EgoHand: Monocular 3D Hand Reconstruction from Egocentric Video

NVIDIA[1]

arXiv:2610.30293v3 [cs.CV] 18 Nov 2026

Abstract

We introduce EgoHand, a monocular 3D hand reconstruction model that achieves state-of-the-art performance on egocentric video. Central to our approach is the curation of a high-quality training corpus of 14 million images, combining synthetic data, in-the-wild video, and multi-camera motion capture to balance scene diversity and supervision quality. The model adopts SOMA-Hand as its parametric representation and processes 2D hand crops with a transformer encoder, incorporating camera cues and an auxiliary 2D branch to improve geometric consistency and image alignment. Zero-shot evaluations on HOT3D, ARCTIC, and H2O demonstrate strong out-of-domain generalization, with lower joint-position, root-translation, and acceleration errors than competing baselines at an inference latency of approximately 9.01 ms. A blinded user preference study further shows that participants favor EgoHand's reconstructed hand poses over those from Apple EgoDex and Ropedia Xperience-10M. We release the EgoHand model and source code under the OpenMDW license, allowing free commercial use of the model.

Model: https://huggingface.co/nvidia/EgoHand-v1

Inference: https://github.com/NVIDIA/IsaacCapture/tree/main/src/python/isaaccapture_egopost

Training: https://github.com/NVlabs/GEM-X

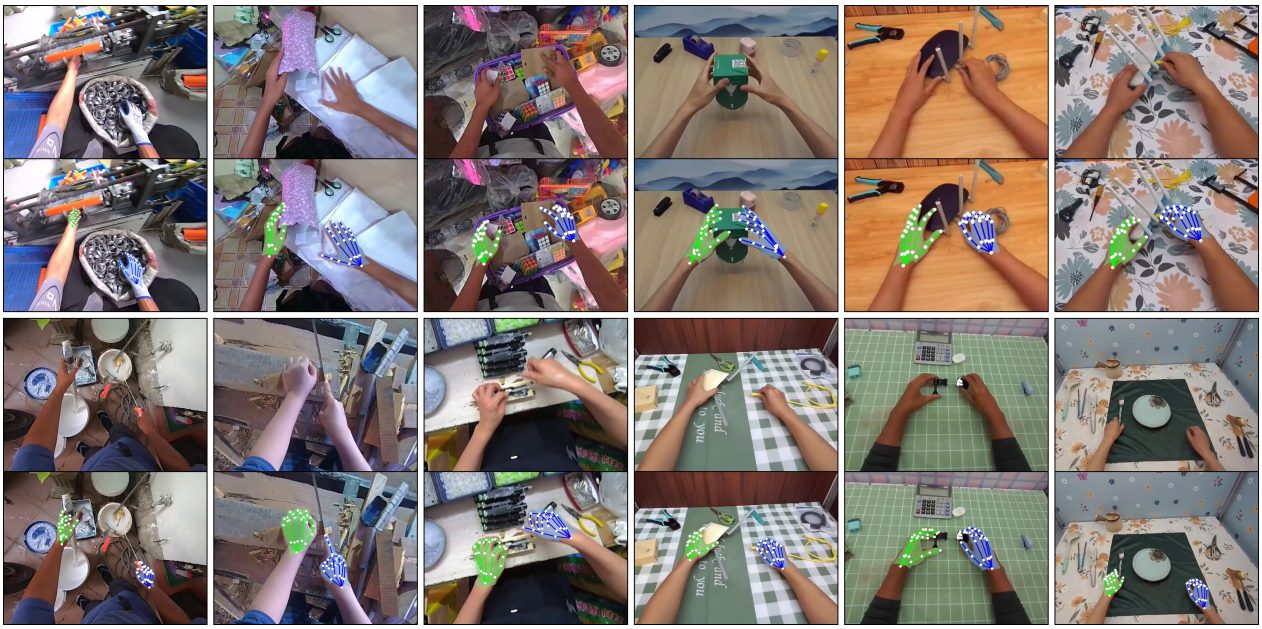


Figure 1: **EgoHand** hand reconstruction results. Our approach achieves robust reconstructions under challenging poses across diverse environments.

[1]A detailed list of contributors and acknowledgments can be found in Section A of this paper.

© 2026 NVIDIA. All rights reserved.

1. Introduction

Egocentric human videos paired with 3D hand annotations have emerged as a scalable source of supervision for pretraining vision-language-action (VLA) and world-action models (WAMs) (Zheng et al., 2026). By capturing human manipulation across diverse objects, tasks, and environments, these recordings offer a path to reducing reliance on costly robot teleoperation. EgoScale (Zheng et al., 2026), for example, demonstrates that scaling action-labeled human video improves dexterous manipulation performance and, together with aligned human–robot training, enables adaptation with few robot demonstrations. In these pipelines, reconstructed hand trajectories provide action supervision that connects visual observations to manipulation behavior. Accurate and efficient 3D hand reconstruction is therefore an important enabling capability for scaling robot learning from human demonstrations.

Despite substantial progress in monocular hand reconstruction (Pavlakos et al., 2024; Potamias et al., 2025; Wang et al., 2026; Yu et al., 2025; Zhang et al., 2025), producing reliable annotations across diverse egocentric recordings remains challenging. Models must generalize across hand appearances, backgrounds, objects, and camera configurations, yet obtaining accurate 3D supervision with comparable diversity is difficult. Combining heterogeneous datasets further introduces differences in joint definitions, annotation reliability, and geometric consistency. At inference time, reconstruction must also preserve image alignment and recover camera-relative hand placement from cropped observations, while remaining efficient enough to process large video collections.

We introduce EgoHand, a monocular 3D hand reconstruction model that addresses these challenges through camera-aware prediction and the systematic curation of diverse, high-quality training data. Our approach combines a geometrically grounded reconstruction architecture with a corpus of 14 million images that brings together complementary forms of visual diversity and 3D supervision.

Model. We adopt SOMA-Hand (Saito et al., 2026) with the MHR shape backend (Ferguson et al., 2025), which provides explicit skeletal scale controls alongside surface-shape parameters. This representation supports fitting heterogeneous landmark annotations while maintaining a consistent hand identity across each sequence. Following prior methods (Pavlakos et al., 2024; Potamias et al., 2025), EgoHand processes 2D hand crops with a transformer encoder. Because cropping and resizing remove explicit information about the hand's original image location and extent, we combine bounding-box conditioning (Li et al., 2022) with ray positional encoding (Prakash et al., 2024) to retain the geometric relationship between the crop and the camera. A separate 2D joint prediction branch supplies image-space correspondences that refine lateral wrist translation while retaining the depth predicted by the 3D branch. Joint training through this refinement couples image alignment with reconstructed geometry and camera-relative placement. Section 3 describes the model.

Data. We construct our training corpus from three complementary sources. First, we use high-quality hand motion captured with gloves to animate synthetic hands with diverse shapes and appearances. This provides abundant training images with accurate geometric labels and controlled variation in hand appearance. Second, we incorporate in-the-wild egocentric videos to capture the breadth of backgrounds, objects, and interactions needed for robust generalization. To make these recordings suitable for training, we develop a comprehensive pseudo-annotation pipeline that combines available 3D landmarks, image-derived hand segmentation masks, and temporal constraints. The pipeline estimates a shared hand shape within each sequence, optimizes per-frame articulation and wrist motion, and refines difficult intervals using alignment and motion checks. Third, we collect hand motion using micro-markers synchronized with multiview video, yielding real images paired with high-quality 3D ground-truth annotations. This source complements the appearance diversity of synthetic data and the scene and object diversity of in-the-wild video with precise supervision on real imagery, helping bridge the synthetic-to-real domain gap. We express all annotations in a common SOMA-Hand representation to support consistent training across sources. Section 4 details the data collection and annotation pipeline.

EgoHand achieves state-of-the-art 3D hand reconstruction with an inference latency of approximately 9.01 ms on an NVIDIA RTX Pro 6000 GPU. It demonstrates strong zero-shot generalization on HOT3D (Banerjee et al.,

2025), H2O (Kwon et al., 2021), and ARCTIC (Fan et al., 2023) without training on any of these datasets. On H2O, it achieves the lowest mean and p90 joint-position, root-translation, and acceleration errors among HaMeR (Pavlakos et al., 2024), WiLoR (Potamias et al., 2025), HaWoR (Zhang et al., 2025), and our best-effort reproduction of Dyn-HaMR (Yu et al., 2025). On HOT3D and ARCTIC, which are part of the training set of WiLoR and HaWoR, EgoHand outperforms HaMeR and our Dyn-HaMR reproduction across all five evaluated metrics in zero-shot. A blinded side-by-side study further shows that human graders prefer EgoHand 's projected hand poses over the supplied annotations on 77.0% of Apple EgoDex clips and 95.5% of Ropedia Xperience-10M clips, while reporting comparable visual quality with Lightwheel EgoStandard. We release the EgoHand model and source code under the OpenMDW license, allowing commercial use of the model.

2. Related Work

3D Hand Representations. MANO is widely used for hand pose and shape reconstruction (Romero et al., 2017). Its learned shape space couples skeletal proportions and surface geometry, limiting direct control of individual bone lengths. Momentum Human Rig (MHR) decouples skeletal parameters from surface shape and includes 23 joints per hand, counting the wrist and fingertip end joints (Ferguson et al., 2025). SOMA accommodates different skeletons and shape backends in a common framework (Saito et al., 2026). For both annotation and prediction, we use native SOMA-Hand with 25 joints per hand and the MHR shape backend with explicit skeletal scale controls (NVIDIA SOMA-X, 2026).

Hand Reconstruction Models. HaMeR (Pavlakos et al., 2024) reconstructs hands using a large vision transformer, while WiLoR (Potamias et al., 2025) combines hand localization and reconstruction. For video, Dyn-HaMR (Yu et al., 2025) recovers world-coordinate hand trajectories through motion priors and multi-stage optimization; HaWoR (Zhang et al., 2025) combines temporal reconstruction, camera tracking, and motion infilling. ViDiHand (Wang et al., 2026) draws on scene and motion priors from a pretrained video diffusion model to reconstruct heavily occluded hands. Its reported throughput of 5.5 frames per second on four A100 GPUs limits its practicality for large-scale annotation. EgoHand combines camera conditioning and a separate 2D joint branch to constrain camera-space hand placement, using a DINOv3 encoder initialized from SAM 3D Body (Siméoni et al., 2025; Yang et al., 2026). Compared with the evaluated baselines, it improves joint-position and root-translation accuracy while retaining efficient single-frame inference.

Hand Reconstruction Datasets. Hand reconstruction across varied poses and imaging conditions benefits from large, diverse training sets (Pavlakos et al., 2024; Potamias et al., 2025). Existing resources span real captures, including ARCTIC and HOT3D (Banerjee et al., 2025; Fan et al., 2023), and synthetic renderings such as RHD and BEDLAM (Black et al., 2023; Zimmermann and Brox, 2017). FreiHAND and WiLoR fit parametric hand models to skeletal keypoints to obtain pose and shape annotations (Potamias et al., 2025; Zimmermann et al., 2019). To scale training with diverse appearances and 3D supervision, we combine synthetic images, in-the-wild video, and multi-camera motion capture, totaling roughly 14 million images. Synthetic data retain SOMA-Hand parameters; real recordings are fitted to 3D landmarks in the same representation (Section 4).

3. EgoHand

3.1. Network Architecture

Given a 2D hand crop, EgoHand extracts image features using a DINOv3 ViT-H/16+ encoder (Siméoni et al., 2025). As shown in Figure 2, the **3D hand reconstruction branch** (Section 3.2) combines these features with geometric cues derived from camera intrinsics and the 2D bounding box to predict hand pose, shape, and wrist translation. A separate **2D joint prediction branch** (Section 3.3) uses the image features to predict 2D joint locations. Translation refinement uses correspondences between predicted 2D and reconstructed 3D joints to adjust translation, positioning the reconstructed hand in camera coordinates. Together, the two branches recover hand geometry and its wrist translation in camera coordinates.

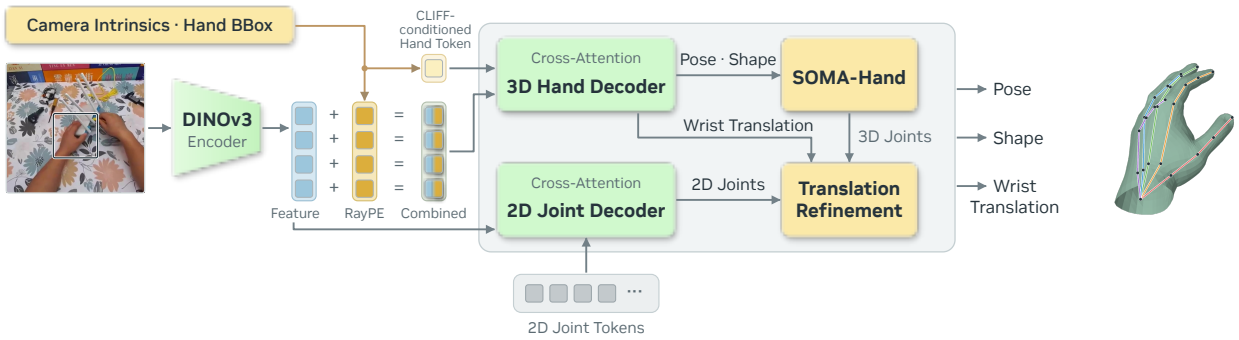


Figure 2: EGOHAND model architecture. A DINOv3 encoder supports two branches. The 3D hand reconstruction branch uses a CLIFF-conditioned hand query to cross-attend to image features augmented with ray positional encoding, predicting SOMA-Hand pose, shape, and wrist translation. The 2D joint prediction branch uses separate learned joint queries to cross-attend to image features. Its predicted 2D joints, together with the SOMA-Hand 3D joints and camera intrinsics, refine lateral wrist translation while preserving the predicted depth, yielding hand geometry in camera coordinates.

3.2. 3D Hand Reconstruction

Resizing a hand crop standardizes the network input but removes explicit information about its position and extent in the original image. We retain this geometric context through bounding-box conditioning (CLIFF) (Li et al., 2022), which encodes the crop's location and scale relative to the camera, and ray positional encoding (RayPE) (Wang et al., 2026), which associates image features with their viewing directions. As illustrated in Figure 2, bounding-box conditions the initial decoder token, while ray features are added to the image features. From the 3D hand decoder output, regression heads predict SOMA-Hand pose and shape parameters. The pose parameters describe wrist orientation and finger articulation, while the shape parameters control hand shape and proportions. SOMA-Hand maps these predictions to a hand mesh and 3D joints expressed relative to the wrist. A separate camera head predicts the translation that positions this geometry in camera coordinates.

For a square crop with center (c_x, c_y) and side length b, let (f_x, f_y) and (p_x, p_y) denote the camera focal lengths and principal point, respectively. The camera head predicts (ℓ, d_x, d_y), where $\ell = \log(bZ/f_x)$ parameterizes wrist depth Z, and d_x, d_y are metric lateral offsets. These parameters yield a direct estimate of wrist translation in camera coordinates:

$$\begin{aligned} Z &= \frac{f_x}{b}\exp(\ell), \\ T_x^{\text{reg}} &= d_x + \frac{c_x - p_x}{f_x} Z, \\ T_y^{\text{reg}} &= d_y + \frac{c_y - p_y}{f_y} Z. \end{aligned} \tag{1}$$

3.3. 2D Joint Prediction and Translation Refinement

The 2D joint prediction branch estimates joint locations from learnable 2D joint tokens by cross-attenting to the image features, providing correspondences between image observations and the hand geometry reconstructed by the 3D branch. These correspondences constrain lateral translation while retaining the wrist depth predicted by the 3D branch.

Let $\widehat{q}_j$ denote a predicted 2D joint mapped from crop-normalized coordinates to image pixels, and J_j the corresponding reconstructed wrist-relative 3D joint. Given camera intrinsics K and predicted wrist depth Z, we estimate lateral translation by minimizing reprojection error over selected correspondences $\mathcal{V}$:

$$(\widehat{T}_x, \widehat{T}_y) = \underset{(T_x,T_y)\in\mathbb{R}^2}{\operatorname{argmin}} \sum_{j\in\mathcal{V}} \|\pi_K(J_j + (T_x, T_y, Z)) - \widehat{q}_j\|_2^2 + \rho(T_x^2 + T_y^2). \tag{2}$$

The projection π_K uses each joint's camera-space depth. The coefficient ρ controls a small regularization term for numerical stability. When the solve is valid, its solution replaces the lateral components of the direct translation estimate T^{reg} from Section 3.2, giving $T = (\widehat{T}_x, \widehat{T}_y, Z)$. Otherwise, we retain T^{reg}.

The 2D decoder also predicts uncertainty scales and a spatial evidence map for each joint. The uncertainty scales characterize horizontal and vertical localization uncertainty, while the normalized evidence map receives spatial localization supervision. Its total evidence determines a presence score that estimates whether the joint lies within the valid crop, regardless of occlusion. These auxiliary outputs support the training objectives in Section 3.4, and presence scores help select correspondences at inference.

3.4. Training Objective

We jointly train the model using a weighted combination of hand reconstruction, direct 2D prediction, and translation losses, $\mathcal{L} = \sum_i \lambda_i \mathcal{L}_i$.

Hand reconstruction. We apply annotation-confidence-weighted L_1 losses to projected 2D joints and wrist-relative 3D joint positions, and squared-error losses to SOMA-Hand pose and shape parameters. The camera head receives smooth L_1 supervision on the standardized logarithmic-depth residual and squared-error supervision on lateral offsets. Together, these losses constrain hand geometry and image alignment while training the direct translation estimate independently of the correspondence solve.

Direct 2D joint prediction. We supervise predicted joint coordinates using residual log-likelihood estimation with learned coordinate uncertainty (Li et al., 2021), including labeled joints outside the crop. Binary cross-entropy trains presence scores to indicate whether annotated joints lie within the valid crop. For joints inside this region, a spatial loss supervises the normalized evidence maps using the continuous ranked probability score on their horizontal and vertical marginals.

Translation supervision. We apply squared-error supervision to the translation obtained from solving valid 2D–3D correspondence, using ground-truth camera-space translation as the target. Backpropagation through the solve couples the predicted 2D coordinates, reconstructed 3D geometry, and depth. The solved translation also enters the 2D reprojection loss. When the solve is invalid, reprojection uses the direct estimate T^{reg}, and the sample is excluded from the translation loss.

4. Data Sourcing and Annotation

Robust 3D hand reconstruction relies on large-scale datasets with accurate, consistent annotations (Pavlakos et al., 2024; Potamias et al., 2025). Yet commercially usable datasets with accurate 3D hand annotations and diverse scenes, poses, and hand shapes remain scarce.

EGOHAND is trained on 14 million images drawn from three complementary sources — synthetic data, in-the-wild video, and multi-view motion capture (MoCap) — to balance scene and camera diversity with annotation accuracy. Our *annotation-fitting pipeline* brings these sources into a common SOMA-Hand representation by leveraging its decoupled shape and pose parameterization: a single hand shape is optimized per clip, while per-frame poses are fit jointly to 2D hand segmentation masks and available 3D keypoint annotations.

4.1. Data Sources

Synthetic data. Synthetic images provide accurate hand geometry and controlled variation in camera configurations. We use NVIDIA Omniverse (NVIDIA Omniverse) to generate approximately 49 hours of synthetic hand sequences, varying hand size, motion, scene content, camera viewpoint, and intrinsic parameters.

In-the-wild video. In-the-wild footage contributes diverse appearances, scenes, object interactions, and hand sizes. We source around 63 hours licensed video, covering everyday manipulation and industrial tasks, including pick-and-place, sorting, and the assembly and disassembly of small rigid parts. The recordings use head-mounted cameras with horizontal FOV ranging from $101°$ to $165°$, encompassing both wide-angle pinhole and fisheye camera models.

MoCap data. Multi-camera motion capture systems, such as Vicon (Vicon), track reflective markers to provide precise hand motion annotations, *including when hands are occluded in the egocentric view*. We collected approximately 16 hours of hand-object manipulation sequences from Noitom Robotics (Noitom Robotics) and Bones (Bones), covering seated and standing activities with varying levels of hand visibility. Motion capture is performed using micro-markers to maximize visibility of hand without using full-hand gloves.

4.2. Data Annotation Pipeline

Our pipeline converts varied real-world annotations into a common SOMA-Hand representation, accounting for differences in anatomical joint definitions. We fit SOMA-Hand using calibrated video, supplied 3D hand landmarks, and SAM 3 segmentation masks. Anatomically corresponding landmarks guide the fit, while 2D mask alignment and 2D–3D consistency checks help assess agreement with the observations.

SOMA-Hand's decoupled shape and pose parameters allow us to estimate a single hand shape, including its proportions and overall size, shared across each clip. With shape fixed, we optimize per-frame poses, comprising finger articulation, wrist orientation, and wrist position, with temporal constraints. This maintains consistent hand geometry while capturing motion throughout the sequence. We first construct image-space fitting targets and estimate shared hand shape from selected frames, then fit poses across the full clip and refine difficult intervals before export.

Constructing image-space fitting targets. We use camera calibration to transform the supplied 3D landmarks into camera coordinates and project them onto the corresponding video frames, producing 2D keypoint targets. Fisheye images are rectified when needed, and landmark projection uses the intrinsics of the rectified view. SAM 3 provides segmentation masks of the visible hands. Forearm regions and ambiguous overlap between hands are excluded from mask comparison so that they do not distort the fitted hand silhouette.

Estimating a consistent hand shape. We estimate a single hand shape from selected frames with clear visibility and diverse poses. The supplied 3D landmarks initialize hand size and wrist placement, while measurements of palm width and finger lengths constrain hand proportions during refinement. Each selected frame has its own pose, but all frames share the same shape parameters. We alternate between adjusting these poses with shape fixed and adjusting the shared shape with poses fixed. Landmark alignment and agreement between the rendered hand silhouette and SAM 3 masks guide these updates, while the dimension constraints discourage unrealistic changes in hand proportions. Combining evidence across frames reduces dependence on any single pose or partially occluded observation. The resulting shape, including overall size, is then held fixed during pose fitting across the full clip, preventing frame-level noise from causing artificial changes in hand proportions.

Fitting hand motion. With shape fixed, we optimize each frame's pose using the supplied 3D landmarks, selected 2D keypoint correspondences, and SAM 3 masks. The keypoints guide finger placement, while alignment between the rendered hand silhouette and the masks constrains the visible hand outline. We use anatomically corresponding keypoints and exclude ambiguous matches, such as differing wrist definitions. Temporal constraints connect neighboring frames to reduce jitter while preserving alignment with the observations.

Refinement and quality checks. Intervals with poor alignment or unstable motion receive additional pose refinement, with shape remaining fixed. Updates are accepted only when alignment and motion checks pass. Before export, we validate frame coverage and model parameters. The final annotations contain one shared hand shape and size per clip, together with per-frame finger articulation and wrist placement. Their accuracy remains dependent on the source observations.

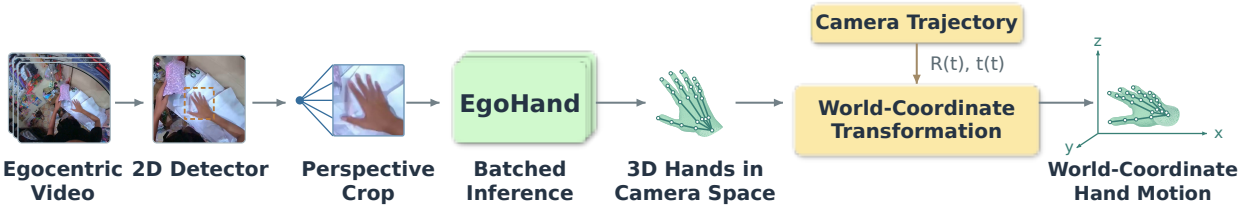


Figure 3: Inference pipeline from egocentric video to world-coordinate hand motion, comprising hand detection, crop preparation, batched EgoHand inference, and transformation of predictions to world coordinates. Fisheye regions are resampled into virtual pinhole views; pinhole images use standard crops. Predictions are assembled in frame order, their hand parameters are smoothed in an adaptive window. A synchronized external camera trajectory supplies the camera-to-world rotations and translations that place the hands in a shared world frame. Benchmark annotations may supply hand boxes in place of detections. The EgoHand network is detailed in Section 3. Illustrations are schematic.

5. End-to-End Inference Pipeline

We provide an example modular inference pipeline that uses the EgoHand model from Section 3 to annotate egocentric video. Given a video and camera calibration, the pipeline detects hands, prepares image crops, reconstructs hand geometry, and assembles the predictions into adaptively smoothed sequences. A synchronized external camera trajectory enables conversion from camera-relative reconstructions to world-coordinate hand annotations for downstream policy learning.

Figure 3 summarizes the workflow. Hand detection supplies bounding boxes for crop preparation. Pinhole images use standard crops, while fisheye images are resampled into virtual pinhole views with corresponding intrinsics. The pipeline batches these crops for EgoHand inference and transforms predictions from virtual views back to the source-camera frame. Each prediction retains its source-frame identifier and handedness so that batch outputs can be assembled into left- and right-hand sequences. Adaptive postprocessing operates on the hand parameters before geometry is decoded for world-coordinate conversion and export.

The stages communicate through hand boxes, camera parameters, and reconstructed hand sequences. This organization allows the detector, camera-pose source, postprocessing, and execution infrastructure to be adapted while retaining the same reconstruction model.

2D Hand Detection. We provide an example 2D hand detector by fine-tuning RT-DETRv2-S (Lv et al., 2024) on datasets listed in Section 4, with bounding-box annotations for both hands derived from the 3D annotations.

Fisheye Support. To handle fisheye distortion, we apply *perspective cropping* (Han et al., 2022) using a virtual pinhole camera that shares the source camera's optical center and points toward the hand bounding-box center. The resulting undistorted crop and virtual-camera intrinsics are then passed to EgoHand for reconstruction.

Adaptive Smoothing. Independent frame predictions can jitter, so we apply offline smoothing to each hand's SOMA-Hand parameters using video timestamps. Rotation and translation filtering adapts to local motion. We fit local quadratic motion curves, adjusting the smoothing window according to fitting error and estimated noise. Windows narrow when a broader fit cannot capture local motion, helping preserve motion changes while reducing jitter. Shape and scale are also smoothed. We regenerate joints and meshes from the filtered parameters.

World-coordinate Reconstruction. We use camera-to-world poses from ViPE (Huang et al., 2025) or the recording to place hands in a shared world frame. The trajectory must be aligned with the video frames and use the same camera convention and metric scale as the hand predictions. For source-camera joint $X^c_{t,j}$ and camera-to-world pose $(R_t^{c\to w}, t_t^{c\to w})$,

$$X^w_{t,j} = R_t^{c\to w} X^c_{t,j} + t_t^{c\to w}. \tag{3}$$

An optional refinement step uses both hands jointly to adjust camera poses and reduce shared world-space jitter, with additional smoothing of the resulting keypoint trajectories. Without a camera trajectory, outputs remain camera-relative.

5.1. Hand Reconstruction at Production-scale

We establish a hand reconstruction production pipeline that organizes video decoding, camera-pose estimation, hand detection, crop preparation, batched reconstruction, postprocessing, and LeRobot dataset assembly for annotation at scale. It targets collections exceeding 10,000 hours of egocentric video across tens of thousands of clips. At this scale, predictable resource use and efficient scheduling are essential to processing long or unusually large inputs reliably.

Bounded memory footprint. Decoding and processing operate on fixed-size chunks, so decoded-frame memory depends on chunk size rather than video length. Camera-pose estimation retains a fixed window of SLAM keyframes in GPU memory and retires older keyframes into a compact CPU-side trajectory record. Temporal hand postprocessing uses local windows rather than retaining full-track history on the GPU. Per-job resource caps and prompt cleanup bound resource use and allow multiple workers to share a machine.

Batch processing and throughput. Hand crops are pooled into fixed-size inference batches, and downstream postprocessing and GPU-to-CPU transfers are also batched to reduce per-item overhead. Forward passes use CUDA autocast with bfloat16, while numerically sensitive accumulation remains in float32. Each job processes multiple videos, amortizing startup, CUDA initialization, and model loading. In early production runs, batching videos per job reduced this fixed overhead by roughly an order of magnitude. Input downloads and output uploads run concurrently with computation, and results are streamed as each processing unit finishes.

Deployment and output assembly. Containers pin the CUDA toolchain and supported GPU architectures. Model checkpoints are included at build time, with a public default as fallback, to avoid repeated runtime downloads. Builds are identified by immutable SHA-256 digests, allowing outputs to be traced to the software version that produced them. NVIDIA OSMO workflows distribute inference to GPU pools and CPU-bound stages to CPU-rich pools. The final assembly stage associates hand annotations with their source frames and prepares LeRobot datasets for downstream policy-learning workflows.

6. Evaluation

6.1. Experimental Setup

Datasets. We evaluate on the HOT3D (Banerjee et al., 2025), H2O (Kwon et al., 2021), and ARCTIC (Fan et al., 2023) datasets. None of these datasets is included in our training data. Our evaluation covers approximately 2.11 hours of egocentric video from HOT3D, 1.06 hours from H2O, and 0.23 hours from ARCTIC, all at 30 fps.

Baselines. On H2O, we compare with single-image methods HaMeR (Pavlakos et al., 2024) and WiLoR (Potamias et al., 2025) and video-based methods HaWoR (Zhang et al., 2025) and Dyn-HaMR (Yu et al., 2025). None of these methods was trained on H2O. On HOT3D and ARCTIC, we compare with HaMeR and Dyn-HaMR, excluding WiLoR and HaWoR because their training data include HOT3D and ARCTIC. In our evaluation, we provide HaWoR and Dyn-HaMR with ground-truth camera poses to exclude camera-pose estimation error from the comparison. Our evaluation compares hand reconstruction from ground-truth hand boxes and does not assess completion of missing hand trajectories. We therefore omit HaWoR's separate motion-infilling module. We exclude Dyn-HaMR's hand motion prior (HMP) refinement because we could not run this stage successfully with the official code.

Evaluation protocol. All methods receive ground-truth hand boxes to evaluate reconstruction independently of detection. EgoHand uses FP16 inference with 256×256 inputs. On H2O, all methods use hand crops from the RGB images. On HOT3D, EgoHand uses perspective crops from fisheye images. On ARCTIC, all

methods use hand crops from globally undistorted RGB images, with boxes obtained by projecting ground-truth MANO mesh vertices. We apply adaptive smoothing to EGOHAND predictions unless a result is labeled "without smoothing." To compare against MANO annotations, we map EGOHAND's predicted SOMA-Hand surfaces to MANO topology, fit a MANO mesh to the mapped surface, and extract 21 evaluation landmarks.

Metrics. We report the mean and p90 of wrist-relative MPJPE, PA-MPJPE, and root translation error (RTE) in mm, 2D EPE in pixels, and acceleration error in mm/frame^2. See Appendix B for detailed definition.

6.2. Evaluation Results

Tables 1–3 compare EGOHAND at 60k training steps with baseline methods, reporting mean and p90 errors on H2O, HOT3D, and ARCTIC across MPJPE, PA-MPJPE, RTE, 2D EPE, and acceleration error.

H2O. EGOHAND achieves the lowest mean and p90 MPJPE, RTE, and acceleration error. WiLoR retains the lowest mean and p90 PA-MPJPE and 2D EPE. Its two-stage reconstruction refines an initial hand estimate using image-aligned features (Potamias et al., 2025), at a higher inference cost (Section 6.4, Table 5).

Table 1: Reconstruction accuracy on H2O. Lower is better for all metrics.

Method	MPJPE ↓ (mm)		PA-MPJPE ↓ (mm)		RTE ↓ (mm)		2D EPE ↓ (px)		Accel. error ↓ (mm/frame^2)	
	Mean	p90	Mean	p90	Mean	p90	Mean	p90	Mean	p90
HaMeR (Pavlakos et al., 2024)	19.35	28.04	7.13	9.61	37.25	71.41	11.15	15.83	4.53	7.58
HaWoR (Zhang et al., 2025)	17.78	24.05	7.18	9.75	35.63	55.71	17.08	22.08	2.82	5.52
WiLoR (Potamias et al., 2025)	17.77	25.32	**6.06**	**8.16**	30.70	56.78	**8.96**	**12.37**	3.87	6.50
Dyn-HaMR (Yu et al., 2025)	19.32	27.97	7.14	9.61	37.18	71.55	15.50	22.90	4.36	7.28
EGOHAND (Ours) w/o smoothing	13.24	18.91	6.91	8.97	25.70	42.26	9.61	12.68	3.92	6.26
EGOHAND (Ours) w/ smoothing	**13.12**	**18.75**	6.87	8.92	**25.63**	**42.10**	9.59	12.66	**1.90**	**3.92**

HOT3D. EGOHAND achieves lower mean and p90 errors than HaMeR and Dyn-HaMR across all five metrics.

Table 2: Reconstruction accuracy on HOT3D. Lower is better for all metrics.

Method	MPJPE ↓ (mm)		PA-MPJPE ↓ (mm)		RTE ↓ (mm)		2D EPE ↓ (px)		Accel. error ↓ (mm/frame^2)	
	Mean	p90	Mean	p90	Mean	p90	Mean	p90	Mean	p90
HaMeR (Pavlakos et al., 2024)	58.00	79.47	9.47	13.02	51.92	89.60	22.88	36.41	5.92	10.04
Dyn-HaMR (Yu et al., 2025)	58.16	79.78	9.64	13.25	53.51	94.91	22.95	36.50	5.83	9.88
EGOHAND (Ours) w/o smoothing	18.93	26.00	9.10	11.41	36.20	57.87	16.43	**22.06**	4.39	7.29
EGOHAND (Ours) w/ smoothing	**18.82**	**25.78**	**9.04**	**11.32**	**36.13**	**57.63**	**16.42**	22.07	**1.63**	**3.10**

ARCTIC. EGOHAND achieves lower mean and p90 errors than HaMeR and Dyn-HaMR across all five metrics, with and without adaptive smoothing. Smoothing reduces mean acceleration error from 7.64 to 2.64 mm/frame^2 while also improving the spatial metrics. With smoothing, EGOHAND reduces mean MPJPE, RTE, and acceleration error relative to HaMeR by 34.9%, 51.4%, and 67.5%, respectively. Although EGOHAND processes 256 × 256 hand crops, ARCTIC's 2D EPE is measured in the full resolution 2800 × 2000 image coordinates after undistortion. A given relative localization error therefore spans more pixels than in the lower-resolution H2O and HOT3D images, contributing to ARCTIC's larger 2D EPE.

6.3. Ablation Study

We examine the benefits of ray positional encoding (RayPE), bounding-box conditioning (CLIFF), and the 2D joint prediction branch. Table 4 reports results at 40k training steps without adaptive smoothing.

Ray positional encoding. With CLIFF and the 2D branch disabled, adding RayPE reduces MPJPE from 18.45 to 15.74 mm on H2O and from 26.15 to 18.17 mm on HOT3D, while also lowering RTE and acceleration error

Table 3: Reconstruction accuracy on ARCTIC. Lower is better for all metrics.

Method	MPJPE ↓ (mm) Mean	MPJPE ↓ (mm) p90	PA-MPJPE ↓ (mm) Mean	PA-MPJPE ↓ (mm) p90	RTE ↓ (mm) Mean	RTE ↓ (mm) p90	2D EPE ↓ (px) Mean	2D EPE ↓ (px) p90	Accel. error ↓ (mm/frame^2) Mean	Accel. error ↓ (mm/frame^2) p90
HaMeR (Pavlakos et al., 2024)	26.44	38.05	10.33	15.18	48.35	87.70	69.59	100.68	8.12	13.30
Dyn-HaMR (Yu et al., 2025)	26.60	38.41	10.46	15.40	50.40	91.61	69.34	100.94	7.72	12.93
EgoHand (Ours) w/o smoothing	17.53	24.78	9.04	12.27	23.88	42.26	45.04	66.82	7.64	12.88
EgoHand (Ours) w/ smoothing	**17.21**	**24.42**	**8.95**	**12.13**	**23.51**	**41.49**	**44.84**	**66.11**	**2.64**	**4.74**

Table 4: Component ablations on H2O and HOT3D without adaptive smoothing. All entries are mean errors.

RayPE	CLIFF	2D	H2O MPJPE (mm) ↓	H2O PA-MPJPE (mm) ↓	H2O RTE (mm) ↓	H2O 2D EPE (px) ↓	H2O Accel. error $\frac{mm}{frame^2}$ ↓	HOT3D MPJPE (mm) ↓	HOT3D PA-MPJPE (mm) ↓	HOT3D RTE (mm) ↓	HOT3D 2D EPE (px) ↓	HOT3D Accel. error $\frac{mm}{frame^2}$ ↓
✗	✗	✗	18.45	8.20	27.03	13.40	4.56	26.15	9.10	38.30	26.63	6.04
✓	✗	✗	15.74	8.31	**25.09**	12.90	3.91	**18.17**	8.99	**35.19**	17.33	4.91
✗	✓	✗	16.06	8.38	26.16	12.55	3.90	22.09	9.29	36.03	21.51	5.30
✗	✗	✓	18.89	7.56	25.12	10.53	5.67	23.45	9.17	41.51	18.78	6.67
✓	✓	✗	15.85	8.24	25.83	12.67	**3.54**	18.41	9.00	35.63	16.91	**4.43**
✓	✗	✓	14.55	7.23	25.77	9.65	4.28	19.05	**8.94**	39.14	16.61	4.68
✗	✓	✓	14.30	7.07	25.62	9.67	4.30	19.18	9.19	38.72	17.09	5.14
✓	✓	✓	**13.31**	**6.97**	26.00	**9.51**	4.06	18.90	9.02	38.44	**16.47**	4.69

on both datasets. These comparisons demonstrate its contribution to pose accuracy across different component configurations.

Bounding-box conditioning. With RayPE and the 2D branch disabled, adding CLIFF reduces MPJPE from 18.45 to 16.06 mm on H2O and from 26.15 to 22.09 mm on HOT3D, with improvements in RTE and acceleration error. With RayPE and the 2D branch enabled, adding CLIFF further reduces MPJPE from 14.55 to 13.31 mm on H2O and from 19.05 to 18.90 mm on HOT3D. These results show that bounding-box conditioning complements ray encoding and 2D joint prediction to improve hand-pose accuracy.

2D joint prediction. With RayPE and CLIFF enabled, adding the 2D branch reduces MPJPE from 15.85 to 13.31 mm, PA-MPJPE from 8.24 to 6.97 mm, and 2D EPE from 12.67 to 9.51 px on H2O. On HOT3D, 2D EPE also decreases from 16.91 to 16.47 px. These gains come with higher RTE and acceleration error on both datasets, and a small increase in HOT3D MPJPE from 18.41 to 18.90 mm. Since correspondence-based refinement adjusts lateral translation while retaining the predicted depth, better image alignment does not necessarily imply lower RTE. These results highlight complementary benefits in camera-aware pose estimation and image alignment.

6.4. Inference Latency

Table 5 reports per-frame latency on an NVIDIA L40 GPU, with an additional EgoHand measurement on an NVIDIA RTX PRO 6000 Blackwell GPU. For each method and GPU configuration, we load the model once and run the pipeline repeatedly in the same Python process. We exclude initial warm-up runs and report mean latency per source frame. For EgoHand, inference includes batch preparation (batch size 32), the forward pass, 2D keypoint projection, and mirroring right-hand predictions back from the model's left-hand convention. It excludes 2D hand detection, cropping, and rendering. Post-processing comprises adaptive smoothing for EgoHand. Dyn-HaMR (Yu et al., 2025) initializes hand predictions with HaMeR; its post-processing consists of offline temporal optimization.

Table 5: Per-frame latency on the indicated NVIDIA GPUs.

Method	GPU	Inference ↓ (ms/frame)	Post-processing ↓ (ms/frame)	Total ↓ (ms/frame)
HaMeR (Pavlakos et al., 2024)	L40	26.00	—	26.00
WiLoR (Potamias et al., 2025)	L40	27.43	—	27.43
HaWoR (Zhang et al., 2025)	L40	26.64	—	26.64
Dyn-HaMR (Yu et al., 2025)	L40	26.00	144.03	170.03
EgoHand (ours)	L40	**18.35**	**2.90**	**21.25**
EgoHand (ours)	RTX PRO 6000	**9.01**	**1.86**	**10.87**

6.5. Human Preference Study

We conducted a blinded A/B study comparing the visual accuracy of 2D overlays of EgoHand predictions and dataset-provided hand poses. We randomly selected 200 manipulation clips from each of the three public egocentric datasets: Apple EgoDex (Hoque et al., 2026), Ropedia Xperience-10M (Ropedia, 2026), and Lightwheel EgoStandard, part of EgoSuite-Open100K (Lightwheel, 2026). The 2D hand bounding boxes provided to EgoHand were derived by projecting the datasets' 3D hand poses and applying a 30% padding.

We overlaid the 2D projections of the reconstructed 3D joints on the videos, presented side-by-side with method identities concealed. We used three independently randomized assignment sets. Each of the 600 clip pairs was independently rated by three graders with hand-annotation experience, yielding 1,800 primary ratings. Graders scored preference on a five-point scale and rated each side from 1 (unusable) to 5 (excellent). A majority required at least two graders to agree on direction, grouping slight and clear preferences; a three-way split was considered inconclusive.

Table 6: Majority preference (%, ↑).

	EgoDex	Xperience-10M	EgoStandard
Dataset poses	2.0	1.0	2.5
Tie / Inconclusive	14.0 / 7.0	2.0 / 1.5	**91.0** / 2.5
EgoHand (Ours)	**77.0**	**95.5**	4.0

Table 7: Mean user rated quality (1–5, ↑).

	EgoDex	Xperience-10M	EgoStandard
Dataset poses	3.54	2.50	4.66
EgoHand (Ours)	**4.26**	**3.54**	**4.70**

EgoHand achieved clear gains over the supplied annotations, earning majority preference on 77.0% of EgoDex clips and 95.5% of Xperience-10M clips (Table 6). On EgoStandard, which provides high-quality hand annotations, EgoHand achieved comparable visual quality, with a majority of graders reporting a tie or an inconclusive preference on 93.5% of clips. Mean quality ratings followed the same pattern (Table 7). These results demonstrate EgoHand's effectiveness for producing high-quality hand annotations across diverse egocentric recordings.

7. Conclusion

We presented EgoHand, a monocular 3D hand reconstruction system for scalable annotation of egocentric video. Trained on 14 million images spanning synthetic scenes, in-the-wild recordings, and motion capture unified through SOMA-Hand, the model combines camera geometry cues with auxiliary 2D joint prediction and translation refinement. Zero-shot evaluations on HOT3D, H2O, and ARCTIC demonstrate strong reconstruction accuracy, with adaptive smoothing further reducing acceleration error. Component ablations characterize the benefits and tradeoffs of camera conditioning and 2D prediction. A blinded human study shows a preference for EgoHand's projected poses over supplied EgoDex and Xperience-10M annotations, and predominantly no meaningful difference on EgoStandard. Together with the modular video-processing pipeline, these results support large-scale egocentric hand annotation. Nevertheless, several fundamental and practical challenges remain:

Monocular Ambiguity & Occlusion. Monocular inputs inherently suffer from depth-scale ambiguity. In single-frame egocentric settings, severe self- and object occlusions require predicting occluded joints primarily from learned priors. To mitigate depth ambiguity and enhance robustness under occlusion, we are investigating stereo setups and wrist-mounted cameras to introduce additional geometric constraints and viewpoint diversity.

Upstream Sensitivity. Detection errors in upstream bounding boxes propagate through crop-based geometric features, and inaccurate camera intrinsics impair 3D reconstruction. To reduce upstream dependencies, we are exploring end-to-end detection-reconstruction frameworks and leveraging stereo cues to refine 2D hand localization.

Future work will also explore the impact of annotation quality on downstream robot learning.

References

[1] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3d from egocentric multi-view videos. In *CVPR*, 2025. URL https://arxiv.org/abs/2411.19167. 2, 3, 8

[2] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *CVPR*, 2023. URL https://arxiv.org/abs/2306.16940. 3

[3] Bones. Bones Studio. URL https://bones.studio/. Accessed October 15, 2026. 6

[4] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation. In *CVPR*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Fan_ARCTIC_A_Dataset_for_Dexterous_Bimanual_Hand-Object_Manipulation_CVPR_2023_paper.html. 3, 8

[5] Aaron Ferguson, Ahmed A. A. Osman, Berta Bescos, Carsten Stoll, Chris Twigg, Christoph Lassner, David Otte, Eric Vignola, Fabian Prada, Federica Bogo, Igor Santesteban, Javier Romero, Jenna Zarate, Jeongseok Lee, Jinhyung Park, Jinlong Yang, John Doublestein, Kishore Venkateshan, Kris Kitani, Ladislav Kavan, Marco Dal Farra, Matthew Hu, Matthew Cioffi, Michael Fabris, Michael Ranieri, Mohammad Modarres, Petr Kadlecek, Rawal Khirodkar, Rinat Abdrashitov, Romain Prévost, Roman Rajbhandari, Ronald Mallet, Russell Pearsall, Sandy Kao, Sanjeev Kumar, Scott Parrish, Shoou-I Yu, Shunsuke Saito, Takaaki Shiratori, Te-Li Wang, Tony Tung, Yichen Xu, Yuan Dong, Yuhua Chen, Yuanlu Xu, Yuting Ye, and Zhongshi Jiang. MHR: Momentum human rig, 2025. URL https://arxiv.org/abs/2511.15586. 2, 3

[6] Shangchen Han, Po chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, Randi Cabezas, Luan Tran, Muzaffer Akbay, Tsz-Ho Yu, Cem Keskin, and Robert Wang. UmeTrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. doi: 10.1145/3550469.3555378. URL https://arxiv.org/abs/2211.00099. 7

[7] Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video. In *ICLR*, 2026. URL https://arxiv.org/abs/2505.11709. 11

[8] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. ViPE: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. URL https://arxiv.org/abs/2508.10934. 7

[9] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *ICCV*, 2021. URL https://arxiv.org/abs/2104.11181. 3, 8

[10] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *ICCV*, pages 11025–11034, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Li_Human_Pose_Regression_With_Residual_Log-Likelihood_Estimation_ICCV_2021_paper.html. 5

[11] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. CLIFF: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. URL https://arxiv.org/abs/2208.00571. 2, 4

[12] Lightwheel. EgoSuite-Open100K: A Fully Annotated Open Egocentric Human Dataset. https://huggingface.co/collections/LightwheelAI/egosuite-open100k, August 2026. Dataset. 11

[13] Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. RT-DETRv2: Improved baseline with bag-of-freebies for real-time detection transformer, 2024. URL https://arxiv.org/abs/2407.17140. 7

[14] Noitom Robotics. Noitom Robotics. URL https://noitomrobotics.com/. Accessed October 15, 2026. 6

[15] NVIDIA Omniverse. NVIDIA Omniverse. URL https://www.nvidia.com/en-us/omniverse/. Accessed October 15, 2026. 5

[16] NVIDIA SOMA-X. SOMA-X: Body and hand representation software. Official software repository, 2026. URL https://github.com/NVlabs/SOMA-X. Accessed September 23, 2026; includes SOMAHandLayer. 3

[17] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *CVPR*, 2024. URL https://arxiv.org/abs/2312.05251. 2, 3, 5, 8, 9, 10, 11

[18] Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3d hand localization and reconstruction in-the-wild. In *CVPR*, 2025. URL https://arxiv.org/abs/2409.12259. 2, 3, 5, 8, 9, 11

[19] Aditya Prakash, Ruisen Tu, Matthew Chang, and Saurabh Gupta. 3d hand pose estimation in everyday egocentric images. In *ECCV*, 2024. URL https://arxiv.org/abs/2312.06583. 2

[20] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017. doi: 10.1145/3130800.3130883. 3

[21] Ropedia. Xperience-10M: A Large-Scale Egocentric Multimodal Dataset with Structured 3D/4D Annotations. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/ropedia-ai/xperience-10m. Dataset card accessed September 30, 2026. 11

[22] Jun Saito, Jiefeng Li, Michael de Ruyter, Miguel Guerrero, Edy Lim, Ehsan Hassani, Roger Blanco Ribera, Hyejin Moon, Magdalena Dadela, Marco Di Lucca, Qiao Wang, Xueting Li, Jan Kautz, Simon Yuen, and Umar Iqbal. SOMA: Unifying parametric human body models, 2026. URL https://arxiv.org/abs/2603.16858. 2, 3

[23] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104. 3

[24] Vicon. Vicon Motion Capture Systems. URL https://www.vicon.com/. Accessed October 15, 2026. 6

[25] Yuxi Wang, Chengkai Jin, Yufei Liu, Wenqi Ouyang, Tianyi Wei, Zhiwei Zeng, Siyuan Huang, Zhiqi Shen, and Xingang Pan. The surprising effectiveness of video diffusion models for hand motion reconstruction, 2026. URL https://arxiv.org/abs/2606.30308. 2, 3, 4

[26] Xitong Yang, Devansh Kukreja, Don Pinkus, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jia-Wei Liu, Nicolás Ugrinovic, Anushka Sagar, Jitendra Malik, Matt Feiszli, Piotr Dollár, and Kris Kitani. SAM 3D Body: Robust full-body human mesh recovery. In *CVPR*, pages 7209–7219, 2026. URL https://arxiv.org/abs/2602.15989. 3

[27] Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. Dyn-HaMR: Recovering 4d interacting hand motion from a dynamic camera. In *CVPR*, pages 27716–27726, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yu_Dyn-HaMR_Recovering_4D_Interacting_Hand_Motion_from_a_Dynamic_Camera_CVPR_2025_paper.html. 2, 3, 8, 9, 10, 11

[28] Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *CVPR*, pages 1805–1815, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhang_HaWoR_World-Space_Hand_Motion_Reconstruction_from_Egocentric_Videos_CVPR_2025_paper.html. 2, 3, 8, 9, 11

[29] Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026. URL https://arxiv.org/abs/2602.16710. 2

[30] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. URL https://arxiv.org/abs/1705.01389. 3

[31] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape From Single RGB Images. In *ICCV*, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Zimmermann_FreiHAND_A_Dataset_for_Markerless_Capture_of_Hand_Pose_and_ICCV_2019_paper.html. 3

A. Contributors and Acknowledgments

Core Contributors
Lin Duan, *NVIDIA*
Jiefeng Li, *NVIDIA*
Xiufeng Xie, *NVIDIA*
Yu-Hsiang Huang, *NVIDIA*
Yanchen Liu, *NVIDIA*
Shao Su, *NVIDIA*
Hang Qiu, *NVIDIA*
Yihan Pang, *NVIDIA, UIUC*
Tianyi Hu, *NVIDIA, Duke University*

Contributors
Edy Lim, *NVIDIA*
Hui Kang, NVIDIA
Juana Du, NVIDIA
Jun Saito, *NVIDIA*
Megha Maheshwari, *NVIDIA*

Leadership
David Chu, *NVIDIA*
Karen Zhu, *NVIDIA*
Spencer Huang, *NVIDIA*
Trisha Tripathi, *NVIDIA*

Core Leadership
Chao-Yeh Chen, *NVIDIA*
Tiffany Chen, *NVIDIA*
Umar Iqbal, *NVIDIA*
Qingsi Wang, *NVIDIA*

The roles are defined as follows:

Core Contributor: Individuals who had a very significant and sustained impact throughout the project, across multiple subsystems, with accountability for key design and implementation decisions. Core contributors are listed in order of merit.

Contributor: Individuals who made important contributions to the project. Contributors are listed in alphabetical order.

Leadership: Individuals who contributed to the organizational and technical direction of efforts related to EGOHAND. Leadership members are listed in alphabetical order.

Core Leadership: Individuals with primary responsibility for the organizational and technical direction of EGOHAND. Core leadership members are listed in alphabetical order.

Acknowledgments

We are especially grateful to Ben Levin, Ryan Punamiya, Aly Khater, Danfei Xu, Danyi Chen, Anna Vargas De Inoue, Yuke Zhu, Cyrus Hogg, Sandeep Desai, Ariana Lerena Riofrio, Naema Bhatti, Yan Chang, Xinghao Zhu, and Soha Pouya for their constructive feedback and valuable guidance.

We also acknowledge the use of large language models, including OpenAI GPT-6, to improve the manuscript's language and clarity. All technical content and analyses were developed by the authors. Recognizing that these models may introduce biases, errors, or omissions, the authors take full responsibility for the manuscript's accuracy and appropriateness.

B. Metric Definitions

On HOT3D, H2O, and ARCTIC, we evaluate the reconstructed hand's joint configuration, wrist location, and motion using **mean per-joint position error (MPJPE)**, **root translation error (RTE)**, and **acceleration error (Accel.)**, respectively. We also report **Procrustes-aligned MPJPE (PA-MPJPE)** and **2D endpoint error (2D EPE)** as secondary metrics. Lower values indicate better performance for all five metrics.

We represent each evaluated hand using $N = 21$ landmarks, including the wrist. For one hand in one frame, let $P_j, G_j \in \mathbb{R}^3$ denote predicted and ground-truth joint positions in the source-camera frame, expressed in millimeters, with wrist index $j = 0$. Define wrist-relative coordinates $p_j = P_j - P_0$ and $g_j = G_j - G_0$.

MPJPE measures the mean distance between corresponding wrist-relative joints:

$$E_{\text{MPJPE}} = \frac{1}{N} \sum_{j=0}^{N-1} \|p_j - g_j\|_2 . \tag{4}$$

Subtracting each hand's own wrist removes global translation. No rotation or scale alignment is applied, so this metric retains errors in hand orientation, scale, and articulation. The wrist contributes zero error but remains in the 21-joint average. MPJPE is reported in millimeters.

RTE measures the distance between the predicted and ground-truth wrists in camera coordinates:

$$E_{\text{RTE}} = \|P_0 - G_0\|_2 . \tag{5}$$

It is reported in millimeters and measures absolute hand localization using the wrist of the evaluated joint set.

Acceleration error measures agreement with ground-truth hand motion over time. For each hand and clip, we transform predicted and ground-truth joints into a common world frame using the dataset's camera-to-world transforms, then subtract each sequence's wrist independently at every frame. Writing the transformed positions in millimeters as $P^w_{t,j}$ and $G^w_{t,j}$, define wrist-relative world-space joints $\bar{P}_{t,j} = P^w_{t,j} - P^w_{t,0}$ and $\bar{G}_{t,j} = G^w_{t,j} - G^w_{t,0}$. With the centered second difference $\Delta^2 \bar{P}_{t,j} = \bar{P}_{t+1,j} - 2\bar{P}_{t,j} + \bar{P}_{t-1,j}$, and analogously for ground truth, the error at the middle frame is

$$E_{\text{Accel},t} = \frac{1}{N} \sum_{j=0}^{N-1} \left\|\Delta^2 \bar{P}_{t,j} - \Delta^2 \bar{G}_{t,j}\right\|_2 . \tag{6}$$

We report this quantity in **mm/frame²**, using consecutive steps in the ordered evaluation sequence without dividing by elapsed time in seconds. Comparisons therefore require consistent frame rates and sampling intervals. The metric measures wrist-relative motion agreement, excluding acceleration of the global wrist trajectory. A smoother prediction alone need not have a lower error.

PA-MPJPE measures joint error after aligning prediction to ground truth with a similarity transform fitted independently for each hand and frame. Let $(s^\star, Q^\star, b^\star)$ minimize $\sum_j \|sQp_j + b - g_j\|_2^2$, where $s \geq 0$ is a uniform scale, $Q \in \text{SO}(3)$ is a rotation without reflection, and $b \in \mathbb{R}^3$ is a translation. The reported error is

$$E_{\text{PA-MPJPE}} = \frac{1}{N} \sum_{j=0}^{N-1} \|s^\star Q^\star p_j + b^\star - g_j\|_2 . \tag{7}$$

The alignment minimizes squared distances, while the score averages Euclidean distances in millimeters. This metric measures the residual joint-configuration error after alignment and does not retain absolute localization or global scale error.

2D EPE measures agreement between the image projections of the evaluated 3D joints:

$$E_{\text{2D}} = \frac{1}{N} \sum_{j=0}^{N-1} \|\pi(P_j) - \pi(G_j)\|_2 , \tag{8}$$

where π is the calibrated projection associated with the evaluation image and the error is in pixels. Both joint sets use the same projection within each run. This metric evaluates projections of the reconstructed 3D joints, rather than outputs of the separate 2D prediction branch. Comparisons across methods require the same image projection and resolution.

For ARCTIC, π projects into the full-resolution 2800×2000 globally undistorted image using the original intrinsic matrix.