Unified Category-Level Object Detection and Pose Estimation from RGB Images using 3D Prototypes

Tom Fischer*[1,2] Xiaojie Zhang*[1,2] Eddy Ilg[2]
[1] Saarland University [2] University of Technology Nuremberg

Abstract

Recognizing objects in images is a fundamental problem in computer vision. Although detecting objects in 2D images is common, many applications require determining their pose in 3D space. Traditional category-level methods rely on RGB-D inputs, which may not always be available, or employ two-stage approaches that use separate models and representations for detection and pose estimation. For the first time, we introduce a unified model that integrates detection and pose estimation into a single framework for RGB images by leveraging neural mesh models with learned features and multi-model RANSAC. Our approach achieves state-of-the-art results for RGB category-level pose estimation on REAL275, improving on the current state-of-the-art by 22.9% averaged across all scale-agnostic metrics. Finally, we demonstrate that our unified method exhibits greater robustness compared to single-stage baselines. Our code and models are available at github.com/Fischer-Tom/unified-detection-and-pose-estimation.

1. Introduction

Finding out whether an object is present in an image, where it is located, and how it is oriented in space, are fundamental problems of scene understanding. They are commonly referred to as object detection and pose estimation, and are critically relevant for applications in robotics [27, 56, 57], autonomous driving [28, 46] and augmented reality [41]. In object pose estimation, one distinguishes between methods that work only on known instances [19, 29, 32, 47, 63], and methods that are able to generalize within object categories [12, 15, 33, 35, 54, 55, 61, 62].

Category-level pose estimation has made significant progress in recent years [12, 15, 33, 35, 54, 55, 61, 62]. However, most current work still relies on accurate depth information to recover precise poses [12, 61], restricting the use to devices and working conditions where depth can be captured. Pose estimation from RGB images alone [15, 30, 36, 55, 62] is less explored and a much harder task.

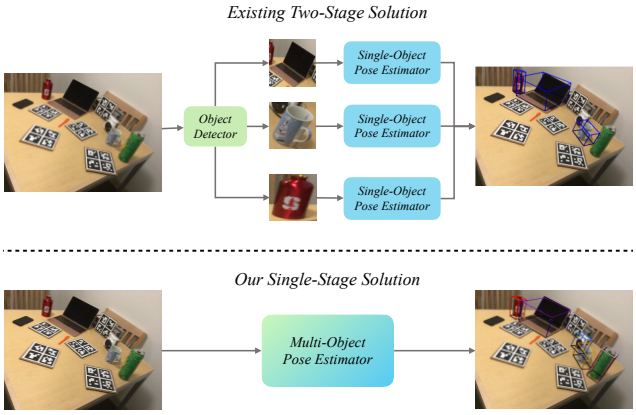

Figure 1. The top shows existing pipelines that are made up of separate stages for object detection and and single-object pose estimation. In the bottom, we show our pipeline that uses a single representation and model for detection and pose estimation.

Methods can be further classified according to the used detection approach. Although there are some RGB-D methods that utilize a shared representation with individual detection and pose heads [24, 54], most methods completely decouple the problem and use separate models and representations. All current RGB methods fall into this category of two-stage methods and use separate models per task, where as shown in Figure 1, objects are first detected, cropped, and then passed into a single-object pose estimator. This naturally places strong emphasis on the accuracy of the detection model, as failures cannot be recovered later, and comes with the downside of having to maintain two separate models.

For the first time, we show that a single unified model can be established for both tasks in the case of RGB images, and that this approach leads to improved accuracy and robustness. Our model leverages neural mesh models [2, 40] as a 3D representation of object categories, where a 2D feature extractor for images and the neural features on the

*Equal contribution. Corresponding email: {tom.fischer, xiaojie.zhang}@utn.de.

meshes are learned jointly during training. During inference, the image features are matched with features of the 3D meshes and multi-model RANSAC PnP [5] is used to detect different object instances and estimate their poses.

Our approach achieves a new state of the art and outperforms the current RGB-only state-of-the-art [62] in pose estimation by 22.9% when averaged across all scale-agnostic metrics. The results show that a common error source in two-stage pipelines is the detector and that our single-stage model is significantly more robust than its two-stage counterparts. In summary, our contributions are:

- We propose the first single-stage framework for category-level multi-object pose estimation from single RGB images.
- For the first time, we leverage neural mesh models jointly for object detection and pose estimation by combining them with multi-model RANSAC PnP and subsequent pose refinement.
- We achieve state-of-the-art performance on pose estimation benchmark REAL275 w.r.t. to accuracy and robustness, verifying the effectiveness of our object representation and single-stage multi-object inference pipeline.

2. Related Work

Both object detection [8, 44] and object pose estimation [37] are long-standing tasks in computer vision. Although the first has converged to broadly accepted models, a variety of approaches are still being explored for the latter. Early pose estimation methods [6, 14, 45] operate by matching hand-crafted features to instance-level 3D models. Today, deep learning has replaced feature descriptors, but many works are still restricted to estimate poses of object instances that were present during training [19, 29, 32, 47, 63].

Category-level pose estimation [12, 35, 54, 61, 62] was introduced by [54] to allow generalization to unseen instances within a set of established categories without requiring additional training or accurate CAD models during inference. The strong variation in geometry and scale within categories makes this a highly challenging problem. Shape-prior-free methods [12, 33, 35, 54] try to learn a generalizable representation without any guidance from CAD models during training.

Shape-prior-based methods, on the other hand, use CAD models and/or simple derived properties such as sizes or bounding boxes to integrate geometric priors. The pose is then either regressed directly from a trained network, or solved for using predicted correspondences in the form of a Normalized Object Coordinate Space (NOCS) [54] with the (often deformed) shape prior using a non-differentiable alignment algorithm such as PnP [21, 31] or the Umeyama algorithm [52]. Our method falls into the above category, but estimates the correspondences from feature matching with 3D neural meshes.

Neural Mesh Models [2] are shape-prior-based methods and were initially proposed for robust category-level single-object rotation estimation. Follow-up work extended it to 6D pose estimation [40], however, they still relied on single-object training datasets and both methods [2, 40] are constrained to per-category networks that have to be calibrated against each other or require a (multi-label) classification model to identify present object categories. Later works removed the need for separate feature representations to allow robust classification, but they cannot estimate translations or handle multiple objects [16, 26]. Our work removes all these restrictions and can both detect objects and estimate their poses with a single, shared representation.

2.1. Two-Stage Methods

The vast majority of object pose estimation methods first locates the objects using off-the-shelf detection models [18, 44] to detect Regions of Interest (RoI) as 2D segmentation masks or bounding boxes. The objects are then cropped from the image and processed by a second model to predict the final poses, splitting the approaches into two stages.

RGB-D Approaches have access to the incomplete point clouds of the observed instances derived from the depth map. These can be used for regression in combination with [34] or without [12, 35, 53] shape priors, or to align them with a shape prior [9, 51] using the Umeyama algorithm [52]. In contrast to the above, our method does not require depth as input.

RGB-Only Approaches face a more challenging problem due to the inherent ambiguity in this setting, which is why these methods are less prominent. In particular, the lack of depth information introduces a fundamental ambiguity between distance and scale, making it impossible to recover these properties from observations. To this end, Zhang *et al.* [62] introduced a set of scale-agnostic metrics that we also utilize in our work. OLD-Net [15] and DMSR [55] leverage a shape prior-based alignment technique and LaPose [62] uses a shape prior-based regression approach. MSOS [30], on the other hand, does not utilize any shape prior and directly estimates NOCS shape and the metric-scale shape of the object for downstream alignment. Although these two-stage approaches are successful in practice, they learn separate representations for both stages and are prone to errors of the first stage. In contrast to the above, we present a single-stage method with higher accuracy and robustness.

2.2. Single-Stage Methods

Single-stage methods couple detection and pose estimation into an end-to-end architecture. Existing work on end-to-end pose estimation focuses mainly on the instance-level

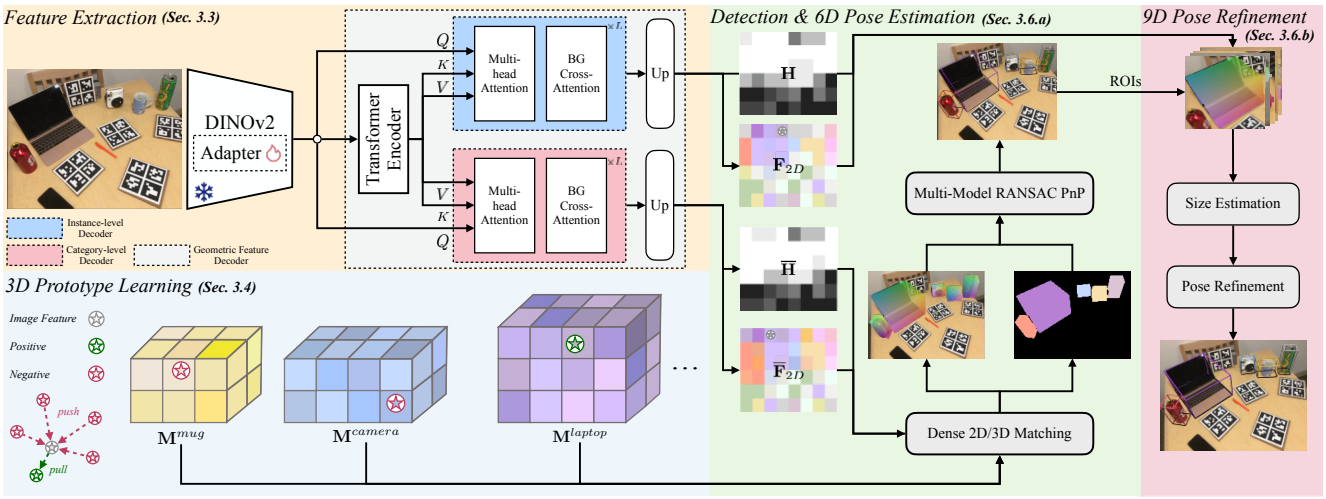

Figure 2. We present a unified framework for solving detection and pose estimation with 3D object prototypes. *Feature Extraction* (Sec. 3.3): Given an RGB image, a frozen DINOv2 model with trainable adapters is combined with a Geometric Feature Decoder to produce two feature maps $\overline{\mathbf{F}}_{2D}, \mathbf{F}_{2D}$ and foreground segmentations $\overline{\mathbf{H}}, \mathbf{H}$. *3D Prototype Learning* (Sec. 3.4): During training, the Adapters and the Geometric Feature Decoder are jointly trained with a set of Neural Mesh Models [2] using a contrastive loss. *Detection & 6D Pose Estimation* (Sec. 3.6 a): After training, 6D object poses are obtained with a multi-model fitting algorithm from dense 2D/3D correspondences between category-level features $\overline{\mathbf{F}}_{2D}$ and all vertex features. *9D Pose Refinement* (Sec. 3.6 b): For each identified instance, a deformation parameter is optimized for by aligning correspondences from instance-level features $\mathbf{F}_{2D}$ with its 6D pose.

setting [7, 11, 22, 23, 49, 50, 58, 60]. However, there are category-level methods that utilize a shared representation and task-specific heads. NOCS [54] uses a shared feature extraction backbone that provides ROIs paired with individual heads for NOCS-map and instance segmentation. The category-level approaches [24, 25] require RGB-D as input and are shape-prior-based regression methods that perform joint center-point detection, reconstruction, and pose estimation. CenterPose [36] comes close to an RGB-only single-stage method, but requires the class labels of the objects present in an image and therefore is not a stand-alone method. In contrast, our work proposes the first RGB-only single-stage approach to unify the complete detection and pose estimation in a shared representation.

3. Method

We present a single-stage framework for RGB-only object detection and category-level 9D pose estimation. Our method aligns image features with a set of 3D object prototypes to infer object properties from 2D/3D correspondences obtained from feature matching. An overview is shown in Fig. 2.

3.1. Task Definition

Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ and camera intrinsics $\mathbf{K} \in \mathbb{R}^{3 \times 3}$, our goal is to estimate the set of object properties:

$$\mathcal{O}(\mathbf{I}, \mathbf{K}) = \{(c_i, \mathbf{s}_i, \mathbf{R}_i, \mathbf{t}_i)\}_{i=1}^{N}, \quad (1)$$

where each object i is described by category c_i, 3D size $\mathbf{s}_i \in \mathbb{R}^3$, rotation $\mathbf{R}_i \in SO(3)$, and translation $\mathbf{t}_i \in \mathbb{R}^3$. We define its scalar instance scale as $\sigma_i = \|\mathbf{s}_i\|_2$. Throughout the remaining parts of this section, we refer to sets in script font (i.e. $\mathcal{O}$); vectors, tuples, and tensors in bold font (i.e. $\mathbf{R}$); and scalars in standard math font (i.e. c).

3.2. Representation

We represent 3D object prototypes with Neural Mesh Models [2, 16, 26, 40], which have shown high generalization and robustness for single-object pose estimation. Each category c has a prototypical mesh $\mathbf{M}^c = (\mathbf{V}^c, \mathbf{A}^c, \mathbf{F}^c_{3D})$, where $\mathbf{V}^c \in \mathbb{R}^{V \times 3}$ are vertex coordinates in object space, $\mathbf{A}^c \in \mathbb{Z}^{A \times 3}$ are triangle indices, and $\mathbf{F}^c_{3D} \in \mathbb{R}^{V \times D}$ are learned vertex features. The geometry of each mesh is obtained by regularly sampling the surface of the bounding box obtained from the average size $\bar{\mathbf{s}}^c$ of the training set, which we normalize to have $\|\bar{\mathbf{s}}^c\|_2 = 1$. We also keep a scaling factor $\bar{\sigma}^c$ that transforms the mesh into one that has the mean scale of the category. If camera poses are available, scenes can be composed from the object prototypes and rendered into an image.

3.3. Feature Extraction

Accurate object localization from 2D/3D correspondences requires that the object geometry aligns well with the visual observations. In category-level pose estimation, one does not have access to instance-level geometries. To address this, we train a 2D feature extractor Φ that learns to map the

input image $\mathbf{I}$ to a rendering of a scene composed of cuboid prototypes parameterized by $\mathcal{O}(\mathbf{I}, \mathbf{K})$.

Modeling. As solving for 9D poses from correspondences alone is challenging and tends to fail in practice, we first perform an initial 6D pose estimation, where we rely on a feature map $\overline{\mathbf{F}}_{2D}$ that is aligned with the prototypes that follow category-level sizes $\overline{\mathbf{s}}^c$. Next, we perform a size refinement with a feature map $\mathbf{F}_{2D}$, which is aligned with deformed prototypes that match the instance-level sizes $\mathbf{s}_i$. Since both feature maps are conditioned on the same neural textures, $\overline{\mathbf{F}}_{2D}$ can be imagined as a warped variant of $\mathbf{F}_{2D}$.

Backbone. The feature extractor Φ is trained jointly with the vertex features. This introduces ambiguities that make a good initialization crucial to guide each $\mathbf{F}^c_{3D}$ towards a meaningful representation. Therefore, we build Φ on top of the self-supervised pre-trained DINOv2 ViT [42] and embed task-specific information using Parameter-Efficient Fine-Tuning (PEFT) [59]. Following [10], we use a low-rank adaptation (Adapter) in the MLP in each transformer block:

$$x'_l = \text{MLP}(\text{LN}(x_l)) + x_l + \lambda(\text{GeLU}(\text{LN}(x_l) \cdot \boldsymbol{D}) \cdot \boldsymbol{U}, \quad (2)$$

where x_l is the output of the previous multi-head attention, λ is a scaling factor [10] and $\boldsymbol{D}, \boldsymbol{U}$ are the down- and up-projection matrices.

Geometric Feature Decoder. To obtain the features corresponding to the object prototype geometry, we first apply a shared transformer encoder to the backbone features, and subsequently two transformer decoders to obtain $\overline{\mathbf{F}}_{2D}, \mathbf{F}_{2D}$. Finally, both feature maps are upsampled to a certain output resolution using a combination of bilinear upsampling and shared convolution layers.

3.4. Training

We consider a training sample with N objects and outline the loss calculation for an object i with prototypical mesh $\mathbf{M}_i = (\mathbf{V}, \mathbf{A}, \mathbf{F}_{3D})$, intrinsic matrix $\mathbf{K}$, and pose $\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i$.

Setup. We first obtain pixel-level training annotations from $\texttt{rasterize}(\mathbf{M}_i, \mathbf{K}, \hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i)$ to obtain projected image location p_k and binary visibility o_k for each vertex $v_k \in \mathbf{V}$. Given a 2D feature map $\mathbf{F}$, we define the annotation set as $\mathcal{Y}(\mathbf{M}_i, \mathbf{F}) = \{(\theta_k, f_k, o_k) \mid \forall v_k \in \mathbf{V}\}$, where f_k is the pixel feature at projection p_k in $\mathbf{F}$, and θ_k is the corresponding vertex feature from $\mathbf{F}_{3D}$.

Optimizing Φ. Following [2], we model the probability distribution of any feature f being generated from vertex v_k by defining $P(f|\theta_k)$ using a von Mises-Fisher (vMF) distribution to express the likelihood:

$$P(f|\theta_k, \kappa) = C(\kappa) e^{\kappa(f^\top \cdot \theta_k)}, \quad (3)$$

with mean θ_k, concentration parameter κ, and normalization constant $C(\kappa)$.

Accurate object poses are obtained when f_k can be matched to θ_k, leading to precise 2D/3D correspondences. We optimize for this task using contrastive learning by maximizing Eq. (3) with the feature f_k, and minimize the likelihood of all other vertices:

$$\max \quad P(f_k|\theta_k, \kappa), \quad (4)$$

$$\min \sum_{\theta_m \in \overline{\theta}_k} P(f_k|\theta_m, \kappa), \quad (5)$$

where the negatives $\overline{\theta}_k$ are the features of all non-corresponding vertices including those from different classes. Equations 4 and 5 can be combined into a single loss by taking the negative log-likelihood:

$$\mathcal{L}(\mathcal{Y}(\mathbf{M}_i, \mathbf{F})) = -\sum_k o_k \cdot \log\left(\frac{e^{\kappa(f_k \cdot \theta_k)}}{\sum_{\theta_m \in \overline{\theta}_k} e^{\kappa(f_k^\top \cdot \theta_m)}}\right), \quad (6)$$

where considering κ as a global hyperparameters allows canceling out the normalization constants $C(\kappa)$. We compute the full training loss as:

$$\mathcal{L}_{train} = \sum_i \frac{\mathcal{L}(\mathcal{Y}(\mathbf{M}^{c_i}, \overline{\mathbf{F}}_{2D})) + \mathcal{L}(\mathcal{Y}(\phi(\mathbf{M}^{c_i}), \mathbf{F}_{2D}))}{2N}, \quad (7)$$

where $\phi(\mathbf{M}) = (\phi(\mathbf{V}), \mathbf{A}, \mathbf{F}_{3D})$ is the deformation operation we discussed in Sec. 3.3. For mean and deformed meshes we scale the vertices with ground-truth scale σ_i before rasterizing.

Optimizing Vertex Features. Previous work [2, 3, 16, 40] advocates for an exponential moving average strategy when updating vertex features. We found that this approach is sensitive to initialization and hyperparameters that lead to highly varying performance between training runs. For a more stable and consistent training, we optimize them through backpropagation with Φ using the same loss in Eq. (7).

3.5. Foreground Detection

Although the contrastive loss in Eq. (7) helps localize objects implicitly, background regions still produce false-positive matches. Inspired by the emergent behavior of foreground segmentation in ViT attention maps [1], we extract foreground masks $\overline{\mathbf{H}}, \mathbf{H}$ using cross-attention between foreground tokens and decoder features.

We initialize two foreground tokens $\overline{\boldsymbol{b}}, \boldsymbol{b}$ with the **CLS**−token of the backbone ViT. After each decoder layer or upsampling step, the per-pixel affinity to the foreground token is measured via cross attention:

$$\boldsymbol{b}^{i+1}, \boldsymbol{W}^{i+1} = \text{CrossAttn}(\boldsymbol{b}^i, F^i), \quad (8)$$

where F^i are the current image features, $\boldsymbol{b}^{i+1}$ is the updated foreground token, and $\boldsymbol{W}^{i+1}$ are the attention weights. At

the final level, we use min-max normalization to obtain the segmentation $\mathbf{H} = \texttt{minmax}(\boldsymbol{W}^N)$. The mean-shape segmentation $\overline{\mathbf{H}}$ is predicted in the same way.

We supervise the final segmentations with masks obtained from rendering the prototypes using ground-truth poses. Since it is a popular choice for segmentation [13, 18] and we have found practical success with it, we adopted the dice loss [48] to optimize the attention weights:

$$\mathcal{L}_{mask} = \frac{\mathcal{L}_{dice}(\mathbf{H}, \mathbf{H}_{GT}) + \mathcal{L}_{dice}(\overline{\mathbf{H}}, \overline{\mathbf{H}}_{GT})}{2}. \quad (9)$$

3.6. Inference

During inference time, we determine the objects contained in an image $\mathbf{I}$ using the trained feature extractor Φ and the prototypes through correspondences from feature matching. First, Φ is queried for the feature maps $\overline{\mathbf{F}}_{2D}, \mathbf{F}_{2D}$ and the foreground heatmaps $\overline{\mathbf{H}}, \mathbf{H}$, which are thresholded using a confidence parameter t_1 to obtain binary foreground segmentations. Then, two dense correspondence mappings $\bar{\mathcal{N}}^{2D}_{3D}, \mathcal{N}^{2D}_{3D}$ are established between the prototypes and the feature maps. Considering $\overline{\mathbf{F}}_{2D}$, the set of correspondences is established as:

$$\begin{aligned} \bar{\mathcal{N}}^{2D}_{3D} = \{(\boldsymbol{p}_i, v_k, \rho(v_k)) \mid \boldsymbol{p}_i \in \overline{\mathbf{F}}_{2D}, \\ v_k = \underset{\theta_k \in \bigcup_c \mathbf{F}^c_{3D}}{\arg\max} f_i^\top \theta_k\}, \end{aligned} \quad (10)$$

where the membership function $\rho(v_k) = c$ returns the category label c of the mesh to which the matched vertex v_k belongs. We remove noisy correspondences which fall outside of $\overline{\mathbf{H}}$ or have a similarity score below a second confidence threshold t_2.

a) Detection and 6D Pose Estimation. We use the correspondence set $\bar{\mathcal{N}}^{2D}_{3D}$ to simultaneously detect objects and estimate their initial 6D pose. If it is known a-priori that there is only a single instance per category in the scene, one could group correspondences by category label and solve this problem via sequential application of PnP [31] paired with RANSAC [17]. However, this assumption is rarely valid in practice. To handle scenes with multiple instances from the same category, we use Progressive-X (ProgX) [5], a multi-model fitting algorithm that enables PnP to detect multiple object poses. ProgX extends Graph-Cut RANSAC [4] to the multi-model setting via a sequential hypothesis generation algorithm that can separate correspondences into individual instances. For each class c, we run ProgX on the subset $\bar{\mathcal{N}}^c \subset \bar{\mathcal{N}}^{2D}_{3D}$ to obtain a set of 6D poses:

$$\mathcal{P}_{6D} = \bigcup_c \left\{(\overline{\mathbf{R}}, \bar{\mathbf{t}}, c) \mid \forall (\overline{\mathbf{R}}, \bar{\mathbf{t}}) \in \text{MM-PnP}(\bar{\mathcal{N}}^c)\right\}, \quad (11)$$

b) 9D Pose Refinement. For each detected object i, we refine the initial pose $(\overline{\mathbf{R}}_i, \bar{\mathbf{t}}_i, c) \in \mathcal{P}_{6D}$ to account for instance-specific size and improve their accuracy using $\mathcal{N}^{2D}_{3D}$. We select a subset $\mathcal{N}_i \subset \mathcal{N}^{2D}_{3D}$, where 1) the 2D point lies within $\mathbf{H}$, 2) its similarity score is above the threshold t_2, and 3) its 2D coordinate was considered an inlier according to the 6D pose and category-level geometry.

We deform the category-level mesh by scaling vertices along the principal axes with a deformation multiplier $\mathbf{d} \in \mathbb{R}^3$. Refinement is done by minimizing the reprojection error:

$$\boldsymbol{E}(\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{d}, \mathcal{N}_i) = \sum_{v_k, \boldsymbol{p}_k \in \mathcal{N}_i} \|(\mathbf{K}\mathbf{R}(\mathbf{d} \odot v_k) + \mathbf{t}) - \boldsymbol{p}_k\|^2_2. \quad (12)$$

The optimization proceeds in two steps, where the deformation is obtained using the 6D rotation and translation, which are subsequently refined using the deformation:

$$\mathbf{d}_i = \min_{\mathbf{d}} \boldsymbol{E}(\mathbf{K}, \overline{\mathbf{R}}_i, \bar{\mathbf{t}}_i, \mathbf{d}, \mathcal{N}_i) \quad (13)$$

$$\mathbf{R}_i, \mathbf{t}_i = \min_{\mathbf{R}, \mathbf{t}} \boldsymbol{E}(\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{d}_i, \mathcal{N}_i). \quad (14)$$

The final pose of object i is then given as $(\mathbf{R}_i, \mathbf{t}_i, \mathbf{s}_i, c_i)$, where $\mathbf{s}_i$ is obtained as the size of the bounding box of $\mathbf{V}^{c_i}$ multiplied with $\mathbf{d}_i$.

4. Experiments

In the following, we explain our experimental setup and then discuss the results of our single-stage pose estimation approach on real, synthetic, and corrupted data. For further results and more detailed comparisons, we refer the reader to our supplemental material.

4.1. Experiment Setup

Datasets. We evaluate our method on the popular category-level 9D pose estimation benchmarks CAMERA25 [54] and REAL275 [54]. REAL275 is a real-world dataset that consists of 4.3K training images of 7 scenes and 2.75K testing images of 6 scenes. CAMERA25 supplements the real data with 275K synthetic training images and 25K evaluation images of rendered objects with real-world backgrounds. Both datasets contain objects from six categories: bottle, bowl, camera, can, laptop, and mug.

Evaluation Metrics. Following previous work [62], we use scale-agnostic evaluation metrics to remove the scale-depth ambiguity. To evaluate 3D detection and object size estimation, we report the mean average of the Normalized 3D Intersection over Union (NIoU) metric at thresholds 25%, 50%, and 75%. For pose estimation, we individually report the rotation error and scale-normalized translation error at the different thresholds $0.2d$, $0.5d$, $5°$, $10°$, and on joint thresholds $5°0.2d$, $5°0.5d$, $10°0.2d$, $10°0.5d$. In addition, we also report results on absolute metrics for comparable thresholds. Our work focuses on joint detection and pose estimation; therefore, we do not remove objects with

Method	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
MSOS [30]	36.9	9.7	0.7	-	-	3.3	15.3	10.6	50.8	-	17.0
OLD-Net [15]	35.4	11.4	0.4	0.9	3.0	5.0	16.0	12.4	46.2	4.2	20.9
DMSR [55]	57.2	38.4	9.5	15.1	23.7	25.6	45.2	35.0	67.3	27.4	52.0
LaPose [62]	70.7	47.9	15.8	15.7	21.3	37.4	57.4	46.9	78.8	23.4	60.7
Ours	**75.2**	**53.7**	**19.2**	**25.1**	**31.8**	**43.7**	**66.1**	**53.5**	**83.7**	**32.1**	**68.8**

Table 1. Comparison with state-of-the-art methods on REAL275 using scale-agnostic evaluation metrics. As visible, we set a new state of the art by outperforming the baseline methods in all metrics.

Method	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
MSOS [30]	35.1	9.9	0.8	-	-	5.9	31.6	8.9	47.2	-	48.6
OLD-Net [15]	48.7	19.5	1.7	4.7	15.9	12.9	39.6	19.1	60.0	20.5	50.5
DMSR [55]	73.6	45.1	11.1	26.1	43.6	38.2	67.5	42.5	79.6	47.3	74.2
LaPose[62]	**74.1**	45.2	12.5	29.4	53.9	39.2	74.4	41.2	80.2	58.5	**82.7**
Ours	73.9	**47.6**	**18.1**	**37.2**	**57.6**	**45.2**	**75.5**	**47.6**	**83.4**	**60.5**	80.8

Table 2. Comparison with state-of-the-art methods on CAMERA25 using scale-agnostic evaluation metrics. We assume that the lower $10°$ performance is due to small objects in CAMERA25, yielding less correspondences. Our method achieves the highest accuracy on 9 out of the 11 metrics.

Method	IoU_{50}	IoU_{75}	$5°5cm$	$5°10cm$	$10°5cm$	$10°10cm$
MSOS [30]	13.6	1.0	-	-	-	11.8
OLD-Net [15]	8.1	0.3	0.5	2.0	2.8	10.4
DMSR [55]	15.9	**3.3**	6.1	13.4	10.5	25.0
LaPose [62]	**17.5**	2.6	6.3	13.6	**12.5**	**30.5**
Ours	**17.5**	2.6	**7.2**	**17.7**	10.4	29.4

Table 3. Results on REAL275 using absolute metrics. Predicting object scales from RGB images alone is challenging and not having 2D bounding boxes during inference exacerbates this problem, leading to worse performance of our trained scale prediction network, and consequently less precise metric scale translations. Nevertheless, our method is competitive in performance, either matching or outperforming the baselines in 3 out of the 6 considered metrics.

low detection thresholds as in previous work [54], but evaluate performance on all detected objects as proposed by [62].

Implementation Details. To train our model on the REAL275 dataset, we use a mixture of 25% real-world images and 75% synthetic images from the CAMERA25 training set, similar to [54]. For the results on CAMERA25, we follow prior work [15, 30, 55, 62] and train a separate model without the REAL275 images.

For each baseline, we use the detection results from [51] for a fair comparison and reproduce the numbers with the public checkpoints of their methods where possible. Notably, MSOS [30] does not have code publicly available at the time of writing, which is why we use the numbers provided by [62]. For more implementation details about our method, we refer the reader to our supplementary material.

4.2. Comparison with State-of-the-Art Methods

Results with Scale-Agnostic Metrics. In Tab. 1, we compare our method with state-of-the-art RGB-only category-level pose estimation methods on the REAL275 benchmark under scale-agnostic evaluation metrics. As visible, our method consistently outperforms all baselines under scale-agnostic metrics.

As can be seen in Tab. 2, our method also shows strong performance on synthetic data. While it falls behind on coarse rotation accuracy, it still outperforms baselines in 9 out of 11 metrics. A likely reason is that CAMERA25 contains many smaller objects (high depth or small scale), and performance could likely be increased by using higher-resolution features at the cost of slower inference speed.

Results with Absolute Metrics. To measure pose estimation quality under metric scale, we follow [62] and decouple scale prediction from pose estimation. Since we do not use a dedicated detection model, we cannot use their pre-trained model. Instead, we train the same scale network using their code to predict a per-instance scale parameter, given the 2D projection of our object prototype with the estimated pose.

We show the performance of our method in combination with the scale prediction network Tab. 3. Our results are competitive due to the strong performance in the normalized space. However, we could not reproduce the same level of performance from[62] for the scale estimation network. We believe that this is due to the network that was trained on 2D bounding boxes being more robust during inference than our version, which was trained on the prototype projections.

Qualitative Results. Finally, Fig. 3 shows estimated

Source	Method	$\Delta NIoU_{25}$	$\Delta NIoU_{50}$	$\Delta NIoU_{75}$	$\Delta 5°0.2d$	$\Delta 5°0.5d$	$\Delta 10°0.2d$	$\Delta 10°0.5d$	$\Delta 0.2d$	$\Delta 0.5d$	$\Delta 5°$	$\Delta 10°$
ROI	OLD-Net[15]	-14.4%	-24.6%	-50.0%	-44.4%	-36.7%	-28.0%	-23.8%	-16.1%	-10.8%	-23.8%	-18.2%
	DMSR[55]	- 3.3%	- 7.3%	-13.7%	-17.9%	-18.1%	- 8.2%	-10.4%	- 4.6%	- 2.7%	-17.5%	-11.5%
	LaPose[62]	- 9.3%	-16.9%	-18.4%	-28.7%	-23.9%	-25.4%	-16.6%	-18.3%	- 6.2%	-21.4%	-14.3%
Image	OLD-Net[15]	-28.0%	-38.6%	-50.0%	-55.6%	-43.3%	-46.0%	-41.3%	-34.7%	-26.4%	-23.8%	-24.9%
	DMSR[55]	-13.8%	-14.3%	-22.1%	-24.5%	-27.8%	-16.4%	-22.1%	-13.1%	-15.2%	-26.6%	-22.5%
	LaPose[62]	-22.9%	-24.6%	-22.8%	-29.3%	-28.6%	-29.4%	-27.7%	-25.6%	-21.2%	-26.1%	-25.9%
	Ours	**-11.0%**	**-14.0%**	**-21.7%**	**-17.2%**	**-13.5%**	**-16.0%**	**-12.9%**	**-12.7%**	**- 9.5%**	**-13.5%**	**-12.6%**

Table 4. Robustness experiments on the REAL275 [54] benchmark. We report loss in accuracy compared to the model evaluated on uncorrupted data averaged over 8 types of corruptions. The top shows results for corruption-free detection with only corrupted crops during pose estimation and the bottom shows corruptions during both stages. Corruptions during pose estimation alone already reduce the performance significantly. However, the on the image level the two-stage approaches suffer even more, indicating that failures in the detection are severe. Our single-stage approach is much less affected by noise and shows the greatest robustness.

Detection Source	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
GT Object Mask	72.1	50.3	17.8	23.0	29.6	40.4	61.4	50.4	81.0	29.9	63.7
GT Proto. Mask	**79.5**	**55.3**	**19.5**	**25.6**	**32.2**	**44.1**	**66.7**	53.4	**84.2**	**32.5**	**69.2**
Mask-RCNN [18]	70.4	48.8	17.7	22.4	28.7	38.9	58.4	48.9	79.1	28.9	60.4
None (ours)	75.2	53.7	19.2	25.1	31.8	43.7	66.1	**53.5**	83.7	32.1	68.8

Table 5. To isolate the pose estimation accuracy of our method, we frame the problem as single-object pose estimation by providing different ROIs. As visible, our method comes close to the upper bound (GT Proto. Mask), showing our model can detect objects well and simultaneously reason about their poses with high accuracy. Our method also works well on the tight "GT Object Masks" but the performance drops significantly when using the fine-tuned Mask-RCNN, indicating that our approach outperforms it.

poses on REAL275 of our method and the baselines. As can be seen, two-stage approaches suffer from the error propagation of detection failures, whereas our method identifies objects much more confidently. Additionally, our method significantly improves the rotation accuracy which is most notable in the laptop category. For more detailed per-category results, we refer the reader to our supplemental material.

4.3. Robustness to Image Degradations

Setting. To verify the robustness of two-stage vs. single-stage pose estimation, we conduct experiments on the REAL275 dataset with added corruptions from the imagecorruptions [20] library. We choose 8 corruptions of different types and report the average performance in the main part of the paper. For the choice of corruptions and per-corruption performance, we refer the reader to the supplemental material. For consistency, each method is evaluated using the same set of images and has been trained with the same data augmentations.

We design two sets of experiments to study the robustness of the two-stage baselines. To measure the robustness of the pose estimator in isolation, we run the detector on clean images and apply the corruption only on its provided RoI. To evaluate the robustness of the whole framework, we apply corruptions on an image level. As it is the most commonly used model, we use the pre-trained Mask-RCNN [18] from [51] to get detection results for all baselines. Note that we omit MSOS [30] from the robustness experiments, since to the best of our knowledge their code is not (or no longer) public.

Results. We report the decrease in performance under image corruptions for the full task at the bottom of Tab. 4. Our performance drops by only 14% when considering the mean and median on all metrics. In comparison, baselines suffer much more, losing anywhere from $19\% - 37\%$ mean or $22\% - 38\%$ median performance.

4.4. Pose Estimation Quality

By framing detection and pose estimation as a single-stage problem, it is hard to quantify the success of either component individually. To measure the quality of our pose estimation in isolation, we try to decouple our framework into two stages. We do so by obtaining masks from three different sources. First, we use ground-truth object segmentation masks. Second, we use projected object prototype masks using ground-truth 9D poses. Third, we use bounding boxes provided by the same fine-tuned Mask-RCNN [18] as our baselines. The masks are then used to identify per-object correspondences for single-object pose estimation. To be consistent with our main method, we also use Progressive-X [5] as the solver, but we limit the maximum of predicted models to one per RoI.

Tab. 5 shows the results of providing RoIs to our method. It is important to note that using instance-level detections from either Mask-RCNN or the ground-truth segmentation

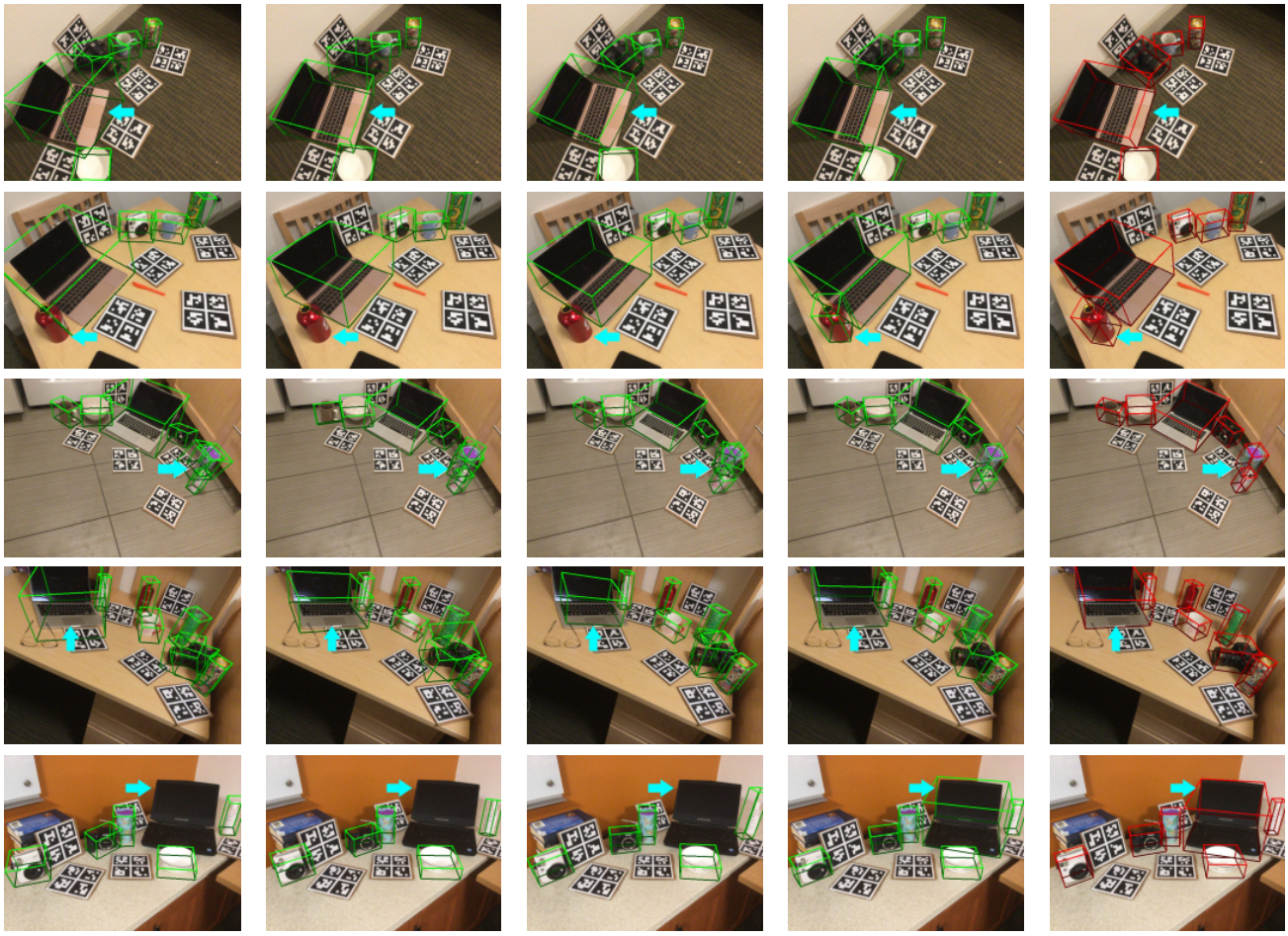

Figure 3. Qualitative comparison on REAL275[54]. We compare our model (second to last column) with all baselines and with ground truth (last column). Some failures due to poor detection results of the baselines are highlighted (row 2, 3, and 5). In rows 1 and 4 our method outperforms baselines in terms of rotation accuracy.

mask is not ideal for our method as it is optimized to work with the projected prototype masks. However, we find that our method also works well with the tightly fitting ground-truth segmentation masks (first row) but the performance drops steeply when using the predicted Mask-RCNN masks. The upper bound is shown in the second row, which provides ground-truth detections and 2D correspondences to our method. As can be seen, our method (fourth row) only falls noticeably behind in the $NIoU_{25}$ and performs close to the upper bound on all other metrics. This shows that our joint approach outperforms Mask-RCNN in detection performance.

5. Conclusion

In this work, we introduced the first unified RGB-only category-level detection and pose estimation framework that operates with a single representation. Using neural mesh models as 3D object prototypes leads to robust feature representations that can be used for 2D/3D correspondence estimation and downstream multi-model pose estimation to jointly solve the detection and pose estimation tasks. We achieve state-of-the-art results for pose estimation on the popular REAL275 benchmark, and our evaluation has shown that our single-stage model is significantly more robust to image degradations than its two-stage counterparts.

Acknowledgements

We gratefully acknowledge the stimulating research environment of the GRK 2853/1 “Neuroexplicit Models of Language, Vision, and Action”, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under project number 471607914. The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Com-

puting Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the BayernKI project b266be. BayernKI funding is provided by Bavarian state authorities.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 4

[2] Wang Angtian, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. In *Proceedings International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3, 4

[3] Yutong Bai, Angtian Wang, Adam Kortylewski, and Alan Yuille. Coke: Contrastive learning for robust keypoint detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 65–74, 2023. 4

[4] Daniel Barath and Jiri Matas. Graph-cut RANSAC. In *CVPR*, 2018. 5

[5] Daniel Barath and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3780–3788, 2019. 2, 5, 7

[6] Tolga Birdal and Slobodan Ilic. Point pair features based object detection and pose estimation revisited. In *2015 International conference on 3D vision*, pages 527–535. IEEE, 2015. 2

[7] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach. *arXiv preprint arXiv:2011.04307*, 2020. 3

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

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

[10] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 35:16664–16678, 2022. 4

[11] Wei Chen, Jinming Duan, Hector Basevi, Hyung Jin Chang, and Ales Leonardis. Pointposenet: Point pose network for robust 6d object pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2824–2833, 2020. 3

[12] Yamei Chen, Yan Di, Guangyao Zhai, Fabian Manhardt, Chenyangguang Zhang, Ruida Zhang, Federico Tombari, Nassir Navab, and Benjamin Busam. Secondpose: Se (3)-consistent dual-stream feature fusion for category-level pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9959–9969, 2024. 1, 2, 12

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 5

[14] Changhyun Choi and Henrik I Christensen. 3d pose estimation of daily objects using an rgb-d camera. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3342–3349. IEEE, 2012. 2

[15] Zhaoxin Fan, Zhenbo Song, Jian Xu, Zhicheng Wang, Kejian Wu, Hongyan Liu, and Jun He. Object level depth reconstruction for category level 6d object pose estimation from monocular rgb image. In *European Conference on Computer Vision*, pages 220–236. Springer, 2022. 1, 2, 6, 7, 14, 15, 18

[16] Tom Fischer, Yaoyao Liu, Artur Jesslen, Noor Ahmed, Prakhar Kaushik, Angtian Wang, Alan Yuille, Adam Kortylewski, and Eddy Ilg. inemo: Incremental neural mesh models for robust class-incremental learning. In *ECCV*, 2024. 2, 3, 4

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 5, 7

[19] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020. 1, 2

[20] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 7, 13

[21] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390. IEEE, 2011. 2

[22] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3385–3394, 2019. 3

[23] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020. 3

[24] Muhammad Zubair Irshad, Thomas Kollar, Michael Laskey, Kevin Stone, and Zsolt Kira. Centersnap: Single-shot multi-object 3d shape reconstruction and categorical 6d pose and size estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10632–10640. IEEE, 2022. 1, 3

[25] Muhammad Zubair Irshad, Sergey Zakharov, Rares Ambrus, Thomas Kollar, Zsolt Kira, and Adrien Gaidon. Shapo: Implicit representations for multi-object shape, appearance, and pose optimization. In *European Conference on Computer Vision*, pages 275–292. Springer, 2022. 3

[26] Artur Jesslen, Guofeng Zhang, Angtian Wang, Wufei Ma, Alan Yuille, and Adam Kortylewski. Novum: Neural object

volumes for robust object classification. In *European Conference on Computer Vision*, pages 264–281. Springer, 2025. 2, 3

[27] Daniel Kappler, Franziska Meier, Jan Issac, Jim Mainprice, Cristina Garcia Cifuentes, Manuel Wüthrich, Vincent Berenz, Stefan Schaal, Nathan Ratliff, and Jeannette Bohg. Real-time perception meets reactive motion generation. *IEEE Robotics and Automation Letters*, 3(3):1864–1871, 2018. 1

[28] Nikunj Kothari, Misha Gupta, Leena Vachhani, and Hemendra Arya. Pose estimation for an autonomous vehicle using monocular vision. In *2017 Indian control conference (ICC)*, pages 424–431. IEEE, 2017. 1

[29] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020. 1, 2

[30] Taeyeop Lee, Byeong-Uk Lee, Myungchul Kim, and In So Kweon. Category-level metric scale object shape and pose estimation. *IEEE Robotics and Automation Letters*, 6(4): 8575–8582, 2021. 1, 2, 6, 7, 14

[31] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International journal of computer vision*, 81:155–166, 2009. 2, 5

[32] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 683–698, 2018. 1, 2

[33] Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia, and Yuanqing Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3560–3569, 2021. 1, 2

[34] Jiehong Lin, Zewei Wei, Changxing Ding, and Kui Jia. Category-level 6d object pose and size estimation using self-supervised deep prior deformation networks. In *European Conference on Computer Vision*, pages 19–34. Springer, 2022. 2

[35] Xiao Lin, Wenfei Yang, Yuan Gao, and Tianzhu Zhang. Instance-adaptive and geometric-aware keypoint learning for category-level 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21040–21049, 2024. 1, 2

[36] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A Vela, and Stan Birchfield. Single-stage keypoint-based category-level object pose estimation from an rgb image. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1547–1553. IEEE, 2022. 1, 3

[37] Jian Liu, Wei Sun, Hui Yang, Zhiwen Zeng, Chongpei Liu, Jin Zheng, Xingyu Liu, Hossein Rahmani, Nicu Sebe, and Ajmal Mian. Deep learning-based object pose estimation: A comprehensive survey. *arXiv preprint arXiv:2405.07801*, 2024. 2

[38] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 12

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 12

[40] Wufei Ma, Angtian Wang, Alan Yuille, and Adam Kortylewski. Robust category-level 6d pose estimation with coarse-to-fine rendering of neural features. In *European Conference on Computer Vision*, pages 492–508. Springer, 2022. 1, 2, 3, 4

[41] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015. 1

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4, 12

[43] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 12, 13

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 91–99, 2015. 2

[45] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, pages 2155–2162. IEEE, 2010. 2

[46] Yongzhi Su, Jason Rambach, Nareg Minaskan, Paul Lesur, Alain Pagani, and Didier Stricker. Deep multi-state object pose estimation for augmented reality assembly. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 222–227. IEEE, 2019. 1

[47] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022. 1, 2

[48] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017. 5

[49] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 292–301, 2018. 3

[50] Stefan Thalhammer, Timothy Patten, and Markus Vincze. Cope: End-to-end trainable constant runtime object pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2860–2870, 2023. 3

[51] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 530–546. Springer, 2020. 2, 6, 7

[52] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04): 376–380, 1991. 2

[53] Boyan Wan, Yifei Shi, and Kai Xu. Socs: Semantically-aware object coordinate space for category-level 6d object pose estimation under large shape variations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14065–14074, 2023. 2

[54] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 1, 2, 3, 5, 6, 7, 8, 13, 16, 18

[55] Jiaxin Wei, Xibin Song, Weizhe Liu, Laurent Kneip, Hongdong Li, and Pan Ji. Rgb-based category-level object pose estimation via decoupled metric scale recovery. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2036–2042. IEEE, 2024. 1, 2, 6, 7, 14, 15, 18

[56] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6401–6408. IEEE, 2022. 1

[57] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *arXiv preprint arXiv:2201.12716*, 2022. 1

[58] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 3

[59] Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv preprint arXiv:2402.02242*, 2024. 4

[60] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019. 3

[61] Jiyao Zhang, Mingdong Wu, and Hao Dong. Genpose: generative category-level object pose estimation via diffusion models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 54627–54644, 2023. 1, 2

[62] Ruida Zhang, Ziqin Huang, Gu Wang, Chenyangguang Zhang, Yan Di, Xingxing Zuo, Jiwen Tang, and Xiangyang Ji. Lapose: Laplacian mixture shape modeling for rgb-based category-level object pose estimation. In *European Conference on Computer Vision*, pages 467–484. Springer, 2024. 1, 2, 5, 6, 7, 14, 15, 18

[63] Jun Zhou, Kai Chen, Linlin Xu, Qi Dou, and Jing Qin. Deep fusion transformer network with weighted vector-wise keypoints voting for robust 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13967–13977, 2023. 1, 2

Unified Category-Level Object Detection and Pose Estimation from RGB Images using 3D Prototypes

Supplementary Material

A. Further Implementation Details

In this section, we provide additional details on the training regimen of our method.

Parameters. We train the model on a single NVIDIA H100 GPU for 150k iterations using the *AdamW* [39] optimizer with weight decay of 0.05 and a Cosine Annealing schedule [38] that decays the learning rate from $1e-4$ to $1e-7$. As training and inference-specific hyperparameters, we choose $\kappa = \frac{1}{0.07}$, $t_1 = 0.5$, and $t_2 = 0.7$. The adapters are included in each transformer block of the DINOv2 [42] model with a low-rank dimensionality of 128.

Training Data. To improve generalization, we use a similar data augmentation strategy during training as [12]. In each training iteration, we use a batch size of 10 images which are stacked into dynamic batches according to the present objects.

Annotation Generation. We use PyTorch3D [43] to rasterize the object prototypes into the annotation set. For each object, we render the given pose individually and then extract the per-pixel annotations. To account for objects occluding each other (which happens frequently for the category-level feature map) we mask out pixels where more than one object is visible by setting the visibility to 0. Alternatively, one could also opt for supervising with the closest object. However, we found that this results in the model to miss small, or partially visible objects more frequently.

B. Architectural choices

We show an ablation on the core design choices of our method in Tab. 6.

Adapter. The PEFT strategy to introduce dataset- and task-specific information into the pre-trained feature extractor massively benefits our method. Only by modifying the feedforward part of the transformer blocks shows significant performance improvements which is especially noticeable for rotation accuracy.

Foreground Modeling. Next, we evaluated the effect of our foreground modeling via CrossAttention. Specifically, we compare two variants. First, we evaluate the effect of focusing the model onto the foreground region during training via a baseline that has all CrossAttention layers removed and filters outliers during inference with confidence scores. We found that the same threshold $t_2 = 0.7$ we used for the full model is not ideal in this case and a more robust segmentation is obtained with $t_2 = 0.8$. This naive baseline achieves good results, indicating that our method can identify vertices with high likelihood. Next, we use our full model but ignore the provided mask during inference by setting $t_1 = 0$. This variant consistently outperforms the previous, indicating that focusing the model on the foreground region explicitly during training leads to better representation learning. However, utilizing the foreground mask still provides consistent improvements across all metrics, indicating its importance for correspondence estimation.

Pose refinement. Finally, we ablate over the components in or 9D pose refinement stage that utilizes the features that follow the instance-level prototype geometries. We show that returning the poses obtained from ProgX (i.e. 6D poses) leads to consistently worse performance. Even an instance level refinement using the category-level correspondences $\bar{\mathcal{N}}_{3D}^{2D}$ leads to performance improvements (see row "w refinement"). However, to obtain strong results for the tightest bounding box threshold $NIoU_{75}$ the size optimization using the instance-level correspondences $\mathcal{N}_{3D}^{2D}$ is required (see row "w size estimation"). Best performance, however, is obtained when refining the deformed 2D/3D correspondences, leading to our full pipeline.

C. Inference Speed

We compare the inference speed of our method with the two-stage baselines in Fig. 4. In contrast to baselines, our method requires only a single forward pass. Two-stage methods require one forward pass of the detection model and one call for each detected object. Our method, on the other hand, is more reliant on the choice of output resolution and found correspondences. With a more aggressive outlier rejection or subsampling of correspondences inference speed can be greatly improved with only minor loss in accuracy. In this work, however, we did not optimize for inference speed and consider speed-up strategies as future work.

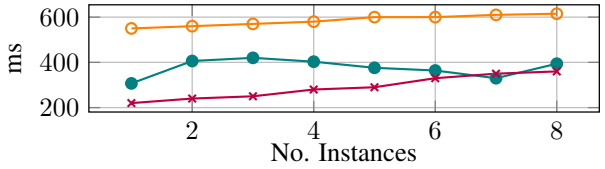

Figure 4. Average inference runtime our method and the baselines LaPose and DMSR. Runtime of our method depends on the found correspondences instead of present instances.

Method	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
Ours	75.2	53.7	19.2	25.1	31.8	43.7	66.1	53.5	83.7	32.1	68.8
w/o adapter	-16.4	- 21.0	-15.9	-19.7	-22.1	-25.8	-31.1	-21.6	-8.7	- 21.8	-30.8
w/o CA - $t_1 = 0, t_2 = 0.7$	- 3.8	- 9.4	- 4.8	- 7.7	- 4.0	- 7.3	- 4.5	- 8.7	- 0.1	- 3.4	- 3.8
w/o CA - $t_1 = 0, t_2 =$opt.	+ 0.8	- 4.9	- 2.6	- 5.7	- 1.9	- 5.6	- 2.9	- 6.3	- 0.3	- 1.7	- 2.9
w CA - $t_1 = 0, t_2 = 0.7$	- 1.4	- 4.9	- 3.0	- 4.2	- 1.9	- 4.5	- 2.6	- 5.4	+ 0.2	- 1.8	- 2.1
w CA - $t_1 = 0, t_2 =$opt.	+ 0.6	- 0.9	- 0.7	- 0.9	- 0.3	- 1.9	- 1.2	- 1.7	- 0.2	- 0.3	- 1.2
w/o refinement + w/o size	- 0.7	- 2.7	- 3.9	- 2.1	- 2.1	- 2.4	- 3.9	- 2.1	- 1.0	- 1.8	- 3.5
w refinement	- 0.3	- 2.4	- 2.8	- 1.3	- 1.1	- 1.9	- 2.7	- 1.7	- 0.8	- 0.8	- 2.2
w size estimation	- 1.0	- 2.2	+ 0.1	- 2.4	- 2.4	- 2.5	- 3.9	- 2.2	- 0.9	- 2.2	- 3.6

Table 6. Ablation over the key design choices in Φ. The adapters are crucial for precise pose estimation and their removal leads to massive performance drops. In the second block, we evaluate without using the foreground mask obtained from the CrossAttention layers and solely from confidence values. "w/o CA" was trained without any CrossAttention layers, with the rest of the architecture being identical. Confidence measures alone lead to reasonable performance, especially with the optimal threshold parameter ($t_2 = 0.8$). However, it is still consistently worse than the network trained with CrossAttention ("w CA"), even without using the mask during inference when setting $t_1 = 0$. In the third block, we ablate over the choices in the instance-level 9D pose refinement part of our pipeline. "w/o refinement + w/o size" refers to directly outputting the poses after ProgX, yielding consistently worse results. "w refinement" refers to the instance level pose refinement given the category-level correspondences, which gives a marginal improvement across all metrics. "w size estimation" includes the size optimization from the instance-level correspondences and shows that this is crucial for good performance on the tight $NIoU_{75}$ metric.

D. Pose Estimation for Overlapping Objects

REAL275 [54] does not contain many overlapping objects of the same category. To showcase that our method can deal with intra-category overlaps we captured in-the-wild images with a smartphone and approximated its intrinsic matrix. We show the predictions of our method in Fig. 5.

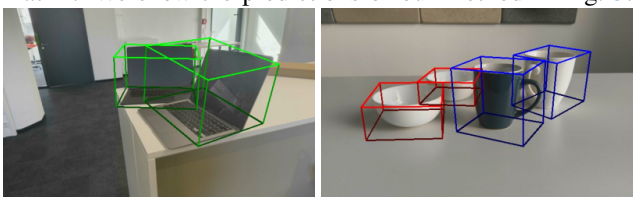

Figure 5. Detections and poses from our model with same-category occlusions on self-captured images.

E. Additional Quantitative Results on Robustness Study

In Tab. 7, we show the pose estimation accuracy under scale-agnostic metrics averaged over all corruption types. The corrupted images are generated using the method proposed in [20] and using their public code. We consider four types of image degradations, encompassing a total of eight corruption types: noise (speckle noise, Gaussian noise), blur (Gaussian blur, defocus blur), digital artifacts (JPEG compression, elastic transformation), and weather effects (frost, fog). The corruption strength follows the default setting, varying in severity per image basis. For a fair comparison, we run each method on the same set of images.

We report the mean scale-agnostic 3D Intersection over Union (NIoU), rotation, and translation metrics for all methods. Additionally, we show results for each corruption type at ROI level in Tab. 9 and at image level in Tab. 8.

F. Qualitative Results on Corrupted Images

Fig. 6 presents qualitative examples of object detection and pose estimation on the corrupted REAL275 dataset. We can observe that image degradations negatively affect the detection and in turn, the final pose estimation accuracy. For instance, when applying elastic transformations the reduced detector accuracy causes the laptop to be missed entirely and introduces redundant detections of the camera. This shows that for degraded images the detection model is a performance bottleneck in current two-stage approaches. In the single-stage approach, we benefit from significantly more robust detection and pose estimation quality.

G. Qualitative Results of Dense Matching

In Fig. 7, we present qualitative results of dense 2D-3D correspondence matching. We render all NOCS-maps with the geometry of our object prototypes using PyTorch3D [43]. To be consistent with our main inference, we remove correspondences with a confidence score below $t_2 = 0.7$. To address object overlap, we prioritize rendering the object closer to the camera. We observe that our method generates less confident correspondences near object edges, likely due to the ambiguity of these regions during optimization. Overall, our method produces high-quality correspondences using a simple nearest-neighbor matching approach, indicating that the contrastive training approach produces descriptive features.

Source	Method	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
None	MSOS [30]	36.9	9.7	0.7	-	-	3.3	15.3	10.6	50.8	-	17.0
	OLD-Net [15]	35.4	11.4	0.4	0.9	3.0	5.0	16.0	12.4	46.2	4.2	20.9
	DMSR [55]	57.2	38.4	9.5	15.1	23.7	25.6	45.2	35.0	67.3	27.4	52.0
	LaPose [62]	70.7	47.9	15.8	15.7	21.3	37.4	57.4	46.9	78.8	23.4	60.7
	Ours	**71.6**	**50.5**	**18.3**	**21.6**	**28.6**	**41.9**	**61.6**	**51.0**	**80.9**	**29.1**	**64.1**
ROI	OLD-Net[15]	30.3	8.6	0.2	0.5	1.9	3.6	12.2	10.4	41.2	3.2	17.1
	DMSR[55]	55.3	35.6	8.2	12.4	19.4	23.5	40.5	33.4	65.5	22.6	46.0
	LaPose[62]	64.1	39.8	12.9	11.2	16.2	27.9	47.9	38.3	73.9	18.4	52.0
Image	OLD-Net[15]	25.5	7.0	0.2	0.4	1.7	2.7	9.4	8.1	34.0	3.2	15.7
	DMSR[55]	49.3	32.9	7.4	11.4	17.1	21.4	35.2	30.4	57.1	20.1	40.3
	LaPose[62]	54.5	36.1	12.2	11.1	15.2	26.4	41.5	34.9	62.1	17.3	45.0
	Ours	**63.8**	**43.4**	**14.3**	**17.9**	**24.7**	**35.2**	**53.7**	**44.5**	**73.2**	**25.2**	**56.0**

Table 7. Ablation study of robustness under image noises. We report the performance of all methods on clean data (top), as well as ROI corruptions for baselines (middle), and image-level corruptions (bottom). Note, that performance of our method on clean data is different due to the changed training regiment to make this comparison fair.

Method	Noise	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
LaPose	Speckle Noise	56.7	35.8	**13.3**	8.6	11.7	25.1	39.8	34.9	64.9	13.9	44.2
DMSR		48.8	31.1	6.4	8.9	15.2	17.9	33.4	27.9	55.7	19.4	40.3
Old-Net		23.2	6.7	0.2	0.2	1.3	2.6	0.3	7.7	32.1	3.2	17.0
Ours		**61.8**	**40.3**	12.6	**15.0**	**22.6**	**31.1**	**50.6**	**40.8**	**71.6**	**23.2**	**53.5**
LaPose	Gaussian Blur	58.2	41.9	17.3	13.4	18.0	31.1	44.1	40.1	64.0	20.2	47.6
DMSR		54.3	40.4	10.3	14.1	18.5	27.7	39.4	36.6	60.3	21.2	43.5
Old-Net		30.7	7.4	0.3	0.4	2.6	3.3	13.2	9.7	39.1	4.1	18.5
Ours		**69.9**	**48.9**	**17.4**	**19.6**	**25.9**	**39.7**	**59.3**	**49.5**	**78.9**	**26.2**	**61.3**
LaPose	Gaussian Noise	52.2	32.5	11.9	8.3	11.3	23.9	36.4	31.9	60.8	13.0	40.0
DMSR		45.2	30.1	7.2	10.6	17.1	19.3	34.0	28.0	52.9	19.3	39.2
Old-Net		24.8	8.7	0.2	0.6	2.2	3.3	11.4	9.0	32.3	4.1	17.2
Ours		**56.9**	**37.8**	**12.2**	**14.9**	**21.0**	**30.4**	**47.1**	**38.7**	**65.4**	**21.7**	**49.7**
LaPose	Defocus Blur	54.1	39.3	11.4	12.2	16.1	29.0	42.1	36.9	60.2	17.4	44.3
DMSR		53.2	37.9	8.8	10.8	15.4	24.5	37.2	36	61.8	17.4	41.1
Old-Net		22.6	5.9	0.1	0.2	1.3	2.1	8.8	7.6	30.0	3.1	15.1
Ours		**63.2**	**42.3**	**14.7**	**17.8**	**24.6**	**34.6**	**53.5**	**43.1**	**73.0**	**24.9**	**55.9**
LaPose	JPEG Compression	57.3	37.6	9.9	11.1	16.6	27.1	45.1	36.1	66.9	19.3	49.2
DMSR		51.3	32.4	8.1	11.3	14.8	21.5	32.3	31.7	60.3	16.5	34.9
Old-Net		35.9	12.4	0.3	0.8	2.7	5.0	15.8	13.5	44.8	4.1	20.1
Ours		**70.4**	**50.7**	**18.5**	**22.1**	**29.2**	**41.4**	**59.5**	**51.6**	**80.2**	**29.7**	**62.4**
LaPose	Elastic Transform	58.6	36.9	10.5	12.0	17.3	27.7	47.1	35.2	66.4	19.5	50.8
DMSR		49.0	30.9	6.4	11.2	17.4	19.5	35.0	27.9	57.5	20.5	40.8
Old-Net		25.3	5.6	0.2	0.4	2.6	2.2	11.9	6.2	34.8	4.3	17.6
Ours		**68.0**	**45.4**	**15.3**	**17.8**	**24.7**	**36.9**	**57.9**	**45.9**	**78.5**	**25.1**	**60.5**
LaPose	Frost	49.4	32.3	11.5	11.6	15.1	23.7	38.5	31.9	56.8	17.7	42.0
DMSR		39.5	25.1	4.8	10.1	16.9	16.4	29.0	23.4	46.6	20.0	34.7
Old-Net		23.8	6.6	0.2	0.2	0.9	2.2	8.9	7.4	31.6	1.6	12.8
Ours		**60.9**	**41.1**	**12.8**	**18.4**	**25.4**	**33.9**	**51.2**	**43.8**	**70.7**	**25.7**	**53.0**
LaPose	Fog	**66.9**	**46.5**	**17.2**	15.2	19.9	**35.0**	**53.9**	**46.0**	**75.3**	22.0	**57.0**
DMSR		53.1	35.2	7.5	14.0	21.7	24.1	41.1	32.1	61.7	26.3	47.9
Old-Net		17.8	2.4	0.1	0.1	0.3	1.1	4.9	4.0	27.6	0.7	7.3
Ours		58.9	40.6	11.1	**17.4**	**24.4**	33.4	50.2	42.6	67.5	**24.9**	51.8

Table 8. We show per corruption accuracy of all methods. Corruptions are applied to the full image, affecting both detection and pose estimation. Our method outperforms the baselines on a majority of the corruption types.

Method	Noise	$NIoU_{25}$	$NIoU_{50}$	$NIoU_{75}$	$5°0.2d$	$5°0.5d$	$10°0.2d$	$10°0.5d$	$0.2d$	$0.5d$	$5°$	$10°$
LaPose	Speckle Noise	**62.2**	**36.5**	**12.6**	7.0	10.1	**23.7**	40.2	**34.5**	**71.6**	11.9	45.3
DMSR		54.9	35.4	8.9	**10.9**	**18.0**	22.9	**40.5**	32.8	65.4	**21.1**	**46.0**
Old-Net		30.9	9.1	0.2	0.5	2.0	4.1	13.4	10.6	41.6	3.6	19.1
LaPose	Gaussian Noise	**62.6**	**34.3**	**12.0**	7.6	10.8	**22.5**	37.2	**33.8**	**72.4**	12.3	41.5
DMSR		55.3	**34.3**	7.6	**11.8**	**19.0**	22.3	**39.5**	32.0	65.3	**21.4**	**44.9**
Old-Net		33.3	10.4	0.2	0.7	2.2	4.6	13.6	11.6	43.5	3.5	19.0
LaPose	Gaussian Blur	**61.8**	**41.5**	**15.3**	13.3	19.0	**31.4**	**49.9**	**40.5**	**71.9**	22.1	**54.5**
DMSR		57.9	39.8	10.1	**14.4**	**22.2**	26.7	45.1	37.5	67.7	**26.1**	51.4
Old-Net		30.6	7.6	0.3	0.5	2.9	3.5	13.7	9.9	41.5	4.6	18.9
LaPose	Defocus Blur	**60.8**	**38.8**	**11.0**	**11.3**	**16.7**	**27.3**	**48.0**	**36.1**	**71.3**	**18.8**	**51.8**
DMSR		56.3	37.9	8.7	10.6	15.7	24.0	38.9	35.8	66.2	18.5	44.0
Old-Net		25.9	6.2	0.1	0.2	1.3	1.9	8.6	9.0	36.6	3.4	15.7
LaPose	JPEG Compression	**58.3**	**34.2**	**8.9**	9.9	**16.3**	**24.6**	**46.3**	**33.3**	**70.5**	**19.1**	**51.3**
DMSR		50.6	29.9	6.4	**10.6**	14.9	20.4	33.5	29.1	61.9	16.7	36.9
Old-Net		35.7	11.5	0.4	0.9	2.9	5.0	15.8	12.9	46.5	4.3	20.4
LaPose	Elastic Transform	**69.1**	**44.5**	**13.4**	**13.6**	18.9	**33.8**	**54.8**	**42.7**	**77.8**	21.2	**58.2**
DMSR		56.8	37.6	8.7	13.4	**20.2**	25.0	42.1	35.0	66.5	**23.3**	48.2
Old-Net		33.2	9.5	0.3	0.7	2.6	4.1	14.4	11.1	44.2	3.8	19.5
LaPose	Frost	**68.0**	**41.6**	**13.4**	11.8	16.8	23.3	**49.7**	**40.0**	**76.9**	18.7	**52.9**
DMSR		55.6	35.3	8.0	**13.3**	**21.8**	**23.5**	40.6	33.5	66.2	**24.9**	45.6
Old-Net		35.6	12.0	0.3	0.4	1.2	4.8	13.1	14.0	47.6	1.9	17.0
LaPose	Fog	**70.3**	**47.1**	**16.2**	**15.3**	20.9	**36.2**	**57.4**	**45.8**	**78.8**	23.1	**60.4**
DMSR		55.0	34.6	7.5	14.2	**23.7**	23.1	43.8	31.1	65.1	**28.6**	50.9
Old-Net		17.3	2.3	0.1	0.1	0.3	1.1	4.8	3.9	27.7	0.6	7.3

Table 9. Ablation study of robustness with corrupted ROIs.

H. Per Category Results

Fig. 8 shows category-level pose estimation results using our method and each baseline which has public code [15, 55, 62]. Notably, our approach consistently improves the mean performance. Furthermore, our method outperforms others in the challenging non-symmetric camera and laptop categories. This result highlights the effectiveness of our object representation and the single-stage modeling strategy.

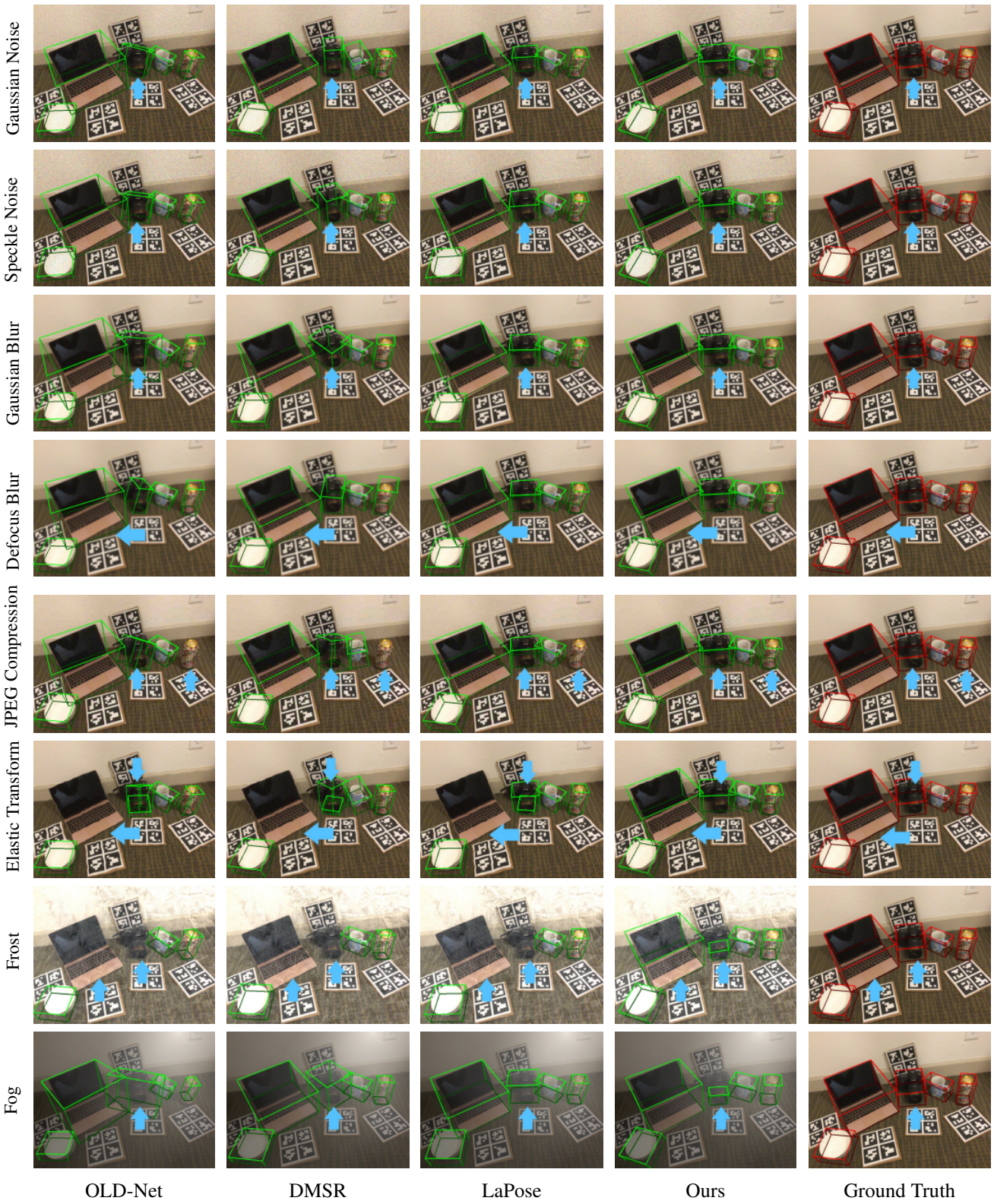

Figure 6. Qualitative comparison on Corrupted NOCS-REAL275[54]. We compare our model with all baselines (first to third columns) and with ground truth (last column) across 8 types of corruption.

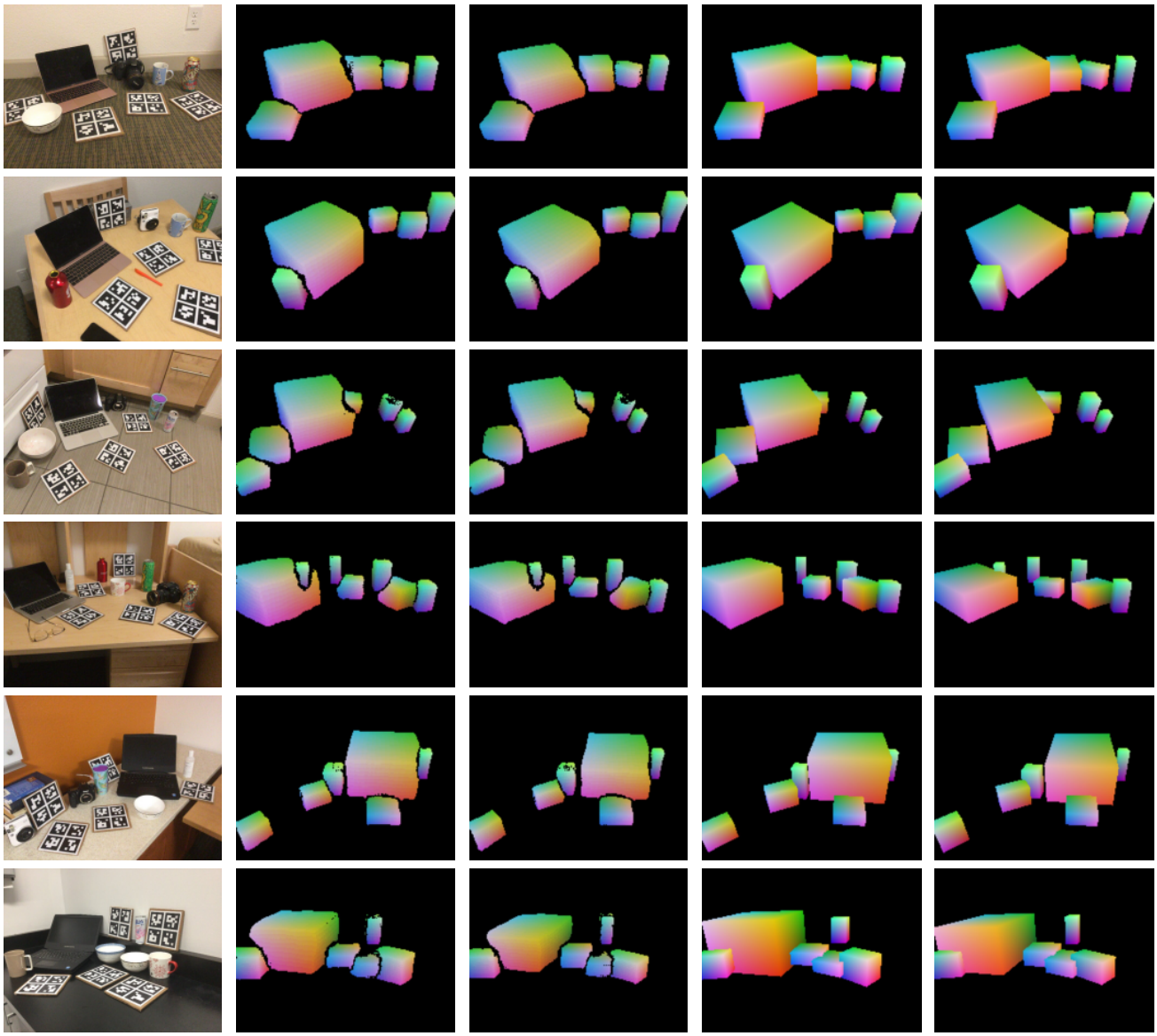

Figure 7. Visualization of the our dense matching results. We estimate 2D-3D correspondences between image features and our object prototypes with mean scales and instance-level scales. We remove low noisy correspondences using our foreground modeling strategy and confidence scores. Our method produces reliable and mostly noise-free correspondences in object regions.

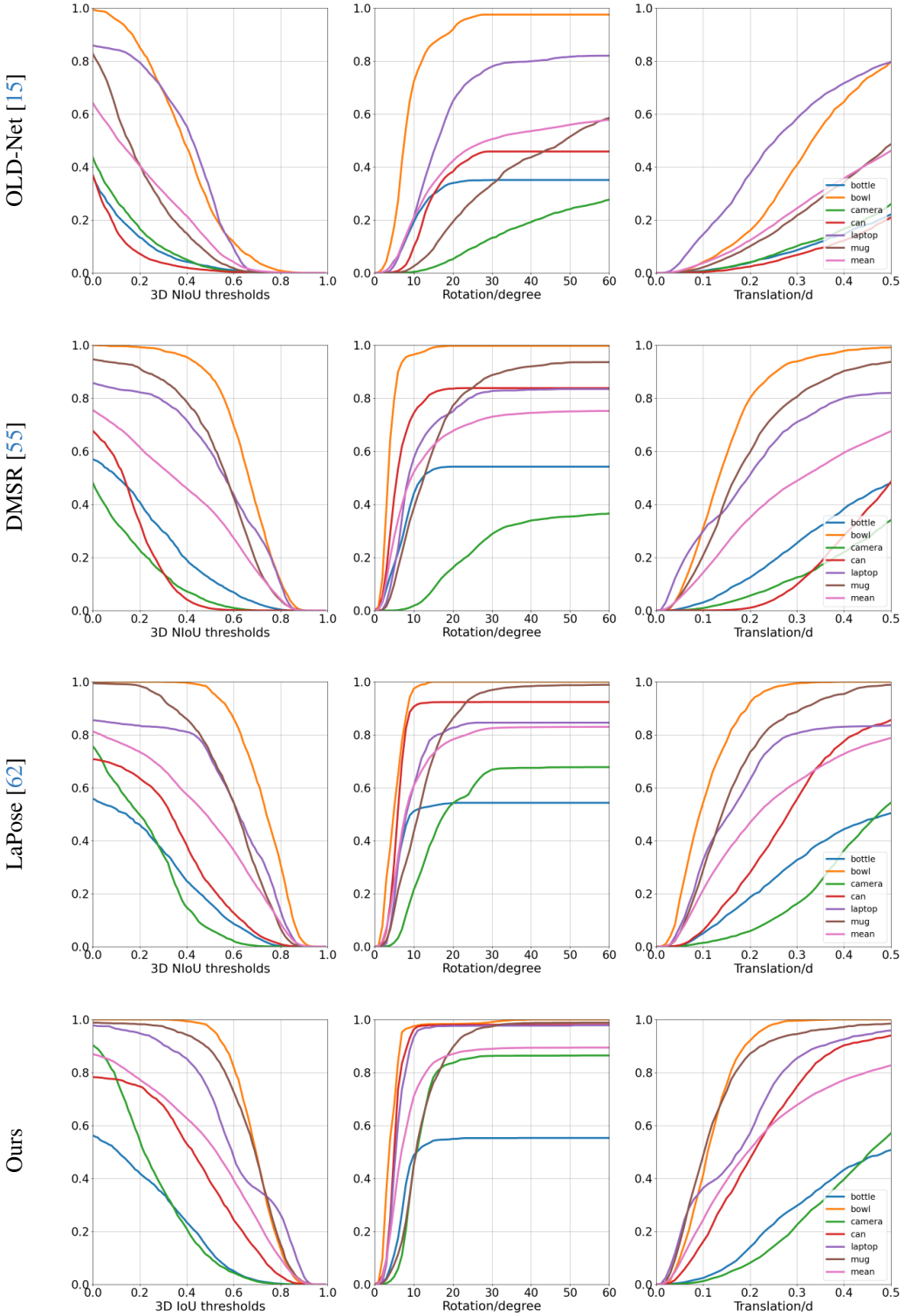

Figure 8. We show mean Average Precision (mAP) on REAL275[54] using scale-agnostic metrics. We compare our model with all baselines that have public code. Noticeably, our method has significantly increased rotation accuracy on the challenging non-symmetric categories camera and laptop.