

EgoArt: Learning 3D Articulation Affordances from Egocentric Human Interaction Videos

Anonymous Authors

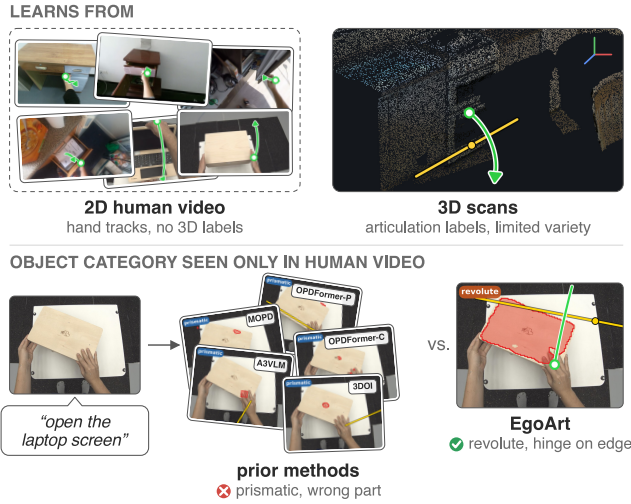


Fig. 1: EgoArt learns 3D articulation from 2D human video. Trained on hand trajectories (top left) and 3D articulation labels (top right), it recovers the hinge of a laptop lid, a category seen only in human video, while other 3D-supervised methods fail.

Abstract—Manipulating articulated objects requires identifying the relevant part and predicting how it can move. We present EgoArt, an instruction-conditioned model that grounds the task-relevant part from a single RGB image and predicts its 3D articulation affordance, including the mask, interaction point, motion type, axis, hinge, and motion extent. These predictions directly define a 3D interaction trajectory. To reduce reliance on limited 3D annotations, we bridge 2D human hand trajectories and 3D articulation through an analytic trajectory generator, allowing egocentric videos without 3D labels to supervise explicit articulation parameters. The same formulation also supports joint training with 3D-labelled data, for which we derive a closed-form trajectory loss that requires neither explicit trajectory sampling nor annotated motion extent. Trained jointly on SceneFun3D, HOI4D, EPIC-KITCHENS, and ARCTIC, EgoArt improves axis and hinge estimation over prior image-based articulation methods while generalizing to object categories seen only in 2D human video.

I. INTRODUCTION

Understanding 3D articulation is fundamental to reasoning about the functionality of objects. Articulation describes how an object part can move, including its motion type, axis, hinge, and extent [1], [2]. These parameters capture the geometric constraints of object interaction and allow a robot to reason about action consequences. This understanding

is essential for planning and manipulating across different interaction points, directions, and motion extents [3]–[5].

Learning 3D articulation, however, remains limited by the availability of real-world 3D annotations. Existing approaches typically rely on point clouds [1], [6], [7], RGB-D scans [8], CAD models [9], or explicitly annotated 3D articulation parameters [10]. Such annotations are *expensive* to acquire and cover only a limited range of scenes and object categories. Consequently, current articulation models are largely trained on relatively narrow distributions of 3D-labelled data [10]–[13], limiting their ability to *generalize* to the diversity of objects encountered in daily life.

With the rapid development and growing adoption of egocentric view recording glasses, egocentric video provides a promising source of broader supervision. Egocentric datasets [14]–[16] capture people interacting with diverse objects at a scale that existing 3D functional annotations cannot match. However, they typically provide only RGB observations and 2D motion cues rather than explicit 3D articulation labels. Such 2D trajectories contain rich information about how an object moves [17], [18], but existing articulation models do not directly use them to supervise explicit 3D articulation parameters. This raises the question we answer in this paper: *how can 2D human interaction trajectories supervise a model that predicts 3D articulation?*

We address this challenge with EgoArt, an instruction-conditioned model that identifies the task-relevant actionable part and predicts its complete 3D articulation affordance from a single RGB image. The predicted mask, interaction point, motion type, axis, hinge, and motion extent jointly define a 3D interaction trajectory for manipulating that part. This analytic connection allows both 3D articulation annotations and 2D hand trajectories from human video to supervise the same underlying articulation parameters. As a result, human interaction videos can be used to train a 3D articulation model **without** requiring 3D articulation labels for those videos, allowing the model to learn from objects and interactions beyond those covered by existing 3D articulation datasets.

In summary, our contributions are:

- We introduce EgoArt, an instruction-conditioned model that grounds the task-relevant actionable part in a scene and predicts its complete 3D articulation affordance from a single RGB image, including the part mask, interaction point, motion type, axis, hinge, and motion extent. These predictions directly define a 3D interaction trajectory for manipulating the selected part.
- We propose a unified supervision framework that con-

nects 2D human interaction trajectories and 3D articulation through an analytic trajectory generator, enabling egocentric videos without 3D articulation labels to supervise explicit 3D articulation parameters.

- We derive a closed-form 3D articulation loss that couples the axis, hinge, and interaction point through their induced motion, improving articulation accuracy over direct parameter supervision while requiring neither trajectory sampling nor ground-truth motion extent.

II. RELATED WORK

A. Articulation estimation

Most methods estimate articulation from 3D input. Some works segment movable parts and predict joint type, axis and origin from point clouds or depth [1], [2], [6], [7], [19], trained on synthetic assets [9], [20]; recent datasets bring the labels to real scans [8], [21], [22], where per-point voting averaged over a part mask is the strongest method [8]. A second line of work, driven by manipulation, predicts how a part will move as a dense flow or affordance and recovers axis and origin from it [3]–[5], rendering the motion analytically from the parameters, a design our decoder shares. Image-based estimation exists too. OPD and its successors detect openable parts with type, axis and origin from RGB or RGB-D [10], [11], [23], 3DOI predicts how a queried point moves [12], and A3VLM writes the articulation as text [13]; OPD and A3VLM are trained on 3D-labelled data alone and are limited to what that annotation covers, and 3DOI on video frames annotated by hand with a 2D axis line, so it learns from images of interaction but not from motion. AFUN [24] is the closest to us: it converts human-video motion into pseudo-3D supervision by lifting 2D trajectories with estimated depth, and learns a free-form 3D motion curve. In contrast, EgoArt predicts explicit 3D articulation parameters and supervises them directly in 2D by projecting their induced trajectory onto the observed hand track. The resulting representation explicitly exposes the motion type, axis, hinge, and extent for downstream manipulation.

B. Learning from human video

Egocentric and hand-object datasets record how people move objects at a scale no 3D annotation reaches [16], [25]–[29]. Prior work mines them for 2D affordances: where a hand will touch and how it will move [18], [30], [31], or point tracks that serve as targets for manipulation policies [32]–[34]. FunRec [35] recovers explicit articulation from egocentric RGB-D video by fitting each sequence independently to 3D point tracks, without learning a predictive model. In contrast, EgoArt turns 2D hand motion into supervision for a learned 3D articulation predictor, enabling scalable learning from human video and single-RGB-image inference at test time.

C. Language-conditioned heads on frozen backbones

Our model builds on referring segmentation, the task of producing the mask that a phrase refers to. Prior work generally follows one of two designs: the text is fused into

the image features and a decoder predicts the mask from the fused features [36], or a general segmentation model is prompted with the phrase or with points and boxes derived from it [37]. Recent methods keep the image backbone frozen and self-supervised [38], [39]: its features are strong and dense enough that a light adapter reading them at several scales [40], [41] suffices for pixel-level tasks, and a text encoder trained to match the same feature space [42] lets a phrase select image regions without retraining the backbone. This has been used to ground affordances by name in open-vocabulary settings [43]. We adopt a frozen DINOv3 backbone with its matched text encoder and replace the segmentation head with articulation prediction heads; our contribution lies in the heads and their supervision rather than the backbone.

III. METHOD

A. Task Definition

Given an RGB image $\mathbf{I}$, an instruction $\mathbf{T}$ specifying a part and an action (e.g., “open the left door of the closet”), and the intrinsics $\mathbf{K}$, the model predicts the articulation affordance tuple

$$(M, \mathbf{p}, c, \mathbf{n}, \mathbf{q}, L), \quad (1)$$

i.e., the mask M of the actionable part, the interaction point $\mathbf{p}$, the motion type $c \in \{\text{rot}, \text{trans}\}$, the unit axis $\mathbf{n}$, a point $\mathbf{q}$ on the axis (the hinge; unused for prismatic parts) and the sweep L , the arc length the interaction point travels. M and $\mathbf{p}$ are predicted in the image; $\mathbf{p}$ and $\mathbf{q}$ are lifted to the camera frame with predicted depths z_p, z_q and $\mathbf{K}$. Human videos, which lack 3D articulation labels, supervise the model only through 2D hand motion. For 2D supervision, we express trajectory displacements in units of the predicted interaction-point depth z_p ((5)). Human video does not determine absolute metric scale; 3D-labelled sources provide metric supervision for the shared model.

B. Language-Conditioned Feature Encoder

The language-conditioned feature encoder takes an RGB image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$ and an instruction $\mathbf{T}$ as inputs and transforms them into a single language-conditioned feature map. Following the CRIS layout [36], we replace the CLIP pair with a frozen DINOv3 and its matched text encoder. The encoder comprises a frozen vision-language encoder, a pyramid adapter, a text-gated feature pyramid, and a transformer decoder.

Frozen encoder. Images are encoded by DINOv3 ViT-L/16 [39], while instructions are encoded by the text encoder [42], with the resulting visual and textual representations aligned in a shared 2048-dimensional embedding space. Both encoders are frozen; all subsequent modules are trained from scratch. The text encoder provides a sequence of projected text tokens $\mathbf{w} \in \mathbb{R}^{L \times 2048}$ and a global sentence representation $\mathbf{s} \in \mathbb{R}^{2048}$. Images are resized to 512×512 , yielding a 32×32 grid of visual patch tokens.

Pyramid adapter. DINOv3 produces a feature map at stride 16. Following the feature pyramid of ViTDet [44], we derive

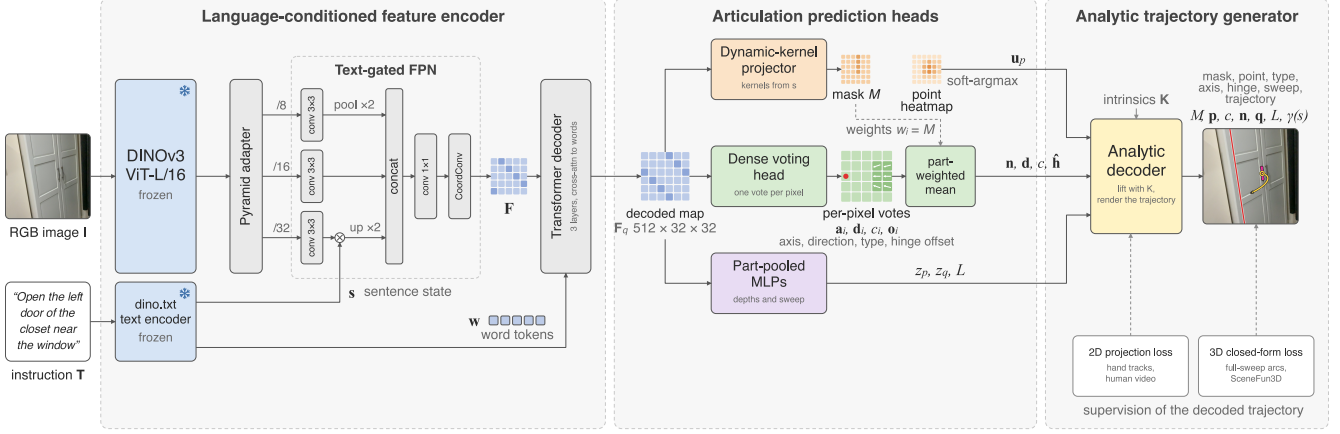


Fig. 2: Overview of EgoArt. A language-conditioned encoder produces a feature map, from which articulation prediction heads estimate the part mask, interaction point, motion type, axis, hinge, and motion extent. An analytic decoder renders the corresponding interaction trajectory, enabling supervision from both 2D human hand tracks and 3D articulation annotations.

three feature levels from the final ViT features: a transposed convolution upsamples them to stride 8, a 1×1 projection preserves stride 16, and a strided convolution downsamples them to stride 32. The resulting feature maps have 512, 1024, and 1024 channels, respectively.

Text-gated feature pyramid. The three pyramid levels are projected with 3×3 convolutions and aligned to stride 16. The coarsest level is gated by the projected sentence representation s , after which all levels are concatenated and fused by a 1×1 convolution and CoordConv to produce $\mathbf{F} \in \mathbb{R}^{512 \times 32 \times 32}$.

Transformer decoder. The fused feature map is flattened into 1024 visual tokens and processed by three transformer layers. Each layer applies self-attention over the visual tokens, followed by cross-attention to the projected text tokens. We use 8 attention heads, a hidden dimension of 512, a feed-forward dimension of 1024, and sinusoidal positional encodings for both modalities. The output tokens are reshaped into the decoded feature map $\mathbf{F}_q \in \mathbb{R}^{512 \times 32 \times 32}$, which is conditioned on the instruction and serves as the shared representation for all subsequent prediction heads.

C. Articulation Prediction Heads

The decoded feature map $\mathbf{F}_q$ is processed by three prediction heads (Fig. 2, middle): a dynamic-kernel projector, a dense voting head, and part-pooled MLPs.

Dynamic-kernel projector. The decoded map is upsampled to stride 4, while the sentence representation s generates 3×3 convolutional kernels and biases. Applying these instruction-conditioned kernels produces the part mask M and an interaction-point heatmap at 128×128 , together with an auxiliary hinge heatmap used only during training. A soft-argmax gives the 2D interaction point $\mathbf{u}_p \in [0, 1]^2$.

Dense voting head. A convolutional head predicts at each spatial location i of $\mathbf{F}_q$, with normalized image coordinate $\mathbf{u}_i$, an axis direction $\mathbf{a}_i$, a prismatic direction $\mathbf{d}_i$, motion-type logits $\mathbf{c}_i$, and a 2D offset $\mathbf{o}_i$ from $\mathbf{u}_i$ to the projected hinge.

These votes are aggregated using the part mask as weights; for the hinge,

$$\hat{\mathbf{h}} = \frac{\sum_i w_i (\mathbf{u}_i + \mathbf{o}_i)}{\sum_i w_i}, \quad (2)$$

where w_i is the ground-truth mask during training and the predicted mask at test time; axis, direction, and motion type are aggregated analogously.

Part-pooled MLPs. Small MLPs predict the interaction-point depth z_p , hinge depth z_q , and sweep L from part-pooled features together with global image and sentence features. The predicted 2D interaction point $\mathbf{u}_p$ and hinge $\hat{\mathbf{h}}$ are then lifted using $\mathbf{K}$ and their depths to obtain the 3D points $\mathbf{p}$ and $\mathbf{q}$.

D. Analytic Trajectory Generation and Supervision

Analytic trajectory decoder. Within the trajectory generator, we construct the interaction trajectory analytically from the predicted articulation parameters:

$$\gamma(s) = \begin{cases} \mathbf{p} + (\cos s\theta - 1)\mathbf{r} + \sin s\theta(\mathbf{n} \times \mathbf{r}) & \text{rot,} \\ \mathbf{p} + sL\mathbf{d} & \text{trans,} \end{cases} \quad (3)$$

where $\mathbf{r} = (\mathbf{I} - \mathbf{n}\mathbf{n}^\top)(\mathbf{p} - \mathbf{q})$, $\theta = \min(L/\|\mathbf{r}\|, \pi)$, and $s \in [0, 1]$. The ground-truth motion type selects the branch during training and the predicted type at test time. Thus, γ provides a differentiable mapping from the predicted articulation parameters to their induced motion.

2D hand-track supervision. Human videos provide 2D hand trajectories but no 3D articulation labels. We express the decoded displacement in units of the predicted interaction-point depth:

$$\Delta\bar{\gamma}(s) = \frac{\gamma(s) - \mathbf{p}}{z_p}, \quad z_p > 0. \quad (4)$$

For an observed hand track $\mathbf{x}_0, \dots, \mathbf{x}_N$ in image coordinates, we back-project its first position at unit depth, add this

normalized displacement, and reproject it using the camera intrinsics:

$$\mathcal{L}_{2D} = \frac{\sum_k \|\pi_{\mathbf{K}}(\mathbf{K}^{-1}[\mathbf{x}_0^\top, 1]^\top + \Delta\bar{\gamma}(s_k)) - \mathbf{x}_k\|^2}{\sum_k \|\mathbf{x}_k - \mathbf{x}_0\|^2 + \epsilon}. \quad (5)$$

Here $\pi_{\mathbf{K}}$ denotes perspective projection with intrinsics $\mathbf{K}$ in the same image coordinate system as the track, and $\epsilon > 0$ stabilizes the denominator for short tracks. Under a common rescaling of $\mathbf{p}$, $\mathbf{q}$, L , and z_p , the normalized displacement remains unchanged. This objective therefore supervises articulation without recovering absolute metric scale. We assume that the tracked hand point follows the actionable part during the selected interaction segment. Motion type, mask, and interaction point are additionally supervised with cross-entropy, Dice/BCE, and heatmap losses.

3D articulation supervision. For 3D-labelled data, the axis and hinge are available but the motion extent is not. We use $(\cdot)^*$ to denote quantities induced by the ground-truth articulation parameters. We therefore integrate trajectory error over a full rotation, avoiding an annotated or arbitrarily chosen sweep. For revolute joints, let $\mathbf{r} = (I - \mathbf{n}\mathbf{n}^\top)(\mathbf{p} - \mathbf{q})$ and $\mathbf{t} = \mathbf{n} \times \mathbf{r}$. Parameterizing the relative trajectory by rotation angle ϕ ,

$$\int_0^{2\pi} \|\Delta\gamma(\phi) - \Delta\gamma^*(\phi)\|^2 d\phi = \pi (3\|\Delta\mathbf{r}\|^2 + \|\Delta\mathbf{t}\|^2),$$

and normalizing by the ground-truth trajectory energy gives

$$\mathcal{L}_{3D} = \frac{3\|\Delta\mathbf{r}\|^2 + \|\Delta\mathbf{t}\|^2}{4\|\mathbf{r}^*\|^2} + \lambda(1 - \mathbf{n}^\top \mathbf{n}^*), \quad (6)$$

where $\Delta\mathbf{r} = \mathbf{r} - \mathbf{r}^*$ and $\Delta\mathbf{t} = \mathbf{t} - \mathbf{t}^*$. For prismatic joints, the trajectory term reduces to $\|\mathbf{d} - \mathbf{d}^*\|^2$. The axis term resolves sign ambiguity, while the 3D interaction point and hinge are supervised with squared errors.

Joint training. We jointly train a single model on SceneFun3D and the three human-video datasets (HOI4D, EPIC-KITCHENS, ARCTIC). Each mini-batch comes from one source and uses its available supervision: the full 3D objective for SceneFun3D, and the 2D projection, type, mask, and heatmap losses for human video. All sources share the same model parameters. To balance the smaller human-video datasets, we sample them ten times per epoch with augmentation.

IV. EXPERIMENTS

A. Implementation Details

Datasets. SceneFun3D [22] provides task descriptions, motion types, and 3D articulation axes and origins in registered RGB-D scans. We convert it to image-level records with an instruction, part mask, interaction point, and articulation parameters in the camera frame. After filtering invalid, small, border-clipped, and revolute elements with lever radius below 10 cm, we obtain 59,174 frames from 224 scenes, 22.5 % of which are revolute.

For 2D supervision, we construct records from HOI4D [16], EPIC-KITCHENS [14], and ARCTIC [28], containing an initial frame, actionable-part mask, instruction,

TABLE I: Training data. Splits are by scene; 10 % of SceneFun3D scenes and 15 % of each hand source form the test splits. Hand sources are repeated ten times per epoch with augmentation.

Source	Train	Test	Part mask	Type	3D axis
SceneFun3D	54,086	5,088	scan splat	annotated	train
HOI4D	2,383	438	HOI4D labels	rules	none
EPIC-KITCHENS	306	53	VISOR+SAM 2	VLM	none
ARCTIC	2,230	329	rendered	CAD	eval



Fig. 3: Records mined from the three human-video sources. Each record is a frame with the mask of the actionable part (red), the track of the acting hand’s knuckle over the interaction (green; ring at the interaction point, arrowhead at the end), a motion-type label and an instruction.

motion type, and hand trajectory estimated by WiLoR [45]. Camera motion is compensated using ground-truth poses when available and homographies otherwise; we use the first hand position as the interaction point and the remaining trajectory as 2D motion supervision. For HOI4D, motion types are derived from category and action; for EPIC-KITCHENS, VISOR masks [27] are propagated with SAM 2 [46] and motion types are inferred by a vision-language model; for ARCTIC, we extract monotonic joint-motion segments and render the actionable-part masks. Its 3D articulation annotations are used only for evaluation. Table I summarizes the datasets.

Evaluation. On SceneFun3D, each test frame has one annotated element with type, axis $\mathbf{n}^*$ and hinge line, and each method makes one prediction per frame. *Axis* is the signed angle between the predicted direction (rotation axis or direction of travel, by predicted type) and $\mathbf{n}^*$, over detected frames; *origin* is the distance from the predicted hinge to $\mathbf{q}^*$, the point of the annotated axis nearest the interaction point, over revolute frames with a hinge prediction. mIoU is the mean mask IoU and PDet the fraction of frames with $\text{IoU} \geq 0.25$. MA is the fraction of frames with the correct type and the axis within 10° (sign included); MAO also requires the hinge within 10 cm of the annotated axis line on revolute frames. +M, +MA and +MAO are the same tests counted on detected frames only, as in OPD, whose IoU 0.5 and 25 cm thresholds we adapt to handle-sized elements. Detectors are scored on their instance that best overlaps the element and 3DOI is given the interaction point, which favours the baselines.

On human video, masks are scored by mIoU, and the axis by the unsigned angle to a 3D ground truth the model

TABLE II: Results on the SceneFun3D test split (best in bold, second best underlined). Detectors are retrained on our split, the OPD-based image methods at 512×384 , USDNet at its 2 cm voxels. $\dagger$: oracle-matched instance (best IoU per element). $\ddagger$: A3VLM’s box hull (caps PDet at 77.1). Rows given the ground-truth part location are upper bounds. $\S$: 3DOI’s axis is undirected, so these ignore the sign (23.2 MA with it). 3DOI’s axis column is unsigned.

Method	Input	MA $\uparrow$	MAO $\uparrow$	axis $\downarrow$	origin $\downarrow$	mIoU $\uparrow$	PDet $\uparrow$	+M $\uparrow$	+MA $\uparrow$	+MAO $\uparrow$
OPDFormer-C [11]	RGB-D	26.0	21.6	28.2	0.370	0.340 $\ddagger$	57.5 $\dagger$	53.8	22.8	19.2
OPDFormer-P [11]	RGB	14.4	9.4	41.4	0.725	0.372 $\ddagger$	63.7 $\dagger$	60.4	13.1	8.6
MOPD [23]	RGB	17.2	11.6	42.1	0.666	0.357 $\ddagger$	61.4 $\dagger$	56.5	14.9	10.3
USDNet [8]	scan	21.8	18.0	16.3	0.988	0.071 $\dagger$	5.4 $\dagger$	5.4	3.4	2.5
A3VLM [13]	RGB+text	<u>43.8</u>	35.5	21.5	0.434	<u>0.240$\ddagger$</u>	<u>46.9$\ddagger$</u>	<u>45.5</u>	<u>22.6</u>	<u>20.3</u>
<i>Given the ground-truth part location</i>						<i>not comparable: the part location is an input</i>				
A3VLM [13]	RGB+GT box	45.7	39.1	26.6	<u>0.361</u>	0.408 $\ddagger$	77.1 $\dagger$	74.2	35.3	32.1
3DOI [12]	RGB-D+GT pt	35.7 $\S$	33.1 $\S$	30.5 $\S$	0.811	0.595	96.2	81.1	33.5 $\S$	31.0 $\S$
EgoArt (ours)	RGB+text	42.7	<u>35.9</u>	<u>16.5</u>	0.248	0.277	51.9	49.9	29.1	24.8

never saw: the part poses of HOI4D (325 articulated records), the CAD axes of ARCTIC and our own annotations on the 53 EPIC-KITCHENS test records; the sources define no opening direction, so the sign is not scored, chance is 57.3° , and MA uses a 30° tolerance on the unsigned angle. *Contact* is the image distance (fraction of the image) between the predicted interaction point and the hand’s first position; *endpoint* is the same distance between the end of the decoded trajectory, projected with the model’s own depth, and the end of the hand track, a scale-free test of extent, direction and shape. Hinges are not scored on video, where metric depth is unsupervised.

Training. Our model has 939M parameters, of which 72M are trainable; the DINOv3 vision and text encoders remain frozen. We train with Adam for 20 epochs using batch size 64, learning rate 2×10^{-5} , weight decay 10^{-4} , and mixed precision. The learning rate is reduced by $10\times$ after epochs 16 and 19. Training takes about 5 h on one RTX PRO 6000.

B. Results on SceneFun3D

Table II compares EgoArt with prior articulation estimators retrained on SceneFun3D. The previous methods were designed for parts, not functional elements: a SceneFun3D handle is about 12 pixels wide at OPDFormer’s 256 by 192 input. To make the comparison fair while preserving the 4:3 input aspect ratio assumed by the OPD-based architectures, we retrain them at 512×384 , keeping all other settings unchanged; EgoArt uses 512×512 inputs. OPDFormer-C, which regresses the axis in the camera frame, is the best of them; OPDFormer-P, whose axis comes from a predicted object pose, and MOPD, which adds an object-level branch to it, land at 14.4 and 17.2 MA with origins 0.7 m off. USDNet has the best axis in the table, 16.3 degrees, but on the 5% of frames in which it finds the element at all: these are the large elements whose scan geometry fixes the axis, and its ungated MA of 21.8 shows what happens on the rest. Halving its voxels to 1 cm finds four times as many elements and raises its axis error to 42 degrees, so the number reflects which frames it answers, not a better axis estimator.

The strongest prior method is A3VLM, a 13B vision-language model retrained on SceneFun3D: given our instruction it reaches 43.8 MA, one point above EgoArt, but

its axes are 5° less accurate, its origin 19 cm further off, and it localises the part far less often (PDet 46.9 against a 77.1 ceiling for its box output), at roughly 180 times the trainable parameters. Handing it the ground-truth box lifts its MA only to 45.7, suggesting that grounding is not its primary bottleneck. The detectors’ mask and PDet numbers are oracle-matched and not comparable to ours: the oracle hands each detector the ground-truth element and its best-overlapping instance at any confidence, two things a text-grounded model has to get right itself (with a confidence threshold of 0.5 the image detectors’ PDet falls to 5.4 to 7.1). Even with that head start, which puts the image detectors above us on mIoU and PDet, EgoArt is ahead on every articulation column: its gated axis and hinge rates are six and five points above the best detector’s, its ungated MA 17 points above, its axis error 12° lower and its origin 12 cm closer. A3VLM is the only baseline that, like ours, takes the instruction; 3DOI is prompted with the ground-truth interaction point, which gives it the best masks of any baseline but only 35.7 MA even with its axis sign ignored: it answers “free-form” for a fifth of the elements and finds fewer than half of the hinges. Fig. 4 shows the same comparison on test frames. With the oracle the detectors find the handle, but their axes are tilted across the oven door and off the drawers’ travel; A3VLM’s axes are close but its part is a box; 3DOI segments the part but calls the door free-form and misdirects a drawer; EgoArt puts the hinge line on the door and oven seams and the travel along the drawers.

C. Does human video help 3D articulation?

To isolate the contribution of human video, we retrain the final model without the three hand-video sources and evaluate both models on SceneFun3D and on the hand-video test records, where the SceneFun3D-only model is zero-shot (Table III). On SceneFun3D, joint training largely preserves performance: MA and MAO stay within one point (43.7 against 42.7, 35.6 against 35.9), the gated rates rise, and the axis and origin errors improve from 19.2° to 16.5° and from 0.262 m to 0.248 m.

The effect is much larger on human-video data. On HOI4D and ARCTIC the joint model passes the 30° MA test on 29 and 30% of the records against 20 and 12% for the

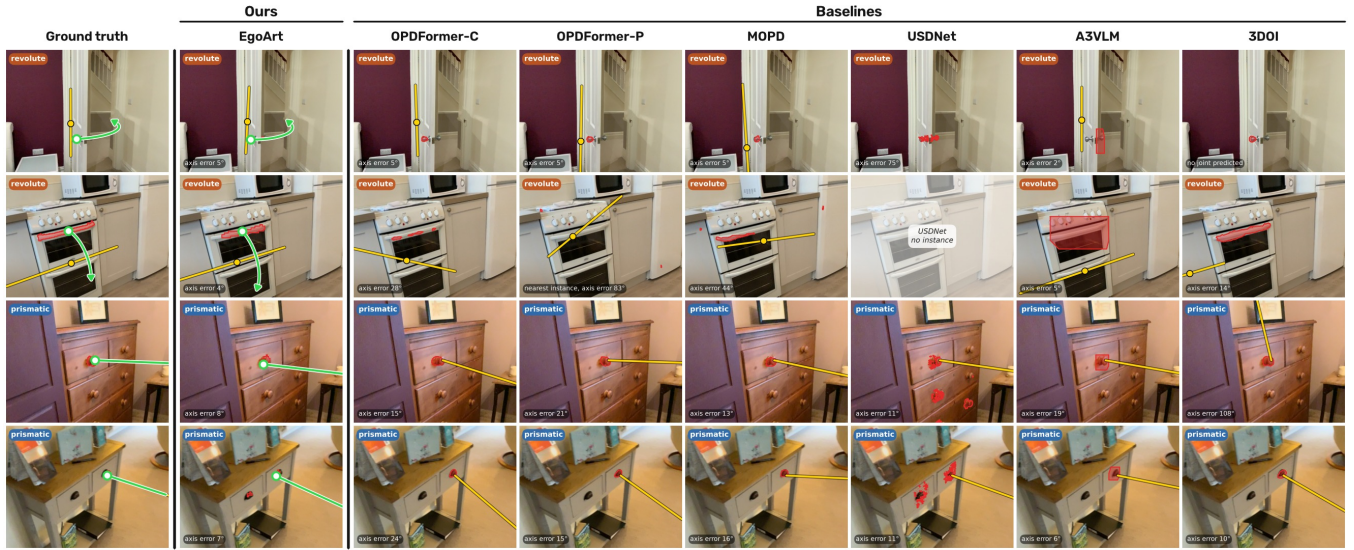


Fig. 4: SceneFun3D test frames, one per row. Red: predicted part; yellow: axis, drawn as the hinge line with its foot for revolute parts and as the direction of travel for prismatic ones; green: motion of the interaction point (white) over the annotated extent. Green trajectories are shown only for the ground truth and EgoArt, since the compared baselines do not predict interaction points; 3DOI instead takes an interaction point as input. None of the compared baselines predicts motion extent, which EgoArt additionally estimates. Detectors show their oracle-matched instance (the nearest detection if none overlaps), and A3VLM its box hull.

TABLE III: Training with and without human video. Top: SceneFun3D test split, protocol of Table II. Bottom: hand-video test records, zero-shot for the SceneFun3D-only row; MA with a 30° tolerance (*: our own EPIC annotations, 53 records, indicative only).

Data	SceneFun3D									
	MA $\uparrow$	MAO $\uparrow$	axis $\downarrow$	origin $\downarrow$	mIoU $\uparrow$	PDet $\uparrow$	+M $\uparrow$	+MA $\uparrow$	+MAO $\uparrow$	
SceneFun3D only	43.7	35.6	19.2	0.262	0.263	48.8	44.9	26.5	21.6	
+ human video	42.7	35.9	16.5	0.248	0.277	51.9	49.9	29.1	24.8	

Data	HOI4D					EPIC*					ARCTIC				
	MA $\uparrow$	axis $\downarrow$	mIoU $\uparrow$	contact $\downarrow$	endpoint $\downarrow$	MA $\uparrow$	axis $\downarrow$	mIoU $\uparrow$	contact $\downarrow$	endpoint $\downarrow$	MA $\uparrow$	axis $\downarrow$	mIoU $\uparrow$	contact $\downarrow$	endpoint $\downarrow$
SceneFun3D only	20.3	53.4	0.01	0.217	0.361	56.6	35.7	0.01	0.203	0.356	11.6	56.6	0.03	0.213	0.391
+ human video	28.6	42.9	0.51	0.020	0.056	47.2	36.7	0.19	0.055	0.219	29.5	45.8	0.51	0.060	0.164

SceneFun3D-only model, whose axes sit near chance (53.4° and 56.6° against 57.3° for a random direction); the joint model brings them to 42.9° and 45.8° . The SceneFun3D-only model does not find the parts (0.01 mIoU on HOI4D and EPIC-KITCHENS) and puts its interaction point a fifth of the image away from the hand, while the joint model segments them (0.51 and 0.19 mIoU), places the contact point within 0.02 to 0.06 of the image of the hand, and ends its decoded motion within 0.06 to 0.22 of the image of where the hand ended. Human video is what makes the model work on objects that scanned 3D data does not cover.

EPIC-KITCHENS has no official 3D articulation labels, so we manually annotate axes for its 53 test records. Given this small evaluation set, the differences between the two models are indicative only and we do not draw conclusions from their ordering. On HOI4D and ARCTIC, the jointly trained model performs clearly better than chance, demonstrating that human-video supervision can transfer to 3D articulation. The transfer is strongest for box lids and laptop screens (19°

mean axis error on each), whose longer motion arcs constrain the axis more strongly, and weaker for small parts with short trajectories such as scissors and espresso levers.

Fig. 5 illustrates this transfer. On categories represented in SceneFun3D, such as microwave doors and drawers, the SceneFun3D-only model already degrades when moving from scanned scenes to human video. The difference is clearer for categories absent from SceneFun3D, such as a laptop lid and a toy car: the SceneFun3D-only model fails to recover their actionable parts or motion, whereas the jointly trained model segments the parts, predicts the correct motion type, and moves the interaction point consistently with the observed hand trajectory. These examples show how human-video supervision extends articulation prediction beyond the object categories covered by 3D-labelled scans.

D. In the wild Results

Fig. 6 shows both models on phone photos from environments absent from both SceneFun3D and the hand-

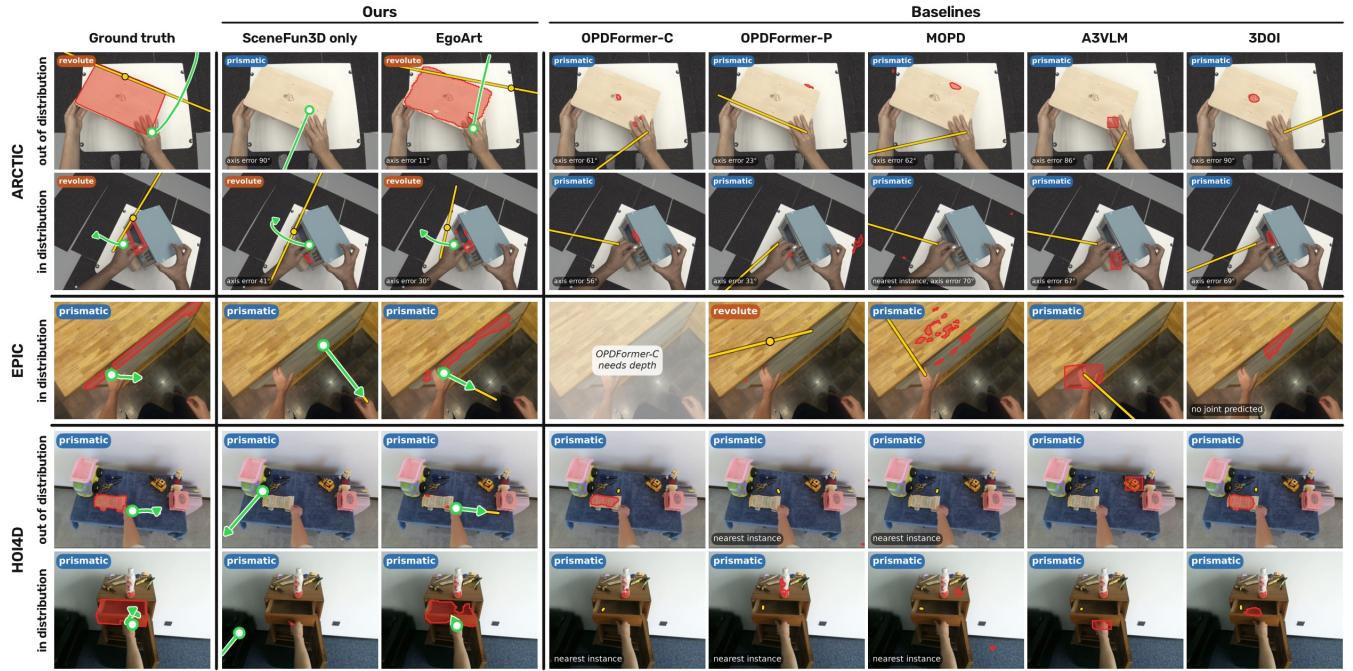


Fig. 5: Hand-video test frames, grouped by source and marked in or out of distribution by whether SceneFun3D contains the category. Overlays as in Fig. 4, with the hand track as ground-truth motion and EgoArt’s decoded trajectory as its own. OPDFormer-C needs depth, which EPIC-KITCHENS lacks; USDNet needs a scan and is omitted.

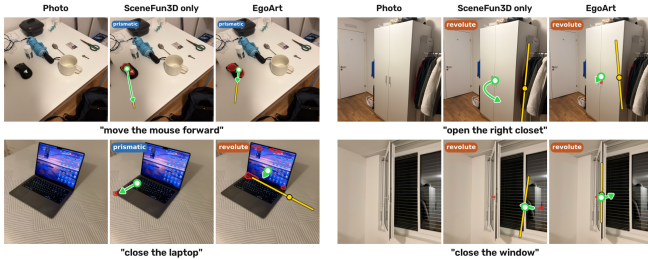


Fig. 6: Phone photos outside every training set, with the SceneFun3D-only model and EgoArt. Both handle the room door, a category SceneFun3D covers: handle found, revolute, a near-vertical hinge line, though inside the door rather than on its edge. On the laptop only EgoArt finds the screen and calls it revolute; the SceneFun3D-only model segments the external keyboard and predicts a slide.

video datasets. Both models generalize, to some extent, to objects absent from all training sources, such as the mouse, possibly aided by the pretrained vision-language backbone. On the closet door, both models also succeed, consistent with doors and cabinets being well represented in SceneFun3D, although EgoArt predicts the articulation more precisely. The clearest difference appears on the laptop, a category absent from SceneFun3D but present in human video: the SceneFun3D-only model predicts prismatic motion on the keyboard, whereas EgoArt identifies the screen, its hinge, and the correct revolute motion. This example highlights how human-video supervision extends articulation prediction to categories not covered by 3D training data.

TABLE IV: Ablations on the SceneFun3D test split. All rows use the final method and data; one factor changes per block.

Variant	MA $\uparrow$	MAO $\uparrow$	axis $\downarrow$	origin $\downarrow$	mIoU $\uparrow$
<i>3D loss</i>					
Direct parameter losses	36.7	28.5	20.1	0.305	0.270
Closed form (ours)	42.7	35.9	16.5	0.248	0.277
<i>Articulation head</i>					
Pooled MLP	36.9	32.0	24.7	0.301	0.239
Attention pooling	32.4	27.2	24.4	0.345	0.273
Dense voting (ours)	42.7	35.9	16.5	0.248	0.277

E. Ablation Study

The 3D loss. We compare our closed-form 3D loss with direct supervision of the articulation parameters, keeping all other settings unchanged. The direct baseline independently supervises the axis, hinge, and interaction point, whereas our loss couples them through the induced motion trajectory. As shown in Table IV (top), our loss improves MA from 36.7 to 42.7 and MAO from 28.5 to 35.9, while reducing the axis error by 4° and the origin error by 6cm. It also requires neither trajectory sampling nor annotated motion extent.

The articulation head. We compare several articulation head designs on the same decoded feature map, including pooled MLPs, attention-based pooling, and our dense voting head. Pooling compresses the spatial features of the part into a global representation, whereas dense voting preserves spatial information by predicting per-pixel votes for the motion type, axis, and hinge offset. As shown in Table IV, dense voting performs best across all metrics: 42.7 MA and 35.9 MAO against 36.9 and 32.0 for the pooled head, with the origin

error reduced to 0.248 m. The lower origin error suggests that spatial voting better preserves the local geometric evidence needed to locate the articulation axis.

V. CONCLUSION

We presented EgoArt, an instruction-conditioned model that grounds the task-relevant part in a scene and predicts a complete 3D articulation affordance from a single RGB image. By analytically mapping the predicted articulation to an interaction trajectory, EgoArt can learn jointly from 3D articulation annotations and 2D hand trajectories in human video. Our experiments show that human-video supervision, enabled by the hand-trajectory bridge, improves 3D axis prediction on human-video data and extends articulation prediction to object categories absent from the 3D training data, while preserving performance on the 3D benchmark. These results show that human interaction video can serve as scalable supervision for 3D articulation, reducing reliance on annotated 3D scans.

REFERENCES

- [1] X. Li, H. Wang, L. Yi, L. Guibas, A. L. Abbott, and S. Song, "Category-Level articulated object pose estimation," in *CVPR*, 2020.
- [2] A. Jain, R. Lioutikov, C. Chuck, and S. Niekum, "ScrewNet: Category-independent articulation model estimation from depth images using screw theory," in *ICRA*, 2021.
- [3] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2Act: From pixels to actions for articulated 3D objects," in *ICCV*, 2021.
- [4] B. Eisner, H. Zhang, and D. Held, "FlowBot3D: Learning 3D articulation flow to manipulate articulated objects," in *RSS*, 2022.
- [5] Q. Yu, J. Wang, W. Liu, C. Hao, L. Liu, L. Shao, W. Wang, and C. Lu, "GAMMA: Generalizable articulation modeling and manipulation for articulated objects," in *ICRA*, 2024.
- [6] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2Motion: joint analysis of motion parts and attributes from 3D shapes," in *CVPR*, 2019.
- [7] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building digital twins of articulated objects from interaction," in *CVPR*, 2022.
- [8] A.-M. Halacheva, Y. Miao, J.-N. Zaeche, X. Wang, L. V. Gool, and D. P. Paudel, "Articulable3D: holistic understanding of 3D scenes as universal scene description," in *ICCV*, 2025.
- [9] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A SimulAteD part-based interactive ENvironment," in *CVPR*, 2020.
- [10] H. Jiang, Y. Mao, M. Savva, and A. X. Chang, "OPD: single-view 3D openable part detection," in *ECCV*, 2022.
- [11] X. Sun, H. Jiang, M. Savva, and A. X. Chang, "OPDMulti: openable part detection for multiple objects," in *3DV*, 2024.
- [12] S. Qian and D. F. Fouhey, "Understanding 3D object interaction from a single image," in *ICCV*, 2023.
- [13] S. Huang, H. Chang, Y. Liu, Y. Zhu, H. Dong, A. Boularias, P. Gao, and H. Li, "A3VLM: Actionable articulation-aware vision language model," in *CoRL*, 2024.
- [14] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100," *IJCV*, vol. 130, pp. 33–55, 2022.
- [15] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan *et al.*, "Ego4D: Around the world in 3,000 hours of egocentric video," in *CVPR*, 2022.
- [16] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi, "HOI4D: a 4D egocentric dataset for Category-Level Human-Object interaction," in *CVPR*, 2022.
- [17] S. Liu, S. Tripathi, S. Majumdar, and X. Wang, "Joint hand motion and interaction hotspots prediction from egocentric videos," in *CVPR*, 2022.
- [18] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023.
- [19] Z. Yan, R. Hu, X. Yan, L. Chen, O. van Kaick, H. Zhang, and H. Huang, "RPM-Net: Recurrent prediction of motion and parts from point cloud," *ACM TOG*, vol. 38, no. 6, pp. 240:1–240:15, 2019.
- [20] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang, "GAPartNet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts," in *CVPR*, 2023.
- [21] Y. Mao, Y. Zhang, H. Jiang, A. X. Chang, and M. Savva, "Multi-Scan: Scalable RGBD scanning for 3D environments with articulated objects," in *NeurIPS*, 2022.
- [22] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann, "SceneFun3D: fine-grained functionality and affordance understanding in 3D scenes," in *CVPR*, 2024.
- [23] S. Li, X. Chen, H. Cheng, G. Zhou, H. Zhao, and G. Tian, "Locate n' rotate: Two-stage openable part detection with geometric foundation model priors," in *ACCV*, 2024.
- [24] Z. Wang, Y. Zhong, J. Fu, H. I. Christensen, and J. Gao, "AFUN: Towards an affordance foundation model for functionality understanding," *arXiv preprint arXiv:2606.02551*, 2026.
- [25] X. Wang, T. Kwon, M. Rad, B. Pan, I. Chakraborty, S. Andrist, D. Bohus, A. Feniello, B. Tekin, F. V. Fruejri *et al.*, "Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world," in *ICCV*, 2023.
- [26] T. Kwon, B. Tekin, J. Stuhmer, F. Bogo, and M. Pollefeys, "H2O: Two hands manipulating objects for first person interaction recognition," in *ICCV*, 2021.
- [27] A. Darkhalil, D. Shan, B. Zhu, J. Ma, A. Kar, R. Higgins, S. Fidler, D. Fouhey, and D. Damen, "EPIC-KITCHENS VISOR benchmark: Video segmentations and object relations," in *NeurIPS*, 2022.
- [28] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges, "ARCTIC: a dataset for dexterous bimanual Hand-Object manipulation," in *CVPR*, 2023.
- [29] T. Yoon, Y. Han, S. Ji, J. Park, S. Kim, T. Kwon, and H.-S. Kim, "Egextreme: a dataset for robust object pose estimation in egocentric views under extreme conditions," in *CVPR*, 2026.
- [30] T. Nagarajan, C. Feichtenhofer, and K. Grauman, "Grounded Human-Object interaction hotspots from video," in *ICCV*, 2019.
- [31] M. Liu, S. Tang, Y. Li, and J. Rehg, "Forecasting human-object interaction: Joint prediction of motor attention and actions in First Person Video," in *ECCV*, 2020.
- [32] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," in *RSS*, 2024.
- [33] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation," in *ECCV*, 2024.
- [34] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," in *CoRL*, 2024.
- [35] A. Delitzas, C. Zhang, A. Gavryushin, T. D. Mario, B. Sun, R. Dabral, L. Guibas, C. Theobalt, M. Pollefeys, F. Engelmann, and D. Barath, "FunRec: Reconstructing functional 3D scenes from egocentric interaction videos," in *CVPR*, 2026.
- [36] Z. Wang, Y. Lu, Q. Li, X. Tao, Y. Guo, M. Gong, and T. Liu, "CRIS: CLIP-Driven referring image segmentation," in *CVPR*, 2022.
- [37] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," in *ICCV*, 2023.
- [38] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: learning robust visual features without supervision," *TMLR*, 2024.
- [39] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolán, J. Brandt, C. Couprie, J. Mairal, H. Jegou, P. Labatut, and P. Bojanowski, "DINOv3," *TMLR*, 2026.
- [40] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *ICCV*, 2021.
- [41] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," in *ICLR*, 2023.

- [42] C. Jose, T. Moutakanni, D. Kang, F. Baldassarre, T. Darcet, H. Xu, D. Li, M. Szafraniec, M. Ramamonjisoa, M. Oquab, O. Siméoni, H. V. Vo, P. Labatut, and P. Bojanowski, “DINOv2 meets text: A unified framework for image- and pixel-level vision-language alignment,” in *CVPR*, 2025.
- [43] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani, “One-shot open affordance learning with foundation models,” in *CVPR*, 2024.
- [44] Y. Li, H. Mao, R. Girshick, and K. He, “Exploring plain vision transformer backbones for object detection,” in *ECCV*, 2022.
- [45] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou, “WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild,” in *CVPR*, 2025.
- [46] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, “SAM 2: Segment anything in images and videos,” in *ICLR*, 2025.