UNIPOSE9D: UNIVERSAL CATEGORY-AGNOSTIC OBJECT POSE ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Object pose estimation is a fundamental problem in 3D vision. Although recent state-of-the-art approaches achieve strong performance, generalization to novel categories and unseen scenes remains challenging. We propose **UniPose9D**, a unified model for category-agnostic 9D object pose estimation: given an instance mask/ROI and either an RGB-D observation or an RGB image with predicted depth, the model estimates rotation, translation, and metric size without category labels, CAD models, mean-shape priors, or reference views. Specifically, UniPose9D samples point pairs from the observed object geometry and uses DINOv2 and PointNet features to predict NOCS coordinates for each pair. To improve accuracy, we introduce a point-pair-based RANSAC *N-hop* Kabsch–Umeyama algorithm with an adaptive threshold. We further employ flow matching to address symmetric ambiguities and construct a large-scale training set by curating and aligning pose annotations from existing public datasets. Experiments across eight datasets show that a single unified model achieves competitive performance on standard benchmarks while generalizing to unseen objects, unseen categories, and in-the-wild scenarios.

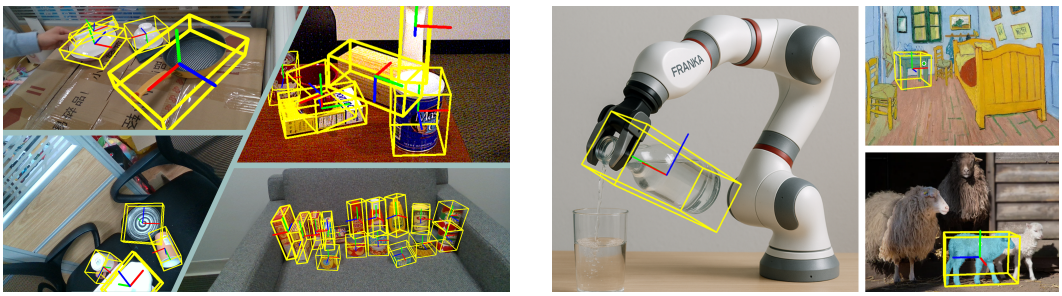

Figure 1: **One model for category-agnostic 9D pose estimation.** Given an instance mask/ROI and measured depth (left four examples) or predicted depth (right three), UniPose9D estimates rotation, translation, and metric box size. No category prompt, target mesh, mean shape, or reference view is supplied at inference. Yellow cuboids show predicted boxes; red, green, and blue axes show orientation.

1 INTRODUCTION

Object pose estimation is a fundamental problem in computer vision with wide impact on robotic manipulation (Wong et al., 2017; Wen et al., 2022; Zeng et al., 2017; Du et al., 2021) and augmented reality (Marder-Eppstein, 2016; Runz et al., 2018; Marchand et al., 2015). Traditional instance-level methods (Haugaard & Buch, 2022; Xiang et al., 2018; Wang et al., 2021) are tailored to specific objects and typically rely on textured CAD models to synthesize training data, limiting their generalization to unseen instances at test time. Category-level approaches (Wang et al., 2019; Zheng et al., 2023; Lin et al., 2022; 2021b; Chen & Dou, 2021) remove the need for instance-wise training and meshes, yet they are usually trained and evaluated within predefined categories, and some additionally rely on category-specific models or mean-shape priors.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To mitigate these limitations, recent works aim to estimate the pose of arbitrary objects at inference time by providing a reference 3D textured model or a set of reference images. While this setting enables generalization to unseen objects, it still requires scanning the entire object or carefully selecting reference viewpoints, which is impractical under occlusion and in robotic scenarios where the camera cannot freely move around the target.

This motivates us to ask: do we truly need instance-level meshes/reference views or category labels (e.g., a category name or template)? We call a model *category-agnostic* if, at inference time, it estimates 9D pose for a detected object without receiving its category name, a category template, a mean shape, an instance mesh, or reference views. As discussed in Orient Anything (Wang et al., 2025b), many objects have a naturally defined canonical orientation aligned with human perception, up to rotational symmetries. Consequently, a single observation can provide appearance and geometry cues to predict object pose. We learn these canonical conventions from aligned pose annotations while sharing the same prediction head across categories.

Therefore, we present **UniPose9D**, a universal model that predicts an object's 9D pose, including rotation, translation, and metric bounding box size, from the input described above. Our method goes beyond Orient Anything, which predicts rotation only.

Inspired by CPPF (You et al., 2022b), we avoid per-pixel NOCS (Wang et al., 2019) regression. Instead, we sample point pairs from the observed object point cloud, extract trainable geometric features with PointNet (Qi et al., 2017) and visual features with DINOv2 (Oquab et al., 2023), and predict the NOCS coordinates for each pair. Pairwise sampling provides a *quadratic* space of candidate pairs, from which we sample a fixed budget to improve the robustness of the subsequent RANSAC-based Kabsch–Umeyama (Lawrence et al., 2019) estimation. For metric-scale recovery, rather than deriving scale directly from Kabsch–Umeyama, we first calibrate it using the ratio between predicted NOCS pair differences and input point-pair differences. We further introduce an *N-hop* Kabsch–Umeyama algorithm that iteratively refines the alignment under a scale-aware inlier threshold. To handle object symmetries, we replace explicit symmetry enumeration with flow matching to model multimodal NOCS distributions.

UniPose9D is trained as a single model on a mixture of public pose datasets and attains strong performance on standard benchmarks while generalizing to unseen categories and in-the-wild scenes. We study whether one category-agnostic model can provide a reusable 9D pose estimator across datasets without dataset-specific tuning. Additional evaluation on unseen-category subsets of Omni6D and HANDAL extends our transfer experiments beyond YCB-Video and HOPE. Our model also accepts RGB input by using predicted metric depth from an off-the-shelf depth estimation model. Our contributions are summarized as follows:

- We introduce **UniPose9D**, a unified framework that predicts rotation, translation, and metric scale from a single masked RGB-D observation or RGB with predicted depth, without category labels, CAD models, mean-shape priors, or reference images at inference.
- We propose a point-pair-driven NOCS prediction pipeline, pairwise metric-scale calibration, and an *N-hop* Kabsch–Umeyama scheme with an adaptive inlier threshold, improving robustness and accuracy.
- Experiments across eight datasets show that UniPose9D achieves competitive performance on standard benchmarks and generalizes to unseen categories and in-the-wild images.

2 Related Work

2.1 Instance-Level Pose Estimation

Instance-level methods assume access to a CAD model for each target object. Classical pipelines such as PPF (Drost et al., 2010) use depth geometry for matching. Modern systems learn pose from RGB or RGB-D: PoseCNN (Xiang et al., 2018) learns 6D pose, CosyPose (Labbé et al., 2020) integrates coarse-to-fine refinement, SurfEmb (Haugaard & Buch, 2022) learns dense surface embeddings for 2D–3D matching, and GDR-Net (Wang et al., 2021) uses geometry-aware regression. These methods are strong when instance models are available, but remain tied to target-specific CAD supervision.

2.2 Category-Level and Category-Agnostic Pose Estimation

Category-level pose estimation removes the requirement of instance-specific meshes and aims to predict 9D pose for unseen objects within a category. NOCS (Wang et al., 2019) introduces a shared canonical space and recovers pose via alignment between predicted coordinates and observed geometry. Many works then extend the paradigm with stronger priors or learning objectives: SGPA (Chen & Dou, 2021) adapts structure-guided priors, SAR-Net (Lin et al., 2022) performs shape alignment to handle symmetries and scale, and HS-Pose (Zheng et al., 2023) designs an operator for extracting hybrid-scope features on point clouds. Alternative formulations target prior-free generalization. VI-Net (Lin et al., 2023) decouples viewpoint and in-plane rotations to stabilize orientation prediction. IST-Net (Liu et al., 2023) learns an implicit transformation that aligns camera-space features to a canonical frame without category priors. CPPF (You et al., 2022b) and CPPF++ (You et al., 2022a) use point-pair voting for category-level pose, but their reported models are still trained and evaluated in category-specific settings. Diffusion and generative modeling have also been explored, for example, GenPose (Zhang et al., 2023) and GenPose++ (Zhang et al., 2024a), to model uncertainty and improve robustness.

SecondPose (Chen et al., 2024) combines semantic and geometric features, while CleanPose (Lin et al., 2025) improves category-level estimation through causal learning and knowledge distillation. UniPose9D shares the broad correspondence-and-alignment philosophy of these methods. It combines point-pair NOCS prediction with pairwise metric-scale calibration and an adaptive N-hop Kabsch–Umeyama solver, using one model trained across heterogeneous public datasets without category labels or category-specific heads at inference.

2.3 Reference-Based and Open-Vocabulary Pose Estimation

Reference-based methods avoid category-specific training by using images or videos of the target object at inference. Gen6D (Liu et al., 2022) estimates pose from calibrated reference images, while OnePose (Sun et al., 2022) and OnePose++ (He et al., 2022a) reconstruct sparse 3D geometry from multi-view inputs using structure-from-motion, then solve pose via 2D–3D matching. FS6D (He et al., 2022b) and FoundationPose (Wen et al., 2024) extend the idea to RGB-D and achieve strong results when reliable depth and camera poses are available. To reduce data acquisition requirements, recent work explores single-reference or partial-overlap settings. NOPE (Nguyen et al., 2024a) estimates relative orientation from a single anchor image. GigaPose (Nguyen et al., 2024b) matches rendered templates of target objects, while Any6D (Lee et al., 2025) uses image-to-3D generation to obtain geometric cues from a single reference. These methods introduce a target-specific reference image, video, reconstruction, or 3D model at inference. UniPose9D targets the complementary setting in which only the current masked RGB-D observation, or RGB with predicted depth, is available.

Open-vocabulary pose estimation Oryon and Horyon (Corsetti et al., 2024b;a) estimate relative 6D pose from two RGB-D observations and object text, while ConceptPose (Kuang et al., 2026) uses category-derived concepts for cross-view matching. OV9D (Cai et al., 2024) estimates absolute pose and size from a localized RGB-D observation and category description. Pov9D (Wang & Wang, 2025) and ProM-Pose (Li et al., 2026) use language-conditioned shape priors, with ProM-Pose also using temporal observations. Orient Anything V2 (Wang et al., 2025c) extends orientation estimation to relative rotation and symmetry. UniPose9D estimates absolute 9D pose from one localized observation without a category prompt, target reference, or supplied shape prior.

3 Method

Given a masked RGB-D observation or RGB with predicted metric depth and camera intrinsics, UniPose9D back-projects the masked depth to an object point cloud. It predicts point-pair NOCS coordinates and aligns them with observed 3D points to recover rotation $\mathbf{R} \in SO(3)$, translation $\mathbf{t} \in \mathbb{R}^3$, and metric box size $\mathbf{s} \in \mathbb{R}^3$.

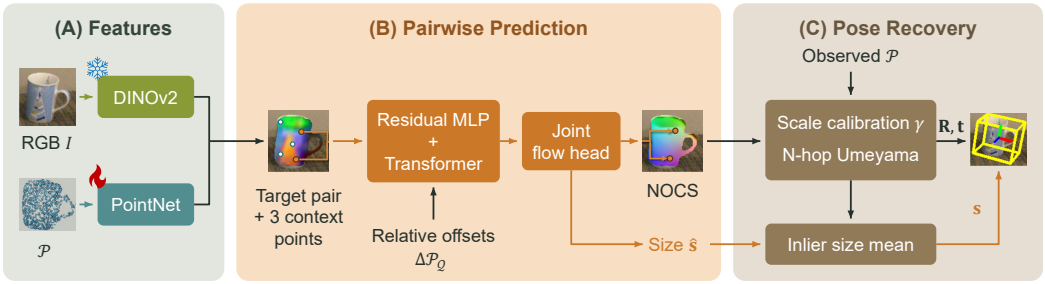

Figure 2: **Pipeline of UniPose9D.** (A) DINOv2 and PointNet extract visual and geometric descriptors. (B) Each target pair is augmented with context points; an MLP and transformer encode their descriptors and relative coordinates. A joint flow head predicts the target pair's NOCS coordinates $\hat{\mathbf{x}}_{i_1}, \hat{\mathbf{x}}_{i_2}$ and metric box size $\hat{\mathbf{s}}$. (C) We calibrate scale from pair differences and run N-hop Umeyama with an adaptive threshold to recover $(\mathbf{R}, \mathbf{t})$; averaging box sizes over inlier pairs gives $\mathbf{s}$.

3.1 Preliminary: NOCS

NOCS canonicalizes each object inside a unit cube by uniformly scaling its CAD model so the longest side of its tight bounding box has length 1, centering it, and consistently aligning objects by center and orientation within each category. These canonical annotations provide training supervision; no category label or target CAD model is needed at inference. Given predicted NOCS coordinates, rotation and translation can be estimated with Kabsch–Umeyama. A NOCS coordinate $\mathbf{x}$ relates to its corresponding 3D point $\mathbf{p}$:

$$\mathbf{p} = \gamma \mathbf{R}\mathbf{x} + \mathbf{t}, \tag{1}$$

where γ is the scalar normalization equal to the longest side of the metric 3D bounding box, and the final anisotropic metric box size is denoted by $\mathbf{s} \in \mathbb{R}^3$.

3.2 Overview

Figure 2 shows the three parts of UniPose9D: visual and geometric feature extraction, point-pair NOCS and box-size prediction, and N-hop Kabsch–Umeyama pose recovery with an adaptive inlier threshold.

3.3 Feature Extraction

For visual feature extraction, we use DINOv2-S/14 (Oquab et al., 2023) enhanced by 3DCorrEnhance (You et al., 2024a), which provides semantic correspondences across different objects and backgrounds. Given an ROI crop and the image location of each observed point, we bilinearly sample its visual descriptor. For geometric features, a lightweight, multi-radius PointNet aggregates local neighborhoods using residual MLPs and max pooling. We project both modalities to a common width d and concatenate them as $\mathbf{f}_i = [\mathbf{W}_v \mathbf{f}_i^{\text{img}} \oplus \mathbf{W}_g \mathbf{f}_i^{\text{geo}}] \in \mathbb{R}^{2d}$. Full feature-extraction equations and architecture details are given in Appendix B.1.

3.4 Point-Pair Sampling and NOCS Prediction

Point pair construction We uniformly sample target pairs $\mathcal{T}=(i_1, i_2)$ and augment each with three context points, forming a five-point tuple $\mathcal{Q}_\mathcal{T}$. We combine all ten pairwise differences $\Delta\mathcal{P}_\mathcal{Q}$ in metric coordinates with projected visual and geometric descriptors:

$$\mathbf{f}_\mathcal{T} = \Delta\mathcal{P}_\mathcal{Q} \oplus \mathbf{W}_p \left(\bigoplus_{j \in \mathcal{Q}_\mathcal{T}} \mathbf{f}_j \right). \tag{2}$$

Five residual MLP blocks followed by a four-layer transformer produce contextual pair features $\mathbf{h}_\mathcal{T}$, with attention across the sampled tuples. Architecture dimensions are given in Appendix B.

Why point pairs Switching from pixel-based NOCS to point-pair-based NOCS offers three advantages: (1) *Quadratic candidate space.* From N sparse samples we obtain $O(N^2)$ possible point pairs, from which we sample T pairs for NOCS prediction and RANSAC. These pairs provide different contexts rather than independent new observations; (2) *Relative invariance.* Pair differences remove common translation biases and make scale calibration easier (Section 3.5); (3) *Occlusion robustness.* Pairwise correspondences allow RANSAC to reject inconsistent hypotheses while retaining geometrically consistent pairs under partial visibility.

Joint flow-matching head The head jointly models two NOCS points and metric box size, with target $\mathbf{z}_0 = [2\mathbf{x}^*_{i_1} \oplus 2\mathbf{x}^*_{i_2} \oplus \log \mathbf{s}^*] \in \mathbb{R}^9$. A time-conditioned Gaussian-mixture flow (Chen et al., 2025) models multimodal correspondence distributions arising from object symmetries. All nine coordinates follow the linear-noise marginals

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\boldsymbol{\epsilon}, \quad t \in [0,1), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}_9). \tag{3}$$

Training samples a bridged pair $(\mathbf{z}_{t_{\text{low}}}, \mathbf{z}_t)$ with $t_{\text{low}} < t$. Conditioned on the contextual feature, noisy state, and learned time embedding $\phi(t)$, the head predicts nine-dimensional flow means, a shared standard deviation, and mixture weights:

$$\{\mathbf{u}_g, \sigma_u, \pi_g\}_{g=1}^{G} = \mathrm{GM}_\theta(\mathbf{h}_\mathcal{T}, \mathbf{z}_t, \phi(t)). \tag{4}$$

These outputs are mapped analytically to reverse-transition means $\mathbf{m}_g$ and variance v, as detailed in Appendix B.3. We minimize the joint negative log-likelihood, averaged over images and tuples:

$$\mathcal{L}_{\text{joint}} = -\log \sum_{g=1}^{G} \pi_g \mathcal{N}(\mathbf{z}_{t_{\text{low}}} \mid \mathbf{m}_g, v\mathbf{I}_9). \tag{5}$$

This objective supervises both NOCS and log-size. At test time, an ODE-based sampler produces the joint state; dividing its first six coordinates by two gives the predicted NOCS pair.

Metric box-size output Exponentiating the final three sampled coordinates gives the metric box size $\hat{\mathbf{s}} \in \mathbb{R}^3$, learned through the same joint likelihood. This vector scale is distinct from the scalar NOCS normalization factor γ used during correspondence alignment.

3.5 9D Pose Recovery via N-hop Kabsch–Umeyama

Scale calibration Let $\mathcal{C}=\{\mathcal{C}_\mathcal{T}\}_\mathcal{T}$ collect the predicted NOCS–metric correspondences, with $\mathcal{C}_\mathcal{T}=\{(\hat{\mathbf{x}}_{i_1}, \mathbf{p}_{i_1}), (\hat{\mathbf{x}}_{i_2}, \mathbf{p}_{i_2})\}$ for $\mathcal{T}=(i_1, i_2)$. Point-pair sampling lets us decouple scalar normalization from Kabsch–Umeyama. For each pair, we calibrate γ by

$$\gamma_\mathcal{T} = \frac{\|\mathbf{p}_{i_2} - \mathbf{p}_{i_1}\|_2}{\|\hat{\mathbf{x}}_{i_2} - \hat{\mathbf{x}}_{i_1}\|_2}, \tag{6}$$

and replace the original coordinates with scaled NOCS $\hat{\mathbf{x}}^s = \gamma_\mathcal{T}\hat{\mathbf{x}}$, so that $\mathbf{p}_i = \mathbf{R}\hat{\mathbf{x}}^s_i + \mathbf{t}$. We then apply Kabsch–Umeyama to estimate $(\mathbf{R}, \mathbf{t})$ without estimating scale.

Finally, we form a flattened correspondence list by taking both points of every pair and running RANSAC to solve the optimal $SE(3)$ transformation between the scaled NOCS and metric spaces.

Adaptive threshold and N-hop refinement We iterate the above RANSAC process for $H=3$ hops. Starting from an initial inlier threshold τ_0, at each hop we:

1. Run RANSAC–Umeyama at τ_h to obtain $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ and point-level inliers.
2. If more than 10% of the individual correspondences are inliers, update the threshold using the largest dimension of the mean box-size vector over selected pairs:

$$\tau_{h+1} = 0.05 \cdot \left\| \frac{1}{|\mathcal{I}|} \sum_{\mathcal{T}_i \in \mathcal{I}} \hat{\mathbf{s}}_{\mathcal{T}_i} \right\|_\infty, \tag{7}$$

where $\mathcal{I}$ contains pairs with at least one inlier endpoint; otherwise, set $\tau_{h+1} = 1.5 \cdot \tau_h$.

Finally, we average box sizes $\hat{\mathbf{s}}$ over the retained pairs, counting each pair once. Pseudocode is provided in the supplementary material.

3.6 Implementation Details

We train one model on the PACE, Omni6DPose, NOCS, and HouseCat6D training splits with balanced sampling and minimize $\mathcal{L}_{\text{joint}}$ using AdamW. At inference, we sample $N = 1{,}024$ points and $T = 10{,}000$ pairs by default, with 16 ODE steps and 2 substeps per step. Full architecture and optimization settings are in Appendix B; Section 4.5 examines the pair budget.

4 Experiments

We evaluate a single unified UniPose9D model on the held-out test sets of PACE (You et al., 2024b), Omni6DPose (Zhang et al., 2024a), NOCS REAL275 (Wang et al., 2019), and HouseCat6D (Jung et al., 2024); the training splits of these four datasets contribute to our training mixture. We then test cross-domain generalization on YCB-Video (Xiang et al., 2018) and HOPE (Tyree et al., 2022), evaluate unseen categories in Omni6D (Zhang et al., 2024b) and HANDAL (Guo et al., 2023), and report ablations.

4.1 Experimental Setup

We evaluate without test-dataset fine-tuning, using each dataset's standard protocol. UniPose9D takes an instance mask/ROI and back-projects measured depth within the mask, or predicted metric depth for RGB input. Appendix C.2 summarizes the benchmark mask/ROI conventions.

Baselines and metrics We tag methods as **Cat** (category-level specialist, trained for a fixed taxonomy with category-specific inputs or branches) or **Gen** (generalizable, requiring no category labels, mean shapes, CAD models, or reference views at inference). Specialist results retain their native training protocols; UniPose9D uses one shared model across all four benchmarks. We also retrain baselines on our four-dataset mixture in Appendix A.

We retain each benchmark's native metrics: PACE reports 3D-box AP at $\text{IoU}_{25/50}$ and AP averaged over rotation/translation thresholds; Omni6DPose reports AUC for box overlap and VUS for joint pose thresholds; REAL275 and HouseCat6D report pose mAP. For YCB-Video and HOPE, we report median rotation error and accuracy within 5° or 2 cm. The RGB setting uses metric depth predicted by MoGe-2 (Wang et al., 2025a). Baseline scores are taken from the cited papers or reproduced under the corresponding protocol; metric definitions and additional comparisons are provided in the appendix.

4.2 Pose Estimation In-Domain

PACE Table 1 summarizes results on the PACE test split. UniPose9D attains the best IoU_{50} (**40.1**) by a wide margin and leads every reported AP metric, with especially large gains over GenPose++ on the joint thresholds AP@0:20°,0:5 cm (**+20.7**) and AP@0:60°,0:15 cm (**+22.1**), as well as the pure-angle APs AP@0:20° (**+18.6**) and AP@0:60° (**+17.7**). Qualitative comparisons against NOCS, HS-Pose, CPPF++, and GenPose++ are shown in Figure S1.

Omni6DPose Table 2 shows that UniPose9D ranks first across AUC and VUS, surpassing GenPose++ on all three AUC@IoU thresholds while leading or tying on every VUS metric. The improvement is larger for 3D-box overlap (AUC@IoU_{50}: 19.1 to 23.0) than for the joint pose metrics (VUS@5°5 cm: 15.1 to 15.8).

NOCS REAL275 Table 3 compares UniPose9D with category-level pose estimation methods. UniPose9D attains a competitive IoU_{75} of 48.1, within 1.6 points of SecondPose, and comparable mAP at looser angular thresholds. The recent specialist CleanPose (Lin et al., 2025) achieves a higher published 5°5 cm mAP (67.6 versus 60.2). Our model uses the same category-agnostic weights across all four benchmarks without mean-shape priors at inference.

Table 1: **PACE benchmark.** 3D-box AP and threshold-averaged pose AP (%). Cat: category-level specialist; Gen: generalizable. Bold marks the best score.

Method	Type	IoU_{25} ↑	IoU_{50} ↑	AP ↑					
				0:20°	0:60°	0:5 cm	0:15 cm	0:20°, 0:5 cm	0:60°, 0:15 cm
NOCS (Wang et al., 2019)	Cat	0.2	0.0	1.2	4.3	17.0	33.5	1.1	4.2
HS-Pose (Zheng et al., 2023)	Cat	36.6	2.7	5.3	8.6	48.7	81.0	3.9	8.1
SGPA (Chen & Dou, 2021)	Cat	2.6	1.2	5.6	11.4	16.1	50.7	3.3	10.1
DualPoseNet (Lin et al., 2021b)	Cat	24.2	0.1	5.5	8.1	18.6	63.4	1.8	6.5
SAR-Net (Lin et al., 2022)	Cat	22.8	0.2	5.3	8.8	48.6	79.8	3.7	8.2
CPPF++ (You et al., 2022a)	Cat	44.5	4.4	15.2	27.3	35.3	74.0	9.9	24.9
GenPose++ (Zhang et al., 2024a)	Gen	52.5	18.6	21.8	39.7	52.3	79.0	15.7	33.9
Ours	Gen	**61.7**	**40.1**	**40.4**	**57.4**	**65.2**	**87.7**	**36.4**	**56.0**

Table 2: **Omni6DPose benchmark.** AUC for box overlap and VUS for joint pose thresholds (%). The same UniPose9D model is used across all four in-domain datasets.

Method	Type	AUC ↑			VUS ↑			
		IoU_{25}	IoU_{50}	IoU_{75}	5°2 cm	5°5 cm	10°2 cm	10°5 cm
NOCS (Wang et al., 2019)	Cat	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SGPA (Chen & Dou, 2021)	Cat	10.5	2.0	0.0	4.3	6.7	9.3	15.0
IST-Net (Liu et al., 2023)	Cat	28.7	10.6	0.5	2.0	3.4	5.3	8.8
HS-Pose (Zheng et al., 2023)	Cat	31.6	13.6	1.1	3.5	5.3	8.4	12.7
GenPose++ (Zhang et al., 2024a)	Gen	39.0	19.1	2.0	10.0	15.1	**19.5**	29.4
Ours	Gen	**43.1**	**23.0**	**2.9**	**10.5**	**15.8**	**19.5**	**29.7**

Table 3: **NOCS REAL275 benchmark.** IoU_{75} and pose mAP (%). Dashes denote unreported values; bold marks the best score.

Method	Type	IoU_{75}	5°2 cm	5°5 cm	10°2 cm	10°5 cm
FS-Net (Chen et al., 2021)	Cat	–	–	28.2	–	60.8
DualPoseNet (Lin et al., 2021b)	Cat	30.8	29.3	35.9	50.0	66.8
GPV-Pose (Di et al., 2022)	Cat	–	32.0	42.9	–	71.6
SS-ConvNet (Lin et al., 2021a)	Cat	–	36.6	43.4	52.6	63.5
HS-Pose (Zheng et al., 2023)	Cat	–	41.3	50.4	72.3	82.7
IST-Net (Liu et al., 2023)	Cat	47.5	53.4	63.2	72.1	80.5
VI-Net (Lin et al., 2023)	Cat	48.3	50.0	57.6	70.8	82.1
SecondPose (Chen et al., 2024)	Cat	49.7	56.2	63.6	74.7	86.0
CleanPose (Lin et al., 2025)	Cat	**62.7**	**61.7**	**67.6**	**78.3**	**86.3**
Ours	Gen	48.1	45.4	60.2	63.6	82.9

HouseCat6D Table 4 shows strong performance in cluttered household scenes: UniPose9D improves 5°5 cm mAP from SecondPose's 13.4 to 19.0, while CleanPose reports a higher score of 24.1. These native-protocol comparisons are complemented by retraining baselines on our four-dataset mixture in Table S1.

Table 4: **HouseCat6D benchmark.** 3D-box AP and pose mAP (%) under the native protocol. Bold marks the best score.

Method	Type	IoU_{25}	IoU_{50}	5°2 cm	5°5 cm	10°2 cm	10°5 cm
FS-Net (Chen et al., 2021)	Cat	74.9	48.0	3.3	4.2	17.1	21.6
GPV-Pose (Di et al., 2022)	Cat	74.9	50.7	3.5	4.6	17.8	22.7
VI-Net (Lin et al., 2023)	Cat	80.7	56.4	8.4	10.3	20.5	29.1
AG-Pose (Lin et al., 2024)	Cat	81.8	62.5	11.5	12.0	32.7	35.8
SecondPose (Chen et al., 2024)	Cat	83.7	66.1	11.0	13.4	25.3	35.7
CleanPose (Lin et al., 2025)	Cat	**89.2**	**79.8**	**22.4**	**24.1**	**51.6**	**56.5**
Ours	Gen	86.4	70.6	16.9	19.0	37.5	43.2

4.3 Generalization to Unseen Instances

To test generalization to unseen instances, we evaluate on the full YCB-Video and HOPE test sets. Both contain seen and unseen categories, while all object instances and scenes are absent from our pose-training mixture. We compare with Orient Anything (rotation-only) and GenPose++, calibrating rotations to a common canonical convention. In Table 5, UniPose9D improves RGB-D accuracy over GenPose++ by **28.0/31.3** percentage points at 5° and **45.4/35.8** points at 2 cm on YCB-Video/HOPE, respectively, with median rotation errors around 4° on both. RGB rotation accuracy also improves. Qualitative comparisons appear in the supplementary material.

With depth predicted by MoGe-2 (Wang et al., 2025a), metric translation remains difficult: Acc@2 cm falls from 91.9% to 3.4% on YCB-Video and from 91.5% to 1.5% on HOPE when moving from RGB-D to RGB. On HOPE with RGB input, GenPose++ has higher translation accuracy (2.3%). Rotation is less affected: our median error changes from 4.0° to 8.9° on YCB-Video and from 4.2° to 7.6° on HOPE. Predicted-depth scale errors directly affect metric translation.

Table 5: **YCB-Video and HOPE cross-domain evaluation.** Median rotation error (Med°), Acc@5°, and Acc@2 cm (%). OriAny predicts rotation only.

Setting	Method	Type	YCB-Video			HOPE		
			Med°↓	5°↑	2 cm↑	Med°↓	5°↑	2 cm↑
RGB	OriAny (Wang et al., 2025b)	Gen	91.4	1.4	–	77.4	1.1	–
	GenPose++ (Zhang et al., 2024a)	Gen	15.5	14.4	1.5	63.6	19.8	**2.3**
	Ours	Gen	**8.9**	**27.9**	**3.4**	**7.6**	**33.4**	1.5
RGB-D	GenPose++ (Zhang et al., 2024a)	Gen	10.3	33.9	46.5	53.4	25.7	55.7
	Ours	Gen	**4.0**	**61.9**	**91.9**	**4.2**	**57.0**	**91.5**

4.4 Generalization to Unseen Categories

YCB-Video and HOPE We next evaluate categories absent from our four-dataset pose-training mixture. YCB-Video contains five such categories (3,543 examples), and HOPE contains two (225 examples), with joint 5°5 cm accuracies of 40.8% and 64.9%, respectively. Category lists and complete subset results are in Appendix A.2.

Omni6D Omni6D (Zhang et al., 2024b) is distinct from the Omni6DPose training dataset. The baselines in Table 6 train on Omni6D_train and test on its held-out categories. UniPose9D transfers zero-shot to the same unseen split without Omni6D training, achieving 50.71% accuracy at IoU_{50} and 26.42% at 5°5 cm on 9,530 examples from 13 categories absent from our pose-training mixture.

Table 6: **Omni6D unseen-split evaluation.** Scores (%); baselines train on Omni6D_train, while ours is zero-shot. Baseline results are from Zhang et al. (2024b); metric definitions and evaluation details are in Appendix A.2.

Method	IoU_{50}	Joint pose				Rotation		Translation	
		5°2 cm	5°5 cm	10°2 cm	10°5 cm	5°	10°	2 cm	5 cm
SGPA	7.05	0.07	0.28	0.19	0.82	0.53	1.69	3.87	28.28
DualPoseNet	36.85	3.24	3.37	8.04	8.51	3.39	8.64	78.00	98.60
HS-Pose	36.75	1.54	1.66	4.67	5.16	1.75	5.38	79.95	98.27
Ours (zero-shot)	50.71	15.61	26.42	20.26	32.26	29.77	35.89	48.96	86.20

HANDAL We further evaluate five unseen categories in HANDAL (Guo et al., 2023): ladle, power drill, spatula, strainer, and whisk. Table 7 compares the same 25 examples for all methods. UniPose9D obtains a median rotation error of 3.3°, compared with 33.9° for GenPose++ and 68.1° for Orient Anything. Because Orient Anything predicts only rotation, it receives ground-truth translation in this comparison. The sample is small and serves as a diagnostic rather than a full HANDAL benchmark.

Table 7: **HANDAL unseen-category evaluation.** Joint-threshold scores (%) and median rotation error on 25 examples from five categories absent from pose training. Orient Anything receives ground-truth translation.

Method	$5°2\,\text{cm}$	$5°5\,\text{cm}$	$10°2\,\text{cm}$	$10°5\,\text{cm}$	Median rot. (°)
Orient Anything (Wang et al., 2025b)	0.0	0.0	8.0	8.0	68.1
GenPose++ (Zhang et al., 2024a)	0.0	4.0	16.0	28.0	33.9
Ours	**56.0**	**64.0**	**60.0**	**76.0**	**3.3**

4.5 Ablation Studies

Table 8: **Ablation study on the NOCS REAL275 dataset.** Full model, component ablations, and plain RANSAC.

Variant	IoU_{75}	$5°2\,\text{cm}$	$5°5\,\text{cm}$	$10°2\,\text{cm}$	$10°5\,\text{cm}$
Full model (Ours)	**48.1**	**45.4**	**60.2**	**63.6**	**82.9**
w/o PointNet features	22.9	24.5	34.2	40.0	70.3
w/o point-pair sampling	44.7	36.9	51.6	55.1	77.5
w/o flow matching	41.5	25.9	33.2	50.1	65.1
w/o scale decoupling	44.5	44.0	57.5	61.2	80.5
Plain RANSAC	43.4	43.7	58.2	59.7	78.9

Component analysis Table 8 compares component ablations and plain RANSAC. Excluding geometric PointNet features causes a large drop in IoU_{75} and pose mAP. Removing point-pair sampling, flow matching, or scale decoupling reduces accuracy. Our full recovery also outperforms plain RANSAC. These results support the learned correspondences and geometric recovery; point pairs provide local hypotheses for robust fitting.

Runtime and solver settings On one RTX A6000, the reported pose-stage latency is 2.784 s with 10,000 pairs; 3,000 pairs gives similar REAL275 accuracy at 0.979 s. Tables S3–S5 report pair-count, profiling, and solver-sensitivity results. Three refinement hops remain the default; one to three hops gives $5°5\,\text{cm}$ mAP of 59.9–60.2.

Mask and depth sensitivity We test controlled input corruptions on the full REAL275 test set. Relative to clean-input $5°5\,\text{cm}$ mAP of 60.2, mask erosion by 10% of the equivalent object radius gives 57.8, and 10% depth outliers give 56.6. Dilation and ragged boundaries at the tested severity give 62.9 and 62.1. The perturbation definitions and complete results are in Table S6.

5 Conclusion

We introduced UniPose9D, a universal model for category-agnostic 9D object pose estimation from a masked RGB-D observation or RGB with predicted depth. UniPose9D combines point-pair NOCS prediction, pairwise metric-scale calibration, and adaptive N-hop Kabsch–Umeyama recovery. A single model trained across public datasets achieves competitive benchmark performance and transfers to unseen objects and categories without category labels, CAD models, mean-shape priors, or reference views at inference. Ablations confirm the contributions of point-pair reasoning, flow matching, geometric features, and the pose-recovery components.

Limitations Our method currently focuses on rigid indoor objects of moderate size and assumes an instance mask/ROI. Its accuracy depends on mask and depth quality; monocular depth errors particularly affect metric translation. Flow sampling dominates inference latency, limiting applications that require frequent pose updates. We plan to incorporate more outdoor objects and datasets and extend UniPose9D to articulated and deformable categories.

Reproducibility Statement

Section 3 specifies the learned representation and pose solver; Section 4 defines the evaluation settings. The appendix provides network and training details, dataset descriptions, additional comparisons, and solver pseudocode. Additional analyses report point-pair budgets, profiling hardware, and input-perturbation settings. Scores are retained under their stated protocols rather than pooled across benchmarks.

AI Use Statement

An AI assistant was used to aid and polish the writing of this paper, including improving clarity, wording, and organization. The authors are responsible for the final content.

References

Junhao Cai, Yisheng He, Weihao Yuan, Siyu Zhu, Zilong Dong, Liefeng Bo, and Qifeng Chen. OV9D: Open-vocabulary category-level 9D object pose and size estimation. *arXiv preprint arXiv:2403.12396*, 2024. URL https://arxiv.org/abs/2403.12396.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian mixture flow matching models. In *ICML*, 2025.

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*, pp. 2773–2782, 2021.

Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. FS-Net: Fast shape-based network for category-level 6D object pose estimation with decoupled rotation mechanism. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1581–1590, 2021.

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*, pp. 9959–9969, 2024.

Jaime Corsetti, Davide Boscaini, Francesco Giuliari, Changjae Oh, Andrea Cavallaro, and Fabio Poiesi. High-resolution open-vocabulary object 6D pose estimation. *arXiv preprint arXiv:2406.16384*, 2024a. URL https://arxiv.org/abs/2406.16384.

Jaime Corsetti, Davide Boscaini, Changjae Oh, Andrea Cavallaro, and Fabio Poiesi. Open-vocabulary object 6D pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18071–18080, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Corsetti_Open-Vocabulary_Object_6D_Pose_Estimation_CVPR_2024_paper.html.

Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. GPV-Pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6781–6791, 2022.

Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 998–1005. Ieee, 2010.

Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, 54(3):1677–1734, 2021.

Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. HANDAL: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023. URL https://nvlabs.github.io/HANDAL/.

Rasmus Laurvig Haugaard and Anders Glent Buch. SurfEmb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6749–6758, 2022.

Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. OnePose++: Keypoint-free one-shot object pose estimation without CAD models. *Advances in Neural Information Processing Systems*, 35:35103–35115, 2022a.

Yisheng He, Yao Wang, Haoqiang Fan, Jian Sun, and Qifeng Chen. FS6D: Few-shot 6D pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6814–6824, 2022b.

HyunJun Jung, Shun-Cheng Wu, Patrick Ruhkamp, Guangyao Zhai, Hannah Schieber, Giulia Rizzoli, Pengyuan Wang, Hongcheng Zhao, Lorenzo Garattoni, Sven Meier, et al. HouseCat6D-a large-scale multi-modal category level 6D object perception dataset with household objects in realistic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22498–22508, 2024.

Liming Kuang, Yordanka Velikova, Mahdi Saleh, Jan-Nico Zaech, Danda Pani Paudel, and Benjamin Busam. ConceptPose: Training-free zero-shot object pose estimation using concept vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26582–26592, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Kuang_ConceptPose_Training-Free_Zero-Shot_Object_Pose_Estimation_using_Concept_Vectors_CVPR_2026_paper.html.

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*, pp. 574–591. Springer, 2020.

Jim Lawrence, Javier Bernal, and Christoph Witzgall. A purely algebraic justification of the Kabsch-Umeyama algorithm. *Journal of Research of the National Institute of Standards and Technology*, 124:124028, 2019. doi: 10.6028/jres.124.028. URL https://doi.org/10.6028/jres.124.028.

Taeyeop Lee, Bowen Wen, Minjun Kang, Gyuree Kang, In So Kweon, and Kuk-Jin Yoon. Any6D: Model-free 6D pose estimation of novel objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

Yuchen Li, Kai Qin, Haitao Wu, and Xiangjun Qu. ProM-Pose: Language-guided zero-shot 9-DoF object pose estimation from RGB-D with generative 3D priors. *Electronics*, 15(5):1111, 2026. doi: 10.3390/electronics15051111. URL https://www.mdpi.com/2079-9292/15/5/1111.

Haitao Lin, Zichang Liu, Chilam Cheang, Yanwei Fu, Guodong Guo, and Xiangyang Xue. SAR-Net: Shape alignment and recovery network for category-level 6D object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2022.

Jiehong Lin, Hongyang Li, Ke Chen, Jiangbo Lu, and Kui Jia. Sparse steerable convolutions: An efficient learning of SE(3)-equivariant features for estimation and tracking of object poses in 3D space. *Advances in Neural Information Processing Systems*, 34:16779–16790, 2021a.

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*, pp. 3560–3569, 2021b.

Jiehong Lin, Zewei Wei, Yabin Zhang, and Kui Jia. VI-Net: Boosting category-level 6D object pose estimation via learning decoupled rotations on the spherical representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14001–14011, 2023.

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*, pp. 21040–21049, 2024.

Xiao Lin, Yun Peng, Liuyi Wang, Xianyou Zhong, Minghao Zhu, Yi Feng, Jingwei Yang, Chengju Liu, and Qijun Chen. CleanPose: Category-level object pose estimation via causal learning and knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5990–6000, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Lin_CleanPose_Category-Level_Object_Pose_Estimation_via_Causal_Learning_and_Knowledge_ICCV_2025_paper.html.

Jianhui Liu, Yukang Chen, Xiaoqing Ye, and Xiaojuan Qi. IST-Net: Prior-free category-level pose estimation with implicit space transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13978–13988, 2023.

Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. In *European Conference on Computer Vision*, pp. 298–315. Springer, 2022.

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.

Eitan Marder-Eppstein. Project Tango. In *ACM SIGGRAPH 2016 Real-Time Live!*, pp. 25–25. 2016.

Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, and Vincent Lepetit. NOPE: Novel object pose estimation from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17923–17932, 2024a.

Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. GigaPose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9903–9913, 2024b.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

Martin Runz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE international symposium on mixed and augmented reality (ISMAR)*, pp. 10–20. IEEE, 2018.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6825–6834, 2022.

Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-DoF pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16611–16621, June 2021.

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*, pp. 2642–2651, 2019.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025a.

Tianfu Wang and Hongguang Wang. Pov9D: Point cloud-based open-vocabulary 9D object pose estimation. *Journal of Imaging*, 11(11):380, 2025. doi: 10.3390/jimaging11110380. URL https://www.mdpi.com/2313-433X/11/11/380.

Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient Anything: Learning robust object orientation estimation from rendering 3D models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 65556–65574. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wang25er.html.

Zehan Wang, Ziang Zhang, Jiayang Xu, Jialei Wang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient Anything V2: Unifying orientation and rotation understanding. In *Advances in Neural Information Processing Systems*, volume 38, 2025c. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/4bda4364bf10a454769fdbeb8c02df16-Abstract-Conference.html.

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.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17868–17879, 2024.

Jay M Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David MS Johnson, et al. SegICP: Integrated deep semantic segmentation and pose estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5784–5789. IEEE, 2017.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.019.

Yang You, Wenhao He, Jin Liu, Hongkai Xiong, Weiming Wang, and Cewu Lu. CPPF++: Uncertainty-aware Sim2Real object pose estimation by vote aggregation. *arXiv preprint arXiv:2211.13398*, 2022a.

Yang You, Ruoxi Shi, Weiming Wang, and Cewu Lu. CPPF: Towards robust category-level 9D pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6866–6875, 2022b.

Yang You, Yixin Li, Congyue Deng, Yue Wang, and Leonidas Guibas. Multiview equivariance improves 3D correspondence understanding with minimal feature finetuning. *arXiv preprint arXiv:2411.19458*, 2024a.

Yang You, Kai Xiong, Zhening Yang, Zhengxiang Huang, Junwei Zhou, Ruoxi Shi, Zhou Fang, Adam W Harley, Leonidas Guibas, and Cewu Lu. PACE: A large-scale dataset with pose annotations in cluttered environments. In *European Conference on Computer Vision*, pp. 473–489. Springer, 2024b.

Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017.

Jiyao Zhang, Mingdong Wu, and Hao Dong. Generative category-level object pose estimation via diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 54627–54644. Curran Associates, Inc., 2023. doi: 10.52202/075280-2381.

Jiyao Zhang, Weiyao Huang, Bo Peng, Mingdong Wu, Fei Hu, Zijian Chen, Bo Zhao, and Hao Dong. Omni6DPose: A benchmark and model for universal 6D object pose estimation and tracking. *arXiv preprint arXiv:2406.04316*, 2024a.

Mengchen Zhang, Tong Wu, Tai Wang, Tengfei Wang, Ziwei Liu, and Dahua Lin. Omni6D: Large-vocabulary 3D object dataset for category-level 6D object pose estimation. In *European Conference on Computer Vision*, 2024b. URL https://arxiv.org/abs/2409.18261.

Linfang Zheng, Chen Wang, Yinghan Sun, Esha Dasgupta, Hua Chen, Aleš Leonardis, Wei Zhang, and Hyung Jin Chang. HS-Pose: Hybrid scope feature extraction for category-level object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17163–17173, 2023.

APPENDIX

A ADDITIONAL EVALUATION RESULTS

Native metrics. PACE AP@0:20° averages thresholds from 0° to 20° in 1° steps; AP@0:5 cm uses 0.25 cm steps. Joint AP evaluates both constraints. Omni6DPose uses its native AUC and VUS protocol, while REAL275 and HouseCat6D use pose mAP. These metrics are retained separately throughout the comparisons.

A.1 UNIFIED TRAINING OF BASELINES

Table S1 compares methods trained on the same four-dataset mixture. We retain the native metric for each benchmark: threshold-integrated AP on PACE, VUS on Omni6DPose, and pose mAP on REAL275 and HouseCat6D. UniPose9D leads the jointly trained variants in all four columns. The NOCS-only SecondPose and CleanPose checkpoints retain higher REAL275 accuracy; their scores and those of the jointly trained models correspond to different training settings. CleanPose uses a 190-category head for the combined taxonomy. Its NOCS-only diagnostic score is reproduced, distinct from the published result in Table 3.

Table S1: **Baselines under the unified four-dataset training protocol.** "All-DS" denotes training on the same PACE, Omni6DPose, NOCS REAL275, and HouseCat6D mixture as UniPose9D. Type: **Cat** = category-level specialist, **Gen** = generalizable. SecondPose and CleanPose remain category-conditioned under All-DS training. CleanPose (All-DS) uses a 190-category head in this diagnostic. Dashes indicate unreported native-checkpoint results, not measured zero scores.

Method	Type	PACE (0:20°)	Omni6DPose (5°2 cm)	REAL275 (5°2 cm)	HouseCat6D (5°2 cm)
GenPose++ (Omni)	Gen	21.8	10.0	10.3	1.8
GenPose++ (All-DS)	Gen	31.2	7.1	28.5	9.8
SecondPose (NOCS)	Cat	–	–	56.2	–
SecondPose (All-DS)	Cat	24.5	5.1	36.2	7.9
CleanPose (NOCS, reproduced)	Cat	–	–	**61.6**	–
CleanPose (All-DS)	Cat	13.4	2.3	4.0	1.7
UniPose9D	Gen	**40.4**	**10.5**	45.4	**16.9**

A.2 UNSEEN-CATEGORY EVALUATION DETAILS

The category subsets in Table S2 exclude categories represented in the four-dataset pose-training mixture. This does not assert that the same categories were absent from pretrained DINOv2 or MoGe-2 data. We use the category names recorded in the evaluation:

- **YCB-Video (5):** clamp, foam brick, pitcher, power drill, wood block.
- **HOPE (2):** squeeze bottle, milk carton.
- **Omni6D (13):** almond, apricot, artichoke, bagel, chestnut, earplug, fig, ginger, pagoda dish, passiflora edulis, passion fruit, pistachio, taro.
- **HANDAL (5):** ladle, power drill, spatula, strainer, whisk.

Each joint threshold requires both rotation and translation errors to fall below the specified limits. The subset scores are reported separately for each dataset. Omni6D is a separate dataset from Omni6DPose. The baselines train on Omni6D_train and test on the held-out-category split. Our zero-shot evaluation uses 9,530 examples from the listed 13 categories of the same unseen split, without Omni6D pose training.

Published baseline results in Table 6 report mAP. Our Omni6D evaluation uses GT visible masks and the released object symmetry annotations, and reports pooled per-instance accuracy. It uses 1,024 input points, 10,000 point pairs, and two RANSAC fitting passes with up to 1,000 hypotheses per pass. The same-sample HANDAL comparison is given in Table 7.

Table S2: **Category-held-out subsets.** Joint-threshold scores (%) for RGB-D input. Counts denote evaluated examples. The full Omni6D result and same-subset HANDAL comparison appear in Tables 6 and 7.

Dataset	Categories	Count	5°2 cm	5°5 cm	10°2 cm	10°5 cm
YCB-Video	5	3,543	40.6	40.8	63.8	64.2
HOPE	2	225	64.9	64.9	85.8	86.7
Omni6D	13	9,530	15.6	26.4	20.3	32.3
HANDAL	5	25	56.0	64.0	60.0	76.0

A.3 Runtime and Pair Sampling

We draw T pairs from $N = 1{,}024$ sampled object points. The $O(N^2)$ candidate pool is not exhaustively evaluated. Table S3 varies only the pair count in the recorded REAL275 evaluation, keeping $T = 10{,}000$ as the default for the paper's other reported results. Peak GPU memory rises from 1.11 to 1.48 GiB over the tested range.

Table S3: **Sampled-pair cost on REAL275.** Reported pose-stage latency on one RTX A6000, $N = 1{,}024$ object points. All main results retain $T = 10{,}000$ pairs; $T = 3{,}000$ is an alternative operating point. Module-level latency is in Table S4.

Pairs T	mAP@5°5 cm	mAP@10°5 cm	Latency (s)	GPU memory (GiB)
1,000	58.4	82.2	0.837	1.11
3,000	60.3	83.0	0.979	1.17
10,000 (default)	60.2	82.9	2.784	1.48

Table S4 gives the recorded latency components on one RTX A6000.

Table S4: **Per-object module latency.** Default $T = 10{,}000$ on one RTX A6000. These measurements characterize the reported pose pipeline and do not establish end-to-end RGB latency including mask and depth prediction.

Component	Time (s)
Preprocessing and aggregation	0.105
Point features	0.117
Flow sampling	2.360
RANSAC	0.201
Reported total	2.784

A.4 Solver Sensitivity

Table S5 evaluates the threshold and refinement-hop settings on the full REAL275 test set. The adaptive threshold is $\tau = 0.05\bar{s}_\infty$, where $\bar{s}_\infty = \| \, |\mathcal{I}|^{-1} \sum_{\mathcal{T}\in\mathcal{I}} \hat{\mathbf{s}}_\mathcal{T} \|_\infty$ is the largest dimension of the mean box-size vector over selected pairs, as in Equation (7). The threshold is relaxed by a factor of 1.5 when support is insufficient. The refinement loop is summarized in Algorithm 1. Fixed 1 cm and 2 cm thresholds give similar scores on this dataset; a 5 cm threshold reduces accuracy. The difference between one and three hops is 0.3 points at 5°5 cm. These measurements support limited sensitivity over the tested hop range, but do not by themselves validate threshold transfer across all object sizes.

A.5 Mask and Depth Perturbations

We evaluate controlled input corruptions on the full REAL275 test set (2,754 images), with $T = 10{,}000$ pairs and 1,000 RANSAC hypotheses. For a mask of area A, severities are relative to its equivalent radius $r_{\mathrm{eq}} = \sqrt{A/\pi}$. Erosion and dilation use 10% of this radius. The ragged-boundary test flips each pixel in a band of this width around the silhouette with probability 0.5; this operation

Table S5: **Solver sensitivity on REAL275.** Pose mAP (%); the default adaptive threshold and three hops are retained.

Setting	$5°5$ cm	$10°5$ cm
Adaptive threshold, $0.05\bar{s}_\infty$ (default)	60.2	82.9
Fixed threshold, 1 cm	59.8	82.8
Fixed threshold, 2 cm	59.6	81.6
Fixed threshold, 5 cm	50.3	76.4
1 refinement hop	59.9	82.7
2 refinement hops	60.1	82.8
3 refinement hops (default)	60.2	82.9

does not guarantee identical mask area. The depth test corrupts 10% of depth values. Table S6 reports $5°5$ cm mAP. Erosion and depth outliers reduce the score by 2.4 and 3.6 points; the other two perturbations yield higher scores in this recorded run. These tests probe specific corruptions and do not cover missing instances, severe truncation, or arbitrary segmentation failures.

Table S6: **Controlled input perturbations on REAL275.** Changes are in percentage points relative to clean-input $5°5$ cm mAP.

Input condition	mAP@$5°5$ cm	Change
Clean	60.2	–
Mask erosion, $0.1r_{\text{eq}}$	57.8	-2.4
Mask dilation, $0.1r_{\text{eq}}$	62.9	$+2.7$
Ragged boundary, $0.1r_{\text{eq}}$ band	62.1	$+1.9$
Depth outliers, 10%	56.6	-3.6

A.6 Component Ablation Definitions

In Table 8, "w/o flow matching" uses direct regression with an ℓ_2 loss. "w/o point-pair sampling" independently predicts each point's NOCS coordinates from its concatenated PointNet and DINOv2 features. Plain RANSAC directly fits a similarity transform without pairwise scale calibration; $H = 1$ in Table S5 retains the proposed scale calibration and rigid fitting, changing only the hop count.

B Network Implementation Details

B.1 Feature Extraction Formulation

For visual feature extraction, we use DINOv2-S/14 enhanced by 3DCorrEnhance (You et al., 2024a). Given an ROI crop $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ and the image location $\mathbf{u}_i$ of a point, we refine the patch-token map with a 3×3 convolution, bilinearly sample it, and apply ℓ_2 normalization:

$$\mathbf{f}_i^{\text{img}} = \text{norm}_2(\text{bilinear}(\text{Conv}_{3\times 3}(\text{DINOv2}(\mathbf{I})), \mathbf{u}_i)) \in \mathbb{R}^{384}. \tag{8}$$

For geometric features, we compute local point descriptors with a lightweight, multi-radius PointNet. Given the observed point cloud $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$, we aggregate neighborhoods at radii $r_1<r_2<r_3$ using radius grouping. We process each neighborhood with a small residual MLP followed by max pooling, then concatenate and fuse the resulting features:

$$\mathbf{f}_i^{\text{geo}} = \Phi(\oplus_{k=1}^3 \max(\text{MLP}(\mathcal{N}_{r_k}(\mathbf{p}_i)))) \in \mathbb{R}^{d_{\text{geo}}}, \tag{9}$$

where $\oplus$ is concatenation and $\mathcal{N}_{r_k}(\mathbf{p}_i)$ represents the neighborhood points within radius r_k of $\mathbf{p}_i$.

We project both modalities to a common width d and concatenate:

$$\mathbf{f}_i = [\mathbf{W}_v\mathbf{f}_i^{\text{img}} \oplus \mathbf{W}_g\mathbf{f}_i^{\text{geo}}] \in \mathbb{R}^{2d}, \tag{10}$$

B.2 Network Architecture

The network combines visual and geometric descriptors, encodes context-augmented point pairs, and predicts a joint NOCS–size flow. Each residual MLP block (ResLayer) contains two linear layers with ReLU between them and an additive skip connection; the skip uses a linear projection when the input and output widths differ.

Visual Branch We use DINOv2-S/14 (Oquab et al., 2023), enhanced by 3DCorrEnhance (You et al., 2024a), with rank-4 low-rank adapters on the query and value projections of its last four transformer blocks. The final normalized patch tokens form a 384-channel feature map. A 3×3 convolution with 384 input and output channels and padding 1 refines this map. Bilinear interpolation and ℓ_2 normalization give per-point descriptors, which a linear layer projects from 384 to $d = 512$ dimensions.

Geometric Branch We employ a multiscale PointNet (Qi et al., 2017) at radii $\{0.02, 0.04, 0.08\}$, with neighborhood capacities of 32, 64, and 96 points. At each radius, three ResLayers map relative neighborhood coordinates through $3 \to 128 \to 128 \to 128$, followed by max pooling. The three pooled descriptors concatenate to 384 dimensions and pass through ResLayers of widths $384 \to 256 \to 256 \to 512$. A separate $512 \to 512$ linear projection produces the geometric descriptor. Concatenating it with the projected visual descriptor gives 1,024 dimensions per point.

Point Pair Feature Extraction Each target pair $\mathcal{T} = (i_1, i_2)$ is augmented with three context points, forming a five-point tuple $\mathcal{Q}_\mathcal{T}$. We concatenate all ten pairwise 3D differences in metric coordinates (30 dimensions). The five point descriptors concatenate to 5,120 dimensions and are projected linearly to 512. Their concatenation with the coordinate differences gives a 542-dimensional input to five ResLayers with widths $542 \to 512 \to 512 \to 512 \to 512 \to 512$. Four pre-normalized transformer encoder layers then exchange information across the sampled tuples, using eight attention heads and feed-forward width 2,048. Each target pair receives a 512-dimensional contextual feature.

Joint Flow Head The head models a nine-dimensional state $\mathbf{z} = [2\mathbf{x}_{i_1} \oplus 2\mathbf{x}_{i_2} \oplus \log \mathbf{s}]$, comprising two NOCS points and three log box dimensions. A 512-dimensional sinusoidal time encoding passes through a $512 \to 512 \to 512$ MLP with ReLU between its linear layers. Concatenating the contextual feature, learned time embedding, and noisy state gives 1,033 inputs. Three ResLayers of widths $1033 \to 512 \to 512 \to 320$ predict 32 mixture components, each with a nine-dimensional flow mean and one mixture-weight logit. Log-softmax normalizes the logits.

A separate time-only MLP, with SiLU before each linear layer and widths $512 \to 512 \to 512 \to 1$, predicts a scalar log standard deviation shared across tuples, mixture components, and state coordinates. It receives a detached time embedding. After flow sampling, the first six state coordinates are divided by two to recover the target pair's NOCS coordinates, and the final three are exponentiated to recover metric box size. Thus both outputs come from the same joint head.

B.3 Joint Flow Training

Targets and tuple sampling We sample all five tuple indices uniformly with replacement from the observed object points. The first two indices define the target pair, followed by the three context indices in descriptor concatenation. Target NOCS coordinates are clipped to $[-0.5, 0.5]$ and multiplied by two; box dimensions are floored at 10^{-4} before taking their logarithms. Concatenating the two normalized NOCS points and three log box dimensions gives $\mathbf{z}_0 \in \mathbb{R}^9$ for each tuple.

Forward bridge Let ϵ_t and ρ denote the time floor and transition ratio. We draw $q \sim \mathcal{U}[0, 1)$ per image and set

$$t = \max(q, \epsilon_t), \qquad \ell = \max(0, \min((1 - \rho)t, t - \epsilon_t)), \tag{11}$$

where $\ell = t_{\text{low}}$. For the linear schedule $\alpha(t) = 1 - t$, $\sigma(t) = t$, two independent Gaussian vectors $\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_1 \sim \mathcal{N}(0, \mathbf{I}_9)$ construct the paired states:

$$\mathbf{z}_\ell = (1 - \ell)\mathbf{z}_0 + \ell\boldsymbol{\epsilon}_0, \tag{12}$$

$$\mathbf{z}_t = a\mathbf{z}_\ell + \sqrt{t^2 - a^2\ell^2}\,\boldsymbol{\epsilon}_1, \qquad a = \frac{1 - t}{1 - \ell}. \tag{13}$$

Each noise vector is shared across tuples within an image. The paired states have the stated noise marginals and are coupled through this forward bridge. The normalized time is multiplied by a fixed embedding scale before sinusoidal encoding.

Reverse-transition likelihood The network outputs flow-component means $\mathbf{u}_g \in \mathbb{R}^9$, weights π_g, and shared standard deviation σ_u. Following the Gaussian reverse bridge (Chen et al., 2025), define

$$b = 1 - \left(\frac{\ell(1-t)}{t(1-\ell)}\right)^2, \qquad c_1 = \left(\frac{\ell}{t}\right)^2 \frac{1-t}{1-\ell}, \tag{14}$$

$$c_2 = b(1-\ell), \qquad c_3 = b\ell^2. \tag{15}$$

The induced transition mixture retains π_g and has parameters

$$\mathbf{m}_g = c_1\mathbf{z}_t + c_2(\mathbf{z}_t - t\mathbf{u}_g), \qquad v = (tc_2\sigma_u)^2 + c_3. \tag{16}$$

We evaluate the mixture likelihood of the full nine-dimensional target $\mathbf{z}_\ell$ and average its negative logarithm over images and tuples. This jointly supervises NOCS and log-size through an analytical transition, allowing simulation-free training. For numerical stability, the reverse mapping floors denominators and log arguments at 10^{-6}, and the likelihood caps inverse standard deviations at 10^4.

B.4 Implementation Settings

We minimize $\mathcal{L}_{\text{joint}}$ with $1{,}000$ point pairs per training image. We use AdamW with a base learning rate of 10^{-4}, cosine decay with linear warmup, and gradient clipping at norm 1. Visual features come from DINOv2-S/14 enhanced by 3DCorrEnhance (You et al., 2024a); the PointNet is trained from scratch with grouping radii $\{0.02, 0.04, 0.08\}$. We use a trainable geometric encoder to accommodate depth statistics across datasets and sensors; its feature contribution is evaluated in Section 4.5. Training uses the PACE, Omni6DPose, NOCS, and HouseCat6D training splits with balanced sampling. At inference, we sample N=1,024 object points and T=10,000 point pairs by default. Each pair uses an ODE sampler with 16 steps and 2 substeps per step. The pair-count trade-off is evaluated in Section 4.5.

B.5 Training Details

We train our model using the AdamW optimizer with a base learning rate of 10^{-4}. We employ a cosine annealing learning rate schedule with linear warmup during the first epoch. The model is trained for 30 epochs on one NVIDIA A6000 GPU with a batch size of 16. We use a balanced sampling strategy to ensure that each dataset (PACE, Omni6DPose, NOCS, HouseCat6D) contributes equally to the training batches.

Data Augmentation We apply data augmentation to improve generalization. For RGB images, we apply random color jittering (brightness, contrast, saturation, hue) and add random Gaussian noise. For point clouds, we apply random jittering to the points. We also employ a Dynamic Zoom-In (DZI) strategy during training, where we randomly crop and resize the input images and point clouds to simulate different object scales and camera distances.

C Dataset Details

In this section, we summarize the statistics of the training and test datasets used in our experiments.

C.1 Training Datasets

PACE We use the PACE-Sim training split of PACE (You et al., 2024b), which contains 100K photorealistic simulated frames with 2.4M annotations across 931 objects.

Omni6DPose We use the SOPE (Simulated Omni-Pose Estimation) split of Omni6DPose (Zhang et al., 2024a) for training. SOPE contains diverse objects placed in photorealistic simulated environments based on IKEA, Matterport3D, and ScanNet++ scenes. It provides 6D pose and metric-scale annotations for 475K images and 5M object instances, covering 4,162 objects from 149 categories.

NOCS REAL275 We use the real training split of NOCS REAL275 (Wang et al., 2019), which contains approximately 4.3K real-world RGB-D frames from 7 scenes. It covers 6 object categories: bottle, bowl, camera, can, laptop, and mug.

HouseCat6D We use the training split of HouseCat6D (Jung et al., 2024), which contains 34 large-scale scenes with approximately 20K real-world RGB-D frames captured in cluttered household environments. It covers 194 instances across 10 categories, including photometrically challenging classes such as glass and cutlery.

C.2 Test Datasets

Published baseline results retain their corresponding localization protocols, as summarized below.

PACE We use the official PACE test benchmark, which consists of 55K frames with 258K annotations across 300 videos, covering 238 objects from 43 categories. The published benchmark provides ground-truth detections to its category-level baselines, except NOCS, which jointly predicts detections and poses using Mask R-CNN (You et al., 2024b).

Omni6DPose For evaluation, we use the ROPE (Real Omni-Pose Estimation) split of Omni6DPose. ROPE consists of real-world video recordings of objects from the test set, providing a challenging benchmark for generalization to real sensor noise and lighting conditions. It contains 332K images with 1.5M annotations, covering 581 objects from 149 categories. Its published pose benchmark uses ground-truth instance masks for the two-stage methods; NOCS jointly predicts detections and poses (Zhang et al., 2024a). The GenPose++ row in Table 2 is its reported ground-truth-mask result.

NOCS REAL275 We evaluate on the real test split of NOCS REAL275, comprising 2.75K real-world frames from 6 unseen scenes, covering the same 6 categories. Published two-stage evaluations, including SecondPose and CleanPose, use predicted Mask R-CNN instance segmentations as inputs to pose estimation (Chen et al., 2024; Lin et al., 2025).

HouseCat6D We use the official 5-scene test split of HouseCat6D, which comprises approximately 3K real-world RGB-D frames covering the same 10 categories under unseen scene configurations and viewpoints. The original category-level pose benchmark evaluates its baselines using ground-truth detection masks (Jung et al., 2024).

YCB-Video For cross-domain evaluation, we use the official test split of YCB-Video (Xiang et al., 2018), which consists of 12 RGB-D videos covering 21 YCB objects in cluttered tabletop scenes (the full benchmark contains 92 videos with 133,827 frames in total). Following standard practice, we evaluate on the keyframes of the 12 test videos. This dataset is excluded from the pose-training mixture. Dataset transfer and the stricter unseen-category subset evaluation are reported separately in the main paper.

HOPE For cross-domain evaluation, we use the HOPE-Image benchmark (Tyree et al., 2022), which provides RGB-D images annotated with 6-DoF poses of 28 toy grocery objects across 50 scenes from 10 household and office environments, with up to five lighting variations per scene. We evaluate on the 188 test images. This dataset is excluded from the pose-training mixture. Dataset transfer and the stricter unseen-category subset evaluation are reported separately in the main paper.

For YCB-Video and HOPE, Table 5 evaluates pose from provided instance masks/ROIs, reporting median rotation error and rotation/translation threshold accuracies.

D Baseline Results on Other Datasets

Baseline Generalization Category-level methods, such as AG-Pose (Lin et al., 2024), VI-Net (Lin et al., 2023), and SecondPose (Chen et al., 2024), are constrained to the object categories on which they were trained (e.g., the six NOCS REAL275 categories). These methods are normally trained within a fixed taxonomy. We report a checkpoint-transfer diagnostic using their official NOCS REAL275

checkpoints on HouseCat6D. Table S7 reports their scores alongside our shared model. The training exposure differs: our model includes HouseCat6D training data, whereas these checkpoints do not. These numbers therefore do not isolate category generalization or architecture.

Table S7: **Baseline generalization results on HouseCat6D.** The *Type* column matches the main paper: **Cat** = category-level specialist (per-dataset training, possibly with category labels or mean-shape priors), **Gen** = generalizable (no category labels, mean-shape priors, CAD models, or reference views at inference). The Cat baselines use NOCS REAL275 checkpoints; our model has seen HouseCat6D training data. This diagnostic has unequal training exposure.

Method	Type	IoU_{25}	IoU_{50}	$5°2\,\text{cm}$	$5°5\,\text{cm}$	$10°2\,\text{cm}$	$10°5\,\text{cm}$
VI-Net (Lin et al., 2023)	Cat	48.4	19.2	1.9	2.3	3.1	4.5
AG-Pose (Lin et al., 2024)	Cat	79.6	38.9	0.8	1.5	2.2	4.5
SecondPose (Chen et al., 2024)	Cat	49.7	18.4	1.4	2.2	3.6	6.2
Ours	Gen	**86.4**	**70.6**	**16.9**	**19.0**	**37.5**	**43.2**

E N-HOP KABSCH–UMEYAMA PSEUDOCODE

Discussion We combine point-pair-level context, multimodal pairwise NOCS prediction, and an adaptive N-hop estimator to accommodate partial observations and symmetry-induced ambiguity. Symmetric objects can still admit multiple equally valid poses. Pairwise scale calibration estimates the scalar normalization geometrically (Equation 6), while the learned metric box sizes determine the adaptive threshold and final box dimensions. The threshold can increase or decrease as the inlier set changes. The component ablation in Section 4.5 evaluates their contributions to overall pose accuracy.

Algorithm 1 fits a rigid transform to scale-calibrated correspondences. Here $\mathcal{J}$ indexes individual inlier correspondences, and $\text{pairs}(\mathcal{J})$ selects each parent pair once if either endpoint is an inlier. Support and rigid refitting use individual correspondences; size aggregation uses the selected pairs. The loop assumes at least one supported hypothesis.

For observed points in meters, let $\mathbf{m} = \text{med}_i\, \mathbf{p}_i$ and $\mathbf{a} = \text{med}_i\, |\mathbf{p}_i - \mathbf{m}|$, using coordinate-wise absolute values and medians (lower medians for even sample counts). The extent proxy and initial threshold are

$$d_{\text{MAD}} = 4\|\mathbf{a}\|_2, \qquad \tau_0 = \max(0.01, \min(0.05, 0.03 d_{\text{MAD}}))\,. \tag{17}$$

Algorithm 1: N-hop Kabsch–Umeyama with adaptive threshold

Input: Flattened scale-calibrated correspondences $\mathcal{F}^s$ (two per pair), per-pair box sizes $\hat{\mathbf{s}}$, number of hops H, and initial threshold τ_0
Output: $\mathbf{R}$, $\mathbf{t}$, $\mathbf{s}$

$\tau := \tau_0$
$\mathcal{J}_{\text{best}} := \varnothing$
for $h = 1$ **to** H **do**
  $(\hat{\mathbf{R}}, \hat{\mathbf{t}}, \mathcal{J}) :=$ RANSAC-Umeyama$(\mathcal{F}^s, \tau)$
  if $|\mathcal{J}| > |\mathcal{J}_{best}|$ **then**
    $\mathcal{J}_{\text{best}} := \mathcal{J}$
  if $|\mathcal{J}| > 0.1|\mathcal{F}^s|$ **then**
    $\tau := 0.05 \cdot \left\| \underset{\mathcal{T}_i \in \text{pairs}(\mathcal{J})}{\text{mean}} \hat{\mathbf{s}}_{\mathcal{T}_i} \right\|_\infty$
  else
    $\tau := 1.5 \cdot \tau$
$(\mathbf{R}, \mathbf{t}) :=$ Umeyama$(\mathcal{F}^s[\mathcal{J}_{\text{best}}])$
$\mathbf{s} := \underset{\mathcal{T}_i \in \text{pairs}(\mathcal{J}_{\text{best}})}{\text{mean}} \hat{\mathbf{s}}_{\mathcal{T}_i}$
return $\mathbf{R}, \mathbf{t}, \mathbf{s}$

F ADDITIONAL QUALITATIVE COMPARISONS

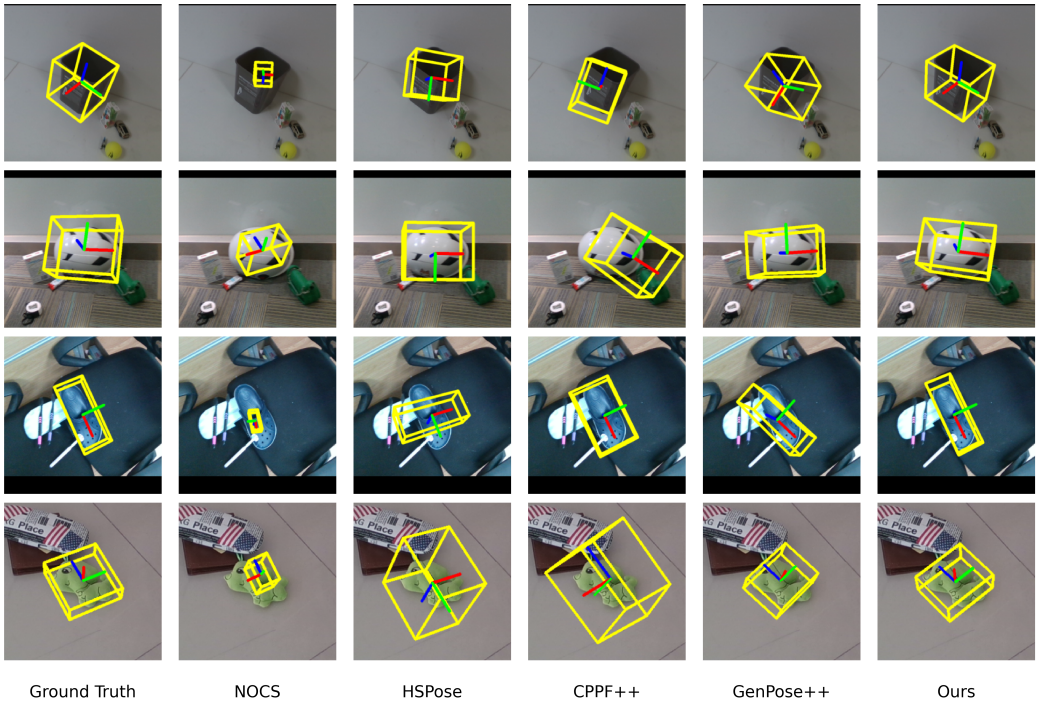

Figure S1: **Qualitative comparison on PACE.** Against NOCS, HS-Pose, CPPF++, and GenPose++, our method yields tighter 3D boxes and more consistent axes.

We present additional qualitative comparisons with Orient Anything and GenPose++ on the cross-domain YCB-Video dataset (Figures S2 and S3) and HOPE dataset (Figures S5 and S4). Both datasets contain diverse objects in cluttered scenes with severe occlusion. Our method recovers the 9D poses of these unseen objects without CAD models. Figure S6 shows results from additional image domains.

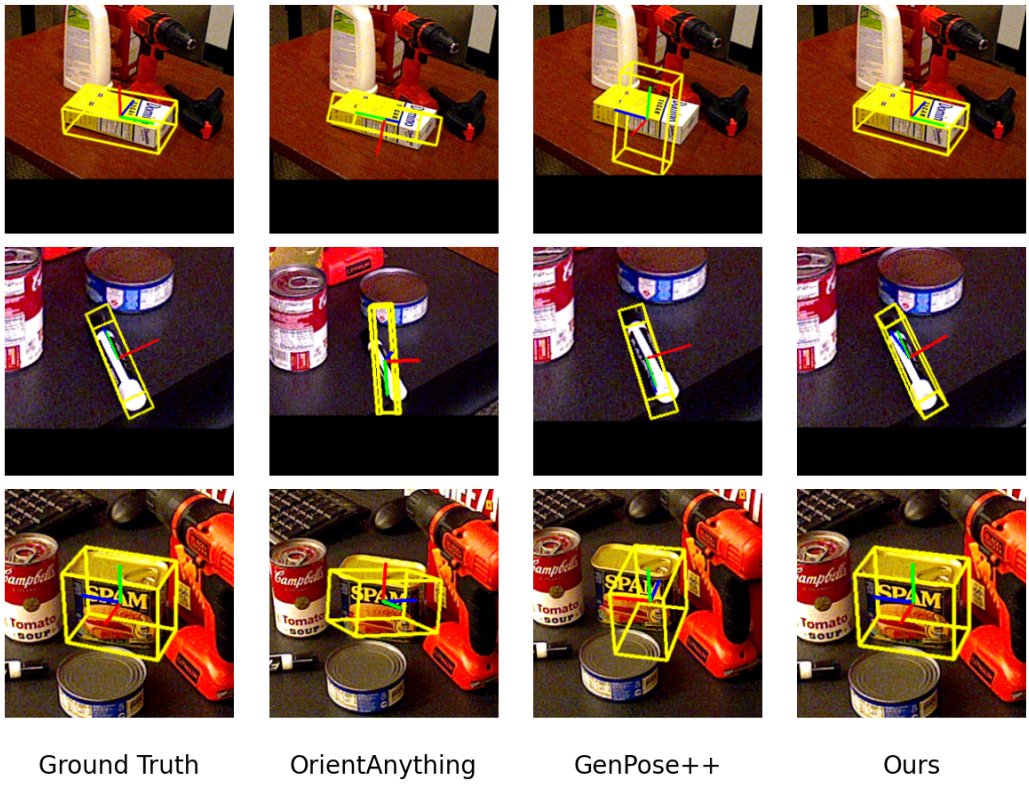

Figure S2: **Qualitative comparison on the YCB-Video dataset.** Compared with Orient Anything and GenPose++, our method estimates 9D poses for unseen objects in cluttered scenes with mutual occlusions.

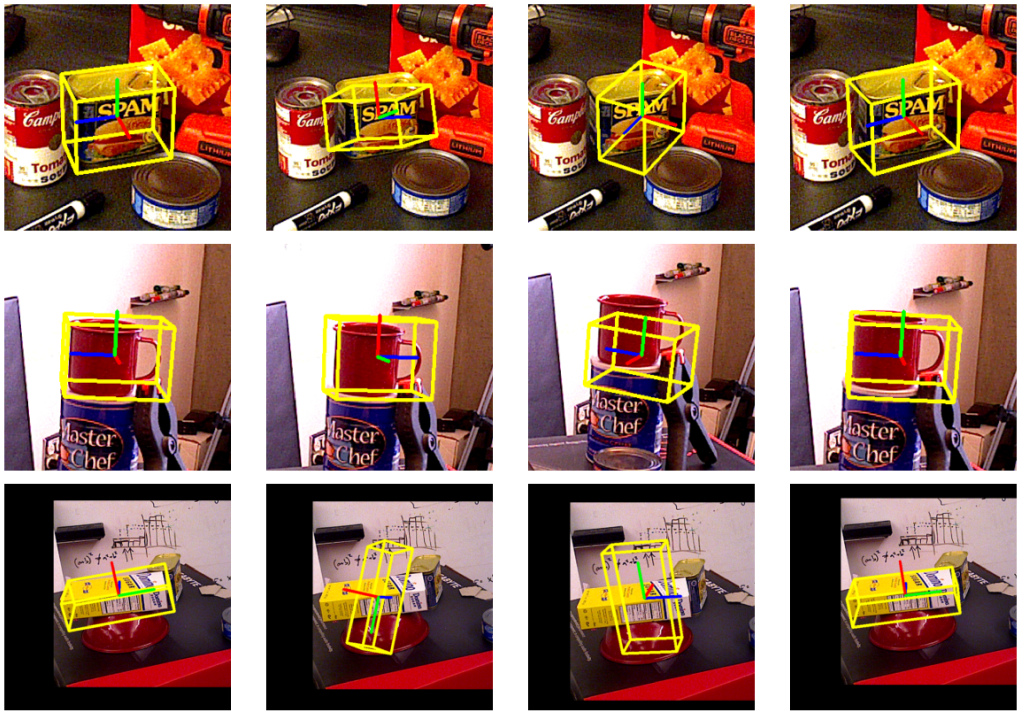

Figure S3: **YCB-Video qualitative comparison with RGB input.** In cluttered scenes with occlusion, our method produces tighter metric boxes and more stable axes than Orient Anything and GenPose++. We use the ground-truth translation and scale for the visualization of Orient Anything.

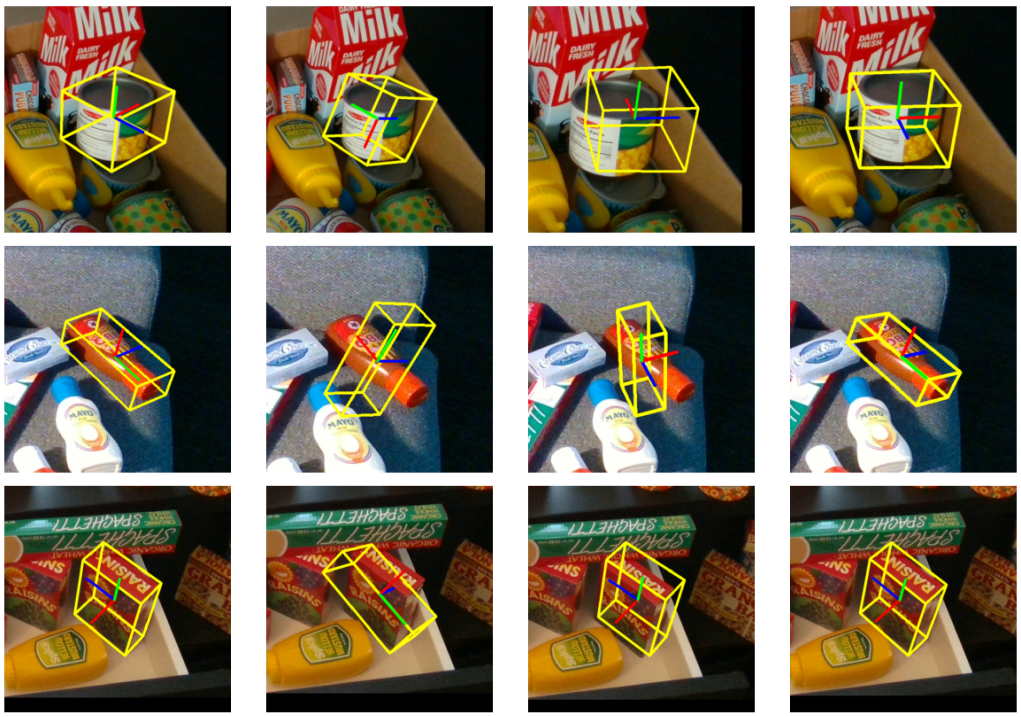

Figure S4: **HOPE qualitative comparison with RGB input.** In cluttered tabletop scenes with heavy occlusion and distractors, our method recovers tighter metric boxes and consistent axes than Orient Anything and GenPose++. We use the ground-truth translation and scale for the visualization of Orient Anything.

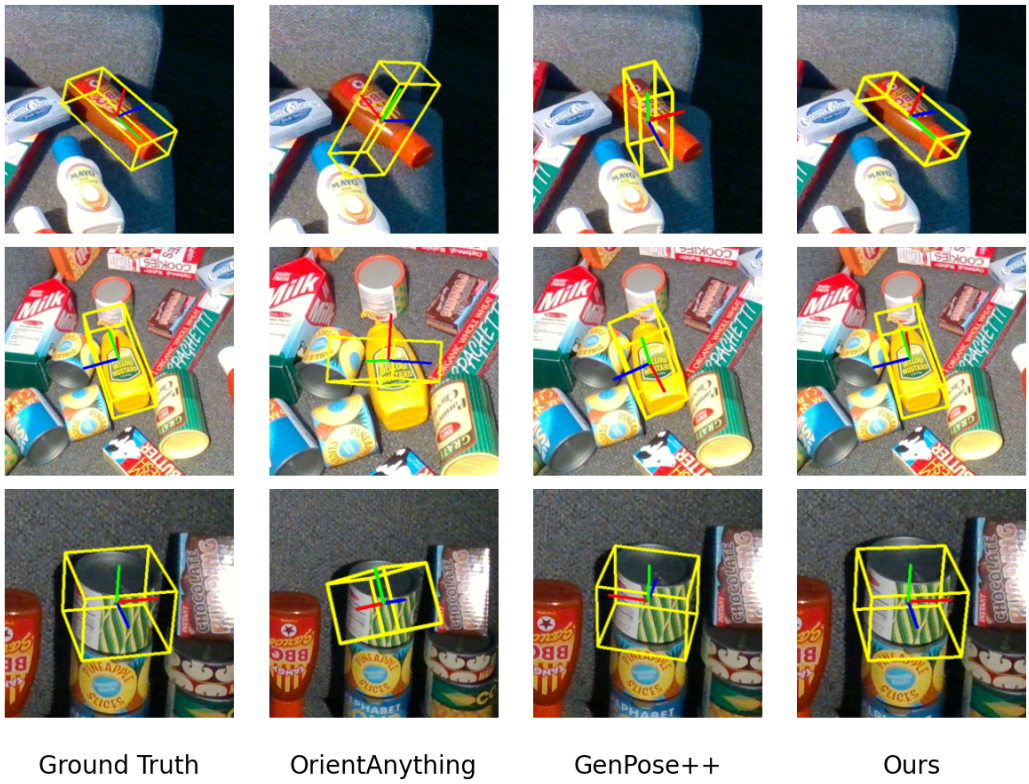

Figure S5: **Qualitative comparison on the HOPE dataset.** Compared with Orient Anything and GenPose++, our method recovers poses and metric box sizes despite heavy occlusions and diverse object types.

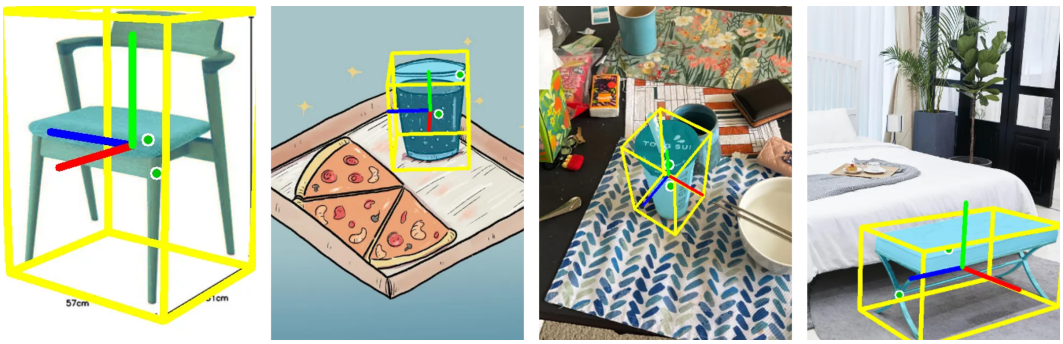

Figure S6: **Pose estimation in the wild.** Our method estimates 9D poses across images with artifacts and domain shifts.

G FAILURE CASES AND LIMITATIONS

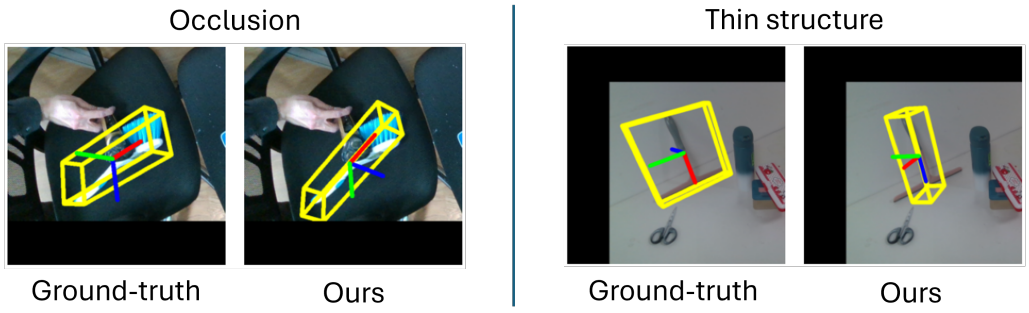

Figure S7: **Failure cases. Left:** Severe occlusion can produce unreliable NOCS predictions and incorrect pose estimates. **Right:** Very thin structures can create ambiguity in bounding box size and orientation.

Two visual failure modes are illustrated here; additional limitations are discussed in Section 5. First, as illustrated in the left panel of Figure S7, severe occlusion can hinder point-pair sampling and leave too few reliable correspondences for RANSAC to recover the correct pose. Second, objects with extremely thin structures (right panel of Figure S7) can create ambiguity in the estimated bounding box size and orientation.