PoseContrast: Class-Agnostic Object Viewpoint Estimation in the Wild with Pose-Aware Contrastive Learning

Yang Xiao[1]   Yuming Du[1]   Renaud Marlet[1,2]

[1]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France   [2]Valeo.ai, Paris, France

Abstract

Motivated by the need for estimating the 3D pose of arbitrary objects, we consider the challenging problem of class-agnostic object viewpoint estimation from images only, without CAD model knowledge. The idea is to leverage features learned on seen classes to estimate the pose for classes that are unseen, yet that share similar geometries and canonical frames with seen classes. We train a direct pose estimator in a class-agnostic way by sharing weights across all object classes, and we introduce a contrastive learning method that has three main ingredients: (i) the use of pre-trained, self-supervised, contrast-based features; (ii) pose-aware data augmentations; (iii) a pose-aware contrastive loss. We experimented on Pascal3D+, ObjectNet3D and Pix3D in a cross-dataset fashion, with both seen and unseen classes. We report state-of-the-art results, including against methods that additionally use CAD models as input. Code is available at `https://github.com/YoungXIAO13/PoseContrast`.

1. Introduction

Object 3D pose (viewpoint) estimation aims at predicting the 3D rotation of objects in images with respect to the camera. Deep learning, as well as datasets containing a variety of pictured objects annotated with 3D pose, have led to great advances in this task [63, 56, 42, 67, 34].

However, they mainly focus on class-specific estimation for few categories, and they mostly evaluate on ground-truth bounding boxes. It is an issue when encountering objects of unseen classes or with out-of distribution appearance, for which no training data was available and no bounding box is given, which is a likely circumstance for robots in uncontrolled environments, e.g., outdoors vs in factories.

Our goal is to address this issue. Given training data for some known classes (images with bounding boxes of multiple objects, class labels and 3D pose annotations), we want to detect and estimate the 3D pose of objects of unknown
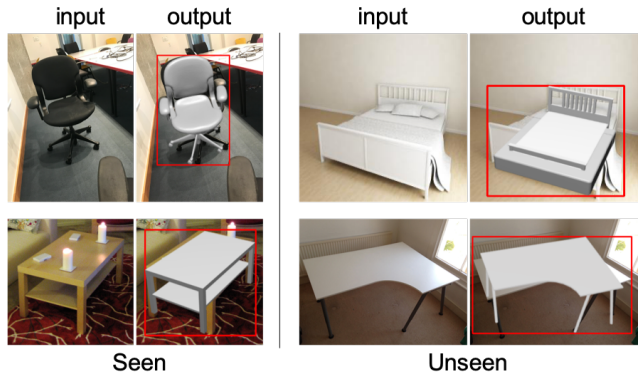

Figure 1. **Task Illustration.** Given an RGB image picturing an object, we aim to estimate its 3D pose (viewpoint) without knowing its class or shape. It is made possible by training a model for all objects in a class-agnostic way and applying it to objects of unseen classes having similar geometries as training objects and similar canonical frames, e.g., an unseen desk being similar to a seen table. (Red boxes are detections of a class-agnostic Mask R-CNN and the 3D models here are only used to visualize the pose.)

classes, given only an RGB image as input (Fig. 1), vs also using CAD models of objects as some methods do [75, 49].

This new task relies on two assumptions. First, it applies to unseen classes that share similarities with seen classes. For example, one may expect to orient an unseen bed when trained on seen chairs and sofas, but not a wrench.

The other assumption is that similar classes have a consistent canonical pose, i.e., have aligned similarities (Figs. 2 and 5). It is somehow a weak assumption, satisfied by all datasets we know of, probably because many objects are used consistently w.r.t. verticality, and feature a notion of left- and right-hand sides, or at least a main vertical symmetry plane, which is enough to define a "natural" canonical frame, possibly up to symmetry. Besides, if similar classes in a training set have inconsistent canonical poses, they can be normalized by a systematic rotation; no 3D shape is needed for that. In this first work, we only consider the general case, disregarding the different forms of symmetry.

Overview. To detect arbitrary objects and estimate their pose, although not in training data, we use a class-agnostic
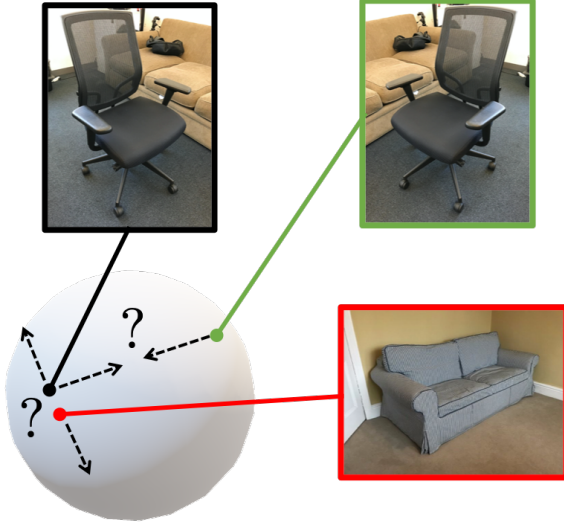
Figure 2. **Pose-Aware Contrastive Loss.** In usual self-supervised contrastive training, the network learns to pull together in feature space the **query** (e.g., chair) and a **positive** variant (e.g., flipped image), while pushing apart the **query** from **negatives** (different objects, e.g., sofa), ignoring pose information. Instead, we exclude flipped **positives**, whose pose actually differ from the **query**, and do not push apart **negatives** with similar poses (e.g., sofa).

approach for both object detection and pose estimation.

Approaches like [18, 78, 50] have already demonstrated the effectiveness of this setting. They detect 2D keypoints regardless of the class of the object, estimate 2D-3D keypoint correspondences, and use a PnP algorithm [32] to compute the pose. But besides being indirect, these methods need a suitable design of class-agnostic keypoints on various object geometries. In contrast, our approach estimates the 3D pose directly from the image embeddings, without any intermediate representation.

Others assume a 3D model of the object is given at test time (sometimes also at training time) [75, 47, 49, 8], either provided by a human or retrieved automatically by an algorithm, which is hard due to the image-shape domain gap and to the number of classes to discriminate [55, 40, 70], and it is limited by the database of possible 3D models to handle. In comparison, our method relies only on RGB images both at train and test time, without any CAD model as input.

To that end, we train a class-agnostic pose estimator by sharing weights across all object classes. And we propose a contrastive-learning approach to learn geometry-aware image embeddings that are optimized for pose estimation.

Recent contrastive-learning approaches create discriminative image features by learning to distinguish pairs of *identical objects* with different appearances thanks to a synthetic transformation (positives), from pairs of *different objects* (negatives). Inspired by image-level discrimination [23, 4, 73], we adapt the common contrastive loss InfoNCE [45] so that it discriminates poses rather than cat-

egories: we propose PoseNCE, a pose-aware contrastive loss that pushes away in latent space the image features of objects having different poses, ignoring the class of these objects as we aim at class-agnostic pose estimation (see Fig. 2). Besides, departing from the binary separation between positives and negatives in classical InfoNCE, PoseNCE takes into account the level of pose difference between two objects as a weighting term to reduce or stress the negativeness of a pair, regardless of the class (see Fig. 5).

Concretely, we use both an angle loss and a contrastive loss. We also curate the contrastive learning transformations to distinguish pose-variant data augmentations, e.g., horizontal flip, and pose-invariant data augmentations, e.g., color jittering. The former is used to actually augment the dataset, while the latter is used to create similar variants to construct positive and negative pairs. And rather than training from scratch on available datasets, that are relatively small, we initialize our network with a contrastive model trained on a large dataset in a self-supervised way.

Last, we propose a class-agnostic approach for both object detection *and* pose estimation. For this, we train a Mask R-CNN in a class-agnostic way for generic object detection, and pipeline it with our pose estimator, thus addressing the coupled problem of generic object detection and pose estimation for unseen objects. It is a more realistic setting w.r.t. existing class-agnostic pose estimation methods, that only evaluate in the ideal case of ground-truth bounding boxes.

Our main contributions are as follows:
- We define a new task suited for uncontrolled settings: class-agnostic object 3D pose estimation, possibly coupled and preceded by class-agnostic detection.
- We propose a contrastive-learning approach for class-agnostic pose estimation, which includes a pose-aware contrastive loss and pose-aware data augmentations.
- We report state-of-the-art results on 3 datasets, including against methods that also require shape knowledge. (And code will be made public upon publication.)

2. Related Work

Class-Specific Object Pose Estimation. While instance-level 3D object pose estimation has long been studied in both robotic and vision communities [25, 2, 28, 53, 59, 54, 72, 58, 44, 31], class-level pose estimation has developed more recently thanks to learning-based methods [56, 63, 62, 42, 30, 66, 18, 17, 67, 78, 61]. These methods can be roughly divided into two categories: pose estimation methods that regress 3D orientations directly [63, 56, 42, 66, 75], and keypoint-based methods that predict 2D locations of 3D keypoints [18, 17, 67, 78, 61].

Still, annotating 3D pose for objects in the wild is a tedious process of searching best-matching CAD models and aligning them to images [71, 70]. It does not scale to large numbers of objects and classes. While good performance is

achieved on supervised classes, generalizing beyond training data remains an important, yet under-explored problem.

Class-Agnostic Object Pose Estimation. To circumvent the problem of limited labeled object classes, a few class-agnostic pose estimation methods have recently been proposed [15, 18, 78, 75, 8, 50, 49]. In contrast to class-specific methods that build an independent prediction branch for each object class, agnostic methods estimate the object pose without knowing its class *a priori*, which is enabled by sharing model weights across all object classes during training.

[15] trains on multiple views of the same object instance on a turntable. [18, 50] use the 3D bounding box corners as generic keypoints for class-agnostic object pose estimation. However, [18] only reports performance on seen classes and [50] focuses on cubic objects with simple geometric shape. Instead of using a fixed set of keypoints for all objects, [78] propose a class-agnostic keypoint-based approach combining a 2D keypoint heatmap and 3D keypoint locations in the object canonical frame. These methods are robust on textured objects but fail with heavy occlusions and tiny or textureless objects. In contrast, our method ignores keypoints, directly infers a pose and is less sensitive to texturelessness.

Rather than relying only on RGB images, another group of class-agnostic pose estimation methods [75, 8, 49] use 3D models, in particular at test time to adapt to objects unseen at training time. [75] aggregates 3D shape and 2D image information for arbitrary objects, representing 3D shapes as multi-view renderings or point clouds. [8] proposes a lighter version of [75] encoding the 3D models into graphs using node embeddings [20]. [49] matches local images embeddings with local 3D embeddings, then use RANSAC and PnP algorithms to recover an object from a database of CAD models, and a pose. In contrast, we need no 3D shape, neither at training nor at testing time.

Pose Loss. 3D pose dissimilarity has been measured indirectly, e.g., with a distance on reprojected features such as keypoints (see above), or directly on pose parameters. In the latter case, the chosen representation and penalty may yield more or less artefacts due to, e.g., discontinuities in the parameterization (Euler angles, quaternions [79]), gimbal lock [19], anti-podal symmetry (quaternions), non-uniform parameter distributions, classification discretization [63, 56, 12], single-mode analysis as with regression [46, 48, 39], or parameter-space biases when penalizing with the L2-norm of the difference of pose parameters, including with the exponential twist representation [80]. We use a combination of classification and regression [42, 21, 38, 33] of Euler angles similar to [75] (offset regression from bin center), which better separates modes in case of pose ambiguities, but we penalize a geodesic distance on the unit sphere rather than the Euclidean distance of parameters, which does not have dimensional biases.

Contrastive Learning. Instead of designing pretext tasks for unsupervised learning [10, 43, 77, 16], powerful image features can be learned by contrasting positive and negative pairs [69, 45, 60, 41, 3, 23, 6, 4, 5, 29]. Among the various forms of the contrastive loss function [22, 64, 26, 69, 45], InfoNCE [45] has become a standard pick in many methods.

While most contrastive-learning approaches work in the unsupervised setting, [29] operates with full supervision. Considering the class label of training examples, features belonging to the same class are pulled together while features from different classes are pushed apart.

Similar to [29], we also propose a contrastive loss that works in the fully-supervised setting. However, instead of focusing on semantic label information, we design it for our geometric task, taking into account the pose distance between different examples. Moreover, we also curate data augmentations as advocated in [73], leaving out those that would be harmful for our pose estimation task.

Besides requiring 3D shapes at training time and operating on RGB-D data, [1] is not pose-aware: in the InfoNCE spirit, it creates positive pairs from the same known shape model and negative pairs from known different shapes, ignoring pose. Besides, it favors features whose L2-distance is *equal to* their pose L2-distance, which is a heavy burden for feature learning, especially for objects with large shape variations. In comparison, we simply contrast features w.r.t. pose dissimilarity. [68], which operates in a class-specific way and also requires known 3D shapes or at least multiple views or renderings of each object, uses a triplet loss whose formulation can be related to our more general PoseNCE loss, but it does not take into account the level of pose dissimilarity nor pose-aware data augmentation.

Coupled Detection and Pose Estimation. Very few works consider the realistic scenario of detecting unknown objects in images *and* inferring their pose. [49] trains a class-agnostic Mask R-CNN and pipelines it with a pose estimator, as we do, but it applies to industrial objects and requires knowing the 3D shapes, including for novel instances. [15], which trains with objects on a turntable, does not do any detection but somehow also applies to ImageNet, i.e., with well-centered, single-object images. None of these methods is thus applicable to objects in the wild. And although [18] predicts a 3D box size (not location) for PnP reprojection, it operates on ground-truth 2D bounding boxes. We can only compare in the class-specific detection and pose estimation setting [66, 17] and, in the class-agnostic setting, against methods also requiring an input 3D shape [75].

3. Method

Given an RGB image I containing an object at a given (known or detected) image location, we aim to estimate the 3D pose $\mathbf{R}$ of the object with no prior knowledge of its class
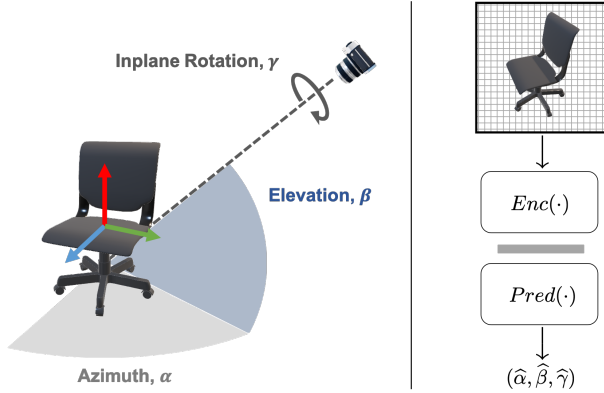
Figure 3. **Pose Parameters** (left). **Network Architecture** (right): from an image crop, the encoder Enc produces an embedding, which is given to the predictor $Pred$ to produce pose angles.

or shape. To that end, we crop the image region containing the object, encode it to produce class-agnostic features, from which the object 3D pose is directly predicted (Fig. 3).

3D Pose Parameterization. To predict the 3D rotation matrix $\mathbf{R}$ of the pictured object, we decompose it into three Euler angles as in [56, 75]: azimuth α, elevation β, and in-plane rotation γ, with $\alpha, \gamma \in [-\pi, \pi)$ and $\beta \in [-\pi/2, \pi/2]$.

Recent work on pose estimation shows a higher performance with a more continuous formulation (cf. [79]) mixing angular bin classification and within-bin offset regression [75, 74]. Concretely, we split each Euler angle $\theta \in \{\alpha, \beta, \gamma\}$ uniformly into discrete bins i of size B ($= \pi/12$ in our experiments). The network outputs bin classification scores $p_{\theta,i} \in [0, 1]$ and offsets $\delta_{\theta,i} \in [0, 1]$ within the bin.

Angle Loss. We use a cross-entropy loss for angle bin classification and a smooth-L1 loss for bin offset regression:

$$\mathcal{L}_{\text{ang}} = \sum_{\theta \in \{\alpha, \beta, \gamma\}} \mathcal{L}_{\text{cls}}(\text{bin}_\theta, p_\theta) + \lambda\, \mathcal{L}_{\text{reg}}(\text{offset}_\theta, \delta_\theta) \quad (1)$$

where bin_θ is the ground-truth bin and offset_θ is the offset for angle θ. The relative weight λ is set to 1 in our experiments. The final prediction for angle θ is obtained as:

$$\widehat{\theta} = (j + \delta_{\theta,j})B \quad \text{with} \quad j = \arg\max_i p_{\theta,i} \quad (2)$$

where $i \in [-12..11]$ for α, γ, and $i \in [-6..5]$ for β. The angle loss is complemented by a contrastive loss (cf. Sect. 3).

Network Architecture. The architecture of our network is depicted in Figure 3 (right). It consists of two modules: an image encoder $Enc(\cdot)$ and a pose predictor $Pred(\cdot)$.

For feature extraction, we use a standard CNN, namely ResNet-50. We crop the input image to the targeted object and pass it through the encoder network until the max-pooling layer. It provides a 2048-dimension feature vector.
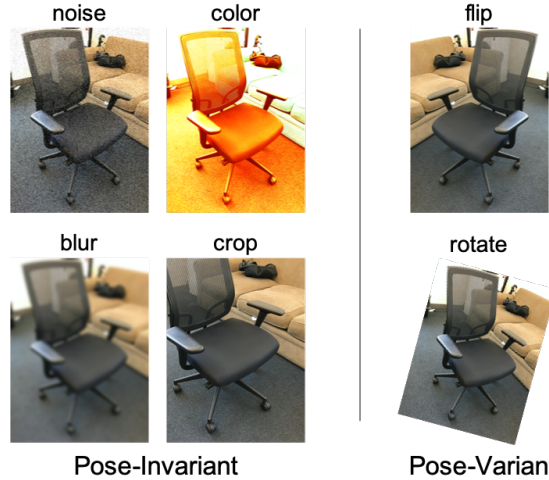

Figure 4. **Pose-Aware Data Augmentations:** while *pose-invariant* data augmentations do not alter the pose, *pose-variant* augmentations modify it and cannot be used as positives.

We then pass the image embedding through the pose predictor, which is a multi-layer perceptron (MLP) with 3 hidden layers of size 800-400-200, each followed by batch normalization and ReLU activation. Contrary to class-specific methods [56, 63, 34, 42] that use one prediction branch per class, we use a single prediction branch for all objects.

Contrastive Features. Datasets of images with pose annotations are scarce and small. (One of the reasons is probably that pose is much harder to annotate than class, especially for images in the wild.) It makes it difficult to learn a high-quality pose estimator. Rather than learning a network from scratch, as most other methods do, or from an initial ImageNet classifier, whose bias is not particularly suited for pose estimation, we initialize our predictor using a pre-trained contrast-based network [6]. We show that it plays a significant role in our high performance (cf. Sect. 4.2).

Self-Supervised Contrastive Loss. In self-supervised contrastive learning [4, 23], the contrastive loss serves as an unsupervised objective for training an image encoder that maximizes agreement between different transformations of the same sample, while minimizing the agreement with other samples. Concretely, we consider a batch $(I_k)_{k \in [1..N]}$ of training samples, transformed into $(\tilde{I}_k)_{k \in [1..N]}$ by data augmentation, and encoded as $f_k = Enc(\tilde{I}_k)$. For any index $k^+ \in [1..N]$, we consider an alternative augmentation $\tilde{I}_q$ of I_{k^+} (the query), with embedding $f_q = Enc(\tilde{I}_q)$, and we separate the *positive pair* (q, k^+) from the *negative pairs* $(q, k^-)_{k^- \in [1..N] \setminus \{k^+\}}$ with the following InfoNCE loss:

$$\mathcal{L}_{\text{infoNCE}} = -\log \frac{\exp(f_q \cdot f_{k^+} / \tau)}{\sum_{k \in [1..N]} \exp(f_q \cdot f_k / \tau)} \quad (3)$$

where τ is a temperature parameter [4] (0.5 in experiments).

Figure 5. **Pose-Aware Contrast.** Instead of treating all negatives (right images) equally, as for positives (left images), we give more weight to **negatives** with a large rotation (rightmost) and less to those with a small rotation, *regardless of their semantic class*.

Pose-Aware Data Augmentations.   As illustrated in Figure 4, we divide data augmentations into two categories. *Pose-invariant* augmentations transform the image without changing the 3D pose of objects: color jittering, blur, crop, etc. On the contrary, *pose-variant* augmentations change the 3D pose at the same time: rotation and horizontal flip. More precisely, an image rotation of angle ϕ corresponds to an in-plane rotation of angle $\gamma + \phi$ for the object, and a horizontal image flip corresponds to a change of sign of azimuth α and in-plane rotation γ. (We assume mirror-imaged objects are realistic objects with identical canonical frame.) In our experiments, ϕ varies in $[-15°, 15°]$ since 95% of the images in Pascal3D+ [71] fall in this range.

Unlike self-supervised learning methods that make use of all data augmentation techniques at the same time, we distinguish two augmentation times. At *batch creation time*, we only apply pose-variant data augmentations, i.e., a small rotation or a horizontal flip, and update the object pose information accordingly. At *contrast time*, i.e., when creating *positive* and *negative* pairs, we only apply pose-invariant data augmentations. The latter is motivated by [73]: a blind use of any data augmentation could be harmful.

Pose-Aware Contrastive Loss.   The contrastive loss in Eq. (3) is designed for unsupervised learning with no annotation involved. While efficient for learning instance-discriminative image embeddings, it is not particularly suited for contrasting geometric cues towards pose estimation: the query embedding is contrasted away from the embeddings of negative samples even if their pose is identical or similar to the pose of the query object. While the case of

different views of identical or similar instances can be disregarded in usual contrastive learning because of its practical rarity, similar and even identical poses are common in a single batch. What we want, instead of contrasting the object semantics, is to learn pose-variant image features.

We thus introduce a new pose-aware contrastive loss, illustrated in Figure 5. It takes into account the level of sample negativeness: the larger the pose difference, the higher the weight in the loss. (There is thus no need to define a notion of *pose similarity*.) Concretely, for each pair (q, k), we compute a normalized distance $\mathrm{d}(\mathbf{R}_q, \mathbf{R}_k) \in [0, 1]$ between the associated 3D pose rotations $\mathbf{R}_q, \mathbf{R}_k$ and we use it as a weight in our *pose-aware contrastive loss PoseNCE*:

$$\mathcal{L}_{\mathrm{poseNCE}} = -\log \frac{\exp(f_q \cdot f_{k+}/\tau)}{\sum_{k \in [1..N]} \mathrm{d}(\mathbf{R}_q, \mathbf{R}_k) \exp(f_q \cdot f_k/\tau)} \tag{4}$$

with $\mathrm{d}(\mathbf{R}_q, \mathbf{R}_{k+}) = 0$ as $\mathbf{R}_q = \mathbf{R}_{k+}$. Our distance is defined as the normalized angle difference between the rotations, which is akin to a geodesic distance on the unit sphere:

$$\mathrm{d}(\mathbf{R}_q, \mathbf{R}_k) = \Delta(\mathbf{R}_q, \mathbf{R}_k)/\pi \quad \text{with}$$
$$\Delta(\mathbf{R}_q, \mathbf{R}_k) = \arccos\left(\frac{\mathrm{tr}(\mathbf{R}_q^{\mathrm{T}} \mathbf{R}_k) - 1}{2}\right) \tag{5}$$

Following [76], PoseNCE can be seen as softer or smoother version of InfoNCE [45], which itself is softer than hard pairwise or triplet losses. It can also be seen as a soft treatment of easy and hard negatives [65].

Last, the total loss adds the angle and contrastive losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{ang}} + \kappa \mathcal{L}_{\mathrm{poseNCE}} \tag{6}$$

The relative weight κ is set to 1 in our experiments.

4. Experiments

In this section, we introduce our experimental setup, analyze results on three commonly-used datasets, provide an ablation study and discuss the limitations. The supplementary material include details on datasets, splits, training, implementation, and more classwise results. It also provides a study on the temperature parameter, and more visual results.

Datasets. We experiment with 3 commonly-used datasets for object pose estimation. Pascal3D+ [71] contains the 12 rigid classes of PASCAL VOC 2012 [13], with approximate poses. ObjectNet3D has slightly more accurate poses for 100 object classes. Pix3D [57] features 9 classes, two of them ('tool' and 'misc') not appearing in Pascal3D+ nor ObjectNet3D, with even more accurate poses.

Evaluation Metrics. Unless otherwise stated, ground-truth object bounding boxes are used in all experiments. We compute the most common metrics [63, 56]: Acc30 is the percentage of estimations with rotation error less than 30 degrees; MedErr is the median angular error in degrees.

	Method	w/ 3D	PnP	Backbone	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
MedErr →	Grabner et al. [18]		✓	ResNet-50	10.9	**12.2**	23.4	9.3	3.4	5.2	15.9	16.2	12.2	11.6	6.3	11.2	11.5
	StarMap [78]		✓†	ResNet-18 *	10.1	14.5	30.3	9.1	3.1	6.5	11.0	23.7	14.1	11.1	7.4	13.0	12.8
	3DPoseLite [8]	✓		ResNet-18	–	–	–	–	–	–	–	–	–	–	–	–	13.4
	PoseFromShape [75]	✓		ResNet-18	11.1	14.4	22.3	7.8	3.2	5.1	12.4	13.8	11.8	**8.9**	5.4	8.8	10.4
	PoseFromShape [75]	✓		ResNet-50	10.9	14.5	21.5	7.5	3.3	5.0	11.2	14.2	11.6	9.2	5.5	9.0	10.3
	PoseContrast (ours)			ResNet-50	**10.0**	13.6	**18.3**	**7.2**	**2.8**	**4.6**	**9.8**	**9.2**	**11.5**	11.0	5.6	11.6	**9.6**
Acc30 ↑	Grabner et al. [18]		✓	ResNet-50	0.80	0.82	0.57	0.90	**0.97**	0.94	0.72	0.67	0.90	0.80	**0.82**	**0.85**	0.81
	StarMap [78]		✓†	ResNet-18 *	0.82	**0.86**	0.50	0.92	**0.97**	0.92	0.79	0.62	0.88	**0.92**	0.77	0.83	0.82
	3DPoseLite [8]	✓		ResNet-18	0.80	0.82	0.58	0.93	0.96	0.92	0.77	0.57	0.88	0.82	0.80	0.79	0.80
	PoseFromShape [75]	✓		ResNet-18	0.83	**0.86**	0.60	**0.95**	0.96	0.91	0.79	0.67	0.85	0.85	**0.82**	0.82	0.83
	PoseFromShape [75]	✓		ResNet-50	0.83	**0.86**	0.61	**0.95**	0.96	0.92	0.80	0.67	0.84	0.82	**0.82**	0.83	0.83
	PoseContrast (ours)			ResNet-50	**0.85**	0.84	**0.64**	0.94	**0.97**	**0.95**	**0.86**	**0.71**	**0.91**	0.90	**0.82**	**0.85**	**0.85**

Table 1. **3D Pose Estimation of Class-Agnostic Methods on Pascal3D+ [71] (all classes seen).** All methods are evaluated with ground-truth bounding boxes. *The authors observe similar or worse performance with ResNet-50 [78]. †StarMap actually obtains the rotation by solving for a similarity transformation between the image frame and world frame, weighting keypoint distances by the heatmap value.

	Method	w/ 3D	2D Bbox	46 tool	61 misc	130 b-case	166 w-drobe	297 desk	394 bed	739 table	1092 sofa	2894 chair	mean	5818 global
Acc30 ↑	3DPoseLite [8]	✓	GT	**0.09**	0.10	0.62	0.57	0.66	0.58	0.40	0.94	0.50	0.50	0.58
	PoseFromShape [75] *	✓	GT	0.07	**0.28**	0.71	0.65	0.71	0.54	0.53	0.94	0.79	0.58	0.75
	PoseContrast (ours)		GT	**0.09**	0.18	**0.81**	**0.68**	**0.78**	**0.68**	**0.54**	**0.97**	**0.86**	**0.62**	**0.80**
	PoseFromShape [75]	✓	pred	0.07	**0.23**	0.68	0.55	0.71	0.51	**0.53**	0.93	0.77	0.55	0.73
	PoseContrast (ours)		pred	**0.09**	0.16	**0.72**	**0.58**	**0.77**	**0.65**	**0.53**	**0.97**	**0.85**	**0.59**	**0.79**

Table 2. **Cross-dataset 3D Pose Estimation of Class-Agnostic Methods on Pix3D [57].** The methods are trained on the Pascal3D+ [71] train set and tested on Pix3D, where 6 classes are unseen (novel) and 3 classes are already seen (table, sofa, chair). As the classes are heavily unbalanced, we also report the global average (instance-wise rather than class-wise). We consider two kinds of input: ground-truth (GT) 2D object bounding box, and predicted (pred) bounding box by a class-agnostic Mask R-CNN detector. *3DPoseLite [8] reports much worse figures for PoseFromShape [75] than what we got here with our own runs, probably due to a wrong experimental setting.

4.1. Main Results

Upper Bound: Performance on Seen Classes. To check our performance before considering unseen classes, we first evaluate on seen classes. We follow the common protocol [18, 78] to train our model on the train split of Pascal3D+ [71] and test it on the val split. Both train and val splits share the same 12 object classes. In Table 1, we compare with state-of-the-art class-agnostic object pose estimation methods, using ground-truth bounding boxes. As we leverage on contrast-based features, we use the available MOCOv2 pre-trained ResNet-50 model [6]. But no MOCO pre-trained ResNet-18 is available for comparison (see also Table 4).

For most categories, our class-agnostic approach consistently outperforms other class-agnostic methods [18, 78], including those that leverage a 3D shape as additional input [75, 8]. In particular, our direct pose estimation method achieves a clear improvement for the class 'chair', which features a higher variety and geometric complexity than other classes. It suggests that keypoint-based methods as [18, 78] may fail to capture detailed shape information for accurate 2D-3D correspondence prediction, while model-based methods as [75, 8] do not construct powerful-enough embeddings despite their access to an actual 3D shape.

Overall, we achieve the best average performance in both metrics. In fact, we even outperform class-specific methods [37, 63, 42, 56, 52, 18, 38] except one [34], that reaches MedErr 9.2° and Acc30 88%, while we get 9.6° and 85%.

Stressing Class Agnosticism: Cross-Dataset Evaluation. To show our generalization ability, we follow the recent protocol proposed in [8] and conduct a cross-dataset object pose estimation. We train on the 12 classes of Pascal3D+ (that has approximate pose annotations) and test on the 9 classes of Pix3D (with accurate poses), where only 3 classes coincide with Pascal3D+. Hence, 6 classes are totally unseen while 3 are already seen. Besides, methods that report cross-dataset results on Pix3D usually assume that ground-truth bounding boxes and 3D object models are given for testing [75, 8]. We compare here in that same setting (see below for using detected objects). Results are in Table 2.

For the three seen classes ('table', 'sofa', 'chair'), our method outperforms all compared methods. It is consistent with results on Pascal3D+ (Table 1), including for the difficult class 'chair'. More interestingly, we achieve a significantly better performance for certain unseen classes, even though there is no prior knowledge of the testing objects for our network. As expected, it applies to unseen classes that share a similar shape and canonical frame as seen classes, e.g., 'desk' and 'table'. By sharing weights across different classes during training, our class-agnostic pose estimation network learns a direct mapping from image embeddings to 3D poses and can apply to unseen objects when they have a similar shape as the training objects. But when the target
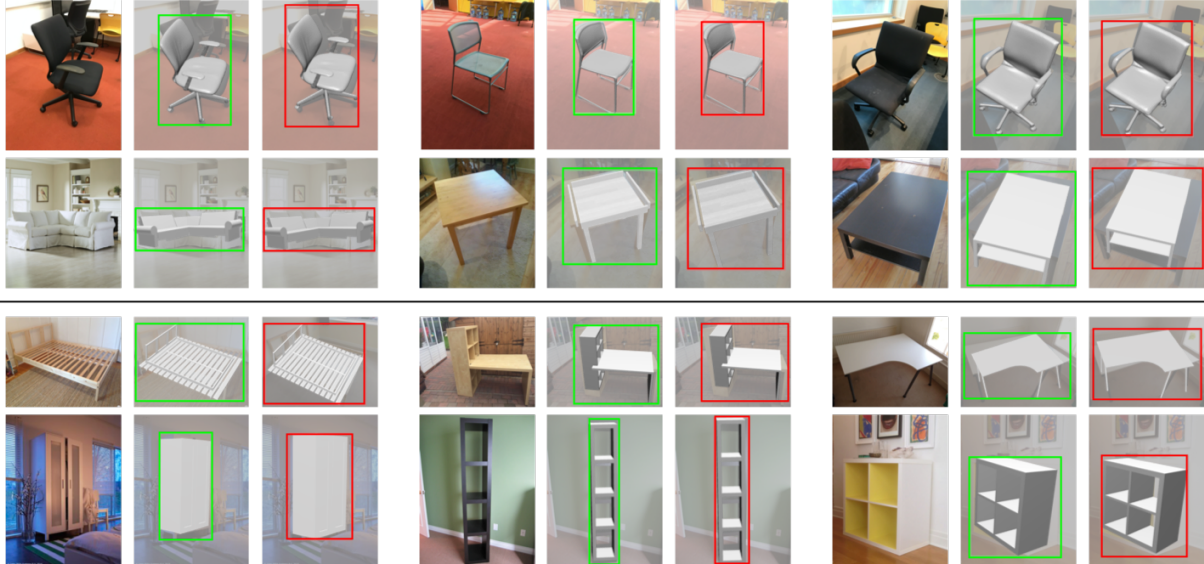
Figure 6. **Qualitative Results on Pix3D.** For each sample, we first plot the original image, then we visualize the pose prediction obtained from the ground-truth bounding box and the detected bounding box, respectively. The top two rows show results for seen classes that intersect with training data in Pascal3D+ ('chair', 'sofa', 'table'), while the bottom two rows show results for novel classes. Note that the 3D CAD object models are only used here for pose visualization purpose; our approach does not rely on them for object pose prediction.

Method	Setting	w/ 3D	Acc30 ↑	MedErr ↓
StarMap [78]	no-shot		0.44	55.8
PoseContrast (ours)	no-shot		**0.55**	**42.6**
PoseFromShape [75]	no-shot	✓	0.62	42.0
MetaView [61]	10-shot		0.48	43.4
PoseContrast (ours)	10-shot		**0.60**	**38.7**
FSDetView [74]	10-shot	✓	0.63	32.1

Table 3. **Few-Shot Object Pose Estimation on ObjectNet3D [70].** We report results on the 20 novel classes of ObjectNet3D as defined in [78, 61]. We compare both in no-shot and 10-shot settings.

objects possess a geometry widely differing from the training ones, such as 'tool' and 'misc', our purely image-based method usually fails; PoseFromShape does a bit better because it leverages a shape model, but accuracy remains poor. Some failure cases of our method can be seen in Figure 8.

Few-Shot Regime on ObjectNet3D. We first follow the no-shot setting proposed in [78]: we train on 80 seen classes and test on 20 unseen (novel) classes, cf. Table 3 (top). Compared to PoseFromShape [75], both our approach and StarMap [78] do not rely on 3D object models at test time, but exploit geometric similarities shared between seen and unseen classes. However, while StarMap struggles to predict precise 3D object coordinates and depth values, our simpler network achieve a higher performance.

We then evaluate in the 10-shot setting as in [61, 74]: the networks are first trained on the 80 seen classes, and then fine-tuned with a few labeled images from the 20 novel classes. Results are shown in Table 3 (bottom). Compared to MetaView [61], that relies on class-specific keypoint pre-

diction, we again find that our approach can obtain a better performance by sharing weights across all object classes.

In both settings, the best performing methods additionally use 3D object models [75, 74]. Such a prior knowledge of the geometry makes sense, especially for unseen objects with shapes widely different shapes from training classes. Yet, our method nonetheless achieves promising results on these unseen classes, even compared to using a 3D model.

Class-Agnostic Object Detection and Pose Estimation. To evaluate the coupling of generic object detection *and* generic pose estimation, we train a Mask R-CNN with backbone ResNet-50 on COCO in a class-agnostic way, merging all classes into a single one, then apply it directly on Pix3D without fine-tuning. All 9 Pix3D classes can thus be detected by our network, whether they are in COCO or not. To compare with other methods, we adopt the well-established metric $Acc_{D_{0.5}}$ [17], that computes the percentage of objects for which the Intersection-over-Union (IoU) between the ground-truth and the predicted boxes is larger than 50% (ignoring false positives), to focus on objects of interest.

Compared to class-specific methods that predict object localization together with their class [66, 17], our class-agnostic detector localizes objects without classifying them into categories, relying less on semantic information for prediction. As shown in Figure 7, it provides a better detection accuracy and, more importantly, it enables the efficient detection of objects that are not included in COCO classes.

Qualitative results are shown in Figure 6. We find that both our object detector and our pose estimator can generalize to unseen objects (two bottom rows). Quantitative results are given in Table 2. We observe that our object pose
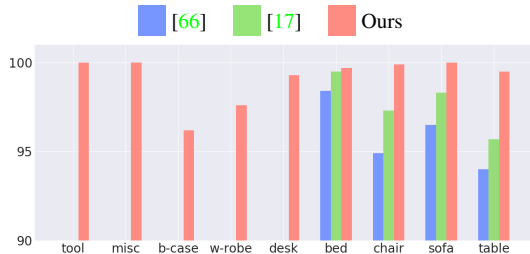
Figure 7. **Object Detection on Pix3D.** Results are given in $Acc_{D_{0.5}}$ as defined in [17]. We compare with two methods [66, 17] that train a class-specific Mask R-CNN on COCO, then fine-tune on a subset of Pix3D containing the same classes as COCO. In contrast, our agnostic Mask R-CNN is only trained on COCO and can generalize to classes not included in COCO.

Initialization	Method	Pre-train data	Epochs	Acc30 ↑	MedErr ↓
from scratch	random	—	15*	0.76	12.8
from scratch	random	—	75	0.81	11.9
supervised	classification	ImageNet	15	0.83	10.7
unsupervised	SimCLR [4]	ImageNet	15	0.83	11.0
unsupervised	SWAV [3]	ImageNet	15	0.84	10.2
unsupervised	MOCOv1 [23]	ImageNet	15	0.84	10.3
unsupervised	MOCOv2 [6]	ImageNet	15	**0.85**	**9.6**

Table 4. **Network Initializations Evaluated on Pascal3D+.** We compare different initializations, training until convergence (*except for the first line), showing the number of epochs required.

estimation, evaluated using predicted boxes, can outperform existing methods evaluated using ground-truth boxes only. This promising results suggests it is possible to develop autonomous systems that perform class-agnostic object detection and pose estimation on unknown objects in the wild.

4.2. Ablation Study

Pre-trained Features. We initialize our image encoder network with MOCOv2 [6] to transfer rich features to the down-stream task of object pose estimation. Yet, other pre-trained features could be used [23, 4, 3], or learning from scratch. Table 4 reports results with various initializations.

Learning from scratch is suboptimal, probably due to the small dataset size, hence the relevance of using a pre-trained network. Convergence is also 5 times faster. Also, contrast-based pre-trained networks tend to perform best. In comparison, [18] also pre-trains on ImageNet while [75] has similar results with or without ImageNet pre-training. [78] uses a ResNet-18 trained from scratch for its keypoint-based 2-stack hourglass network. Pre-training is not known for [8].

Adding a Contrastive Loss. Table 5 shows the relevance of adding a contrastive loss to the angle loss for pose estimation. However, adding the original InfoNCE loss only brings a minor improvement. A larger performance gap is obtained with our pose-aware contrastive loss of Eq. (4).

Alternative Pose Distances. Our contrastive loss relies on a distance between two poses $d(\mathbf{R}_q, \mathbf{R}_k)$, defined as the

Loss	$d(\mathbf{R}_i, \mathbf{R}_{k-})$	Pascal3D+ Acc30	Pascal3D+ MedErr	Pix3D Acc30	Pix3D MedErr
$\mathcal{L}_{ang}$	N/A	0.83	10.2	0.56	36.1
$\mathcal{L}_{ang}+\mathcal{L}_{infoNCE}$	1	0.83	10.0	0.57	35.2
$\mathcal{L}_{ang}+\mathcal{L}_{poseNCE}$	$(\Delta(\mathbf{R}_i, \mathbf{R}_{k-})/\pi)^{\frac{1}{2}}$	0.84	9.8	0.61	31.3
$\mathcal{L}_{ang}+\mathcal{L}_{poseNCE}$	$(\Delta(\mathbf{R}_i, \mathbf{R}_{k-})/\pi)^2$	**0.85**	10.0	**0.62**	32.6
$\mathcal{L}_{ang}+\mathcal{L}_{poseNCE}$	$\Delta(\mathbf{R}_i, \mathbf{R}_{k-})/\pi$	**0.85**	**9.6**	**0.62**	**29.3**

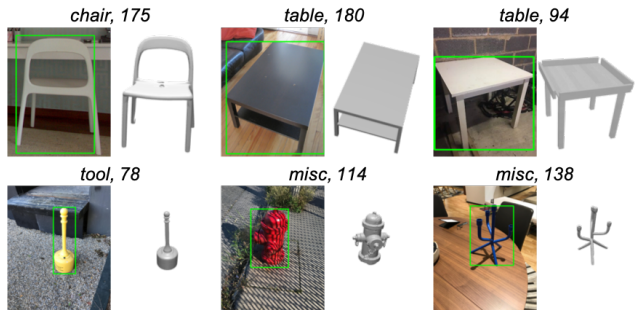Table 5. **Adding a Contrastive Loss, Alternative Pose Distances.**

Figure 8. **Visualization of Failure Cases.** We show input image crops and predicted object poses, with class name and prediction error displayed at the top. Common failures come from ambiguous appearances of symmetrical objects, or shapes out of distribution.

normalized rotation difference $\Delta(\mathbf{R}_q, \mathbf{R}_k)/\pi$. Table 5 compares this definition to two variants: square and square root of this distance. All three perform better than the InfoNCE loss of Eq. (3), but the linear distance performs best.

4.3. Discussion

To understand where prediction errors come from, we show some common failure cases in Fig. 8. Many mistakes originate from symmetric objects, as their symmetry is neither modeled explicitly nor taken into account for metric evaluation (e.g., defining or measuring a table orientation by 180°). See a more detailed study in the supp. mat.

In fact, a few other works specifically treat symmetries [27, 11, 1, 7, 51]. It is largely orthogonal to our proposal and left for future work. Note that it concerns only about 10-15% of the classes (e.g., table, bottle in Pascal3D+) and has little impact here as annotations generally assume the orientation with the smallest angle(s) w.r.t. the viewpoint.

Our approach also fails on unseen objects with shapes differing completely from training ones, e.g., 'tool' and 'misc' of Pix3D. But it actually is a problem to all the RGB-only class-agnostic methods [78, 18], not specifically to ours, as generalizing towards unseen objects mainly relies on similarities. Even shape-based methods [75, 8], that exploit extra shape knowledge, nevertheless also struggle to get a good performance on these two classes.

5. Conclusion

We presented a new class-agnostic object pose estimation approach based on a pose-aware contrastive learning.

Our network is trained end-to-end, leveraging on existing unsupervised contrastive features. We empirically show on various benchmarks that our method constitutes a strong baseline for class-agnostic object pose estimation. We also pave the way to more practical applications by successfully combining it with a class-agnostic object detector.

References

[1] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, R. Kouskouridas, and Tae-Kyun Kim. Pose guided RGBD feature learning for 3D object pose estimation. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3, 8

[2] Eric Brachmann, Alexander Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 8

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, 2020. 2, 3, 4, 8

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning. *ArXiv*, 2020. 3, 4, 6, 8, 15

[7] Enric Corona, Kaustav Kundu, and Sanja Fidler. Pose estimation for objects with rotational symmetry. In *International Conference on Intelligent Robots and Systems (IROS)*, 2018. 8

[8] Meghal Dani, Karan Narain, and Ramya Hebbalaguppe. 3DPoseLite: A Compact 3D Pose Estimation Using Node Embeddings. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2, 3, 6, 8

[9] Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 12

[10] C. Doersch, A. Gupta, and Alexei A. Efros. Unsupervised Visual Representation Learning by Context Prediction. *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[11] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Härtinger, and Carsten Steger. Introducing MVTec ITODD — a dataset for 3D object recognition in industry. In *International Conference on Computer Vision Workshops (ICCVW)*, 2017. 8

[12] Mohamed Elhoseiny, Tarek El-Gaaly, Amr Bakry, and Ahmed M. Elgammal. A comparative analysis and study of multiview CNN models for joint object categorization and pose estimation. In *International Conference on Machine Learning (ICML)*, 2016. 3

[13] M. Everingham, L. Gool, C. K. Williams, J. Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 2009. 5, 12

[14] Chelsea Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning (ICML)*, 2017. 14

[15] Yunhao Ge, Jiaping Zhao, and Laurent Itti. Pose augmentation: Class-agnostic object pose transformation for object recognition. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 3

[17] A. Grabner, P. Roth, and Vincent Lepetit. GP2C: Geometric Projection Parameter Consensus for Joint 3D Pose and Focal Length Estimation in the Wild. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 7, 8

[18] Alexander Grabner, Peter M. Roth, and Vincent Lepetit. 3D pose estimation and 3D model retrieval for objects in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 6, 8

[19] F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, Mar. 1998. 3

[20] Aditya Grover and J. Leskovec. node2vec: Scalable Feature Learning for Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. 3

[21] Rıza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. DenseReg: Fully convolutional dense shape regression in-the-wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[22] Raia Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 3

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4, 8

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 12

[25] 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 *Asian Conference on Computer Vision (ACCV)*, 2012. 2

[26] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua

Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[27] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *ECCV Workshops (ECCVw)*, 2016. 8

[28] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[29] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[30] Abhijit Kundu, Yin Li, and James M. Rehg. 3D-RCNN: Instance-Level 3D Object Reconstruction via Render-and-Compare. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[31] Yann Labbé, J. Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[32] Vincent Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision (IJCV)*, 2008. 2

[33] Chi Li, Jin Bai, and Gregory D. Hager. A unified framework for multi-view multi-class object pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[34] Shuai Liao, Efstratios Gavves, and Cees G. M. Snoek. Spherical Regression: Learning Viewpoints, Surface Normals and 3D Rotations on N-Spheres. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 4, 6

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 12

[36] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, 2014. 12

[37] S. Mahendran, H. Ali, and R. Vidal. 3D Pose Regression Using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 6

[38] Siddharth Mahendran, Haider Ali, and René Vidal. A Mixed Classification-Regression Framework for 3D Pose Estimation from 2D Images. In *British Machine Vision Conference (BMVC)*, 2018. 3, 6

[39] Francisco Massa, Renaud Marlet, and Mathieu Aubry. Crafting a multi-task CNN for viewpoint estimation. In *British Machine Vision Conference (BMVC)*, 2016. 3

[40] Francisco Massa, Bryan C. Russell, and Mathieu Aubry. Deep Exemplar 2D-3D Detection by Adapting from Real to Rendered Views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[41] I. Misra and L. V. D. Maaten. Self-Supervised Learning of Pretext-Invariant Representations. *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[42] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D Bounding Box Estimation Using Deep Learning and Geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 4, 6

[43] M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[44] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[45] A. Oord, Y. Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, 2018. 2, 3, 5

[46] M. Osadchy, Y. LeCun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *The Journal of Machine Learning Research (JMLR)*, 8:1197–1215, 2007. 3

[47] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. LatentFusion: End-to-End Differentiable Reconstruction and Rendering for Unseen Object Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[48] H. Penedones, R. Collobert, F. Fleuret, and D. Grangier. Improving object classification using pose information. Technical report, Idiap Research Institute, 2011. 3

[49] Giorgia Pitteri, Aurelie Bugeau, Slobodan Ilic, and Vincent Lepetit. 3D object detection and pose estimation of unseen objects in color images with local surface embeddings. In *Asian Conference on Computer Vision (ACCV)*, 2020. 1, 2, 3

[50] Giorgia Pitteri, S. Ilic, and Vincent Lepetit. CorNet: Generic 3D Corners for 6D Pose Estimation of New Objects without Retraining. In *IEEE International Conference on Computer Vision Workshops (ICCVw)*, 2019. 2, 3

[51] Giorgia Pitteri, Michaël Ramamonjisoa, Slobodan Ilic, and Vincent Lepetit. On object symmetries and 6d pose estimation from images. In *International Conference on 3D Vision (3DV)*, 2019. 8

[52] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. Deep Directional Statistics: Pose Estimation with Uncertainty Quantification. In *European Conference on Computer Vision (ECCV)*, 2018. 6

[53] Mahdi Rad and Vincent Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[54] M. Rad, Markus Oberweger, and Vincent Lepetit. Feature Mapping for Learning Fast and Accurate 3D Pose Inference from Synthetic Images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[55] Hang Su, Subhransu Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view Convolutional Neural Networks for 3D Shape Recognition. *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[56] Hao Su, Charles Ruizhongtai Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 3, 4, 5, 6

[57] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 6, 12, 13, 15, 16

[58] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[59] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[60] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[61] Hung-Yu Tseng, Shalini De Mello, J. Tremblay, Sifei Liu, Stan Birchfield, Ming-Hsuan Yang, and J. Kautz. Few-Shot Viewpoint Estimation. In *British Machine Vision Conference (BMVC)*, 2019. 2, 7, 13, 14

[62] Shubham Tulsiani, João Carreira, and Jitendra Malik. Pose Induction for Novel Object Categories. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[63] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 3, 4, 5, 6

[64] X. Wang and A. Gupta. Unsupervised Learning of Visual Representations Using Videos. *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[65] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[66] Yaming Wang, Xiao Tan, Yi Yang, Xiao Liu, Errui Ding, Feng Zhou, and Larry S Davis. 3D pose estimation for fine-grained object categories. In *European Conference on Computer Vision Workshop (ECCVw)*, 2018. 2, 3, 7, 8

[67] Wang, He and Sridhar, Srinath and Huang, Jingwei and Valentin, Julien and Song, Shuran and Guibas, Leonidas J. Normalized object coordinate space for category-level 6D object pose and size estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[68] Paul Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[69] Zhirong Wu, Yuanjun Xiong, S. Yu, and D. Lin. Unsupervised Feature Learning via Non-parametric Instance Discrimination. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[70] 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 *European Conference on Computer Vision (ECCV)*, 2016. 2, 7, 12, 13, 14

[71] 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 (WACV)*, 2014. 2, 5, 6, 12, 13, 15, 17

[72] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems (RSS)*, 2018. 2

[73] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What Should Not Be Contrastive in Contrastive Learning. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 5

[74] Yang Xiao and Renaud Marlet. Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild. In *European Conference on Computer Vision (ECCV)*, 2020. 4, 7, 13, 14

[75] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3D objects. In *British Machine Vision Conference (BMVC)*, 2019. 1, 2, 3, 4, 6, 7, 8, 14

[76] Haozhi Zhang, Xun Wang, Weilin Huang, and Matthew R. Scott. Rethinking deep contrastive learning with embedding memory, 2021. arXiv preprint 2103.14003. 5

[77] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[78] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. StarMap for Category-Agnostic Keypoint and Viewpoint Estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 6, 7, 8, 13, 14

[79] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4

[80] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

This supplementary material to the "PoseContrast" paper (3DV 2021) provides:

A. Implementation and training details,
B. Dataset information,
C. Details on hyper-parameters,
D. Visualizations of the latent space,
E. Class-wise results on ObjectNet3D,
F. Additional visual results.
G. Histograms of azimuth prediction errors.

Note: reference numbers for citations used here are not the same as references used in the main paper; they correspond to the bibliography section at the end of this supplementary material.

A. Implementation and Training Details

Our experiments are coded using PyTorch. Code will be made available upon publication.

A.1. Training for 3D Pose Estimation

We train our networks end-to-end using Adam optimizer with a batch size of 32 and an initial learning rate of 1e-4, which we divide by 10 at 80% of the training phase. Unless otherwise stated, we train for 15 epochs, which takes less than 2 hours on a single V100-16G GPU.

A.2. Network and Training for Object Detection

We use a class-agnostic Mask R-CNN [24] with a ResNet-50-FPN backbone [35] as our instance segmentation network. The Mask R-CNN is trained on COCO dataset [36], which contains 80 classes and 115k training images. We use the open source repo of Mask R-CNN and follow the training setting of [24], except that we adopt the class-agnostic architecture, where all 80 classes are merged into a single "object" category.

Our backbone network is initialized with weights pre-trained on ImageNet [9]. During training, the shorter edge of images are resized to 800 pixels. Each GPU has 4 images and each image has 512 sampled RoIs, with a ratio of 1:3 of positives to negatives. We train our Mask R-CNN for 90k iterations. The learning rate is set to 0.02 at the beginning and is decreased by 10 at the 60k-th and 80k-th iteration. We use a weight decay of 0.0001 and momentum of 0.9. The entire training is carried out on 4 Nvidia RTX 2080Ti GPUs. During the training, mixed-precision training is used to reduce memory consumption and accelerate training.

B. Datasets

We experimented with three commonly-used datasets for benchmarking object pose estimation in the wild. Table 7 lists their main characteristics.

While all these datasets feature a variety of objects and environments, Pascal3D+ [71] contains only the 12 rigid

Dataset	year	# classes	# img train / val*	quality
Pascal3D+ [71]	2014	12	28,648 / 2,113	+
ObjectNet3D [70]	2016	100	52,048 / 34,375	++
Pix3D [57]	2018	9	0 / 5,818	+++

Table 7. **Experimented Datasets:** images of objects in the wild, with different qualities of pose annotation due to aligned shapes. *Only non-occluded and non-truncated objects, as done usually.

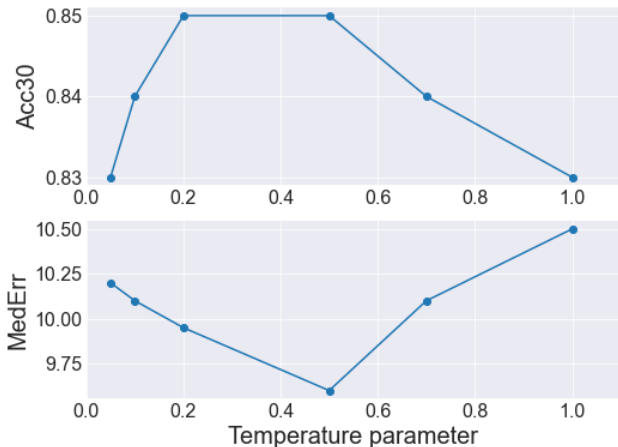

Figure 9. **Parameter Study of Temperature τ in $\mathcal{L}_{\text{poseNCE}}$.** We report the performance on the dataset Pascal3D+ [71] with 30-degree accuracy (Acc30 ↑) and median error (MedErr ↓).

classes of PASCAL VOC 2012 [13], with approximate poses due to coarsely aligned 3D models at annotation time. ObjectNet3D distinguishes 100 classes in a subset of ImageNet [9], with more accurate poses as more and finer shapes were used for annotation. Recently, Pix3D [57] proposes a smaller but more accurate dataset with pixel-level 2D-3D alignment using exact shapes; although it only features 9 classes, two of them ('tool' and 'misc') do not appear in Pascal3D+ nor ObjectNet3D. We test all methods only on non-occluded and non-truncated objects, as other publications do.

C. Hyper-parameters

We use parameters $\lambda = 1$, $\kappa = 1$, $\tau = 0.5$, and the transformation rotation ϕ varies in $[-15°, 15°]$, i.e., $[-\frac{\pi}{24}, \frac{\pi}{24}]$.

Figure 9 shows the influence of temperature parameter τ in the proposed pose-aware contrastive loss $\mathcal{L}_{\text{poseNCE}}$. By varying this parameter from 0.05 to 1.0, we obtain the best performance on Pascal3D+ when $\tau = 0.5$. While training without this pose-aware contrastive loss can still reach an overall accuracy at 0.83 and an overall median error at 10.2, we note that the performance can be improved using the propose loss $\mathcal{L}_{\text{poseNCE}}$ with a temperature parameter between 0.2 and 0.6, which is quite robust.

D. Visualizations of the Latent Space

To better understand the effect of contrastive learning, we use t-SNE to visualize the features obtained by different backbones. Results are presented in Figure 10.

The features are extracted from val images of Pascal3D+. Considering the fact that the distributions of elevation and inplane-rotation are highly centered around a specific value compared to that of azimuth, we split the visualized features into different clusters, with each cluster corresponding to objects with similar azimuth angles. More specifically, we divide the 360 degrees of azimuth angle into 24 bins, and objects within the same azimuth bin are shown by the same color.

As seen in Figure 10 (left), the features extracted using a randomly-initialized network are more or less uniformly distributed across different locations in the latent space, and regardless of their 3D poses. On the contrary, the features extracted using networks trained with contrastive learning (MOCOv2 and PoseContrast) tend to form clusters, where each cluster groups objects with similar azimuth angles. Arguably, feature clusters are less spread with PoseContrast, compared to MOCOv2, and actual azimuths are more consistent within clusters.

When doing the same kind of visualization on Pix3D, as shown in Figure 10 (right), we observe more or less the same kind of distribution for the random initialization. However, MOCOv2 has a harder time clustering the features of Pix3D images, including regarding pose. Yet, PoseContrast manages to produce clusters, and with a better pose consistency.

E. Class-Wise Results on ObjectNet3D

In this section, we present the class-wise quantitative results on the dataset ObjectNet3D [70]. As detailed in Table 8, we follow previous work [78, 61] and split the 100 object classes of ObjectNet3D into 80 base classes and 20 novel classes. We conduct both no-shot and 10-shot viewpoint estimation, as described in the paper, and report the performance on each novel classe in Table 9.

We see that our method, that only relies on RGB images, significantly outperforms all other RGB-based methods [78, 61] on both tasks and both metrics.

The best overall performance is however achieved by methods that additionally make use of 3D models. It was expected since these methods are designed to extract information regarding geometry and canonical frame from the aligned 3D object models. In fact, the performance of methods using CAD models can somehow be regarded as an upper bound with respect to RGB-based methods. Nevertheless, we outperform CAD-model-based methods on a few classes, e.g., 'filling_cabinet', 'guitar' and 'wheelchair'. Besides, the relative gap between our method and these CAD-model-based methods is mostly due to a few classes, such as 'rifle', 'iron' or 'shoe', for which base classes offer limited help in terms of geometrical cue or canonical frame. In fact, if we put aside 'iron' and 'shoe', our method is on par with FSDetView [74] on 10-shot viewpoint estimation, despite not using any extra shape information.

Moreover, our class-agnostic network directly estimates the viewpoint from image embeddings, without relying on any keypoint prediction. This direct estimation network thus can predict the viewpoint for all classes, while keypoint-based methods struggle to get a reasonable prediction for certain classes, e.g., 'door', 'pen', and 'shoe'.

F. Additional Visual Results

We present in Figure 11 some additional visual results of our class-agnostic method on the cross-dataset 3D pose estimation, training on Pascal3D+ and testing on Pix3D [57].

G. Histograms of Azimuth Prediction Errors

Finally, we present in Figure 12 the histograms of azimuth angle prediction errors on Pascal3D+ [71]. As shown in the figure, the largest viewpoint prediction errors come from the ambiguity caused by the symmetric objects, e.g., two-fold symmetric for 'boat' and four-fold symmetric for 'diningtable'.

	Base classes								Novel classes	
aeroplane	ashtray	backpack	basket	bench	bicycle	blackboard	boat		bed	bookshelf
bottle	bucket	bus	cabinet	camera	can	cap	car		calculator	cellphone
chair	clock	coffee_maker	comb	cup	desk_lamp	diningtable	dishwasher		computer	door
eraser	eyeglass	fan	faucet	file_extinguisher	fish_tank	flashlight	fork		filling_cabinet	guitar
hair_dryer	hammer	headphone	helmet	jar	kettle	key	keyboard		iron	knife
laptop	lighter	mailbox	microphone	motorbike	mouse	paintbrush	pan		microwave	pen
pencil	piano	pillow	plate	printer	racket	refrigerator	remote_control		pot	rifle
road_pole	satellite_dish	scissors	screwdriver	shovel	sign	skate	skateboard		shoe	slipper
sofa	speaker	spoon	stapler	suitcase	teapot	telephone	toaster		stove	toilet
toothbrush	train	train_bin	trophy	tvmonitor	vending_machine	washing_machine	watch		tub	wheelchair

Table 8. **Dataset split of ObjectNet3D [70]:** 80 base classes (left) and 20 novel classes (right). Some novel classes share similar geometries and canonical frames as base classes, e.g., 'door'/'blackboard', 'filling_cabinet'/'cabinet', 'wheelchair'/'chair'.

	Method $Acc30(\uparrow)/MedErr(\downarrow)$	bed	bookshelf	calculator	cellphone	computer	door	filling_cabinet
no-shot	StarMap [78]	0.37 / 45.1	0.69 / 18.5	0.19 / 61.8	0.51 / 29.8	0.74 / 15.6	– / –	0.78 / 14.1
	PoseContrast (ours)	0.62 / 17.4	0.89 / 6.7	0.65 / 17.7	0.57 / 15.8	0.85 / 14.5	0.91 / 2.7	0.88 / 10.4
	PoseFromShape [75]	0.65 / 15.7	0.90 / 6.9	0.88 / 12.0	0.65 / 10.5	0.84 / 11.2	0.93 / 2.3	0.84 / 12.7
10-shot	StarMap* [78]	0.32 / 42.2	0.76 / 15.7	0.58 / 26.8	0.59 / 22.2	0.69 / 19.2	– / –	0.76 / 15.5
	MetaView [61]	0.36 / 37.5	0.76 / 17.2	0.92 / 12.3	0.58 / 25.1	0.70 / 22.2	– / –	0.66 / 22.9
	PoseContrast (ours)	0.67 / 13.9	0.90 / 7.0	0.85 / 11.0	0.58 / 15.2	0.85 / 10.9	0.91 / 2.5	0.89 / 8.4
	FSDetView [74]	0.64 / 14.7	0.89 / 8.3	0.90 / 8.3	0.63 / 12.7	0.84 / 10.5	0.90 / 0.9	0.84 / 10.5
	Method $Acc30(\uparrow)/MedErr(\downarrow)$	guitar	iron	knife	microwave	pen	pot	rifle
no-shot	StarMap [78]	0.64 / 20.4	0.02 / 142	0.08 / 136	0.89 / 12.2	– / –	0.50 / 30.0	0.00 / 104
	PoseContrast (ours)	0.73 / 14.4	0.03 / 124	0.25 / 122	0.93 / 7.5	0.45 / 39.8	0.76 / 9.2	0.00 / 102
	PoseFromShape [75]	0.67 / 20.8	0.02 / 145	0.29 / 138	0.94 / 7.7	0.46 / 37.3	0.79 / 13.2	0.15 / 110
10-shot	StarMap* [78]	0.59 / 21.5	0.00 / 136	0.08 / 117	0.82 / 17.3	– / –	0.51 / 28.2	0.01 / 100
	MetaView [61]	0.63 / 24.0	0.20 / 76.9	0.05 / 97.9	0.77 / 17.9	– / –	0.49 / 31.6	0.21 / 80.9
	PoseContrast (ours)	0.73 / 14.7	0.03 / 126	0.23 / 116	0.94 / 6.9	0.45 / 41.3	0.78 / 10.6	0.04 / 90.4
	FSDetView [74]	0.72 / 17.1	0.37 / 57.7	0.26 / 139	0.94 / 7.3	0.45 / 44.0	0.74 / 12.3	0.29 / 88.4
	Method $Acc30(\uparrow)/MedErr(\downarrow)$	shoe	slipper	stove	toilet	tub	wheelchair	TOTAL
no-shot	StarMap [78]	– / –	0.11 / 146	0.82 / 12.0	0.43 / 35.8	0.49 / 31.8	0.14 / 93.8	0.44 / 55.8
	PoseContrast (ours)	0.23 / 58.9	0.25 / 138	0.91 / 12.0	0.43 / 30.8	0.53 / 24.0	0.42 / 43.4	0.56 / 42.6
	PoseFromShape [75]	0.54 / 28.2	0.32 / 158	0.89 / 10.1	0.61 / 21.8	0.68 / 17.8	0.39 / 57.4	0.62 / 42.0
10-shot	StarMap* [78]	– / –	0.15 / 128	0.83 / 15.6	0.39 / 35.5	0.41 / 38.5	0.24 / 71.5	0.46 / 50.0
	MetaView [61]	– / –	0.07 / 115	0.74 / 21.7	0.50 / 32.0	0.29 / 46.5	0.27 / 55.8	0.48 / 43.4
	PoseContrast (ours)	0.24 / 56.7	0.23 / 155	0.92 / 8.1	0.64 / 22.2	0.55 / 18.6	0.45 / 36.7	0.60 / 38.7
	FSDetView [74]	0.51 / 29.4	0.25 / 96.4	0.92 / 9.4	0.69 / 17.4	0.66 / 15.1	0.36 / 64.3	0.63 / 32.1

Table 9. **Few-shot viewpoint estimation on ObjectNet3D [70].** All models are trained and evaluated on ObjectNet3D. For each method, we report Acc30($\uparrow$) / MedErr($\downarrow$) on the same 20 novel classes of ObjectNet3D, while the remaining 80 classes are used as base classes. *StarMap network trained with MAML [14] for few-shot viewpoint estimation, with numbers reported in [61]. Methods additionally using 3D object models are shown in gray.

Pascal3D [71]

Pix3D [57]

(a) Random

(b) Random

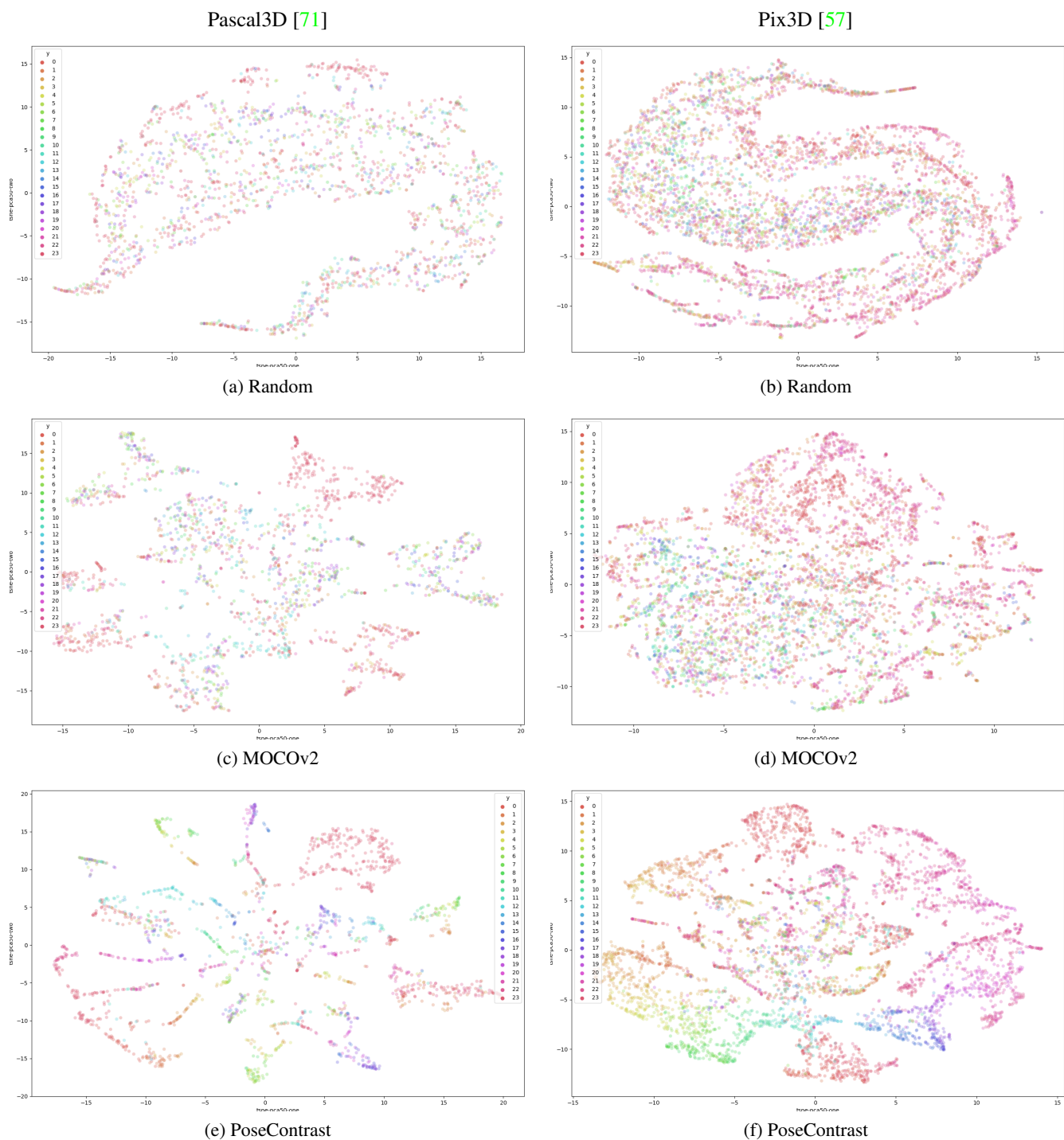(c) MOCOv2

(d) MOCOv2

(e) PoseContrast

(f) PoseContrast

Figure 10. **Feature Visualization.** We visualize image features from the val set of Pascal3D+ [71] (left) and Pix3D [57] (right) by t-SNE (preceded by PCA) for three different ResNet-50 backbones: (a,b) randomly initialized network (top); (c,d) network pre-trained on ImageNet by MOCOv2 [6] (middle); and (e,f) network trained on Pascal3D+ with PoseContrast (bottom). We divide the 360 degrees of azimuth angle into 24 bins of 15° and use one color for each bin. The figure is better viewed in color with zoom-in.
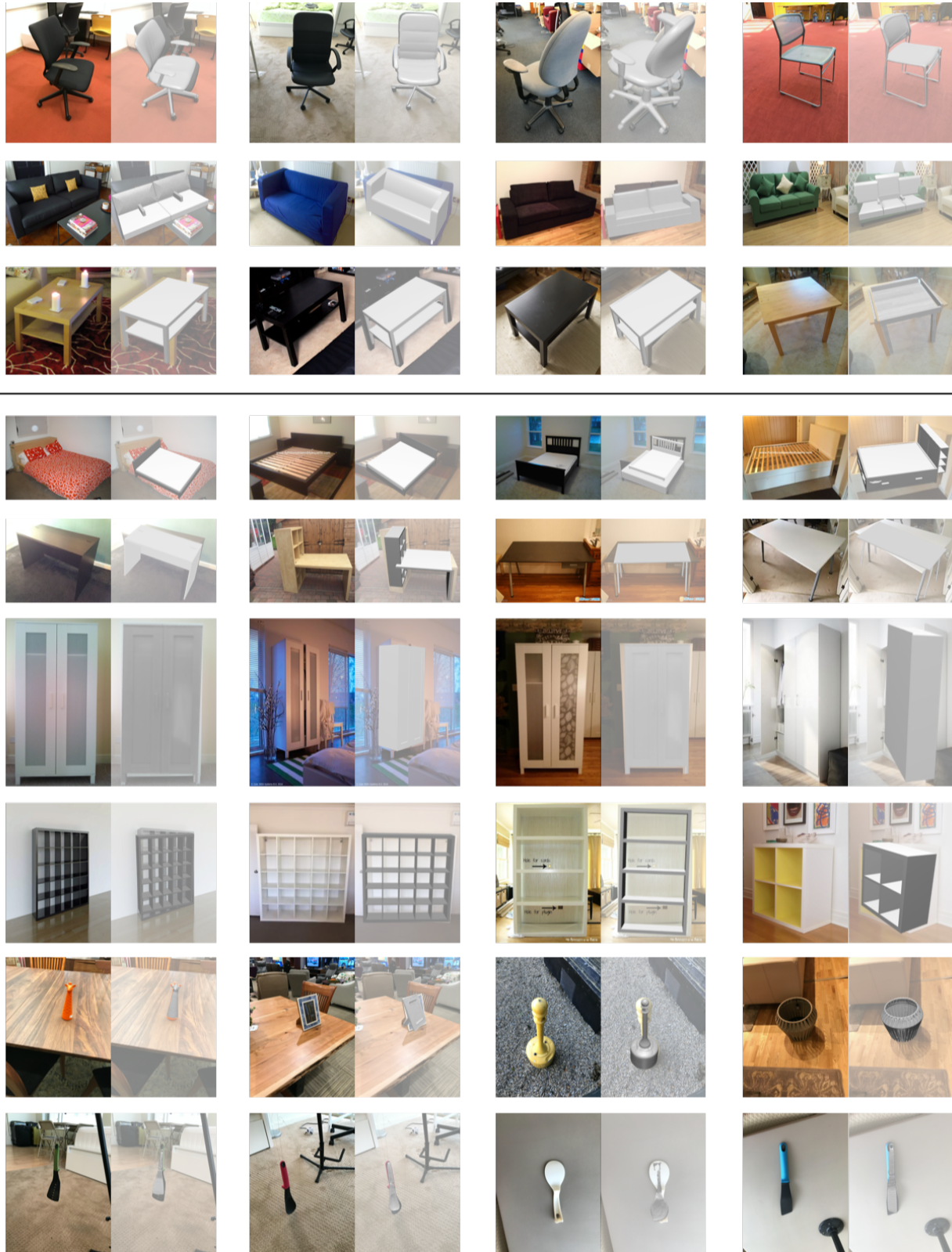
Figure 11. **Additional qualitative results on the 9 object classes of Pix3D [57].** The network is trained on the 12 object classes of Pascal3D+ and directly tested on Pix3D. From top to bottom: 'chair', 'sofa', 'table', 'bed', 'desk', 'wardrobe', 'bookcase', 'misc', and 'tool'. 3D object models here are only used to visualize the pose.
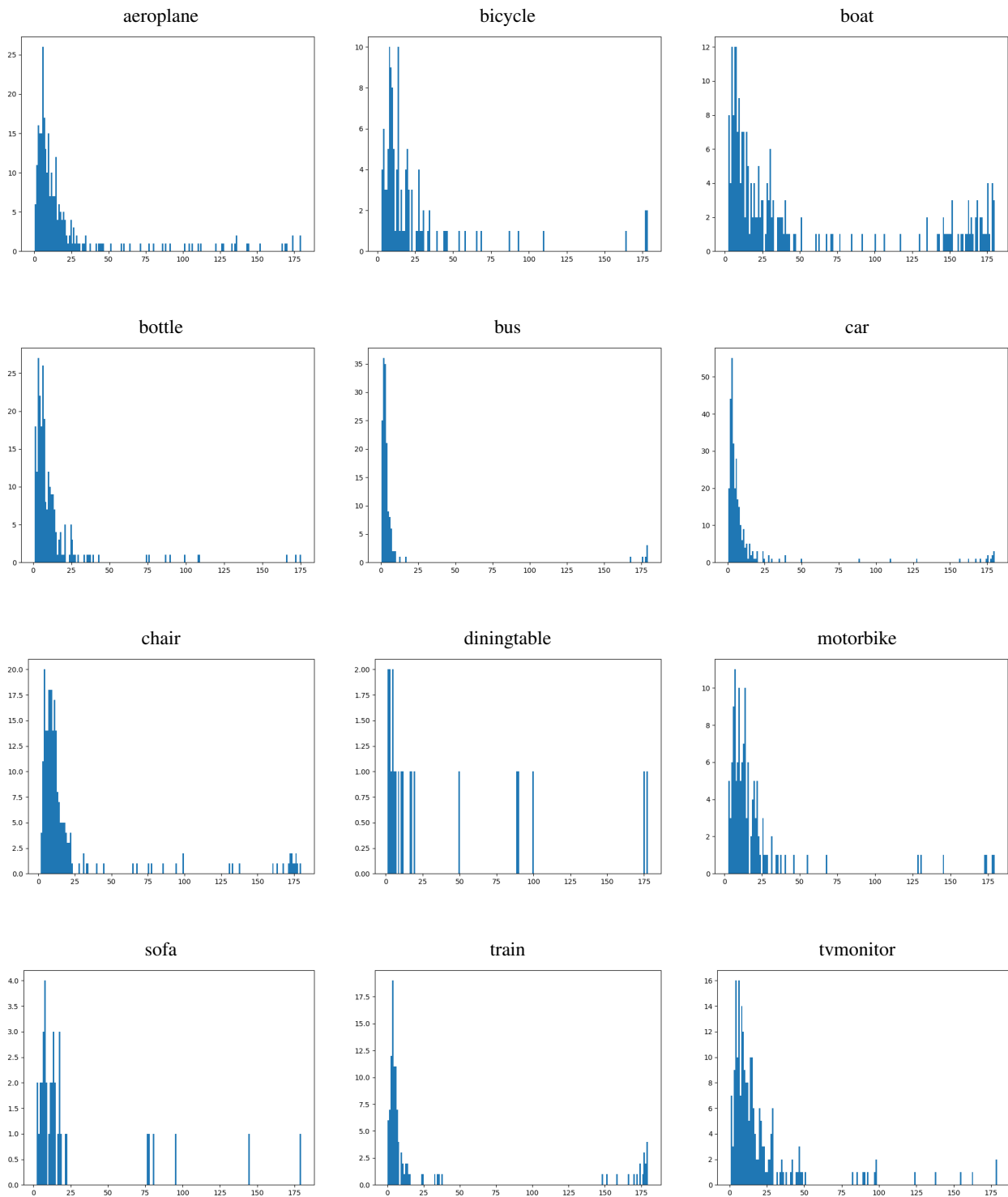
Figure 12. **Histograms of azimuth angle prediction errors on the 12 object classes of Pascal3D+ [71].**