: 1) A multi-view alignment procedure that determines canonical camera poses across videos with a novel and robust cyclic distance formulation for geometric and appearance matching. 2) Learning dense correspondences between images and a prototypical 3D template by predicting, for each pixel in a 2D image, a feature vector of the corresponding vertex in the template mesh. The results showed that our proposed method achieves large improvements over all baselines, and we hope that our work will pave the ground for future advances in this important research direction.

6. Acknowledgement

Adam Kortylewski acknowledges support for his Emmy Noether Research Group funded by the German Science Foundation (DFG) under Grant No. 468670075.

References

[1] Shuichi Akizuki and Manabu Hashimoto. Asm-net: Category-level pose and shape estimation using parametric deformation. In *Proceedings of the British Machine Vision Conference*, pages 1–13, 2021. 1

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. 2

[3] Kai Chen and Qi Dou. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2773–2782, 2021. 1, 2

[4] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 1

[5] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. Zero-shot category-level object pose estimation. In *European Conference on Computer Vision*, pages 516–532. Springer, 2022. 2, 4, 5, 6

[6] Walter Goodwin, Ioannis Havoutis, and Ingmar Posner. You only look at one: Category-level object representations for pose estimation from a single example, 2023. 2, 6

[7] Alexander Grabner, Peter M Roth, and Vincent Lepetit. 3d pose estimation and 3d model retrieval for objects in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3022–3031, 2018. 1, 2

[8] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Inferring 3d object pose in rgb-d images. *arXiv preprint arXiv:1502.04652*, 2015. 1

[9] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021. 1, 3

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[11] D Kirkpatrick and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983. 1

[12] Adam Kortylewski, Ju He, Qing Liu, and Alan L Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8940–8949, 2020. 4

[13] Georgios Kouros, Shubham Shrivastava, Cédric Picron, Sushruth Nagesh, Punarjay Chakravarty, and Tinne Tuytelaars. Category-level pose retrieval with contrastive features learnt with occlusion augmentation. *arXiv preprint arXiv:2208.06195*, 2022. 2

[14] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022. 1, 2

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[16] Natalia Neverova, David Novotny, Marc Szafraniec, Vasil Khalidov, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. *Advances in Neural Information Processing Systems*, 33:17258–17270, 2020. 1, 3

[17] Natalia Neverova, David Novotny, Marc Szafraniec, Vasil Khalidov, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. *Advances in Neural Information Processing Systems*, 33:17258–17270, 2020. 2

[18] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 5

[19] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 1, 5

[20] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 3, 5

[21] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE international conference on computer vision*, pages 2686–2694, 2015. 2

[22] Hung-Yu Tseng, Shalini De Mello, Jonathan Tremblay, Sifei Liu, Stan Birchfield, Ming-Hsuan Yang, and Jan Kautz. Few-shot viewpoint estimation. *arXiv preprint arXiv:1905.04957*, 2019. 2

[23] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 2

[24] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04): 376–380, 1991. 4

[25] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12645–12654, 2020. 5

[26] Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. *arXiv preprint arXiv:2101.12378*, 2021. 1, 2, 3

[27] Angtian Wang, Peng Wang, Jian Sun, Adam Kortylewski, and Alan Yuille. Voge: a differentiable volume renderer using gaussian ellipsoids for analysis-by-synthesis. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 6, 7

[28] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 1

[29] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. 1, 2

[30] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. 5

[31] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 160–176. Springer, 2016. 1, 2, 5

[32] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3d objects. *BMVC*, 2019. 1, 2

[33] Yang Xiao, Yuming Du, and Renaud Marlet. Posecontrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning. In *2021 International Conference on 3D Vision (3DV)*, pages 74–84. IEEE, 2021. 1, 2

[34] Yang Xiao, Vincent Lepetit, and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3090–3106, 2022. 2

[35] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020. 6

[36] Yanjie Ze and Xiaolong Wang. Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset. *Advances in Neural Information Processing Systems*, 35:27469–27483, 2022. 1, 3

[37] Kaifeng Zhang, Yang Fu, Shubhankar Borse, Hong Cai, Fatih Porikli, and Xiaolong Wang. Self-supervised geometric correspondence for category-level 6d object pose estimation in the wild. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2

[38] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018. 1, 2, 5, 6, 7

[39] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019. 8