NOPE: Novel Object Pose Estimation from a Single Image

Van Nguyen Nguyen[1], Thibault Groueix[2], Georgy Ponimatkin[1], Yinlin Hu[3], Renaud Marlet[1,4], Mathieu Salzmann[5], Vincent Lepetit[1]

[1]LIGM, Ecole des Ponts, [2]Adobe, [3]MagicLeap, [4]Valeo.ai, [5]EPFL

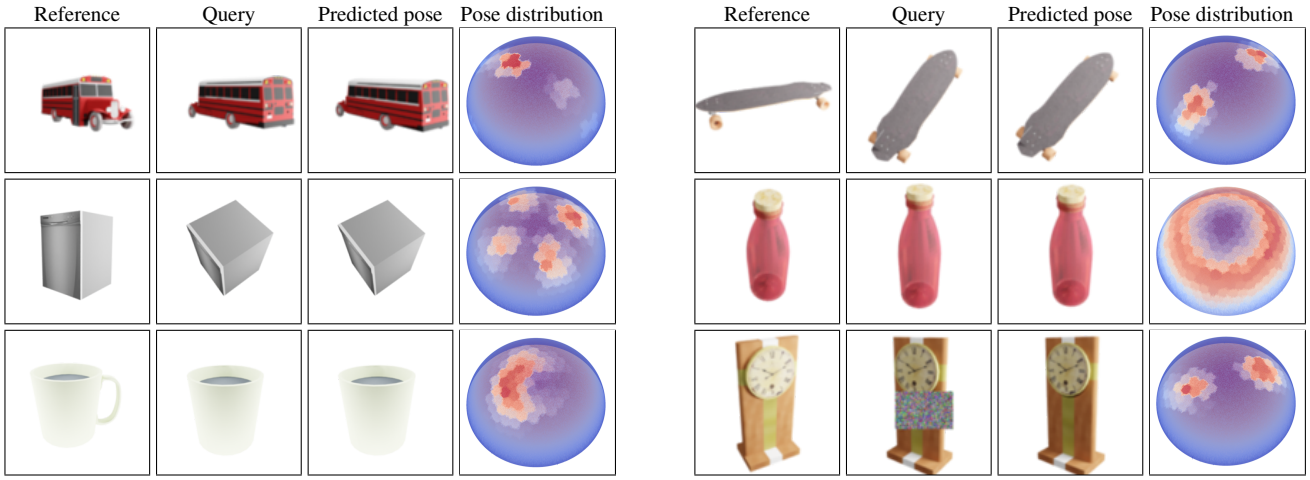

Figure 1. Given as input a single reference view of a novel object, our method predicts the relative 3D pose (rotation) of a query view and its ambiguities. **We visualize the predicted pose by rendering the object from this pose, but the 3D model is only used for visualization purposes, not as input to our method.** Our method works by estimating a probability distribution over the space of 3D poses, visualized here on a sphere centered on the object. **We use the canonical pose of the 3D model to visualize this distribution, but not as input to our method.** From this distribution, we can also identify the pose ambiguities: For example, in the case of the bottle, any pose with the same pitch and roll is possible; in the case of the mug, a range of poses are possible as the handle is not visible in the query image. Our method is also robust to partial occlusions, as shown on the clock hidden in part by a rectangle in the query image.

Abstract

The practicality of 3D object pose estimation remains limited for many applications due to the need for prior knowledge of a 3D model and a training period for new objects. To address this limitation, we propose an approach that takes a single image of a new object as input and predicts the relative pose of this object in new images without prior knowledge of the object's 3D model and without requiring training time for new objects and categories. We achieve this by training a model to directly predict discriminative embeddings for viewpoints surrounding the object. This prediction is done using a simple U-Net architecture with attention and conditioned on the desired pose, which yields extremely fast inference. We compare our approach to state-of-the-art methods and show it outperforms them both in terms of accuracy and robustness.

1. Introduction

Estimating the 3D pose of objects has seen significant progress in the past decade with regard to both robustness and accuracy [12, 16, 37, 50, 58]. Specifically, there has been a considerable increase in robustness to partial occlusions [8, 33, 34], and the need for large amounts of real annotated training images has been relaxed through the use of domain transfer [1], domain randomization [14, 20, 47, 51], and self-supervised learning techniques [49] that leverage synthetic images for training.

Unfortunately, the practicality of 3D object pose estimation remains limited for many applications, including robotics and augmented reality. Typically, existing approaches require a 3D model [15, 31, 32, 55], a video sequence [5, 46], or sparse multiple images of the target ob-

ject [59], and a training stage. Several techniques aim to prevent the need for retraining by assuming that new objects fall into a recognized category [4, 53], share similarities with the previously trained examples as in the T-LESS dataset [47], or exhibit noticeable corners [35].

In this paper, we introduce an approach, which we call **NOPE** for **N**ovel **O**bject **P**ose **E**stimation, that only requires a single image of the new object to predict the relative pose of this object in any new images, without the need for the object's 3D model and without training on the new object. This is a very challenging task, as, by contrast with the multiple views used in [46, 59] for example, a single view only provides limited information about the object's geometry.

To achieve this, we train NOPE to predict the appearance of the object under novel views. We use these predictions as 'templates' annotated with the corresponding poses. Matching these templates with new input views lets us estimate the object relative pose with respect to the initial view. This approach is motivated by the good performance of recent related work [32, 44]. In particular, [32] showed that template matching can be extremely fast and robust to partial occlusions. This contrasts with methods that rely on a deep network to predict the probability of a pose [59].

Since our method relies on predicting the appearance of the target object, it relates to recent developments in novel view synthesis. However, it has two critical differences: The first difference is that instead of predicting color images, we directly predict discriminative embeddings of the views. These embeddings are extracted by passing the input image through a U-Net architecture with attention and conditioned on the desired pose for the new view.

The second main difference of our approach with novel view synthesis is more fundamental. We first note that generating novel views given a single view of an object is ambiguous. Novel view synthesis usually focuses on generating a single possible image for a given point of view. This is however not suitable for our purpose: The view synthesis method will "invent" the parts that were not visible in the input view. As illustrated in Figure 2, these invented parts create a plausible novel view but there is no guarantee this view actually corresponds to the actual view. For our goal of pose estimation, the invented parts will not match in general the query view and this will result in incorrect pose estimation. The limitations of using novel view synthesis for pose estimation will further be quantitatively demonstrated in our experiments (see Table 1).

Our approach to handling the ambiguities in novel view synthesis for template matching is to consider the *distribution* of all the possible appearances of the object for the target viewpoint. More exactly, we train NOPE to predict the average of all the possible appearances of the object. We then treat the predicted average as a template: Under some simple assumptions, the distance between this template and the query view is directly related to the probability of the query view to be a sample from the distribution of the possible appearances of the object. This approach allows us to deal with the ambiguities of novel view prediction in a robust and efficient way: Predicting the average views is just a direct inference of NOPE and is thus very fast, and robust to partial occlusions thank to template-matching.

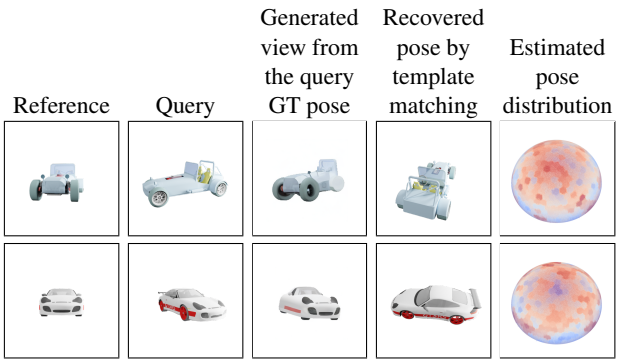

Figure 2. **The limit of novel view synthesis for pose prediction.** While the images generated by Wonder3D [21] look very realistic, they have to invent unseen parts, impairing the similarity computation between the query image and the generated view, and hence the pose estimation: The probability distributions computed by template matching do not peak on the right pose but show many wrong local maxima. This is not a limitation of Wonder3D but of view synthesis from a single view in general.

Furthermore, our approach can identify the pose ambiguities due, for example, to symmetries [23], even if we do not have access to the object 3D model but only to a single view. To this end, we estimate the distribution over all poses for the query, which becomes increasingly less peaked as the pose suffers from increasingly many ambiguities. Figure 1 depicts a variety of ambiguous and unambiguous cases with their pose distributions.

In summary, our main contribution is to show we can efficiently and reliably recover the relative pose of an unseen object in novel views given only a single view of that object as reference. To the best of our knowledge, our approach is the first to predict ambiguities due to symmetries and partial occlusions of unseen objects from only a single view.

2. Related Work

In this section, we first review various approaches to novel view synthesis. We then shift our focus to pose estimation techniques that aim to achieve generalization.

2.1. Novel view synthesis from a single image

Our method generates discriminative feature views, which are conditioned on a reference view and the relative pose between the views. This relates to the pioneering work of NeRFs [28] since it performs novel-view synthesis. While recent advancements have improved the speed of NeRFs [29, 43, 56], our approach is still orders of magnitude faster as it does not require the creation of a full 3D volumetric model. Furthermore, our approach only requires a single input view, whereas a typical NeRF setup necessi-

tates around 50 views. Reducing the number of views required for NeRF reconstruction remains an active research area, especially in the single-view scenario [27, 57].

Recent works [27, 62] have had successes generating novel views via NERFs using a sparse set of views as input by leveraging 2D diffusion models. For images, the breakthrough in diffusion models [6,45] have unlocked several workflows [39, 41, 42]. For 3D applications, DreamFusion [36] pioneered a score-distillation sampling that allows for the use of a 2D diffusion model as an image-based loss, leveraged by 3D applications via differentiable rendering. This has resulted in significant improvements for tasks previously trained with a CLIP-based image loss [9, 10, 13, 17, 38, 52]. By building on top of score-distillation sampling, SparseFusion [62] reconstructs a NeRF scene with as few as two views with relative pose, while the concurrent work RealFusion [27] does it from a single input view, although the reconstruction time is impractical for real-time applications. Our approach is much faster as we do not create a 3D representation of the object.

Closest to us, 3DiM [54] and Zero-1-to-3 [18] generate novel views of an object by conditioning a diffusion model on the pose. Instead of leveraging foundation diffusion models in 2D like DreamFusion [36] does, they retrain a diffusion model specifically for this task. While they have not applied their approach to template-based pose estimation, we design such a baseline and compare against it. We find that the diffusion model tends to change the texture or invent wrong details which hinders the performance of the template-based approach. In contrast, our approach generates average novel views directly in an embedding space instead of a pixel space, which is much more efficient [32].

Finally, several methods [25,26] generate novel views by conditioning a feed-forward neural network on the 3D pose, which we also do with a U-Net. We share with these methods an advantage in speed: such feed-forward neural network are one or two orders of magnitude faster than current diffusion models. However, the way we perform pose estimation is fundamentally different. We use novel-view synthesis in a template-based matching approach [32], while they use it in a regression-based optimization. In practice, we found these methods to work well on a limited number of object categories, and we observed their performance to deteriorate significantly when testing on novel categories.

2.2. Generalizable object pose estimation

Several techniques have been explored to generalize better to unseen object pose estimation, such as generic 2D-3D correspondences [35], an energy-based strategy [59], keypoint matching [46], or template matching [15, 19, 31, 32, 44, 60]. Despite significant progress, these methods either need an accurate 3D model of the target or they rely on multiple annotated reference images from different viewpoints. These 3D annotations are challenging to obtain in practice. By contrast, we propose a strategy that works with neither the 3D model of the target nor the annotation of multiple views. More importantly, our method predicts accurate poses with only a single reference image, and generalizes to novel objects without retraining.

3. Method

In this section, we first introduce our formalism, then describe our architecture and how we train it, and finally how we use it for pose prediction and for identifying pose ambiguities.

3.1. Formalization

Given a reference image I_r of a target object and a query image I_q of the same object, we would like to estimate the probability $p(\Delta R \mid I_r, I_q)$ that the relative motion between I_r and I_q is a certain discretized relative pose ΔR. We assume that this probability follows a normal distribution in the embedding space of the images:

$$p(\Delta R \mid I_r, I_q) = \mathcal{N}(\mathbf{e}_q \mid \mathbf{e}(\mathbf{e}_r, \Delta R), \Sigma(\mathbf{e}_r, \Delta R)) \,, \quad (1)$$

where $\mathbf{e}_q$ and $\mathbf{e}_r$ are the embeddings for query image I_q and reference image I_r respectively, $\mathbf{e}(\mathbf{e}_r, \Delta R)$ is the mean of the normal distribution, and $\Sigma(\mathbf{e}_r, \Delta R)$ its covariance. This approach allows us to handle the fact that the object can have various appearances from viewpoint ΔR given the reference image, as discussed in the introduction.

We take the mean $\mathbf{e}(\mathbf{e}_r, \Delta R)$ as the average embedding for the appearance of the object from pose ΔR over the possible 3D shapes for the object:

$$\mathbf{e}(\mathbf{e}_r, \Delta R) = \int_{\mathcal{M}} \mathbf{e}(\Delta R, \mathcal{M}) p(\mathcal{M} | \mathbf{e}_r) d\mathcal{M} \,, \quad (2)$$

with $\mathcal{M}$ a 3D model of testing object and $\mathbf{e}(\Delta R, \mathcal{M})$ the image embedding of same object under pose ΔR. $\mathbf{e}(\mathbf{e}_r, \Delta R)$ may look complicated to compute, but it is in fact easy to train a deep network to predict it using the L2 loss:

$$\sum_{(\mathbf{e}_1, \mathbf{e}_2, \Delta R)} \| F(\mathbf{e}_r, \Delta R) - \mathbf{e}_2 \|^2 \,. \quad (3)$$

F denotes the network, $(\mathbf{e}_1, \mathbf{e}_2, \Delta R)$ is a training sample where $\mathbf{e}_1$ is the embedding for a view of a training object and $\mathbf{e}_2$ the embedding for the view of the same object after pose change ΔR. During training, given enough samples, $F(\mathbf{e}_r, \Delta R)$ will converge naturally towards $\mathbf{e}(\mathbf{e}_r, \Delta R)$.

3.2. Framework

Figure 3 gives an overview of our approach. We train a deep architecture to predict the average embeddings of novel views of an object using pairs of images of objects and the corresponding pose changes from a first set of object categories. In practice, we consider embeddings computed from the pretrained VAE of [40], as it was shown to be robust for template matching. To generate these embeddings, we use a U-Net-like network with a pose conditioning mechanism that is very close to the one of 3DiM [54].

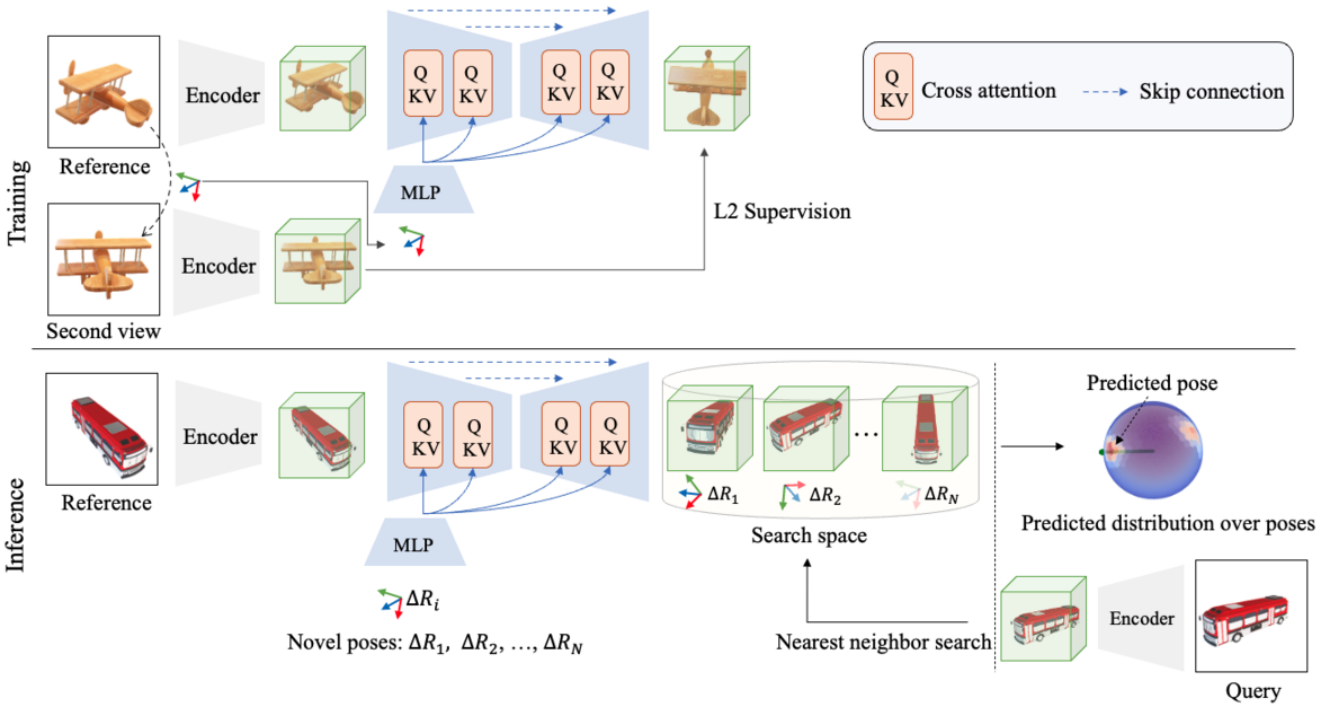

Figure 3. **Overview.** During training, we train a U-Net to predict the embedding of a novel view of an object, given a reference image of the object and a relative pose. The U-Net is conditioned on an embedding of the relative pose computed using an MLP, which we train jointly with the U-Net. At inference, our method first takes as input a reference image of a new object and predicts the embeddings of views of the object under many relative poses. This inference takes around 1 second on a single GPU V100. Then, given a query image of the object, we first compute its embedding and match it against the set of predicted embeddings. This gives us a distribution over the possible relative poses between the reference and query images, where the maximum corresponds to the predicted pose.

More precisely, we first use an MLP to convert the desired relative viewpoint ΔR with respect to the object pose in the reference view to a pose embedding. We then integrate this pose embedding into the feature map at every stage of our U-Net using cross-attention, as in [40].

Training. At each iteration, we build a batch composed of N pairs of images, a reference image and another image of the same object with a known relative pose. The U-Net model takes as input the embedding of the reference image and as conditioning the embedding of the relative pose to predict an embedding for the second image. We jointly optimize the U-Net and the MLP by minimizing the Euclidean distance between this predicted embedding and the embedding of the query image. Note that we freeze the pretrained VAE network of [40] during the training.

By training it on a dataset of diverse objects, this architecture generalizes well to novel unseen object categories. Interestingly, our method does not explicitly learn any symmetries during training, but it is able to detect pose ambiguities during testing as discussed below.

3.3. Pose prediction

Template matching. Once our architecture is trained, we can use it to generate the embeddings for novel views: Given a reference image I_r and a set of N relative viewpoints $\mathcal{P} = (\Delta R_1, \Delta R_2, \ldots, \Delta R_N)$, we can obtain a corresponding set of predicted embeddings $(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N)$. To define these viewpoints, we follow the approach used in [32]: We start with a regular icosahedron and subdivide each triangle recursively into four smaller triangles twice to get 342 final viewpoints. Finally, we simply perform a nearest neighbor search to determine the reference point that has the embedding closest to the embedding of the query image.

Detecting pose ambiguities. Pose ambiguities arise when the object has symmetries or when an object part that could remove the ambiguity is not visible, as for the mug in Figure 1. By considering the distance between the embedding of the query image and the generated embeddings, we not only can predict a single pose but also identify all the other poses that are possible given the reference and query views.

This can be done simply by relying on the normal distribution introduced in Eq. (1):

$$\log p(\Delta R \mid I_r, I_q) \propto \|F(\mathbf{e}_r, \Delta R) - \mathbf{e}_q\|^2 \,. \qquad (4)$$

To illustrate this, we show in Figure 4 three distinct types of symmetry and visualize the pose distribution for corresponding pairs of reference and query images (not shown). The number of regions with high similarity scores is consistent with the number of symmetries and pose ambiguities: If an object has no symmetry, the probability distribution has

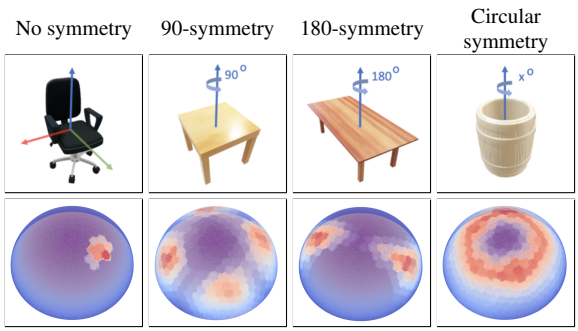

Figure 4. **Object symmetries and the pose ambiguities they may generate**, as estimated by our method given a pair of reference and query images.

a clear mode. The probability distribution for objects with symmetries have typically several modes or even a continuous high-probability region in case of rotational symmetry. We provide additional qualitative results in Section 4.

4. Experiments

In this section, we first describe our experimental setup in Section 4.1. We then compare our method to others [24, 25, 30, 32, 47, 54] on both synthetic and real-world datasets in Section 4.2. Section 4.3 reports an evaluation of the robustness to partial occlusions. We provide the runtime in Section 4.4. Finally, we discuss failure cases in Section 4.5. An ablation study is provided in the supp. mat.

4.1. Experimental setup

To the best of our knowledge, we are the first method addressing the problem of object pose estimation from a single image when the object belongs to a category not seen during training: PIZZA [30] evaluated on the DeepIM refinement benchmark, which is made of pairs of images with a small relative pose; SSVE [25] and ViewNet [24] evaluated only on objects from categories seen during training. We therefore had to create a new benchmark to evaluate our method.

Synthetic dataset. We created a dataset as in FORGE [11] using the same ShapeNet [2] object categories. For the training set, we randomly select 1000 object instances from each of the 13 categories as done in FORGE (*airplane, bench, cabinet, car, chair, display, lamp, loudspeaker, rifle, sofa, table, telephone, and vessel*), resulting in a total of 13,000 instances. We build two separate test sets for evaluation. The first test set is the "novel instances" set, which contains 50 new instances for each training category. The second test set is the "novel category" set, which includes 100 models per category for the 10 unseen categories selected by FORGE (*bus, guitar, clock, bottle, train, mug, washer, skateboard, dishwasher, and pistol*). For each 3D model, we randomly select camera poses to produce five reference images and five query images. We use BlenderProc [3] as rendering engine.

Figure 5 illustrates the categories used for training our architecture and the categories used for testing it. The shapes and appearances of the categories in the test set are very different from the shapes and appearances of the categories in the training set, and thus constitute a good test set for generalization to unseen categories.

Real-world dataset. We evaluate on the T-LESS dataset [7] following the evaluation protocol of [47]: we train only on objects 1-18 and test on the full PrimeSense test set using the ground-truth masks. At inference, we randomly sample a non-occluded reference image either from *all views* or only from *front views* (-45°$\leq$ azimuth $\leq$ 45°), which often offers more information on the object and illustrates the influence of the reference view.

Metrics. For the ShapeNet dataset, we report two different metrics based on relative camera pose error as done in [25]. Specifically, we provide the median pose error across instances for each category in the test set, and the accuracy Acc 30 for which a prediction is treated as correct when the pose error is $\leq$ 30°. Additionally, we present the results of our method for the top 3 and 5 nearest neighbors retrieved by template matching.

For the T-LESS dataset, as most objects are symmetric, we report the recall VSD metric as done in [47]. Please note that for the evaluation on the T-LESS dataset, we also predict the translation by using the same formula "projective distance estimation" as SSD-6D [12], as done in [47, 48]. This translation is deduced from the retrieved template and the relative scale factor between the two input images, as detailed in Section 8 of [32].

Baselines. We compare our work with all previous methods that aim to predict a pose from a single view: PIZZA [30], a regression-based approach that directly predicts the relative pose, as well as SSVE [25] and ViewNet [24], which employ semi-supervised and self-supervised techniques to treat viewpoint estimation as an image reconstruction problem using conditional generation. We also compare our method with the recent diffusion-based method 3DiM [54], which generates pixel-level view synthesis. Since 3DiM originally only targets view-synthesis and is not designed for 3D object pose, we use it to generate templates and perform nearest neighbor search to estimate a 3D object pose. To make 3DiM work in the same setting as us, we retrain it using relative pose conditioning instead of canonical pose conditioning.

Implementation. Only the code of PIZZA is available. The other methods did not release their code at the time of writing, however we re-implemented them. We use a ResNet18 backbone as in [30] for PIZZA, SSVE, and ViewNet. We train all models on input images with a resolution of 256×256 except for 3DiM for which we use a resolution of 128×128 since 3DiM performs view synthesis in pixel space, which takes much more memory. Our re-implementations achieve similar performance as the original papers when evaluated on the same data for seen cat-

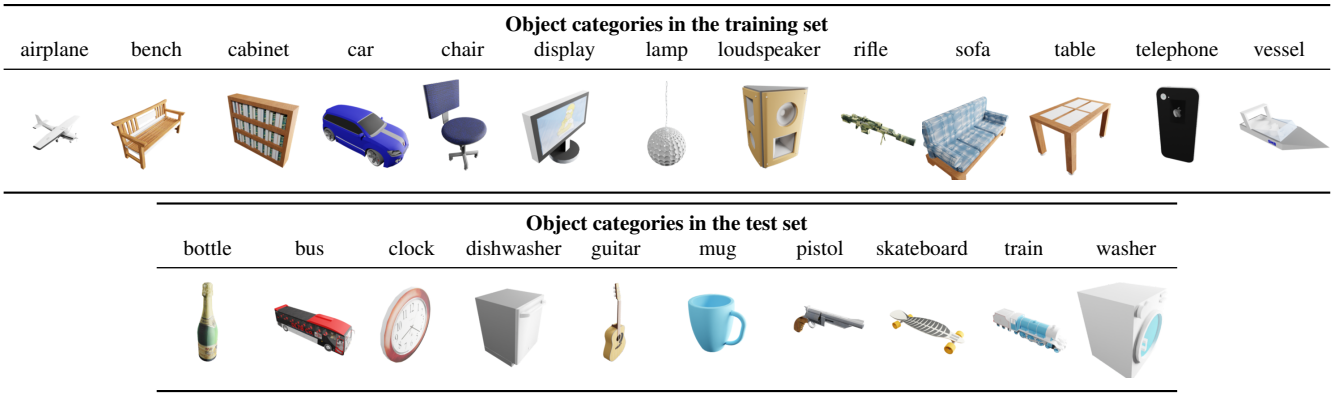

Figure 5. **Visualization of training and test sets from the ShapeNet dataset [2].** The shapes and appearances in the training and test sets are very different and thus constitute a good test bed for generalization to unseen categories.

	Method	novel inst.	bottle*	bus	clock	dishwasher	guitar	mug	pistol	skateboard	train	washer	mean
Acc 30 ↑	ViewNet [24]	**77.5**	48.4	36.2	23.5	16.4	37.8	31.3	17.9	33.9	44.8	25.1	35.7
	SSVE [25]	75.3	61.5	38.2	41.8	21.3	46.8	38.4	36.8	**62.3**	41.5	50.8	46.8
	PIZZA [30]	72.3	76.0	38.6	38.5	32.6	30.8	35.6	40.4	58.3	52.9	**61.0**	48.8
	3DiM [54]	77.3	95.1	43.5	23.6	24.5	36.0	32.0	31.9	50.3	37.0	56.1	46.1
	Ours (top 1)	75.5	**96.0**	**53.6**	**48.0**	**48.0**	**49.0**	**44.6**	**69.0**	57.8	**55.2**	60.6	**59.8**
	Ours (top 3)	92.0	97.4	83.8	73.4	78.5	66.8	56.0	83.8	86.2	86.0	84.4	80.8
	Ours (top 5)	95.5	97.8	89.8	80.4	88.2	74.6	62.8	88.4	92.8	95.4	93.4	87.1
Median ↓	ViewNet [24]	6.6	26.7	35.8	40.3	96.3	50.6	51.6	42.8	37.4	26.8	44.3	41.7
	SSVE [25]	6.1	23.8	45.2	41.9	90.4	47.6	49.6	24.0	13.5	24.9	48.1	37.7
	PIZZA [30]	5.8	25.5	26.4	43.2	80.6	40.2	45.5	23.4	17.3	20.3	38.5	33.3
	3DiM [54]	**5.7**	**1.8**	19.8	47.3	98.8	35.2	35.7	21.2	**12.5**	**17.6**	19.2	28.6
	Ours (top 1)	8.1	**1.8**	**18.4**	**39.9**	**77.6**	**31.6**	**35.5**	**13.4**	15.5	18.3	**8.5**	**24.4**
	Ours (top 3)	5.0	1.3	5.8	9.1	4.8	16.0	22.6	8.1	6.5	6.7	5.7	8.3
	Ours (top 5)	4.5	1.2	4.5	7.1	4.4	11.6	18.4	6.1	5.6	4.9	5.0	6.6

Table 1. **Quantitative results on ShapeNet.** *We treat "bottle" as a symmetric category, i.e., the error is only the difference of elevation angle. Since the quality of prediction may depend on the reference image, we report the score as the average over 5 runs with 5 different reference images.

egories, as shown in Table 1, which validates our comparisons. Our method also uses the frozen encoder from [40] to encode the input images into embeddings of size $32\times32\times8$. In all settings, we train the baselines and our method using the same training set and AdamW [22] with an initial learning rate of 5×10^{-5}. Training takes about 20 hours on 4 V100 GPUs for each method.

4.2. Comparison with the state of the art

4.2.1 Results on ShapeNet

Table 1 summarizes the results of our method compared with the baselines discussed above. Under both the Acc30 and Median metrics, our method consistently achieves the best overall performance, outperforming the baselines by more than 10% in Acc30 and 10^o in Median. In particular, while other works produce reasonable results on unseen instances of seen training categories, they often struggle to estimate the 3D pose of objects from unseen categories. By contrast, our method works well in this case, demonstrating a better generalization ability on unseen categories.

	Method	Ref. image sampling	Recall VSD		
			Seen obj.	Novel obj.	Avg
GT CAD	Nguyen et al. [32]	-	**60.15**	**58.70**	**59.57**
	MultiPath [47]	-	43.17	43.33	43.24
1 ref. image (avg 5 runs)	PIZZA [30]	all views	20.05	15.90	18.39
	Ours	all views	**47.03**	**45.69**	**46.49**
	PIZZA [30]	front views	21.63	15.55	19.19
	Ours	front views	**49.30**	**48.46**	**48.96**

Table 2. **Comparison to PIZZA [30] and CAD-based methods [32, 47]** on seen (obj. 1-18) and novel (obj. 19-30) objects of T-LESS. We report numbers averaged over 5 different samplings and runs.

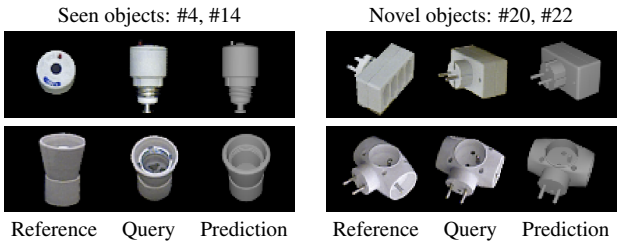

Figure 6. **Qualitative results on real images of T-LESS.** For each sample, we show in the last column the predicted poses.

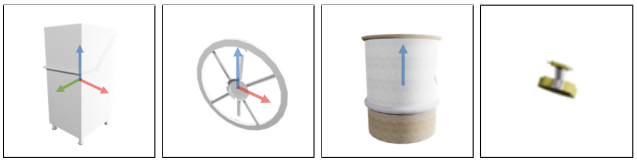

Figure 7. **Failure cases.** "Dishwashers", "clocks", and "dishwashers" are "nearly symmetrical" while "guitars" are barely visible from some viewpoints. This makes the pose estimation very challenging, and all the methods perform poorly on these categories.

Figure 8 shows some visualization results of our method on unseen categories, with and without symmetries. Our method produces more accurate 3D poses than the baselines when there is a symmetry axis.

4.2.2 Results on T-LESS

Table 2 shows our comparison with [30, 32, 47] on real images of T-LESS. While our method focuses on the more challenging case of using *a single reference image*, [32, 47] rely on ground-truth CAD models. Our method consistently outperforms the baseline PIZZA by a large margin. Interestingly, although there is still a gap compared to the SOTA [32], our method outperforms MultiPath [47]. Figure 6 shows results on seen and unseen objects of T-LESS.

4.3. Robustness to occlusions

To evaluate the robustness of our method against occlusions, we added random rectangle filled with Gaussian noise to the query images over the objects, in a similar way to Random Erasing [61]. We vary the size of the rectangles to cover a range betwen 0% to 25% of the bounding box of the object. Figures 1 and 8 show several examples.

Table 3 compares PIZZA, the best second performing method in our previous evaluation, to our method for different occlusion rates. Our method remains robust even under large occlusions, thanks to embedding matching. Figure 8 shows that our pose probabilities remain peaked on the correct maximum and shows clearly the symmetries.

Acc 30 ↑ Method	0%	5%	10%	15%	20%	25%
PIZZA [30]	48.9	44.6	33.3	24.5	18.2	14.6
NOPE (ours)	**59.8**	**54.3**	**48.4**	**45.1**	**43.7**	**40.5**

Table 3. **Robustness to partial occlusions.** We add rectangles of Gaussian noise to the query image, and vary the ratio between the area of the rectangle and the area of the object's 2D bounding box. Our method remains robust under large occlusions, while PIZZA's performance decreases significantly.

Method	Memory	Run-time	
		Processing	Neighbors search
3DiM [54]	358.6 MB	13 min	0.31 s
NOPE (ours)	22.4 MB	1.01 s	0.18 s

Table 4. **Average run-time** of our method and 3DiM [54] on a single GPU V100. We report the memory used for storing novel views, the time taken to generate novel views, and the time taken for nearest neighbor search to obtain the final prediction.

4.4. Runtime analysis

We report the running time of NOPE and 3DiM in Table 4. Our method is significantly faster than 3DiM, thanks to our strategy of predicting the embedding of novel viewpoints with a single step instead of multiple diffusion steps.

4.5. Failure cases

All the methods fail to yield accurate results when evaluated on "clock", "dishwasher", "guitar", and "mug" categories, as indicated by the high median errors. As shown in Figure 7, these categories except "guitar" are "almost symmetric", in the sense that only small details make the pose non-ambiguous. Our predictions using the top-3 and top-5 nearest neighbors significantly improves median errors for 90-symmetrical, 180-symmetrical objects, but not circular-symmetrical as mug objects. Additionally, guitar objects can appear very thin under certain viewpoints.

5. Conclusion

Our experiments have shown that direct inference of average view embeddings from a single view, as in NOPE, leads to accurate object pose estimation. This is true even for objects from unseen categories, while requiring neither retraining nor a 3D model. NOPE also lets us estimate the pose ambiguities that arise for many objects.

Acknowledgments. The authors thank Elliot Vincent, Mathis Petrovich and Nicolas Dufour for helpful discussions. This project was funded in part by the European Union (ERC Advanced Grant explorer Funding ID #101097259). This work was performed using HPC resources from GENCI–IDRIS 2022-AD011012294R2 and 2022-AD011012294R3.

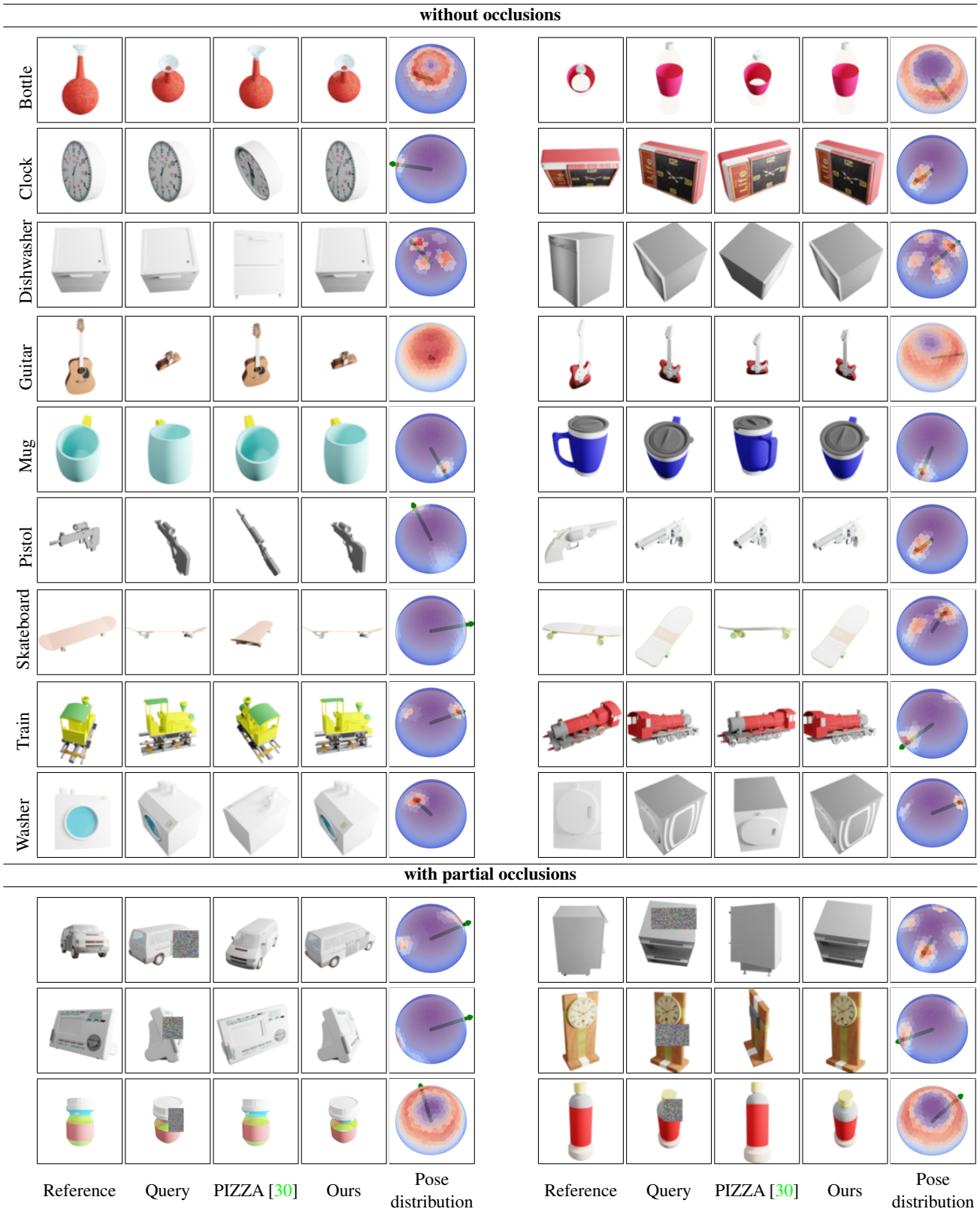

Figure 8. **Visual results on unseen categories** from ShapeNet. An arrow indicates the pose with the highest probability as recovered by our method. We visually compare with PIZZA, which is the method with the second best performance. **We visualize the predicted poses by rendering the object from these poses, but the 3D model is only used for visualization purposes, not as input to our method. Similarly, we use the canonical pose of the 3D model to visualize this distribution, but not as input to our method.**

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-Supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects. In *CVPR*, 2020. 1

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, and Others. ShapeNet: An Information-Rich 3D Model Repository. In *arXiv*, 2015. 5, 6

[3] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. BlenderProc. In *arXiv*, 2019. 5

[4] Alexander Grabner, Peter M. Roth, and Vincent Lepetit. 3D Pose Estimation and 3D Model Retrieval for Objects in the Wild. In *CVPR*, 2018. 2

[5] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. OnePose++: Keypoint-Free One-Shot Object Pose Estimation Without CAD Models. In *NeurIPS*, 2022. 1

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *NeurIPS*, 2020. 3

[7] Tomas Hodan, Pavel Haluza, Stepan Obdrzalek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects. In *WACV*, 2017. 5

[8] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-Driven 6D Object Pose Estimation. In *CVPR*, 2019. 1

[9] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-Shot Text-Guided Object Generation with Dream Fields. In *CVPR*, 2022. 3

[10] Ajay Jain, Amber Xie, and Pieter Abbeel. VectorFusion: Text-to-SVG By Abstracting Pixel-Based Diffusion Models. In *SIGGRAPH*, 2022. 3

[11] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-View Object Reconstruction with Unknown Categories and Camera Poses. In *3DV*, 2022. 5

[12] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *ICCV*, 2017. 1, 5

[13] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. CLIP-Mesh: Generating Textured Meshes from Text Using Pretrained Image-Text Models. In *SIGGRAPH*, 2022. 3

[14] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-View Multi-Object 6D Pose Estimation. In *ECCV*, 2020. 1

[15] 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. In *CoRL*, 2022. 1, 3

[16] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *ECCV*, 2018. 1

[17] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-Resolution Text-to-3D Content Creation. In *arXiv*, 2022. 3

[18] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 3

[19] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6D: Generalizable Model-Free 6DoF Object Pose Estimation from RGB Images. In *ECCV*, 2022. 3

[20] Vianney Loing, Renaud Marlet, and Mathieu Aubry. Virtual training for a real application: Accurate object-robot relative localization without calibration. *IJCV*, 126(9):1045–1060, Sept. 2018. 1

[21] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 2

[22] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 6

[23] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the Ambiguity of Object Detection and 6D Pose From Visual Data. In *ICCV*, 2019. 2

[24] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. ViewNet: Unsupervised Viewpoint Estimation from Conditional Generation. In *ICCV*, 2021. 5, 6

[25] Octave Mariotti and Hakan Bilen. Semi-Supervised Viewpoint Estimation with Geometry-aware Conditional Generation. In *ECCV Workshop*, 2020. 3, 5, 6

[26] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. ViewNeRF: Unsupervised Viewpoint Estimation Using Category-Level Neural Radiance Fields. In *BMVC*, 2022. 3

[27] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. RealFusion: 360^o Reconstruction of Any Object from a Single Image. In *arXiv*, 2023. 3

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 2

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *TRA*, 41(4), July 2022. arXiv:2201.05989 [cs]. 2

[30] Van Nguyen Nguyen, Yuming Du, Yang Xiao, Michael Ramamonjisoa, and Vincent Lepetit. PIZZA: A Powerful Image-only Zero-Shot Zero-CAD Approach to 6 DoF Tracking. In *3DV*, 2022. 5, 6, 7, 8

[31] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence. In *CVPR*, 2024. 1, 3

[32] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7

[33] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *ECCV*, 2018. 1

[34] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In *CVPR*, 2019. 1

[35] Giorgia Pitteri, Aurélie Bugeau, Slobodan Ilic, and Vincent Lepetit. 3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface Embeddings. In *ACCV*, 2020. 2, 3

[36] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D Using 2D Diffusion. In *arXiv*, 2022. 3

[37] 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 *ICCV*, 2017. 1

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 3

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. In *arXiv*, 2022. 3

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022. 3, 4, 6

[41] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-To-Image Diffusion Models for Subject-Driven Generation. In *arXiv*, 2022. 3

[42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, and Others. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *arXiv*, 2022. 3

[43] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. VoxGRAF: Fast 3D-Aware Image Synthesis with Sparse Voxel Grids. In *arXiv*, 2022. 2

[44] Ivan Shugurov, Fu Li, Benjamin Busam, and Slobodan Ilic. OSOP: A Multi-Stage One Shot Object Pose Estimation Framework. In *CVPR*, 2022. 2, 3

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *ICLR*, 2021. 3

[46] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. OnePose: One-Shot Object Pose Estimation Without CAD Models. In *CVPR*, 2022. 1, 2, 3

[47] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O. Arras, and Rudolph Triebel. Multi-Path Learning for Object Pose Estimation Across Domains. In *CVPR*, 2020. 1, 2, 5, 6, 7

[48] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *ECCV*, 2018. 5

[49] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, and Rudolph Triebel. Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection. *IJCV*, 128(3), 2020. 1

[50] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*, 2018. 1

[51] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. In *CoRL*, 2018. 1

[52] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. CLIPasso: Semantically-Aware Object Sketching. In *SIGGRAPH*, 2022. 3

[53] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In *CVPR*, 2019. 2

[54] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel View Synthesis with Diffusion Models. In *ICLR*, 2023. 3, 5, 6, 7

[55] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. In *BMVC*, 2019. 1

[56] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields Without Neural Networks. In *CVPR*, 2022. 2

[57] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*, 2021. 3

[58] Sergey Zakharov, Ivan S. Shugurov, and Slobodan Ilic. DPOD: 6D Pose Object Detector and Refiner. In *ICCV*, 2019. 1

[59] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild. In *ECCV*, 2022. 2, 3

[60] Chen Zhao, Yinlin Hu, and Mathieu Salzmann. Fusing Local Similarities for Retrieval-based 3D Orientation Estimation of Unseen Objects. In *ECCV*, 2022. 3

[61] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 7

[62] Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling View-conditioned Diffusion for 3D Reconstruction. In *arXiv*, 2023. 3