Emergence of a Shared Canonical Object Frame from In-the-Wild Videos

Tom Fischer[1] **Martin Sundermeyer**[2] **Adam Kortylewski**[3] **Eddy Ilg**[1,†]
[1]University of Technology Nuremberg
[2]Google
[3]CISPA Helmholtz Center for Information Security

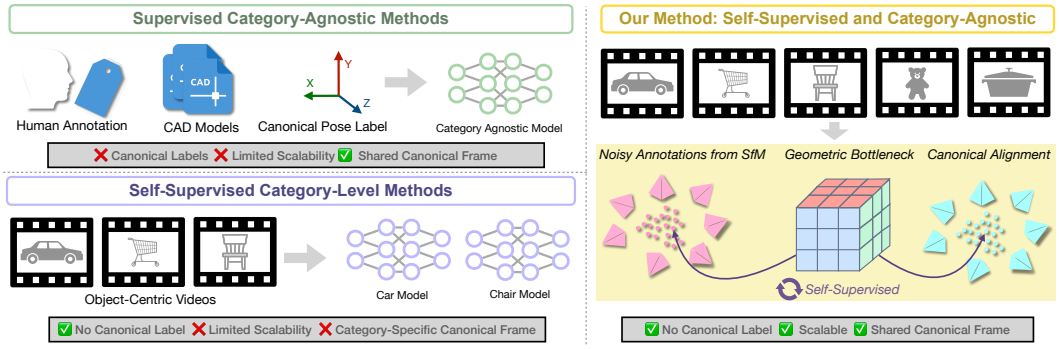


Figure 1: **Three approaches for learning of canonical object frames.** *Top left:* Supervised methods can obtain a shared canonical frame from rendering aligned CAD models or manual pose annotation, which is expensive to scale. *Bottom left:* Self-supervised category-level methods learn from object-centric videos without canonical labels, but train separate models per category, limiting scalability and producing no shared frame across categories. *Right:* Our approach trains a single model across all categories by routing noisy SfM reconstructions through a shared geometric bottleneck, producing a shared canonical frame without canonical pose labels or category conditioning.

Abstract

Comparing object orientations and positions across different instances requires their poses to be expressed in a shared canonical frame. Establishing such frames has traditionally required manual annotation, creating a scaling bottleneck that limits category and instance diversity. We show that a shared canonical frame can instead emerge from self-supervised training on object-centric videos of rigid objects captured in the wild, using only noisy camera poses from Structure-from-Motion. Our key idea is to route all training sequences through a shared geometric bottleneck: a coarse canonical mesh that carries no category-specific detail. By learning dense correspondences from image pixels to this mesh, and estimating per-sequence alignments from noisy SfM geometry, a common canonical frame emerges from multi-view consistency and the semantic priors of the feature extractor, without any canonical pose labels or category conditioning. Trained in a self-supervised manner on $\sim$160,000 in-the-wild object videos, our method achieves competitive accuracy on category-level pose estimation benchmarks compared to methods that rely on canonical pose supervision. The code and checkpoint are available on github.com/Fischer-Tom/Emergent-Canonical-Frame/.

[†]Now at Google; work done while at the University of Technology Nuremberg.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

A shared canonical frame enables consistent reasoning about object orientation across different instances. This is essential in applications like augmented reality [1], robotic manipulation [2, 3], category-level pose estimation [4], shape retrieval [5], and controlled 3D generation [6, 7]. However, establishing canonical frames has traditionally required substantial manual supervision, either via aligned CAD models [8, 9], controlled rendering of synthetic assets [10, 11], or careful real-world pose annotation [4, 12, 13]. This reliance on canonical supervision creates a scaling bottleneck: synthetic data introduces a domain gap, while annotated real data remains limited in category and instance diversity. Learning canonical object representations that generalize broadly across real-world data therefore remains an open challenge.

A natural question is whether canonical frames can emerge from self-supervised training on readily available data, following trends in representation learning where scale and data diversity drive generalization [14, 15]. Object-centric videos of rigid objects are abundant, and Structure-from-Motion (SfM) can recover camera poses and coarse geometry from them without manual annotation. However, SfM reconstructions are sequence-specific: each video yields internally consistent poses, but in an arbitrary coordinate system. Two videos of different cars may each produce accurate relative poses, yet their coordinate frames bear no relation to each other or to any shared notion of canonical orientation. Geometric registration (e.g., ICP) can align sufficiently similar shapes, but cannot establish shared notions of front, up, or left across objects with different geometry or topology. Learning a shared frame therefore requires a signal that transfers semantic orientation across sequences, which per-sequence geometry alone does not provide.

In this work, we show that a shared canonical frame can emerge from object-centric videos without canonical pose labels (see Figure 1). Our approach routes all training sequences through a shared geometric bottleneck: a coarse canonical mesh defined in a fixed coordinate frame. A correspondence network learns to map image pixels to vertices of this mesh, while estimated per-sequence alignments transform noisy SfM camera poses into the mesh's frame before each loss computation. SfM thus enforces multi-view consistency within each sequence, while the shared correspondence network transfers semantic orientation cues across sequences through visual appearance. Because the canonical mesh carries no category-specific detail, similar object parts across different instances consistently map to the same surface regions, and a common canonical frame emerges from multi-view consistency and the semantic priors of the feature extractor. At inference, our model takes a single RGB image and predicts the object's 3D rotation and scale-normalized translation in a canonical frame shared across all categories.

We demonstrate the utility of the learned correspondences on category-level pose estimation, achieving accuracy competitive with methods trained using canonical pose labels across multiple benchmarks. Our training uses $\sim$160k in-the-wild object-centric videos without category conditioning or canonical pose annotations, enabling generalization to object categories not seen during training (Section 4.5). In summary, our contributions are as follows:

- **Self-supervised canonical frame emergence.** We show that a shared canonical frame can emerge from object-centric videos using only noisy SfM camera poses, without any canonical pose annotation.
- **Geometric bottleneck.** We show that routing all training sequences through a shared, coarse canonical mesh induces cross-sequence consistency and enables a common reference frame to emerge through a combination of multi-view alignment and strong pre-trained features.
- **Category-agnostic generalization.** Without canonical pose supervision, our self-supervised approach achieves competitive performance with canonically supervised methods across multiple pose estimation benchmarks, including categories unseen during training.

2 Related Work

Canonical frames for pose supervision. Category-level pose estimation typically assumes a shared canonical object frame across instances. Wang *et al.* [4] introduced NOCS, which predicts normalized correspondences and recovers pose via Umeyama [16]. Subsequent work improves accuracy using RGB-D cues such as shape priors and deformation models [17, 18, 19], or stronger



Figure 2: **Qualitative pose predictions.** Predicted poses (3D rotation and scale-normalized translation) visualized as canonical axes overlaid on images from diverse categories and benchmarks. A single fixed rotation is applied uniformly to all predictions for visualization, without per-category or per-dataset adjustments. Semantic directions such as "front" and "up" of object instances are not explicitly supervised, but tend to consistently map onto the same axes in the learned canonical frame. We also show limitations discussed in Section 5: the laptop is articulated, so its predicted frame follows the dominant part; left and right shoes share one canonical frame, since chirality is not modeled.

architectures [20, 21]. RGB-only methods [22, 23, 24, 25, 26] operate on appearance alone, enabling shape-agnostic representations at the cost of metric accuracy. Neural mesh models [27, 28, 29] attach learned features to a geometric scaffold and recover pose by aligning rendered features to image observations, with extensions to full 6D and joint detection-and-pose [30, 25]. Recent open-world orientation estimation methods [31, 32] demonstrate impressive generalization across diverse categories, but still rely on canonical supervision through large-scale synthetic data with predefined canonical frames. Across these approaches, the dependence on canonical pose labels creates a scaling bottleneck: real-world datasets remain limited in category and instance diversity [4, 33, 34, 12, 13], and synthetic pipelines inherit a domain gap. We show that canonical frames can instead emerge from category-agnostic training on noisy in-the-wild data, without canonical pose labels or synthetic samples.

Novel object pose estimation without canonical frames. A separate line of work addresses pose estimation for objects unseen during training, but relies on additional object information at test time. Onboarding-based methods use CAD renderings [35, 36, 37] or multi-view captures [38, 39, 40, 41] to build object-specific templates or reconstructions, and then estimate pose by matching test images to these references. Reference-based methods instead predict relative transformations between a test image and a reference view, either directly [42, 43, 44] or via single-view 3D reconstructions [45, 46]. Recent methods such as MFOS [47] and BoxDreamer [48] also use a cuboid primitive for generalizable pose estimation, but work in the reference view setting rather than recovering a shared canonical frame. Both families avoid canonical supervision, but the resulting coordinate frame is instance- or reference-defined: two different chairs yield unrelated orientations that are not directly comparable across objects. We instead seek a shared canonical frame consistent across instances and categories, without canonical labels or test-time onboarding.

Emergence of canonical frames from object-centric videos. The most closely related line of work reduces supervision by exploiting multi-view geometry from object-centric videos. SHIC [49] predicts 2D-3D correspondences by conditioning features on detailed templates. UOP3D [50] shows that such templates can instead be reconstructed from video, removing the need for manual template design, and Common3D [51] extends this to category-specific morphable models learned entirely from object-centric sequences. While these methods progressively reduce supervision, they still require category labels and train separate models per category, introducing a scaling bottleneck as the number of categories grows. Canonical shape learning [52, 53] shows that consistent frames can emerge without supervision on 3D point clouds, but does not extend to RGB. Multi-Path Encoders [54] learn shared latent spaces from RGB without pose supervision, but scale linearly with object count.

Self-supervised correspondence learning has shown that vision foundation features can produce semantically meaningful correspondences without manual keypoint annotation [55], and recent work couples features to geometric primitives such as spheres to reduce ambiguities [56, 57]. While these methods share our use of a geometric bottleneck, they target pairwise keypoint matching rather than canonical pose recovery, which additionally requires a globally coherent 3D frame.

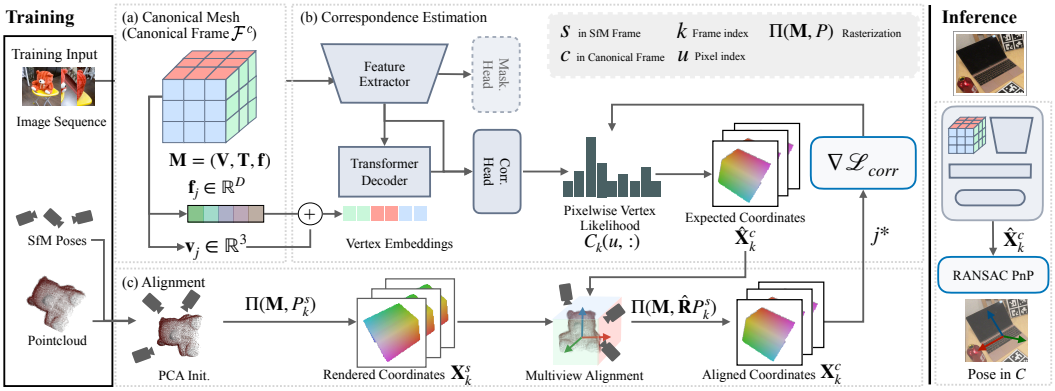


Figure 3: **Method overview.** (a) We define the shared geometric bottleneck as a canonical mesh in a fixed coordinate frame. (b) Each image is processed independently: the correspondence network predicts dense 2D–3D correspondences from pixels to mesh vertices. (c) During training, we estimate a per-sequence rotation that aligns each sequence's SfM frame to the canonical mesh frame. This alignment transforms SfM-derived correspondences into the canonical frame, producing pseudo-labels that supervise the correspondence network. During inference, correspondences are predicted from a single view, and the 3D rotation and scale-normalized translation are recovered via PnP.

Summary. Existing approaches either (a) assume canonical frames as supervision targets, (b) adopt instance-specific reference frames or test-time onboarding, or (c) use self-supervised training to recover canonical frames, but only within category-specific pipelines. Our work addresses the gap among these directions: a shared canonical frame emerges from self-supervised training on in-the-wild object-centric video through a geometric bottleneck, enabling category-agnostic pose estimation that transfers to unseen object categories.

3 Method

Our method learns dense correspondences from image pixels to the surface of a shared canonical mesh. Because this mesh is fixed and shared across all training sequences regardless of object category, it acts as a geometric bottleneck: consistent correspondences give rise to a shared canonical frame from which object pose can be recovered. We assume rigid objects throughout; articulated and deformable objects are outside the scope of this work.

Training relies on object-centric video sequences with SfM camera poses and off-the-shelf segmentation masks. No canonical pose annotations or category labels are used to condition our model. The method has three components: (a) the canonical mesh that defines the coordinate frame, (b) a correspondence network that maps image pixels to vertices of this mesh, and (c) a per-sequence alignment step that transforms SfM camera poses into the mesh's frame before loss computation. An overview is shown in Figure 3.

3.1 Canonical Mesh

The canonical mesh $\mathbf{M} = (\mathbf{V}, \mathbf{T}, \mathbf{f})$ is a coarse 3D shape with M vertices uniformly sampled on its surface and a set of triangular faces $\mathbf{T}$ (see Figure 3 (a)). Each vertex $\mathbf{v}_j \in \mathbf{V}$ carries a learnable embedding $\mathbf{f}_j \in \mathbb{R}^D$ that serves as a query in the correspondence network (Section 3.2). The mesh defines a fixed canonical frame $\mathcal{F}^c$: once the correspondence network maps object pixels to mesh vertices consistently, an object's orientation can be recovered from the resulting 2D-3D correspondences.

To generate supervision for the correspondence network during training, we need to know which 3D point on the mesh surface is visible at each pixel under a given camera pose. We obtain this by rasterizing the mesh under camera pose P (denoted $\Pi(\mathbf{M}, P)$) and interpolating vertex positions via barycentric coordinates over the mesh faces. This produces a dense coordinate map $\mathbf{X} \in \mathbb{R}^{N \times 3}$, where each visible pixel stores a 3D point on the mesh surface, along with a binary visibility mask.

The canonical mesh is a coordinate scaffold, not a geometric model of the object. It carries no category- or instance-specific geometric detail, so that the correspondence network must rely on appearance rather than shape matching. Consequently, the object-to-mesh mapping need not preserve topology or be bijective: a tire, for example, is mapped onto the mesh as a whole, and its hole is simply treated as part of the object. We use a unit cube as the default. Its faces, edges, and corners provide discrete orientation anchors, while its equal side lengths leave the three axes interchangeable, so that semantic axes must be assigned from visual evidence rather than from object aspect ratio. We compare against alternative primitives in Section 4.4.

3.2 Correspondence Prediction

Given an input image $\mathbf{I}$, the correspondence network predicts, for each pixel, a distribution over the M vertices of the canonical mesh (see Figure 3 (b)). From these distributions, we obtain dense 2D-3D correspondences between image pixels and 3D locations on the mesh surface.

A feature extractor produces dense features $\mathbf{F} \in \mathbb{R}^{N \times D}$, where N is the number of pixels and D is the feature dimension. To obtain per-vertex features, we use a transformer decoder that takes the pixel features $\mathbf{F}$ as keys and values, and learnable vertex embeddings $\mathbf{f} \in \mathbb{R}^{M \times D}$ as queries. Vertex positions $\mathbf{V} \in \mathbb{R}^{M \times 3}$ provide positional encoding for the queries, grounding each to a specific location on the canonical mesh. Given the decoded vertex features $\tilde{\mathbf{f}} \in \mathbb{R}^{M \times D}$, a soft correspondence matrix is computed as:

$$\mathbf{C} = \mathrm{softmax}\left(\frac{(\mathbf{F}\mathbf{W}_F)(\tilde{\mathbf{f}}\mathbf{W}_f)^\top}{\tau}\right) \in \mathbb{R}^{N \times M}, \tag{1}$$

where $\mathbf{W}_F, \mathbf{W}_f \in \mathbb{R}^{D \times D'}$ are learned projections and τ is a temperature parameter. Each row $\mathbf{C}(u, :)$ is a distribution over mesh vertices for pixel u, and the expected canonical 3D coordinate is:

$$\hat{\mathbf{x}}^c_u = \sum_{j=1}^{M} \mathbf{C}(u, j)\, \mathbf{v}_j \quad \text{or} \quad \hat{\mathbf{X}}^c = \mathbf{C}\mathbf{V} \in \mathbb{R}^{N \times 3}. \tag{2}$$

We frame correspondence as classification over discrete vertices rather than regression of continuous 3D coordinates; under a softmax, the cross-entropy objective concentrates probability on the target vertex while suppressing all others, encouraging discriminative correspondences.

3.3 Per-Sequence Alignment

Each SfM reconstruction lives in an arbitrary coordinate frame $\mathcal{F}^s$. To supervise the correspondence network, we need to express SfM camera poses in the canonical mesh frame $\mathcal{F}^c$. This requires estimating, for each training sequence, a rotation $\hat{\mathbf{R}}$ that maps $\mathcal{F}^s$ to $\mathcal{F}^c$. The process is illustrated in Figure 3 (c) and its effect on the pseudo-labels in Figure 5 (appendix).

We initialize each sequence's alignment via PCA on the masked SfM point cloud, which assigns consistent axes based on geometric extent (e.g., elongated objects share a dominant axis). This geometric prior captures coarse canonical structure but cannot resolve semantic orientation (e.g., front-vs-back) and fine-grained alignment. The per-sequence alignment refines this initialization during training using the correspondence network's predictions.

Concretely, for each of K sampled views, every pixel where the canonical mesh is visible provides a 3D–3D correspondence pair: one point from rasterizing the mesh under the SfM pose P^s_k (in $\mathcal{F}^s$), and one from the network's dense prediction $\hat{\mathbf{X}}^c_k = \mathbf{C}\mathbf{V}$ (in $\mathcal{F}^c$). Since all views in a sequence share the same SfM frame, we estimate a single rotation from these pairs jointly across views. Since the mesh is centered around the origin and at the same scale, we normalize each 3D vector to unit length and solve:

$$\hat{\mathbf{R}} = \underset{\mathbf{R} \in \mathrm{SO}(3)}{\arg\min} \sum_{k=1}^{K} \sum_{u=1}^{N_k} w_{ku} \left\| \hat{\mathbf{x}}^c_{k,u} - \mathbf{R}\, \mathbf{x}^s_{k,u} \right\|^2, \tag{3}$$

where $w_{ku} = H(\mathbf{C}_k(u, :))^{-1}$ with $H(\mathbf{p}) = -\sum_j p_j \log p_j$ is the inverse entropy of the correspondence distribution at pixel u, so that confident predictions contribute more to the alignment. We add a small constant ($\epsilon = 10^{-8}$) for numerical stability in the entropy computation and normalize weights

within each view to unit mean. This is Wahba's problem [58] and admits a closed-form solution via SVD, which we wrap in a standard RANSAC loop to reject outlier correspondences.

The estimated rotation transforms the SfM poses into the canonical frame: $P_k^c = \hat{\mathbf{R}} P_k^s$. These aligned poses are then used to rasterize the canonical mesh and generate pseudo-labels for the correspondence loss, as described in Section 3.4. Since the alignment depends on the network's current predictions, and the resulting pseudo-labels in turn update the network, both improve together during training.

3.4 Training Objective

Once the SfM poses have been transformed into the canonical frame, we use them to generate pseudo-labels for the correspondence network. For each view k, we rasterize the canonical mesh under the aligned pose P_k^c, producing a canonical 3D coordinate $\mathbf{x}_{k,u}^c$ for each visible pixel u. We assign each coordinate to its nearest mesh vertex:

$$j_{k,u}^* = \arg\min_j \|\mathbf{x}_{k,u}^c - \mathbf{v}_j\|_2, \tag{4}$$

and supervise the correspondence distribution with cross-entropy:

$$\mathcal{L}_{\text{corr}} = \sum_{k=1}^{K} \sum_{u=1}^{N_k} -\log \mathbf{C}_k(u, j_{k,u}^*). \tag{5}$$

This encourages the network to predict a distribution that concentrates on the mesh vertex visible at that location under the current alignment.

We additionally train a mask prediction head to identify which pixels belong to the object. During training, this head is supervised by the binary visibility mask obtained from rasterizing the canonical mesh under the estimated alignment (Section 3.3). At inference, the predicted mask determines the object region from which pose is recovered. The mask head is supervised using binary cross-entropy and Dice loss against the visibility mask $\mathbf{m}^c$ obtained from the same rasterization:

$$\mathcal{L}_{\text{mask}} = \text{BCE}(\hat{\mathbf{m}}, \mathbf{m}^c) + \text{Dice}(\hat{\mathbf{m}}, \mathbf{m}^c), \tag{6}$$

where $\hat{\mathbf{m}}$ is the predicted mask (boundary upweighting details in appendix). The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{corr}} + \lambda \mathcal{L}_{\text{mask}}, \tag{7}$$

where λ controls the relative contribution of the mask loss.

3.5 Inference

At test time, the correspondence network predicts $\mathbf{C}$ and the mask head identifies object pixels in a single forward pass. From the 2D-3D correspondences $\hat{\mathbf{X}}^c = \mathbf{CV}$, we recover the object's rotation and translation via EPnP [59] with RANSAC. No per-sequence alignment or PCA is involved at inference, since the correspondences directly yield a continuous pose in the canonical frame. Since metric scale is ambiguous from a single RGB image, translation is estimated only up to a global scale factor, matching the convention of prior RGB-only methods [26, 60, 25]. We therefore refer to the output as 3D rotation and scale-normalized translation rather than metric 6D pose.

4 Experiments

We evaluate our method on category-level pose estimation benchmarks, studying transfer performance, the effect of data scale, the role of design choices, and properties of the learned frame. Unless stated otherwise, we report rotation metrics since translation is recovered only up to scale and most baselines predict rotation only.

4.1 Experimental Setup

Training. We train on UCO3D [62], which provides ∼160k object-centric videos with SfM camera poses, sparse point clouds, segmentation masks, and optional PCA-based coordinate alignments.

Model	Can.	Train	REAL275		Omni6DPose		Objectron		PASCAL3D+		ImageNet3D		Avg.	
			Med↓	Acc30↑	Med↓	Acc30↑	Med↓	Acc30↑	Med↓	Acc30↑	Med↓	Acc30↑	Med↓	Acc30↑
QWEN3-VL[61]	✔	R.+S.	38.7	37.6	70.6	18.8	49.0	31.1	61.1	27.0	66.5	24.0	57.2	27.7
OriAny.V1[31]	✔	S.	28.4	52.2	62.8	36.7	18.4	60.4	18.1	71.0	29.7	50.3	31.5	54.1
OriAny.V1†[31]	✔	R.+S.	26.7	54.1	54.5	31.5	**15.1**	67.7	**15.7**	78.4	25.7	**55.6**	27.5	57.5
OriAny.V2†[32]	✔	R.+S.	**21.3**	57.0	**47.7**	**39.2**	19.8	66.2	17.5	76.5	28.1	54.1	26.9	58.6
Ours	✘	R.	21.8	**70.0**	49.2	35.2	15.3	**69.2**	**15.7**	**79.8**	**25.5**	55.0	**25.5**	**61.8**

Table 1: **Transfer across benchmarks.** Median rotation error and Acc30° on REAL275, Omni6DPose, Objectron, PASCAL3D+, and ImageNet3D. Avg. reports the macro-average over all five datasets. All methods are evaluated with the same symmetry-aware protocol and convention mapping discussed in Section 4.1. Despite not using canonical pose labels and training only on real video, our method achieves the best all-dataset average. Methods with a dagger (†) were trained on ImageNet3D (overlapping with PASCAL3D+) and are therefore given an unfair advantage on the gray metrics.

From each sequence, we use the masked point cloud (centered at the origin and normalized to unit scale), the segmentation masks for cropping, and the SfM camera poses. We optionally use the provided PCA alignment as initialization (Section 3.3). We do not use canonical pose annotations and train a single model across all categories without category conditioning.

Evaluation. We evaluate on REAL275 [4], Omni6DPose [63], PASCAL3D+ [12], ImageNet3D [13], and Objectron [64]. For category-constrained self-supervised comparisons, we use PASCAL3D+ and ObjectNet3D [65]. All methods assume single-object images, so for multi-object datasets we predict per-object pose from crops of the dataset-provided bounding boxes.

Baselines. For category-agnostic rotation estimation, we compare against OrientAnything V1/V2 [31, 32] and QWEN3-VL [61], all trained with canonical pose supervision (QWEN3-VL prompt details in appendix). For category-specific self-supervised comparisons, we use ZSP [66], UOP3D [50], and Common3D [51], which train separate models per category.

We report median geodesic rotation error (deg) and $Acc@30^\circ$ with symmetry-aware evaluation following [32]. Symmetry class labels for each category are derived automatically using an LLM (details in appendix), used only at evaluation time and are applied uniformly across all methods. In result tables, *Can.* indicates whether canonical pose annotations are used during training (✔/✘), and *Train* indicates whether the training data was real (R), synthetic (S), or mixed (R+S).

Our canonical frame can differ from a dataset's annotation convention by a fixed rotation. Recovering a particular benchmark convention is not our objective, since such conventions are arbitrary and can differ across datasets per category. For evaluation, we estimate this mapping per category via rotation averaging [67] on held-out data from the training split, never on test data, and apply it uniformly to all test predictions. In practice, this mapping is largely consistent across categories (see Figure 4b), and it reduces to selecting among discrete axis conventions (Section 4.5). Since such a post-hoc alignment is required whenever a method's convention differs from the dataset's, we apply the same mapping protocol to all baselines for a fair comparison.

Implementation details. We describe all implementational details including architecture, hyperparameters, and training schedule in the appendix.

4.2 Transfer Across Benchmarks

Table 1 compares our method to OrientAnything V1/V2 and QWEN3-VL across five benchmarks. We train once on UCO3D and evaluate on each benchmark without dataset-specific finetuning. Despite using no canonical pose labels and training only on real video, our method achieves the best average performance. The strongest results appear on Objectron and PASCAL3D+, where most categories have distinctive semantic axes that provide clear orientation signals from multi-view consistency. On REAL275, all methods benefit from limited pose variability in the dataset, but our method shows the largest margin in Acc@30°. On Omni6DPose, our performance falls behind OrientAnything V2, which we attribute to its large percentage of symmetric objects (see Section 4.5). Figure 2 shows qualitative pose predictions across diverse categories and benchmarks.

Model	Method	Training Videos	PASCAL3D+	ObjectNet3D	Mean
cat.-specific	ZSP [66]	50 seq. / cat.	46.0	42.2	44.1
	UOP3D [50]	50 seq. / cat.	69.2	52.4	60.8
	Common3D [51]	50 seq. / cat.	75.3	56.8	66.1
cat.-agnostic	$\text{Ours}^{\text{cat}}_{50}$	50 seq. / cat.	68.1	53.6	60.9
	$\text{Ours}^{\text{cat}}_{\text{full}}$	all seq. / cat.	<u>81.6</u>	<u>63.1</u>	<u>72.4</u>
	Ours	∼160k seq.	**85.9**	**70.0**	**77.9**

Table 2: **Effect of data scale.** Acc@30° on the 7-category PASCAL3D+ and 20-category ObjectNet3D subsets used by category-specific baselines. Note that these category subsets differ from the full-dataset evaluation in Table 1, which accounts for the different PASCAL3D+ scores.

				Objectron		**PASCAL3D+**		**ImageNet3D**	
Mesh	PCA	Align.	K	Med↓	Acc30↑	Med↓	Acc30↑	Med↓	Acc30↑
Cube	✔	✔	4	15.3	69.2	15.7	79.8	25.5	55.0
Cube	✘	✔	4	17.7	65.1	17.5	76.6	27.5	53.1
Cube	✔	✘	4	32.0	49.1	25.0	55.0	33.6	45.0
Rect. prism	✔	✔	4	17.0	66.8	15.4	79.5	28.5	52.0
Sphere	✔	✔	4	19.6	63.3	17.6	75.5	27.1	53.8
Cylinder	✔	✔	4	20.9	62.9	18.4	74.1	29.4	50.1
Sphere	✘	✔	4	21.4	62.2	23.1	72.4	29.7	50.4
Cube	✔	✔	2	18.2	64.4	19.6	72.5	27.0	53.9
Cube	✔	✔	1	57.3	32.5	53.0	28.0	50.9	33.9
Mapping: 24 axis conv.				15.3	69.1	15.7	79.8	25.7	54.9
Mapping: single global				22.7	55.3	15.6	77.5	31.5	49.0

Table 3: **Ablations.** Each row varies one design choice from the full model (first row). We ablate PCA initialization, learned alignment, mesh shape, and the number of training views K per sequence. The last two rows keep the full model and only restrict the evaluation-time convention mapping (Section 4.5).

4.3 Effect of Data Scale

Table 2 compares our category-agnostic model to self-supervised baselines that learn canonical frames without canonical pose labels but train *separate* models per category (ZSP/UOP3D/Common3D), using category supervision and relatively little per-category multi-view training data ($\leq$50 sequences per category). To separate method from data effects, we include two controls trained on the same categories: Ours^{cat}_{50} trains with the same 50-sequence cap, while Ours^{cat}_{full} trains on all available UCO3D sequences.

Our full model achieves the best average Acc@30°, improving over Common3D on both PASCAL3D+ (85.9% vs. 75.3%) and ObjectNet3D (70.0% vs. 56.8%). Notably, Ours^{cat}_{50} remains competitive even without a scale advantage, while scaling within the fixed categories yields substantial gains ($\text{Ours}^{cat}_{50} \rightarrow \text{Ours}^{cat}_{full}$). The strongest results are obtained with category-agnostic training (Ours), which highlights that data scale is critical to learn a generalizable representation.

4.4 Ablations

Table 3 ablates the canonical mesh shape, PCA initialization, the per-sequence alignment step, and the number of training views K per sequence, reporting transfer accuracy on three benchmarks.

PCA and learned alignment are complementary. PCA initialization without per-sequence alignment (row 3) maintains reasonable performance on ImageNet3D (45.0% vs. 55.0% for the full model), which contains a high proportion of symmetric objects. When we stratify by symmetry class, PCA-only actually improves on continuously symmetric objects (59.1% $\rightarrow$ 65.0% for $s{=}\infty$) while degrading severely on non-symmetric objects (53.6% $\rightarrow$ 41.7% for $s{=}1$). This reflects the

distinct roles of the two components: PCA captures geometric structure, such as elongation, flatness, spatial extent, which is often sufficient to determine orientation for symmetric objects, but cannot resolve semantic axes such as front-vs-back on objects where multiple orientations are geometrically equivalent. The learned alignment adds this semantic disambiguation through appearance-based correspondences (see Figure 5 for an example), producing substantial gains on PASCAL3D+ and Objectron ($55.0\% \rightarrow 79.8\%$, $49.1\% \rightarrow 69.2\%$), where most categories have distinctive semantic axes. Conversely, removing PCA while keeping learned alignment (row 2) causes smaller but consistent drops of 2–4 points across benchmarks, confirming that the geometric initialization aids convergence but is not the primary driver, and that the canonical frame emerges from the self-supervised alignment through the bottleneck.

Mesh shape matters, but the bottleneck principle is robust. Rows 4–7 compare the cube to a rectangular prism, a sphere, and a cylinder. The results suggest that the primitive must provide enough geometric structure to anchor orientation, while remaining neutral about which axis acquires which semantic meaning; the cube is the only tested primitive that satisfies both. The rectangular prism provides strong orientation structure, but its unequal side lengths prescribe long, medium, and short axes, encouraging an extent-based convention that does not transfer consistently across categories. The sphere is axis-neutral but its continuous symmetry provides no orientation anchors, and the cylinder is both axis-biased and continuously ambiguous around its principal axis. On the heterogeneous ImageNet3D, the sphere even outperforms the prism (53.8% vs. 52.0%), suggesting that avoiding a misleading axis prior can be more valuable than stronger geometric structure. All primitives nevertheless remain reasonably effective, indicating that the geometric bottleneck concept is not tied to a specific shape.

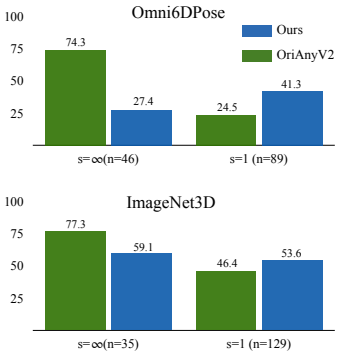
Multi-view coverage matters. Reducing the number of training views per sequence from $K{=}4$ to $K{=}2$ causes moderate drops across benchmarks (e.g., $79.8\% \rightarrow 72.5\%$ on PASCAL3D+), while $K{=}1$ leads to severe degradation ($79.8\% \rightarrow 28.0\%$). With a single view, the per-sequence alignment becomes underconstrained and causes mode collapse as a trivial solution. Two views recover much of the performance, but four views provide the multi-view diversity needed for reliable alignment.

4.5 Analysis: Frame Consistency, Unseen Categories, and Symmetry

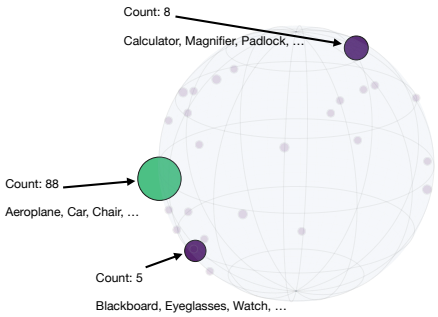
Frame consistency. To assess whether the emerged canonical frame is shared across categories, we estimate per-category convention mappings on ImageNet3D and visualize their agreement. For each category c, we compute the mapping $\hat{\mathbf{A}}_c$ (Appendix C) and apply it to a fixed reference vector $\mathbf{v} = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})^\top$, plotting the resulting direction $\hat{\mathbf{A}}_c\mathbf{v}$ on the unit sphere. If the learned frame is consistent, these directions should cluster into a small number of modes. Figure 4b shows this visualization restricted to non-symmetric categories ($s{=}1$), removing ambiguity due to rotational symmetry. The dominant mode contains 88 of 129 non-symmetric categories, indicating that the learned frame is largely shared across diverse object types. The remaining modes are small and largely correspond to categories with geometrically ambiguous structure (e.g., watches, padlocks), where multiple axis assignments are plausible.

Reliability of the convention mapping. To test how much work the evaluation mapping does, we restrict each per-category mapping from an unrestricted rotation to one of the 24 proper rotational symmetries of the cube, i.e., handedness-preserving permutations and sign changes of the coordinate axes, which cannot absorb fine-grained per-category errors (Table 3, last two rows). This changes accuracy by at most 0.1 points: the mapping only selects among discrete axis conventions rather than compensating for inconsistent predictions. A single global rotation shared by all categories is stricter and performs worse, because a few categories are internally consistent but disagree with the dataset's choice of semantic axes (e.g., books and laptops on Objectron, dining tables on PASCAL3D+). Per-category results are provided in the appendix.

Unseen categories. To isolate category-level novelty, we evaluate on the 32 ImageNet3D categories with no exact or synonymous match in the 1,067-category UCO3D vocabulary (4,648 images; e.g., jinrikisha, ferris wheel, hovercraft). Our method reaches 52.0% Acc@30° on these categories, compared to 55.5% on the remaining 157. Both groups have a similar symmetry composition ($22/66/13/0\%$ vs. $18/69/11/2\%$ for $s{=}\infty/1/2/4$), so the 3.5-point gap reflects category novelty rather than a harder symmetry mix. While we cannot verify that the pretrained backbone has never encountered these categories, the result supports generalization beyond the training vocabulary.



(a) **Symmetry-stratified accuracy.** Acc30° by symmetry class s on Omni6DPose (top) and ImageNet3D (bottom).



(b) **Convention mappings cluster across categories.** Per-category convention mappings (estimated via rotation averaging) concentrate into a few modes, indicating that the learned frame largely agrees with the conventions in ImageNet3D, up to a single rotation.

Figure 4: **Analysis of symmetry and convention mapping.** Our main performance gap concentrates on symmetry-heavy objects (left), while per-category convention mappings cluster into a small number of modes (right), supporting that the emerged canonical frame is shared across categories.

Symmetry. The emerged frame is most consistent for objects with distinctive semantic axes, where appearance and multi-view constraints provide clear orientation signals. For a symmetric object, the reference frame is only defined up to its symmetry group, and any element of that group is a valid choice. At evaluation, we account for this by minimizing the error over symmetry-equivalent ground-truth rotations. During training, however, symmetry is not modeled: the alignment (Eq. 3) selects one valid element of the symmetry group, and the multi-view loss then enforces this single correspondence assignment even though several are equally valid. The resulting pseudo-labels are geometrically correct but not predictable from appearance, since symmetry-equivalent views are virtually indistinguishable. Methods supervised with canonical poses do not suffer from this problem, making it significantly easier to learn robust defaults for symmetric objects. Figure 4a confirms this: our method falls behind on continuously symmetric objects ($s{=}\infty$) but shows consistently stronger performance on non-symmetric objects ($s{=}1$). Optimizing over symmetry-equivalent alignments is not a trivial extension: symmetry is only definable relative to a stable frame, which is what training aims to establish, and unconstrained self-supervised symmetry prediction trivially minimizes the loss by declaring an overly broad symmetry group.

5 Conclusion

We showed that a shared canonical frame can emerge from object-centric video of rigid objects without canonical pose labels or category conditioning, by routing all training through a shared geometric bottleneck. Across multiple benchmarks, the resulting frame enables competitive pose estimation compared to methods trained with canonical supervision, generalizes to categories outside the training vocabulary, and improves with data volume and category diversity. Our analysis shows that the emerged frame is largely consistent across categories, and that the evaluation mapping to benchmark conventions reduces to selecting discrete axis conventions.

Limitations. Our formulation assumes rigid objects: articulated and deformable objects would require a representation beyond a single canonical frame, and for articulated instances (e.g., laptops at different hinge angles) the predicted frame follows whichever part dominates the correspondence evidence. Chirality is not modeled, so mirrored instances such as left and right shoes share the same canonical frame, consistent with the conventions of the evaluated benchmarks. Translation is recovered only up to scale, and accuracy drops sharply once more than roughly half of the object is occluded (Appendix F). Finally, symmetry is handled at evaluation but not during training, which causes our main performance gap on symmetric objects. Jointly learning the canonical frame and the symmetry group without canonical pose or symmetry labels remains the principal open problem left by this work.

Acknowledgments and Disclosure of Funding

AK acknowledges support via his Emmy Noether Research Group funded by the German Research Foundation (DFG) under grant number 468670075. TF gratefully acknowledges the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the BayernKI project ***v114be***. BayernKI funding is provided by Bavarian state authorities.

References

[1] P. Aguirrezabal, I. Aguinaga, and A. Alvarez-Gila, "Monocular rgb 6d object pose estimation for augmented reality: a survey," *Virtual Reality*, 2026.

[2] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6394–6400, IEEE, 2022.

[3] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong, "Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17359–17369, 2025.

[4] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *CVPR*, pp. 2642–2651, 2019.

[5] J. Tangelder and R. Veltkamp, "A survey of content based 3d shape retrieval methods," *Proceedings Shape Modeling Applications, 2004.*, pp. 145–156, 2004.

[6] Y. Lu, Y. Tian, Z. Jiang, Y. Zhao, Y. Yang, H. Ouyang, H. Hu, H. Yu, Y. Shen, and Y. Liao, "Orientation matters: Making 3d generative models orientation-aligned," *arXiv preprint arXiv:2506.08640*, 2025.

[7] R. Parihar, V. Agrawal, S. VS, and V. B. Radhakrishnan, "Compass control: Multi object orientation control for text-to-image generation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2791–2801, 2025.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[9] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023.

[10] M. Zhang, T. Wu, T. Wang, T. Wang, Z. Liu, and D. Lin, "Omni6d: Large-vocabulary 3d object dataset for category-level 6d object pose estimation," in *European Conference on Computer Vision*, pp. 216–232, Springer, 2024.

[11] J. Cai, Y. He, W. Yuan, S. Zhu, Z. Dong, L. Bo, and Q. Chen, "Ov9d: Open-vocabulary category-level 9d object pose and size estimation," *arXiv preprint arXiv:2403.12396*, 2024.

[12] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE winter conference on applications of computer vision*, pp. 75–82, IEEE, 2014.

[13] W. Ma, G. Zhang, Q. Liu, G. Zeng, A. Kortylewski, Y. Liu, and A. Yuille, "Imagenet3d: Towards general-purpose object-level 3d understanding," *Advances in Neural Information Processing Systems*, vol. 37, pp. 96127–96149, 2024.

[14] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.

[16] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[17] K. Chen and Q. Dou, "Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation," in *ICCV*, pp. 2773–2782, 2021.

[18] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, and F. 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.

[19] J. Lin, Z. Wei, C. Ding, and K. Jia, "Category-level 6d object pose and size estimation using self-supervised deep prior deformation networks," in *European Conference on Computer Vision*, pp. 19–34, Springer, 2022.

[20] J. Liu, Y. Chen, X. Ye, and X. 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.

[21] Y. Chen, Y. Di, G. Zhai, F. Manhardt, C. Zhang, R. Zhang, F. Tombari, N. Navab, and B. Busam, "Secondpose: Se (3)-consistent dual-stream feature fusion for category-level pose estimation," in *CVPR*, pp. 9959–9969, 2024.

[22] Z. Fan, Z. Song, J. Xu, Z. Wang, K. Wu, H. Liu, and J. He, "Object level depth reconstruction for category level 6d object pose estimation from monocular rgb image," in *European Conference on Computer Vision*, pp. 220–236, Springer, 2022.

[23] T. Lee, B.-U. Lee, M. Kim, and I. S. Kweon, "Category-level metric scale object shape and pose estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8575–8582, 2021.

[24] J. Wei, X. Song, W. Liu, L. Kneip, H. Li, and P. Ji, "Rgb-based category-level object pose estimation via decoupled metric scale recovery," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2036–2042, IEEE, 2024.

[25] T. Fischer, X. Zhang, and E. Ilg, "Unified category-level object detection and pose estimation from rgb images using 3d prototypes," in *ICCV*, pp. 9790–9800, 2025.

[26] R. Zhang, Z. Huang, G. Wang, C. Zhang, Y. Di, X. Zuo, J. Tang, and X. Ji, "Lapose: Laplacian mixture shape modeling for rgb-based category-level object pose estimation," in *European Conference on Computer Vision*, pp. 467–484, Springer, 2024.

[27] A. Wang, A. Kortylewski, and A. Yuille, "Nemo: Neural mesh models of contrastive features for robust 3d pose estimation," in *ICLR*, 2021.

[28] T. Fischer, Y. Liu, A. Jesslen, N. Ahmed, P. Kaushik, A. Wang, A. Yuille, A. Kortylewski, and E. Ilg, "inemo: Incremental neural mesh models for robust class-incremental learning," in *ECCV*, 2024.

[29] A. Jesslen, G. Zhang, A. Wang, W. Ma, A. Yuille, and A. Kortylewski, "Novum: Neural object volumes for robust object classification," in *European Conference on Computer Vision*, pp. 264–281, Springer, 2025.

[30] W. Ma, A. Wang, A. Yuille, and A. Kortylewski, "Robust category-level 6d pose estimation with coarse-to-fine rendering of neural features," in *ECCV*, pp. 492–508, Springer, 2022.

[31] Z. Wang, Z. Zhang, T. Pang, C. Du, H. Zhao, and Z. Zhao, "Orient anything: Learning robust object orientation estimation from rendering 3d models," *arXiv preprint arXiv:2412.18605*, 2024.

[32] Z. Wang, Z. Zhang, J. Xu, J. Wang, T. Pang, C. Du, H. Zhao, and Z. Zhao, "Orient anything v2: Unifying orientation and rotation understanding," *arXiv preprint arXiv:2601.05573*, 2026.

[33] Y. Fu and X. Wang, "Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27469–27483, 2022.

[34] H. Jung, S.-C. Wu, P. Ruhkamp, G. Zhai, H. Schieber, G. Rizzoli, P. Wang, H. Zhao, L. Garattoni, S. 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.

[35] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan, "Foundpose: Unseen object pose estimation with foundation features," in *ECCV*, pp. 163–182, Springer, 2024.

[36] V. N. Nguyen, T. Groueix, M. Salzmann, and V. 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 (CVPR)*, pp. 9903–9913, June 2024.

[37] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, "Megapose: 6d pose estimation of novel objects via render & compare," *arXiv preprint arXiv:2212.06870*, 2022.

[38] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, pp. 17868–17879, 2024.

[39] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. 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.

[40] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou, "Onepose++: Keypoint-free one-shot object pose estimation without cad models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35103–35115, 2022.

[41] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang, "Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images," in *European Conference on Computer Vision*, pp. 298–315, Springer, 2022.

[42] X. Liu, G. Wang, R. Zhang, C. Zhang, F. Tombari, and X. Ji, "Unopose: Unseen object pose estimation with an unposed rgb-d reference image," in *CVPR*, pp. 22023–22034, 2025.

[43] M. Liu, S. Li, A. Chhatkuli, P. Truong, L. Van Gool, and F. Tombari, "One2any: One-reference 6d pose estimation for any object," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6457–6467, 2025.

[44] L. Kuang, Y. Velikova, M. Saleh, J.-N. Zaech, D. P. Paudel, and B. Busam, "Conceptpose: Training-free zero-shot object pose estimation using concept vectors," *arXiv preprint arXiv:2512.09056*, 2025.

[45] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon, "Any6d: Model-free 6d pose estimation of novel objects," in *CVPR*, pp. 11633–11643, 2025.

[46] Z. Geng, N. Wang, S. Xu, C. Ye, B. Li, Z. Chen, S. Peng, and H. Zhao, "One view, many worlds: Single-image to 3d object meets generative domain randomization for one-shot 6d pose estimation," in *Conference on Robot Learning*, pp. 168–197, PMLR, 2025.

[47] J. Lee, Y. Cabon, R. Brégier, S. Yoo, and J. Revaud, "Mfos: Model-free & one-shot object pose estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 2911–2919, 2024.

[48] Y. Yu, X. He, C. Zhao, J. Yu, J. Yang, R. Hu, Y. Shen, X. Zhu, X. Zhou, and S. Peng, "Boxdreamer: Dreaming box corners for generalizable object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9374–9384, 2025.

[49] A. Shtedritski, C. Rupprecht, and A. Vedaldi, "Shic: Shape-image correspondences with no keypoint supervision," in *European Conference on Computer Vision*, pp. 129–145, Springer, 2024.

[50] L. Sommer, A. Jesslen, E. Ilg, and A. Kortylewski, "Unsupervised learning of category-level 3d pose from object-centric videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22787–22796, 2024.

[51] L. Sommer, O. Dünkel, C. Theobalt, and A. Kortylewski, "Common3d: Self-supervised learning of 3d morphable models for common objects in neural feature space," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6468–6479, 2025.

[52] X. Li, Y. Weng, L. Yi, L. Guibas, A. L. Abbott, S. Song, and H. Wang, "Leveraging se(3) equivariance for self-supervised category-level object pose estimation," *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[53] W. Sun, A. Tagliasacchi, B. Deng, S. Sabour, S. Yazdani, G. E. Hinton, and K. M. Yi, "Canonical capsules: Self-supervised capsules in canonical pose," *Advances in Neural information processing systems*, vol. 34, pp. 24993–25005, 2021.

[54] M. Sundermeyer, M. Durner, E. Y. Puang, Z.-C. Marton, N. Vaskevicius, K. O. Arras, and R. Triebel, "Multi-path learning for object pose estimation across domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13916–13925, 2020.

[55] J. Zhang, C. Herrmann, J. Hur, L. Polania Cabrera, V. Jampani, D. Sun, and M.-H. Yang, "A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence," *Advances in Neural Information Processing Systems*, vol. 36, pp. 45533–45547, 2023.

[56] O. Mariotti, O. Mac Aodha, and H. Bilen, "Improving semantic correspondence with viewpoint-guided spherical maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19521–19530, 2024.

[57] O. Dünkel, T. Wimmer, C. Theobalt, C. Rupprecht, and A. Kortylewski, "Do it yourself: Learning semantic correspondence from pseudo-labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5834–5844, October 2025.

[58] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[59] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," *International journal of computer vision*, vol. 81, pp. 155–166, 2009.

[60] Z. Huang, G. Wang, C. Zhang, R. Zhang, X. Li, and X. Ji, "Givepose: Gradual intra-class variation elimination for rgb-based category-level object pose estimation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22055–22066, 2025.

[61] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[62] X. Liu, P. Tayal, J. Wang, J. Zarzar, T. Monnier, K. Tertikas, J. Duan, A. Toisoul, J. Y. Zhang, N. Neverova, *et al.*, "Uncommon objects in 3d," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14102–14113, 2025.

[63] J. Zhang, W. Huang, B. Peng, M. Wu, F. Hu, Z. Chen, B. Zhao, and H. Dong, "Omni6dpose: A benchmark and model for universal 6d object pose estimation and tracking," in *European Conference on Computer Vision*, pp. 199–216, Springer, 2024.

[64] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7822–7831, 2021.

[65] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese, "Objectnet3d: A large scale database for 3d object recognition," in *European conference on computer vision*, pp. 160–176, Springer, 2016.

[66] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner, "Zero-shot category-level object pose estimation," in *European Conference on Computer Vision*, pp. 516–532, Springer, 2022.

[67] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International journal of computer vision*, vol. 103, no. 3, pp. 267–305, 2013.

[68] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," *ACM Transactions on Graphics*, vol. 39, no. 6, 2020.

[69] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, *et al.*, "Dinov3," *arXiv preprint arXiv:2508.10104*, 2025.

[70] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR*, 2022.

[71] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

[72] V. Larsson and contributors, "PoseLib - Minimal Solvers for Camera Pose Estimation," 2020.

[73] A. Wang, Y. Sun, A. Kortylewski, and A. L. Yuille, "Robust object detection under occlusion with context-aware compositionalnets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

A Overview

This supplement provides additional details and analyses referenced in the main paper:

- Implementation details and training hyperparameters (B)
- Evaluation protocol, including symmetry-aware metrics and convention mapping (C)
- Visualization of the learned per-sequence alignment (D)
- Reliability of the convention mapping, with per-category results (E)
- Robustness to occlusion (F)
- Full symmetry-class breakdown (G)
- Mesh-resolution runtime trade-off (H)
- PASCAL3D+ and ObjectNet3D per-category results (I)
- Qualitative results (J)
- LLM-derived symmetry labels (K)

B Implementation Details

B.1 Data and Preprocessing

We train on the train split of UCO3D [62], which provides object-centric video sequences with RGB frames, segmentation masks, SfM camera poses, and sparse point clouds. We undistort images using the provided camera parameters. Per sequence, we apply the dataset's PCA alignment and normalize the object point cloud using a robust bounding box (10th/90th coordinate quantiles) to fit the unit canonical mesh. For details about the alignment, we refer the reader to the supplemental material of [62]. During training, we sample $K{=}4$ frames per sequence using stratified temporal sampling to encourage viewpoint diversity.

B.2 Data Augmentation

We apply photometric augmentation throughout training: color jitter (brightness 0.4, contrast 0.4, saturation 0.2, hue 0.1) with probability 0.8, random grayscale with probability 0.2, and Gaussian blur with probability 0.4. Images are cropped to the object mask with 10% padding and resized to $256{\times}256$ with aspect-ratio-preserving padding, followed by ImageNet normalization. After iteration $7,500$ we enable geometric augmentation: the mask-crop padding is randomly sampled from $[0.1, 0.25]$, random recropping is applied with probability 0.7, in-plane rotation up to 30° with probability 0.3, and random patch occlusion (1–2 patches, $5-30\%$ of the crop each) with probability 0.5. All geometric transforms are accompanied by consistent updates to camera intrinsics and extrinsics to preserve multi-view consistency. The staged schedule avoids destabilizing the co-evolutionary alignment loop during early training, where mode collapse could rarely occur.

B.3 Canonical Mesh

We use a cube mesh of unit scale with $M{=}1016$ vertices and 2028 triangular faces. The mesh is rasterized at $64{\times}64$ resolution using nvdiffrast [68], producing per-pixel object coordinates and a visibility mask for the correspondence loss. For the primitive ablation (Table 3), the rectangular prism, sphere, and cylinder use approximately $1{,}000$ evenly sampled vertices and watertight triangulations, and are trained with the same protocol as the cube.

B.4 Network Architecture

The visual backbone is DINOv3 ViT-L/16 [69] with LoRA [70] (rank 8, $\alpha{=}8$, dropout 0.1) inserted into the fused query/value projections of every transformer block and the remaining backbone parameters are frozen. Features from layers 6, 14, 18, and 23 are aggregated into a feature pyramid and decoded with a UPerNet-style decoder [71] (512 channels) to produce a $64{\times}64$ feature map. A 6-layer transformer decoder (hidden size 512, 8 heads, feed-forward dimension 2048) decodes one

query per mesh vertex. The correspondence head projects both mesh queries and image features to normalized 512-dimensional descriptors and computes vertex-pixel logits via dot product. A separate convolutional head predicts the foreground mask.

B.5 Training

We train for 15 epochs ($\sim$75k iterations) on four H100 GPUs with batch size 8 sequences (effective 32) and mixed precision. We use AdamW with weight decay 0.05 and gradient clipping at norm 1.0. The learning rate is 10^{-4} for non-backbone parameters and 10^{-5} for LoRA parameters. The correspondence softmax temperature τ follows a three-stage cosine schedule: $0.20 \to 0.17$ over the first 30% of training, $0.17 \to 0.13$ over the next 50%, and $0.13 \to 0.10$ over the final 20%.

B.6 Robust Per-Sequence Alignment

The per-sequence rotation is estimated using a robust Wahba solver with RANSAC (100 iterations, 4-point samples, 25° inlier threshold).

B.7 Loss

The total loss is the per-frame averaged correspondence cross-entropy plus the mask loss (λ=1.0). The mask loss combines BCE and Dice loss with boundary-focused weighting: pixels within a 2-pixel band of the mask boundary receive full weight, while non-boundary pixels are downweighted by a factor of 0.1. Padded and synthetically occluded regions are excluded from the mask loss.

B.8 Inference

We use PoseLib [72] for PnP pose recovery with default settings and a maximum reprojection error threshold of 4.0 pixels.

C Evaluation Protocol

We evaluate category-level orientation by aligning each predicted pose to the target dataset's coordinate convention and reporting geodesic rotation error and threshold accuracies.

Symmetry-aware rotation error. We measure angular error with the geodesic distance on $\mathrm{SO}(3)$:

$$d(\mathbf{R}^{\text{gt}}, \mathbf{R}^{\text{pred}}) = \arccos\Big(\tfrac{\mathrm{tr}(\mathbf{R}^{\text{gt}\top}\mathbf{R}^{\text{pred}})-1}{2}\Big), \tag{8}$$

reported in degrees. To account for rotational symmetries about the upright axis, we use the category-specific symmetry label $s \in \{\infty, 1, 2, 4\}$ (continuous / none / 180° / 90° ambiguity) and report the minimum error over the symmetry group:

$$d_{\text{sym}}(\mathbf{R}^{\text{gt}}, \mathbf{R}^{\text{pred}}) = \min_{\mathbf{S}\in\mathcal{S}(s)} d(\mathbf{R}^{\text{gt}}, \mathbf{S}\mathbf{R}^{\text{pred}}), \tag{9}$$

where $\mathcal{S}(1) = \{\mathbf{I}\}$, $\mathcal{S}(2) = \{\mathbf{R}_y(0), \mathbf{R}_y(\pi)\}$, $\mathcal{S}(4) = \{\mathbf{R}_y(k\tfrac{\pi}{2})\}_{k=0}^{3}$, and $\mathbf{R}_y(\theta)$ denotes a rotation around the y-axis. For continuous symmetry (s=∞), yaw is unrecoverable and we compare only the transformed upright axes, i.e., the angle between $\mathbf{R}^{\text{gt}}\mathbf{y}$ and $\mathbf{R}^{\text{pred}}\mathbf{y}$ with $\mathbf{y} = (0, 1, 0)^\top$.

Per-image inference and pose recovery. For each test image, the model predicts dense 2D–3D correspondences from pixels to canonical mesh vertices. We recover the 3D rotation and scale-normalized translation via EPnP [59] with RANSAC using dataset-provided camera intrinsics. This yields a predicted rotation $\mathbf{R}^{\text{pred}}$ and scale-normalized translation $\mathbf{t}^{\text{pred}}$ expressed in the learned canonical frame.

Convention mapping. Canonical frames are defined only up to a global rotation, and different benchmarks use different coordinate conventions. We estimate a per-category convention mapping $\hat{\mathbf{A}}_c$ that aligns our predictions to the dataset convention via rotation averaging [67]:

$$\hat{\mathbf{A}}_c = \operatorname*{arg\,min}_{\mathbf{A}\in\mathrm{SO}(3)} \sum_{i\in c} \big\|\mathbf{R}_i^{\text{gt}} - \mathbf{A}\,\mathbf{R}_i^{\text{pred}}\big\|_F^2, \tag{10}$$

i.e., $\hat{\mathbf{A}}_c$ is the chordal L_2 mean of the relative rotations $\mathbf{R}_i^{\text{gt}}\mathbf{R}_i^{\text{pred}\top}$, which admits a closed-form SVD solution with the $\det(\hat{\mathbf{A}}_c)=+1$ constraint. This mapping is estimated on the training split of each dataset and applied uniformly to all test predictions. As shown in the main paper (Figure 4b), these per-category mappings cluster into a small number of modes, indicating that the learned frame is largely shared across categories. Section E shows that the mapping only selects among discrete axis conventions and does not compensate for fine-grained per-category errors.

Qwen3-VL baseline. For the QWEN3-VL [61] baseline, we prompt the model with the input image and request an orientation estimate as rotation angles. The prompt was taken from the official GitHub repository of the QWEN Team. We additionally provided the category label in the prompt and bounding box location in the output during our experiments, since we found that this yielded noticeably better results and is likely due to the training paradigm of the model. The prompt template is:

> Find the category in this image. Provide its 3D bounding box. The output format required is JSON: ’‘["bbox_3d":[x_center, y_center, z_center, x_size, y_size, z_size, roll, pitch, yaw]]‘.’

D Visualization of the Per-Sequence Alignment

Figure 5 visualizes the pseudo-labels of the per-sequence alignment (Section 3.3) for four UCO3D training sequences. For each sequence, we render the canonical coordinate map under the PCA alignment provided with the dataset, which serves as initialization, and under the learned alignment $\hat{\mathbf{R}}$, which yields the final pseudo-labels. We also show the correspondences predicted by the trained network. The learned alignment replaces the extent-based PCA axes with a rotation estimated from the network’s predictions (Eq. 3), and, as a single rotation per sequence, is applied consistently to all frames.

The colors make the effect on semantic orientation visible. Under the PCA initialization, the red corner of the cube marks a different part of each object: front-right-bottom for the helicopter and the mouse, but back-left-bottom for the cap and the truck. After alignment, it marks the back-right-top of all four objects. The learned alignment thus resolves the front-back and left-right ambiguities left by PCA and places these objects, from four different categories, in a common semantic frame.

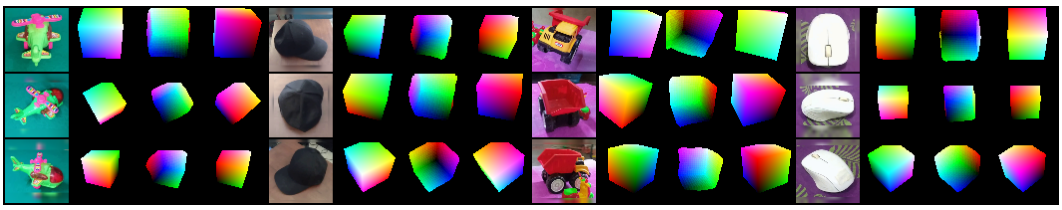


Figure 5: **From PCA initialization to learned alignment.** Four UCO3D training sequences (helicopter, cap, truck, mouse), each shown for three frames (rows). Columns per sequence: input frame, coordinate map under the PCA initialization, correspondences $\hat{\mathbf{X}}^c$ predicted by the trained network, and coordinate map under the learned per-sequence alignment $\hat{\mathbf{R}}$ (final pseudo-label). Colors encode canonical 3D coordinates. Under PCA, the red corner marks different object parts across sequences (e.g., front-right-bottom for the helicopter, back-left-bottom for the cap). After alignment, it marks the back-right-top of all four objects.

E Reliability of the Convention Mapping

All results use the per-category mapping $\hat{\mathbf{A}}_c$ of Eq. (10). An unrestricted rotation could, in principle, absorb category-specific prediction errors. To test this, we compare three variants, all estimated on the training split and never on test data:

- **Unrestricted (per category):** $\hat{\mathbf{A}}_c \in \mathrm{SO}(3)$ as in Eq. (10). This is the default used throughout the paper.

- **24 axis conventions (per category):** $\hat{\mathbf{A}}_c$ is restricted to the 24 proper rotational symmetries of the cube, i.e., handedness-preserving permutations and sign changes of the coordinate axes. We select the element that minimizes the objective of Eq. (10).
- **Single global rotation:** one $\hat{\mathbf{A}}$ shared by all categories of a dataset, obtained by pooling all training samples in Eq. (10).

The aggregate results are reported in the main paper (Table 3, last two rows). The restricted variant cannot correct fine-grained errors, yet it changes Acc@30° by at most 0.1 points. The mapping therefore only selects among discrete axis conventions.

Table 4 shows the per-category results on Objectron and PASCAL3D+. The single global rotation degrades on only a few categories. These categories are predicted consistently in our frame, but their semantic axes are assigned differently by the dataset convention (books and laptops on Objectron, dining tables on PASCAL3D+).

Objectron										
Mapping	bike	book	bottle	camera	cereal box	chair	cup	laptop	shoe	Mean
Unrestricted	98.1	51.3	45.3	75.5	66.3	87.0	37.7	80.5	81.7	69.2
24 axis conv.	98.1	51.3	45.3	75.5	66.3	87.0	37.7	80.5	81.0	69.1
Single global	98.1	1.5	45.3	75.5	66.3	87.0	37.7	6.4	81.0	55.3

PASCAL3D+													
Mapping	aero	bicycle	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Mean
Unrestricted	79.1	80.0	82.3	66.8	97.7	96.7	95.6	47.7	81.5	84.6	80.9	65.1	79.8
24 axis conv.	79.1	80.0	82.3	66.8	97.7	96.7	95.6	47.7	81.5	84.6	80.9	65.1	79.8
Single global	79.1	80.0	82.3	66.8	97.7	96.7	95.6	20.7	81.5	84.6	80.9	65.1	77.5

Table 4: **Per-category convention-mapping results** (Acc@30°). Restricting the per-category mapping to the 24 axis conventions of the cube leaves accuracy virtually unchanged, so the mapping does not absorb per-category errors. A single global rotation affects only a few categories whose semantic axes are defined differently by the dataset.

F Robustness to Occlusion

UCO3D contains little natural occlusion, so we apply random patch occlusion to the input images during training. We evaluate robustness at inference in two settings: realistic occluders from OccludedPASCAL3D+, and a controlled occluder whose size we vary. In both settings, all methods use the convention mapping and evaluation protocol of the main paper.

F.1 OccludedPASCAL3D+

OccludedPASCAL3D+ [73] places occluders on PASCAL3D+ images at three levels: L1 (20–40%), L2 (40–60%), and L3 (60–80% of the object occluded). L0 refers to the original, unoccluded images. Table 5 shows the results. Our method is strongest at L0 and L1, and within 0.5 points of the best result at L2, but falls behind at L3. Resolution and training data both differ between methods, so this comparison is not controlled for either. OrientAnything V2 uses 518×518 inputs, while ours are 256×256. The daggered models were trained on ImageNet3D, which overlaps with PASCAL3D+. We do not claim that these factors fully account for the L3 gap, but they are relevant for interpreting it. We attribute our robustness at moderate occlusion to patch-occlusion augmentation during training and RANSAC at inference, which rejects correspondences predicted on occluded pixels.

F.2 Controlled Occlusion Sweep

To measure how accuracy degrades with the occluded fraction rather than in coarse bins, we place a single black rectangular occluder inside the ground-truth 2D bounding box of each PASCAL3D+

Method	L0	L1	L2	L3
OriAny.V1 [31]	71.0	66.9	61.3	48.0
OriAny.V1† [31]	78.4	78.0	**75.9**	64.5
OriAny.V2† [32]	76.5	75.0	74.0	**70.6**
Ours	**79.8**	**79.5**	75.4	63.0

Table 5: **Occlusion robustness on OccludedPASCAL3D+** (Acc@30°). L0 denotes the unoccluded images; L1–L3 correspond to 20–40%, 40–60%, and 60–80% occlusion. Daggered methods were trained on ImageNet3D (overlapping with PASCAL3D+, gray). Each method is evaluated at its standard input resolution.

object and increase the fraction of the box it covers from 0% to 80% in steps of 10%. The model, convention mapping, and evaluation protocol match the main paper.

Figure 6 shows the results. Up to $\sim 40\%$ coverage, accuracy degrades gracefully ($79.8\% \rightarrow 74.8\%$). Beyond 50%, it drops sharply (70.8% at 50%, 44.0% at 70%, 20.4% at 80%), once the occluder removes most of the object's discriminative appearance. This boundary is the occlusion limitation stated in Section 5.

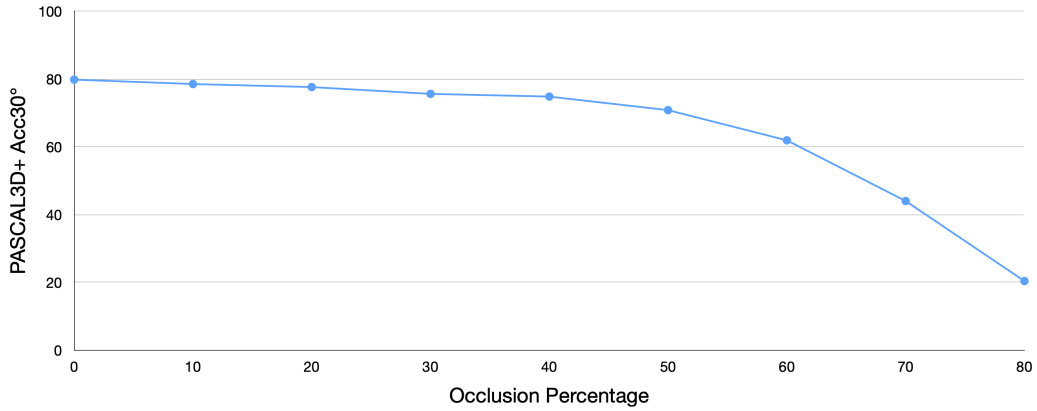


Figure 6: **Controlled occlusion sweep on PASCAL3D+.** A rectangular occluder covers an increasing fraction of the object bounding box. Accuracy degrades gracefully up to $\sim 40\%$ coverage and drops sharply beyond 50%, once most discriminative appearance is removed.

G Full Symmetry Breakdown

The main paper (Figure 4) reports symmetry-stratified accuracy for the two most common classes ($s{=}\infty$ and $s{=}1$). Figure 7 provides the complete breakdown across all symmetry classes $s \in \{\infty, 1, 2, 4\}$ on ImageNet3D and Omni6DPose. Two-fold and four-fold symmetries are rare in both datasets, and performance differences between methods are less systematic for these classes. On two-fold symmetric categories in ImageNet3D ($n{=}22$), our method falls slightly behind OrientAnything V2, which was trained on ImageNet3D and has seen all categories during training.

Figure 8 additionally compares our full method to the PCA-only ablation without per-sequence alignment. For continuously symmetric categories, the PCA-only model outperforms the full method, showing that PCA is an effective gravity prior. For non-symmetric categories, the per-sequence alignment is necessary to disambiguate semantic axes.

H Mesh Resolution vs. Runtime Trade-off

Our correspondence target is a fixed primitive mesh with M surface vertices. Increasing M can improve correspondence granularity and downstream pose accuracy, but increases compute and memory through the vertex-query decoding and matching steps. Figure 9 quantifies this trade-off

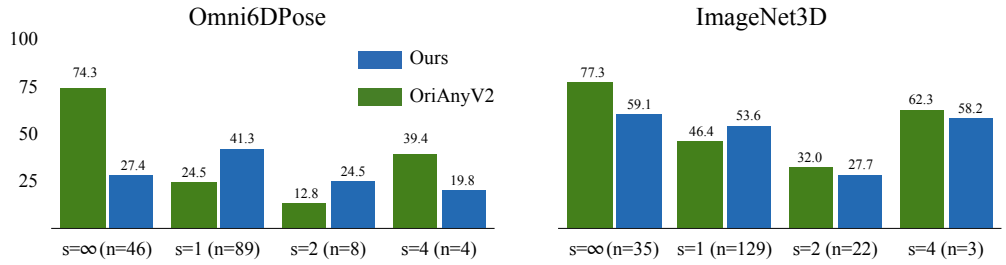


Figure 7: **Accuracy by symmetry class (full breakdown).** Symmetry-aware Acc@30° stratified by $s \in \{\infty, 1, 2, 4\}$ for Omni6DPose and ImageNet3D. Bars report accuracy, group sizes are shown alongside each class.

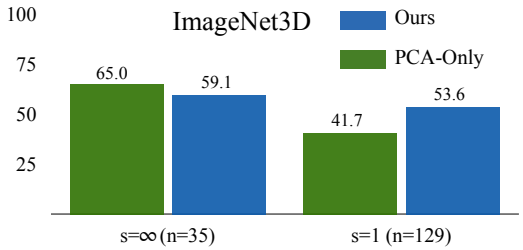


Figure 8: **Accuracy with and without per-sequence alignment.** Symmetry-aware Acc@30° stratified by $s \in \{\infty, 1\}$ for ImageNet3D. Bars report accuracy, group sizes are shown alongside each class.

by sweeping M and reporting ImageNet3D Acc@30° together with inference throughput (FPS). Accuracy improves with mesh resolution and begins to saturate, while FPS decreases monotonically. Higher vertex counts would likely benefit more if we increased the model's output resolution beyond the 64×64 grid, but this would substantially impact runtime. We select $M{=}1000$ as a practical operating point that captures most of the accuracy gains while maintaining favorable throughput. Runtime was measured on a NVIDIA A6000 GPU.

I Per-Category Results on ObjectNet3D and PASCAL3D+

Figure 10 reports per-category $Acc@30°$ on the 20-category ObjectNet3D split and the 7-category PASCAL3D+ split used by Common3D [51]. Baselines train separate models per category on approximately 50 sequences each. On ObjectNet3D, our full model achieves the highest average (70.0%) and leads on 15 of 20 categories. The largest gains appear on categories with distinctive geometric and semantic structure, such as bicycle (89.1%), chair (98.8%), and couch (98.7%), where the combination of geometric and semantic priors provides strong orientation signals. Categories where our method trails tend to be small, geometrically ambiguous objects where category-specific training can exploit shape priors that a shared model cannot. The scaling trend is consistent across most categories: $\text{Ours}^{\text{cat}}_{50} \rightarrow \text{Ours}^{\text{cat}}_{\text{full}} \rightarrow \text{Ours}$ improves performance progressively, with the full category-agnostic model often substantially outperforming the data-matched $\text{Ours}^{\text{cat}}_{50}$, indicating that cross-category training transfers useful structure even to categories seen during training.

On PASCAL3D+, the pattern is similar: our full model leads on 6 of 7 categories and achieves 85.9% average. The largest gains over Common3D appear on chair ($+27.8$), bus ($+14.8$), and TV monitor ($+14.0$). TV monitor remains the hardest category (65.1%), as its near-symmetric geometry limits the orientation signal available from multi-view consistency.

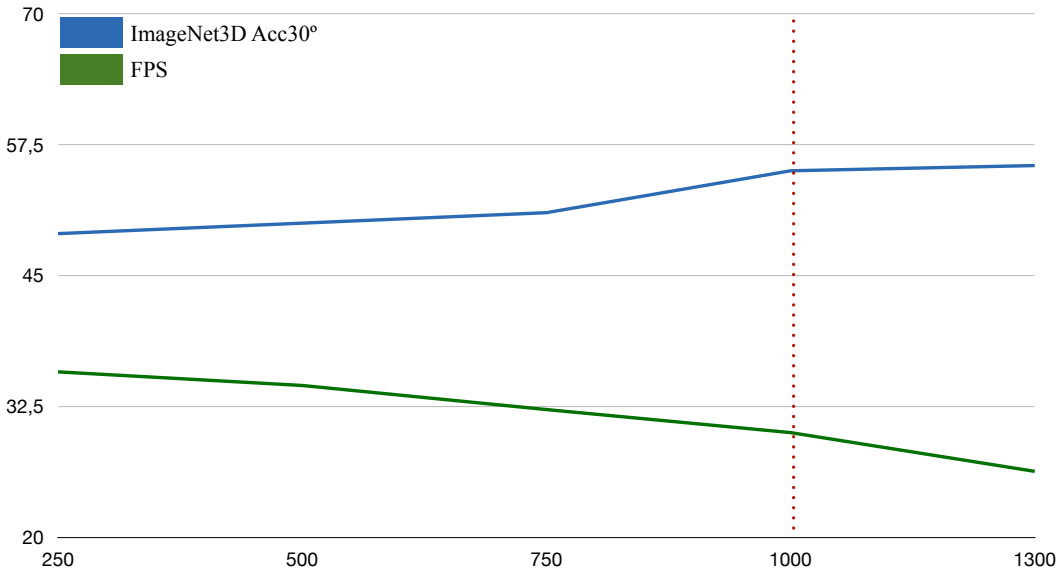


Figure 9: **Accuracy–runtime trade-off.** ImageNet3D Acc@30° (blue) and inference speed in FPS (green) as a function of mesh vertices M. Red dotted line: our default (M=1000).

	ObjectNet3D (20 categories)										AVG
									cup		
ZSP [66]	23.1	50.8	58.6	30.5	60.3	46.4	36.8	55.5	33.0	21.7	42.2
UOP3D [50]	18.0	62.1	57.8	78.3	98.1	**54.6**	52.2	76.6	38.2	14.1	52.4
Common3D [51]	25.5	77.0	65.7	81.1	99.0	51.8	65.9	79.0	44.7	19.4	56.8
Ours$^{cat}_{50}$	24.7	68.2	76.0	72.9	89.7	20.0	54.6	86.8	38.2	9.0	53.6
Ours$^{cat}_{full}$	41.5	74.4	83.6	93.7	98.2	36.8	94.2	91.6	66.0	4.9	63.1
Ours	**45.5**	**86.7**	**89.1**	**95.0**	**99.1**	45.1	**98.8**	**98.7**	**68.5**	**30.0**	**70.0**
ZSP [66]	**46.8**	60.5	50.5	50.3	28.8	41.6	25.8	28.8	56.3	37.9	42.2
UOP3D [50]	26.9	53.3	**80.3**	69.0	44.7	**54.4**	15.5	60.6	39.6	53.2	52.4
Common3D [51]	34.5	54.2	76.1	81.0	38.5	52.4	15.7	57.3	65.0	52.4	56.8
Ours$^{cat}_{50}$	28.9	45.4	68.3	80.6	79.9	32.2	20.1	51.3	57.3	67.9	53.6
Ours$^{cat}_{full}$	38.3	60.5	58.4	87.3	82.5	34.0	26.1	**63.6**	81.3	44.6	63.1
Ours	43.2	**74.1**	66.6	**88.3**	**86.3**	36.9	**56.0**	36.3	**84.9**	**71.6**	**70.0**

(a) **ObjectNet3D.** Acc@30°(%) on the 20-category Common3D[51] split.

	PASCAL3D+ (7 categories)							AVG
ZSP	61.7	21.4	61.6	42.6	52.9	43.1	39.0	46.0
UOP3D	58.4	79.3	98.2	51.9	76.6	67.0	53.1	69.2
Common3D	67.2	82.9	**99.3**	67.8	80.9	78.1	51.1	75.3
Ours$^{cat}_{50}$	67.4	68.8	81.3	50.7	79.6	69.0	59.6	68.1
Ours$^{cat}_{full}$	74.4	93.8	90.4	89.9	82.9	78.7	61.0	81.6
Ours	**80.0**	**97.7**	96.7	**95.6**	**84.6**	**81.5**	**65.1**	**85.9**

(b) **PASCAL3D+.** Acc@30° (%) on the 7-category protocol of Common3D [51].

Figure 10: **Category-level pose estimation results.** Left: ObjectNet3D results on the 20-category Common3D split. Right: PASCAL3D+ results on 7 categories. Baselines train separate models per category on roughly 50 sequences. Our fairness controls show scaling effects: Ours$^{cat}_{50}$ trains on the same target categories with a 50-sequence-per-category cap, Ours$^{cat}_{full}$ trains on the same target categories using all available UCO3D sequences, and **Ours** trains category-agnostically on all UCO3D.

J Qualitative Results

J.1 Pose Predictions Across Benchmarks

Figure 11 visualizes predicted poses across the five evaluation benchmarks. For each dataset we show input images overlaid with predicted canonical axes (top) and a ShapeNet [8] mesh rendered under the predicted pose (bottom). Because RGB-only pose is evaluated in a scale-normalized coordinate system, rendered meshes use a fixed canonical scale and translations are comparable only up to a global scale factor. The examples illustrate that the learned canonical frame induces consistent axis conventions across categories and datasets.

K LLM-Derived Symmetry Labels

Several evaluations use symmetry-aware rotation metrics, where the error for a prediction is the minimum over symmetry-equivalent ground-truth rotations (following [32]). This requires assigning each category a discrete yaw symmetry class. We derive these labels automatically using a large language model to avoid manual annotation. These labels are used only at evaluation time and are applied uniformly across all compared methods; they are not used during training.

K.1 Symmetry Classes

We assign each category to one of four classes:

1. **Class** ∞ (continuous): rotationally symmetric about the vertical axis; yaw is unrecoverable.
2. **Class 1 (asymmetric)**: a unique identifiable front; full yaw determination is possible.
3. **Class 2 (**180° **ambiguity)**: front and back are indistinguishable but side views differ.
4. **Class 4 (**90° **ambiguity)**: all four cardinal yaw orientations are indistinguishable.

K.2 LLM and Prompt Details

We generated symmetry labels using ChatGPT (GPT-5.2), queried in February 2026 with default decoding settings. We used a single fixed prompt and queried the model with the full list of categories from all considered datasets. We did not iterate on the prompt or revise individual labels based on downstream results. To simplify the prompt, we replaced symmetry class name ∞ with 0 in the prompt. The complete category-to-symmetry mapping is provided as a CSV file with the released code.

Prompt:

> I have a list of object categories that I need labeled by their object pose symmetry class. Each object category is assigned a discrete symmetry class based on the degree of pose ambiguity an observer would face when estimating the object's yaw from a single view:
>
> - Class 0 (Continuous): The object is rotationally symmetric about its vertical axis, making yaw unrecoverable. Applies to cylindrical (bottle, can), spherical (ball, orange), round flat (plate, pizza), and generally blob-shaped objects (pineapple, hamburger, hat) — even if minor surface details exist. - Class 1 (Asymmetric): The object has a unique, identifiable front face, allowing full pose determination. This is the default class and covers most man-made objects including vehicles, electronics, furniture with a clear front, labeled containers, tools, and figures. - Class 2 (180° ambiguity): Front and back views are indistinguishable, but side views differ. Applies to bilaterally symmetric elongated objects (barbell, pen, sausage) and objects symmetric through their dividing plane (bench, pillow, radiator). - Class 4 (90° ambiguity): All four cardinal views are identical. Applies to objects with square cross-sectional symmetry (table, box, dice, stool).
>
> Please label the following categories with their symmetry class (0, 1, 2, or 4) and output the result as a CSV with columns category,symmetry_class:

K.3 Sample Annotations

```
category,symmetry_class
aeroplane,1
air_hammer,1
airship,1
ambulance,1
ashtray,0
ax,1
backpack,1
ball,0
banana,2
barbell,2
baseball_bat,0
basket,0
```

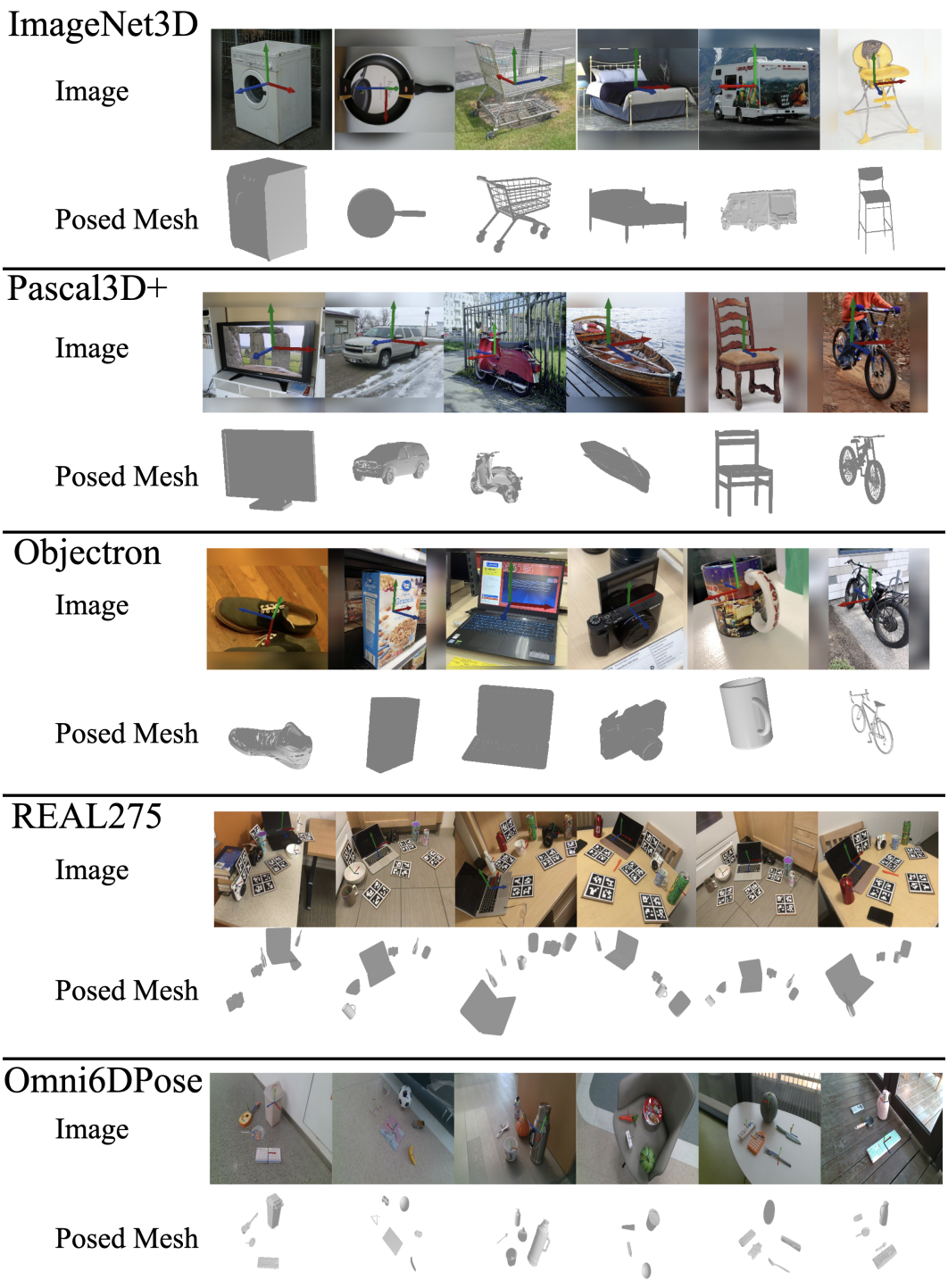


Figure 11: **Qualitative pose predictions across benchmarks.** For each dataset: input images with predicted canonical axes (top) and ShapeNet meshes rendered under the predicted pose (bottom). Blue axis indicates the predicted "front" direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim that a shared canonical frame can emerge from self-supervised training on in-the-wild object-centric videos without canonical pose labels or category conditioning, achieving competitive performance on category-level pose estimation benchmarks. These claims are supported by experiments across five benchmarks (Table 1), data scaling analysis (Table 2), and ablation studies (Table 3).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 contains a dedicated Limitations paragraph covering: (1) the rigid-object assumption, including the behavior on articulated objects; (2) chirality, since mirrored instances share one canonical frame; (3) translation being recovered only up to scale; (4) the sharp drop in accuracy beyond roughly 50% occlusion (Appendix F); (5) symmetry being handled at evaluation but not during training, which causes the main performance gap on symmetric objects (Section 4.5, Figure 4a). We further discuss that evaluation requires a convention mapping to each benchmark's frame, and show that it only selects among discrete axis conventions (Section 4.5, Table 3, Appendix E). We also note that we cannot verify that the pretrained backbone has never seen the held-out categories (Section 4.5), and that the occlusion comparison is not controlled for input resolution or training data (Appendix F). The ablations (Section 4.4) analyze when PCA initialization alone is insufficient.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results or proofs. The method is empirical, with the alignment step based on the well-known Wahba's problem which admits a standard closed-form SVD solution.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main paper describes the method, architecture, training setup, and evaluation protocol in full. The appendix provides complete implementation details including all hyperparameters (learning rates, augmentation schedules, temperature schedules, RANSAC parameters, loss weights), network architecture specifications, and the evaluation protocol with symmetry-aware metrics and convention mapping. The code and model checkpoint are publicly available at `https://github.com/Fischer-Tom/Emergent-Canonical-Frame`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Justification: Training and evaluation code, model checkpoints, the unseen-category split (Section 4.5), and the LLM-derived symmetry labels are publicly available at `https://github.com/Fischer-Tom/Emergent-Canonical-Frame`. The training data (UCO3D) is publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 provides the training setup, evaluation datasets, baselines, metrics, and convention mapping protocol. The appendix provides full details on data preprocessing, augmentation schedules, network architecture, optimizer (AdamW with weight decay 0.05), learning rates, temperature schedules, RANSAC parameters, and loss formulation.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report single-run results following standard practice in category-level pose estimation. Training the full model requires approximately 75,000 iterations on four H100 GPUs, making multiple-seed runs prohibitively expensive. We mitigate this by evaluating on five diverse benchmarks and reporting both median error and accuracy metrics, providing robustness through breadth of evaluation rather than repeated runs. We pick a single fixed random seed and keep it consistent throughout all runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B states that we train on four H100 GPUs for 15 epochs ($\sim$75k iterations) and reports batch size and mixed-precision settings. Appendix H reports inference throughput (FPS) on an NVIDIA A6000 GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets (UCO3D, REAL275, Omni6DPose, Pascal3D+, ImageNet3D, Objectron, ObjectNet3D) for their intended research purposes. No human subjects are involved. The work does not raise privacy or fairness concerns beyond those common to computer vision research.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work addresses canonical object pose estimation from video, which is foundational computer vision research. We do not foresee direct negative societal impacts beyond those common to general-purpose 3D understanding (e.g., potential use in surveillance systems). The method does not generate content, process personal data, or make decisions affecting individuals.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: The released model estimates object pose from images and does not generate content, manipulate images, or produce outputs that pose a risk for misuse. The training data (UCO3D) is a publicly available research dataset of object-centric videos.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and pretrained models used are cited. UCO3D, REAL275, Omni6DPose, Pascal3D+, ImageNet3D, Objectron, and ObjectNet3D are cited with their original publications. The DINOv3 backbone, LoRA, PoseLib, and nvdiffrast are cited. All assets are used in accordance with their respective licenses for research purposes.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release training and evaluation code, model checkpoints, and LLM-derived symmetry label mappings. The symmetry labels are documented in the appendix (Appendix K), including the LLM used (GPT-5.2), the exact prompt, and sample annotations. A CSV file with the complete category-to-symmetry mapping is included with the code release.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use an LLM (GPT-5.2) to derive per-category symmetry class labels for evaluation. This is a non-standard use that affects the reported metrics. The LLM, prompt, and complete output are documented in Appendix K. These labels are used only at evaluation time and applied uniformly to all compared methods; they do not affect training.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.