EGOH4RT: BRINGING EGOCENTRIC HANDS AND CAMERA 4D RECONSTRUCTION TOGETHER

Anonymous authors
Paper under double-blind review

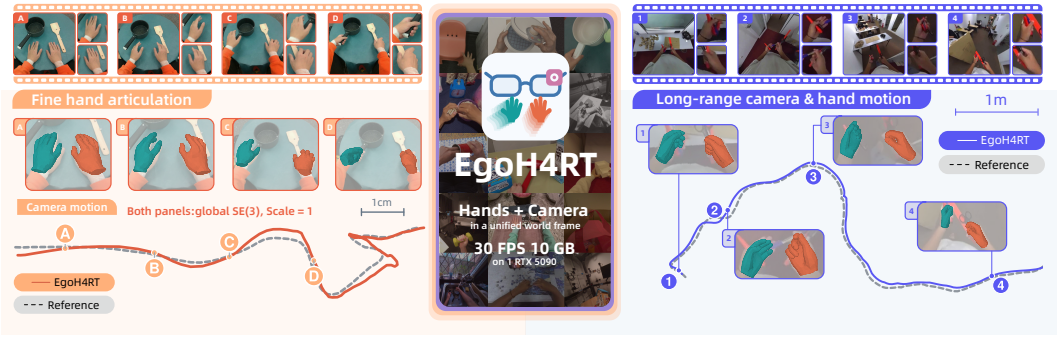


Figure 1: **EgoH4RT overview.** Articulated hands and camera poses are reconstructed from egocentric video in one metric world frame. **Left:** A short clip demonstrates fine hand articulation and reprojection, along with camera motion. **Right:** A long sequence demonstrates camera and bimanual trajectories in a shared world frame.

ABSTRACT

Reconstructing hands from egocentric video requires preserving fine hand articulation while recovering camera motion in a consistent world coordinate system. This becomes more difficult over long videos, where errors in local estimates can accumulate. To address these challenges, we introduce EgoH4RT, a framework that combines the articulation priors of a pretrained hand model with the geometric understanding of a 3D foundation model. Hand queries connect these complementary representations, drawing on evidence across frames to recover detailed bimanual poses alongside metric camera motion. To carry these local predictions into a coherent reconstruction of the full video, we jointly optimize camera constraints spanning short and long time intervals, then refine world-space hand motion within the camera path. The model learns from approximately 2,600 hours of video across 14 egocentric sources through a training recipe that accommodates different joint definitions and incomplete annotations. Experiments show that EgoH4RT outperforms multi-stage reconstruction pipelines in world-space hand and camera reconstruction and methods built on pretrained video diffusion models in camera-space hand reconstruction.

1 INTRODUCTION

Learning from egocentric human video is emerging as an important paradigm for training embodied intelligence models (Hoque et al., 2026; Punamiya et al., 2026). In these videos, observed hand motion reflects both hand movement and motion of the head-mounted camera. Even a hand that is stationary in the world can appear to move as the camera moves. Recovering hand motion in a shared world coordinate system therefore requires estimating both camera-space hand poses and the camera trajectory.

Most hand reconstruction methods recover camera-space hand poses without estimating camera motion (Pavlakos et al., 2024; Potamias et al., 2025; Wang et al., 2026d; Liu et al., 2026). World-space pipelines combine hand reconstruction with camera trajectories obtained from separate SLAM

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

or geometry estimators (Zhang et al., 2025a; Xu et al., 2026; Zeng et al., 2026). Errors in the hand and camera estimates can compound in world-space trajectories, while separate camera estimation adds runtime overhead. Recent unified models jointly predict human and camera motion within short video windows (Lyu et al., 2026; Zhu et al., 2026). However, extending these predictions by chaining neighboring windows can accumulate drift without directly constraining motion between distant frames. These limitations motivate an estimator that combines *accurate camera-space hand reconstruction* with *consistent world-space camera trajectories*, both within short windows and across long sequences.

To this end, we present **EgoH4RT**, a unified framework that brings **Ego**centric **H**and and Camera **4**D **R**econstruction **T**ogether. Motivated by recent advances in camera motion estimation within 3D foundation models (3DFMs) (Wang et al., 2025a; 2026c), we adopt a pretrained 3D geometry backbone to transfer this capability to our framework. However, hand reconstruction requires recovering many articulated degrees of freedom under frequent occlusion. We therefore incorporate features from a pretrained hand mesh reconstruction (HMR) model (Potamias et al., 2025; Pavlakos et al., 2024), complementing the geometric backbone with priors on hand pose and shape.

EgoH4RT takes egocentric RGB video as input and recovers a **metric-scale** camera trajectory and **bimanual** MANO-hand motion (Romero et al., 2017). For short-window reconstruction, we construct a query for each hand and frame within a 16-frame window. Each query combines WiLoR features for HMR (Potamias et al., 2025), CLIFF-style camera conditioning (Li et al., 2022), and target-frame context from the 3DFM: Pi3X (Wang et al., 2026c). Temporal attention aggregates hand evidence across frames, while cross-attention retrieves visual and geometric context for hand decoding. Camera poses and metric scale are predicted from the shared video representation.

For long videos, we apply the same window model to consecutive frames and frames sampled farther apart. The resulting relative camera motion constraints cover different temporal spans and are jointly optimized to recover the full camera trajectory. We then fix the camera trajectory and refine the world-space hand motion.

We train EgoH4RT on approximately 2,600 hours of egocentric video (184 million frames) from 14 sources. We handle different joint definitions with anatomical mappings and compute losses only for annotated targets. Available metric camera translation and depth annotations also supervise a single scale factor for each 16-frame window. On the HOT3D (Banerjee et al., 2025) evaluation recordings, EgoH4RT reduces W-MPJPE for the right hand over complete sequences by **58.9%** relative to HaWoR (Zhang et al., 2025a). As video duration increases, EgoH4RT's processing time grows at less than one-quarter the rate of HaWoR's.

In summary, our contributions are:

- **A single feed-forward model** that jointly estimates bimanual MANO motion and a metric camera trajectory from egocentric RGB video.
- **A deterministic post-processing method** that combines predictions from short windows into complete camera and hand trajectories for long videos.
- **An egocentric training recipe** that integrates sources with different combinations of camera pose, hand joint, MANO, and metric depth annotations.

2 Related Work

Camera-space hand reconstruction. MANO provides a compact differentiable hand representation (Romero et al., 2017). Single-image methods recover parametric hands from monocular observations (Pavlakos et al., 2024; Potamias et al., 2025; Yang et al., 2026); video methods add temporal context for bimanual reconstruction, visibility reasoning, and motion completion (Wang et al., 2026d; Liu et al., 2026; Sun et al., 2026). Unlike these camera-space methods, EgoH4RT jointly recovers hands and metric ego-camera motion.

Feed-forward and long-horizon video geometry. Classical structure-from-motion uses feature matching and iterative optimization (Schönberger & Frahm, 2016), whereas recent feed-forward models predict cameras and dense geometry directly (Wang et al., 2025a; 2026c; Lin et al., 2026;

Wang et al., 2026a). Related query-based models read geometry or appearance from scene representations (Zhang et al., 2026; Jiang et al., 2025). Long-sequence systems either assemble window reconstructions (Zhu et al., 2026; Zhang et al., 2025b; Han et al., 2026) or maintain streaming state (Zhuo et al., 2026; Chen et al., 2026; Cheng et al., 2026). Adjacent-window chains and causal updates remain susceptible to drift. EgoH4RT instead jointly optimizes redundant relative-motion constraints from Pi3X windows of consecutive frames and windows of frames sampled farther apart.

World-space and unified egocentric reconstruction. Egocentric systems recover camera, depth, and dynamic geometry through explicit assembly or shared prediction (Yuan et al., 2025; Li et al., 2025). World-space hand and body methods often combine hand estimates with separately recovered camera, depth, scene observations, motion priors, or per-video optimization (Zhang et al., 2025a; Xu et al., 2026; Yu et al., 2025; Zeng et al., 2026; Yi et al., 2025); camera-conditioned models may instead require an external metric trajectory (Patel et al., 2025). ReViV uses motion priors to complete unobserved hand intervals (Lyu et al., 2026). MINT directly predicts camera, field of view, bimanual MANO, and hand presence, while RESELF couples Pi3X scene and camera predictions to body reconstruction (Zhu et al., 2026; Guan et al., 2026). EgoH4RT jointly recovers camera-space hands and metric ego-camera motion without separately running hand and camera systems.

Scaling geometric supervision across datasets. Video-geometry models combine datasets and camera, depth, or motion supervision (Wang et al., 2025a; 2026c; Zhang et al., 2026; Lu et al., 2026). Egocentric hand–camera data additionally differ in joint conventions, MANO availability, and physical camera/depth scales. MINT's final metric camera requires Stage-2 camera-head fine-tuning on a small single-rig corpus; its reported correction reduces under-scaling on HOT3D but overshoots on ARCTIC. StableHand uses benchmark-specific motion priors (Zhu et al., 2026; Zeng et al., 2026). Other geometry models use scale heads or tokens, or an auxiliary metric-depth estimator (Wang et al., 2026b; Cheng et al., 2026; Lin et al., 2026). EgoH4RT instead trains once across datasets, applying available joint, MANO, camera, and depth labels and mapping camera motion and depth geometry into a common metric space.

3 METHOD

3.1 PROBLEM FORMULATION AND SYSTEM OVERVIEW

EgoH4RT recovers a metric camera trajectory and the motion of both hands from a monocular egocentric video with known camera intrinsics. The input consists of $N \geq 16$ RGB frames $\mathbf{I}_n$, intrinsics $\mathbf{K}$, and left- and right-hand bounding boxes that may be missing in individual frames. Writing $\mathcal{S} = \{L, R\}$ for the two hand sides, we define the input as

$$\mathcal{V}_N = \left(\mathbf{K}, \{(\mathbf{I}_n, \{\mathbf{b}_n^s\}_{s\in\mathcal{S}})\}_{n=1}^N\right), \tag{1}$$

where $\mathbf{b}_n^s \in [0,1]^4$ is a normalized bounding box and $\mathbf{b}_n^s = \varnothing$ denotes a missing detection. The recovered output is the complete camera trajectory together with left- and right-hand states over the sequence,

$$\left(\{\mathbf{C}_n\}_{n=1}^N, \{\mathbf{h}_{n,L}, \mathbf{h}_{n,R}\}_{n=1}^N\right). \tag{2}$$

Each camera-to-world pose $\mathbf{C}_n \in \mathbb{SE}(3)$ has translation measured in metres, and the first camera pose $\mathbf{C}_1$ is set to the identity transformation to fix the rigid gauge. We represent each hand in camera coordinates by its MANO state,

$$\mathbf{h}_{n,s} = (\mathbf{R}_{n,s}^g, \boldsymbol{\theta}_{n,s}, \boldsymbol{\beta}_{n,s}, \mathbf{p}_{n,s}), \tag{3}$$

where $\mathbf{R}^g$ is the global hand rotation, $\boldsymbol{\theta}$ contains 15 articulated rotations, $\boldsymbol{\beta} \in \mathbb{R}^{10}$ contains the ten MANO shape coefficients, and $\mathbf{p}$ is the camera-space translation in metres. Side-specific MANO layers generate joints and vertices, which $\mathbf{C}_n$ transforms into a common metric world frame.

Within each 16-frame window, EgoH4RT uses constructed hand queries to retrieve hand evidence from the pretrained Pi3X video-geometry representation. The same representation predicts camera motion and geometry within each window. For long videos, relative camera observations from sliding windows sampled at multiple frame strides form a pose graph whose optimization produces the sequence trajectory, followed by hand refinement with the camera fixed. We train the window predictor across egocentric sources with heterogeneous hand, camera, and geometry labels.

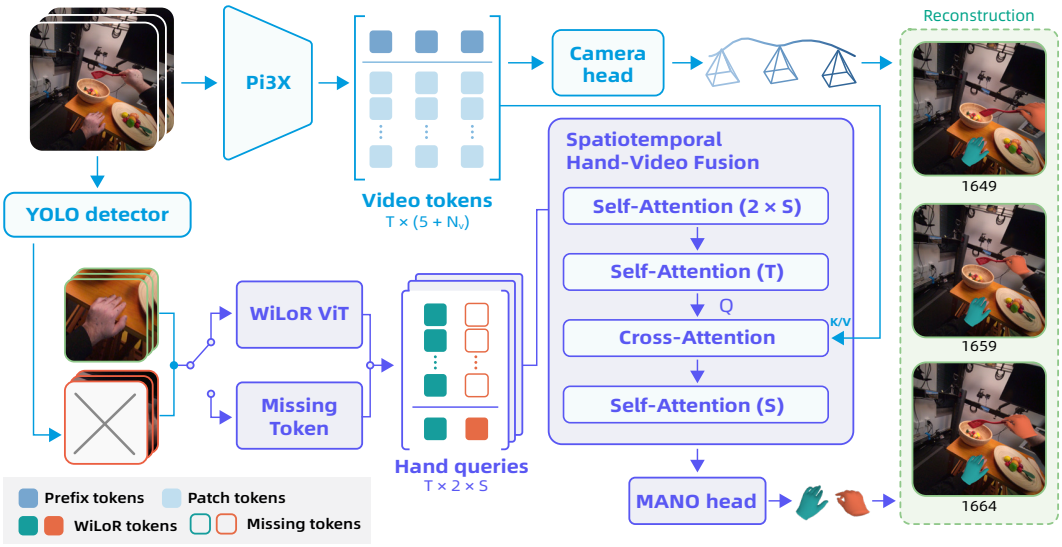


Figure 2: **Short-window reconstruction pipeline.** Constructed hand queries retrieve window-wide Pi3X context for MANO decoding of both hands and metric camera-space translation. Camera motion and metric scale are decoded by the geometry branch.

3.2 Hand queries over video geometry

The 16-frame EgoH4RT window model uses pretrained Pi3X features conditioned on known intrinsics to construct a query for each frame and hand side (Fig. 2). We initialize Pi3X from the video-geometry branch of Track4World (Wang et al., 2026c; Lu et al., 2026) and supply intrinsics through ray conditioning.

Constructing hand queries. Each hand query combines WiLoR-derived structured crop evidence with a constructed instance token (Appendix A.1). The crop evidence encodes hand pose, shape, and weak-camera information. The instance token combines hand side, normalized box location, crop position and size relative to the camera through CLIFF-style cues (Li et al., 2022), camera intrinsics, and a pooled target-frame Pi3X feature; the same feature is added to the crop tokens. When YOLO supplies no detection, learned missing-observation tokens substitute for the crop tokens and box-dependent cues are omitted, while hand side, intrinsics, and target-frame Pi3X content remain.

Retrieving and decoding hand evidence. First, self-attention within each query exchanges crop and instance information. Temporal self-attention then gathers evidence for the same hand from frames with valid hand detections, supplying information to queries with missing observations. The central cross-attention stage retrieves visual and geometric evidence from Pi3X features across the full window. Final within-query attention redistributes the fused evidence (Appendix A.2).

WiLoR-initialized MANO heads decode global and finger rotations and shape. A separate translation head predicts metric camera-space hand translation from the fused instance token. Pi3X geometry decoders predict camera poses, point maps, and a positive per-window scale. Pi3X applies this scale to camera translation and point geometry.

3.3 Training across heterogeneous egocentric sources

We supervise both the direction and physical scale of camera translation. The direction loss compares normalized translation vectors, so it is independent of the predicted per-window scale. Metric translation targets constrain the displacement in physical units, while metric depth or point-geometry targets supervise the predicted scene geometry and its scale.

Combining supervision across sources. We organize the 14 egocentric training sources (Liu et al., 2022; Banerjee et al., 2025; Fan et al., 2023; Wang et al., 2025b; Hoque et al., 2026; Liu et al.,

Table 1: **Training sources by supervision branch.**

Supervision branch	Target type	Sources (sampling probability)
Hand	MANO fields	HOI4D (4.5%), HOT3D (9.8%), ARCTIC (2.7%), HOCap (1.6%), TACO (4.2%), H2O (2.1%), Xperience (9.4%)
	Direct joint annotations	EgoDex (28.0%), EgoVerse (31.0%), Xperience (9.4%)
Camera	Metric camera poses	All sources except HOCap; HD-EPIC (6.6%) is camera-only
Geometry	Metric depth / point geometry	HOI4D (4.5%), TACO (4.2%), H2O (2.1%), Xperience (9.4%)

2024; Kwon et al., 2021; Punamiya et al., 2026; Perrett et al., 2025; Ropedia, 2026) by supervision branch and target type in Table 1. A source may contribute to multiple rows, and the shown sampling probabilities are rounded after grouping HOT3D and EgoVerse variants. Anatomical mappings and joint masks reconcile different hand definitions, and unavailable fields are not supervised. Joint-only supervision uses protocol-verified distal landmarks with comparatively consistent definitions across sources; Appendix A.3 specifies the source-specific masks.

The training objective combines three supervision branches.

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{hand}} + \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{geom}}. \tag{4}$$

The hand branch $\mathcal{L}_{\text{hand}}$ supervises MANO parameters, joints, vertices, metric hand translation, and reprojection, with anatomical and temporal consistency. The camera branch $\mathcal{L}_{\text{cam}}$ supervises relative rotation, translation direction, metric camera translation, and log-displacement magnitude. The geometry branch $\mathcal{L}_{\text{geom}}$ supervises point-map shape, metric scale, and shift. Appendices A.3 and A.4 detail source routing, validity masks, normalized reductions, and losses.

Training clips and missing observations. We optimize the final eight Pi3X decoder layers and the query, fusion, and hand-decoding modules while the WiLoR feature extractor and early Pi3X layers remain frozen. Training clips contain 16 frames at a sampled temporal stride, with each otherwise visible frame–hand observation independently masked with probability 15% while retaining its target. Hidden inputs use the same replacement tokens and omitted box cues as missing detections at inference. Section 4.1 gives the training settings.

3.4 Long-video pose graph optimization and hand refinement

For long videos, we assemble window predictions through camera pose graph optimization (PGO) followed by hand refinement (Fig. 3).

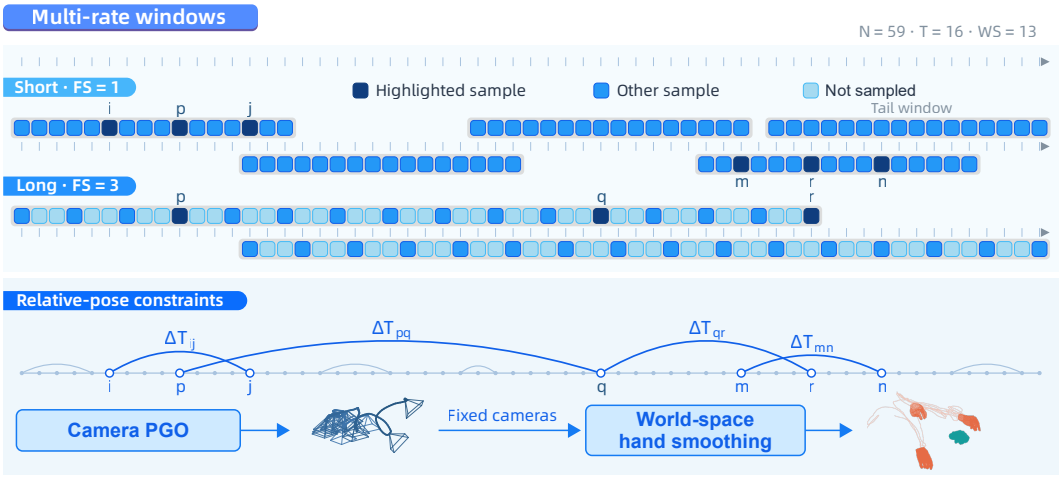


Figure 3: **Full-sequence reconstruction pipeline.** Windows of consecutive frames provide local observations, while windows of frames sampled farther apart cover longer time spans. Pose graph optimization first estimates a camera trajectory for the full sequence, and a second stage then refines the MANO trajectories with that camera trajectory fixed.

Window sampling and camera measurements. Frame stride (FS) specifies the spacing between frames within a window, and window stride (WS) specifies the spacing between window starts. Multiple frame strides provide observations over different temporal spans. For each frame stride, the final legal start is included for tail coverage. Each camera pose is a graph node, and $\mathbf{C}_i^\omega$ denotes the window-local camera-to-world prediction for source frame i. A window ω observing source frames i and j supplies the following relative measurement.

$$\mathbf{Z}_{j\leftarrow i}^\omega = (\mathbf{C}_j^\omega)^{-1}\mathbf{C}_i^\omega \in \mathbb{SE}(3). \tag{5}$$

For a window with frame stride f, edges are constructed at within-window offsets $\delta \in \{1, 2, 4, 8, 15\}$, corresponding to source-frame gaps $j - i = f\delta$. These transforms remove each window's independent rigid gauge, while repeated observations from overlapping windows remain in the graph.

Two-stage optimization. We construct a spanning tree using breadth-first search (BFS) and propagate relative transforms from the first frame to initialize all camera poses. We keep the first pose fixed to the identity to remove the global gauge. Camera PGO fits the relative camera observations with temporal regularization. With the camera trajectory fixed, we refine contiguous visible segments of each hand track using camera-space joint observations and world-space joint temporal regularization. Refinement optimizes MANO translation, global and finger rotations, and a shape correction shared within each run. Appendix B details the objectives, solvers, and corner cases.

4 Experiments

4.1 Experimental setup

Training details. We train EgoH4RT on 16-frame clips using AdamW for 50,000 steps (100 epochs) on 16 H200 GPUs, taking approximately one week. The per-GPU batch size is 8 clips, with gradient accumulation of 1 for the first 10,000 steps and 2 for the remaining 40,000 steps, giving effective batch sizes of 128 and 256, respectively. Appendix A.5 gives further training details.

Evaluation overview. Sec. 4.2.1 evaluates camera-space bimanual reconstruction error on held-out samples. Sec. 4.2.2 evaluates short-window camera pose estimation. Sec. 4.2.3 measures error accumulation of camera poses and world-space hand trajectories reconstruction for long videos. Sec. 4.3 tests the effectiveness of missing tokens when detector does not detect a hand. Sec. 4.4 reports end-to-end efficiency. Analysis and ablation follow in Sec. 4.5.

4.2 Reconstruction accuracy

4.2.1 Camera-space bimanual hand reconstruction

Table 2 evaluates camera-space bimanual reconstruction on the 27-video HOT3D (Banerjee et al., 2025) validation split used by HaWoR (Zhang et al., 2025a) and on out-of-domain OakInk2 (Zhan et al., 2024). None of the evaluated samples are used to train EgoH4RT. Appendix C.1 details frame selection and evaluation counts.

We report MPJPE and Root Error for camera coordinates reconstruction under the following alignment protocols. **TA-MPJPE** aligns wrist position by translation alone and leaves rotation and scale unchanged. **PA-MPJPE** applies Procrustes alignment of rotation, translation, and scale. **Abs-MPJPE** averages Euclidean errors over 21 joints in current camera coordinates without any alignment. **Root Error** measures metric wrist localization error without alignment. Bold and underlined values indicate the best and second best results, respectively.

Compared with WiLoR, EgoH4RT reduces TA-MPJPE by 21.4% on OakInk2, with little change in PA-MPJPE. This suggests that our method estimates the hand's global rotation and scale more accurately while retaining similar local pose accuracy, consistent with combining frozen WiLoR features with geometric supervision. Root Error and Abs-MPJPE also decrease by 25.6% and 15.2%, respectively. ACE-Ego-Hand shows the opposite pattern: its aligned errors are lower, but its wrist and absolute joint errors are much higher. These results suggest that locating the wrist accurately helps place the entire hand correctly in camera space. Our method improves this positioning on

Table 2: **Camera-space bimanual hand reconstruction.**

	Method	TA-MPJPE (mm)↓			PA-MPJPE (mm)↓			Abs-MPJPE (mm)↓			Root Error (mm)↓		
		Left	Right	Both	Left	Right	Both	Left	Right	Both	Left	Right	Both
HOT3D	MINT	19.402	21.217	20.393	8.195	8.273	8.238	44.793	60.666	53.465	45.571	66.409	56.956
	WiLoR	22.535	19.633	20.949	7.073	7.033	7.051	51.688	49.649	50.574	46.254	42.222	44.051
	HaWoR	15.425	15.499	15.465	6.295	6.215	6.251	33.664	36.147	35.021	39.318	42.549	41.083
	HandFlow†	15.462	10.447	12.722	6.594	4.850	5.642	26.107	18.426	21.911	24.410	16.344	20.004
	ACE-Ego-Hand	14.194	8.976	11.343	7.388	4.703	5.921	20.500	18.458	19.384	18.035	16.157	17.009
	EgoH4RT	**10.742**	**8.282**	**9.398**	**4.930**	**4.187**	**4.524**	**18.232**	**14.254**	**16.059**	**16.681**	**13.062**	**14.704**
OakInk2	MINT	24.097	22.804	23.449	7.955	8.186	8.071	179.941	157.652	168.773	168.538	149.198	158.847
	WiLoR	22.563	23.994	23.280	7.152	7.400	7.276	63.704	62.458	63.079	63.521	62.508	63.013
	HaWoR	21.664	20.369	21.015	9.150	8.290	8.719	68.612	67.065	67.837	62.500	62.886	62.694
	HandFlow	27.680	29.723	28.704	8.831	9.168	9.000	72.672	69.029	70.847	87.277	85.464	86.369
	ACE-Ego-Hand‡	**16.468**‡	**14.572**‡	**15.518**‡	**5.198**‡	**5.160**‡	**5.179**‡	140.910	139.254	140.080	127.944	128.683	128.315
	EgoH4RT	18.301	18.299	18.300	7.368	7.540	7.454	**56.928**	**50.111**	**53.512**	**50.176**	**43.578**	**46.870**

† HandFlow training includes 22/27 evaluated HOT3D recordings.

‡ We deliberately exclude OakInk2 from training and reserve it for out-of-domain evaluation. We sample from its training split to obtain a larger, more representative evaluation set. Among the compared baselines, only ACE-Ego-Hand uses OakInk2 for training; it achieves lower TA-MPJPE and PA-MPJPE than all other methods on this evaluation set.

unseen OakInk2 data while preserving aligned pose accuracy, yielding more accurate joint positions without ground-truth alignment.

4.2.2 SHORT-WINDOW CAMERA POSE ESTIMATION

We construct two sets of sliding windows from the same 27 held-out HOT3D videos using different frame strides (FS). Each window contains 16 frames: FS=1 uses consecutive frames, and FS=5 takes every fifth frame. We measure camera translation errors within these windows and report the results in Table 3. Appendix C.2.1 provides more details.

Table 3: **Window-level camera estimation.** Evaluation on matched 16-frame HOT3D windows.

	FS=1					FS=5				
Method	ATE-$\mathbb{SE}(3)$	ATE-$\mathrm{Sim}(3)$	RPE-t@1	RPE-t@8	RPE-t@15	ATE-$\mathbb{SE}(3)$	ATE-$\mathrm{Sim}(3)$	RPE-t@1	RPE-t@8	RPE-t@15
StreamVGGT	6.431	3.430	3.079	15.878	25.285	18.248	10.450	9.034	36.835	47.458
DA3 Giant	4.859	1.146	1.758	9.014	14.889	14.925	1.937	5.687	27.083	35.531
VGGT-Ω	4.227	0.957	1.622	7.701	12.577	12.417	1.880	5.108	22.631	28.435
Pi3X	3.984	1.359	2.029	7.604	12.040	10.844	2.453	4.878	19.986	25.963
HaWoR†	3.274	0.849	1.970	6.107	9.112	12.802	3.722	7.835	24.069	32.385
HandFlow†	2.698	0.875	1.180	5.035	8.148	8.104	1.872	3.587	15.130	19.612
EgoH4RT	**1.342**	**0.847**	**1.053**	**2.709**	**3.916**	**2.667**	**1.634**	**2.000**	**5.277**	**6.628**

† HaWoR and HandFlow run inference on long sequences, not on individual short windows. We extract predictions for the corresponding frames from their long-sequence outputs to compute the errors.

ATE measures camera-centre RMSE after alignment. The suffix -$\mathbb{SE}(3)$ denotes rigid alignment, while -$\mathrm{Sim}(3)$ additionally allows scale alignment. **RPE-t@**k measures relative translation error between input frames k positions apart. Our model retains its advantage in trajectory accuracy and relative translation across both input frame rates. This provides accurate translation constraints over different time spans for the full-sequence reconstruction evaluated next.

4.2.3 LONG-HORIZON WORLD-SPACE CAMERA AND HAND RECONSTRUCTION

Long-horizon reconstruction on HOT3D. From the same 27 HOT3D videos, we extract 520 continuous segments with valid right-hand annotations and detections. They span 16–1,448 frames.

Following Sec. 3.4, we use the complete optimization pipeline to combine window predictions into camera and world-space hand trajectories. We evaluate the resulting camera and world-space hand trajectories and compare them with baseline methods in Table 4.

Table 4: **Full-sequence reconstruction.** Camera and world-space hand results on HOT3D.

	Camera trajectory		World-space hand		
Method	ATE-$\mathbb{SE}(3)$ (mm)↓	ATE-$\mathrm{Sim}(3)$ (mm)↓	W-MPJPE (mm)↓	WA-MPJPE (mm)↓	Accel. (mm/frame^2)↓
MINT	70.996	53.738	132.755	47.953	10.859
HaWoR	80.006	38.624	108.099	36.114	11.691
HandFlow	21.223	5.407	61.099	15.767	7.957
EgoH4RT	**5.316**	**3.336**	**44.465**	**11.607**	**6.646**

Appendix C.2.2 provides more details. For world-space MPJPE, **W-** and **WA-** use similarity alignment fitted to the first two frames and all frames, respectively. **Accel.** measures unaligned world-space joint acceleration error.

Our optimization pipeline combines window predictions into complete hand and camera trajectories in world space. After aggregation, we retain the advantage in world-space hand and camera accuracy.

Long-horizon reconstruction on Xperience. We additionally evaluate camera reconstruction on four held-out Xperience videos, each lasting 235–297 s with a trajectory diameter of approximately 10 m. These videos involve larger camera movements than HOT3D, allowing us to assess stability over longer distances. We compare reconstruction using only short-span windows with reconstruction that also incorporates long-span windows. Table 5 summarizes full-sequence trajectory errors, while Figure 4 shows relative pose errors over different temporal intervals. Adding long-span constraints reduces overall trajectory error and improves consistency over longer intervals, at the cost of higher short-interval errors. Appendix C.2.2 provides more details.

Table 5: **ATE on Xperience.**

Method	ATE-$\mathbb{SE}(3)$ (mm)↓	ATE-Sim(3) (mm)↓
Ours (local)	<u>482.612</u>	<u>421.224</u>
Ours (long span)	**310.193**	**268.601**
HaWoR	1593.024	1500.451
HandFlow	2330.169	2258.552
MINT	2608.152	2345.939
LingBot-Map	1421.826	724.688
ABot-Recon	2371.269	1873.781

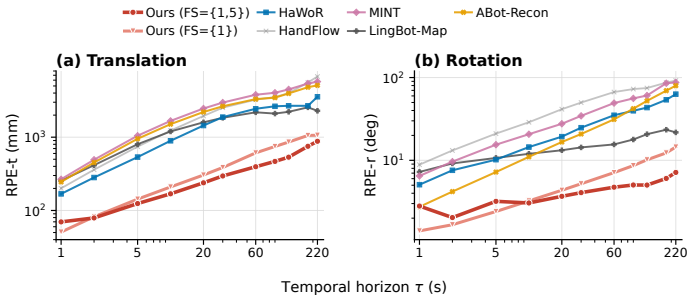


Figure 4: **Xperience multi-horizon RPE.**

4.3 Reconstruction under missing detections

To prevent methods from benefiting by skipping difficult hands, we adopt the coverage-aware evaluation used in prior work such as ACE-Ego-Hand (Liu et al., 2026) and MINT (Zhu et al., 2026), denoted by the "**-p**" suffix (Table 6). Correctly matched predictions contribute their pose errors, while each missed hand contributes the error of a canonical MANO placeholder with identity rotation and mean shape at the camera origin. These errors are averaged over all true positives and false negatives.

We apply this same scoring rule to all methods and vary our consecutive-miss threshold e. At $e = 0$, every detector miss is scored using the placeholder. At $e = 8$, learned missing tokens replace unavailable WiLoR tokens to recover hand poses during short detection gaps. Recovered predictions must still satisfy the evaluator's matching criterion: matched predictions contribute their pose errors, and unmatched ground-truth hands receive the placeholder error. Increasing e from 0 to 8 lowers MPJPE-p from 17.732 to 15.643 mm, supporting the ability of missing tokens to recover hand poses during detection gaps. Longer gaps can also introduce false positives; Appendix C.3 reports the counts and full sweep.

Table 6: **HOT3D coverage-aware reconstruction.**

Method	MPJPE-**p** (mm)↓	PA-MPJPE-**p** (mm)↓
WiLoR	29.496	20.378
HaWoR	42.009	31.377
MINT	24.945	13.098
ACE-Ego-Hand	24.541	12.740
EgoH4RT ($e = 0$)	17.732	12.882
EgoH4RT ($e = 4$)	<u>16.316</u>	<u>11.351</u>
EgoH4RT ($e = 8$)	**15.643**	**10.592**

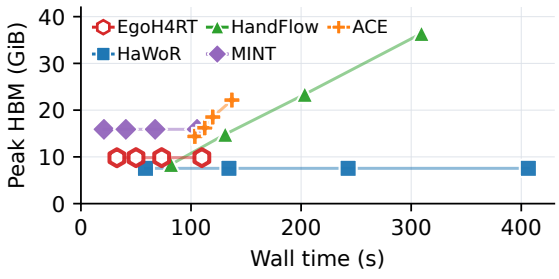


Figure 5: **Runtime and memory.** Wall time and peak GPU memory at 128, 512, 1,024, and 1,802 frames.

4.4 END-TO-END EFFICIENCY

We measure runtime from raw MP4 input to final predictions on a single H100 80GB GPU. Times average two runs after warm-up. Appendix C.4 details the protocol.

Figure 5 shows that our peak memory stays within 9.813–9.837 GiB from 128 to 1,802 frames. At 1,802 frames, our runtime is 109.8 s, close to MINT with less memory. HaWoR uses less memory but takes substantially longer, while HandFlow's memory grows with sequence length. These results show a favorable balance of runtime and memory across the tested lengths.

4.5 ANALYSIS AND ABLATION

Multi-rate PGO sampling. Table 5 and Figure 4 show that long-span windows improve overall trajectory accuracy and long-interval motion accuracy on Xperience. Table 7 shows improved world-space hand accuracy under otherwise identical optimization settings on HOT3D. Appendices C.5.1 and C.5.2 report the complete ablation and sampling sensitivity results.

Fixed-camera hand refinement. Fixed-camera MANO refinement leaves W-MPJPE essentially unchanged, reduces WA-MPJPE by 2.5%, and reduces acceleration error by 30.3% (Table 7).

Table 7: **Optimization ablations on HOT3D.**

Configuration	W-MPJPE (mm)↓	WA-MPJPE (mm)↓	Accel. (mm/frame^2)↓
Full pipeline	**44.465**	**11.607**	6.646
w/o long-span windows	46.285	12.702	**6.358**
w/o MANO refinement	44.489	11.902	9.539

Role of WiLoR features. We evaluated a variant without WiLoR, replacing it with an eight-layer attention model with 0.1B parameters. The replacement constructed only instance tokens and queried Pi3X features directly. It converged, but the PA-MPJPE and WA-MPJPE were about 20% and 25% higher than those of HaWoR, respectively. WiLoR was pretrained on a broader collection of cropped hand images, including non-egocentric images. These results suggest that its features improve MANO parameter prediction and allow the Pi3X backbone to focus on video and scene geometry instead of learning detailed hand shape from scratch.

Frame conditioning. D4RT-style learned temporal embeddings (Zhang et al., 2026) led to slow optimization convergence when constructing the hand queries. Defining each frame's temporal embedding as the mean of its five Pi3X prefix tokens instead reduced the number of optimization steps required for convergence, rather than the wall-clock time per step. The pooled embedding is projected to the query dimension and added to both the instance token and WiLoR tokens.

Metric camera supervision. In development experiments that omitted Pi3X scale-head supervision and translation-direction loss, the camera objective was difficult to optimize and plateaued at a substantially higher value. Scale supervision constrains translation magnitude, while the direction term supervises the orientation of camera displacement. This observation motivated the supervision design, but does not isolate either term's contribution or establish metric-scale accuracy.

5 DISCUSSION

Conclusion. EgoH4RT combines pretrained hand features with video geometry for joint hand and metric camera reconstruction from egocentric video. Multi-rate window fusion produces complete world-space trajectories, while anatomical joint mappings and available annotations enable training across heterogeneous sources. Experiments demonstrate accurate reconstruction within and across windows, robustness to brief detection gaps, and efficient long-video processing.

Limitations. Our method relies on an external detector and MANO, without modeling contact or geometry beyond MANO. Reconstruction is offline, PGO cannot correct biases shared across windows, and accuracy varies across datasets. Future work will explore online reconstruction and how training-data scale and composition affect generalization.

AI USE STATEMENT

Large language models were used to assist with language polishing of the manuscript.

ETHICS STATEMENT

All experiments use released research datasets. Because egocentric video may contain sensitive information, models and reconstructed motion should follow the licenses, consent conditions, and privacy safeguards of their sources. Performance may vary with visibility, appearance, handedness, and device. The system is for research and is unsuitable for safety-critical tracking.

REPRODUCIBILITY STATEMENT

The paper specifies training duration, evaluation sets, detector, coordinate conventions, alignment, aggregation, and optimization. The supplement provides per-sequence predictions, evaluation membership, timing and aggregation rules, and validation checks.

REFERENCES

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7061–7071. IEEE, 2025.

Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, et al. Geometric context transformer for streaming 3d reconstruction. *arXiv preprint arXiv:2604.14141*, 2026.

Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyang Guo, and Hao Wang. Longstream: Long-sequence streaming autoregressive visual geometry. *arXiv preprint arXiv:2602.13172*, 2026.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12943–12954. IEEE, 2023.

Kai Guan, Minchao Jiang, Ruichen WangLi, Wentao Zhu, and Lei Zhang. Seeing the world and the self from egocentric video. *arXiv preprint arXiv:2609.01276*, 2026.

Jiarong Han, Jincheng Xiong, Yuzhou Liu, Linzhe Shi, Changjie Wu, Ning Guo, Mu Xu, Hang Zhang, and Ming Qian. Revisiting local context for long-horizon streaming 3d reconstruction. *arXiv preprint arXiv:2608.27529*, 2026.

Ryan Hoque, Peide Huang, David Yoon, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations*, volume 2026, pp. 4218–4237, 2026.

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4918–4929. IEEE, 2025.

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10138–10148, October 2021.

Gen Li, Yutong Chen, Yiqian Wu, Kaifeng Zhao, Marc Pollefeys, and Siyu Tang. Egom2p: Egocentric multimodal multitask pretraining. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10830–10843. IEEE, 2025.

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 *European Conference on Computer Vision*, pp. 590–606. Springer, 2022.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 141261–141285, 2026.

Yufei Liu, Xixi Wang, Hao Li, Ganlong Zhao, Kaitong Cai, Chengkai Jin, Chunxiao Liu, Jianbo Liu, Siyuan Huang, Xingang Pan, and Hongsheng Li. Ace-ego-hand: Repurposing video diffusion models for occlusion-robust egocentric 3d hand motion recovery, 2026.

Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21740–21751. IEEE, 2024.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20981–20990. IEEE, 2022.

Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4world: Feedforward world-centric dense 3d tracking of all pixels. In *European Conference on Computer Vision*, pp. 184–204. Springer, 2026.

Xiaozhong Lyu, Gen Li, Zhiyin Qian, Xucong Zhang, Marc Pollefeys, and Siyu Tang. Reviv: Reconstructing the viewer and the view in 4d from monocular egocentric video. In *European Conference on Computer Vision*, pp. 428–446. Springer, 2026.

Chaitanya Patel, Hiroki Nakamura, Yuta Kyuragi, Kazuki Kozuka, Juan Carlos Niebles, and Ehsan Adeli. Uniegomotion: A unified model for egocentric motion reconstruction, forecasting, and generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10318–10329. IEEE, 2025.

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9826–9836. IEEE, 2024.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. Hd-epic: A highly-detailed egocentric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12242–12254. IEEE, 2025.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, Aniketh Cheluva, Pranav Kuppili, Yangcen Liu, Dhruv Patel, Aidan Gao, Hye-Young Chung, Ryan Co, Renee Zbizika, Jeff Liu, Xiaomeng Xu, Haoyu Xiong, Geng Chen, Sebastiano Oliani, Wenkai Xuan, Chenyu Yang, Xi Wang, James Fort, Richard Newcombe, Josh Gao, Jason Chong,

Garrett Matsuda, Aseem Doriwala, Marc Pollefeys, Robert Katzschmann, Xiaolong Wang, Shuran Song, Judy Hoffman, and Danfei Xu. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

Ropedia. Xperience-10m: A large-scale egocentric multimodal dataset with structured 3d/4d annotations, 2026. Dataset.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Zhihao Sun, Tong Wu, Ruirui Tu, Daoguo Dong, and Zuxuan Wu. Unihand: A unified model for diverse controlled 4d hand motion modeling. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Jikai Wang, Qifan Zhang, Yu-Wei Chao, Bowen Wen, Xiaohu Guo, and Yu Xiang. HO-cap: A capture system and dataset for 3d reconstruction and pose tracking of hand-object interaction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025b.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026c.

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. *arXiv preprint arXiv:2606.30308*, 2026d.

Mingxi Xu, Bowen Duan, Yi Gu, Zhengyang Shen, Renjing Xu, and Yutao Yue. Handflow: Fully generative 4d hand recovery with flow matching. *arXiv preprint arXiv:2607.11221*, 2026.

Xitong Yang, Devansh Kukreja, Don Pinkus, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jia-Wei Liu, Nicolás Ugrinovic, Anushka Sagar, et al. Sam 3d body: Robust full-body human mesh recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7209–7219, 2026.

Brent Yi, Vickie Ye, Maya Zheng, Yunqi Li, Lea Müller, Georgios Pavlakos, Yi Ma, Jitendra Malik, and Angjoo Kanazawa. Estimating body and hand motion in an ego-sensed world. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7072–7084. IEEE, 2025.

Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. Dyn-hamr: Recovering 4d interacting hand motion from a dynamic camera. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27716–27726. IEEE, 2025.

Chengbo Yuan, Geng Chen, Li Yi, and Yang Gao. Self-supervised monocular 4d scene reconstruction for egocentric videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8863–8874. IEEE, 2025.

Huajian Zeng, Chaohua Yao, Yuantai Zhang, Jiaqi Yang, Rolandos Alexandros Potamias, and Xingxing Zuo. Stablehand: Quality-aware flow matching for world-space dual-hand motion estimation from egocentric video. *arXiv preprint arXiv:2605.18553*, 2026.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 445–456. IEEE, 2024.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K Barral, Raia Hadsell, et al. Efficiently reconstructing dynamic scenes one d4rt at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7382–7392, 2026.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. Hawor: World-space hand motion reconstruction from egocentric videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1805–1815. IEEE, 2025a.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, volume 2025, pp. 82863–82886, 2025b.

Zijie Zhu, Weiren Cai, Yizhou Wang, Zhenjie Yang, Yide Liu, Jiahao Chen, and Guanqi He. Mint: A unified model for world-space camera and hand motion estimation from scalable egocentric pipeline supervision. *arXiv preprint arXiv:2609.04958*, 2026.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026.

A Model and Training Details

A.1 Hand-query construction

For each frame and hand, the hand query consists of 18 structured tokens from WiLoR and one instance token. Each token has $D = 1280$ channels, giving a query of shape $[19, D]$.

The frozen WiLoR ViT-H encoder processes resized hand crops; left-hand crops are mirrored into its canonical right-hand orientation. The WiLoR tokens comprise 16 pose tokens, one shape token, and one weak-camera token. We retain these structured tokens and discard WiLoR's patch tokens.

The instance token encodes hand side, bounding box location, camera intrinsics, and time. We encode these inputs as follows, sum the resulting D-channel features, and apply layer normalization:

- **Hand side.** A learned embedding identifies the left or right hand.
- **Normalized box centre.** For a box centre (u, v) in an image of width W and height H, compute $[u/W, v/H]$. Fourier encoding followed by a learned projection produces a feature of shape $[D]$.
- **CLIFF box features.** Let (c_x, c_y) be the camera principal point, f_x, f_y the focal lengths, and b the square crop side length, all in pixels. Compute $f = (f_x + f_y)/2$ and the three values $[(u - c_x)/f, (v - c_y)/f, b/f]$ (Li et al., 2022), then linearly project them to D channels.
- **Camera intrinsics.** Normalize the intrinsics as $[f_x/W, f_y/H, c_x/W, c_y/H]$ and linearly project these four values to D channels.
- **Time embedding.** Average the five 1024-channel Pi3X prefix tokens for the frame and linearly project the result to D channels. The same time embedding is also added to each of the 18 WiLoR tokens.

A.2 Temporal fusion and prediction heads

After constructing the hand queries, we stack them into $\mathbf{Q}$ of shape $[T, K, D]$ for each hand, where $T = 16$, $K = 19$, and $D = 1280$. Pi3X provides five prefix tokens and S spatial tokens per frame, forming context of shape $[T, 5 + S, D_p]$, where $D_p = 1024$. Separate linear projections map prefix and spatial tokens to D channels. Flattening the frame and token axes gives context of shape $[T(5 + S), D]$. The spatiotemporal fusion module then applies the following operations:

1. **Within-frame self-attention.** Keep the input in shape $[T, K, D]$, treating T as the batch axis and K as the attention sequence axis. The K tokens interact independently within each frame. The output remains $[T, K, D]$, so no reshape is needed.
2. **Temporal self-attention.** Transpose the input from $[T, K, D]$ to $[K, T, D]$, treating K as the batch axis and T as the attention sequence axis. Each frame can attend to frames with valid hand observations and to itself. The attention output has shape $[K, T, D]$; transpose it back to $[T, K, D]$.
3. **Cross-attention.** Reshape the queries from $[T, K, D]$ to $[1, TK, D]$ and the Pi3X context from $[T, 5 + S, D]$ to $[1, T(5 + S), D]$, where the leading dimension is the batch axis. Use the context as keys and values so that every hand token can attend to all window frames. The output has shape $[1, TK, D]$; reshape it back to $[T, K, D]$.
4. **Within-frame self-attention and MLP.** As in step 1, use input shape $[T, K, D]$ with T as the batch axis and attention along K. The output remains $[T, K, D]$. Apply the MLP to each token along the D channels, preserving shape $[T, K, D]$ without an additional reshape.

Both hands follow this sequence independently and share the Pi3X context. Each attention operation and the MLP uses layer normalization before the operation and a residual connection; cross-attention normalizes queries and context separately.

The camera decoder inherited from Pi3X decodes camera extrinsics from Pi3X features. The MANO decoder inherited from WiLoR decodes global and finger rotations from the 16 fused pose tokens and shape from the fused shape token. A separate MLP predicts metric camera-space hand translation from the fused instance token. Pi3X's metric branch predicts the window scale used to scale camera translations and point geometry.

A.3 TRAINING DATA AND ANATOMICAL CORRESPONDENCE

Table 8 lists the 14 training sources, their sampling probabilities, and their available supervision. Sampling probabilities are set based on the training data we actually downloaded and could use. Some downloaded training sets are incomplete, so the mixture reflects the available subsets rather than the full official training sets. Each source contributes losses only for annotations it provides.

Hand supervision. For samples with MANO annotations, we supervise the available global rotation, finger rotations, shape, and camera-space translation. We also use compatible 16-joint and 778-vertex targets when available. These targets are enabled separately, so a sample can supervise only part of the MANO output; for example, HOCap provides finger-rotation and shape supervision. Missing parameters or geometry targets do not contribute to their respective losses.

For joint-only annotations, we match landmarks by anatomical location and arrange them in the OpenPose/MediaPipe 21-joint order. For each hand, 3D and 2D targets have shapes $[T, 21, 3]$ and $[T, 21, 2]$, with separate validity masks of shape $[T, 21]$. EgoDex supervises 14 landmarks: the thumb's interphalangeal joint and tip, and the proximal interphalangeal joint, distal interphalangeal joint, and tip of each other finger. Their zero-based indices are $[3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16, 18, 19, 20]$. EgoVerse additionally supervises the wrist at index 0. For each target, the final mask is the elementwise logical AND of this anatomical selection and the annotation's validity mask. Unselected or invalid joints contribute no loss, and joint-only samples do not receive synthesized MANO parameter targets.

Camera extrinsic supervision. All training sources except HOCap provide metric camera extrinsics. For each eligible clip, camera poses have shape $[T, 4, 4]$, with translations expressed in metres. We derive relative transforms between frames to supervise camera rotation and translation without depending on the dataset's world coordinate system. Camera supervision is enabled per clip and remains available even when hand annotations are absent, as in HD-EPIC.

Point-map supervision. HOI4D, TACO, H2O, and Xperience provide metric depth supervision, with TACO contributing only samples that satisfy the metric-depth validity checks. We back-project valid depth pixels using the camera intrinsics to obtain 3D points in each frame's camera coordinates, measured in metres. These targets supervise point-map geometry, metric scale, and XYZ shift. A pixel contributes only when its depth is valid and the clip is eligible for metric depth supervision. Sources without metric depth contribute no point-map loss but can still supervise hands or camera extrinsics.

Table 8: **Training sources and supervision.**

Source	Sampling (%)	Hand supervision	Camera	Depth
HOI4D	4.5	MANO	Yes	Yes
HOT3D clips	4.5	MANO	Yes	–
HOT3D full Aria	2.2	MANO	Yes	–
HOT3D full Quest	3.1	MANO	Yes	–
ARCTIC	2.7	MANO	Yes	–
HOCap	1.6	MANO	–	–
EgoDex	28.0	Joint-only (no wrist)	Yes	–
TACO	4.2	MANO	Yes	Yes*
H2O	2.1	MANO	Yes	Yes
EgoVerse-MP	14.6	Joint-only (with wrist)	Yes	–
EgoVerse-SfS	10.8	Joint-only (with wrist)	Yes	–
EgoVerse-Aria	5.6	Joint-only (with wrist)	Yes	–
HD-EPIC	6.6	–	Yes	–
Xperience	9.4	MANO	Yes	Yes
Total	100.0			

Sampling probabilities are rounded. Yes denotes available metric supervision; – denotes none. Hand supervision follows the categories described above.
HOT3D: The full Aria and full Quest subsets include samples with hand annotations and samples with camera annotations only. Across HOT3D sources, the left-hand MANO shape convention differs from our prediction model, so direct left-hand shape and translation losses are disabled; annotated joint and mesh geometry still provides supervision.
Partial MANO supervision: HOCap supplies finger rotations and shape only. Xperience supplies global rotation, finger rotations, translation, and 21-joint annotations, but no shape or mesh supervision.
*TACO depth is used only when aligned to RGB and valid for metric supervision.

A.4 TRAINING OBJECTIVES

Table 9: **Training losses.** Criteria and weights for the hand, camera, and point-map branches.

Branch	Loss	Criterion	Weight
Hand	Global rotation	MSE over rotation-matrix entries	1.25
	Finger rotations	MSE over 15 rotation matrices	6.0
	MANO shape	MSE over ten coefficients	0.1
	Shape magnitude	Mean absolute shape coefficient	0.1
	Camera-space translation	L1 over XYZ coordinates	1.5
	Mesh vertices	L1 over 778 vertices and XYZ	1.8
	MANO joints	L1 over 16 joints and XYZ	0.9
	Direct joint annotations	Sum of absolute XYZ errors, averaged over valid joints	0.9
	2D reprojection	Mean squared 2D distance, divided by squared image size	2.0
	Anatomical regularization	Penalty for exceeding joint-angle limits	0.9
	Temporal shape consistency	Variance across valid frames	1.0
Camera	First-frame relative rotation	MSE over rotation-matrix entries	1.5
	First-frame relative translation	L1 over XYZ coordinates	0.35
	All-pairs relative rotation	MSE over rotation-matrix entries	2.5
	All-pairs relative translation	L1 over XYZ coordinates	0.5
	Translation direction	One minus cosine similarity	0.05
	Translation magnitude	Absolute log-magnitude difference	0.05
Point map	Geometry	Confidence-weighted robust error after scale and shift alignment	0.01
	Metric scale	Absolute mean log-depth difference per clip	0.06
	XYZ shift	Robust penalty on the fitted shift	0.01

The training objective is the weighted sum of the losses in Table 9. Each loss uses the validity mask for its own target and averages only over valid entries; an empty mask contributes zero. Unless stated otherwise, L1 and MSE denote the mean absolute and mean squared errors over both valid entries and their coordinate components. Reductions are computed within each device's microbatch before distributed training averages the resulting objectives. All 3D positions are measured in metres.

Hand losses. Rotation losses compare 3×3 matrices, averaging their nine squared entry errors; finger rotation additionally averages over 15 joints. Ground-truth finger rotations include the corresponding MANO model's mean pose before conversion to matrices. The MANO-16 and direct-21 joint losses are reduced separately and each receives weight 0.9. For direct joint annotations, XYZ errors are summed rather than averaged, so the per-joint value is three times a coordinate-wise mean L1. Shape regression averages squared errors over the ten coefficients of each valid hand prediction. When shape annotations are unavailable, a zero-centred L1 prior penalizes the predicted coefficients instead.

For reprojection, project the 21 predicted joints with the known intrinsics and compute the squared Euclidean pixel distance to each valid target. Divide by $\max(H, W)^2$, where H, W are the image height and width, then average over valid joints. The complete microbatch reprojection loss contributes only when its value is below 0.05; the threshold decision receives no gradient. Temporal shape consistency computes the mean squared deviation from the mean shape of each hand within a clip, using only valid frames. It averages over the ten coefficients and over hand tracks with at least two valid frames. Anatomical regularization uses manotorch's Euler-angle constraints on twist, spread, and bend: excess beyond each allowed angle range is penalized linearly. For left hands, twist and spread signs are reversed before applying the shared constraint limits; bend is unchanged.

Camera losses. Convert predicted and target poses to world-to-camera transforms of shape $[T, 4, 4]$. For each pair of frame indices i, j, multiply the transform at j by the inverse transform at i to obtain the relative transform. This produces an array of shape $[T, T, 4, 4]$. The first-frame losses use the first row of this array, while the all-pairs losses use all T^2 entries, including self-pairs. Rotation

MSE and translation L1 are averaged over their nine and three components, respectively, and then over valid frame pairs.

The direction and magnitude losses use all frame pairs with a ground-truth displacement of at least 5 mm. For predicted and target translation vectors $\hat{\mathbf{v}}, \mathbf{v}$, let $a = \|\hat{\mathbf{v}}\|_2$ and $b = \|\mathbf{v}\|_2$ be their magnitudes in metres. Direction loss also requires $a \geq 0.001$, with no gradient through this validity check. The per-pair errors are

$$\begin{aligned} L_{\text{dir}} &= 1 - \text{clip}\left(\frac{\hat{\mathbf{v}} \cdot \mathbf{v}}{\max(a, 0.001)\max(b, 0.001)}, -1, 1\right), \\ L_{\text{mag}} &= |\log(a + 0.001) - \log(b + 0.001)|. \end{aligned} \tag{6}$$

Each is averaged over its own valid pairs. For pairs passing the validity check, the direction term compares only direction; the magnitude and XYZ terms constrain metric translation.

Point-map losses. For each frame, flatten valid predicted and target camera-space points into arrays P, G of shape $[V, 3]$, where V is the number of valid pixels. Subtract each array's mean point to obtain centred arrays P_c, G_c. Fit one scale s and one XYZ offset $\mathbf{b}$ by least squares:

$$s = \text{clip}\left(\frac{\sum P_c \odot G_c}{\max(\sum P_c \odot P_c, 10^{-6})}, 0.1, 10\right), \qquad \mathbf{b} = \text{mean}(G) - s\,\text{mean}(P). \tag{7}$$

Here $\odot$ is elementwise multiplication, and the sums include all valid points and XYZ coordinates. The aligned geometry residual is the Euclidean distance between corresponding rows of $sP + \mathbf{b}$ and G. The fit is detached when computing the geometry residual; the shift loss retains gradients through the fit. Both geometry and shift use the normalized Huber penalty below for a nonnegative distance r in metres:

$$g(r) = \begin{cases} r^2/(2 \times 0.10 \times 0.05), & r \leq 0.10, \\ (r - 0.05)/0.05, & r > 0.10. \end{cases} \tag{8}$$

For each pixel, convert the predicted confidence logit q to $c = 1 + \exp(\min(q, 10))$. The geometry loss averages $c\,g(r) - 0.2 \log c$ over valid metric-depth pixels. The shift loss averages $g(\|\mathbf{b}\|_2)$ over frames with valid depth.

Metric scale is supervised using the unaligned predicted and target depths. Clamp both depths to at least 0.001 m, subtract their logarithms, and average the differences across valid pixels and frames within each clip. Take the absolute value of this clip-level mean, then average over clips with valid metric depth. Thus, the absolute value is applied after the within-clip mean, preserving the sign of a systematic scale error during aggregation.

A.5 Training settings

Optimization settings. We train for 50,000 optimizer updates over 100 epochs using AdamW with momentum coefficients $(0.9, 0.999)$, epsilon 10^{-8}, and weight decay 0.05. The gradient norm is clipped to 1.0. Learning rates increase linearly from 10^{-8} to the peaks in Table 10 during the first 500 updates, then follow cosine decay to the listed final values. The WiLoR feature extractor and early Pi3X layers remain frozen. We train the final eight Pi3X decoder layers together with the query, fusion, and prediction modules.

Table 10: **Learning-rate schedule.** Peak rates after warmup and final rates after cosine decay.

Parameter group	Peak learning rate	Final learning rate
Default parameters and camera translation	3×10^{-5}	3×10^{-6}
Trainable Pi3X backbone layers	10^{-5}	10^{-6}
Camera rotation head	10^{-4}	10^{-5}

Training clips. Each clip contains 16 frames sampled with one integer frame stride shared across the clip. The stride varies across clips and is bounded by the frame rate of the source video, keeping adjacent input frames at most 0.5 s apart. For a 30 fps video, the maximum stride is therefore 15 frames. Each visible hand observation is independently masked with probability 15%, while its supervision targets remain available. Masked observations use learned replacement tokens, and the

features derived from the bounding box are omitted, matching the treatment of missing detections at inference.

Training volume. Training takes approximately one week on 16 H200 GPUs, with 8 clips per GPU per microbatch. The first 10,000 optimizer updates use one microbatch each; the remaining 40,000 accumulate two microbatches per update. The effective batch sizes are therefore 128 and 256 clips, respectively. This schedule totals 90,000 microbatches, each containing $16 \times 8 \times 16 = 2,048$ frames across all GPUs. The cumulative training volume is 184.32 million frame samples, which may include repeated frames; we round this to 184M in the main paper. All EgoH4RT evaluations use the resulting model.

B Long-sequence Reconstruction

B.1 Window sampling and constraints

Window sampling. The input video contains N frames, indexed from 0 to $N-1$. Each inference window contains 16 frames. Following Section 3.4, frame stride f controls the spacing between frames within a window, and window stride w controls the spacing between window starts. For a window starting at a, the input frame indices are $[a, a+f, \ldots, a+15f]$. Generate starts at $0, w, 2w, \ldots$ up to the last legal start $N-1-15f$. If this schedule misses the last legal start, append it once so that the final frame is covered. Repeat this procedure for each chosen frame stride; skip any stride with $15f > N-1$.

Relative camera constraints. Each window predicts camera-to-world poses of shape $[16, 4, 4]$. Form pairs of window positions separated by 1, 2, 4, 8, or 15 steps, keeping only pairs whose two positions lie within the window. For each pair, invert the second camera pose and multiply it by the first, as in Eq. 5. Convert both window positions to frame indices in the source video using the window start and frame stride. Collecting all windows gives frame pairs of shape $[M, 2]$ and relative transforms of shape $[M, 4, 4]$, where M is the total number of measurements. Keep repeated measurements of the same frame pair from different windows as separate constraints. Each edge e stores its two frame indices and the relative transform predicted by its window. We weight each edge by its temporal span: for frame gaps $\mathbf{d}$ of shape $[M]$, compute $\mathbf{r} = \min(\sqrt{\mathbf{d}}, 8)$ elementwise and normalize as $\mathbf{a} = \mathbf{r}/\operatorname{mean}(\mathbf{r})$. The entry a_e is the weight of measurement e, giving longer spans greater weight up to the cap while keeping the mean weight equal to one.

B.2 Camera trajectory optimization

Initialization and variables. We optimize world-to-camera poses $\mathbf{X}$ of shape $[N, 4, 4]$ over the frames covered by the graph. Use breadth-first search from the earliest frame, setting its pose to the identity. For an edge from frame i to frame j with measured relative transform $\mathbf{Z}$, initialize $\mathbf{X}_j = \mathbf{Z}\mathbf{X}_i$; when traversing the edge backward, use $\mathbf{Z}^{-1}$. Each newly reached frame inherits its pose along the traversal tree. Represent the rotations by axis-angle vectors of shape $[N, 3]$ and translations by an array of shape $[N, 3]$. Convert the rotation vectors to matrices when evaluating the objective. The first frame remains fixed throughout optimization.

Relative pose errors. For each measured frame pair (i, j), compute the current relative transform $\mathbf{X}_j\mathbf{X}_i^{-1}$. Extract its rotation and translation and compare them with the corresponding measurement from B.1. Across M edges, the rotation arrays have shape $[M, 3, 3]$ and the translation arrays have shape $[M, 3]$. For edge e, write the current rotation and translation as $R_e, \mathbf{t}_e$, and the measured values as $R_e^0, \mathbf{t}_e^0$; the superscript 0 marks a fixed measurement. The edge objective is

$$L_{\text{edge}} = \sum_e a_e \left(\|R_e - R_e^0\|_F^2 + \|\mathbf{t}_e - \mathbf{t}_e^0\|_2^2 \right). \tag{9}$$

Here a_e is the normalized edge weight from B.1. The squared norms sum the nine rotation-entry errors and three translation-component errors, respectively; translation is measured in metres.

Temporal regularization. Sort graph frames by their indices in the source video and consider triples of neighbouring entries. Keep a triple only when its two frame gaps are equal and both neighbouring pairs have measured edges. For a retained triple (i, j, k), let R_i, R_j, R_k be the current world-to-camera rotations. The two relative rotations are $U = R_j R_i^\top$ and $V = R_k R_j^\top$. For each

frame, compute its world-space camera centre as $\mathbf{c} = -R^\top \mathbf{t}$ from its world-to-camera rotation R and translation $\mathbf{t}$. The regularizers average over the retained triples:

$$\begin{aligned} L_{\text{rot}} &= \text{mean}_{\text{triples}} \, \|V - U\|_F^2, \\ L_{\text{pos}} &= \text{mean}_{\text{triples}} \, \|\mathbf{c}_k - 2\mathbf{c}_j + \mathbf{c}_i\|_2^2. \end{aligned} \tag{10}$$

The first term penalizes changes in relative rotation; the second penalizes changes in camera-centre velocity. Camera centres are used because world-to-camera translations also depend on rotation. If no triple qualifies, both regularizers are zero.

Solver. Let $A = \sum_e a_e$ be the total edge weight. The complete objective is

$$L_{\text{PGO}} = L_{\text{edge}} + A(L_{\text{rot}} + L_{\text{pos}}). \tag{11}$$

Multiplying the mean regularizers by A keeps their scale comparable to the summed edge errors as the number of measurements changes. Both regularization weights are 1. We run 100 Adam updates with a constant learning rate of 0.005, optimizing the camera rotation vectors and translations while keeping MANO predictions fixed. Only connected camera graphs enter evaluation. The resulting camera-to-world trajectory is obtained by inverting the optimized poses.

B.3 Hand trajectory refinement

We fix the optimized camera trajectory and refine each hand independently within consecutive visible runs of at least three frames. Overlapping window predictions initialize each frame using equal-weight quaternion means for rotations and arithmetic means for translation and shape; shorter runs retain this initialization. The variables are per-frame MANO translation, global and finger rotations, and one ten-dimensional shape offset shared across the run and added to the initial shapes.

The observation loss L_{obs} compares the refined camera-space joints with all raw window predictions. For a joint distance r in metres, use the pseudo-Huber penalty $g(r) = 2(0.01)^2(\sqrt{1 + (r/0.01)^2} - 1)$. Sum over the 21 joints, average over window observations within each frame, then average over frames. The smoothness loss L_{smooth} transforms the refined joints into world coordinates using the fixed cameras and computes the second difference of each joint across consecutive frames. Sum the squared XYZ components, then average over joints and valid three-frame groups within the run. Both terms backpropagate through MANO, with total objective

$$L = L_{\text{obs}} + 1.2 L_{\text{smooth}}. \tag{12}$$

We use 200 Adam steps with cosine learning-rate decay, starting at 0.001 for translation, 0.01 for rotations, and 0.1 for the shared shape offset.

C Additional Experimental Details and Results

C.1 Evaluation data and sample selection

HOT3D. We use the 27 HOT3D (Banerjee et al., 2025) videos in the validation split used by HaWoR (Zhang et al., 2025a). None is used to train EgoH4RT. These videos come from the official training partition because official test annotations are withheld. We retain frames with valid right-hand MANO annotations, at least 2 of the 21 projected joints inside the image, and hand detection confidence ≥ 0.2. Grouping consecutive valid frames and discarding segments shorter than 16 frames yields 520 continuous segments of 16–1,448 frames, with 73,627 frames in total. These segments are used for right-hand camera-space evaluation and long-horizon reconstruction; left-hand camera-space samples are selected independently as described below. HandFlow's released training split includes 22 of the 27 videos.

OakInk2. We evaluate zero-shot on 50 OakInk2 (Zhan et al., 2024) sequences; no OakInk2 data is used to train EgoH4RT. We use a deterministic sequence-level hash split with an 80/10/10 ratio. We shuffle the egocentric sequences in its 80% partition with a fixed random seed and retain complete sequences until their combined length reaches a preset budget of about 80,000 RGB frames. This gives 50 sequences with 82,152 frames before hand-validity filtering. The resulting sample is a custom evaluation subset, not the official test split. ACE-Ego-Hand is trained on OakInk2.

Xperience. We select four complete Xperience videos from captures excluded from training, with at most one video per capture. Eligible videos must last at least 30 s, have reference poses for at least 99% of frames, and have at least 30 s of consecutive valid poses. Their reference trajectories must cover at least 8 m with at least 3 m of start-to-end displacement and pass checks for abrupt pose jumps. We select four eligible videos in a fixed hash order, without using model predictions or errors. The selected videos last 235–297 s and provide 21,695 frames with matched metric-SLAM reference poses.

Hand evaluation samples. For the camera-space hand comparison in Table 2, we select left- and right-hand samples independently within each dataset. Each hand must have valid annotations, at least 2 of its 21 projected joints inside the image, and shared YOLO detection confidence ≥ 0.2. We retain continuous segments of at least 16 frames for each hand side. All methods are scored on the same valid hand samples. Table 11 lists their counts: one sample is one valid hand in one frame, so a frame with two valid hands contributes two samples to Both.

Table 11: Valid hand samples for camera-space reconstruction evaluation.

Dataset	Left	Right	Both
HOT3D	61,140	73,627	134,767
OakInk2	62,478	62,750	125,228

C.2 Reconstruction accuracy

C.2.1 Short-window camera pose estimation

Window sampling. We construct 16-frame windows from the HOT3D segments described in Appendix C.1, using frame strides FS=1 and FS=5. Window starts are five frames apart in the source video; we add the last legal window if needed to cover the end of each segment. FS=1 gives 13,795 windows from all 520 segments. FS=5 requires at least 76 frames per segment and gives 8,921 windows from 285 segments. All methods use the same evaluation windows within each FS, but the two FS settings use different sets of windows and segments.

Inference and scoring. EgoH4RT, Pi3X, DA3 Giant, VGGT-Ω, and StreamVGGT process each 16-frame window independently, without cross-window fusion or PGO. For LingBot-Map, ABot-Recon, ViPE-SLAM, HaWoR, and HandFlow, we extract the corresponding frames from long-sequence predictions, which use additional temporal context. We fit the ATE alignment separately in each evaluation window and compute RPE over frame pairs within that window. RPE offsets count positions within the 16-frame window: an offset of k corresponds to $\mathrm{FS} \times k$ frames in the source video. Each reported error is the mean of the per-window RMSEs, with every window weighted equally.

Additional results. Table 12 compares translation errors from long-sequence predictions with our independent short-window results.

Table 12: Camera translation on HOT3D at two sampling rates. Long-sequence predictions evaluated on 16-frame windows, with EgoH4RT short-window results for comparison. Errors are in mm (↓).

	FS=1						FS=5					
Method	ATE	RPE-t@1	RPE-t@2	RPE-t@4	RPE-t@8	RPE-t@15	ATE	RPE-t@1	RPE-t@2	RPE-t@4	RPE-t@8	RPE-t@15
LingBot-Map	6.031	4.365	5.897	8.573	12.993	19.032	16.267	9.646	14.921	22.943	33.068	42.782
ABot-Recon	9.196	2.551	4.811	9.184	17.268	29.972	30.744	11.147	20.931	37.043	57.971	77.591
ViPE-SLAM	2.478	1.183	1.782	2.849	4.678	7.417	7.373	3.413	5.684	9.268	13.629	16.905
HaWoR	3.274	1.970	2.805	4.096	6.107	9.112	12.802	7.835	11.543	16.933	24.069	32.385
HandFlow	2.698	1.180	1.816	2.972	5.035	8.148	8.104	3.587	6.058	10.054	15.130	19.612
EgoH4RT†	**1.342**	**1.053**	**1.359**	**1.868**	**2.709**	**3.916**	**2.667**	**2.000**	**2.786**	**3.924**	**5.277**	**6.628**

RPE offsets count positions within the 16-frame window. Multiplying an offset by FS gives the gap in frames in the source video. Each window spans 15 frames in the source video at FS=1 and 75 at FS=5. FS=1 and FS=5 contain 13,795 and 8,921 windows, respectively. All methods share the same windows within each FS. ATE uses fixed-scale $\mathbb{SE}(3)$ alignment.

† EgoH4RT processes each 16-frame window independently, without cross-window fusion or PGO. The other methods are evaluated on corresponding frames extracted from long-sequence predictions.

C.2.2 LONG-HORIZON WORLD-SPACE CAMERA AND HAND RECONSTRUCTION

HOT3D. We evaluate complete camera and right-hand trajectories on the segments described in Appendix C.1. For a segment of N frames, we use 16-frame windows with FS=$\{1, k\}$ and WS=13, where $k = \min(15, \lfloor (N-1)/15 \rfloor)$. Following Appendix B, we run 100 iterations of camera PGO and 200 iterations of hand refinement with the cameras fixed. Baselines retain their released long-sequence procedures: MINT chains windows, HaWoR uses DROID-SLAM/Metric3D v2 (Teed & Deng, 2021; Hu et al., 2024), and HandFlow uses ViPE-SLAM (Huang et al., 2025) without dynamic masks. We fit trajectory alignments separately within each segment. Camera ATE pools squared position errors over all evaluated frames before taking the square root; hand errors are averaged over valid frames and joints. Acceleration errors use only consecutive frames within the same segment.

Xperience. We evaluate the four complete videos described in Appendix C.1 at their original frame rate of 20 fps. We compare FS=$\{1\}$ with FS=$\{1, 5\}$, using 16-frame windows, WS=12, and 100 camera-PGO iterations for both settings. HandFlow processes chunks of at most 1,024 frames with 32-frame overlap. We fit a rigid alignment to the overlapping camera centres to join adjacent chunks into a complete trajectory. ATE fits one alignment per complete video and pools squared position errors across videos before taking the square root. The relative-motion comparison uses temporal intervals from 1 to 220 s and pools squared errors over all valid frame pairs within videos.

Additional OakInk2 results. The 50-sequence OakInk2 subset described in Appendix C.1 yields 181 continuous segments with valid right-hand annotations and detections, containing 62,750 frames. Table 13 reports camera and right-hand world-space comparisons on this population using the same aggregation as the HOT3D comparison above.

Table 13: **World-space camera and hand reconstruction on OakInk2.** Full-segment camera and right-hand results on the fixed 50-sequence sample. Errors are in millimetres. HandFlow uses ViPE-SLAM without dynamic masks. †HandFlow lacks verified metric translation scale and is excluded from ATE rankings.

Method	ATE-$\mathbb{SE}(3)$	W-MPJPE	WA-MPJPE
EgoH4RT	**18.309**	**94.652**	**23.828**
HaWoR	<u>19.278</u>	<u>101.242</u>	<u>29.853</u>
HandFlow†	1420.469	195.306	47.855

C.3 RECONSTRUCTION UNDER MISSING DETECTIONS

Evaluation clips. We evaluate both hands on 219 HOT3D clips from the timeline used for the MINT evaluation. Each clip contains 81 frames, giving 17,739 frames in total. The clip length accommodates ACE-Ego-Hand's native inference unit. All methods receive the same lossless MP4 clips and camera intrinsics while retaining their native detection and preprocessing pipelines.

Prediction matching. A ground-truth hand is included if at least one of its 21 joints lies inside the image with depth greater than 0.01 m. We match predictions to ground truth by hand side and require box IoU greater than 0.1, using the projected predicted mesh and a ground-truth box enlarged by 10%. Matched predictions are true positives (TP); unmatched ground-truth hands are false negatives (FN), and unmatched predictions are false positives (FP). The pose metrics use all 29,087 valid ground-truth hand instances. Frame accuracy (FAcc) is the fraction of the 17,217 frames containing a visible ground-truth hand with neither false positives nor false negatives.

Threshold sweep. We use YOLO confidence 0.2 and vary the maximum recoverable detection-gap length e over $\{0, 1, 2, 3, 4, 5, 8, 40, 80\}$. For each hand, a consecutive run of missing detections is eligible for prediction only if its entire length is at most e; longer runs are excluded in full. A single valid detection separates two missing runs. Within an eligible gap, learned missing tokens replace unavailable WiLoR tokens, and the resulting predictions must satisfy the same matching criterion. Table 14 reports detection counts and the complete metric sweep. The longer-gap settings recover more hands but also produce more false positives.

Table 14: **Reconstruction under missing detections on HOT3D.** Detection counts and complete threshold sweep on 219 clips. Pose errors are in millimetres. †Three HaWoR clips reproduce the released masked DROID-SLAM empty-graph failure after one retry and are retained as no output.

Method	TP↑	FN↓	FP↓	FAcc↑	Precision↑	Recall↑	F1↑	MPJPE-p↓	PA-MPJPE-p↓
WiLoR	26,150	2,937	1,167	0.816	0.957	0.899	0.927	29.496	20.378
HaWoR†	23,659	5,428	6,135	0.692	0.794	0.813	0.804	42.009	31.377
MINT	27,873	1,214	1,256	0.909	0.957	0.958	0.958	24.945	13.098
ACE-Ego-Hand	27,713	1,374	**630**	0.914	**0.978**	0.953	**0.965**	24.541	12.740
EgoH4RT ($e = 0$)	27,215	1,872	1,051	0.875	0.963	0.936	0.949	17.732	12.882
EgoH4RT ($e = 1$)	27,352	1,735	1,075	0.882	0.962	0.940	0.951	17.151	12.262
EgoH4RT ($e = 2$)	27,417	1,670	1,093	0.885	0.962	0.943	0.952	16.887	11.979
EgoH4RT ($e = 3$)	27,484	1,603	1,107	0.889	0.961	0.945	0.953	16.619	11.675
EgoH4RT ($e = 4$)	27,559	1,528	1,120	0.892	0.961	0.947	0.954	16.316	11.351
EgoH4RT ($e = 5$)	27,597	1,490	1,148	0.894	0.960	0.949	0.954	16.168	11.183
EgoH4RT ($e = 8$)	27,732	1,355	1,179	0.900	0.959	0.953	0.956	15.643	10.592
EgoH4RT ($e = 40$)	28,429	658	1,429	**0.933**	0.952	0.977	0.965	13.034	7.368
EgoH4RT ($e = 80$)	**28,484**	**603**	1,771	0.925	0.941	**0.979**	0.960	**12.845**	**7.149**

C.4 End-to-end efficiency

Inputs and outputs. Figure 5 uses the first 128, 512, 1,024, and 1,802 frames of the same video. EgoH4RT, HaWoR, HandFlow, and MINT output camera trajectories and both hands. ACE-Ego-Hand outputs both hands only; its released causal-VAE implementation processes 125, 509, 1,021, and 1,801 frames for these inputs.

Runtime and memory measurement. Each method runs on a single H100 80GB GPU, with input videos and model files stored on node-local SSD. We measure elapsed time from launching the raw-MP4 inference pipeline to obtaining final predictions, including data loading and hand detection where required. After one warm-up run, we average the times of two measured runs. We sample GPU memory usage every 0.2 s over the same interval and report the highest whole-device reading.

Teaser timing. The inference time shown in Figure 1 measures only the forward inference of our model, using precomputed YOLO detections and inputs already loaded into memory. It excludes YOLO preprocessing and disk reads, and therefore uses a different timing scope from the end-to-end measurements in Figure 5.

C.5 Analysis and ablation

C.5.1 Effect of long-span windows

We remove the long-span windows from the HOT3D configuration in Appendix C.2.2, changing FS from $\{1, k\}$ to $\{1\}$ while keeping WS=13. Both settings use the same evaluation segments and cached network predictions, with identical camera and hand optimization settings. Table 15 supplements Table 7 with the complete camera, world-space hand, and acceleration results.

Table 15: Effect of long-span windows on all 520 HOT3D segments. Both configurations use WS=13 and identical camera and hand optimization. All metrics are lower-is-better.

FS	ATE-$\mathbb{SE}(3)$ (mm)	ATE-$\mathrm{Sim}(3)$ (mm)	RPE-t@1 (mm)	RPE-r@1 (°)	W-MPJPE (mm)	WA-MPJPE (mm)	Accel. (mm/frame^2)
$\{1\}$	7.707	6.047	**0.703**	**0.090**	46.285	12.702	**6.358**
$\{1, k\}$	**5.316**	**3.336**	0.810	0.096	**44.465**	**11.607**	6.646

C.5.2 SENSITIVITY TO FRAME AND WINDOW STRIDES

HOT3D sampling sensitivity. From the 520 HOT3D segments used in Table 4, we retain a fixed common population of 103 segments with $N \geq 226$, so that every configuration can use frame stride 15. All configurations use 100 camera-PGO iterations; camera-space hand predictions are held fixed and transformed by the optimized cameras, without subsequent MANO refinement. The WA-MPJPE values therefore precede the second-stage hand refinement. In Figure 6, dotted lines mark the HOT3D full-sequence settings on this population, where the adaptive frame-stride set is $\{1, 15\}$ and WS=13. For 16-frame windows, WS=13 preserves three-frame overlap and uses $12.2\times$ fewer network windows than WS=1, while the WS=1–15 sweep shows little metric variation. At WS=13, changing FS from $\{1\}$ to $\{1, 15\}$ reduces ATE-$\mathbb{SE}(3)$ from 8.767 to 5.808 mm and WA-MPJPE from 13.235 to 11.667 mm.

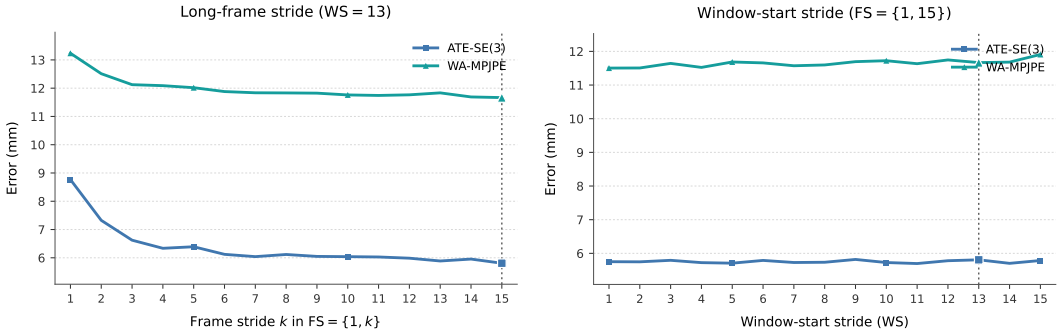


Figure 6: **PGO sampling sensitivity.** Left: long frame stride k in FS=$\{1, k\}$ at WS=13. Right: WS at FS=$\{1, 15\}$. ATE-$\mathbb{SE}(3)$ and WA-MPJPE share a y-axis in millimeters; lower is better.

Xperience stride diagnostic at WS=13. A separate diagnostic evaluates FS=$\{1\}$, $\{1, 10\}$, and $\{1, 15\}$ at WS=13 on the same four Xperience recordings, yielding ATE-$\mathbb{SE}(3)$/ATE-$\mathrm{Sim}(3)$ of $478.596/426.989$, $460.782/439.723$, and $1076.446/1045.318$ mm, respectively. Both longer-stride variants worsen multi-horizon RPE relative to FS=$\{1\}$, with FS=$\{1, 15\}$ showing strong instability. This configuration differs from the main FS=$\{1, 5\}$, WS=12 experiment, whose smaller long stride accommodates Xperience's 20 fps frame rate and larger inter-frame motion.

D QUALITATIVE RESULTS

We show screenshots from four demo videos: one HOT3D sequence, two Xperience sequences, and one in-the-wild video. The examples illustrate hand overlays and camera trajectories; timestamps refer to the displayed clips.

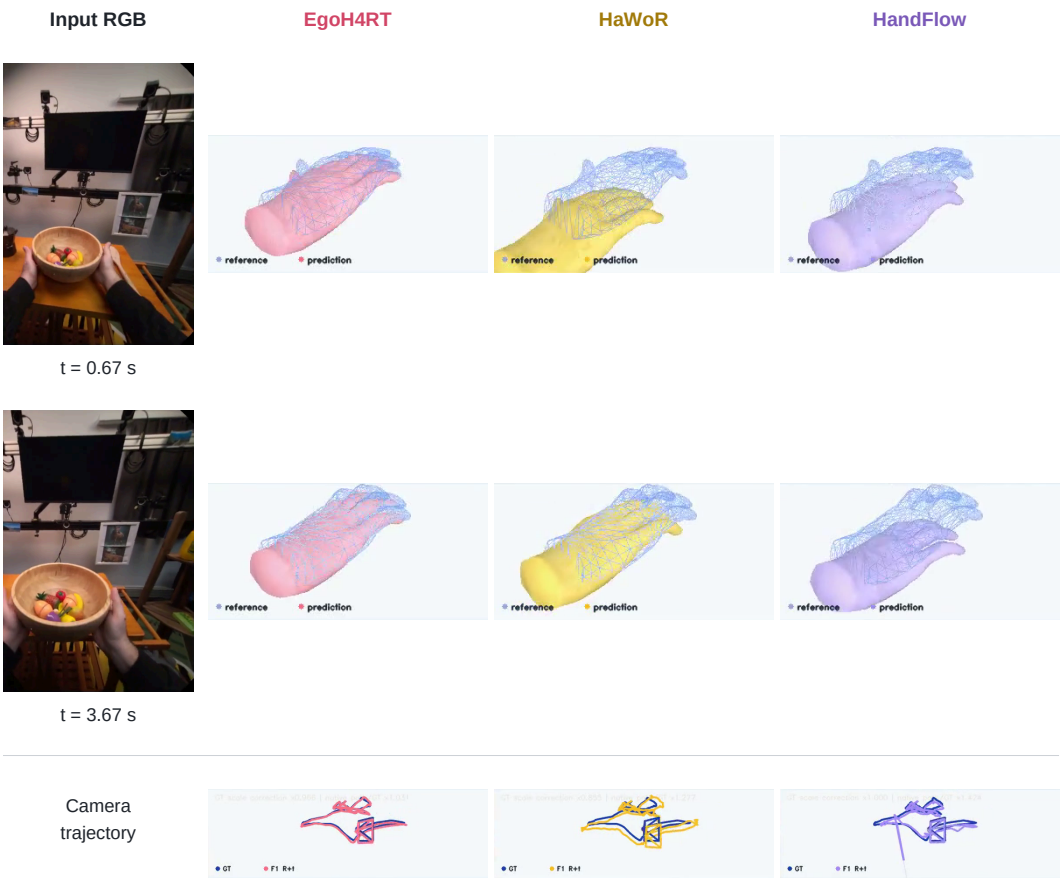


Figure 7: **Hand and camera reconstruction on HOT3D.** Two moments from a sequence involving a bowl. The leftmost column shows input frames; the remaining columns compare methods at the same two moments. Hand meshes are rendered against a light background under a shared view, with the reference mesh in blue wireframe. The bottom row shows camera trajectories at the end of the clip: blue denotes the reference, and each method uses its labelled color. Camera trajectories are rigidly aligned at the first frame without scale fitting.

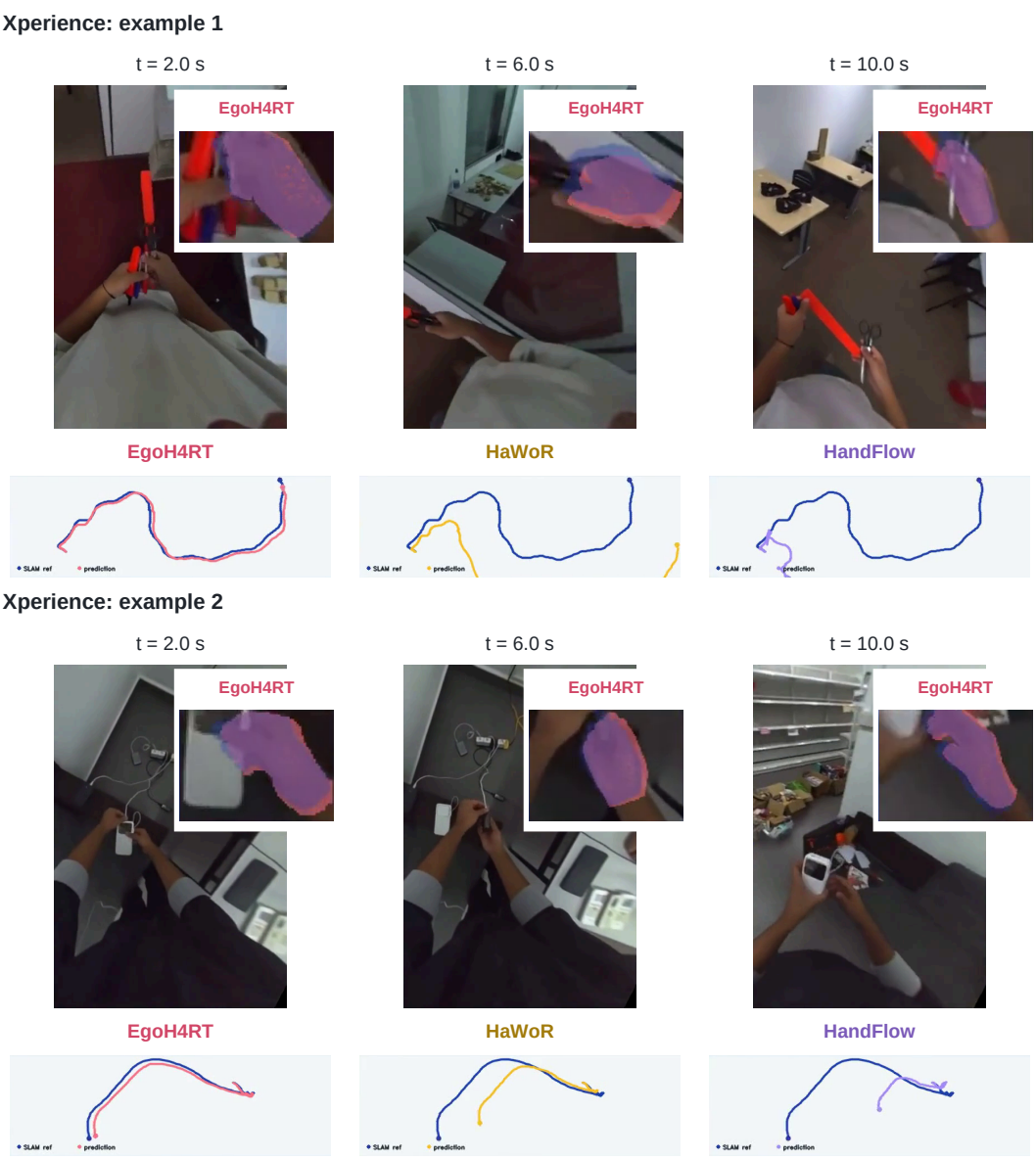


Figure 8: **Camera trajectories on Xperience.** Each example shows three input frames, followed by the camera trajectories at the end of the clip. Insets show EgoH4RT right-hand mesh overlays. Blue denotes the released metric-SLAM reference; the other colors denote predictions. All trajectories are rigidly aligned at the first frame without scale fitting and use the same view within each example.

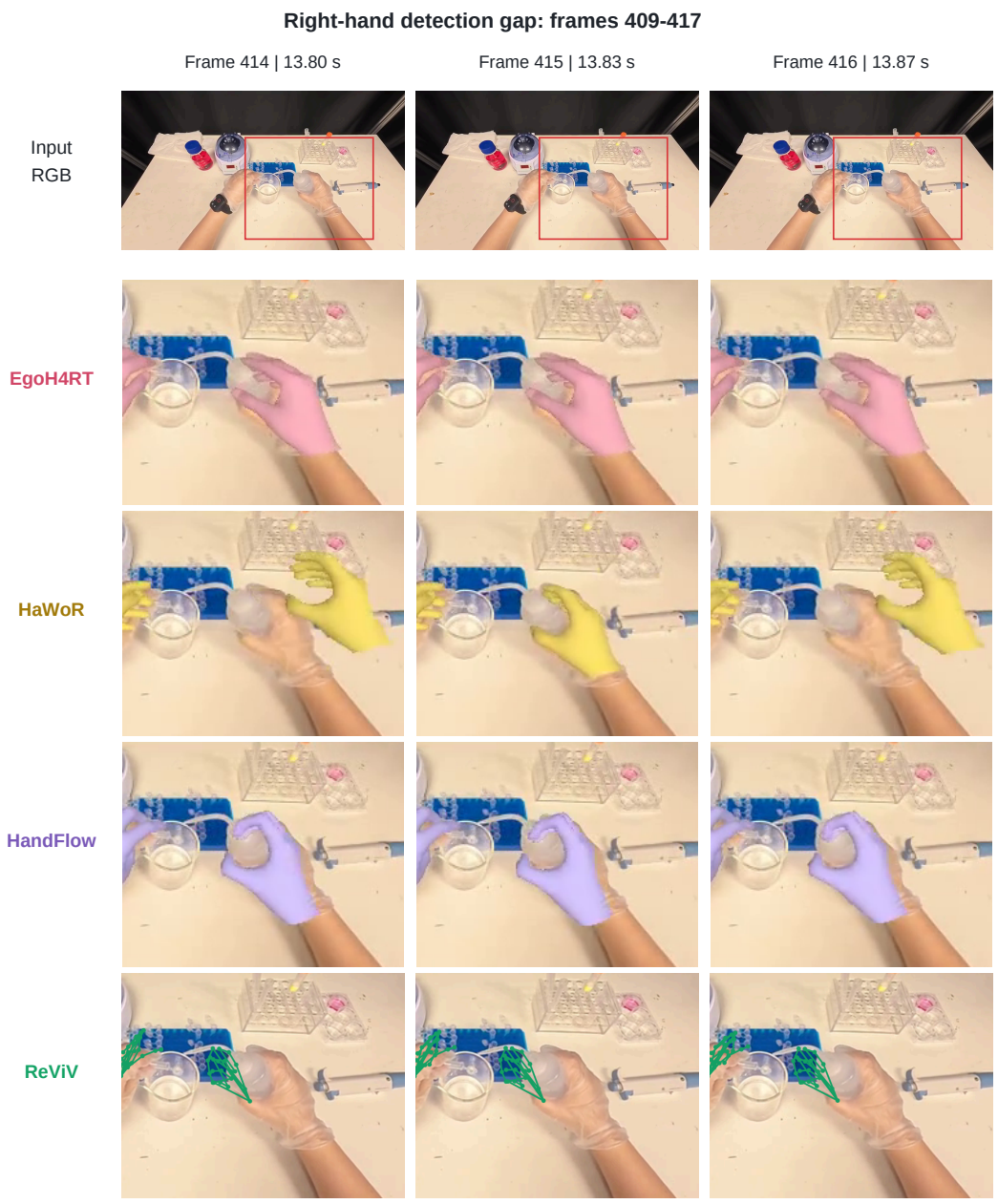


Figure 9: **Hand reconstruction during a detection gap.** Three consecutive frames (414–416) from a nine-frame right-hand YOLO detection gap in an in-the-wild video. The red boxes mark the region enlarged for every method. EgoH4RT, HaWoR, and HandFlow are shown as mesh overlays; ReViV is shown as projected joints. The EgoH4RT demo renderer applies three-frame temporal smoothing to hand parameters. No reference hand annotations are available.