G4D-HAND: GEOMETRY-GROUNDED 4D HAND TRAJECTORY ESTIMATION FROM EGOCENTRIC VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Egocentric 4D hand trajectory estimation aims to recover the 3D positions of hand keypoints over time in a consistent coordinate frame from egocentric video. Most existing methods follow an articulation-centric pipeline in which hands are detected and cropped, reconstructed from independent image patches, assigned wrist translation through a weak-perspective camera, and subsequently placed in the world by an external SLAM system. At no stage do these pipelines represent the hands in relation to the scene in which they move, even though this could help resolve hand depth and occlusion. We argue that scene geometry should shape the hand representation from the outset and introduce G4D-HAND, which grounds hand prediction in a geometric representation. It reconstructs both hands from shared full-frame features, without a hand detector or hand-specific image encoder, jointly recovering articulation and metric wrist translation in a common camera frame. Inter-hand and temporal supervision further encourage consistent relative positioning and coherent motion, and the same backbone estimates camera motion to lift predictions into the world frame without a separate SLAM stage. G4D-HAND runs at 30+ FPS for camera-frame, matches state-of-the-art articulation accuracy, and substantially improves localization on egocentric benchmarks. On ARCTIC, a dataset with frequent hand occlusions, it reduces inter-hand relative wrist error by 50% and camera-frame hand trajectory error by 31%, against the strongest baseline on each metric. Code and checkpoints will be made public.

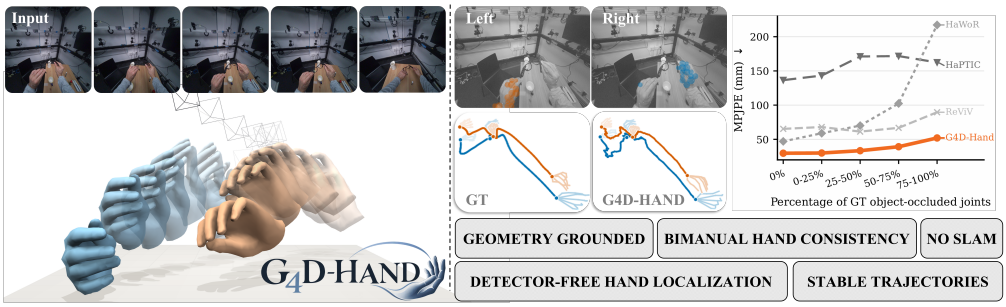

Figure 1: **G4D-HAND reconstructs 4D hand trajectories from monocular egocentric video.** From a shared scene reconstruction backbone, G4D-HAND estimates hand and camera motion without a hand detector, hand encoder, or external SLAM. **Left:** Reconstructed hand motion and camera trajectory. **Middle top:** Gradient attribution maps of the hand-specific feature extractors show that each hand is well localized within the scene features. **Middle bottom:** Predicted bimanual trajectories preserve relative hand positioning and closely follow the ground truth. **Right:** Hand reconstruction error remains low as object occlusion increases.

1 INTRODUCTION

Egocentric videos capture how a person interacts with the world from their own perspective. Understanding these interactions in 3D includes estimating how the hands are posed, where they are located, and how they move through the surrounding scene. In this work, we focus on monocular egocentric 4D hand trajectory estimation, i.e., recovering the 3D positions of hand keypoints over time in a consistent coordinate frame. Doing this well requires accurate hand articulation, precise wrist localization, and coherent motion relative to the moving camera and within the 3D scene.

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.

This task has become increasingly important with the development and adoption of wearable cameras in recent years. Their videos offer a close view of how people manipulate objects, but understanding these interactions automatically in 3D requires recovering structured data from the images. Hand trajectories provide a compact geometric description of human manipulation and can support reasoning about actions in a video (Kwon et al., 2021; Shamil et al., 2024), anchor augmented-reality content to physical interactions (Han et al., 2022; Banerjee et al., 2025), and supervise robots from human demonstrations (Kareer et al., 2025; Li et al., 2026).

Extracting such trajectories from monocular egocentric video, however, remains challenging. A single image does not uniquely determine metric hand depth since different hand sizes and distances from the camera can produce similar projections. Frequent hand-object occlusions and viewpoint changes further limit the visual evidence available for reconstruction. Moreover, recovering trajectories in a consistent coordinate frame requires separating hand motion from the motion of the recording camera. Together, these challenges motivate reasoning beyond the hands themselves by using the surrounding 3D scene to provide additional geometric and temporal context.

Limitations of previous approaches. Existing 4D hand-reconstruction pipelines largely model the hands and the scene as separate problems. Large-scale transformers have advanced image-based hand pose and shape estimation (Pavlakos et al., 2024; Potamias et al., 2025), and recent methods extend these articulation-centric representations to recover 4D trajectories (Zhang et al., 2025a; Yu et al., 2025; Ye et al., 2026). Pipelines such as HaWoR (Zhang et al., 2025a) and Dyn-HaMR (Yu et al., 2025) detect and crop each hand independently, often excluding much of the surrounding scene and the other hand from the initial input of hand reconstruction models. Wrist translation is initialized from a predicted weak-perspective camera, making metric depth sensitive to scale errors even when camera intrinsics are known (Ye et al., 2026). To recover world-frame trajectories, these methods further rely on a separate SLAM system, but do not use its scene representation to condition hand reconstruction. This separation leaves potentially useful geometric context unused, even though hands move within and interact with a 3D scene. In this work, we hypothesize that representations encoding scene geometry provide a strong foundation for recovering 4D hand trajectories as an alternative to hand-centric image representations.

Our work: G4D-HAND. We introduce G4D-HAND, a geometry-grounded method for 4D hand trajectory estimation. Figure 1 shows reconstructed hand and camera trajectories from a monocular egocentric video. Building on a frozen 4D reconstruction backbone, the method extracts hand-specific features from full-frame image tokens, without hand detectors or dedicated hand encoders, and directly regresses metric camera-frame wrist translations from geometric feature pyramids. Inter-hand and temporal supervision encourage consistent relative positioning and coherent motion. The same backbone estimates camera motion to lift the hands into a shared world frame without external SLAM. Across egocentric benchmarks, G4D-HAND substantially improves trajectory accuracy, hand localization, and relative hand positioning, while matching state-of-the-art articulation accuracy.

2 RELATED WORK

3D hand pose estimation. Single-image methods regress MANO (Romero et al., 2017) pose, shape, and translation from a hand crop. HaMeR (Pavlakos et al., 2024) uses a fully transformer-based architecture and WiLoR (Potamias et al., 2025) scales up training data and adds a refinement module for hand-image alignment. WildHands (Prakash et al., 2024) encodes camera intrinsics and crop location to handle egocentric perspective distortion, and EgoForce (Millerdurai et al., 2026) recovers metric wrist position conditioned on forearm geometry and intrinsics. These models provide strong articulation priors, but rely on per-frame prediction, leading to temporally inconsistent motion.

4D hand trajectory estimation. Recent methods extend image-based reconstruction to temporally coherent camera- or world-frame trajectories (Zhang et al., 2025a; Yu et al., 2025; Ye et al., 2026), as in moving-camera body reconstruction (Ye et al., 2023; Shin et al., 2024; Wang et al., 2024). The pioneering works HaWoR (Zhang et al., 2025a) and Dyn-HaMR (Yu et al., 2025) pair camera-frame hand models (Pavlakos et al., 2024; Potamias et al., 2025) with SLAM-estimated camera trajectories. HaWoR adds temporal attention and a motion infiller for occluded or out-of-view hands; Dyn-HaMR handles interacting hands via costly optimization against 2D observations. Both output world-frame trajectories, so their camera-frame predictions inherit camera estimation errors through projection. HaPTIC (Ye et al., 2026) instead predicts in the camera frame, estimating depth relative to the

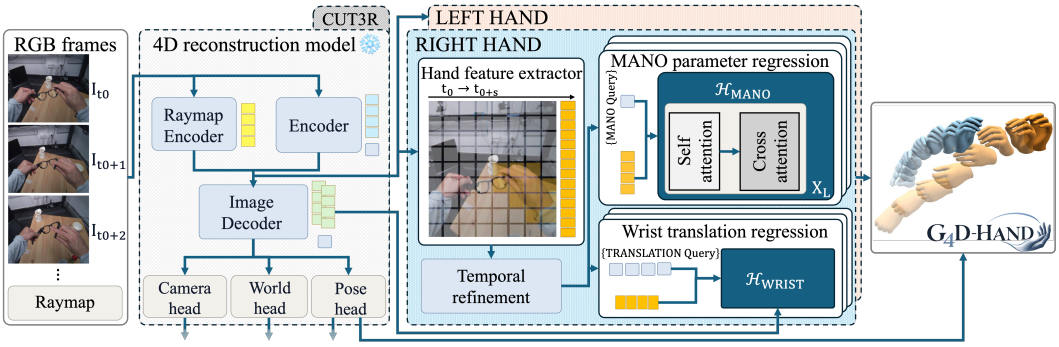

Figure 2: **Overview of G4D-HAND.** Given an egocentric video, the frozen CUT3R backbone extracts shared full-frame scene representations conditioned on camera intrinsics through raymap encoding. Separate left- and right-hand branches extract hand-specific tokens, which are temporally refined and used to predict MANO articulation and orientation. Metric wrist translation is predicted by cross-attending the hand representation to CUT3R's geometric feature pyramid. The frozen backbone estimates camera motion, so reconstructed hands are lifted from the camera to world frame.

first frame, and ReViV (Lyu et al., 2026) predicts camera-frame hands within a masked generative transformer for multimodal egocentric reconstruction. G4D-HAND predicts hand pose and metric hand location from scene geometry features, aiming for accurate localization and trajectory.

Repurposing geometry foundation models. MonST3R (Zhang et al., 2025b), CUT3R (Wang et al., 2025), and VGGT-Ω (Wang et al., 2026) recover scene geometry and camera motion from dynamic videos. CUT3R's persistent recurrent state integrates observations over time to predict metric-scale pointmaps and camera poses in a shared frame. Their priors have been adapted beyond reconstruction: GAM (Han et al., 2026) repurposes a geometric foundation model as a shared backbone for robot perception, future prediction, and action decoding. Closest to our work, Human3R (Chen et al., 2026) augments CUT3R tokens with Multi-HMR (Baradel et al., 2024) features to inject human-specific priors for online 4D human-scene reconstruction. In contrast, we investigate whether scene geometry itself can support hand pose prediction and metric wrist localization from egocentric video, without a hand-specific image encoder.

3 Our method: G4D-Hand

3.1 Preliminaries

Parametric hand representation: MANO. In our work, a human hand is represented in 3D using MANO (Romero et al., 2017), a low-dimensional differentiable model of hand shape and articulation. The method predicts MANO pose $\boldsymbol{\Theta} = [\boldsymbol{\Theta}_0, \ldots, \boldsymbol{\Theta}_{15}] \in \mathbb{R}^{16\times3}$, consisting of the axis-angle root rotation $\boldsymbol{\Theta}_0$ and the relative rotations $\boldsymbol{\Theta}_{1:15}$ of 15 finger joints, shape parameters ($\boldsymbol{\beta} \in \mathbb{R}^{10}$), and metric camera-frame hand translation ($\boldsymbol{\tau} \in \mathbb{R}^3$). Given these, the MANO function outputs a hand mesh with 778 vertices and 16 skeletal joints ($\mathbf{J} \in \mathbb{R}^{16\times3}$), representing the hand with three joints per finger, in the camera frame.

4D reconstruction foundation model: CUT3R. CUT3R (Wang et al., 2025) is a feed-forward recurrent model that jointly estimates metric-scale scene geometry and camera motion from unposed image sequences. Given an image I_t at frame t, a ViT (Dosovitskiy et al., 2021) encoder $\mathcal{E}$ extracts image tokens $\mathbf{F}_t$, which interact with a persistent scene state from the previous frame ($\mathbf{S}_{t-1}$) through two interconnected transformer decoders, jointly denoted by $\mathcal{D}$ as:

$$\mathbf{F}_t = \mathcal{E}(\mathrm{I}_t), \qquad [\mathbf{F}'_t, \mathbf{z}'_t], \mathbf{S}_t = \mathcal{D}\left([\mathbf{F}_t, \mathbf{z}], \mathbf{S}_{t-1}\right).$$

Here, $\mathbf{F}'_t$ represents the final decoder features and $\mathbf{z}$ is a learnable pose token whose output $\mathbf{z}'_t$ captures scene-level information, including camera motion. For dense reconstruction, DPT heads (Ranftl et al., 2021) take the feature pyramid $\mathcal{P}_t$ formed from intermediate and final image decoder features. From the output camera token and pyramid features, CUT3R predicts pointmaps in camera and world frame and camera-to-world transformation. These dense pointmaps include the hands in egocentric videos, therefore our motivation is to use the implicit geometric information in CUT3R features to reconstruct the hands in 3D.

CUT3R also supports novel-view reconstruction by querying its persistent state with a virtual camera, represented by a six-channel raymap of per-pixel ray origins and directions. A lightweight encoder $\mathcal{E}_{\text{ray}}$ maps the raymap into image-token space, from which the decoder and heads predict pointmaps without updating the state. To keep hand predictions for both hands and across frames in a consistent camera-local coordinate system in G4D-HAND, we repurpose CUT3R's raymap encoder. For each observed image, we compute a raymap from its intrinsics and a fixed reference pose, then add the encoded raymap to the input image tokens to provide geometric context for estimating hand pose and absolute translation.

3.2 G4D-HAND

G4D-HAND uses 4D scene geometry as the foundation for egocentric hand reconstruction, recovering both hand articulation and metric localization from a frozen CUT3R representation. The overview of the architecture is shown in Figure 2.

Scene representations. Each image I_t is encoded once into image tokens $\mathbf{F}_t$. To condition these features on camera intrinsics for 3D localization, the encoding of a raymap $\mathbf{r}$ is added as:

$$\widetilde{\mathbf{F}}_t = \mathbf{F}_t + \mathcal{E}_{\text{ray}}(\mathbf{r}).$$

The raymap is constructed from the camera intrinsics using a fixed reference pose across frames, so its rays are always expressed in local camera coordinates. The conditioned tokens $\widetilde{\mathbf{F}}_t$ are used by the hand-specific articulation branches and passed through CUT3R's recurrent decoder to obtain the feature pyramid $\widetilde{\mathcal{P}}_t$ for hand localization. Thus, hand articulation is estimated from encoder tokens that provide semantic and geometric information, while hand localization is conditioned on the pyramid features originally used by CUT3R for dense scene reconstruction. Camera motion is estimated in a second decoder pass without raymap conditioning, using an independent recurrent state. This pass is needed to lift hand trajectories into the world frame: with the raymap fixed to the reference pose, the conditioned decoder predicts static camera poses. The encoder features can be reused, so the most computationally expensive part is computed only once.

Hand-specific feature extraction. To represent both hands within the full scene, G4D-HAND extracts hand-specific tokens directly from CUT3R's raymap-conditioned encoder features $\widetilde{\mathbf{F}}_t$, without detecting or cropping the hands. Inspired by Millerdurai et al. (2024), dedicated left- and right-hand branches with independent parameters map the shared scene features to fixed-size token sets. These are then used for MANO parameter prediction and hand-specific conditioning for localization within the scene. Each branch combines coarse and fine learned-query cross-attention pooling. The coarse path uses a compact query set to aggregate broader spatial context, while the fine path uses one query per output token to capture local articulation details. Both paths aggregate image features according to learned attention weights, and their outputs are combined to form hand-specific features:

$$\widehat{\mathbf{H}}_t^h = \mathrm{LNorm}\left((1-\sigma(g^h))\,\mathbf{W}_m^h\,\mathrm{CAttn}\left(\mathbf{Q}_{\mathrm{c}}^h, \Pi^h\left(\widetilde{\mathbf{F}}_t\right)\right) + \sigma(g^h)\,\mathrm{CAttn}\left(\mathbf{Q}_{\mathrm{f}}^h, \Pi^h\left(\widetilde{\mathbf{F}}_t\right)\right)\right),$$

for $h \in \{\mathrm{L}, \mathrm{R}\}$. Here, Π^h and CAttn denote the projection of the scene features into the hand-feature space and learned-query cross-attention pooling over the projected scene tokens, respectively. $\mathbf{Q}_{\mathrm{c}}^h$ and $\mathbf{Q}_{\mathrm{f}}^h$ denote the coarse and fine-grained learned queries, $\mathbf{W}_m^h$ maps the compact coarse representation to the output token set, g^h is a trainable scalar gate, and σ the sigmoid function. The two branches have independent parameters, allowing each to specialize in the corresponding hand while retaining access to the same full-scene context. Gradient-attribution visualizations (Figure 1, Appendix E.1) show that the most influential tokens are the ones representing the respective hand within the scene.

Temporal refinement. Training and inference operate on sequences of S frames $\{\mathrm{I}_{t_0}, \mathrm{I}_{t_0+s}, \ldots, \mathrm{I}_{t_0+s(S-1)}\}$ sampled at a fixed temporal stride s. Within a sequence, the per-frame hand-specific representations $\widehat{\mathbf{H}}_t^h$ are refined by a temporal transformer $\mathcal{T}$, with shared weights between hands, that applies attention along the temporal axis as follows:

$$\mathbf{H}^h_{t\in\{t_0,\ldots,t_0+s(S-1)\}} = \mathcal{T}\big(\widehat{\mathbf{H}}^h_{t\in\{t_0,\ldots,t_0+s(S-1)\}}\big), \qquad h \in \{\mathrm{L}, \mathrm{R}\},$$

to obtain final hand representation tokens ($\mathbf{H}_t$). Attending across the sequence propagates information between frames, producing more coherent predictions when individual frames offer limited visual evidence, e.g., under motion blur or occlusion, and reducing jitter in trajectories and poses.

MANO parameter regression. G4D-HAND adopts HaMeR's pretrained transformer decoder $\mathcal{H}_{\text{MANO}}$ (Pavlakos et al., 2024), whose learnable query $\mathbf{q}_{\text{MANO}}$ cross-attends to hand-specific features $\mathbf{H}_t^h$ to regress pose and shape as:

$$\left(\boldsymbol{\Theta}_t^h, \boldsymbol{\beta}_t^h\right) = \mathcal{H}_{\text{MANO}}\left(\mathbf{q}_{\text{MANO}}, \mathbf{H}_t^h\right), \qquad h \in \{\text{L}, \text{R}\}.$$

To adapt the head from cropped-hand encoder features to geometric features while retaining its articulation prior, we re-initialize and train its cross-attention layers and pre-feed-forward normalization layers, keeping all other weights frozen (details in Appendix A). The weak-perspective camera prediction from $\mathcal{H}_{\text{MANO}}$ is discarded. Instead, absolute camera-frame hand translation $\boldsymbol{\tau}_t^h$ is regressed using the dedicated head and scene features. Head weights are shared between hand branches.

Translation regression. Metric camera-frame hand translation is regressed through cross-attention between learned queries conditioned on hand tokens $\mathbf{H}_t^h$ and CUT3R's feature pyramid $\widetilde{\mathcal{P}}_t$:

$$\mathbf{C}_t = \text{Fuse}(\widetilde{\mathcal{P}}_t), \qquad \boldsymbol{\tau}_t^h = \mathcal{H}_{\text{wrist}}\big(\text{CAttn}\big(\text{Query}(\mathbf{H}_t^h), \mathbf{C}_t\big)\big), \qquad h \in \{\text{L}, \text{R}\}.$$

Here, Fuse combines pyramid levels into shared scene context $\mathbf{C}_t$, Query conditions learned queries on the target hand features, and CAttn aggregates the corresponding scene information. The MLP $\mathcal{H}_{\text{wrist}}$ maps the concatenated output tokens to translation $\boldsymbol{\tau}_t^h \in \mathbb{R}^3$. Using the same pyramid levels as CUT3R's DPT pointmap head allows translation regression to rely on features trained to output metric scene pointmaps. All parameters in this component are shared between hands.

Lifting camera-frame trajectories to the world frame. Camera trajectory is estimated through a separate CUT3R decoder pass that reuses $\mathbf{F}_t$ without raymap conditioning and maintains an independent state. This pass outputs dense scene pointmaps and camera-to-world transformation.

Loss functions. Training combines per-hand reconstruction losses with inter-hand and temporal supervision. The per-hand terms supervise 3D joints, projected 2D joints, articulated joint rotations, global orientation, and metric hand translation. Rotation errors are measured by geodesic distance on $\text{SO}(3)$. Inter-hand terms supervise relative joint displacement vectors, relative hand translation, and consistency between the predicted left- and right-hand shapes, linking the two hand predictions during training. Temporal supervision constrains changes in articulation over time while retaining the ground-truth dynamics. The overall loss is computed as:

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\sum_{h\in\{\text{L},\text{R}\}} \Big(\lambda_{\text{3D}}\mathcal{L}_{\text{3D}}^h + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}}^h + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}}^h + \lambda_{\text{orient}}\mathcal{L}_{\text{orient}}^h + \lambda_{\text{transl}}\mathcal{L}_{\text{transl}}^h + \lambda_{\text{accel}}\mathcal{L}_{\text{accel}}^h\Big) \\ &+ \lambda_{\text{3D}}^{\text{ih}}\mathcal{L}_{\text{3D}}^{\text{ih}} + \lambda_{\text{transl}}^{\text{ih}}\mathcal{L}_{\text{transl}}^{\text{ih}} + \lambda_{\text{shape}}^{\text{ih}}\mathcal{L}_{\text{shape}}^{\text{ih}},\end{aligned}$$

with full loss definitions provided in Appendix B.4.

Training setup and implementation details. The method is trained jointly on three egocentric video datasets: HOT3D (Banerjee et al., 2025), ARCTIC (Fan et al., 2023), and H2O (Kwon et al., 2021), using camera-frame 3D joint, 2D joint, and MANO ground-truth annotations. All evaluation subjects are excluded from the training set. Each training batch contains multiple sequences of a fixed length $S = 16$. Training includes temporal stride augmentation, with a constant stride within each sequence, and random crop zoom and position augmentation with corresponding updates to the camera intrinsics. Appearance augmentations applied on a per-sequence basis include color jitter, Gaussian blur, Gaussian noise, and random erasing. All CUT3R components remain frozen to preserve the pretrained scene representation and camera-motion estimation capabilities. In the MANO head initialized from HaMeR, cross-attention to the hand features and pre-feed-forward layer normalizations are reinitialized and trained to adapt to the new feature representation, while all remaining layers are frozen. The model contains 27 million trainable parameters and 818 million frozen parameters. Further implementation details are provided in Appendix B.

4 EXPERIMENTS

Datasets. We evaluate on three egocentric datasets: ARCTIC (Fan et al., 2023), H2O (Kwon et al., 2021), and HOT3D (Banerjee et al., 2025). For ARCTIC and H2O, we follow the official evaluation splits (validation and test, respectively). Since the HOT3D test set is not publicly available, we instead evaluate on sequences from three held-out subjects that do not appear in our training set.

G4D-HAND inference details. Inference for G4D-HAND uses non-overlapping sequences of $S = 16$ frames with temporal stride $s = 1$ and fixed camera intrinsics. Running the model on a sequence of $T = 128$ frames therefore requires 8 non-overlapping windows of $S = 16$ frames. CUT3R's recurrent state is reset every 16 frames for camera-frame evaluation and every 64 frames for camera trajectory estimation, which is the maximum sequence length used in CUT3R training. G4D-HAND processes 16-frame clips jointly and achieves 39 FPS with batched clip processing and 31 FPS with sequential processing on an RTX 4090 for camera-frame predictions.

Baseline inference details. We compare against single-frame methods (HaMeR (Pavlakos et al., 2024), WiLoR (Potamias et al., 2025), EgoForce (Millerdurai et al., 2026)), and temporal methods targeting camera-frame (HaPTIC (Ye et al., 2026), ReViV (Lyu et al., 2026)) and world-frame hand trajectories (Dyn-HaMR (Yu et al., 2025), HaWoR (Zhang et al., 2025a)). For fair comparison, all methods, except ReViV that predicts metric location directly, use ground truth camera intrinsics.

Evaluation metrics. For camera-frame evaluation, we report Mean Per Joint Position Error (RA-MPJPE) after root-alignment to measure 3D hand pose accuracy, wrist 3D error (Wrist err.) to measure absolute hand localization, and Mean Relative-Root Position Error (MRRPE) to measure the relative root translation between the hands. To evaluate the local consistency of predicted and ground truth dynamics, we use acceleration error (Acc.). To evaluate hand trajectory accuracy in camera and world coordinates, we report G-MPJPE and GA-MPJPE. Following Dyn-HaMR, both metrics align the predicted trajectory to the ground truth using the Umeyama method (Umeyama, 1991): G-MPJPE estimates the alignment from the first two frames, while GA-MPJPE uses the entire sequence. We evaluate camera-frame trajectories over 128-frame clips and world-frame trajectories over 64-frame clips, the maximum sequence length used to train CUT3R. Evaluation is performed on 16 MANO skeletal joints (except for the ARCTIC test web server). Results on 21 keypoints are provided in Appendix D.2, with method rankings similar to those for the 16-keypoint evaluation. Metric definitions are provided in Appendix C.2.

4.1 CAMERA-FRAME EVALUATION

ARCTIC is a dataset with frequent hand-object interactions and hand occlusions. The validation-set results are shown in the first eight rows of Table 1. G4D-HAND predicts stable trajectories through occlusions, reducing G-MPJPE by 31% relative to the best baseline, HaPTIC. Furthermore, G4D-HAND errors remain low as hand occlusion increases (Figure 1, Appendix D.3). For localization, G4D-HAND reduces wrist error by 55% compared to ReViV, which learns metric translation during training. Although ReViV ranks second in wrist error and MRRPE, its trajectory error is higher than most temporal methods, possibly because of high RA-MPJPE. G4D-HAND instead achieves strong localization while maintaining competitive articulation accuracy. Frame-based methods (HaMeR, WiLoR, and EgoForce) predict each frame independently from a hand crop, so under occlusion they have little visual evidence and no temporal context. Dyn-HaMR and HaWoR build on these models and add temporal modeling through motion infilling, which can produce jittery or overly extended trajectories (see Figure 3). Occasional high wrist errors for HaWoR and Dyn-HaMR could also be due to optimizing world hand trajectories and transforming them back to the camera frame using estimated external SLAM poses. In contrast, G4D-HAND predicts directly in the metric camera frame before lifting trajectories to world coordinates. Results on the ARCTIC test web server are presented in Table 3. G4D-HAND achieves 62% lower MRRPE relative to the second-best and a negligible 2% higher RA-MPJPE. Additional visuals are shown in Appendix E.

H2O is a dataset in which both hands are mostly in view. Results are shown in the middle eight rows of Table 1. Compared with Dyn-HaMR, which jointly optimizes the two hand trajectories using an interacting-hand prior, G4D-HAND achieves $2.3\times$ lower MRRPE. Compared to HaPTIC, G4D-HAND reduces both wrist error and MRRPE by $\sim 65\%$, suggesting that HaPTIC's wrist-depth formulation relative to the first frame limits localization accuracy even under favorable visibility conditions.

HOT3D is a challenging dataset with rapid camera motion and frequent changes in hand visibility. Both hands are visible in 73% of frames, while only one hand is visible in 26%, none in 1%. Results are presented in the bottom seven rows of Table 1. G4D-HAND achieves approximately $2\times$ lower trajectory and $4.4\times$ lower acceleration error than WiLoR. We observe large errors for WiLoR near image boundaries and occasional left–right detection confusion for HaMeR. This also affects methods that extend these models.

Table 1: **Camera-frame comparison on three egocentric benchmarks without scale alignment.** We report trajectory (rigid G-MPJPE, GA-MPJPE), wrist position (Wrist Err.), inter-hand (MRRPE), articulation (RA-MPJPE), and temporal consistency (Acc.) errors. Best and second-best results per dataset are **bold** and underlined, respectively. Lower is better; errors are in mm, except Acc. in m/s^2. G4D-HAND improves trajectory and inter-hand accuracy over all previous works.

		G-MPJPE	GA-MPJPE	Wrist err.	MRRPE	RA-MPJPE	Acc.
ARCTIC	HaMeR	96.9	60.3	88.3	86.3	25.6	7.1
	WiLoR	81.0	53.0	65.9	87.7	17.6	6.7
	EgoForce	98.2	62.8	69.9	90.4	20.4	7.1
	Dyn-HaMR	74.2	38.6	86.5	99.2	25.0	2.2
	HaWoR	87.1	46.2	64.0	82.3	32.3	3.3
	HaPTIC	60.8	31.6	144.4	92.3	**16.9**	3.1
	ReViV	85.6	44.6	62.6	56.8	28.9	3.9
	G4D-HAND	**41.7**	**22.0**	**28.3**	**28.7**	17.2	**2.1**
H2O	HaMeR	35.3	18.5	28.6	38.8	15.8	4.2
	WiLoR	29.9	14.2	21.5	30.6	13.8	3.2
	EgoForce	27.1	16.8	23.8	33.8	13.9	3.7
	Dyn-HaMR	30.8	17.5	41.6	48.8	15.5	1.7
	HaWoR	31.1	19.5	35.4	52.1	17.2	2.7
	HaPTIC	34.0	18.8	57.9	62.0	**13.4**	2.3
	ReViV	41.7	23.1	40.6	50.1	24.6	2.7
	G4D-HAND	**21.4**	**11.2**	**20.9**	**21.5**	13.7	**1.5**
HOT3D	HaMeR	199.1	88.5	100.6	132.1	56.1	11.6
	WiLoR*	103.5	54.1	71.0	94.9	20.1	11.1
	EgoForce*	198.9	91.8	115.8	166.4	37.0	15.4
	Dyn-HaMR	96.6	45.4	127.3	197.1	34.2	**2.1**
	HaPTIC	152.0	74.8	187.1	250.1	51.2	6.5
	ReViV*	121.4	51.8	85.1	92.9	39.2	5.1
	G4D-HAND	**48.6**	**21.0**	**36.7**	**42.3**	**18.7**	2.5

* Possible overlap between their training and our evaluation HOT3D set. We omit HaWoR results on HOT3D as it is its main train set and the subject split is not disclosed in paper.

Table 2: **World-frame hand pose evaluation on three egocentric benchmarks.** We report trajectory errors using G-MPJPE and GA-MPJPE (aligned with scale). Best and second-best results per dataset are **bold** and underlined. Lower is better; all errors are in mm. G4D-HAND achieves the best performance across all metrics and datasets without relying on an external SLAM system.

	ARCTIC		H2O		HOT3D	
	G-MPJPE	GA-MPJPE	G-MPJPE	GA-MPJPE	G-MPJPE	GA-MPJPE
HaMeR & CUT3R	92.3	37.5	29.8	13.8	150.8	53.5
WiLoR & CUT3R*	84.2	36.0	26.6	11.8	106.0	46.1
HaPTIC & CUT3R	88.5	38.6	32.3	15.4	169.7	71.1
EgoForce & CUT3R*	94.6	38.1	24.8	12.8	171.5	62.8
Dyn-HaMR	74.1	32.7	35.4	13.0	137.3	50.1
HaWoR*	71.1	30.9	26.9	15.4	–	–
G4D-HAND	**58.1**	**29.5**	**20.0**	**10.3**	**80.8**	**34.2**

* Possible overlap between their training and our evaluation HOT3D set. We omit HaWoR results on HOT3D as it is its main train set and the subject split is not disclosed in paper.

4.2 WORLD-FRAME EVALUATION

Evaluation with ground-truth camera trajectories. When camera motion is available from, for instance, a tracked headset, absolute camera-frame hand predictions can be directly transformed into world coordinates. We evaluate this setting on ARCTIC with rigid-aligned GA-MPJPE, using ground-truth camera extrinsics, and vary the trajectory alignment window over 64, 128, and all frames (Figure 4, full numbers in Appendix D.4). All baselines degrade as the alignment window grows. G4D-HAND stays more stable across all three settings, and its margin over the best baseline grows from 6.0 mm at 64 frames to 27.7 mm over the full clip. These results support the stability

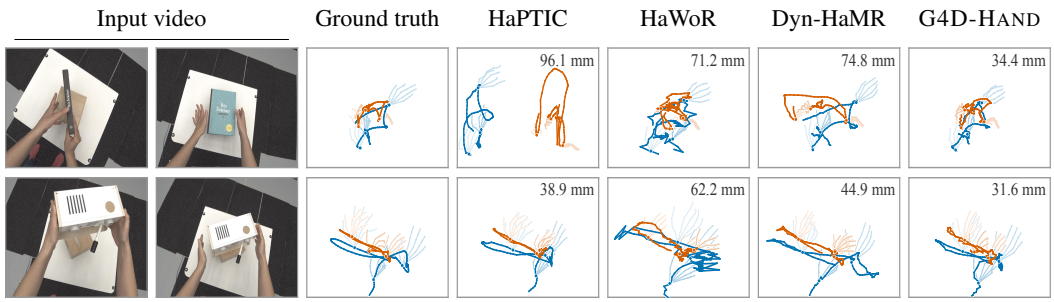

Figure 3: **Qualitative comparison of 4D hand trajectories on ARCTIC.** Each panel shows the left- and right-hand wrist trajectories over that window, with the hand poses at the two input frames, drawn from faint to solid in temporal order. The number in each panel reports G-MPJPE over the shown window. All methods are mapped to the first-frame camera coordinates using ground-truth poses, isolating hand reconstruction from camera-trajectory estimation. G4D-HAND yields stable trajectories that follow ground-truth dynamics and inter-hand positioning. Video results are provided in the supplementary material.

Table 3: **Camera-frame evaluation on the ARCTIC test web server.** Best and second-best results are **bold** and underlined. Lower is better; errors are in mm, except Acc. in m/s^2; evaluated on 21 keypoints.

	MRRPE	RA-MPJPE	Acc.
HaMeR	63.4	27.2	7.1
WiLoR	54.1	**15.6**	5.5
HaWoR	88.5	27.2	3.1
HaPTIC	325.7*	19.6	3.1
G4D-HAND	**20.7**	15.9	**1.9**

* HaPTIC predicts wrist depth relative to the first frame, leading to high localization errors when hands are initially absent (as in some ARCTIC test sequences).

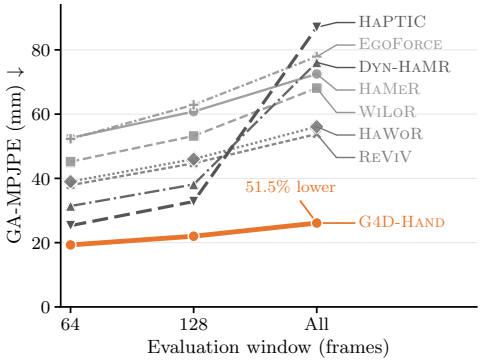

Figure 4: **ARCTIC world evaluation with ground-truth extrinsics.** As windows grow, G4D-HAND remains stable, with 51.5% lower all-frames rigid GA-MPJPE than the second-best.

of our method, likely because raymap conditioning provides a consistent geometric reference for camera-frame predictions.

Estimated camera trajectories. We compare against native world-frame methods (HaWoR, Dyn-HaMR) on the benchmarks in Section 4.1 and lift camera-frame baselines (HaMeR, WiLoR, HaPTIC, EgoForce) using CUT3R poses. CUT3R is trained to produce metric geometry, but its predictions are not perfectly metric. We recover absolute camera translation from ratios of predicted 3D keypoint depths to CUT3R depths sampled near the corresponding 2D predicted keypoints. We take the median across 16 joints per frame, skipping hands with fewer than 8 valid joints, then the median across frames within each chunk after one round of outlier rejection. The same procedure applies to G4D-HAND, which estimates camera motion with its frozen geometry backbone. This pipeline combines hand reconstruction, camera pose, and scale-recovery errors. Table 2 measures the accuracy of the combined pipeline (scale aligned, Dyn-HaMR SLAM is up to scale). G4D-HAND achieves the best world-frame trajectory accuracy, including 18% lower G-MPJPE than HaWoR on ARCTIC. HaPTIC generally has the highest errors among the lifted baselines and its relative depth predictions may contribute to this increase through scale recovery.

4.3 QUALITATIVE COMPARISON

Figure 3 illustrates the differences in trajectory estimation between G4D-HAND and previous works. In the first row, we visualize an example where HaPTIC distorts the inter-hand configuration, possibly because it does not enforce any localization consistency between hands and its relative depth estimation. HaWoR and Dyn-HaMR produce overly extended trajectories, with HaWoR also showing substantial jitter. In contrast, G4D-HAND preserves the spatial relationship between the hands

Table 4: **Ablation analysis on ARCTIC.** Raymap conditioning improves metric localization and trajectories. Replacing CUT3R with Pow3R retains strong performance. Per-hand temporal modeling reduces acceleration error. Lower is better; errors are in mm, except Acc. in m/s^2.

Model variant	G-MPJPE	GA-MPJPE	Wrist err.	MRRPE	RA-MPJPE	Acc.
G4D-HAND (full)	41.7	22.0	28.3	28.7	17.2	2.1
w/o raymap conditioning	53.8	28.9	41.4	33.6	17.3	2.2
w/ Pow3R backbone	42.2	22.7	26.1	27.9	17.5	2.1
w/o temporal transformer	44.8	23.4	28.5	28.9	16.9	3.7
+ w/o acceleration loss	44.1	23.4	27.8	28.4	16.6	4.6

and closely follows the ground-truth paths, even under occlusion. These examples are consistent with the trajectory and MRRPE results in Table 1.

4.4 ABLATION ANALYSIS

Does camera conditioning improve metric localization? Our idea for adding raymap conditioning was to provide the geometric features with an explicit camera reference for localizing hands across frames. Removing it increases wrist error by 46% (28.3 to 41.4 mm) and G-MPJPE by 29% (41.7 to 53.8 mm), while RA-MPJPE remains nearly unchanged (Table 4). The model thus retains its ability to reconstruct hand articulation, based on semantic encoder features, but is substantially less accurate at localizing the hands in 3D. Furthermore, removing raymap conditioning increases MRRPE, suggesting that its shared geometric reference also encourages consistent localization of the two hands within the same camera coordinate frame. This supports our design of conditioning the scene representation on camera intrinsics to recover metric hand positions and consistent trajectories.

Does G4D-HAND extend beyond CUT3R? Our approach reconstructs hands from pretrained scene features. To test whether this idea extends beyond CUT3R, we replace it with Pow3R (Jang et al., 2025), keeping the backbone frozen and the hand architecture unchanged. Trajectory accuracy is on par, while wrist error and MRRPE improve slightly, likely because Pow3R is natively trained with intrinsics. These results support our hypothesis. We use CUT3R for G4D-HAND because it processes video sequentially and is trained on dynamic scenes.

Is temporal modeling on hand features needed on top of a recurrent scene backbone? Although CUT3R already aggregates scene information over time, removing our hand-specific temporal transformer increases acceleration error from 2.1 to 3.7 m/s^2 and G-MPJPE from 41.7 to 44.8 mm. Removing acceleration supervision as well further increases acceleration error to 4.6 m/s^2, more than twice that of the original model. These results show that scene-level recurrence benefits from explicit modeling of hand dynamics: temporal refinement and supervision improve motion consistency beyond what the frozen geometric backbone provides.

5 CONCLUSION

We propose G4D-HAND, a geometry-grounded approach to 4D hand trajectory estimation. Unlike prior modular pipelines that rely on hand-specific appearance features, detection, motion infilling, and separate SLAM, G4D-HAND uses a single scene reconstruction backbone for both hand and camera trajectory estimation, leveraging 3D representations learned through large-scale reconstruction pretraining. It extracts hand-specific features without detectors or dedicated hand encoders, regresses MANO parameters, and directly predicts metric hand translation in camera coordinates, with raymap conditioning promoting consistency across hands and frames. G4D-HAND achieves the lowest localization and trajectory errors on three egocentric benchmarks while maintaining comparable articulation accuracy, training only 27M parameters, and running at over 30 FPS.

Limitations and future work. The current model is designed for egocentric videos and focuses on estimating the 3D trajectories of the wearer's two hands. For future work, we plan to extend it to handle an arbitrary number of hands and out-of-view scenarios. Another direction is to jointly finetune the reconstruction backbone for egocentric scene and hand estimation, which could improve camera trajectories and, by that, also world hand trajectories.

AI USE STATEMENT

In our work, we have used LLMs as a general-purpose assistant tool to enhance the clarity of writing, phrasing, and grammar. All ideas, experiments, and analyses were carried out by the authors.

ETHICS STATEMENT

This work uses existing egocentric video benchmarks and does not introduce new data collection. Because wearable cameras can capture other people and private spaces, applications of 4D hand tracking should respect the consent and privacy of everyone recorded. Hand trajectories may also reveal details about a person's activities. We will release code and model checkpoints to support research, but encourage users to consider these privacy risks before applying the method to new recordings.

REPRODUCIBILITY STATEMENT

To support reproducibility, we will release our code and checkpoints. Details on the architecture, training, and evaluation are provided in the Appendix.

REFERENCES

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 *Comp. Vis. Patt. Recognition*, 2025.

Fabien Baradel, Matthieu Armando, Salma Galaaoui, Romain Brégier, Philippe Weinzaepfel, Grégory Rogez, and Thomas Lucas. Multi-HMR: Multi-person whole-body human mesh recovery in a single shot. In *Proc. European Conf. Computer Vision*, 2024.

Yue Chen, Xingyu Chen, Yuxuan Xue, Anpei Chen, Yuliang Xiu, and Gerard Pons-Moll. Human3R: Everyone everywhere all at once. In *Proc. Int. Conf. Learning Representations*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learning Representations*, 2021.

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 *Comp. Vis. Patt. Recognition*, 2023.

Jisang Han, Seonghu Jeon, Jaewoo Jung, René Zurbrügg, Honggyu An, Tifanny Portela, Marco Hutter, Marc Pollefeys, Seungryong Kim, and Sunghwan Hong. Geometric action model for robot policy learning. In *arXiv preprint arXiv:2606.17046*, 2026.

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 *Proc. ACM Siggraph Asia*, 2022.

Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3R: Empowering unconstrained 3D reconstruction with camera and scene priors. In *Comp. Vis. Patt. Recognition*, 2025.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *Int. Conf. Robotics and Automation*, 2025.

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *Int. Conf. Computer Vision*, 2021.

Richard Li, Aditya Prakash, Andrew Wen, Saurabh Gupta, Yilun Du, and Pulkit Agrawal. What matters when cotraining robot manipulation policies on everyday human videos? In *RSS 2026 Workshop on Data-Centric Robotics: What Data Do Robots Really Need?*, 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 *Proc. European Conf. Computer Vision*, 2026.

Christen Millerdurai, Diogo Luvizon, Viktor Rudnev, André Jonas, Jiayi Wang, Christian Theobalt, and Vladislav Golyanik. 3D pose estimation of two interacting hands from a monocular event camera. In *Proc. of the Int. Conf. 3D Vision*, 2024.

Christen Millerdurai, Shaoxiang Wang, Yaxu Xie, Vladislav Golyanik, Didier Stricker, and Alain Pagani. EgoForce: Forearm-guided camera-space 3D hand pose from a monocular egocentric camera. In *Proc. ACM Siggraph*, 2026.

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *Comp. Vis. Patt. Recognition*, 2024.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3D hand localization and reconstruction in-the-wild. In *Comp. Vis. Patt. Recognition*, 2025.

Aditya Prakash, Ruisen Tu, Matthew Chang, and Saurabh Gupta. 3D hand pose estimation in everyday egocentric images. In *Proc. European Conf. Computer Vision*, 2024.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Int. Conf. Computer Vision*, 2021.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In *Proc. ACM Siggraph Asia*, 2017.

Md Salman Shamil, Dibyadip Chatterjee, Fadime Sener, Shugao Ma, and Angela Yao. On the utility of 3D hand poses for action recognition. In *Proc. European Conf. Computer Vision*, 2024.

Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J. Black. WHAM: Reconstructing world-grounded humans with accurate 3D motion. In *Comp. Vis. Patt. Recognition*, 2024.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Comp. Vis. Patt. Recognition*, 2026.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Comp. Vis. Patt. Recognition*, 2025.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. HoloAssist: An egocentric human interaction dataset for interactive AI assistants in the real world. In *Int. Conf. Computer Vision*, 2023.

Yufu Wang, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. TRAM: Global trajectory and motion of 3D humans from in-the-wild videos. In *Proc. European Conf. Computer Vision*, 2024.

Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *Comp. Vis. Patt. Recognition*, 2023.

Yufei Ye, Yao Feng, Omid Taheri, Haiwen Feng, Shubham Tulsiani, and Michael J. Black. Predicting 4D hand trajectory from monocular videos. In *Proc. of the Int. Conf. 3D Vision*, 2026.

Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. Dyn-HaMR: Recovering 4D interacting hand motion from a dynamic camera. In *Comp. Vis. Patt. Recognition*, 2025.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *Comp. Vis. Patt. Recognition*, 2025a.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *Proc. Int. Conf. Learning Representations*, 2025b.

APPENDIX

The appendix is organized as follows. Section A provides additional implementation details of G4D-HAND, and Section B describes the training and inference configuration and data augmentation. Section C complements the experiments in Section 4 with details on the evaluation protocol and metrics. For additional insights, Section D extends the experimental section with different evaluation protocols. Finally, Section E provides additional trajectory and hand pose reconstruction visualizations.

A EXTENDED ARCHITECTURE DETAILS

This section supplements Section 3.2 with the architecture, parameter sharing, and initialization details of G4D-HAND. It is divided into separate subsections corresponding to G4D-HAND components.

A.1 SCENE REPRESENTATION

G4D-HAND uses the final public CUT3R (Wang et al., 2025) checkpoint trained with 4–64 views and a DPT (Ranftl et al., 2021) head for pointmap prediction. All CUT3R parameters, including the image encoder, raymap encoder, and recurrent decoder, remain frozen. Input frames have height $H = 512$ and width $W = 384$, matching one of CUT3R's training resolutions.

The raymap $\mathbf{r} \in \mathbb{R}^{H\times W\times 6}$ concatenates a zero ray origin with a unit viewing direction at each pixel:

$$\mathbf{r}_{uv} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{d}_{uv} \end{bmatrix}, \qquad \mathbf{d}_{uv} = \frac{\mathbf{K}^{-1}[u, v, 1]^\top}{\|\mathbf{K}^{-1}[u, v, 1]^\top\|_2},$$

where (u, v) are the horizontal and vertical pixel coordinates, respectively, and $\mathbf{K}$ is the camera intrinsic matrix. The fixed reference pose expresses rays in local camera coordinates so that the raymap encodes intrinsics without encoding camera motion.

Hand-specific features are extracted from the raymap-conditioned encoder tokens. For wrist translation, the conditioned tokens are passed through the recurrent decoder, and pyramid features are taken at indices 0, 6, 9, and 12, matching the features used by CUT3R's DPT head. Camera motion is estimated through a separate decoder pass using the original image tokens without raymap conditioning and an independent recurrent state. Thus, the hand predictions are expressed in each frame's camera coordinates and can subsequently be transformed into the world frame.

A.2 HAND-SPECIFIC FEATURE EXTRACTION

The left- and right-hand extractors have independent parameters. Firstly, scene tokens are projected through a channel projection of widths $1024 \to 512 \to 1280$, with layer normalization, GELU activation, and dropout. Coarse and fine pooling paths then produce 192 tokens of dimension 1280, each using one layer of single-head learned-query cross-attention. The coarse path aggregates features with 32 queries and maps the resulting tokens to 192 output tokens through a learned linear map. The fine path directly uses 192 queries, one per output token. A learned sigmoid gate combines the two outputs, followed by layer normalization. The attention maps are initialized to be approximately uniform, and the gate is initialized to give equal weight to both paths.

A.3 TEMPORAL REFINEMENT

The temporal module is applied to each hand with shared weights. It uses a two-layer transformer encoder with four attention heads, hidden width 512, feed-forward width 1024, GELU activations, and dropout. Fixed sinusoidal frame-position encodings are added at the hidden width. A residual connection and initialization close to the identity preserve the input features at the start of training. Attention operates along the temporal axis independently at each token position. It is bidirectional within a sequence, allowing every frame to attend to all $S = 16$ sampled frames. The same sequence length is used during training and inference.

A.4 MANO PARAMETER REGRESSION

The MANO regressor follows HaMeR's transformer-decoder head and is initialized from its released checkpoint (Pavlakos et al., 2024). It contains six layers, eight attention heads with head dimension 64, feed-forward width 1024, and context dimension 1280.

To adapt the pretrained head to the hand-specific CUT3R features, the query, key-value, and output projections of every cross-attention layer are reinitialized and trained. The layer-normalization modules preceding the feed-forward blocks are also reinitialized and trained. All remaining head parameters retain their pretrained values and remain frozen. HaMeR's weak-perspective camera branch is frozen and its predictions are discarded, since metric wrist translation is estimated separately.

A.5 HAND TRANSLATION REGRESSION

The translation head $\mathcal{H}_{\text{wrist}}$ takes two inputs for each hand h at frame t: the full-scene feature pyramid $\widetilde{\mathcal{P}}_t$ from CUT3R's raymap-conditioned decoder pass, and the temporally refined hand tokens $\mathbf{H}_t^h$. The pyramid provides spatial context for localization, while the hand features condition the head on the target hand. All head parameters are shared between hands.

Scene context. The pyramid contains four feature tensors with channel dimensions $(1024, 768, 768, 768)$, taken from CUT3R's feature sequence at indices $(0, 6, 9, 12)$, excluding the camera token. Index 0 corresponds to the encoder features before decoder projection. All levels share the same spatial token grid. Each level is independently layer-normalized and linearly projected to width 512. The projected features are summed at corresponding token positions and layer-normalized again to form the cross-attention context.

Hand-conditioned queries. The hand tokens $\mathbf{H}_t^h$ are averaged over token positions to obtain a 1280-dimensional vector, then linearly projected to width 512 and layer-normalized. This vector is added to each of four learned queries.

Translation prediction. The conditioned queries attend to the scene context through a single cross-attention layer with eight heads, followed by a residual connection and layer normalization. The four output queries are concatenated and passed through an MLP of widths $2048 \to 512 \to 3$, with GELU activation and dropout, to predict the camera frame wrist translation $\boldsymbol{\tau}_t^h \in \mathbb{R}^3$ in meters.

B EXTENDED TRAINING AND INFERENCE DETAILS

B.1 TRAINING DATA AND OPTIMIZATION

A single model is trained jointly on HOT3D-Aria, ARCTIC-ego, H2O, and HOT3D-Quest, with sampling weights $(0.4, 0.3, 0.15, 0.15)$, respectively. For ARCTIC-ego and H2O we use the official released training sets. For HOT3D-Aria and HOT3D-Quest, we split the participant list, such that participants with indices (0, 1, 2) are used for evaluation, while the rest is used for training. The two HOT3D sources correspond to different capture devices within the same dataset. Table 5 lists the optimization settings.

Table 5: **Training configuration details.** Batch size is global across both GPUs.

Setting	
Hardware	2× NVIDIA RTX 4090
Training steps	80,000
Batch size	16 sequences × 16 frames
Optimizer	AdamW, $\boldsymbol{\beta} = (0.9, 0.999)$
Weight decay	0.01
Gradient clipping	Global norm 1.0
Learning rate (initial / peak / final)	1.41×10^{-5} / 1.41×10^{-4} / 10^{-6}
Schedule	Linear warm-up (10%), then cosine decay

B.2 SEQUENCE SAMPLING AND AUGMENTATION

During training, to improve the generalization and robustness of our method, we apply the augmentation detailed below. Beyond standard image augmentations and feature dropout, we include temporal stride sampling for generalization to different video frame rates, and spatial augmentations for improving robustness to different camera parameters.

Temporal sampling. Training sequences are sampled on the fly. A temporal stride is drawn from $\{1, 2, 4\}$ with probabilities $(0.5, 0.3, 0.2)$ and held constant within each sequence, ensuring uniform frame spacing.

Spatial augmentation. A single crop rectangle is applied to all frames in a sequence. The zoom factor is sampled log-uniformly from $[1.0, 1.6]$, with random position jitter. Each crop is resampled from the pre-crop image to $H \times W = 512 \times 384$, and the camera intrinsics are updated accordingly. Consequently, each sequence uses one intrinsic matrix and one raymap. Crops that remove more than 70% of the visible hand keypoints are rejected and resampled.

Appearance augmentation. With probability 0.67, one of color jitter, Gaussian blur, Gaussian noise, or random erasing is selected uniformly. The same augmentation parameters are applied to every frame in the sequence.

Feature regularization. The frozen scene features are regularized with token dropout at rate 0.15, excluding tokens in hand regions, and channel dropout at rate 0.05.

B.3 COMPUTE ANALYSIS

We measure G4D-HAND's runtime on 15 HOT3D (Banerjee et al., 2025) sequences of 128 frames each, using an NVIDIA GeForce RTX 4090, one CPU core, and 32 GB of RAM. For processing the sequence sequentially as eight windows of length $S = 16$, our method takes 4.13 s (31 FPS), with a peak GPU memory usage of 4.1 GB. Batching all eight windows reduces the runtime to 3.30 s (39 FPS) at the cost of increased peak memory usage of 10.6 GB. Running CUT3R accounts for approximately 95% of the total runtime.

B.4 LOSS DEFINITIONS

This section defines the terms in the training objective given in the main text. Hats denote predictions, and $h \in \{\mathrm{L}, \mathrm{R}\}$ indexes the hand. The two hand losses are weighted equally, and the remaining weights are $\lambda_{\mathrm{3D}} = 0.125$, $\lambda_{\mathrm{2D}} = 0.02$, $\lambda_{\mathrm{pose}} = 0.0075$, $\lambda_{\mathrm{orient}} = 0.075$, $\lambda_{\mathrm{transl}} = 5$, $\lambda_{\mathrm{accel}} = 0.08$, $\lambda^{\mathrm{ih}}_{\mathrm{3D}} = \lambda^{\mathrm{ih}}_{\mathrm{transl}} = 7.8$ and $\lambda^{\mathrm{ih}}_{\mathrm{shape}} = 0.078$. For the following subsections, $\mathbf{J}^h_t$ and $\hat{\mathbf{J}}^h_t$ denote ground-truth and predicted joint positions for hand h at frame t, expressed in the coordinate frame being evaluated. Let $\bar{\mathbf{J}}^h$ and $\hat{\bar{\mathbf{J}}}^h$ denote the root-relative ground truth and predicted joints.

B.4.1 PER-HAND RECONSTRUCTION

Joint positions are supervised in 3D and in the image plane:

$$\mathcal{L}^h_{\mathrm{3D}} = \|\hat{\bar{\mathbf{J}}}^h - \bar{\mathbf{J}}^h\|_1, \;\; \mathcal{L}^h_{\mathrm{2D}} = \|\hat{\mathbf{J}}^h_{\mathrm{2D}} - \mathbf{J}^h_{\mathrm{2D}}\|_1,$$

where $\mathbf{J}^h_{\mathrm{2D}}$ denotes projection of 3D joints into the image.

Articulated pose, wrist orientation, and metric wrist translation are supervised by:

$$\mathcal{L}^h_{\mathrm{pose}} = \textstyle\sum_{k=1}^{15} d_{\mathrm{geo}}\big(R(\hat{\boldsymbol{\Theta}}^h_k), R(\boldsymbol{\Theta}^h_k)\big), \;\; \mathcal{L}^h_{\mathrm{orient}} = d_{\mathrm{geo}}\big(R(\hat{\boldsymbol{\Theta}}^h_0), R(\boldsymbol{\Theta}^h_0)\big), \;\; \mathcal{L}^h_{\mathrm{transl}} = \|\hat{\boldsymbol{\tau}}^h - \boldsymbol{\tau}^h\|^2_2,$$

where $\boldsymbol{\Theta}^h_0$ denotes the root (wrist) rotation and $\boldsymbol{\Theta}^h_k$, $k \in \{1, \ldots, 15\}$, the relative finger-joint rotations. $R(\cdot)$ converts the axis-angle pose representation defined in the main text to a rotation matrix. The geodesic distance between rotations is

$$d_{\mathrm{geo}}(\mathbf{R}_1, \mathbf{R}_2) = \arccos\left(\frac{\mathrm{tr}(\mathbf{R}_1^\top \mathbf{R}_2) - 1}{2}\right).$$

B.4.2 INTER-HAND CONSISTENCY

Inter-hand supervision constrains relative joint positions and wrist translations, while a shape-consistency term encourages similar shape parameters for the two hands:

$$\mathcal{L}_{\text{3D}}^{\text{ih}} = \|(\hat{\mathbf{J}}^{\text{L}} - \hat{\mathbf{J}}^{\text{R}}) - (\mathbf{J}^{\text{L}} - \mathbf{J}^{\text{R}})\|_2^2, \mathcal{L}_{\text{transl}}^{\text{ih}} = \|(\hat{\boldsymbol{\tau}}^{\text{L}} - \hat{\boldsymbol{\tau}}^{\text{R}}) - (\boldsymbol{\tau}^{\text{L}} - \boldsymbol{\tau}^{\text{R}})\|_2^2, \;\; \mathcal{L}_{\text{shape}}^{\text{ih}} = \|\hat{\boldsymbol{\beta}}^{\text{L}} - \hat{\boldsymbol{\beta}}^{\text{R}}\|_2^2.$$

B.4.3 TEMPORAL CONSISTENCY

The temporal loss matches second-order changes in predicted and ground-truth joint positions:

$$\mathcal{L}_{\text{accel}}^{h} = \|\Delta^2 \hat{\bar{\mathbf{J}}}^{h} - \Delta^2 \bar{\mathbf{J}}^{h}\|_1.$$

This term penalizes deviations from the ground-truth dynamics rather than directly penalizing the magnitude of hand motion.

C EXTENDED EVALUATION DETAILS

C.1 DATASETS AND EVALUATION SPLITS

We evaluate on the Aria subset of HOT3D (Banerjee et al., 2025), H2O (Kwon et al., 2021), and ARCTIC (Fan et al., 2023). For HOT3D, we use a subject-level split of HOT3D-clips (curated set with verified annotations). As the official test set is unavailable, we create our split in which participants with indices $(0, 1, 2)$ are used for evaluation in this work and excluded from training sets of both HOT3D-Aria and HOT3D-Quest. For H2O, we evaluate on the official test split. For ARCTIC, we report results on the official validation and test splits, trajectory metrics are evaluated on the validation split, as they are not supported by the official test evaluation website. For the ARCTIC validation set, we start at the first frame with both hands visible to avoid HaPTIC's sensitivity to empty frame initialization. This excludes about 3% of valid frames per hand. On the ARCTIC test web server, the evaluation is done on all frames. Evaluation is performed on in-the-view hands.

C.2 EVALUATION DETAILS AND METRICS

Let $\mathbf{J}_{t,j}^h$ and $\hat{\mathbf{J}}_{t,j}^h$ denote the ground-truth and predicted positions of joint j of hand h at frame t, expressed in the coordinate frame being evaluated. Let $\bar{\mathbf{J}}_{t,j}^h$ and $\hat{\bar{\mathbf{J}}}_{t,j}^h$ denote the corresponding root-relative ground-truth and predicted joint positions. Our main results use the original $J = 16$ MANO skeletal joints ($j = 0$ denotes the wrist), excluding the five fingertips estimated from the mesh, as our primary focus is hand localization and trajectory recovery. Results using the full 21-joint convention are provided in Table 7. Positional errors are reported in mm and acceleration errors in $\mathrm{m/s^2}$. We follow the evaluation metrics and aggregation procedure of Fan et al. (2023).

Joint position error. MPJPE measures joint localization error, while root-aligned MPJPE (RA-MPJPE) evaluates articulation after independently subtracting the predicted and ground-truth wrist positions:

$$\text{MPJPE}_t^h = \frac{1}{J}\sum_{j=0}^{J-1} \left\| \hat{\mathbf{J}}_{t,j}^h - \mathbf{J}_{t,j}^h \right\|_2, \quad \text{RA-MPJPE}_t^h = \frac{1}{J}\sum_{j=0}^{J-1} \left\| (\hat{\mathbf{J}}_{t,j}^h - \hat{\mathbf{J}}_{t,0}^h) - (\mathbf{J}_{t,j}^h - \mathbf{J}_{t,0}^h) \right\|_2.$$

Wrist position error. The wrist error (WE) isolates the global localization of each hand, independent of articulation, by measuring the Euclidean distance between the predicted and ground-truth wrist joints:

$$\text{WE}_t^h = \left\| \hat{\mathbf{J}}_{t,0}^h - \mathbf{J}_{t,0}^h \right\|_2.$$

While RA-MPJPE removes the wrist translation to evaluate hand pose, WE captures only the translational component that RA-MPJPE discards, and is therefore sensitive to errors in depth and scale estimation.

Inter-hand position error. MRRPE measures the error in the displacement between the two wrists:

$$\mathrm{MRRPE}_t = \left\| (\hat{\mathbf{J}}^{\mathrm{L}}_{t,0} - \hat{\mathbf{J}}^{\mathrm{R}}_{t,0}) - (\mathbf{J}^{\mathrm{L}}_{t,0} - \mathbf{J}^{\mathrm{R}}_{t,0}) \right\|_2 .$$

It is evaluated only on frames where both hands have valid ground-truth annotations.

Trajectory errors. G-MPJPE and GA-MPJPE measure MPJPE after aligning each predicted hand trajectory to the ground truth with Umeyama algorithm independently for each hand and evaluation clip. G-MPJPE estimates the transform from the first two frames, while GA-MPJPE uses all frames, considering only valid ground-truth annotations in both cases. The transform is then applied to the entire clip. Evaluation uses non-overlapping clips of 128 frames in camera coordinates and 64 frames in world coordinates. We use non-scale aligned GA-MPJPE, unless otherwise noted in the text.

Acceleration error. Acceleration is computed from root-relative joints using second-order finite differences:

$$\mathbf{a}^h_{t,j} = \frac{\bar{\mathbf{J}}^h_{t-1,j} - 2\bar{\mathbf{J}}^h_{t,j} + \bar{\mathbf{J}}^h_{t+1,j}}{\Delta t^2}, \quad \mathrm{Acc}^h_t = \frac{1}{J}\sum_{j=0}^{J-1} \left\| \hat{\mathbf{a}}^h_{t,j} - \mathbf{a}^h_{t,j} \right\|_2 .$$

Predicted accelerations are computed analogously, with joint positions in meters and $\Delta t = 1/\nu$, where ν denotes the video frame rate. For all datasets in our evaluation, $\nu = 30$ FPS, and hence $\Delta t = 1/30$ seconds. Only triplets with valid ground truth for the same hand in all three frames are evaluated, and triplets crossing evaluation-chunk boundaries are excluded. This metric quantifies temporal motion consistency with the ground truth.

Aggregation. For each per-hand metric, errors are first averaged over valid hands within each frame, then over valid frames across all sequences. MRRPE is averaged over frames with both hands valid.

C.3 Baselines details

All baselines are evaluated using their publicly released checkpoints. For a fair comparison on metrics that are not root-aligned, all baselines are run using ground-truth camera intrinsics. For HaMeR (Pavlakos et al., 2024) and WiLoR (Potamias et al., 2025), the ground-truth intrinsics are used when converting wrist translation in weak-to-full. EgoForce's (Millerdurai et al., 2026) architecture takes intrinsics as input. HaPTIC (Ye et al., 2026) similarly uses ground-truth intrinsics for its weak-perspective wrist translation initialization in the first frame. ReViV learns metric hand positions based on the training datasets, and intrinsics can be used only to project 3D to 2D keypoints. Dyn-HaMR (Yu et al., 2025) and HaWoR (Zhang et al., 2025a) use ground-truth intrinsics both for their camera-space hand predictions, which rely on weak-perspective camera models, and within their SLAM pipelines. Both of these models predict the hand pose for the full sequence, due to their motion infilling. Our model uses ground-truth intrinsics to construct the raymap conditioning, while wrist translation is predicted directly in metric camera space, without weak-to-full. For Dyn-HaMR and HaWoR evaluation in the camera frame, we transform their final world-frame predictions using their estimated camera extrinsics. These methods perform key refinement stages directly in world space, including motion infilling and, for Dyn-HaMR, interacting-hand optimization. Evaluating their predictions before these stages would therefore omit core components of the methods and would not be a fair comparison. WiLoR, HaWoR, ReViV, and EgoForce were trained on HOT3D, but their training subject splits are not reported. HaWoR uses HOT3D as its main training set and the only one for the motion infiller network. Since we cannot verify whether our evaluation subjects were excluded from training, we omit HaWoR from the HOT3D comparison. We still report WiLoR, ReViV, and EgoForce for reference, as they were trained on broader collections of datasets that include HOT3D.

D Extended quantitative analysis

D.1 Camera frame evaluation with scale alignment

In Table 6, we compare against the state of the art on camera-frame trajectory metrics after Umeyama alignment with a similarity transform (including scale), a standard metric in hand pose evaluation.

Table 6: **Camera-frame trajectory evaluation with scale alignment.** We report G-MPJPE and GA-MPJPE after similarity alignment, including scale. Best and second-best results per dataset are **bold** and underlined. Lower is better; all errors are in mm.

	ARCTIC		H2O		HOT3D	
Method	**G-MPJPE**	**GA-MPJPE**	**G-MPJPE**	**GA-MPJPE**	**G-MPJPE**	**GA-MPJPE**
HaMeR	85.4	37.2	34.9	16.5	155.6	53.4
WiLoR*	77.1	35.6	30.4	13.3	93.0	42.2
EgoForce*	94.4	39.9	25.9	14.0	158.1	57.1
Dyn-HaMR	70.6	29.5	35.6	12.9	104.4	35.7
HaWoR*	79.2	35.1	30.7	17.8	–	–
HaPTIC	61.5	27.0	34.4	16.7	129.2	48.0
ReViV*	86.8	40.2	42.1	21.9	114.0	43.7
G4D-HAND	**42.1**	**21.1**	**21.4**	**10.7**	**47.4**	**19.1**

* Possible overlap between the method's training data and our HOT3D evaluation set. We omit HaWoR on HOT3D as HOT3D is its main training set and the subject split is not disclosed.

However, in our setting all methods either receive ground-truth intrinsics or directly predict in a metric camera frame (ReViV). We therefore report results with rigid alignment only in the main paper (Table 1), since a scale correction would hide errors in the recovered hand depth and trajectory scale, which a metric method is expected to estimate correctly. For completeness, the results with scale alignment show that G4D-HAND remains the best on every dataset and metric, while its errors are almost unchanged compared to rigid alignment (e.g., 41.7 vs. 42.1 mm G-MPJPE on ARCTIC).

The large improvements of the single-frame methods HaMeR and WiLoR (e.g., GA-MPJPE on ARCTIC and HOT3D) can be explained by how they predict hand depth. Both estimate a weak-perspective camera on the hand crop and obtain metric depth only indirectly from the apparent hand size, so any error in the estimated size translates into a proportional depth error, a global scale error of the trajectory. This effect is amplified in egocentric view, especially if hands are close to the camera.

D.2 EVALUATION ON 21 KEYPOINTS

In Table 7, we repeat the camera-frame comparison of Table 1 on all 21 hand joints, including the fingertips. All errors increase slightly for every method, but the overall ranking remains similar. G4D-HAND still achieves the lowest G-MPJPE and GA-MPJPE on all three datasets, and the best or on-par articulation accuracy (RA-MPJPE).

D.3 ROBUSTNESS TO HAND-OBJECT OCCLUSIONS

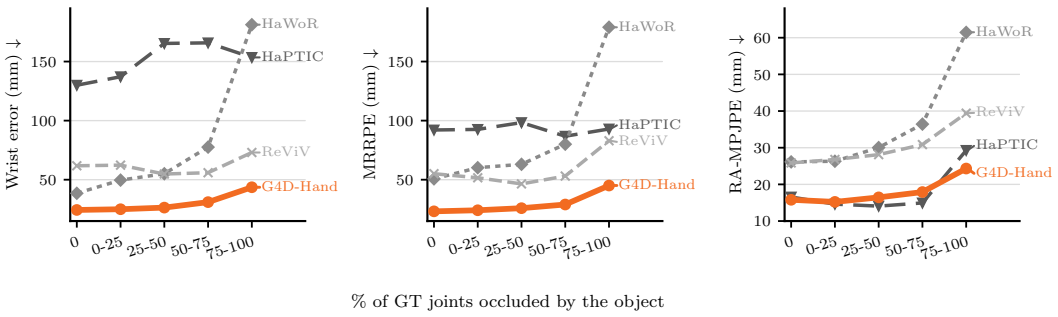

Figure 5: **Robustness to hand-object occlusion on ARCTIC validation set.** We group hand frames by the fraction of ground-truth joints occluded by the interacting object. G4D-HAND maintains the lowest wrist error and MRRPE across all occlusion levels, and achieves the lowest articulation errors under the strongest occlusion. For MRRPE, frames are binned by the fraction of occluded joints across both hands.

Table 7: **Camera frame SOTA comparison on three egocentric benchmarks; evaluated on 21 joints.** We report trajectory (rigid G-MPJPE, GA-MPJPE), articulation (RA-MPJPE), and temporal consistency (Acc.) errors. MRRPE and Wrist err. do not depend on number of keypoints. Best and second-best results per dataset are **bold** and underlined. Lower is better; errors are in mm, except Acc. in m/s^2. The ranking is consistent with the main evaluation as G4D-HAND achieves the lowest trajectory errors on all datasets.

		G-MPJPE	GA-MPJPE	RA-MPJPE	Acc.
ARCTIC	HaMeR	98.6	61.0	29.0	8.3
	WiLoR	82.0	53.9	20.0	7.7
	EgoForce	100.2	64.1	23.4	8.4
	Dyn-HaMR	75.0	39.8	28.0	2.7
	HaWoR	88.2	47.6	35.9	3.9
	HaPTIC	62.5	32.7	**19.6**	3.6
	ReViV	90.1	47.3	33.5	4.6
	G4D-HAND	**44.3**	**23.7**	20.0	**2.5**
H2O	HaMeR	36.4	19.6	18.1	4.9
	WiLoR	31.2	15.4	16.0	3.8
	EgoForce	28.2	17.9	15.8	4.3
	Dyn-HaMR	31.8	18.5	17.3	2.0
	HaWoR	31.7	20.5	18.9	3.0
	HaPTIC	34.6	19.8	**15.5**	2.6
	ReViV	43.8	25.0	27.6	3.1
	G4D-HAND	**22.7**	**12.4**	**15.5**	**1.8**
HOT3D	HaMeR	200.1	89.9	62.6	13.1
	WiLoR*	105.6	55.6	22.7	12.4
	EgoForce*	201.0	93.6	41.6	17.4
	Dyn-HaMR	97.3	47.0	38.2	**2.5**
	HaPTIC	152.8	75.6	56.8	7.5
	ReViV*	126.0	54.5	44.0	5.8
	G4D-HAND	**51.1**	**22.7**	**21.4**	2.9

* Possible overlap between their training and our evaluation HOT3D set. We omit HaWoR results on HOT3D as it is its main train set and the subject split is not disclosed in paper.

We analyze robustness to hand-object occlusions on the ARCTIC validation set. We benchmark against trajectory methods ReViV (Lyu et al., 2026), HaPTIC (Ye et al., 2026), and HaWoR (Zhang et al., 2025a). We compare wrist error, MRRPE, and RA-MPJPE across five bins corresponding to different percentages of occluded joints. Results are shown in Figure 5. For wrist error and RA-MPJPE, each hand is binned by the fraction of its ground-truth joints occluded by the object. For MRRPE, which involves both hands, the fraction is computed over their joints combined.

G4D-HAND achieves the lowest wrist error and MRRPE in every occlusion bin. Under 75-100% occlusion, its wrist error is 43.6 mm, compared with 73.0 mm for the next-best baseline, while MRRPE is 45.1 mm compared with 83.0 mm. In the same occlusion range, G4D-HAND also achieves the best RA-MPJPE of 24.3 mm. Interestingly, despite being trained with motion infilling for occluded hands, HaWoR degrades substantially under stronger occlusion, with the highest RA-MPJPE in the two most occluded bins. This suggests that the occlusion robustness learned largely from HOT3D may not transfer as effectively to ARCTIC.

D.4 EXTENDED WORLD FRAME EVALUATION WITH GROUND TRUTH EXTRINSICS

In Table 8, we report the full analysis of world frame evaluation when transforming from camera to world with ground truth extrinsics. Devices such as Project Aria glasses and Meta Quest headsets provide reliable camera trajectory estimation, so the camera-to-world transform can be treated as known at inference time. This shifts the difficulty of world-space hand tracking away from camera localization and onto the accuracy of the camera-frame prediction itself: any error in the predicted hand position relative to the camera is carried directly into the world frame, and errors can accumu-

late. G4D-HAND achieves the lowest error in every window, and its margin over the best baseline grows from 6.0 mm (64 frames) to 27.7 mm (all frames).

Table 8: **World-frame hand pose evaluation on ARCTIC with ground-truth camera trajectory.** We report rigid-aligned G-MPJPE and GA-MPJPE over 64-, 128-, and all-frame evaluation windows, without scale alignment. Best and second-best results are **bold** and underlined. Lower is better; all errors are in mm.

	64-frame		128-frame		all frames	
Method	**G-MPJPE**	**GA-MPJPE**	**G-MPJPE**	**GA-MPJPE**	**G-MPJPE**	**GA-MPJPE**
HaMeR	85.2	52.6	99.5	60.8	105.6	72.5
WiLoR	68.3	45.2	81.8	53.2	<u>79.0</u>	68.1
EgoForce	80.9	52.3	97.3	62.9	84.5	77.9
HaPTIC	<u>48.8</u>	<u>25.3</u>	<u>62.1</u>	<u>32.8</u>	170.1	87.0
ReViV	71.9	37.9	87.7	44.7	90.4	<u>53.8</u>
Dyn-HaMR	59.7	31.4	77.7	38.1	115.4	76.1
HaWoR	73.0	39.0	88.1	46.0	80.2	56.1
G4D-HAND	**36.6**	**19.3**	**42.0**	**22.0**	**37.9**	**26.1**

D.5 EXTENDED ABLATION ANALYSIS

We remove all inter-hand supervision while keeping the architecture and per-hand losses unchanged. Results on ARCTIC are presented in Table 9. MRRPE increases by 8%, whereas wrist error changes only marginally. This supports that the inter-hand loss improves the relative configuration of the two hands by coupling their predictions.

	Wrist err [mm] ↓	**MRRPE** [mm] ↓
G4D-HAND	28.3	28.7
w/o inter-hand loss	27.9	31.1

Table 9: **Inter-hand loss ablation on ARCTIC.** We report wrist, and inter-hand (MRRPE) errors.

E EXTENDED QUALITATIVE RESULTS

E.1 GRADIENT ATTRIBUTION

Unlike prior hand-reconstruction methods (Potamias et al., 2025; Pavlakos et al., 2024; Zhang et al., 2025a; Yu et al., 2025), G4D-HAND operates without a hand detector or hand-centered crop. Our proposed hand-specific extractors read full-frame scene features and must therefore learn to localize each hand on their own. To test whether this specialization in fact emerges, and to identify which image regions influence each hand reconstruction, we measure how sensitive the predicted hands are to individual image tokens. For each hand $h \in \{\mathrm{L}, \mathrm{R}\}$ we form a scalar summary of its predicted joints in MANO canonical frame, $\mathcal{S}^h = \sum_{t=1}^{S} \sum_{j=0}^{J-1} \left\| \hat{\mathbf{J}}^h_{t,j} \right\|_2$, and backpropagate it to the frozen CUT3R encoder tokens $\mathbf{e}_i \in \widetilde{\mathbf{F}}_t$, assigning each token the attribution $a^h_i = \left\| \partial \mathcal{S}^h / \partial \mathbf{e}_i \right\|_2$.

Figure 6 visualizes the 15 highest-attribution tokens per branch on frames from the out-of-distribution HoloAssist dataset (Wang et al., 2023). The maps are strongly concentrated: the top-15 tokens carry roughly 40% of the total attribution magnitude in a frame, against 2.0% under a uniform distribution. More importantly, this mass falls predominantly on the branch's own hand, across different scenarios, including two-handed grasps, asymmetric manipulation, motion blur, and hands truncated by the image boundary. Despite receiving full-frame features and no explicit attention/localization supervision on the extractors, these modules discover and specialize to their respective hands without detectors or crops, which indicates that the frozen scene-reconstruction features retain spatially localized hand information. In Figure 7, we visualize attribution maps on ARCTIC (Fan et al., 2023) validation frames under varying levels of hand occlusion. When a hand becomes fully occluded, the corresponding branch no longer concentrates exclusively on the hand region; instead, its attribution shifts primarily toward the visible forearm and less toward the occluding object.

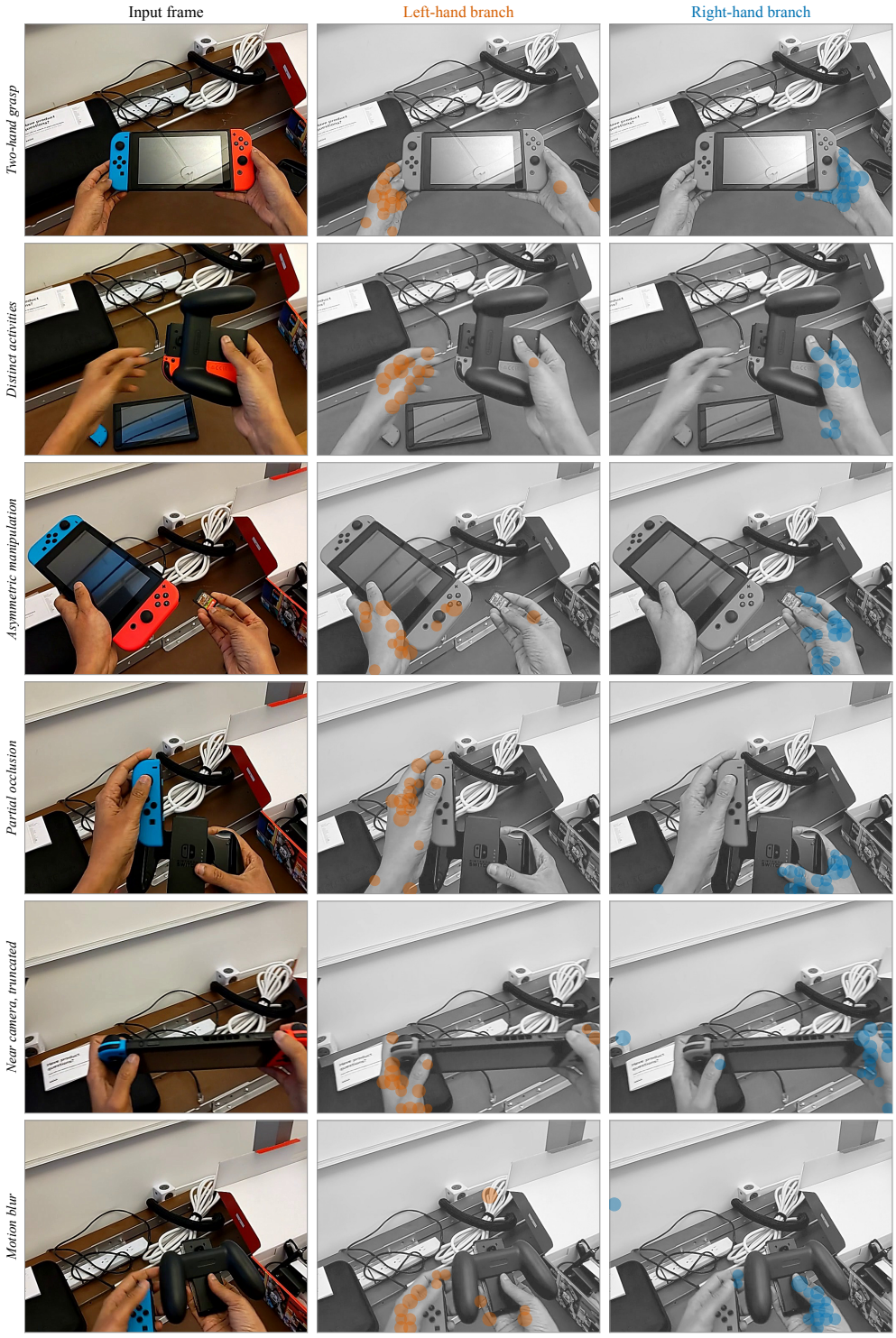

Figure 6: **Gradient attribution of the hand-specific feature extractors on out-of-distribution frames from HoloAssist.** We visualize the 15 CUT3R tokens with the highest gradient attribution separately for the left- and right-hand branches, with circle size indicating attribution rank. Across diverse scenarios, the highest-attribution tokens predominantly localize around the corresponding hand, despite the extractors operating on full-frame features without hand detections or crops.

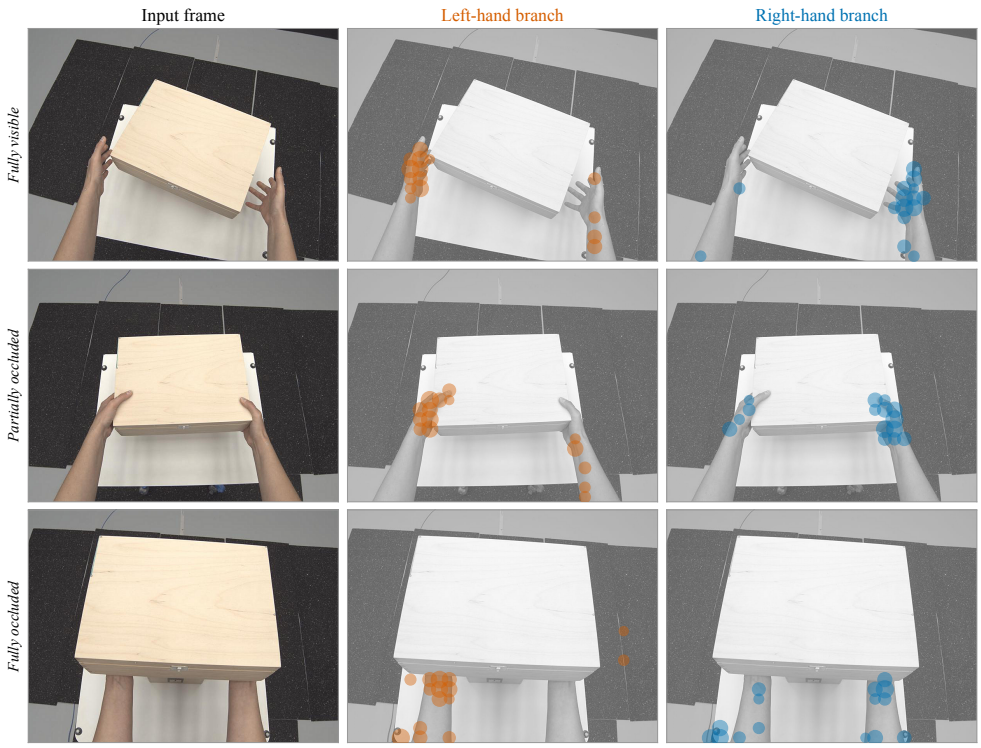

Figure 7: **Gradient attribution of the hand-specific feature extractors on ARCTIC frames with increasing hand occlusion.** We visualize the 15 CUT3R tokens with the highest gradient attribution separately for the left- and right-hand branches, with circle size indicating attribution rank. When the object hides both hands completely, each branch falls back mainly to the visible forearm on its own side.

E.2 Trajectory visualization

Camera-frame trajectories on ARCTIC. Figure 8 visualizes the camera-frame hand trajectories predicted by G4D-HAND with the ground truth on ARCTIC (Fan et al., 2023) validation sequences. Across different hand motion scenarios, G4D-HAND accurately recovers both the overall shape and spatial extent of the trajectories, including inter-hand configuration and changes in depth. The predicted hand poses also remain temporally consistent along the trajectory. The remaining errors primarily appear as small high-frequency deviations around the ground-truth path.

Comparison with ReViV on HOT3D. Figure 9 shows ground truth, ReViV, and G4D-HAND predictions for five egocentric clips from HOT3D. Sequences cover both-hand visibility and left- and right-hand-only cases. G4D-HAND closely follows the ground-truth motion, while ReViV shows larger trajectory deviations, occasional drift, and spurious loops.

World-frame trajectories on ARCTIC. Figure 10 compares HaWoR with our method on four 64-frame sequences from ARCTIC, covering both-hand and single-hand visibility. Methods are run without provided ground truth extrinsics. Both are expressed in the first-frame camera of each 64-frame chunk using each method's own camera motion. HaWoR's trajectories show frequent jumps and artifacts; G4D-HAND recovers smooth paths that closely follow the ground truth in shape and placement, performing well on the world-frame task.

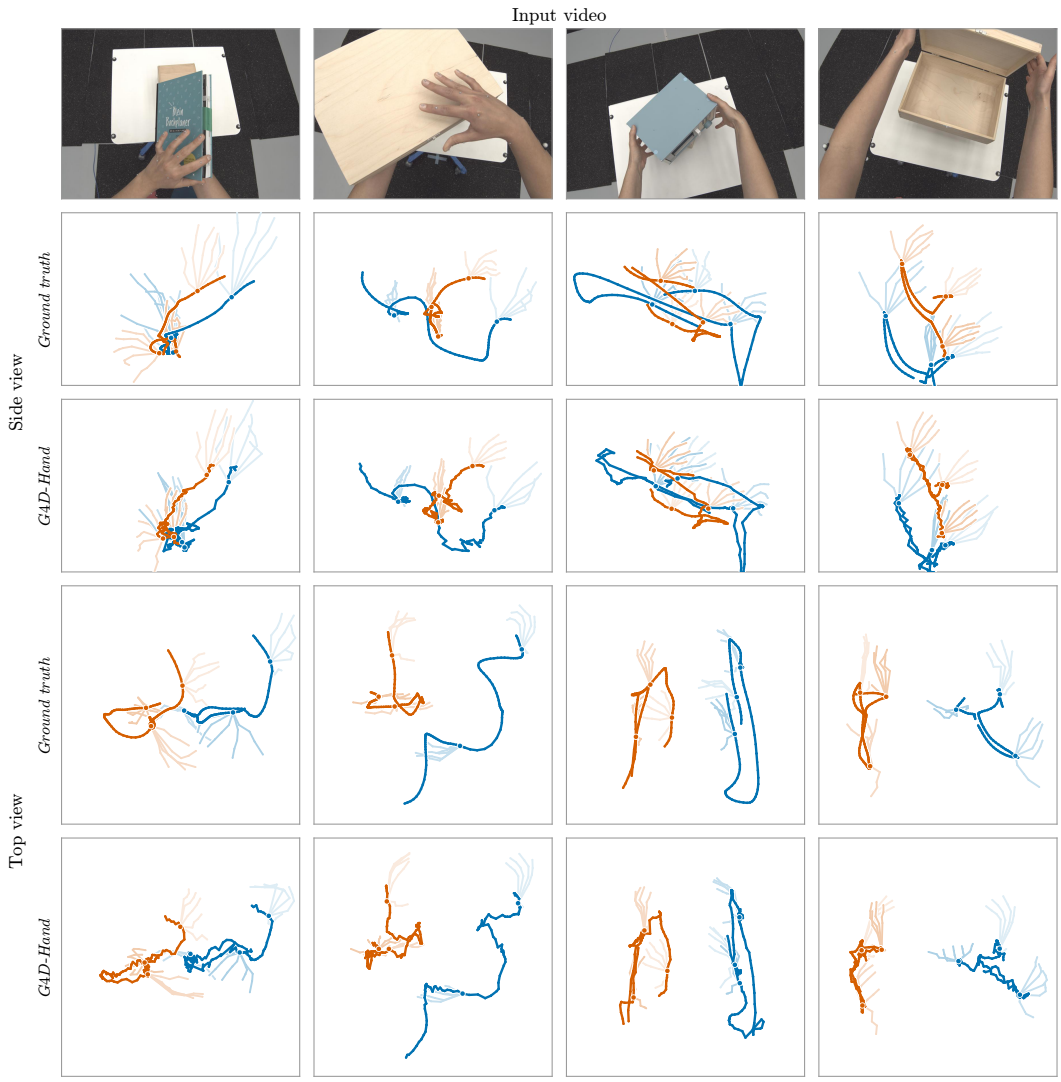

Figure 8: **Estimated 4D hand trajectories of G4D-HAND compared with ground truth on four ARCTIC validation sequences.** The top row shows the middle input frame of the ~ 4 s video. The remaining rows visualize the left- and right-hand wrist trajectories and hand poses, rendered from faint to solid in temporal order. Side and top views are shown, with ground truth above G4D-HAND prediction in each pair. Results are mapped to the first-frame camera coordinates using ground-truth camera poses, isolating hand reconstruction from camera-trajectory estimation. G4D-HAND closely follows the shape and extent of the ground-truth trajectories from both viewpoints, with only minor high-frequency deviations. Video results are provided in the supplementary material.

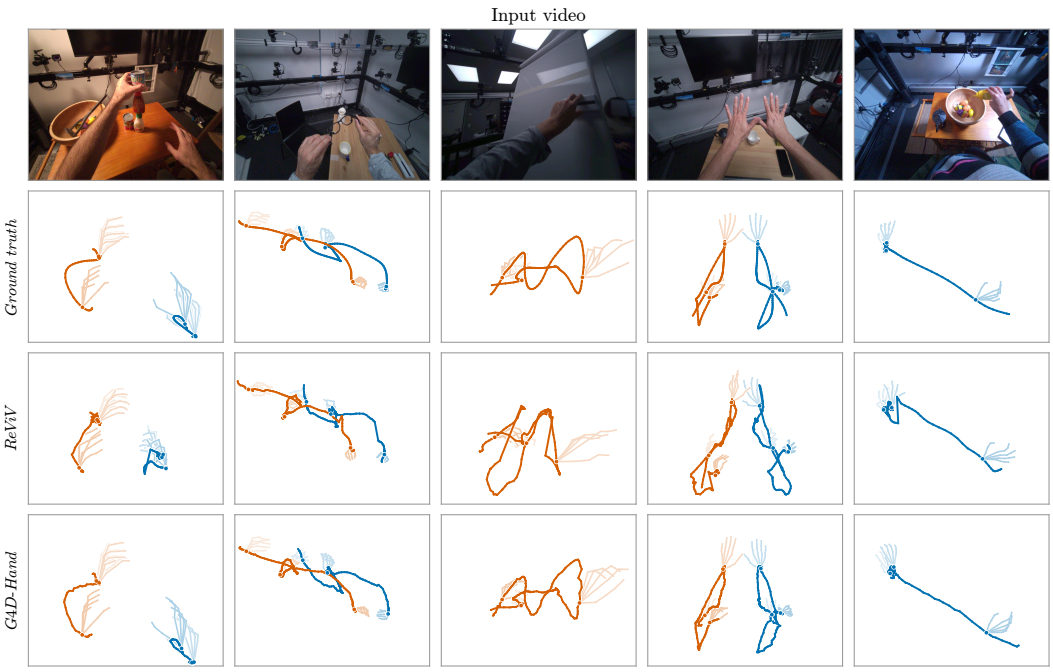

Figure 9: **Top-view comparison between ReViV and G4D-HAND on five sequences of HOT3D, with both, only left, and only right hand visible.** The top row shows an input frame from the ~ 4 s video, the next one shows ground truth trajectory, followed by ReViV and G4D-HAND predictions. Visualizations show left- and right-hand wrist trajectories and hand poses, rendered from faint to solid in temporal order. Results are mapped to the first-frame camera coordinates using ground-truth camera poses, isolating hand reconstruction from camera-trajectory estimation. G4D-HAND closely follows the ground-truth motion across different hand-visibility scenarios and under complex hand motion. Video results are provided in the supplementary material.

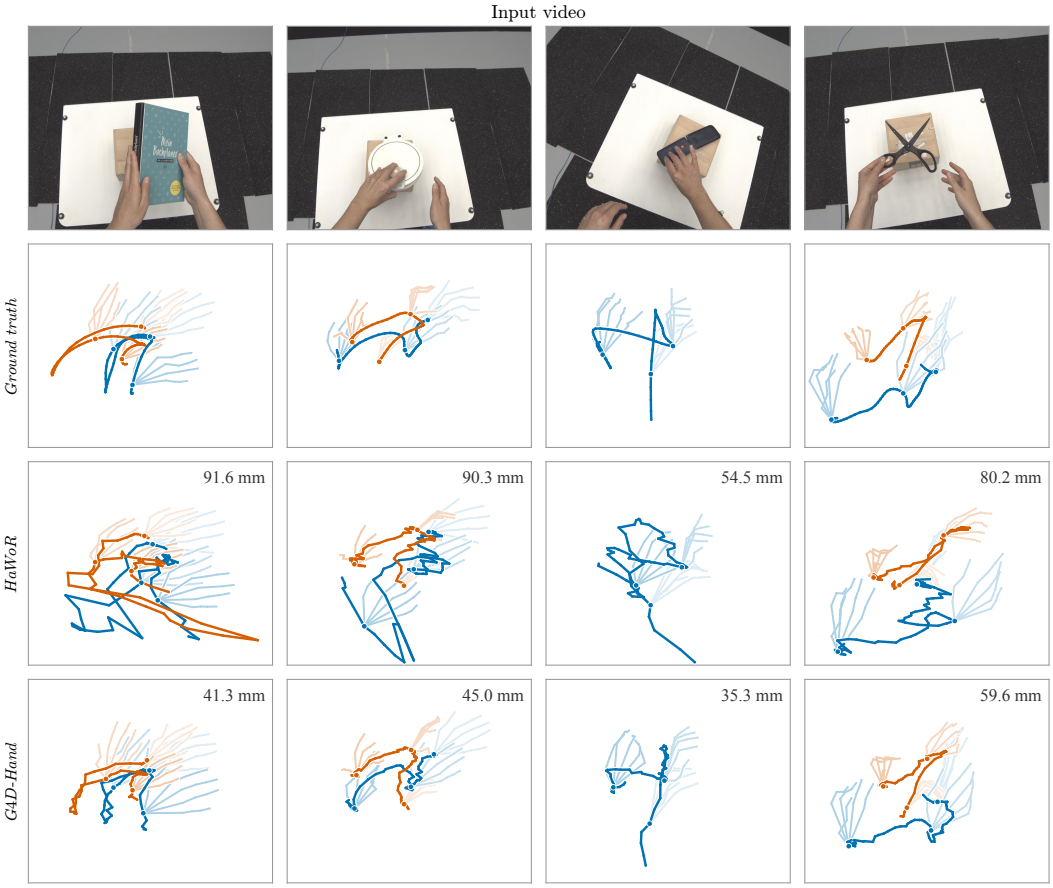

Figure 10: **Side-view comparison between HaWoR and G4D-HAND on ARCTIC in world frame.** Methods run without given camera extrinsics. The top row shows an input frame from the ~ 2 s video, the next one shows ground truth trajectory, followed by HaWoR and G4D-HAND predictions. Visualizations show left- and right-hand wrist trajectories and hand poses, rendered from faint to solid in temporal order with denoted mean 3D world MPJPE in the chunk in top right corner. Despite relying on CUT3R rather than an external SLAM system for camera trajectory estimation, G4D-HAND achieves accurate world-frame performance. Video results are provided in the supplementary material.