

Towards in-the-wild Egocentric 3D Hand-Object Pose Estimation

Siddhant Bansal¹ Zhifan Zhu¹ Shashank Tripathi² Jiahe Zhao¹
Michael J. Black² Dima Damen¹

¹ University of Bristol, Bristol, United Kingdom

² Max Planck Institute for Intelligent Systems, Tübingen, Germany
<https://sid2697.github.io/epic-contact>

Abstract. Estimating accurate 3D hand-object pose from in-the-wild egocentric RGB remains challenging due to severe occlusions and ambiguous contact. Existing learning-based methods often struggle to generalise to in-the-wild scenes and are limited by the scarcity of supervision. We address these issues with two contributions. First, we introduce EPIC-Contact, an in-the-wild egocentric dataset of 2.3K clips (62.3K frames) with dense, bijective 3D hand-object contact correspondences and posed meshes. Second, we propose HOPformer, an end-to-end transformer that jointly predicts bi-manual hand and object pose in a single forward pass. A cross-attention decoder conditions object features on hand priors, producing robust pose estimation. We test HOPformer on the in-lab 3D dataset, ARCTIC, as well as our newly introduced EPIC-Contact dataset. HOPformer reaches 82.4% success rate on ARCTIC (+6.2 pts over current SOTA). On EPIC-Contact, it nearly doubles the success rate while reducing contact deviation by 75%. EPIC-Contact, HOPformer code and checkpoints are released: <https://sid2697.github.io/epic-contact>.

1 Introduction

We routinely use our two hands to interact with the physical world. Everyday activities like washing dishes at the sink, grasping a bottle, or moving cookware on the stove require precise coordination between hands and objects, highlighting the remarkable dexterity and adaptability of the human hand. Modelling such interactions is essential for human-centric applications such as AR/VR, robotics, human-computer interaction/collaboration, and assistive technologies. Yet most existing work relies on controlled, scripted settings that fail to capture the complexity of real-world use [1, 6, 16, 25, 47]. In this work, we aim to recover the pose of both the hands and the object in 3D from a single forward pass, including from images sourced from unscripted egocentric videos (Fig. 1).

The task is inherently challenging due to the wide diversity of objects and hand poses commonly present in hand-object interaction scenarios. The difficulty is further compounded by strong mutual occlusions and ambiguous contact regions. Existing methods focus on controlled (in-lab) settings [2, 8, 16, 35, 41, 60] and as shown in Fig. 1 (*Right*), often struggle in the wild [6, 25, 47].

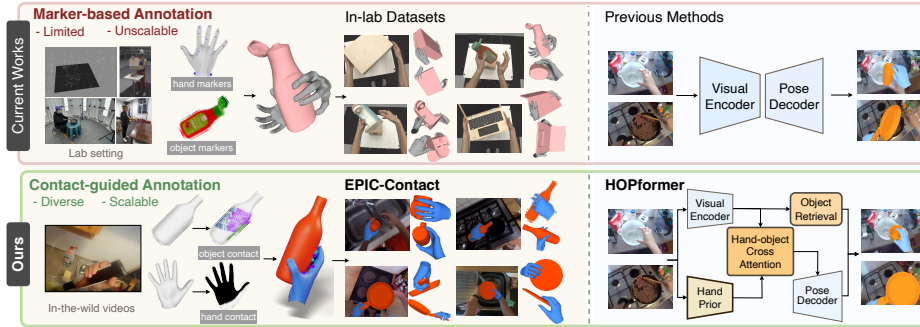


Fig. 1: (*Left*) We introduce EPIC-Contact, an in-the-wild egocentric dataset for 3D hand-object pose estimation. Unlike typical in-lab MoCap datasets that require specialised equipment and capture limited backgrounds/object instances, EPIC-Contact provides diverse, cluttered real-world interactions with posed 3D hand-object meshes derived from dense, bijective contact annotations. (*Right*) Existing learning-based approaches do not leverage strong hand priors and hence do not generalise to in-the-wild scenarios. In contrast, our proposed HOPformer network enriches object features with hand priors to achieve superior performance on in-the-wild data.

A central bottleneck is the availability of supervision. Training (or evaluating) on in-the-wild interactions requires images paired with ground-truth 3D poses of both the hands and the object. As highlighted in Fig. 1 (*Left*), such 3D ground-truth has, to date, been acquired using expensive motion capture (MoCap) [2, 16, 35, 41, 55, 60], which requires careful calibration, making it unsuitable for wider adoption. Such data typically has limited, minimalist uncluttered backgrounds that do not reflect the complexity of real-world egocentric scenes [2, 16, 35, 41].

To move beyond these limitations, we propose a novel approach to obtain 3D annotations for hand-object interaction in images, thus enabling scalable supervision. Specifically, we introduce *EPIC-Contact*, a dataset of 2.3K egocentric video clips, capturing hand-object grasps with 9 object categories, which we annotate with contact vertices and paired 3D hand-object meshes (Fig. 1, *Left*). EPIC-Contact captures diverse backgrounds and cluttered environments, where objects may be small or transparent, heavily occluded, in challenging natural interactions. Our key idea is to annotate bijective 3D contact on *both* the hands and the object, in line with prior work on full-body contact [12, 57]. Once we have bijective contact points, we use an optimisation pipeline to translate these annotations into posed 3D hand-object meshes. With this approach, we obtain 62.3K annotated frames, a one-of-a-kind dataset for training and evaluation.

While learning-based 3D hand pose estimation has progressed rapidly [3, 33, 34, 48–50, 53], performing well in diverse and challenging scenarios, the same cannot be said of joint hand-object pose estimation methods [1, 16]. Such methods fail to model the interactions between hand and object poses, impacting their performance as shown in Fig. 1 (*Right*).

Our intuition is to leverage robust data-driven 3D hand reconstruction methods [48, 49] to guide joint 3D hand and object pose estimation. We propose HOPformer, an end-to-end learning framework for joint 3D bi-manual hand-object

pose estimation from an RGB image in a single pass (Fig. 1, *Right*). Specifically, we design a cross-attention framework that injects structural hand priors into object features, creating a powerful set of interaction features. HOPformer generalises well across both in-lab [16] and in-the-wild images, and achieves state-of-the-art results. Together, EPIC-Contact and HOPformer represent a step towards scalable, robust 3D hand-object pose estimation. To summarise, our contributions are:

1. We annotate 3D hand-object contact points on 2.3K video clips from the EPIC-Kitchens [13, 74] dataset and propose EPIC-Contact. This enables training and evaluation of hand-object pose estimation methods on challenging in-the-wild images.
2. We propose a learning-based transformer model, Hand-Object Pose transformer (HOPformer), that leverages pre-trained hand priors to regress the pose of the two hands and the object in a single forward pass.
3. HOPformer achieves state-of-the-art results on ARCTIC [16] and EPIC-Contact, outperforming prior work on most metrics by large margins.

2 Related Works

Hand-Object Pose Estimation: Jointly estimating the 3D pose of hands and objects has gained significant interest in past years [1, 8, 16, 19, 24, 25, 29, 35, 56, 61]. Methods assume the object template is known - whether rigid [1, 19, 24, 25, 61, 74] or articulated [16]. While a majority of the proposed approaches are optimisation-based [6, 19, 25, 47, 61, 74], a few are learning-based [1, 16, 24, 56]. The closest works to HOPformer are ArcticNet-SF [16] and JointTransformer [1], to which we compare. Similar to HOPformer, both methods use an encoder-decoder architecture and regress both the hand and object poses. While ArcticNet-SF [16] uses a ResNet-50 [26] backbone, JointTransformer replaces it by DINOv2 [46] to achieve superior results. Different from both, HOPformer leverages hand priors to enrich features and improve results.

Another set of works explore CAD-free object reconstruction [10, 15, 21, 27, 50, 58, 59, 65–67]. While promising generalisation results are reported, we find that, in practice, the results are not satisfactory; see the Sup. Mat. for qualitative comparison with [9, 15, 65, 67]. Tackling learning-based pose estimation for CAD-free models remains a future direction.

Hand Pose Estimation: Hand pose estimation has been a topic of exploration for decades [5, 18, 20, 28, 35, 37, 38, 40, 44, 45, 51, 54]. Recently, with large 3D hand pose datasets [2, 8, 16, 30, 43, 75] and better architectures [39, 48, 49, 64, 70], HaMeR [48] and WiLoR [49] scale up model size and training data to achieve robust hand pose estimation on diverse scenes. However, unlike HOPformer, these methods only target hand pose estimation. Our key insight is to leverage these strong hand priors for joint hand and object pose estimation.

Hand-Object Contact Annotations: Most datasets for hand-object reconstruction are collected in controlled in-lab settings [2, 4, 8, 16, 17, 41, 60, 68]. MOW [6] is the only in-the-wild dataset providing paired hand-object meshes.

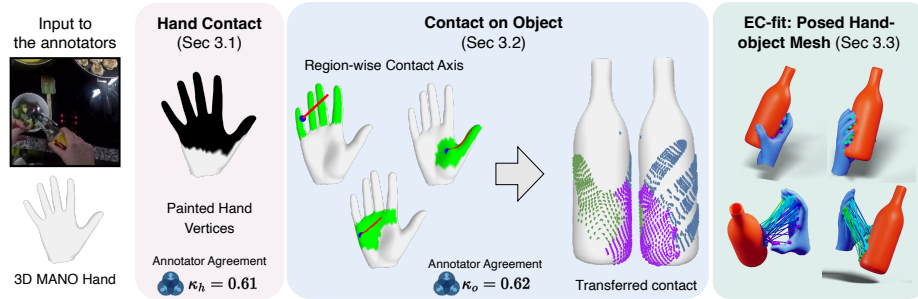


Fig. 2: EPIC-Contact annotation process. Given a hand–object interaction clip, annotators (i) paint contact vertices on a subdivided MANO hand mesh (Sec. 3.1); (ii) parametrise each contact region with a 2-DoF contact axis (blue sphere/red line) and transfer it to the object surface with two clicks per axis, yielding bijective hand–object correspondences (Sec. 3.2); and (iii) fit posed hand and object meshes with EC-fit (Sec. 3.3). The EC-fit column visualises the fitted meshes and the one-to-one vertex correspondences. We perform quantitative quality checks on annotations for each stage.

However, the ground-truth object poses in MOW are only coarsely verified, while the fine-grained contacts between the hand and the object are ignored. Body pose estimation works, on the other hand, have explored various ways to obtain contact annotations for in-the-wild images. For example, DECO [57] uses a “vertex painting” approach to label contact regions on the body. PICO [12] extends this by transferring the contact regions to objects. Motivated by this framework, we propose a pipeline to annotate hand and object contact in egocentric images.

3 EPIC-Contact Dataset

Training and evaluation for hand-object reconstruction require images paired with 3D hand and object pose labels. We collect new annotations that provide 3D posed hand-object meshes for in-the-wild egocentric images, which we refer to as the EPIC-Contact dataset. Importantly, our pipeline includes a novel approach to manually and efficiently collect bijective 3D contact correspondences. We then exploit these in an optimisation framework to estimate both hand and object poses. We describe our annotation pipeline and dataset statistics in this section.

3.1 Annotating Hand Contact Regions

We use videos from the EPIC-Grasps dataset [74] as it has challenging and diverse hand-object interactions and is paired with 3D meshes for 9 object classes. Importantly, the dataset is manually labelled with stable grasp temporal segments, where the hand maintains a steady contact with the object. This allows annotating a single frame then applying the same contact across the video.

First step is to annotate regions of the hand that are in contact with the object. Following previous work [57, 62, 63], we label contact regions on 3D meshes

using “vertex painting” tools. We annotate the hand mesh because its consistent topology (MANO [52]) simplifies the annotation task and provides a canonical representation, unlike annotating objects with varied and irregular geometries.

We create an interface (see Sup. Mat.) where we show the video of a hand in contact with the object and 3D MANO mesh [52]. Unlike prior work that relies on a single image as input [57], a video clip provides richer context for annotators, particularly as the clip allows multiple views of the contact to be observed during a consistent grasp. We ask the annotators to “paint” contact labels on $N_V = 3106$ vertices of the MANO template mesh, $\tilde{\mathcal{H}} \in \mathbb{R}^{3106 \times 3}$. Note that we uniformly subdivide the standard 778-vertex MANO mesh to $N_V = 3106$ vertices, allowing us to capture fine-grained contact regions with high precision. In case of both hands being in contact, we annotate one hand at a time, specifying the hand side for each annotation. Furthermore, the annotators are instructed to infer and label all contact regions, including those occluded in the egocentric view, using the video context and grasp motion. As shown in Fig. 2, for the bottle in the right hand, vertices are painted on the hand in black.

We validate the accuracy of our annotations using inter-annotator agreement. We calculate the Fleiss’ Kappa score (κ_h) as used in DECO [57]. As shown in Fig. 2, we report $\kappa_h = 0.61$ across 10 videos annotated by 12 annotators (compared to 0.65 in [57]). Additional details on calculation of κ_h and a figure showing the highest and lowest κ_h examples is in the Sup. Mat.

3.2 Contact Regions on Objects

Bijjective Contact Transfer: Once we have the contact vertices on the hand, the next step is to get corresponding contact vertices on the object. We need a bijective mapping between these object vertices and the hand’s contact vertices for obtaining posed hand-object meshes through optimisation.

We follow ContactEdit [36] and represent the contact regions from Sec. 3.1 with a contact “axis”. This allows us to parametrise each region with a 2-DoF axis, which reduces the problem of mapping the contact region onto the object’s surface to transferring this axis with two clicks: the start of the contact axis and its direction. Using this axis, we can transfer the contact region to the object while preserving the correspondence. This pipeline is implemented in an interactive web-based tool.

A key challenge is to identify which contact regions on the hand to parametrise for transfer. Treating each hand vertex independently is extremely tedious and expensive. At the other extreme, using a single contact axis for the entire hand is unintuitive and prone to annotation mistakes. To balance speed, convenience and accuracy, we divide the hand into three regions: the thumb, the four fingers, and the palm (see Fig. 2). We estimate the spacing between fingers using the pose parameters from WiLoR [49]. Therefore, for each clip, we have up to three contact axes (depending on contact regions) to transfer, *i.e.*, with at most six clicks, we get the full 3D hand-object contact correspondences. Further details on the interface are in the Sup. Mat.

Scaling the Objects: The object meshes from [74] have a standard scale. However, for transferring the contact region correctly, the scale of the mesh should match that of the object in the image. We utilise a VLM (Gemini 2.5 [11]) to estimate the scale using a class-specific prompt. We prompt the VLM to estimate multiple degrees-of-scale for non-uniform scaling, rather than a single isotropic scale factor. For example, for a “pan”, we query for both its diameter and its handle length. This allows us to scale the various parts of the template object mesh to more accurately match the instance in the video.

To verify these scale predictions, we sample 30 objects covering all 9 classes and manually compare these to ground truth object sizes. We identify objects of a known brand (*e.g.* a specific bottle of oil) and measure the dimensions of the same physical object. This allows us to evaluate the VLM scale estimates against ground truth dimensions, achieving 0.94 cm MAE (5.9% relative error) with 82.5% of samples falling within $\pm 10\%$ of the true dimensions. Additional details on prompts used, degrees-of-scale, and verification analysis are in Sup. Mat.

Validating the object contact annotations: We compute the inter-annotator agreement (κ_o) as 0.62 on 10 videos annotated by 4 annotators. This confirms that the annotators consistently map hand regions to object surfaces and are able to understand contact despite monocular ambiguities. Note that κ for bijective correspondences on object is not reported in [12, 57] for direct comparison. Additional details along with visualisation are in Sup. Mat.

3.3 EC-fit Pipeline: From Contact to Posed Hand-Object Meshes

Finally, we build EPIC-Contact Fitting (EC-fit), an optimisation-based pipeline to derive the posed 3D hand-object interaction meshes from contact annotations. EC-fit aligns the mesh geometry with contact constraints and robustly models the interactions captured in the video space.

Pose Initialisation: We initialise the 3D hand pose by applying WiLoR [49] to the central frame I of the clip, yielding a MANO mesh $\mathcal{H}$ with pose Θ_0 . For initialising the object pose, we diverge from previous works [12, 22] that utilise standard single-pose initialisation, and find it necessary to use multiple initialisations, including random poses and category-aware pose priors from [73].

Contact-based Alignment: Firstly, we leverage the contact information to guide the object pose alignment with the hand. This is formulated as an optimisation problem. Given the set of bijective vertex pairs $\mathbb{C} := \{(h_i, o_i)\}$ where $h_i \in \mathcal{H}$ and $o_i \in \mathcal{O}$ are the hand and object vertices in contact, we optimise the object rotation $r_o \in \mathbb{R}^6$ and translation $t_o \in \mathbb{R}^3$ by minimising the contact loss: $\mathcal{L}_{con} = \frac{1}{|\mathbb{C}|} \sum_{i=1}^{|\mathbb{C}|} \|h_i - o_i\|_2$.

Image-guided Refinement: We use the central frame of the clip (I) and an **occlusion-aware mask loss** to refine the hand and object poses. Traditional approaches [12, 23] render the predicted object mesh to a 2D mask $\hat{M}_o$, and directly align it with the object mask M_o [14] in the image frame. However, masks are incomplete when objects are occluded, which is common in our dataset (*e.g.* plates with food). To this end, we identify an occlusion mask M_{occ} that

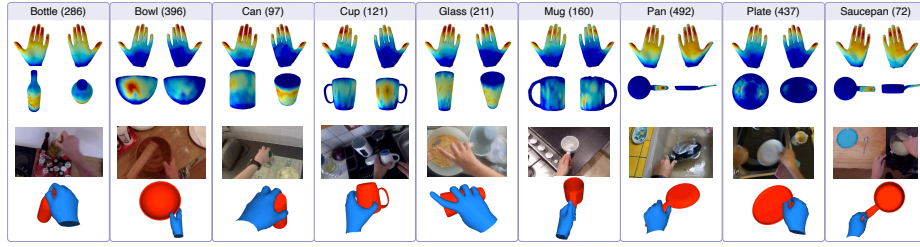


Fig. 3: EPIC-Contact dataset. For each object category (# of annotated clips). Middle: contact-frequency heatmaps on the canonical MANO hand mesh and object template mesh. Bottom: example frames with EC-fit posed hand and object meshes.

includes both hand-object occlusions and inter-object occlusions. Our occlusion-aware mask loss only aligns object regions that are not occluded by excluding M_{occ} : $\mathcal{L}_m^o = 1 - IoU([\hat{M}_o \setminus M_{occ}], [M_o \setminus M_{occ}])$. Moreover, to enforce physical realism, a penetration loss $\mathcal{L}_p$ is added to prevent hand-object interpenetration.

Next, we update the hand pose, minimising a hand mask loss $\mathcal{L}_m^h$ using the rendered hand mask $\hat{M}_h$ and update the MANO pose Θ . A regularisation loss $\mathcal{L}_{reg} = \|\Theta - \Theta_0\|_2$ prevents the refined pose from deviating significantly from the initial pose prediction. We also retain the penetration loss $\mathcal{L}_p$ and contact loss $\mathcal{L}_{con}$. We manually verify the posed meshes, and correct errors identified by human annotators manually. Details of the manual correction are in Sup. Mat.

Clip-level annotations: So far, we operate at the frame level. EPIC-Grasps [74] provides clips with a stable grasp and EC-fit yields the object pose relative to the hand in the central frame of the stable grasp, this allows us to *automatically* broadcast the relative object-to-hand pose across the clip. We obtain the clip-level object-to-camera poses leveraging hand poses from [49]. In case of temporal jitter, frames are labelled with lower confidence.

To summarise, EPIC-Contact consists of egocentric interactions with bijective hand-object contact correspondences and posed 3D hand-object meshes, collected via hand contact painting, contact transfer, and EC-fit with quality checks. Figure 3 shows the number of samples per object class, heatmaps of aggregate contact on hand and object, showing diversity of grasps as well as a sample of posed hand and object meshes. In the following sections we describe how we leverage these annotations for training and evaluation.

4 HOPformer: Hand-Object Pose transformer

In this section, we describe HOPformer, a learning-based network for estimating hand and object poses. Figure 4 provides an overview. HOPformer can be trained using either 3D ground truth or posed hand-object meshes obtained from in-the-wild datasets (Sec. 3).

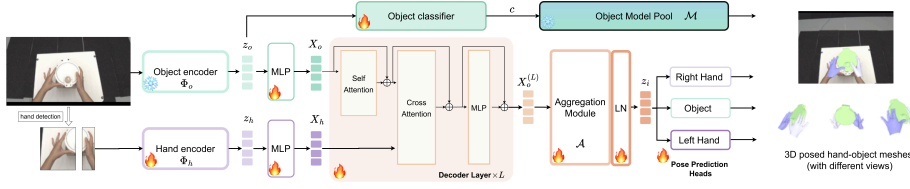


Fig. 4: Overview of HOPformer for learning-based hand-object pose estimation. Given an image with hand-object interaction, HOPformer conditions object features on features from hand pose encoder using an L -layer transformer decoder ($\mathcal{D}_\theta$). The learned features after the aggregation module $\mathcal{A}$ are used to estimate the pose of both hands and interacting object, through dedicated output heads. The object head estimates rotation and translation relative to a canonical object pose. Furthermore, we predict the object’s class c and retrieve the object mesh $\mathcal{O}_c$ from a model pool $\mathcal{M}$. The estimated pose is applied to the retrieved object mesh.

4.1 Overview and Notations

Problem Formulation: Given an image containing one object out of a set of known categories, being manipulated by one or both hands or visible without active manipulation (*e.g.*, resting on a surface), we estimate the pose of each visible hand and the object. HOPformer learns to predict hand and object poses across diverse manipulation patterns, including articulated objects when present.

Notations: We denote the i^{th} frame in k^{th} video as f_k^i . We use r , l , and o to denote the right hand, left hand, and object, respectively. For hands we use the parametric MANO model [52] to represent the hand’s pose and shape by $\Theta = \{\theta, \beta\}$. MANO maps Θ to a 3D posed and shaped mesh $\mathcal{H}(\Theta) \in \mathbb{R}^{778 \times 3}$, where $\theta \in \mathbb{R}^{48}$ (including global orientation of the hand) and $\beta \in \mathbb{R}^{10}$. For each object, we predict the object pose ω , which consists of rotation $\mathbf{R}_o \in \mathbb{R}^6$ [72], translation $\mathbf{T}_o \in \mathbb{R}^3$, and 1D rotation in radians for objects with articulation. Given this pose ω , we output a posed 3D mesh, $\mathcal{O}_c(\omega) \in \mathbb{R}^{|V_c| \times 3}$ where $|V_c|$ denotes the number of vertices of the object c .

4.2 Learning-based Pose Estimation

Our goal is a learning-based model for joint hand-object pose estimation, where the estimate of one can inform the other. During interactions, this context is particularly helpful under occlusion.

We first extract a generic sequence of object tokens z_o (via Φ_o) and a specialised sequence of hand pose tokens z_h (via Φ_h). These are linearly projected into a query sequence ($X_o = W_o z_o$) and memory sequence ($X_h = W_h z_h$), and passed into an L -layer Transformer Decoder $\mathcal{D}_\theta$. We use a full decoder stack (rather than a single cross-attention layer) to iteratively refine the object features. As shown in Fig. 4, in each of the L layers, the object representations:

1. communicate via self-attention, 2. attend to hand context X_h via cross-at-

tention, and 3. pass through a position-wise feed-forward network. This yields a refined object sequence $X_o^{(L)} = \mathcal{D}_\theta(X_o, X_h)$.

To obtain the final, aggregated interaction features z_i , this output is passed through a residual connection, followed by a learnable linear aggregation module $\mathcal{A}$ and layer normalisation as: $z_i = \text{LayerNorm}(\mathcal{A}(X_o^{(L)} + X_o))$. The resulting z_i is not merely ‘fused’, but is a representation where object features have been progressively modulated by the hand pose. The core cross-attention mechanism within any layer ℓ of our decoder $\mathcal{D}_\theta$ is defined as

$$\text{CA}(X_o^{(\ell-1)}, X_h) = \delta \left(\frac{(X_o^{(\ell-1)} W_Q^\ell)(X_h W_K^\ell)^\top}{\sqrt{d_k}} \right) (X_h W_V^\ell) \quad (1)$$

where $X_o^{(\ell-1)}$ is the object query sequence input to that layer (with $X_o^{(0)} = X_o$), δ is the Softmax operation, and W_Q^ℓ , W_K^ℓ , and W_V^ℓ are the learnable projection matrices for the query, key, and value, respectively.

Finally, as shown in Fig. 4, the feature sequence z_i is routed to a set of dedicated prediction heads, which are implemented as small, independent MLPs. Following the design in [7], different tokens from the sequence z_i are disentangled and learned to predict different components of the hand and object pose. The first 16 tokens are used to regress θ_r (with global orientation), the 17th token the root, and the 18th token is used to get the β_r . The next 18 tokens are used to regress pose, orientation (θ_l), root, and shape (β_l) for the left hand. The remaining three tokens are used to regress the articulated object pose ω .

4.3 Object Mesh Retrieval

The primary decoder regresses the object’s pose ω by utilising interaction features, without being hard-coded to a specific object’s 3D geometry. However, to render the 3D hand-object reconstruction, the specific mesh geometry $\mathcal{O}_c$ is required. We learn to retrieve the correct mesh $\mathcal{O}_c$ from a predefined model pool, $\mathcal{M}$. We leverage the rich, semantic information encoded in the object features z_o from Φ_o , and add a classification head to predict the object’s category, c .

The classifier is trained jointly with the pose estimators. At inference, the predicted class c is used to retrieve the 3D mesh $\mathcal{O}_c$ from $\mathcal{M}$. The posed mesh is then generated by applying the regressed pose ω to the retrieved mesh. This design efficiently reuses features and cleanly decouples the regression of pose from the identification of object class.

4.4 Training Signal

To train HOPformer, we use various frame-level losses. For hands, we utilise the following losses: $\mathcal{L}_h = \lambda_{2D}^h \mathcal{L}_{2D}^h + \lambda_{3D}^h \mathcal{L}_{3D}^h + \lambda_\theta^h \mathcal{L}_\theta^h + \lambda_\beta^h \mathcal{L}_\beta^h + \lambda_T^h \mathcal{L}_T^h$ where $h = \{r, l\}$ stands for handedness, $\mathcal{L}_{3D}$ is a supervised loss on the 3D joints (after subtracting the root), $\mathcal{L}_\theta^h$ and $\mathcal{L}_\beta^h$ are losses on MANO pose and shape parameters, $\mathcal{L}_{2D}^h$ is a 2D loss on the projected 3D points in the image, and $\mathcal{L}_T^h$ is the loss on the

weak-perspective camera parameters. Finally, λ^h are the weighting coefficients for each of the losses when optimising the hand poses.

Similarly, for the object, there are losses for 3D keypoints $\mathcal{L}_{3D}^o$, 2D projection of 3D keypoints $\mathcal{L}_{2D}^o$, weak-perspective camera parameters, $\mathcal{L}_T^o$, classification $\mathcal{L}_c^o$, and pose $\mathcal{L}_\omega^o$ on the object: $\mathcal{L}_o = \lambda_{3D}^o \mathcal{L}_{3D}^o + \lambda_{2D}^o \mathcal{L}_{2D}^o + \lambda_T^o \mathcal{L}_T^o + \lambda_c^o \mathcal{L}_c^o + \lambda_\omega^o \mathcal{L}_\omega^o$.

For frames where the hand is in contact with an object, we add a CDev-based interaction loss (details in Sec. 5.3): $\mathcal{L}_{int} = \lambda_{ro}^{int} \mathcal{L}_{ro}^{int} + \lambda_{lo}^{int} \mathcal{L}_{lo}^{int}$. The final loss term for training HOPformer is then: $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_l + \mathcal{L}_o + \mathcal{L}_{int}$. The losses above use MSE except for $\mathbf{R}_o$ in $\mathcal{L}_\omega^o$ which uses geodesic loss.

5 Experiments and Results

5.1 Datasets

ARCTIC [16]: This in-the-lab dataset is captured in a constrained setting where a subject is recorded manipulating one object. It consists of 2.1M RGB frames across 9 camera views where 10 participants manipulate 11 objects with both hands. Participants either “use” or “grasp” the object during the interactions. The poses of the hands and objects are captured using a MoCap system. ARCTIC offers 8 exocentric views for pre-training and 1 egocentric view to fine-tune and evaluate. We follow the original train and validation splits [16]. As this dataset uses MoCap ground-truth, we use it for ablation experiments.

EPIC-Contact: In EPIC-Contact (which we collect following the details in Sec. 3) we have 2,272 videos consisting of 62.3K frames across 9 objects. Note that the same posed hand-object mesh is used for training all frames within the same video clip, as these clips are temporally annotated as a continuous stable grasp [74]. We use 237 videos for testing and 2,035 videos as training, keeping the distribution of hand side and object categories identical between the sets.

5.2 Implementation Details

For all experiments, we use DINOv2 [46] with ViT-G backbone [69] as our object feature extractor (Φ_o) and WiLoR [49] as Φ_h for hand features. The decoder depth L is set to 12. The hyper-parameters λ s in the loss terms are all set to 1 except for $\lambda_{2D}^h = 5.0$, $\lambda_{3D}^r = \lambda_{3D}^o = 5.0$, $\lambda_\theta^h = \lambda_{3D}^l = 10.0$ and $\lambda_\beta^h = \lambda_c^o = 0.001$.

We train using the AdamW [42] optimiser. Following [16], HOPformer is trained in two stages, the first stage is trained on exocentric views and then fine-tuned on egocentric views from ARCTIC. While training on exocentric views, we use a linear warm-up of the learning rate, from $1e-7$ to $5e-5$ in first 5% of steps, and cosine decay from $5e-5$ to $1e-7$ for the remaining steps. The batch size is 256 across 4 NVIDIA GH200 GPUs and we train for 25 epochs. For egocentric training, we use a learning rate of $3e-5$ with cosine decay from $1e-7$, batch size of 128 across 4 GPUs, trained for 30 epochs with early stopping. The same hyper-parameters are used for training on EPIC-Contact for 125 epochs with early stopping. We use a weak perspective camera model [3, 16, 31, 32, 71] for translation and a 6D representation for rotation [72].

Table 1: ARCTIC (Egocentric; in-lab). HOPformer outperforms baselines by a clear margin on all metrics (except AAE). **Bold** numbers are best performance.

Method	Contact and Relative Positions			Motion		Hand	Object		
	CDev [mm] ↓	MRRPE _{rl/ro} [mm] ↓	MDev [mm] ↓	ACC _{h/o} [m/s ²] ↓	MPJPE [mm] ↓	AAE [°] ↓	SR@0.05 [%] ↑	Cls [%] ↑	
ArcticNet-SF [16]	44.1	33.9 / 36.8	11.8	6.3 / 11.3	22.9	8.0	59.0	-	
JointTransformer [1]	35.0	34.0 / 29.9	10.4	6.9 / 10.1	20.0	4.9	76.2	-	
HOPformer (<i>ours</i>)	31.9	31.1 / 29.4	7.3	4.8 / 6.2	16.1	5.0	82.4	99.5	

5.3 Quantitative Metrics

We follow [16] and report metrics capturing contact/relative pose, motion, hand accuracy, and object pose estimation. 1. **Contact Deviation (CDev, mm)**: The mean distance between corresponding hand-object contact vertex pairs; 2. **Mean Relative-Root Position Error (MRRPE_{rl/ro}, mm)**: The relative root translation error for hand-hand and hand-object; 3. **Motion Deviation (MDev, mm)**: Measuring disagreement in motion of vertices in stable contact across consecutive frames; 4. **Acceleration Error (ACC_{h/o}, m/s²)**: Measuring smoothness via acceleration differences for hand/object vertices; 5. **Mean Per-Joint Position Error (MPJPE, mm)**: The mean 3D error over 21 hand keypoints; 6. **Average Articulation Error (AAE, °)**: Articulation error for articulated objects; 7. **Success Rate (SR@0.05/SR@0.1, %)**: The fraction of object vertices within 5%/10% of the object diameter; 8. **Object Classification Accuracy (Cls)**: The accuracy of the object classification head. For EPIC-Contact, where symmetric objects are used (*e.g.* bottle, can), CDev, MDev, ACC and Success Rate are updated to be symmetry aware. Details in Sup. Mat.

5.4 Baselines

We use learning-based baselines closest to HOPformer. 1. **ArcticNet-SF [16]**: Encoder-decoder architecture with ResNet-50 [26] as its backbone. 2. **Joint-Transformer [1]**: Builds on ArcticNet-SF and replaces the CNN-based backbone with DINOv2 [46]. As this uses the same feature backbone as us, it also serves as the best baseline for direct comparison. Improvements over this baseline are results of our proposed HOPformer design choices. Importantly, JointTransformer remains the current SOTA method on the ARCTIC dataset reconstruction leaderboard. For fair comparison, we retrain both methods on ARCTIC and EPIC-Contact using the same splits and evaluation protocol as ours.

5.5 Results

In-lab ARCTIC Results: As shown in Tab. 1, HOPformer achieves the best performance on ARCTIC’s egocentric split, improving contact consistency, motion, and object pose estimation over prior learning-based methods. Compared to JointTransformer [1], we reduce CDev from 35.0→31.9 mm and MDev from

Table 2: EPIC-Contact (Egocentric; in-the-wild). HOPformer improves interaction and object pose estimation over all baselines. **Bold** numbers are best performance. MRRPE_{rl} is invalid for EPIC-Contact due to one hand per sequence.

Method	Contact and Relative Positions			Motion		Hand	Object		
	CDev [mm] ↓	MRRPE _{ro} [mm] ↓	MDev [mm] ↓	ACC _{h/o} [m/s ²] ↓	MPJPE [mm] ↓	SR@0.05 [%] ↑	SR@0.1 [%] ↑	Cls [%]	
ArcticNet-SF [16]	94.2	166.7	70.1	3.9 / 5.3	44.8	16.5	55.9	-	
JointTransformer [1]	30.1	78.6	20.0	3.1 / 9.5	22.9	17.6	56.9	-	
HOPformer (ours)	20.7	65.8	11.4	2.5 / 4.1	19.9	29.8	69.7	52.9	

10.4→7.3 mm, while improving SR@0.05 from 76.2→82.4; hand accuracy also improves substantially (MPJPE 20.0→16.1 mm), with comparable articulation error (AAE 4.9→5.0). We also present results on the exocentric split in Sup. Mat., where HOPformer outperforms JointTransformer by an equal or larger margin on most metrics. These gains establish HOPformer as the new SOTA on this established benchmark and support our central design choice of conditioning object features on strong hand priors for robust hand-object pose estimation.

In-the-wild EPIC-Contact Results: Table 2 shows that HOPformer improves pose-estimation and interaction consistency on EPIC-Contact, outperforming learning-based baselines under in-the-wild occlusions and clutter. Compared to JointTransformer [1], HOPformer reduces CDev from 30.1 → 20.7 mm and MDev from 20.0 → 11.4 mm, while improving SR@0.05 from 17.6 → 29.8; hand accuracy also improves (MPJPE 22.9 → 19.9 mm). We additionally predict object category to enable fully automatic inference; however, to match prior work that assumes oracle object meshes, we report pose with oracle mesh and provide fully automatic results in the Sup. Mat.

These results also highlight the additional challenge of EPIC-Contact compared to established benchmarks, as expected. For example, JointTransformer drops SR@0.05 from 76.2% on ARCTIC (Ego) to 17.6% on EPIC-Contact. This is not unexpected given ARCTIC shares the same object instances between train and val split, while EPIC-Contact shows novel instances of the known object categories, often transparent, occluded and in a cluttered scene.

Qualitative Results: We demonstrate qualitative comparison of models’ predictions on EPIC-Contact in Fig. 5. HOPformer estimates plausible hand and object pose under heavy occlusion, for diverse objects in challenging scenes. Prior works fail in most cases, even in securing contact between the hand and the object. Failure case (bottom) is impacted by strong priors from learnt data, where a plate is expected to be flat. Additional results are provided in the Sup. Mat.

5.6 Ablations

All ablations are reported on the ARCTIC dataset consisting of 3D MoCap ground truth (egocentric split).

Ablating the losses: In Tab. 3, we ablate each training term and observe consistent drops, with the largest effects coming from object supervision and interaction constraints. Removing the object losses ($\mathcal{L}_o$) reduces SR@0.05 from

Table 3: Architectural and loss ablations on ARCTIC (egocentric). We ablate HOPformer components and training objectives. Results show that hand-conditioned cross-attention decoding and the interaction/object supervision terms are key to performance across pose-estimation, contact, and motion metrics.

Setting	Contact and Relative Positions		Motion		Hand	Object	
	CDev [mm] ↓	MRRPE _{rt/ro} [mm] ↓	MDev [mm] ↓	ACC _{h/o} [m/s ²] ↓	MPJPE [mm] ↓	AAE [°] ↓	SR@0.05 [%] ↑
Loss ablations							
w/o Interaction Loss ($\mathcal{L}_{int}$)	40.7	33.3 / 36.5	10.8	6.2 / 8.4	18.3	7.7	79.2
w/o 3D keypoints ($\mathcal{L}_{3D}^{o,h}$)	36.8	33.7 / 33.8	10.0	6.1 / 8.7	18.7	7.8	78.5
w/o 2D keypoints ($\mathcal{L}_{2D}^{o,h}$)	35.6	35.9 / 33.0	9.5	5.9 / 8.1	18.3	7.4	79.5
w/o Beta ($\mathcal{L}_\beta^h$)	35.4	33.2 / 32.6	9.6	5.8 / 8.4	18.1	7.2	79.3
w/o Translation ($\mathcal{L}_T^{o,h}$)	35.0	32.8 / 31.5	9.6	5.9 / 8.3	18.0	7.2	79.6
w/o Object Pose ($\mathcal{L}_o^o$)	34.5	31.7 / 31.0	9.8	5.7 / 9.5	17.3	7.4	72.5
w/o Object Losses ($\mathcal{L}_o$)	33.8	31.3 / 31.4	9.5	5.6 / 9.4	17.3	8.6	66.8
Architecture ablations							
Concat + MLP	74.4	48.1 / 66.5	32.0	6.6 / 25.5	19.6	18.7	29.4
DINOv2 as hand features	44.2	36.8 / 37.1	16.2	7.4 / 13.0	20.6	6.3	70.0
No SA	61.3	74.0 / 60.4	17.9	7.3 / 14.4	22.9	10.9	55.8
No Aggr. Module ($\mathcal{A}$)	40.4	35.9 / 36.6	14.0	6.0 / 11.4	16.8	6.3	74.0
$L = 1$	101.6	93.5 / 90.8	34.9	15.0 / 25.4	46.7	23.4	21.9
$L = 6$	59.2	54.0 / 49.4	19.5	9.8 / 15.2	24.0	8.2	58.6
$L = 9$	50.0	47.1 / 42.8	15.7	6.8 / 11.5	21.5	7.9	69.0
Main result							
HOPformer (<i>ours</i>)	31.9	31.1 / 29.4	7.3	4.8 / 6.2	16.1	5.0	82.4

82.4 $\rightarrow$ 66.8 and increases AAE from 5.0 $\rightarrow$ 8.6, while removing the object pose term ($\mathcal{L}_o^o$) also notably reduces SR (82.4 $\rightarrow$ 72.5). Removing the CDev-based interaction loss ($\mathcal{L}_{int}$) primarily hurts physical consistency, increasing CDev from 31.9 $\rightarrow$ 40.7 and MDev from 7.3 $\rightarrow$ 10.8. For hand accuracy, keypoint supervision is most critical: removing $\mathcal{L}_{o,h}^{3D}$ yields the largest MPJPE increase (16.1 $\rightarrow$ 18.7), with $\mathcal{L}_{o,h}^{2D}$ and $\mathcal{L}_{int}$ showing similar degradation ($\rightarrow$ 18.3), whereas $\mathcal{L}_\beta^h$ and $\mathcal{L}_{o,h}^T$ provide smaller but consistent gains.

Architectural Ablations: Also in Tab. 3, we ablate the architectural design of HOPformer. **Concat + MLP.** Replacing the Cross-Attention decoder with naive fusion causes a sharp collapse (SR 82.4 $\rightarrow$ 29.4, CDev 31.9 $\rightarrow$ 74.4), despite using the same strong backbones. This confirms that HOPformer’s gains come from conditioning features and not from the backbone. **DINOv2 as hand features.** Using generic DINOv2 tokens instead of WiLoR weakens the hand prior (MPJPE 16.1 $\rightarrow$ 20.6) and propagates to object quality (SR 82.4 $\rightarrow$ 70.0). This highlights that pose-specialised hand features are key to effective hand-conditioned object refinement. **No self-attention.** Disabling self-attention, which enriches the object features, negatively impacts token communication, leading to large degradations (SR 82.4 $\rightarrow$ 55.8, MRRPE 31.1/29.4 $\rightarrow$ 74.0/60.4). This indicates that token-to-token interaction is critical for accurate object estimation. **No Aggregation Module.** Removing the learnable aggregation module, and instead using a fixed subset of the tokens for each output head, consistently hurts performance (SR 82.4 $\rightarrow$ 74.0, CDev 31.9 $\rightarrow$ 40.4). This indicates that HOPformer benefits from learnt pooling into a compact interaction representation. **Varying decoder depth.** Reducing decoder depth sharply reduces performance ($L = 1$: SR 21.9), while increasing

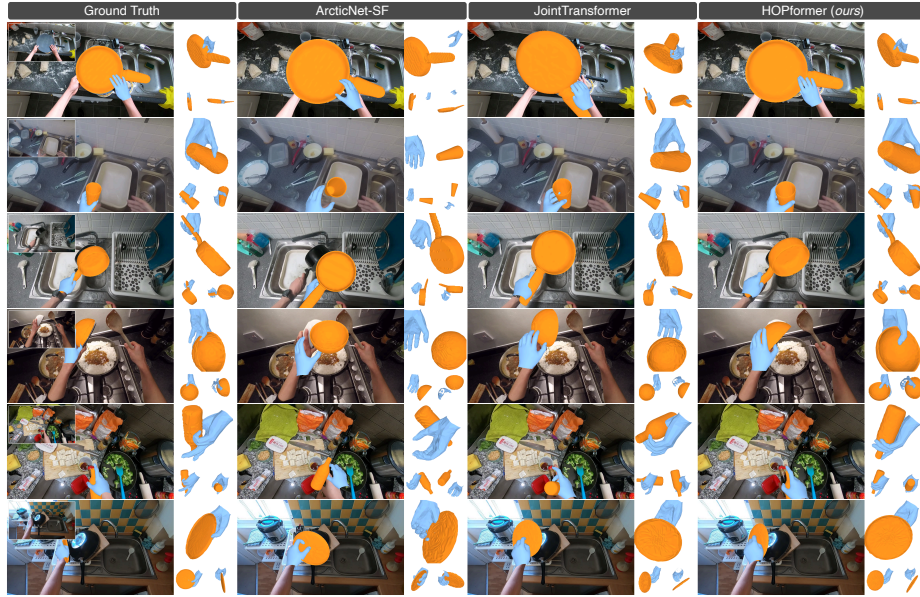


Fig. 5: Qualitative comparison with baselines. Original image (thumbnail), image with projected hand-object mesh, and three views of the posed meshes. HOPformer has visibly superior estimations of pose compared to the baselines.

depth steadily improves it up to $L = 12$ (SR 82.4). This supports our design choice of an L -layer decoder for iterative refinement of object tokens under hand guidance.

Overall, these ablations confirm that HOPformer’s improvements are obtained by iterative, hand-conditioned object refinement.

Scaling the number of object categories:

To evaluate HOPformer’s performance as the number of objects increases, we train HOPformer on $N \in \{3, 6, 9, 11\}$ object classes from ARCTIC. Fig. 6 shows a line plot with $MDev(ho)$ score as number of objects increases. HOPformer consistently benefits as more object categories are added highlighting superior generalisability despite the increase in the challenge. Other metrics show similar gains and are in the Sup. Mat.

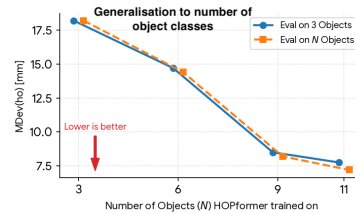


Fig. 6: MDev(ho) improves as the number of objects increases.

Training HOPformer from scratch on EPIC-Contact: Table 4 shows initialising HOPformer with ARCTIC weights yields consistent gains on EPIC-

Table 4: Cross-dataset transfer (ARCTIC $\rightarrow$ EPIC-Contact). Training HOPformer from scratch on EPIC vs. ARCTIC-initialised fine-tuning.

Method	Contact and Relative Positions		Motion		Hand	Object	
	CDev [mm] $\downarrow$	MRRPE _{ro} [mm] $\downarrow$	MDev [mm] $\downarrow$	ACC _{h/o} [m/s ²] $\downarrow$	MPJPE [mm] $\downarrow$	SR@0.05 [%] $\uparrow$	SR@0.1 [%] $\uparrow$
HOPformer (scratch)	24.4	99.5	19.5	3.5 / 7.1	54.8	3.9	24.2
HOPformer (ARCTIC init; <i>ours</i>)	20.7	65.8	11.4	2.5 / 4.1	19.9	29.8	69.7

Contact, notably in relative pose and motion (MRRPE_{ro} 99.5 $\rightarrow$ 65.8, MDev 19.5 $\rightarrow$ 11.4). HOPformer benefits from additional data during the pre-training. **Compute/Runtime:** On one NVIDIA GH200, with batch=1 for online evaluation, HOPformer runs in 106.63 ms median per frame at batch=1 (9.12 frame/s); additional compute and profiling details are provided in the Sup. Mat.

6 Conclusion

This paper contributes to 3D hand-object pose estimation for in-the-wild egocentric images in two ways.

First, we enable training and evaluation in-the-wild by annotating and releasing the EPIC-Contact dataset. We annotate EPIC-Contact video clips with hand contact regions and bijective contact on 2.3K egocentric clips with functional interactions involving 9 object classes. EPIC-Contact is quality checked, diverse and a significantly challenging benchmark to guide future research.

Second, we propose HOPformer – a learning-based approach that, given an RGB image, predicts the pose of hands and object in a single forward pass. HOPformer uses hand priors that enrich the representation, providing superior results in joint estimation of hand, object class, and object pose. We outperform SOTA on the established ARCTIC benchmark and provide a strong starting point for evaluation on EPIC-Contact.

Acknowledgement

This work was supported by EPSRC Programme Grant Visual AI (EP/T028572/1) and EPSRC Fellowship UMPIRE (EP/T004991/1). S Bansal is supported by a Charitable Donation to UoB from Meta. Z Zhu and J Zhao are supported by UoB-CSC Scholarships. While MJB is employed by Epic Games, this work was performed solely at, and funded solely by, the Max Planck Society. We acknowledge the usage of GPU Node hours granted as part of the AIRR Innovator project “5D Hand-Object Interaction Modelling from In-the-wild Videos” (Mar 2026 - Sep 2026), AIRR Gateway project “HOI Foundational Model from Egocentric Data” (Dec 2025 - Mar 2026) and the Sovereign AI Unit call project “Gen Model in Ego-sensed World” (Aug 2025 - Nov 2025).

The authors would like to thank Jacob Chalk, Tomoya Yoshida, Kranti Kumar Parida, Omar Emara, and Balamurugan Thambiraja for their comments on the manuscript. We thank Rajan, Saikiran, Durga Prasad and their team from Elancer for assisting with annotating the EPIC-Contact dataset.

References

1. Abou Zeid, K.: JointTransformer: Winner of the HANDS’2023 ARCTIC Challenge @ ICCV (2023), <https://github.com/kabouzeid/JointTransformer> 1, 2, 3, 11, 12, 26
2. Banerjee, P., Shkodrani, S., Moulon, P., Hampali, S., Han, S., Zhang, F., Zhang, L., Fountain, J., Miller, E., Basol, S., Newcombe, R., Wang, R., Engel, J.J., Hodan, T.: HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. In: CVPR (2025) 1, 2, 3
3. Boukhayma, A., Bem, R.d., Torr, P.H.: 3d hand shape and pose from images in the wild. In: CVPR (2019) 2, 10
4. Brahmbhatt, S., Tang, C., Twigg, C.D., Kemp, C.C., Hays, J.: ContactPose: A dataset of grasps with object contact and hand pose. In: ECCV (2020) 3, 35
5. Cai, Y., Ge, L., Cai, J., Yuan, J.: Weakly-supervised 3D hand pose estimation from monocular RGB images. In: ECCV (2018) 3
6. Cao, Z., Radosavovic, I., Kanazawa, A., Malik, J.: Reconstructing hand-object interactions in the wild. In: ICCV (2021) 1, 3
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: ECCV (2020) 9
8. Chao, Y.W., Yang, W., Xiang, Y., Molchanov, P., Handa, A., Tremblay, J., Narang, Y.S., Van Wyk, K., Iqbal, U., Birchfield, S., Kautz, J., Fox, D.: DexYCB: A benchmark for capturing hand grasping of objects. In: CVPR (2021) 1, 3
9. Chen, X., CHU, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., Lin, A., Liu, J.W., Ma, Z., Sagar, A., Song, B., Wang, X., Yang, J., Zhang, B., Dollár, P., Gkioxari, G., Feiszli, M., Malik, J.: SAM 3D: 3Dfy Anything in Images. In: CVPR (2026) 3, 25
10. Chen, Z., Potamias, R.A., Chen, S., Schmid, C.: HORT: Monocular hand-held objects reconstruction with transformers. In: ICCV (2025) 3
11. Comanici, G., Bieber, E., Schaekermann, M., Sachdeva, N., et al.: Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. arXiv preprint arXiv:2507.06261 (2025) 6, 27
12. Cseke, A., Tripathi, S., Dwivedi, S.K., Lakshmipathy, A., Chatterjee, A., Black, M.J., Tzionas, D.: PICO: Reconstructing 3D people in contact with objects. In: CVPR (2025) 2, 4, 6, 32
13. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Ma, J., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. In: IJCV (2022) 3
14. Darkhalil, A., Shan, D., Zhu, B., Ma, J., Kar, A., Higgins, R., Fidler, S., Fouhey, D., Damen, D.: EPIC-KITCHENS VISOR Benchmark: Video Segmentations and Object Relations. In: NeurIPS (2022) 6, 26
15. Fan, Z., Parelli, M., Kadoglou, M.E., Chen, X., Kocabas, M., Black, M.J., Hilliges, O.: Hold: Category-agnostic 3d reconstruction of interacting hands and objects from video. In: CVPR (2024) 3, 24
16. Fan, Z., Taheri, O., Tzionas, D., Kocabas, M., Kaufmann, M., Black, M.J., Hilliges, O.: ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In: CVPR (2023) 1, 2, 3, 10, 11, 12, 20, 21, 22, 26
17. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In: CVPR (2018) 3, 35

18. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3D hand shape and pose estimation from a single rgb image. In: CVPR (2019) [3](#)
19. Grady, P., Tang, C., Twigg, C., Vo, M., Brahmabhatt, S., Kemp, C.: ContactOpt: Optimizing contact to improve grasps. In: CVPR (2021) [3](#)
20. Guo, Z., Zhou, W., Wang, M., Li, L., Li, H.: HandNeRF: Neural radiance fields for animatable interacting hands. In: CVPR (2023) [3](#)
21. Hampali, S., Hodan, T., Tran, L., Ma, L., Keskin, C., Lepetit, V.: In-hand 3d object scanning from an rgb sequence. In: CVPR (2023) [3](#)
22. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3d annotation of hand and object poses. In: CVPR (2020) [6](#), [35](#)
23. Han, G., Zhai, W., Yang, Y., Cao, Y., Zha, Z.J.: Touch: Text-guided controllable generation of free-form hand-object interactions. In: arXiv preprint arXiv:2510.14874 (2025) [6](#)
24. Hasson, Y., Tekin, B., Bogu, F., Laptev, I., Pollefeys, M., Schmid, C.: Leveraging Photometric Consistency Over Time for Sparsely Supervised Hand-Object Reconstruction. In: CVPR (2020) [3](#)
25. Hasson, Y., Varol, G., Schmid, C., Laptev, I.: Towards unconstrained joint hand-object reconstruction from rgb videos. In: 3DV (2021) [1](#), [3](#), [20](#)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) [3](#), [11](#)
27. Huang, D., Ji, X., He, X., Sun, J., He, T., Shuai, Q., Ouyang, W., Zhou, X.: Reconstructing hand-held objects from monocular video. In: SIGGRAPH Asia (2022) [3](#)
28. Iqbal, U., Molchanov, P., Gall, T., Kautz, J.: Hand pose estimation via latent 2.5D heatmap regression. In: ECCV (2018) [3](#)
29. Jiang, H., Liu, S., Wang, J., Wang, X.: Hand-object contact consistency reasoning for human grasps generation. In: ICCV (2021) [3](#), [33](#), [35](#)
30. Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., Ouyang, W., Luo, P.: Whole-body human pose estimation in the wild. In: ECCV (2020) [3](#)
31. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end Recovery of Human Shape and Pose. In: CVPR (2018) [10](#)
32. Kocabas, M., Huang, C.H.P., Hilliges, O., Black, M.J.: PARE: Part attention regressor for 3D human body estimation. In: ICCV (2021) [10](#)
33. Kulon, D., Guler, R.A., Kokkinos, I., Bronstein, M.M., Zafeiriou, S.: Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: CVPR (2020) [2](#)
34. Kulon, D., Wang, H., Güler, R.A., Bronstein, M.M., Zafeiriou, S.: Single Image 3D Hand Reconstruction with Mesh Convolutions. In: BMVC (2019) [2](#)
35. Kwon, T., Tekin, B., Stühmer, J., Bogu, F., Pollefeys, M.: H2O: Two Hands Manipulating Objects for First Person Interaction Recognition. In: ICCV (2021) [1](#), [2](#), [3](#)
36. Lakshmipathy, A.S., Feng, N., Lee, Y.X., Mahler, M., Pollard, N.: Contact Edit: Artist Tools for Intuitive Modeling of Hand-Object Interactions. In: ACM Transactions on Graphics (2023) [5](#)
37. Lee, J., Sung, M., Choi, H., Kim, T.: Im2hands: Learning attentive implicit representation of interacting two-hand shapes. In: CVPR (2023) [3](#)
38. Li, M., An, L., Zhang, H., Wu, L., Chen, F., Yu, T., Liu, Y.: Interacting attention graph for single image two-hand reconstruction. In: CVPR (2022) [3](#)
39. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: CVPR (2021) [3](#)

40. Liu, R., Ohkawa, T., Zhang, M., Sato, Y.: Single-to-dual-view adaptation for ego-centric 3d hand pose estimation. In: CVPR (2024) [3](#)
41. Liu, Y., Liu, Y., Jiang, C., Lyu, K., Wan, W., Shen, H., Liang, B., Fu, Z., Wang, H., Yi, L.: HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction. In: CVPR (2022) [1](#), [2](#), [3](#)
42. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) [10](#)
43. Moon, G., Yu, S.I., Wen, H., Shiratori, T., Lee, K.M.: InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image. In: ECCV (2020) [3](#)
44. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: GANerated hands for real-time 3D hand tracking from monocular RGB. In: CVPR (2018) [3](#)
45. Ohkawa, T., Furuta, R., Sato, Y.: Efficient annotation and learning for 3d hand pose estimation: A survey. In: IJCV (2023) [3](#)
46. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision. TMLR (2024) [3](#), [10](#), [11](#), [21](#)
47. Patel, A., Wang, A., Radosavovic, I., Malik, J.: Learning to imitate object interactions from internet videos. In: arXiv preprint arXiv:2211.13225 (2022) [1](#), [3](#)
48. Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., Malik, J.: Reconstructing Hands in 3D with Transformers. In: CVPR (2024) [2](#), [3](#)
49. Potamias, R.A., Zhang, J., Deng, J., Zafeiriou, S.: Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In: CVPR (2025) [2](#), [3](#), [5](#), [6](#), [7](#), [10](#), [21](#), [26](#), [32](#), [35](#)
50. Prakash, A., Chang, M., Jin, M., Tu, R., Gupta, S.: 3d reconstruction of objects in hands without real world 3d supervision. In: ECCV (2024) [2](#), [3](#)
51. Rogez, G., Supancic, J.S., Ramanan, D.: Understanding everyday hands in action from rgb-d images. In: ICCV (2015) [3](#), [35](#)
52. Romero, J., Tzionas, D., Black, M.J.: Embodied Hands: Modeling and Capturing Hands and Bodies Together. In: ACM Transactions on Graphics (2017) [5](#), [8](#), [28](#)
53. Rong, Y., Shiratori, T., Joo, H.: Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In: ICCVW (2021) [2](#)
54. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: CVPR (2017) [3](#)
55. Taheri, O., Ghorbani, N., Black, M.J., Tzionas, D.: GRAB: A dataset of whole-body human grasping of objects. In: ECCV (2020) [2](#)
56. Tekin, B., Bogo, F., Pollefeys, M.: H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In: CVPR (2019) [3](#)
57. Tripathi, S., Chatterjee, A., Passy, J.C., Yi, H., Tzionas, D., Black, M.J.: DECO: Dense estimation of 3D human-scene contact in the wild. In: ICCV (2023) [2](#), [4](#), [5](#), [6](#), [30](#), [32](#)
58. Tse, T., Kim, K., Leonardis, A., Chang, H.: Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In: CVPR (2022) [3](#)
59. Yang, L., Li, K., Zhan, X., Lv, J., Xu, W., Li, J., Lu, C.: Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In: CVPR (2022) [3](#)

60. Yang, L., Li, K., Zhan, X., Wu, F., Xu, A., Liu, L., Lu, C.: Oakink: A large-scale knowledge repository for understanding hand-object interaction. In: CVPR (2022) [1](#), [2](#), [3](#)
61. Yang, L., Zhan, X., Li, K., Xu, W., Li, J., Lu, C.: CPF: Learning a contact potential field to model the hand-object interaction. In: ICCV (2021) [3](#)
62. Yang, Y., Zhai, W., Luo, H., Cao, Y., Zha, Z.J.: LEMON: Learning 3D Human-Object Interaction Relation from 2D Images. In: CVPR (2024) [4](#)
63. Yang, Y., Zhai, W., Wang, C., Yu, C., Cao, Y., Zha, Z.J.: EgoChoir: Capturing 3D Human-Object Interaction Regions from Egocentric Views. In: NeurIPS (2024) [4](#)
64. Ye, Y., Feng, Y., Taheri, O., Feng, H., Tulsiani, S., Black, M.J.: Predicting 4d hand trajectory from monocular videos. arXiv preprint arXiv:2501.08329 (2025) [3](#)
65. Ye, Y., Gupta, A., Kitani, K., Tulsiani, S.: G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In: CVPR (2024) [3](#), [24](#), [35](#)
66. Ye, Y., Gupta, A., Tulsiani, S.: What’s in your hands? 3d reconstruction of generic objects in hands. In: CVPR (2022) [3](#)
67. Ye, Y., Hebbar, P., Gupta, A., Tulsiani, S.: Diffusion-guided reconstruction of everyday hand-object interaction clips. In: ICCV (2023) [3](#)
68. Yu, Z., Xu, W., Xie, P., Li, Y., Anthony, B.W., Zhang, Z., Lu, C.: Dynamic reconstruction of hand-object interaction with distributed force-aware contact representation. In: ICCV (2025) [3](#)
69. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: CVPR (2022) [10](#)
70. Zhang, J., Deng, J., Ma, C., Potamias, R.A.: Hawor: World-space hand motion reconstruction from egocentric videos. In: CVPR (2025) [3](#)
71. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular rgb image. In: ICCV (2019) [10](#)
72. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the Continuity of Rotation Representations in Neural Networks. In: CVPR (2019) [8](#), [10](#)
73. Zhu, Z., Bansal, S., Tripathi, S., Damen, D.: Reconstructing Objects along Hand Interaction Timelines in Egocentric Video. In: CVPRW (2026) [6](#)
74. Zhu, Z., Damen, D.: Get a grip: Reconstructing hand-object stable grasps in egocentric videos. In: arXiv (2024) [3](#), [4](#), [6](#), [7](#), [10](#), [20](#), [26](#)
75. Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., Brox, T.: Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In: ECCV (2019) [3](#)