EgoForce: Forearm-Guided Camera-Space 3D Hand Pose from a Monocular Egocentric Camera

CHRISTEN MILLERDURAI, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany
SHAOXIANG WANG, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany
YAXU XIE, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany
VLADISLAV GOLYANIK, Max Planck Institute for Informatics (MPII), Germany
DIDIER STRICKER, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany
ALAIN PAGANI, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany

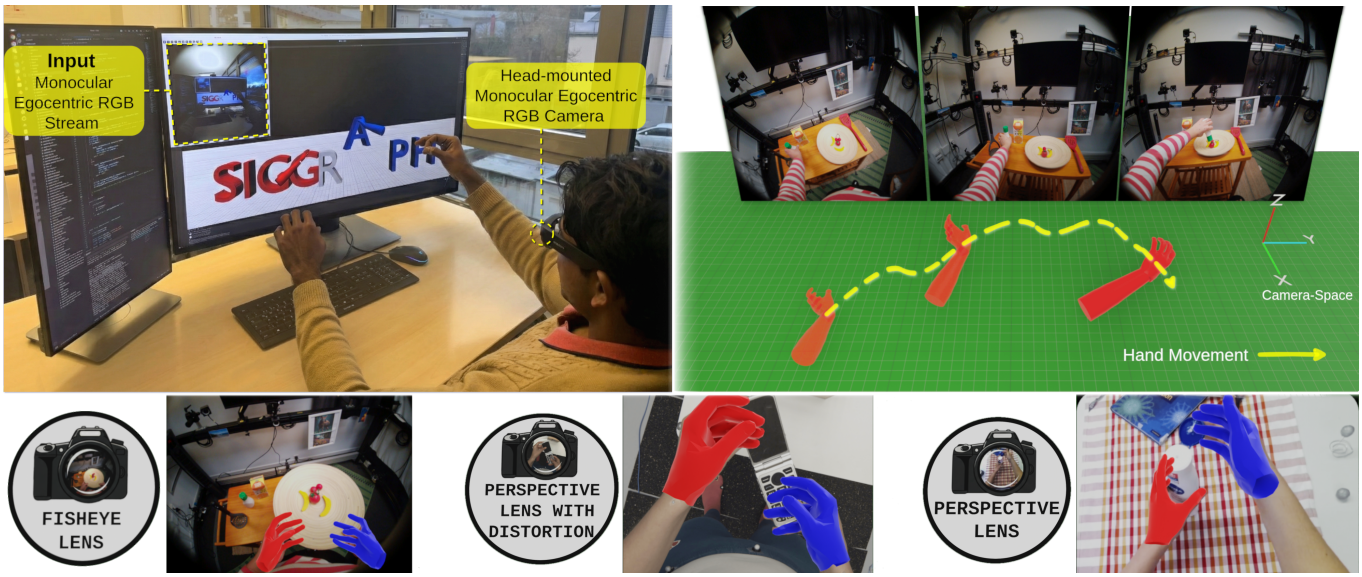


Fig. 1. **EgoForce reconstructs the absolute 3D pose and shape of the hands from the user's viewpoint using a monocular RGB camera from Aria glasses (top left).** With a unified framework, it supports diverse camera models while producing accurate 3D hand pose and shape (bottom), and recovers the absolute 3D hand position in the egocentric frame (top right), enabling metrically meaningful, viewpoint-consistent 3D tracking.

Abstract

Reconstructing the absolute 3D pose and shape of the hands from the user's viewpoint using a single head-mounted camera is crucial for practical egocentric interaction in AR/VR, telepresence, and hand-centric manipulation tasks, where sensing must remain compact and unobtrusive. While monocular RGB methods have made progress, they remain constrained by depth–scale ambiguity and struggle to generalize across the diverse optical configurations of head-mounted devices. As a result, models typically require extensive training on device-specific datasets, which are costly and laborious to acquire. This paper addresses these challenges by introducing EgoForce, a monocular 3D hand reconstruction framework that recovers robust, absolute 3D hand pose and its position from the user's (camera-space) viewpoint. EgoForce operates across fisheye, perspective, and distorted wide-FOV camera models using a single unified network. Our approach combines a differentiable forearm representation that stabilizes hand pose, a unified arm–hand transformer that predicts both hand and forearm geometry from a single egocentric view, mitigating depth–scale ambiguity, and a ray space closed-form solver that enables absolute 3D pose recovery across diverse head-mounted camera models. Experiments on three egocentric benchmarks show that EgoForce achieves state-of-the-art 3D accuracy, reducing camera-space MPJPE by up to 28% on the HOT3D dataset compared to prior methods and maintaining consistent performance across camera configurations. For more details, visit the project page at https://dfki-av.github.io/EgoForce.

CCS Concepts: • **Human-centered computing** → **Mixed / augmented reality**; • **Computing methodologies** → *Motion capture*; Neural networks.

Authors' Contact Information: Christen Millerdurai, Christen.Millerdurai@dfki.de, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern, Germany; Shaoxiang Wang, Shaoxiang.Wang@dfki.de, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern, Germany; Yaxu Xie, Yaxu.Xie@dfki.de, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern, Germany; Vladislav Golyanik, golyanik@mpi-inf.mpg.de, Max Planck Institute for Informatics (MPII), Saarbrücken, Germany; Didier Stricker, didier.stricker@dfki.de, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern, Germany; Alain Pagani, alain.pagani@dfki.de, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern, Germany.

SIGGRAPH Conference Papers '26, July 19–23, 2026, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811047

Additional Key Words and Phrases: egocentric hand pose estimation, hand-arm reconstruction, monocular RGB, egocentric RGB, computer vision

ACM Reference Format:
Christen Millerdurai, Shaoxiang Wang, Yaxu Xie, Vladislav Golyanik, Didier Stricker, and Alain Pagani. 2026EgoForce: Forearm-Guided Camera-Space 3D Hand Pose from a Monocular Egocentric Camera . In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3799902.3811047

1 Introduction

Following the shift from bulky head-mounted AR/VR systems toward compact wearable devices such as Project Aria (top-left of Fig. 1), devices increasingly rely on a lightweight perception stack built around a *single* egocentric RGB camera. Consequently, monocular egocentric 3D hand pose estimation becomes both essential and inherently challenging. This capability is fundamental for applications such as onsite teleoperation and surgical training, where accurate hand motion must be recovered in the headset's metric 3D frame from this single compact head-mounted camera.

Most existing monocular methods [Lin et al. 2021; Pavlakos et al. 2024; Potamias et al. 2024] estimate the 3D hand pose relative to a root joint (e.g., the wrist), recovering joint coordinates only up to an unknown translation and scale. While this simplifies supervision (since only relative annotations are required), it cannot provide the hand's absolute 3D position, which is essential for interaction-centric downstream tasks. In contrast, we aim to recover the full 6-DoF hand pose, *i.e.,* camera-aware hand translation and orientation, directly in the headset's metric coordinate frame, enabling plug-and-play integration with physics engines, grasp planners, collision-avoidance modules, and hand–object compositors without ad-hoc alignment, scale heuristics, or manual calibration. However, achieving this from a single egocentric camera is challenging due to depth–scale ambiguity, frequent self-occlusions, and the strong distortions introduced by wide-FOV and fisheye optics [Millerdurai et al. 2025, 2024a]. These challenges are compounded by the wide variety of camera configurations used in egocentric AR/VR setups (*e.g.,* perspective, fisheye, cylindrical or spherical), whose differing projection models make it difficult to train a single monocular system that generalizes across optics.

To address these challenges, we introduce EgoForce, a camera-space 3D hand-pose estimation framework that jointly leverages hand and forearm imagery to recover absolute 3D hand pose from a single egocentric camera (Fig. 2). Our first key insight is that the forearm provides strong metric cues that help resolve monocular depth-scale ambiguity: by modeling the anthropometric[1] coupling between forearm and hand (*e.g.,* ANSUR [Gordon et al. 1989] reports strong correlations between forearm length and overall arm size) and their coupled movement, EgoForce reduces depth–scale ambiguity far more reliably than hand-only methods. Our second key insight is a camera-model-agnostic ray space lifting formulation that operates on 2D joint observations formulated as rays rather than raw image coordinates, enabling a unified pipeline that generalizes across perspective, fisheye, and distorted wide-FOV optics. Together, these components enable robust absolute (camera-space) 3D hand reconstruction from a single egocentric camera. To realize these insights, we propose the following contributions:

(1) **H**and-**A**rm **L**atent-Shape & **O**rientation (HALO), a unified regression architecture that jointly regresses hand and forearm pose with their shape proxies from monocular images.
(2) A fully differentiable forearm representation that provides metric cues and improves absolute 3D hand-pose estimation through contextual arm–hand reasoning.
(3) A cross-camera ray space solver that recovers absolute 3D hand-forearm placement across fisheye, perspective, and distorted wide-FOV optics, enabling deployment across diverse head-mounted optics.

2 Related Works

2.1 3D Hand Pose Estimation

Monocular 3D hand pose estimation has progressed rapidly, but recovering *absolute* (camera-space) hand pose from a single RGB image remains challenging due to depth ambiguity along with lens and crop-induced distortions. Consequently, most methods [Lin et al. 2021; Pavlakos et al. 2024; Potamias et al. 2024] predict root-relative pose under weak perspective, discarding absolute hand position and often ignoring crop geometry effects [Prakash et al. 2024].

Prior approaches regressing 3D hand poses in the camera space fall into several categories. Single-stage regressors [Millerdurai et al. 2024b] provide an efficient end-to-end formulation, but often struggle to generalize across cameras. Root-depth regressors [Moon et al. 2019; Moon and Lee 2020] lift root-relative 3D poses to camera space by depth but rely on brittle anthropometric assumptions. Methods using known intrinsics [Iqbal et al. 2018; Mueller et al. 2018; Zhou et al. 2020] still require global scale estimation. Implicit neural formulations [Huang et al. 2023] regress camera-space joints via learned distance fields, but depend on accurate masks and manual tuning. Registration-based pipelines [Chen et al. 2022; Park et al. 2022; Valassakis and Garcia-Hernando 2024] decouple 2D detection and 3D lifting. However, many predict a root-relative hand and perform post-hoc iterative registration [Chen et al. 2022; Park et al. 2022], limiting end-to-end camera-space reasoning. HandDGP [Valassakis and Garcia-Hernando 2024] integrates differentiable registration but still relies on operating in a rectified or pinhole-style correspondence setting, making both correspondence learning and backpropagation through nonlinear projections fragile under extreme wide-FOV optics. Our approach follows the registration paradigm but differs from previous registration-based methods in two key ways: (1) Ray space alignment lifts the estimated 2D-3D correspondences directly in *ray space*, i.e., using bearing vectors from the *native calibrated projection model* (including fisheye/distorted wide-FOV), which removes dependence on a specific pixel coordinate system; while point-to-ray fitting is classic [Ansar and Daniilidis 2003; Pless 2003], our novelty is integrating it as a stable lifting module and validating across camera models in egocentric hand tracking; and (2) Our Crop Intrinsics Tokens (CIT) encode normalized crop intrinsics into

[1]We use *anthropometric* to refer to human body measurements and their statistical relationships (*e.g.,* correlations between limb-segment proportions).

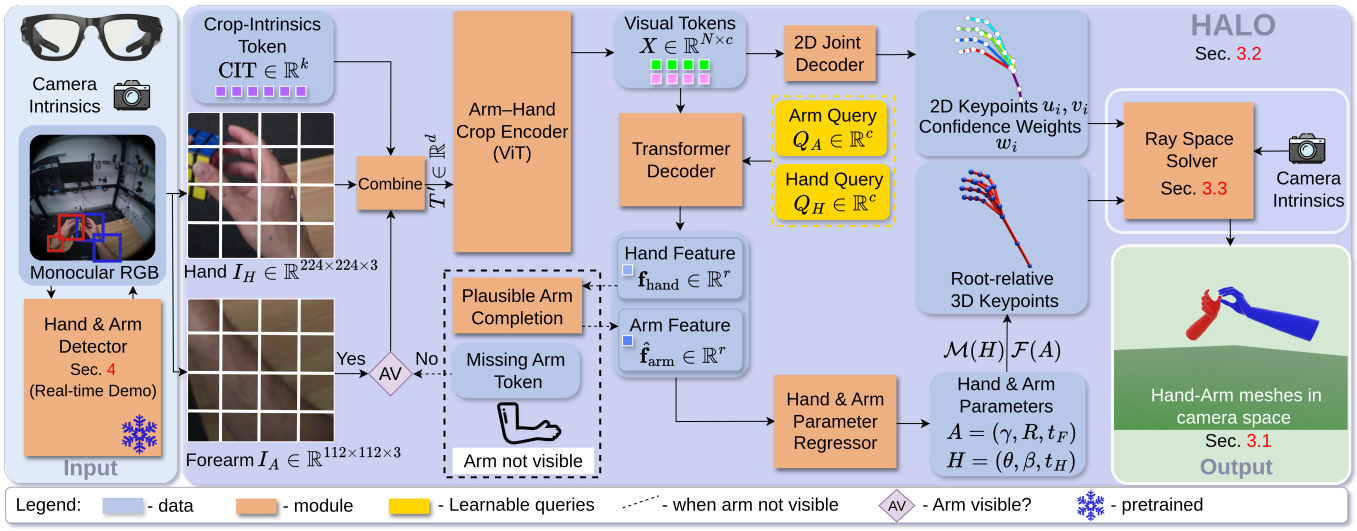


Fig. 2. EgoForce processes a monocular egocentric RGB frame by extracting hand and forearm crops, tokenizing them, and conditioning the features on crop intrinsics (CIT). A transformer jointly infers hand–arm features to predict 2D keypoints (with confidences) and root-relative 3D hand and arm poses, which are lifted to camera-space meshes via the ray space solver. When the forearm is out of view, arm tokens are replaced with missing-arm tokens, and a hand-conditioned variational prior infers a plausible arm representation. We apply this workflow independently to the left and right hand–forearm crops.

transformer tokens to enable consistent geometric reasoning across lenses and crop configurations. While multi-view methods such as UmeTrack [Han et al. 2022] target camera-space hand tracking for VR headsets, we instead focus on monocular hand reconstruction from a single camera, a setting better suited to lightweight smart glasses. Finally, some works estimate monocular world-space hand trajectories, but they rely on SLAM, explicit camera-motion estimation, multi-stage disentanglement frameworks [Yu et al. 2025; Zhang et al. 2025], or scale alignment [Ye et al. 2025], making them sensitive to drift and scale instabilities. In contrast, EgoForce predicts per-frame stable camera-space hand placements without SLAM, odometry, or manual scale calibration thanks to the ray space lifting module that directly solves point-to-ray constraints under the native camera model.

2.2 Hand-Forearm Context Reasoning

The forearm and hand are biomechanically coupled: forearm pose stabilizes orientation and provides a strong spatial prior that narrows the 3D space where the hand can plausibly be [Lee et al. 2024; Liu et al. 2022]. Moreover, forearm size covaries with body scale, providing soft anthropometric priors that help resolve monocular depth–scale ambiguity [Rostamzadeh et al. 2021; Vukotic et al. 2023]. Existing approaches exploit this coupling in different ways. Lee et al. [2024]; Tse et al. [2023] show that including the forearm region alongside the hand improves 3D hand pose estimation, either by regressing unified hand–forearm meshes or by leveraging forearm context for better hand accuracy. However, directly enlarging the hand crop to include the forearm can dilute high-frequency hand details required for precise joint localization (CNNs). Even with tokenization in transformers, careful design is needed to avoid spurious outputs when the forearm is not visible. To address these challenges, we propose a novel modular, crop-based framework that (1) processes hand and forearm regions separately to preserve fine-grained hand geometry; and (2) fuses them via cross-attention to capture the kinematic structure. This design preserves fine-grained hand detail while still leveraging forearm information to reduce depth–scale ambiguity. Moreover, when the forearm is not visible, a generative forearm prior infers a plausible arm orientation, maintaining continuity and realism in 3D hand–forearm motion. Hereafter, "arm" also refers to the forearm for conciseness.

3 The EgoForce Framework

Fig. 2 overviews the proposed EgoForce approach for recovering camera-space 3D hand and forearm meshes from a single egocentric RGB frame. We define the hand and forearm models in Sec. 3.1, describe our Hand-Arm Latent-Shape & Orientation (HALO) architecture in Sec. 3.2 that predicts root-relative meshes and 3D joints, lift these predictions to camera space using our Ray Space Solver in Sec. 3.3, and detail supervision in Sec. 3.4.

3.1 Preliminaries

Hand Model. We represent each hand using MANO [Romero et al. 2017], with pose $\theta \in \mathbb{R}^{16\times6}$ (6D rotations [Zhou et al. 2018]), shape $\beta \in \mathbb{R}^{10}$, and translation $t_H \in \mathbb{R}^3$. We denote hand parameters as $H = (\theta, \beta, t_H)$ and obtain mesh/joints via MANO operator $\mathcal{M}(H)$.
Forearm model (FARM). We introduce a novel formulation for each forearm using FARM (ForeArm Representation Model) with shape $\gamma \in \mathbb{R}^5$, rotation $R \in \mathbb{R}^6$, and translation $t_F \in \mathbb{R}^3$. We denote FARM parameters as $A = (\gamma, R, t_F)$ and obtain mesh/joints via FARM operator $\mathcal{F}(A)$. For details regarding the construction and parameterization of FARM, please refer to the Sup. Sec. 7.1.

Unified hand-arm mesh. We attach FARM to the MANO wrist by aligning FARM's wrist to the MANO wrist with a single translation in the MANO frame (see Sup. Fig. 14). To avoid interpenetration, we apply a small elbow-direction offset (about 3% of the elbow-to-wrist length) while preserving rotation, yielding a clean, anatomically consistent connection. This procedure yields clean, non-intersecting geometry and ensures that the hand and forearm remain rigidly anchored in camera space, enabling stable and physically coherent camera-space estimates.

3.2 **H**and-**A**rm **L**atent-Shape & **O**rientation Architecture

For each limb $\ell \in \{\text{left}, \text{right}\}$, HALO takes a hand crop $\mathbf{I}_H^\ell \in \mathbb{R}^{224\times224\times3}$ and a forearm crop $\mathbf{I}_A^\ell \in \mathbb{R}^{112\times112\times3}$ as input. Both crops are extracted from the original frame using their respective bounding boxes and then resized to the specified resolutions. HALO predicts: (i) 2D joints for hand and arm, (ii) per-joint confidences, (iii) MANO parameters, and (iv) FARM parameters.

Arm–Hand Crop Encoder. We remove lens-specific nonlinear distortions from each image crop using undistortion mapping and split them into N_H and N_A patches. These patches are linearly projected to d-dim tokens and augmented with positional encodings, yielding $\mathbf{T}_H \in \mathbb{R}^{N_H\times d}$ and $\mathbf{T}_A \in \mathbb{R}^{N_A\times d}$. We encode crop-specific intrinsics (normalized crop geometry and viewing parameters; described in Sup. Sec. 7.2) similar to Prakash et al. [2024] and produce *Crop Intrinsics Token* CIT $\in \mathbb{R}^k$. These CIT tokens are combined[2] with every patch token yielding $\mathbf{T}'_H \in \mathbb{R}^{N_H\times d}$ and $\mathbf{T}'_A \in \mathbb{R}^{N_A\times d}$. If the forearm is out of view, we replace $\mathbf{T}'_A$ with `[MASK]` tokens (i.e., missing-arm token). Finally, all the tokens are passed through a ViT backbone [Dosovitskiy et al. 2021] to obtain visual tokens of dimension c: $\mathbf{X} \in \mathbb{R}^{N\times c}$ where $N = N_H + N_A$.

Contextual Decoding of Hand–Arm Interactions. To extract the hand and arm features, we employ two sets of learnable queries—four hand queries (2D joints, global pose, hand shape, hand pose) and three arm queries (2D joints, arm shape, arm pose)—and, denote these query vectors collectively as $\mathbf{Q}_H \in \mathbb{R}^c$ and $\mathbf{Q}_A \in \mathbb{R}^c$, respectively. These queries are stacked to form the target sequence $\mathbf{Q}_0 = [\mathbf{Q}_H; \mathbf{Q}_A] \in \mathbb{R}^{2\times c}$, and decoded with a transformer decoder attending to $\mathbf{X} \in \mathbb{R}^{N\times c}$. After L layers (with $L = 2$ in practice), we obtain $\mathbf{Q}_L = [\mathbf{f}_{\text{hand}}; \mathbf{f}_{\text{arm}}] \in \mathbb{R}^{2\times r}$, where r is the decoded query dimension. The decoder's self-attention provides cross-limb context, letting the model leverage arm cues to resolve hand occlusions (and vice versa) during hand–object interaction. When the arm is not visible, updates to the arm query $\mathbf{Q}_\mathbf{A}$ are masked. Finally, we combine CIT with $\mathbf{f}_{\text{hand}}$ and $\mathbf{f}_{\text{arm}}$ to reinforce geometric conditioning.

Plausible Arm Completion. In egocentric views, the user's arm is frequently outside the camera's FOV, making direct visual localization challenging or even impossible. Although our primary focus is accurate hand tracking, inferring a plausible full arm pose can greatly benefit downstream applications (*e.g.,* AR immersion or physics simulation). We, therefore, introduce a conditional variational prior [Sohn et al. 2015] that models a latent arm code $\mathbf{z}_{\text{arm}}$ conditioned on hand features $\mathbf{f}_{\text{hand}}$. When the arm is not visible (i.e., no arm bounding box is available), we sample $\mathbf{z}_{\text{arm}}$ from this prior and obtain a plausible arm feature $\mathbf{f}_{\text{arm}}^{\text{prior}}$, which replaces the missing arm feature:

$$\hat{\mathbf{f}}_{\text{arm}} = \begin{cases} \mathbf{f}_{\text{arm}}, & \text{if the arm is visible,} \\ \mathbf{f}_{\text{arm}}^{\text{prior}}, & \text{otherwise.} \end{cases}$$

This leverages visual evidence when available and falls back to a learned hand-conditioned kinematic prior otherwise, yielding stable and realistic hand-arm configurations in egocentric scenarios.

2D Joint Decoder. We compute spatial attention between $\mathbf{X} \in \mathbb{R}^{N_H\times c}$ and the hand features to produce a hand-focused spatial map, and compute spatial attention between $\mathbf{X} \in \mathbb{R}^{N_A\times c}$ and the arm features to produce an arm-focused spatial map. We then decode heatmaps with a lightweight CNN as in ViTPose [Xu et al. 2022] for $J_H = 21$ hand joints and $J_A = 3$ forearm joints. Joint locations (u_j, v_j) are then obtained by soft-argmax over the heatmaps. Next, we bilinearly sample the spatial maps at these locations and use an MLP to predict per-joint confidence weights w_j as in Valassakis and Garcia-Hernando [2024]. These sampled features are also combined with the corresponding hand and arm features and passed to the parametric regressor.

Parametric Regressor. The final stage of HALO takes the hand and arm features and applies two parametric regressors [Kanazawa et al. 2018] to regress the hand $H = (\theta, \beta, t_H = \mathbf{0})$ and arm $A = (\gamma, R, t_F = \mathbf{0})$ parameters. The t_H and t_F are set to zero to obtain the root-relative parameters, and the camera-space positions are recovered using our Ray Space Solver.

3.3 Ray Space Solver (RSS)

After obtaining root-relative joints, detected 2D joints and confidence weights from HALO, we estimate a single camera-space translation $\mathbf{t} \in \mathbb{R}^3$ shared by the hand–arm mesh. For every 2D keypoint (u_i, v_i), we back-project it through the calibrated camera model to obtain a unit ray direction $\mathbf{d}_i$. The camera-space 3D joint $\mathbf{P}_i(\mathbf{t}) = \mathbf{t}+\mathbf{J}_i$ must lie on its corresponding ray, *i.e.,* there exists an (unknown) depth λ_i such that the point-on-ray constraint $\mathbf{t} + \mathbf{J}_i = \lambda_i \mathbf{d}_i$ holds. We eliminate the unknown depths λ_i by measuring only the component of each translated joint that is *perpendicular* to its ray, using the orthogonal projector $\mathbf{\Pi}_i = \mathbf{I} - \mathbf{d}_i\mathbf{d}_i^\top$ onto the plane normal to the ray, thereby removing the depth component. We estimate the shared translation by solving the confidence-weighted least-squares (closed-form) problem

$$\min_{\mathbf{t}} \; E(\mathbf{t}) = \sum_{i=1}^{M} w_i \left\| \mathbf{\Pi}_i \mathbf{P}_i(\mathbf{t}) \right\|_2^2, \tag{1}$$

over all joints $i = 1, \ldots, M$, where $\mathbf{\Pi}_i\mathbf{P}_i(\mathbf{t})$ is the depth-free point-to-ray residual and $\|\mathbf{\Pi}_i\mathbf{P}_i(\mathbf{t})\|_2$ equals the perpendicular (*i.e.,* shortest) distance from the camera-space joint $\mathbf{P}_i(\mathbf{t})$ to its ray (see Fig. 3). To prevent occasional unstable camera-space fits from corrupting root-relative learning, we stop gradients through the solver from flowing back into the root-relative predictions. Finally, we apply a Kalman filter to the per-frame translation estimates to improve stability under keypoint noise and occasional spurious solutions. Since ray directions can be computed for any camera projection model, our Ray Space Solver generalizes to arbitrary calibrated cameras. Full

[2] concatenate two vectors, reduce dimensionality via an MLP, then add residually to the original token; see Sup. Fig. 15.

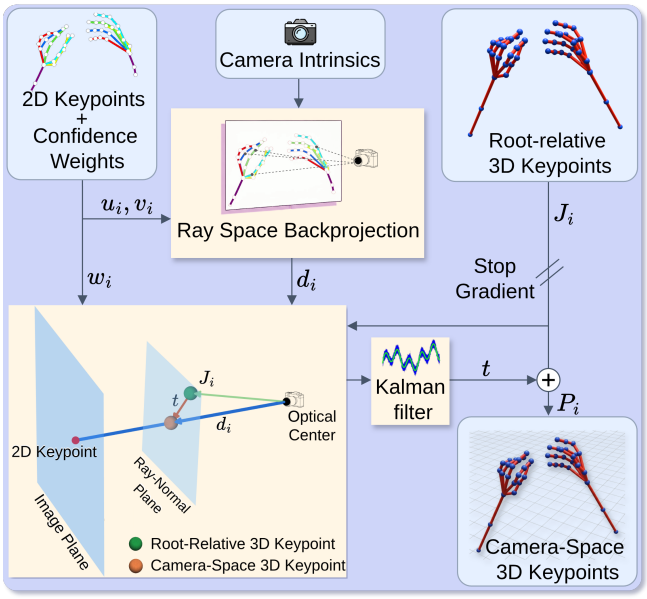


Fig. 3. **The Ray Space Solver** is a cross-camera (calibration-conditioned) module that recovers camera-space translation from 2D–3D correspondences, enabling deployment across devices with different optics.

derivation of the solver is provided in Sup. Sec. 7.3, and Kalman filter details and hyperparameters are reported in Sup. Sec. 7.3.1.

3.4 Loss Functions

2D Heatmap Loss. Squared error between the predicted and ground-truth 2D joint heatmaps for both hand and forearm:

$$\mathcal{L}_{\text{H}} = \lambda_H^{\mathcal{M}} \frac{1}{N_{\mathcal{M}}} \sum_{i=1}^{N_{\mathcal{M}}} \|H_i^{\mathcal{M}} - \hat{H}_i^{\mathcal{M}}\|^2 + \lambda_H^{\mathcal{F}} \frac{1}{N_{\mathcal{F}}} \sum_{i=1}^{N_{\mathcal{F}}} \|H_i^{\mathcal{F}} - \hat{H}_i^{\mathcal{F}}\|^2, \quad (2)$$

where $N_{\mathcal{M}}$ and $N_{\mathcal{F}}$ are the numbers of hand and forearm heatmaps, respectively; $H_i^{\mathcal{M}}, \hat{H}_i^{\mathcal{M}}$ are the predicted and ground-truth 2D joint heatmap for the hand; $H_i^{\mathcal{F}}, \hat{H}_i^{\mathcal{F}}$ are the predicted and ground-truth 2D joint heatmap for the arm; and we set $\lambda_H^{\mathcal{M}} = 20, \lambda_H^{\mathcal{F}} = 100$.

3D Joint Loss. Squared error between the predicted and ground-truth *root-relative* 3D joints:

$$\mathcal{L}_{\text{joints}} = \lambda_J^{\mathcal{M}} \frac{1}{N_{\mathcal{M}}} \sum_{i=1}^{N_{\mathcal{M}}} \|J_i^{\mathcal{M}} - \hat{J}_i^{\mathcal{M}}\|^2 + \lambda_J^{\mathcal{F}} \frac{1}{N_{\mathcal{F}}} \sum_{i=1}^{N_{\mathcal{F}}} \|J_i^{\mathcal{F}} - \hat{J}_i^{\mathcal{F}}\|^2, \quad (3)$$

where $J_i^{\mathcal{M}}, \hat{J}_i^{\mathcal{M}}$ are the predicted and ground-truth joint coordinates; $J_i^{\mathcal{F}}, \hat{J}_i^{\mathcal{F}}$ are the predicted and ground-truth forearm joint coordinates; and we set $\lambda_J^{\mathcal{M}} = 1$ and $\lambda_J^{\mathcal{F}} = 5$.

MANO And FARM Losses. The hand pose θ and shape β, are penalized via an ℓ_2 loss:

$$\mathcal{L}_{\text{MANO}} = \lambda_\theta \left\|\theta - \hat{\theta}\right\|^2 + \lambda_\beta \left\|\beta - \hat{\beta}\right\|^2, \quad (4)$$

where θ, β and $\hat{\theta}, \hat{\beta}$ are the predicted and ground-truth hand parameters, respectively, and we set $\lambda_\theta = 5,\ \lambda_\beta = 0.01$.

Similarly, the forearm shape coefficients γ and root rotation R are supervised with

$$\mathcal{L}_{\text{FARM}} = \lambda_\gamma \left\|\gamma - \hat{\gamma}\right\|^2 + \lambda_R \left\|R - \hat{R}\right\|^2, \quad (5)$$

where $\gamma, \hat{R}$ and $\hat{\gamma}, \hat{R}$ are the predicted and ground-truth forearm parameters, and we choose $\lambda_\gamma = 0.5,\ \lambda_R = 25$.

Hand–Arm Relative Orientation Loss. To enforce consistency between hand and arm orientations, we first compute their relative rotations $R_{\text{rel}} = R_{\text{hand}} R_{\text{arm}}^\top, \hat{R}_{\text{rel}} = \hat{R}_{\text{hand}} \hat{R}_{\text{arm}}^\top$ where are R the predicted rotations (converted from 6D representations) and $\hat{R}$ are the ground-truth rotations. We then measure the mean angular misalignment using the SO(3) geodesic distance, $d_A(R_1, R_2)$ [Mahendran et al. 2017] over the set of $\mathcal{V}$ frames where both the hand and the arm are present:

$$\mathcal{L}_{\text{rel}} = \lambda_{\text{rel}} \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left[d_A\left(R_{\text{rel},i},\ \hat{R}_{\text{rel},i}\right) \right]^2, \quad (6)$$

and set $\lambda_{\text{rel}} = 0.5$.

Forearm Prior Loss. When the VAE prior is used to infill an invisible forearm, we compute the KL-divergence between the predicted prior and a standard Gaussian distribution:

$$\mathcal{L}_{\text{prior}} = \lambda_{\text{prior}}\ \text{KL}\left(\mathcal{N}(\mu_{\text{prior}},\ \sigma_{\text{prior}}^2)\ \|\ \mathcal{N}(0, I)\right), \quad (7)$$

and we set $\lambda_{\text{prior}} = 1$.

Camera-Space 3D Joint Loss. Squared error between the predicted *camera-space* 3D joints and the ground-truth 3D joint:

$$\mathcal{L}_{\text{cs}} = \lambda_P^{\mathcal{M}} \frac{1}{N_{\mathcal{M}}} \sum_{i=1}^{N_{\mathcal{M}}} \|P_i^{\mathcal{M}} - \hat{P}_i^{\mathcal{M}}\|^2 + \lambda_P^{\mathcal{F}} \frac{1}{N_{\mathcal{F}}} \sum_{i=1}^{N_{\mathcal{F}}} \|P_i^{\mathcal{F}} - \hat{P}_i^{\mathcal{F}}\|^2, \quad (8)$$

where $P_i^{\mathcal{M}}, \hat{P}_i^{\mathcal{M}}$ are the predicted and ground-truth hand joint coordinates; $P_i^{\mathcal{F}}, \hat{P}_i^{\mathcal{F}}$ are the predicted and ground-truth forearm joint coordinates; and we set $\lambda_P^{\mathcal{M}} = 0.001$ and $\lambda_P^{\mathcal{F}} = 0.001$.

The Total Loss. Overall, our total loss reads:

$$\mathcal{L} = \mathcal{L}_{\text{H}} + \mathcal{L}_{\text{joints}} + \mathcal{L}_{\text{MANO}} + \mathcal{L}_{\text{FARM}} + \mathcal{L}_{\text{rel}} + \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{cs}}. \quad (9)$$

where the corresponding loss weights are included within each paragraph above. When FARM parameters are unavailable, we set $\mathcal{L}_{\text{FARM}}$ and $\mathcal{L}_{\text{rel}}$ to zero. In addition, hand and forearm keypoints that fall outside the image frame are masked out during supervision.

4 Experimental Evaluation

Training and Evaluation Datasets. We train on six datasets: Re:InterHand [Moon 2023], HandCO [Zimmermann et al. 2021], H2O [Kwon et al. 2021], ARCTIC [Fan et al. 2023], HO3D [Hampali et al. 2020], and HOT3D [Banerjee et al. 2024]. Re:InterHand, H2O, and ARCTIC include egocentric and exocentric views; HandCO and HO3D only exocentric; HOT3D only egocentric. In total, training data contain 3.67M RGB images with MANO parameters and 3D joints. We evaluate mainly on egocentric H2O, HOT3D, and ARCTIC, and additionally on HO3D. Additional details are in Supp. Sec. 8.1.

FARM Generation. Since the datasets contain only MANO annotations, we generate FARM parameters and 3D arm joints per dataset. For ARCTIC, we convert the provided SMPL-X meshes to FARM; for H2O, we triangulate multi-view imagery and fit FARM. FARM recovery is infeasible for HO3D and HOT3D due to monocular depth

ambiguity and frequent forearm occlusions, and for Re:InterHand and HandCO, the forearm is rarely visible, making FARM parameter recovery infeasible. Additional details on FARM parameter generation are provided in Supp. Sec. 8.3.

Evaluation Metrics. We report (i) camera-space mean joint error (CS-MJE) from Valassakis and Garcia-Hernando [2024], (ii) root-relative mean joint error (RS-MJE) from Grauman et al. [2024], (iii) Procrustes-aligned mean joint error (PS-MJE) from Pavlakos et al. [2024], and (iv) acceleration error (ACC) from Kocabas et al. [2020], reported as CS-ACC and RS-ACC. Detailed definitions are provided in the Sup. Sec. 8.4.

Evaluation Methodology. We train three camera-space baselines—MobRecon [Chen et al. 2022], HandOccNet [Park et al. 2022], and HandDGP [Valassakis and Garcia-Hernando 2024]—on our training data. As an additional baseline (HaMeR^{D}), we use the pretrained HaMeR model [Pavlakos et al. 2024] and lift its root-relative predictions to camera space with DepthAnythingV2 [Yang et al. 2024] and ground-truth intrinsics. We use the official implementations for MobRecon and HandOccNet and reimplement HandDGP (Sup. Tab. 5).

Implementation Details. We implement EgoForce in PyTorch [Paszke et al. 2019] and train with AdamW [Loshchilov and Hutter 2017], batch size 27, for 113 epochs. The transformer uses a learning rate of 1×10^{-5}; all other modules use 5×10^{-4}. Training on five NVIDIA H200 GPUs takes about four days. Further details are in Sup. Sec. 10; all baselines use official hyperparameters.

Live Demo and Runtime Performance. We demonstrate interactive performance in the supplementary video. Using the monocular fisheye stream from Aria glasses [Engel et al. 2023], we detect hand-arm crops with RTMDet [Lyu et al. 2022], regress camera-space hand and arm meshes with EgoForce, and stream them to Unity [2026] for live rendering. The full pipeline runs at ~14 FPS on an RTX 3090 for end-to-end two-hand tracking.

4.1 Results

Table 1 reports quantitative comparisons against HaMeR^{D}[Pavlakos et al. 2024], MobRecon [Chen et al. 2022], HandOccNet [Park et al. 2022], and HandDGP [Valassakis and Garcia-Hernando 2024].

ARCTIC. ARCTIC contains challenging egocentric scenarios with strong hand–object occlusions. HaMeR^{D} lifts 2D predictions to camera space using a monocular metric-depth estimator (DepthAnythingV2 [Yang et al. 2024]), but depth estimation in the near field—where egocentric hands typically lie—is unreliable, leading to large CS-MJE. Prior work [Zhang et al. 2025] reports similar degradation at extreme distances and proposes scene-based cues for improved stability, but such methods remain sensitive to motion blur, scene texture, illumination, and occlusion. Two-stage pipelines such as MobRecon and HandOccNet achieve competitive articulation accuracy (PS-MJE) but struggle with camera-space localization. HandDGP and EgoForce both use single-stage, feed-forward inference; HandDGP achieves good camera-space accuracy but weaker articulation, consistent with their report [Valassakis and Garcia-Hernando 2024]. EgoForce achieves state-of-the-art performance in both articulation and camera-space accuracy. As shown in Fig. 4, incorporating arm context improves robustness under occlusion, yielding an overall 3% improvement in RS-MJE and a 2.7% improvement in CS-MJE. Temporal stability also improves notably, with CS-ACC reduced by 22% and RS-ACC by 17%. The largest improvements in all metrics occur when hand-joint visibility is between 25-55% (≈5-12 visible joints), which commonly arises during hand-object manipulation.

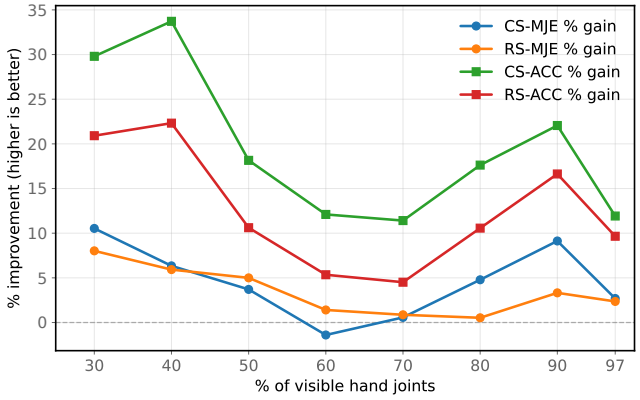


Fig. 4. **Influence of arm on hand-joint occlusion accuracy (ARCTIC dataset).** Adding the arm consistently improves hand pose (RS-MJE), camera-space accuracy (CS-MJE), and temporal stability (RS-ACC, CS-ACC).

HOT3D. HOT3D is challenging due to (1) large-range hand motion during object interaction and (2) severe fisheye distortion combined with wide-FOV imagery, which amplifies depth ambiguity. EgoForce achieves the highest accuracy across all methods, reducing CS-MJE by 28% relative to HandDGP. HandDGP relies on a pinhole-based 2D–3D correspondence formulation, making it sensitive to distortion near the image periphery, where fisheye effects dominate. As illustrated in Fig. 5, our method maintains accurate 3D reconstructions even when the hand moves toward the periphery, whereas HandDGP can deviate significantly despite plausible 2D projections. Furthermore, EgoForce also produces the most accurate camera-space trajectories (Sup. Fig. 16) for sequences, with both the start and end points closely matching the ground truth.

H2O. H2O contains highly dexterous hand–object interactions with pre-rectified (undistorted) images. We outperform all competing methods in CS-MJE and achieve strong results in PS-MJE. As shown in Tab. 2, it also attains the lowest CS-ACC, indicating the best temporal smoothness, and consistently recovers accurate hand orientation and articulation, as reflected by RS-MJE. As illustrated in Fig. 8, robust hand orientation estimates can be observed even under challenging object-interaction scenarios (e.g., holding a book).

HO3D. Although HO3D is captured from an external viewpoint rather than egocentric, we report results for completeness. Our method achieves the lowest CS-MJE, surpassing the previous state of the art, HandDGP. However, our PA-MJE (9.0 *mm*) is slightly higher than HaMeR^{D}'s 7.7 *mm*, likely due to their use of extensive in-the-wild 2D and diverse 3D training data, whereas our training is restricted to 3D-annotated datasets.

Comparison to Other Methods. EgoForce achieves lower CS-MJE on ARCTIC (55.1→49.5 *mm*) and lower PS-MJE on ARCTIC (14.7→8.0 *mm*), HOT3D (12.1→6.6 *mm*), and H2O (11.1→5.6 *mm*) against Han et al. [2022]. EgoForce also outperforms Zhang et al.

Table 1. **Quantitative results on ARCTIC, HOT3D, H2O, and HO3D (in mm).** Gold and bronze denote the best and second-best results, respectively. HO3D PS-MJE metrics of HandOccNet, HaMeR^{D}, and MobRecon are from their official papers; HO3D CS-MJE metrics of MobRecon, HandOccNet, and HandDGP are from Valassakis and Garcia-Hernando [2024].

Method	ARCTIC		HOT3D		H2O		HO3D	
	CS-MJE ↓	PS-MJE ↓	CS-MJE ↓	PS-MJE ↓	CS-MJE ↓	PS-MJE ↓	CS-MJE ↓	PS-MJE ↓
HaMeR^{D}	2067.3	9.2	4493.7	8.3	631.6	6.3	561.5	**7.7**
MobRecon	81.5	9.6	116.3	8.0	49.1	6.2	121.7	9.2
HandOccNet	256.3	**8.0**	284.8	**6.6**	62.1	**5.3**	156.4	9.1
HandDGP	51.7	9.9	61.3	8.6	29.9	6.3	50.3	9.3
EgoForce (Ours)	**49.5**	**8.0**	**43.9**	**6.6**	**25.0**	5.6	**49.5**	9.0

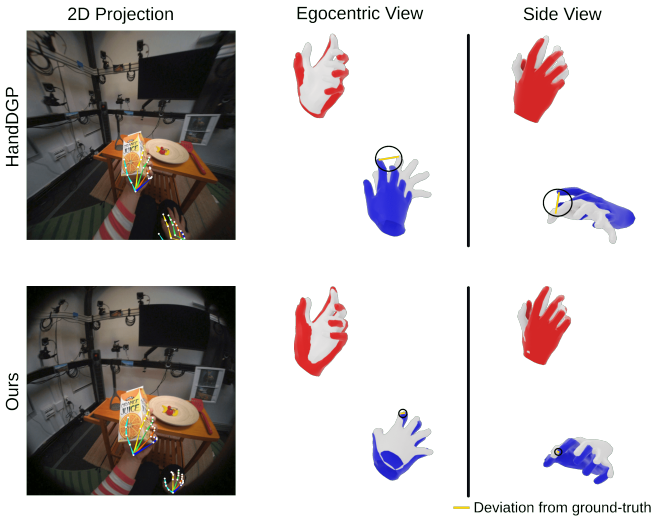


Fig. 5. **Camera-space results on HOT3D.** Left: egocentric input with the predicted 2D joint projections. Right: predicted meshes (left red, right blue) and ground-truth meshes (gray) in camera space.

Table 2. **Camera-space acceleration error (CS-ACC, in m/s^2) and root-relative hand pose error (RS-MJE, in mm) on the H2O dataset.**

Method	CS-ACC ↓	RS-MJE ↓
HaMeR^{D}	55.9	19.0
MobRecon	21.7	22.6
HandOccNet	11.7	17.9
HandDGP	8.5	17.3
EgoForce (Ours)	**5.5**	**14.8**

[2025], reducing CS-MJE from 319.9→49.5 *mm* on ARCTIC and 72.5→25.0 *mm* on H2O. See Sup. Sec. 9.1 for detailed comparisons.

4.2 Ablations

We isolate the key components of our method:

Camera geometry modeling. The ablation on HOT3D (Tab. 3) highlights the importance of accurate camera modeling under extreme fisheye optics. Undistortion alone provides the largest single gain (A→D), reducing CS-MJE from 123.4→48.7 *mm* (60.5% ↓) and RS-MJE from 53.1→19.7 *mm* (62.9% ↓). Introducing the Crop Intrinsics Token (A→B) without undistortion also helps (37.9% ↓ on CS-MJE), but combining undistortion and crop-specific intrinsic conditioning (D→E) yields the best results: 45.8 *mm* CS-MJE and 18.9 *mm* RS-MJE. In contrast, full-frame rectification (B→C) degrades performance due to the peripheral unwarping. Overall, explicit intrinsic modeling reduces CS-MJE by 62.9% and RS-MJE by 64.4% over the baseline, showing that camera geometry handling is crucial for accurate camera-space hand pose in fisheye imagery. Please refer to Sup. Fig. 11, Fig. 12, and the video for qualitative examples illustrating the impact of undistortion and CIT.

Table 3. **Ablation of Camera Geometry Modeling.** CS-MJE, RS-MJE, and PS-MJE on HOT3D (in mm). "CIT" = Crop Intrinsics Token; "Rect." = Rectification; "Un.D" = Undistortion.

Config.	CIT	Rect.	Un.D	CS-MJE ↓	RS-MJE ↓	PS-MJE ↓
(A)	✗	✗	✗	123.4	53.1	7.6
(B)	✓	✗	✗	76.6	29.0	6.8
(C)	✓	✓	✗	77.3	34.2	8.0
(D)	✗	✗	✓	48.7	19.7	6.6
(E)	✓	✗	✓	**45.8**	**18.9**	**6.6**

Incorporating arm context. We analyze the effect of arm context on ARCTIC for frames where the arm is visible and where it is not (Tab. 4). When the arm is visible, adding the arm crop improves hand performance, reducing CS-ACC from 19.3→15.2 m/s^2 (21.2% ↓) and RS-MJE from 18.5→18.0 *mm* (2.7% ↓). Arm accuracy also improves (RS-MJE 20.4→17.0 *mm*), although arm CS-ACC slightly increases (20.7→22.72 m/s^2). This suggests that while the model's prior favors smooth but mean arm configurations and providing true arm evidence triggers more expressive articulation at the cost of a small reduction in temporal smoothness. When the arm is not visible, introducing our hand-conditioned variational prior significantly improves arm estimates: RS-MJE drops from 28.7→12.8 *mm* (55.4% ↓) and CS-ACC from 20.6→18.4 m/s^2 (10.7% ↓), while hand performance remains unchanged. Overall, explicit arm conditioning benefits the hand when the arm is observed, and the learned arm prior is crucial for plausible arm recovery under occlusion, highlighting the importance of arm context for robust egocentric hand–arm pose estimation. Please refer to Fig. 6, Fig. 7, and the supplementary video for qualitative illustrations.

Table 4. **Ablation of Arm.** CS-ACC (in m/s^2) and RS-MJE (in mm) on the ARCTIC dataset."VP" = Variational Prior; "Inp." = Input Crop; "Vis. F" = Arm-visible frames; "Invis. F" = Arm-invisible frames.

	Method	Hand		Arm	
		CS-ACC ↓	RS-MJE ↓	CS-ACC ↓	RS-MJE ↓
Vis. F	w/o Arm Inp.	19.3	18.5	**20.7**	20.4
	w/ Arm Inp. (Ours)	**15.2**	**18.0**	22.7	**17.0**
Invis. F	w/o Arm VP	10.5	6.0	20.6	28.7
	w/ Arm VP (Ours)	**10.5**	**6.0**	**18.4**	**12.8**

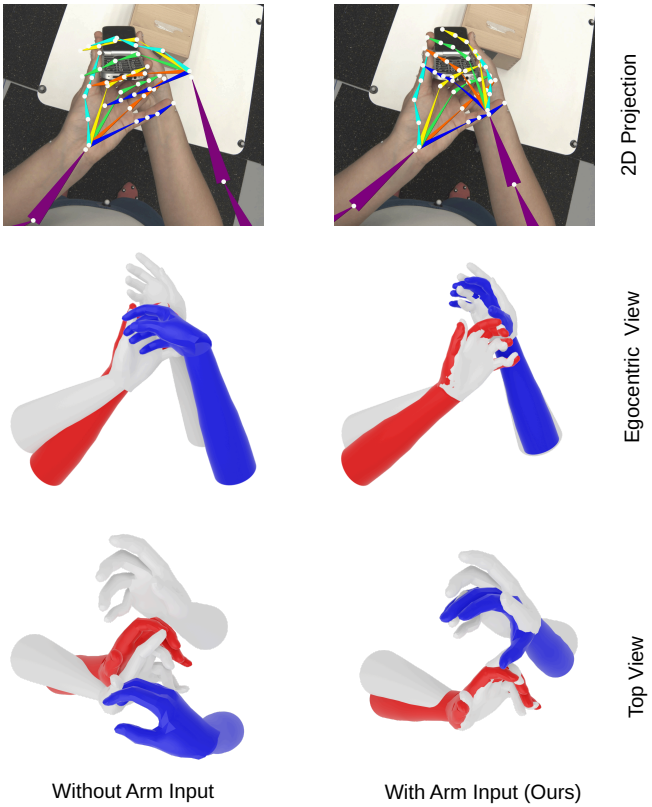


Fig. 6. **Influence of arm input.** Providing the arm crop as an input to the network improves hand pose accuracy. In this example, the right hand is strongly occluded by the phone and the other hand, yet the model recovers a plausible 3D pose, with accurate 2D joint reprojections and a hand-arm mesh closely aligned to ground truth.

Depth–scale mitigation and hand-scale stability. Tab. 8 quantitatively shows that forearm cues mitigate monocular depth–scale ambiguity. In particular, arm input reduces hand-scale error from 4.7 to 2.7 *mm* when the hand lies 200–300 *mm* from the camera (near field). In addition, frame-wise hand-scale variation for the sequences in datasets remains low, at 4 *mm* on HOT3D and 2 *mm* on ARCTIC, across 5 unseen hand sizes. See Sup. Sec. 9.1.

Calibration-mismatch robustness. As shown in Fig. 18, EgoForce remains stable under intrinsic errors. On HOT3D, CS-MJE

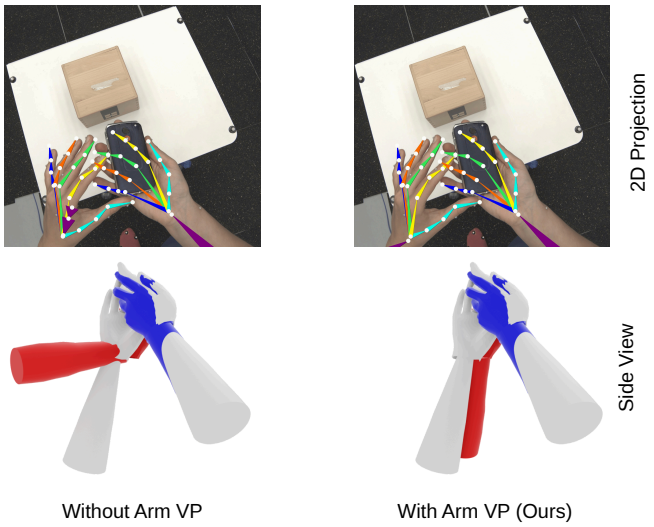


Fig. 7. **Influence of the variational arm prior.** Without the variational prior, the forearm is often mislocalized when it is heavily occluded. With the prior, the model infers a plausible forearm pose; in this example, the forearm is entirely out of view, yet the predicted position and orientation closely match ground truth.

improves from 43.9 to 39.3 *mm* at 50% intrinsic noise, despite a camera-geometry error of 25.3 *mm*, and degrades gracefully only under large mismatches (> 150%). See Sup. Sec. 9.1.

5 Limitations

Our method is not without limitations. It relies on calibrated 3D datasets for training, preventing the use of large 2D hand datasets common in root-relative methods [Pavlakos et al. 2024; Potamias et al. 2024] and limiting generalization to in-the-wild imagery. It also remains sensitive to camera intrinsics (See Fig. 18 and Tab. 9 of Supplement). Extended discussion of limitations are in Sup. Sec. 11.

6 Conclusion

We introduce EgoForce, a monocular egocentric method for absolute camera-space 3D hand pose that leverages forearm context and camera-model–aware ray-space lifting. Across three egocentric benchmarks, it delivers higher camera-space accuracy and stable temporal predictions, even with occlusions caused by hand–hand and hand–object interactions, and remains effective across both perspective and fisheye optics. Ablations show that wide-FOV tracking benefits strongly from *explicit camera geometry*: modeling distortion and conditioning on crop-aware intrinsics consistently improve performance. We hope this work motivates future egocentric hand tracking systems to integrate forearm context and ray-based geometric constraints, especially as AR/VR hardware continues to shift toward compact, wide-FOV wearable cameras.

ACKNOWLEDGMENTS

This work was partially funded by the Horizon Europe programme under the projects dAIEDGE, Grant Agreement No. 101120726, and IRIS-XR, Grant Agreement No. 101298672. The authors thank the anonymous reviewers for their valuable feedback.

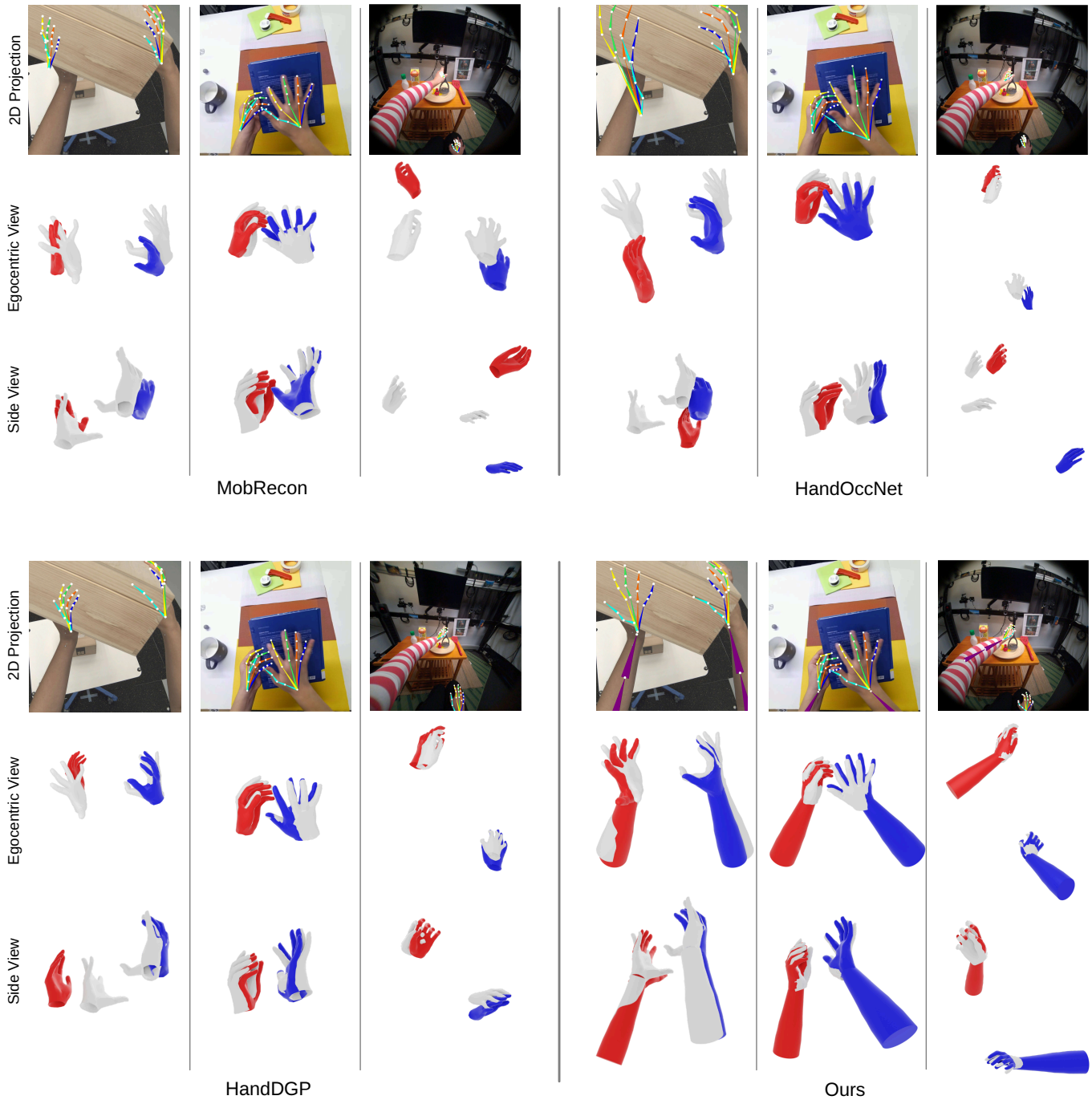


Fig. 8. **Qualitative camera-space results on egocentric datasets.** We compare our method against three state-of-the-art camera-space 3D hand pose methods on three datasets with widely different camera intrinsics. Predicted left and right limb meshes are shown in red and blue, respectively, with ground truth highlighted in gray.

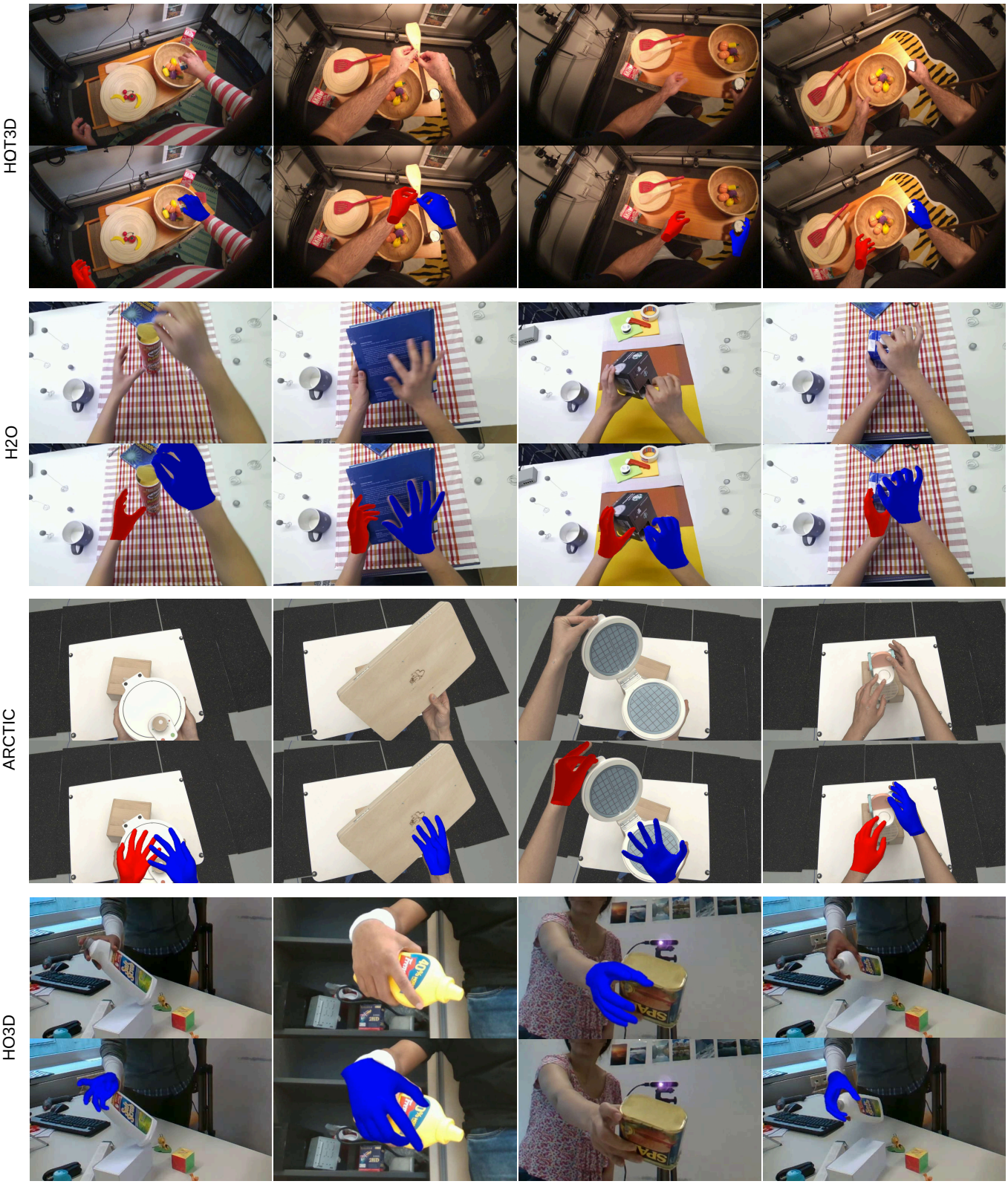


Fig. 9. **Camera-space hand mesh projections on egocentric datasets.** We project predicted hand meshes onto images from three camera types: HOT3D (fisheye), H2O/HO3D (perspective), and ARCTIC (distorted perspective). Our method maintains accurate projections under challenging conditions such as motion blur (H2O) and hand-object occlusions (HOT3D, HO3D, ARCTIC).

References

Adnan Ansar and Konstantinos Daniilidis. 2003. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 5 (2003), 578–589.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. 2024. Introducing HOT3D: An Egocentric Dataset for 3D Hand and Object Tracking. *arXiv preprint arXiv:2406.09598* (2024).

Feng Chen, Ling Ding, Kanokphan Lertniphonphan, Jian Li, Kaer Huang, and Zhepeng Wang. 2024. PCIE_EgoHandPose Solution for EgoExo4D Hand Pose Challenge. *arXiv preprint arXiv:2406.12219* (2024).

Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. 2021. Camera-Space Hand Mesh Recovery via Semantic Aggregationand Adaptive 2D-1D Registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xingyu Chen, Yufeng Liu, Dong Yajiao, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. 2022. MobRecon: Mobile-Friendly Hand Mesh Reconstruction from Monocular Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*.

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De Nardi, and Richard Newcombe. 2023. Project Aria: A New Tool for Egocentric Multi-Modal AI Research. arXiv:2308.13561 [cs.HC] https://arxiv.org/abs/2308.13561

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. 2023. ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Claire C. Gordon, T. E. Churchill, C. E. Clauser, and B. Bradtmiller. 1989. *1988 Anthropometric Survey of U.S. Army Personnel: Summary Statistics*. Technical Report Natick/TR-89/044. U.S. Army Natick Research, Development and Engineering Center.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. 2024. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Computer Vision and Pattern Recognition (CVPR)*.

Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. 2020. HOnnotate: A method for 3D Annotation of Hand and Object Poses. In *Computer Vision and Pattern Recognition (CVPR)*.

Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. 2022. UmeTrack: Unified multi-view end-to-end hand tracking for VR. In *SIGGRAPH Asia 2022 conference papers*. 1–9.

Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.

Lin Huang, Chung-Ching Lin, Kevin Lin, Lin Liang, Lijuan Wang, Junsong Yuan, and Zicheng Liu. 2023. Neural voting field for camera-space 3D hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8969–8978.

Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. 2018. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European conference on computer vision (ECCV)*. 118–134.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7122–7131.

Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. 2019. Learning 3d human dynamics from video. In *Computer Vision and Pattern Recognition (CVPR)*.

Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. 2020. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5253–5263.

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. 2021. H2O: Two Hands Manipulating Objects for First Person Interaction Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 10138–10148.

Jeongho Lee, Jaeyun Kim, Seon Ho Kim, and Sang-Il Choi. 2024. Enhancing 3D hand pose estimation using SHaF: synthetic hand dataset including a forearm. *Applied Intelligence* 54, 20 (2024), 9565–9578.

Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021. End-to-End Human Pose and Mesh Reconstruction with Transformers. In *CVPR*.

Shuying Liu, Wenbin Wu, Jiaxian Wu, and Yue Lin. 2022. Spatial-temporal parallel transformer for arm-hand dynamic estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20523–20532.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. 2022. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. arXiv:2212.07784 [cs.CV] https://arxiv.org/abs/2212.07784

Siddharth Mahendran, Haider Ali, and René Vidal. 2017. 3d pose regression using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision workshops*. 2174–2182.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

Christen Millerdurai, Hiroyasu Akada, Jian Wang, Diogo Luvizon, Alain Pagani, Didier Stricker, Christian Theobalt, and Vladislav Golyanik. 2025. EventEgo3D++: 3D Human Motion Capture from A Head-Mounted Event Camera. *International Journal of Computer Vision (IJCV)* (11 Jun 2025). https://doi.org/10.1007/s11263-025-02489-1

Christen Millerdurai, Hiroyasu Akada, Jian Wang, Diogo Luvizon, Christian Theobalt, and Vladislav Golyanik. 2024a. EventEgo3D: 3D Human Motion Capture from Egocentric Event Streams. In *Computer Vision and Pattern Recognition (CVPR)*.

Christen Millerdurai, Diogo Luvizon, Viktor Rudnev, André Jonas, Jiayi Wang, Christian Theobalt, and Vladislav Golyanik. 2024b. 3D Pose Estimation of Two Interacting Hands from a Monocular Event Camera. In *International Conference on 3D Vision (3DV)*.

Gyeongsik Moon. 2023. Bringing Inputs to Shared Domains for 3D Interacting Hands Recovery in the Wild. In *CVPR*.

Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. 2019. Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. In *The IEEE Conference on International Conference on Computer Vision (ICCV)*.

Gyeongsik Moon and Kyoung Mu Lee. 2020. I2L-MeshNet: Image-to-Lixel Prediction Network for Accurate 3D Human Pose and Mesh Estimation from a Single RGB Image. In *European Conference on Computer Vision (ECCV)*.

Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*. 11 pages. https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/

JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. 2022. HandOccNet: Occlusion-Robust 3D Hand Mesh Estimation Network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. 2024. Reconstructing Hands in 3D with Transformers. In *CVPR*.

Robert Pless. 2003. Using many cameras as one. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Vol. 2. IEEE, II–587.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. 2024. WiLoR: End-to-end 3D Hand Localization and Reconstruction in-the-wild. arXiv:2409.12259 [cs.CV]

Aditya Prakash, Ruisen Tu, Matthew Chang, and Saurabh Gupta. 2024. 3D hand pose estimation in everyday egocentric images. In *European Conference on Computer Vision*. Springer, 183–202.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

Sajjad Rostamzadeh, Mahnaz Saremi, Shahram Vosoughi, Bruce Bradtmiller, Leila Janani, Ali Asghar Farshad, and Fereshteh Taheri. 2021. Analysis of hand-forearm anthropometric components in assessing handgrip and pinch strengths of school-aged children and adolescents: a partial least squares (PLS) approach. *BMC pediatrics* 21, 1 (2021), 39.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems (NeurIPS)* (2015).

Javier Tirado-Garín and Javier Civera. 2025. AnyCalib: On-manifold learning for model-agnostic single-view camera calibration. In *Computer Vision and Pattern Recognition (CVPR)*.

Tze Ho Elden Tse, Franziska Mueller, Zhengyang Shen, Danhang Tang, Thabo Beeler, Mingsong Dou, Yinda Zhang, Sasa Petrovic, Hyung Jin Chang, Jonathan Taylor, et al. 2023. Spectral graphormer: Spectral graph-based transformer for egocentric two-hand reconstruction using multi-view color images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14666–14677.

Unity. 2026. Unity Real-Time Development Platform. https://unity.com/. Accessed: 2026-01-07.

Eugene Valassakis and Guillermo Garcia-Hernando. 2024. HandDGP: Camera-Space Hand Mesh Prediction with Differentiable Global Positioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)* (2017).

Marina Vukotic, Desanka Radojicic, and Djurdja Buric. 2023. Nationwide stature estimation from forearm length measurements in Montenegrin adolescents. *Int. j. morphol* 41, 3 (2023), 764–768.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. 2022. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth anything v2. *Advances in Neural Information Processing Systems (NeurIPS)* (2024).

Yufei Ye, Yao Feng, Omid Taheri, Haiwen Feng, Shubham Tulsiani, and Michael J Black. 2025. Predicting 4d hand trajectory from monocular videos. *arXiv preprint arXiv:2501.08329* (2025).

Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. 2025. Dyn-hamr: Recovering 4d interacting hand motion from a dynamic camera. In *Computer Vision and Pattern Recognition (CVPR)*.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. 2025. HaWoR: World-Space Hand Motion Reconstruction from Egocentric Videos. *arXiv preprint arXiv:2501.02973* (2025).

Yi Zhou, C Barnes, J Lu, J Yang, and H Li. 2018. On the continuity of rotation representations in neural networks. 2019 IEEE. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 3.

Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. 2020. Monocular Real-Time Hand Shape and Motion Capture Using Multi-Modal Data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Christian Zimmermann, Max Argus, and Thomas Brox. 2021. Contrastive representation learning for hand shape estimation. In *German Conference on Pattern Recognition (DAGM)*.

Supplementary Material

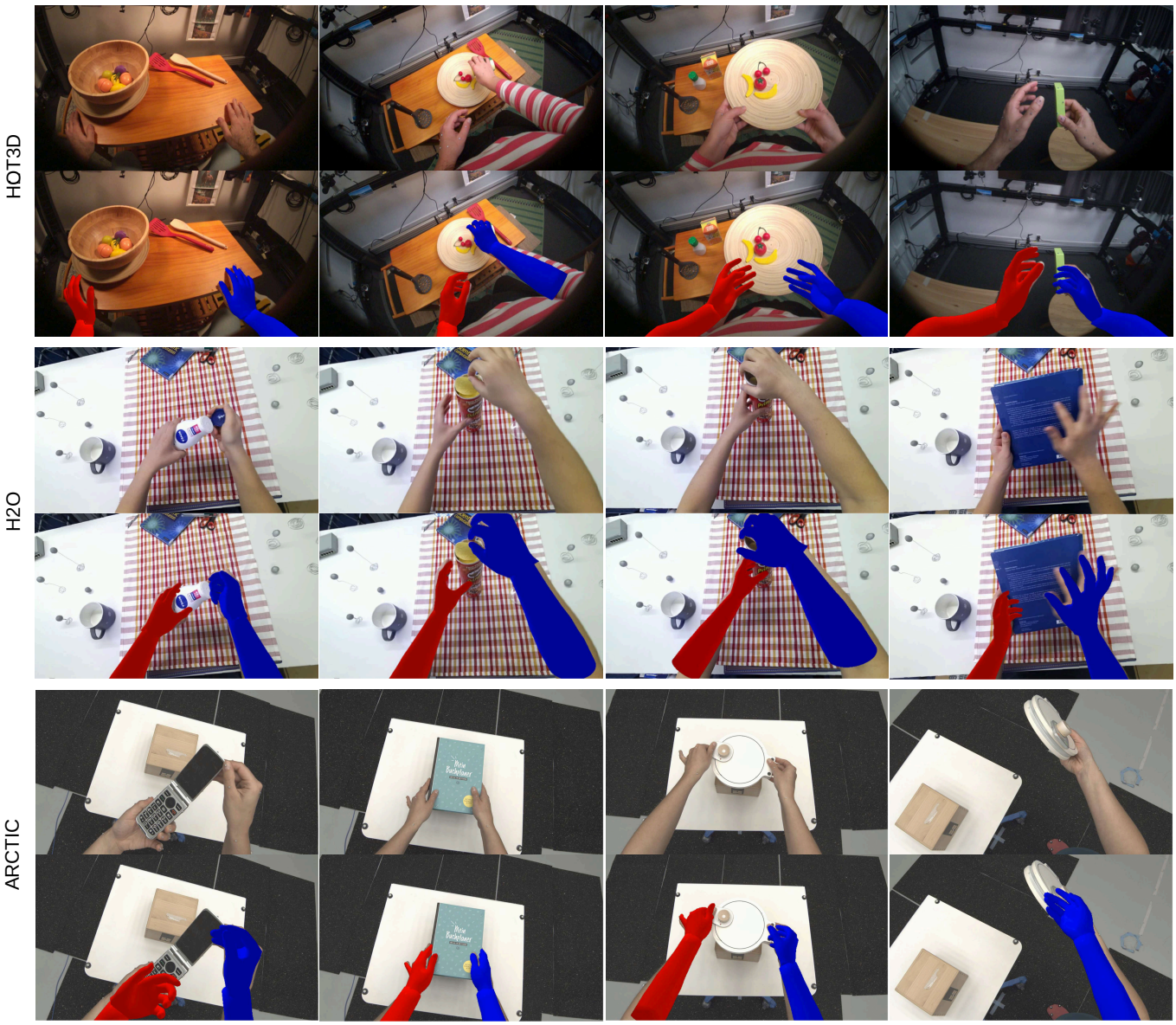


Fig. 10. **Camera-space hand-arm mesh projections on egocentric datasets.** We project predicted hand and arm meshes onto images from three camera types: HOT3D (fisheye), H2O (perspective), and ARCTIC (distorted perspective). Our method maintains accurate projections under challenging conditions such as motion blur (H2O) and hand-object occlusions (HOT3D, ARCTIC).

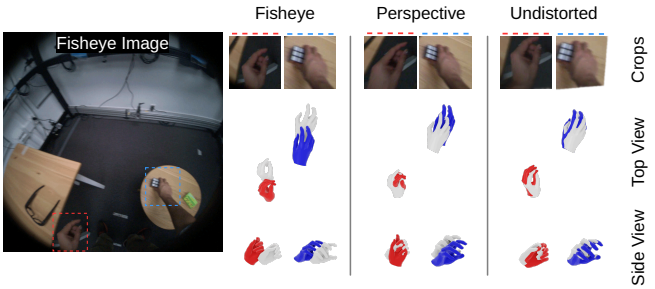


Fig. 11. **Influence of undistortion on input crops.** Direct hand-arm crops from the raw fisheye image lead to large errors as fisheye pixels correspond to highly non-linear viewing rays. Rectifying the full frame to a single perspective view reduces distortion but introduces strong peripheral warping and resampling artifacts that amplify localization noise. In contrast, lens-model undistortion preserves the correct pixel-to-ray geometry, yielding the most accurate camera-space reconstruction, especially near the image periphery.

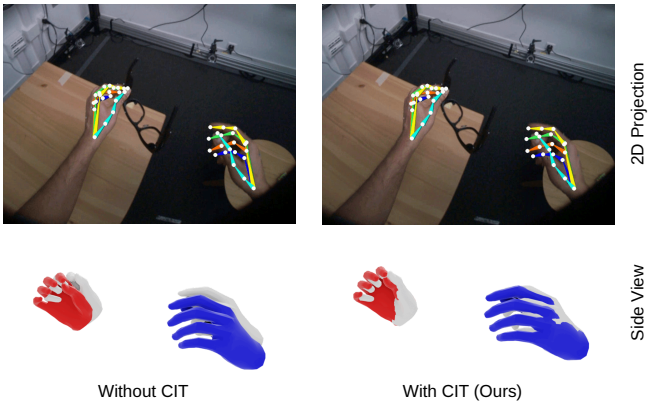


Fig. 12. **Influence of Crop Intrinsics Tokens (CIT).** CIT encodes crop-specific intrinsics as tokens for the hand-arm crop inputs feed to the transformer. This enables explicit local camera-geometry reasoning and reduces camera-space mesh error, leading to closer alignment with ground truth.

7 Additional Details about our Framework

7.1 ForeArm Representation Model

The ForeArm Representation Model (FARM) is a lightweight, fully differentiable, parameterized mesh generator that maps a low-dimensional parameter vector to a watertight triangular mesh $(\mathbf{V}, \mathbb{F})$ suitable for modeling individual limb segments. Geometrically, FARM approximates the limb as a truncated cone and defines three anatomically meaningful 3D joints along its length. In the forearm configuration, these joints correspond to the elbow, mid-forearm, and wrist.

7.1.1 Construction. FARM constructs its mesh by sampling vertices on a regular angular–height lattice:

$$\theta_i = \frac{2\pi\, i}{n_\theta}, \quad z_j = \frac{j}{n_z - 1}\, h,$$

$$i = 0, \ldots, n_\theta - 1, \quad j = 0, \ldots, n_z - 1,$$

where $n_\theta, n_z \in \mathbb{N}$ are the numbers of angular and height subdivisions (e.g. $n_\theta = 50$, $n_z = 12$ in our implementation).

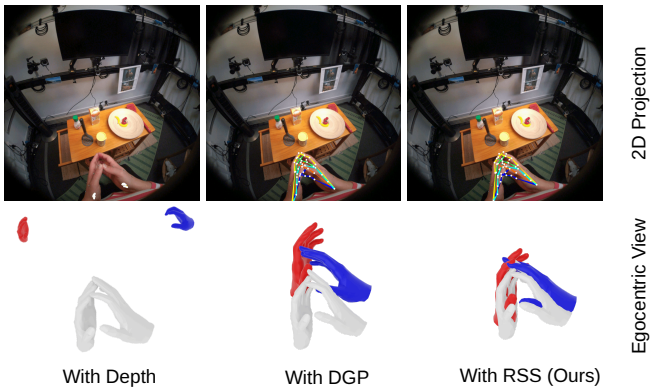


Fig. 13. **Camera-space lifting techniques.** Lifting via monocular depth estimators is brittle in the near field of the camera, where small depth errors lead to large camera-space misplacements. HandDGP's DGP module can yield plausible 2D projections while placing the 3D hand mesh far from the ground truth. In contrast, our RSS produces both accurate 2D projections and camera-space mesh placements that closely match ground truth.

At each height level z_j, the radius is given by

$$r_j \;=\; r_1 + (r_2 - r_1)\,\frac{j}{n_z - 1} + \rho_j, \qquad \boldsymbol{\rho} = (\rho_0, \ldots, \rho_{n_z - 1})^\top,$$

so that the vector $\boldsymbol{\rho}$ serves as a learned radial offset profile, allowing fine sculpting of the limb's cross-section.

We collect the (x, y, z) coordinates as

$$v_{i,j} = \begin{bmatrix} r_j \cos\theta_i \\ r_j \sin\theta_i \\ z_j \end{bmatrix} \in \mathbb{R}^3, \quad V = \{v_{i,j}\}_{\substack{i=0,\ldots,n_\theta - 1 \\ j=0,\ldots,n_z - 1}}.$$

Two additional vertices v_{bottom} and v_{top} cap the ends, and one midpoint vertex v_{mid} is placed at $z = \frac{1}{2}h$. The face set $\mathcal{F}$ comprises $2n_\theta(n_z - 1)$ quadrilaterals—each divided into two triangles—and n_θ triangular faces on each end cap.

7.1.2 Pose. Let $\Omega \in \mathrm{SO}(3)$ be an arbitrary rotation, which we decompose into a *swirl* component Ω_s and a *twist* component Ω_t:

$$\Omega \;=\; \underbrace{\Omega_s}_{\text{swirl}}\;\underbrace{\Omega_t}_{\text{twist}}.$$

Since the truncated-cone geometry is radially symmetric, forearm pronation (twist around the longitudinal axis) cannot be determined from the mesh. We therefore discard the twist component and retain only the swirl:

$$\Omega_s = \Omega\, \Omega_t^{-1}.$$

To apply the rigid transform to each vertex v_k, we first recenter it about the midpoint

$$\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2}h \end{bmatrix},$$

then rotate and translate:

$$\tilde{\mathbf{v}}_{i,j} = \Omega_s\,(\mathbf{v}_{i,j} - \mathbf{c}) + \mathbf{c} + \mathbf{t},$$

where $\mathbf{t} \in \mathbb{R}^3$ is the global translation of the FARM mesh. The same transformation is applied to the three joint centres (elbow,

mid-forearm, and wrist), making them suitable for extracting FARM parameters from existing datasets.

7.1.3 Low-Dimensional Shape Space. The FARM shape parameters are highly expressive, which can make optimization—when relying solely on sparse 3D joints and segmentation masks—ill-posed. To address this, we impose a geometric prior by learning a PCA model over the parameter vector

$$\mathbf{s} = \begin{bmatrix} r_1 & r_2 & h & \rho_0 & \dots & \rho_{n_z-1} \end{bmatrix}^\top \in \mathbb{R}^{3+n_z},$$

Specifically, we introduce a low-dimensional latent code $\mathbf{p} \in \mathbb{R}^d$ (with $d = 5$) and decode it linearly:

$$\mathbf{s} = \mathbf{W}\,\mathbf{p} + \mathbf{b},$$

where $\mathbf{W} \in \mathbb{R}^{(3+n_z)\times d}$ is the PCA loading matrix and $\mathbf{b} \in \mathbb{R}^{3+n_z}$ is the mean. Both $\mathbf{W}$ and $\mathbf{b}$ are learned from data and then frozen to regularize the forearm shape toward plausible geometries.

PCA Space Training. The PCA loading matrix $\mathbf{W}$ and mean vector $\mathbf{b}$ are computed from forearm parameter vectors extracted from the AMASS motion-capture repository [Mahmood et al. 2019]. Specifically, we sample 2806 SMPL body meshes—covering 344 unique subjects performing a wide variety of motions—and isolate the corresponding forearm parameters s. We then apply principal component analysis to this collection and retain the top $d = 5$ components, which together capture approximately 99% of the variance in forearm shape, to form $\mathbf{W}$ and $\mathbf{b}$.

SMPL to FARM Fitting. For each SMPL sample, we perform the following steps:

(1) **Extract forearm vertex set.** Select the SMPL vertices whose 3D coordinates lie between the anatomical elbow and wrist joint centers. Denote this set by

$$V = \{\, v_i \mid v_i \text{ lies between elbow and wrist} \}.$$

(2) **Estimate initial parameters.** From V, extract the two boundary rings

$$V_{\text{elbow}} = \{\, v_i \mid i \in I_{\text{elbow}} \}, \quad V_{\text{wrist}} = \{\, v_i \mid i \in I_{\text{wrist}} \},$$

and let

$$V_{\text{ring}} = V_{\text{elbow}} \cup V_{\text{wrist}}.$$

Perform PCA on these rings to estimate

$$h, \quad r_1, \quad r_2$$

where h is the forearm length, r_1 the elbow radius, and r_2 the wrist radius.

(3) **Instantiate FARM.** Generate the FARM mesh with ($n_\theta = 50$, $n_z = 10$) using the initial shape parameters $\{r_1, r_2, h\}$ and setting $\boldsymbol{\rho} = \mathbf{0}$. Extract the elbow and wrist boundary indices $\hat{I}_{\text{elbow}}, \hat{I}_{\text{wrist}}$ and form the ring set

$$\hat{V}_k = \left\{ \hat{v}_i \mid i \in \hat{I}_{\text{elbow}} \cup \hat{I}_{\text{wrist}} \right\}.$$

(4) **Pose optimization.** Keeping the shape parameters $\mathbf{s}$ fixed, optimize only the global rotation $\boldsymbol{\Omega}$ and translation $\mathbf{t}$ by minimizing

$$\mathcal{L}_{\text{pose}} = \lambda_k\, d_{\text{Chamfer}}\big(\hat{V}_k,\, V_k\big) + \lambda_v\, d_{\text{Chamfer}}\big(\hat{V},\, V\big),$$

with $\lambda_k = 100$ and $\lambda_v = 10$. We employ the Adam optimizer (learning rate 0.1), a ReduceLROnPlateau scheduler (factor 0.9, patience 10), and early stopping (patience 100, $\Delta_{\min} = 10^{-6}$). Optimization terminates when the early-stopping criterion is met, yielding $\boldsymbol{\Omega}^*$ and $\mathbf{t}^*$.

(5) **Shape optimization.** We freeze the optimal pose $\boldsymbol{\Omega}^*, \mathbf{t}^*$ and optimize the shape parameters $\mathbf{s} = [r_1,\, r_2,\, h,\, \boldsymbol{\rho}]^\top$ by minimizing

$$\begin{aligned}\mathcal{L}_{\text{shape}} = {} & \alpha_k\, d_{\text{Chamfer}}\big(\hat{V}_k,\, V_k\big) \\ & + \alpha_v\, d_{\text{Chamfer}}\big(\hat{V},\, V\big) \\ & + \alpha_{\text{vol}} \left| \text{Vol}(r_1, r_2, h, \boldsymbol{\rho}) - V_{\text{mesh}} \right| \\ & + \alpha_{\Delta r} \left| r_2 - r_1 \right| + \alpha_1 \left| r_1 - r_1^{(0)} \right| + \alpha_2 \left| r_2 - r_2^{(0)} \right|,\end{aligned}$$

where the FARM volume

$$\text{Vol}(r_1, r_2, h, \boldsymbol{\rho}) = \frac{\pi}{3} \sum_{j=0}^{n_z-2} \Delta z \left(r_j^2 + r_j\, r_{j+1} + r_{j+1}^2 \right),$$

$$\Delta z = \frac{h}{n_z - 1},$$

is computed as the sum of frusta and the weights are set to $\alpha_k = 10$, $\alpha_v = 1$, $\alpha_{\text{vol}} = 1$, $\alpha_{\Delta r} = -0.1$, $\alpha_{1,2} = 0.01$. We employ the Adam optimizer with learning rates $\{r_1,\, r_2,\, h\}$: 0.001, $\boldsymbol{\rho}$: 0.01, together with the same ReduceLROnPlateau scheduler and early stopping as in pose fitting. Optimization terminates when the early-stopping criterion is met, yielding the final parameters $\{r_1^*,\, r_2^*,\, h^*,\, \boldsymbol{\rho}^*\}$.

(6) **Optimized parameters.** After completing both pose and shape optimization, the final set of FARM parameters is

$$\left\{ r_1^*,\; r_2^*,\; h^*,\; \boldsymbol{\rho}^*,\; \boldsymbol{\Omega}^*,\; \mathbf{t}^* \right\}.$$

7.1.4 Discussion. While FARM offers a compact, differentiable forearm representation, it makes several simplifying assumptions that may limit its fidelity. By modeling the radius and ulna as a single rigid segment, FARM cannot capture true pronation and supination—motions of up to ±90°—and discards all axial twist. Its PCA shape prior is learned from adult SMPL meshes (AMASS), which may not generalize to children, individuals with atypical anatomy, or amputees. To mitigate these limitations, one can: (1) place off-axis markers on the forearm surface to break axial symmetry and make twist observable; or (2) incorporate inexpensive inertial measurement units (IMUs) or EMG straps on the limb to directly measure axial rotation.

7.2 Crop Intrinsics Token

To encode the geometric context of each cropped image patch relative to its camera, we build upon the *Keypoint Encodings* (KPE) of Prakash *et al.* [Prakash et al. 2024] and extend them with a set of camera-independent distribution parameters. Concretely, for each crop we compute the normalized principal-point offset, the crop ratios, and the half-field-of-view angles, and then concatenate these quantities with the original KPE vectors to form our Crop Intrinsics Tokens (CIT). By conditioning the framework on CIT, we enable training on heterogeneous datasets—from wide-FOV egocentric videos to narrow-FOV third-person captures—while remaining robust to camera-specific variations, thereby improving cross-domain generalization..

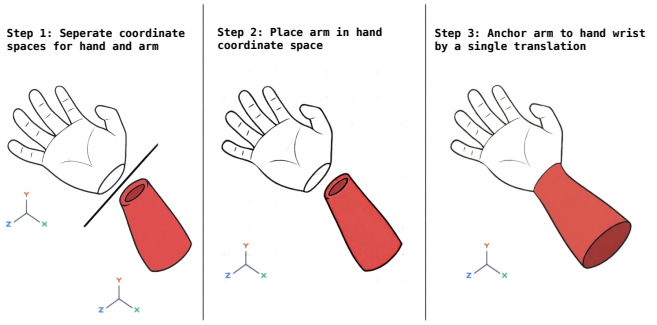


Fig. 14. **Unified hand–arm mesh.** We attach the FARM at the MANO wrist and apply a small elbow-direction offset to avoid overlap and ensure a clean, anatomically consistent connection.

7.2.1 Crop Intrinsics & Distortion Correction. Let the camera intrinsics be

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix},$$

where f_x, f_y are the focal lengths (in pixels) and (c_x, c_y) is the principal point. Let

$$d = (d_1, \ldots, d_m)$$

denote the distortion parameters of the chosen lens model (e.g., Rational Polynomial for ARCTIC, Kannala–Brandt for Reinterhand, FisheyeRadTanThinPrism for HOT3D).

A distorted image point

$$P' = (u', v')^\top \in \mathcal{I},$$

where

$$\mathcal{I} = \{(u, v) \in \mathbb{R}^2 \mid 0 \le u < W,\ 0 \le v < H\},$$

is mapped to its undistorted counterpart

$$P = (u, v)^\top \in \mathcal{I}$$

by the distortion-correction function

$$\phi_d : \mathcal{I} \longrightarrow \mathcal{I}, \qquad \phi_d(u', v') = \begin{pmatrix} u \\ v \end{pmatrix}.$$

Here, ϕ_d implements the inverse of the selected distortion model (e.g. Rational Polynomial, Kannala–Brandt, or FisheyeRadTanThinPrism), ensuring that all subsequent projection and cropping operations use geometrically accurate, undistorted coordinates within the image domain $\mathcal{I}$.

7.2.2 Spatial Context via Crop Geometry. Let the hand bounding box in image $\mathcal{I}$ be given by the pixel coordinates

$$(x_1', y_1', x_2', y_2').$$

Its width and height are then

$$w' = x_2' - x_1', \qquad h' = y_2' - y_1'.$$

The center of this box in the distorted image is

$$(x_c', y_c') = \left(\tfrac{x_1'+x_2'}{2}, \tfrac{y_1'+y_2'}{2}\right).$$

Applying the inverse-distortion mapping $\phi_d : \mathcal{I} \to \mathcal{I}$ to the center yields the undistorted crop midpoint:

$$(u_c, v_c) = \phi_d(x_c', y_c').$$

Similarly, the undistorted coordinates of the four crop corners are

$$(u_{11}, v_{11}) = \phi_d(x_1', y_1'), \quad (u_{12}, v_{12}) = \phi_d(x_1', y_2'),$$
$$(u_{21}, v_{21}) = \phi_d(x_2', y_1'), \quad (u_{22}, v_{22}) = \phi_d(x_2', y_2').$$

Hence, the undistorted width w and height h of the hand crop are

$$w = u_{22} - u_{11}, \qquad h = v_{22} - v_{11}.$$

7.2.3 CIT Formulation. For any undistorted image coordinate $(u, v) \in \mathcal{I}$, the local viewing direction is defined as in Prakash et al. [2024]:

$$\boldsymbol{\theta}(u, v) = \begin{pmatrix} \theta_u \\ \theta_v \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{u - c_x}{f_x}\right) \\ \arctan\left(\frac{v - c_y}{f_y}\right) \end{pmatrix}.$$

We evaluate this mapping at the undistorted crop midpoint (u_c, v_c):

$$\boldsymbol{\theta}_c = \boldsymbol{\theta}(u_c, v_c),$$

and at each undistorted corner (u_{ij}, v_{ij}) for $i, j \in \{1, 2\}$:

$$\boldsymbol{\theta}_{ij} = \boldsymbol{\theta}(u_{ij}, v_{ij}).$$

Together, $\boldsymbol{\theta}_c$ and the set $\{\boldsymbol{\theta}_{ij}\}$ specify the viewing directions at the crop's center and its four corners.

Next, we compute the six scale-normalized crop intrinsics:

$$p_x = \frac{c_x - u_c}{w}, \qquad p_y = \frac{c_y - v_c}{h}, \tag{10}$$

$$r_w = \frac{w}{W}, \qquad r_h = \frac{h}{H}, \tag{11}$$

$$\alpha_x = \arctan\left(\frac{W}{2\, f_x}\right), \qquad \alpha_y = \arctan\left(\frac{H}{2\, f_y}\right). \tag{12}$$

Finally, we concatenate the five local-ray angles $\boldsymbol{\theta}_c,\ \boldsymbol{\theta}_{11},\ \boldsymbol{\theta}_{12},\ \boldsymbol{\theta}_{21},\ \boldsymbol{\theta}_{22}$ (each a $\mathbb{R}^2$ vector) with these six intrinsics values into a single vector:

$$\mathrm{CI} = \begin{bmatrix} \boldsymbol{\theta}_c \\ \boldsymbol{\theta}_{11} \\ \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{21} \\ \boldsymbol{\theta}_{22} \\ p_x \\ p_y \\ \log r_w \\ \log r_h \\ \alpha_x \\ \alpha_y \end{bmatrix} \in \mathbb{R}^{16}.$$

We then apply the standard sinusoidal positional encoding (as in Vaswani et al. [2017]) to **CI**, yielding the *Crop Intrinsics Token*

$$\mathrm{CIT} = \mathrm{PE}(\mathbf{u}) \in \mathbb{R}^{128}.$$

7.2.4 *Semantic Interpretation of CIT Components.*

$\theta_c, \{\theta_{ij}\}$: **Local viewing directions.** Each $\theta \in \mathbb{R}^2$ encodes the angular offset of a ray from the optical axis at a specific point in the crop (center or corner). By supplying these five directions, the network knows the precise local geometry of the patch—how objects tilt or foreshorten— which is essential for accurate 3-D pose recovery.

$p_{x,y}$: **Principal-point offset.** Recall that the camera's principal point (c_x, c_y) is the projection of the *optical axis* onto the image plane. The principal-point offset (Eqn. (10)) locates the centre of projection within the patch in $[0, 1]^2$.
Why is this important? Suppose a hand keypoint moves by Δu pixels to the right in the cropped image. Two things could cause this:
- The hand really moved to the right in 3-D space.
- The camera crop shifted to the left (i.e. x_1 increased).

Without p_x, the network has no way to tell these apart and must implicitly learn a mapping from appearance alone, which couples object motion and crop translation. By supplying p_x:
- A change in x_1 (crop shift) changes p_x but not the underlying feature activations for the hand—so the network can *subtract out* cropping effects.
- A genuine hand motion changes the relative position of pixels *and* the geometric residual after compensating for p_x, so the network correctly attributes that to 3-D movement.

This explicit disambiguation dramatically reduces bias when training on datasets with heterogeneous cropping strategies: the network no longer "confuses" camera recentering for object translation.

$\log r_{w,h}$: **Scale ratios.** The logarithms of the crop's width and height relative to the full image inform the network about how "zoomed in" a given patch is after resizing it to the fixed input resolution (e.g., 224×224). Concretely:
- If the patch occupies a large fraction of the original image ($r_w \approx 1$), resizing this large patch down to the fixed input size effectively reduces the region's apparent size.
- Conversely, if the patch is very small relative to the original image ($r_w \ll 1$), resizing that small patch to the fixed input significantly magnifies the region.

Expressing these ratios in log-space enables the network to linearly and smoothly interpolate across a wide range of zoom levels, facilitating robust generalization to diverse cropping strategies.

$\alpha_{x,y}$: **Half–field–of–view angles.** These angles define the camera's total angular aperture in the horizontal and vertical directions, and can be interpreted as the sensor's angular resolution per pixel For instance, a horizontal displacement of Δu pixels on the image plane corresponds to an angular change of $\Delta u / f_x$ radians. Supplying (α_x, α_y) therefore allows the network to convert image-space displacements into real-world directions in a device-independent manner.

Hence, each component of the CIT plays a complementary role in achieving our objective of camera-model general, cross-camera 3D hand pose estimation.

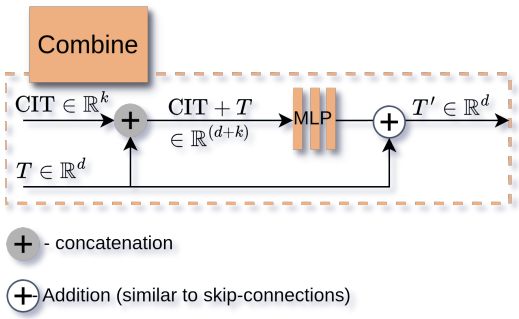


Fig. 15. **Fusing CIT and Crop Tokens.** For each crop (hand/arm), its CIT is broadcast to all patch tokens for that crop and fused in the Combine block via feature concatenation followed by a learnable projection, with a residual addition of the original token embedding. This injects crop-specific geometric context into every patch feature while allowing the model to fall back to the original mapping when the conditioning is unnecessary.

7.3 Ray Space Solver (RSS)

The Ray Space Solver (RSS) takes the 3D joints positions $\boldsymbol{J}_i$, their corresponding estimated 2D image keypoints (u_i, v_i), and the associated 2D confidence weights w_i, to compute the camera-space translation $\mathbf{t}$ by enforcing that each joint lies along its corresponding viewing ray:

$$\mathbf{J}_i + \mathbf{t} \;=\; \lambda_i \, \mathbf{d}_i, \qquad i = 1, \ldots, M, \tag{13}$$

is weighted by w_i, where M is the number of joints and $\boldsymbol{d}_i$ is the unit-direction vector of the ray passing through the 2D keypoint (u_i, v_i).

To compute each ray direction $\boldsymbol{d}_i$, we first back-project the 2D keypoint (u_i, v_i) into a normalized camera coordinate frame:

$$\bar{u}_i \;=\; \frac{u_i - c_x}{f_x}, \qquad \bar{v}_i \;=\; \frac{v_i - c_y}{f_y}, \qquad \rho_i = \sqrt{\bar{u}_i^2 + \bar{v}_i^2},$$

where f_x, f_y are the focal lengths and (c_x, c_y) is the principal point. For a pinhole model,

$$\tilde{\mathbf{d}}_i = (\bar{u}_i, \bar{v}_i, 1)^\top, \quad \mathbf{d}_i = \tilde{\mathbf{d}}_i / \|\tilde{\mathbf{d}}_i\|.$$

For an equidistant fisheye (example),

$$\mathbf{d}_i = \left(\tfrac{\bar{u}_i}{\rho_i} \sin \rho_i, \; \tfrac{\bar{v}_i}{\rho_i} \sin \rho_i, \; \cos \rho_i \right)^\top.$$

(Other calibrated models, e.g., rational polynomial or Kannala–Brandt, provide a similar unprojection function; in all cases we finally L2-normalize $\mathbf{d}_i$.)

A weighted least-squares fit of (13) can be formulated as

$$\min_{\mathbf{t},\lambda}; \sum_{i=1}^{M} w_i \, \left\| (\mathbf{t} + \mathbf{J}_i) - \lambda_i \mathbf{d}_i \right\|^2 \tag{14}$$

for each joint $i = 1, \ldots, M$. For fixed $\mathbf{t}$, the optimal depth obtained in closed form:

$$\lambda_i^\star \;=\; \mathbf{d}_i^\top (\mathbf{t} + \mathbf{J}_i). \tag{15}$$

Substituting (15) into (14) yields the *point-to-ray least-squares* objective

$$\min_{\mathbf{t}} \; E(\mathbf{t}) = \sum_{i=1}^{M} w_i \, \|\mathbf{r}_i\|^2, \qquad \mathbf{r}_i \;=\; \underbrace{\left(I - \mathbf{d}_i \mathbf{d}_i^\top\right)}_{\Pi_i} (\mathbf{t} + \mathbf{J}_i). \tag{16}$$

Here $\mathbf{\Pi}_i$ is the orthogonal projector onto the plane perpendicular to $\mathbf{d}_i$ and maps any vector to its component perpendicular to $\mathbf{d}_i$. In particular,

$$\mathbf{\Pi}_i \mathbf{d}_i \;=\; \big(I - \mathbf{d}_i \mathbf{d}_i^\top\big)\mathbf{d}_i = \mathbf{d}_i - \mathbf{d}_i(\mathbf{d}_i^\top \mathbf{d}_i) = \mathbf{d}_i - \mathbf{d}_i = \mathbf{0},$$

so $\mathbf{\Pi}_i$ *annihilates* all along-ray (depth) components. For any $\mathbf{v} \in \mathbb{R}^3$,

$$\mathbf{v}_\parallel = (\mathbf{v}^\top \mathbf{d}_i)\,\mathbf{d}_i \qquad \text{(“depth” component)},$$
$$\mathbf{v}_\perp = \mathbf{v} - \mathbf{v}_\parallel = (I - \mathbf{d}_i \mathbf{d}_i^\top)\mathbf{v} = \mathbf{\Pi}_i \mathbf{v} \qquad \text{(“sideways” component)}.$$

Hence $\|\mathbf{\Pi}_i \mathbf{v}\| = \|\mathbf{v}\| \sin\theta = \|\mathbf{v} \times \mathbf{d}_i\|$, where θ is the angle between $\mathbf{v}$ and $\mathbf{d}_i$.

In Eqn. 16, $\mathbf{r}_i = \mathbf{\Pi}_i(\mathbf{t} + \mathbf{J}_i)$ is the sideways residual component: it is exactly the shortest vector from the translated point $\mathbf{t} + \mathbf{J}_i$ to the ray through $\mathbf{d}_i$.

Differentiating (16) w.r.t. $\mathbf{t}$ gives the 3×3 weighted equations

$$\Big(\sum_{i=1}^{M} w_i\, \mathbf{\Pi}_i\Big)\mathbf{t} \;=\; -\sum_{i=1}^{M} w_i\, \mathbf{\Pi}_i\, \mathbf{J}_i, \qquad \text{i.e.,} \qquad M\,\mathbf{t} \;=\; -\,\mathbf{m}, \tag{17}$$

with

$$M = \sum_{i=1}^{M} w_i \mathbf{\Pi}_i \in \mathbb{R}^{3\times 3}, \qquad \mathbf{m} = \sum_{i=1}^{M} w_i \mathbf{\Pi}_i \mathbf{J}_i \in \mathbb{R}^3.$$

We solve (17) using a tiny Tikhonov term for stability:

$$\widehat{\mathbf{t}} \;=\; -\,(M + \varepsilon I)^{-1}\mathbf{m}, \qquad \varepsilon \ll 1. \tag{18}$$

After $\widehat{\mathbf{t}}$ is found, each joint's camera-space position are

$$\mathbf{p}_i = \widehat{\mathbf{t}} + \mathbf{J}_i,$$

Notes on robustness. In practice we check the condition number $\kappa(M)$; if $\kappa(M) \geq 10^6$ we damp the weights ($w_i \leftarrow 0.5\, w_i$) and resolve. For more details about the formulation, we refer the readers to Tikhonov regularization [Hoerl and Kennard 1970].

7.3.1 Kalman Filter. We use a 3D constant-velocity Kalman filter on the predicted camera-space translations. The process and measurement noise variances ($q_{\text{pos}}, q_{\text{vel}}, r_{\text{meas}}$) are tuned offline on H2O, HOT3D and ARCTIC datasets. We select the filter hyperparameters by minimizing a simple objective that trades off (i) accuracy w.r.t. ground truth during visible frames and (ii) temporal smoothness:

$$\mathcal{L} \;=\; \lambda\, \mathcal{L}_{\text{fid}} \;+\; (1 - \lambda)\, \mathcal{L}_{\text{smooth}}, \tag{19}$$

where

$$\mathcal{L}_{\text{fid}} \;=\; \Big\|\tilde{\mathbf{p}}_t - \mathbf{p}_t^{\text{gt}}\Big\|_2^2 \tag{20}$$

measures fidelity to the ground-truth translation on visible frames, and

$$\mathcal{L}_{\text{smooth}} \;=\; \|\tilde{\mathbf{p}}_{t+1} - 2\tilde{\mathbf{p}}_t + \tilde{\mathbf{p}}_{t-1}\|_2^2 \tag{21}$$

penalizes second-order temporal differences (acceleration), encouraging smooth motion. We fix λ (e.g., $\lambda = 0.7$).

8 Experimental Evaluation

8.1 Dataset Details

ARCTIC [Fan et al. 2023] contains 393 sequences of hand–object interactions involving 11 articulated objects and 10 subjects, recorded from eight static and one egocentric fisheye view. We use both exocentric and egocentric frames from the official training split (1.7M images) for training, and evaluate exclusively on the egocentric subset (23K images) from the official validation split. Since the official test set is not publicly available and does not support camera-space joint-error evaluation, we report results on the validation set as a proxy for testing.

H2O (Two Hands and Objects) [Kwon et al. 2021] provides synchronized multi-view RGB-D sequences of bimanual hand–object interactions with 3D hand and object poses, camera parameters, and meshes. We use both exocentric and egocentric frames from the official training split (167K images) for training, and evaluate only on the egocentric frames from the official test split (23K images).

HOT3D [Banerjee et al. 2024] offers over 833 minutes of egocentric video from Project Aria glasses [Engel et al. 2023], featuring 19 participants interacting with 33 rigid objects. We use only the monocular RGB stream and divide it into 354K training and 110K test frames (see Sup. Sec. 8.2 for split details).

HO3D [Hampali et al. 2020] consists of hand–object interaction sequences with severe occlusions caused by object manipulation. We use the V2 version of the dataset and follow the official training (66K images) and test (11K images) splits.

For *Re:InterHand* and *HandCO*, we use the entire datasets for training.

8.2 HOT3D Split

Although HOT3D provides an official 20% test split, we ignore it due to the lack of ground-truth annotations for testing the CS-MJE and no official server to test this metric (to the best of our knowledge); instead, we take the remaining 80% of the data and split it into 60% for training and 20% for validation. The full list of image splits will be released upon publication.

8.3 FARM Generation for Datasets

H2O. We first train a dedicated 2D arm-pose network and run it on every camera view of the H2O recordings to obtain per-view arm 2D keypoints. These detections are triangulated across views to recover 3D arm joints. In parallel, we generate multi-view arm segmentation masks. The resulting 3D keypoints and silhouettes are then fed to our optimisation stage, which refines arm pose and shape and returns the corresponding FARM parameters.

ARCTIC. For ARCTIC we directly optimise the arm vertices of an SMPL-X body model using the available 3D supervision, and convert the fitted mesh to the required FARM parameter set.

8.4 Evaluation Metrics

We report the following metrics:

- **Camera-space Mean Joint Error (CS-MJE).** Mean Euclidean distance (mm) between predicted and ground-truth 3D joints

in the camera-space [Chen et al. 2021; Valassakis and Garcia-Hernando 2024], capturing errors in hand pose, translation, scale, and rotation.

- **Root-relative Mean Joint Error (RS-MJE).** Mean Euclidean distance (mm) after subtracting the root joint [Chen et al. 2024; Grauman et al. 2024], capturing pose, scale, and rotation errors while ignoring translation.
- **Procrustes-aligned Mean Joint Error (PS-MJE).** Mean Euclidean distance (mm) after rigid Procrustes alignment (removing scale, rotation, and translation), following [Lin et al. 2021; Pavlakos et al. 2024].
- **Acceleration error (ACC).** ACC measures temporal stability [Kanazawa et al. 2019; Kocabas et al. 2020]. We report CS-ACC in camera space (translation jitter) and RS-ACC in root-relative space (hand jitter), both in m/s^2.

9 Additional Experiments

9.1 Results

Table 5. Comparison between the released HandDGP model weights and our reimplementation (†) on HOT3D (rectified). Since HandDGP does not provide training code, † is trained via our best-effort reproduction under the same specification as HandDGP. Both variants are evaluated with an identical preprocessing (same rectification, cropping, resolution). Similar results on HOT3D for the released HandDGP model weights are also reported in Zhang et al. [2025]. We report CS-MJE (mm), PS-MJE (mm), and CS-ACC (m/s^2); lower is better for all metrics.

Method	CS-MJE ↓	PS-MJE ↓	CS-ACC ↓
HandDGP (released weights)	109.3	16.3	21.9
HandDGP† (reimplemented)	61.3	8.6	20.4

In-the-Wild. We define *in-the-wild* video sequences as videos for which no calibrated camera model is available. In these cases, we estimate camera intrinsics using AnyCalib [Tirado-Garín and Civera 2025]. While this strategy works reasonably well for pinhole optics—as illustrated in Fig. 17(b) and in the supplementary video—our experiments in Table 9 show that accuracy degrades when applying the same procedure to fisheye cameras. Moreover, because our method is trained exclusively on calibrated 3D datasets recorded in controlled laboratory environments, it does not fully generalize to out-of-domain data.

Runtime Comparison. The initial hand-arm detection stage of our pipeline takes around 40 ms to run, followed by the model forward pass at 24.2 ms, and the RSS lifting module at 3.1 ms. Tab. 6 reports compute time (ms) for the model forward pass and the lifting step across methods, measured with a batch size of two (both hands) on an RTX 3090. We report the mean runtime with the standard deviation spread σ in parentheses, where σ denotes three standard deviations and serves as a measure of runtime jitter. Feed-forward approaches, such as HandDGP and our EgoForce, exhibit consistently low lifting cost (2.8–3.1 ms), whereas optimization-based methods incur substantially higher lifting overhead, most notably HandOccNet (220.9 ms with large jitter), reflecting the iteration- and conditioning-dependent nature of per-frame refinement.

Table 6. **Runtime performance metrics on the HOT3D dataset.** We report mean compute time (ms) with the corresponding σ deviation in parentheses. "FF" denotes a feed-forward model, and "Opti" denotes iterative optimization-based 3D-to-2D lifting.

Method	Compute Time (ms)		Type
	Model (σ)	Lifting (σ)	
MobRecon	14.0 (8.8)	18.1 (12.2)	FF + Opti
HandOCCNET	15.3 (2.1)	220.9 (313.8)	FF + Opti
HandDGP	17.5 (3.3)	2.8 (15.8)	FF
OURS	24.2 (2.4)	3.1 (0.5)	FF

Comparison to Additional SOTAs. UmeTrack [Han et al. 2022] and HaWoR [Zhang et al. 2025] did not release training code, providing only pretrained models and evaluation pipelines; our comparisons are based on their released inference and evaluation setups.

UmeTrack. Table 7 compares EgoForce with UmeTrack under two crop settings. When UmeTrack is evaluated with a perspective crop built from ground-truth 3D keypoints, this should be regarded as a best-case setting, since such crop information is unavailable in real deployment. Even under this favorable protocol, EgoForce achieves lower PS-MJE on all three datasets and also lower CS-MJE on ARCTIC, showing that its gains are not limited to global translation recovery but also extend to articulated hand reconstruction. The more realistic comparison is therefore the setting (UmeTrack*) in which the perspective crop is built from 2D hand bounding boxes, which are available at inference time either from dataset annotations or from a hand detector. Under this realistic protocol, EgoForce is clearly superior, reducing CS-MJE by 68.1% on ARCTIC, 83.2% on HOT3D, and 79.9% on H2O, while also improving PS-MJE by 67.3%, 77.2%, and 74.1%, respectively.

Moreover, our hand-scale analysis shows that UmeTrack in single-view evaluation exhibits 6 mm frame-wise scale variability (standard deviation across continuous frame sequences) on its own dataset and 18 mm on HOT3D, whereas EgoForce shows only 4 mm on HOT3D. This gap is expected. UmeTrack was introduced primarily as a multi-view VR hand-tracking system, and its formulation explicitly notes that recovering hand scale for an unknown skeleton requires multi-view features, since single-view input is inherently affected by scale ambiguity. In contrast, EgoForce is explicitly designed for monocular camera-space reconstruction: forearm cues provide metric information to reduce monocular depth–scale ambiguity, and ray-space lifting preserves calibrated camera geometry across different optics. This further indicates that EgoForce is not only more accurate, but also more stable for practical real-world deployment.

As shown in Fig. 19, UmeTrack, even when using crops derived from ground-truth 3D keypoints, does not reliably recover hand pose under hand–object occlusions (ARCTIC, left). Even in non-occluded cases, its 2D finger reprojections are less accurate than ours (H2O, middle). Furthermore, on HOT3D (right), the interacting left hand pose estimate and its reprojection are not faithful to the observed hand.

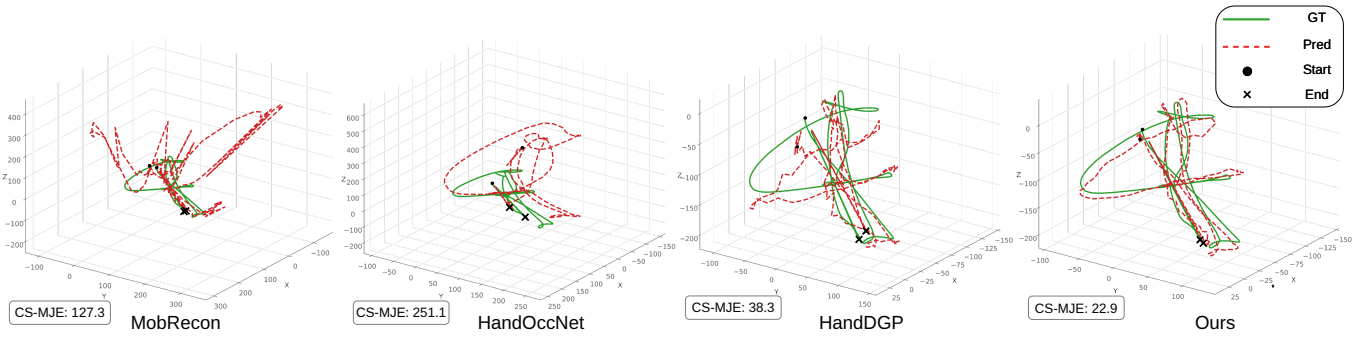


Fig. 16. **Right-hand camera-space trajectory for a HOT3D sequence.** Our approach produces a more accurate hand trajectory in camera space, particularly along the depth (z-axis), compared to competing approaches. We visualize 160 frames from the sequence.

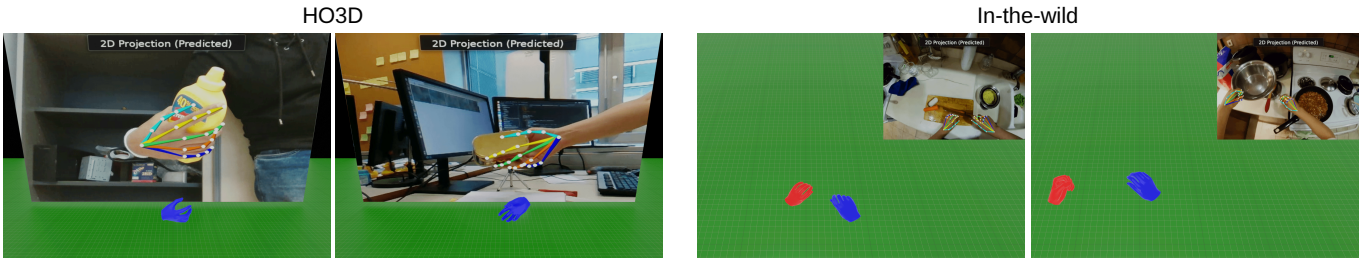


Fig. 17. **Qualitative results on HO3D and in-the-wild data.** Our approach produces accurate hand pose estimates even under hand–object occlusions on HO3D (a), and it generalizes to in-the-wild videos despite not being explicitly trained on those data distributions (b).

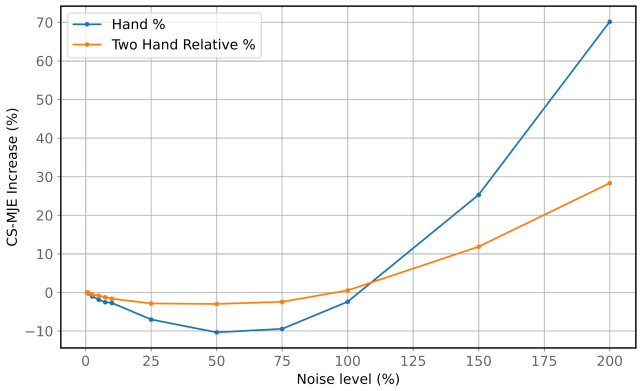
Table 7. **Comparison with UmeTrack on ARCTIC, HOT3D, and H2O (in mm).** We report camera-space mean joint error (CS-MJE) and Procrustes-aligned mean joint error (PS-MJE). For UmeTrack, we evaluate two crop settings: 3D keypoints-based perspective crop and 2D bounding-box-based perspective crop.

Method	ARCTIC		HOT3D		H2O	
	CS-MJE ↓	PS-MJE ↓	CS-MJE ↓	PS-MJE ↓	CS-MJE ↓	PS-MJE ↓
UmeTrack (3D keypoints crop)	55.1	14.7	**31.9**	12.1	**23.8**	11.1
UmeTrack* (2D bounding-box crop)	155.4	24.5	261.7	28.9	124.6	21.6
EgoForce (ours)	**49.5**	**8.0**	43.9	**6.6**	25.0	**5.6**

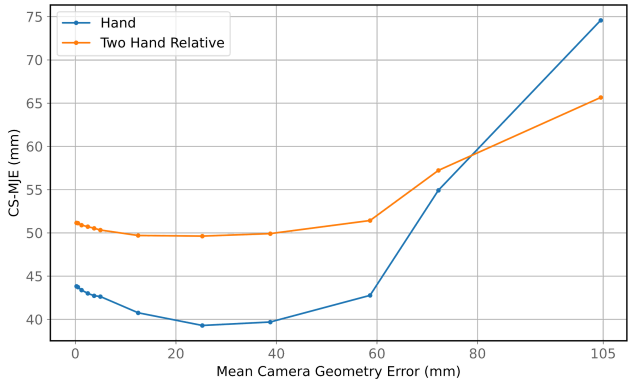
HaWoR. Since HaWoR depends on SLAM for its global hand pose estimation, it introduces additional computational overhead and makes the pipeline sensitive to failures in camera tracking. This is particularly problematic in egocentric hand-interaction scenarios, where the camera is often close to the hands and manipulated objects frequently occupy a large part of the view, reducing the stability of feature matching and pose estimation. Such conditions are common in ARCTIC, where close-up views and frequent object interactions make SLAM especially unreliable in our experiments. As a result, HaWoR performs poorly on ARCTIC, with 319.9 mm CS-MJE and 16.3 mm PA-MJE, whereas EgoForce achieves 49.5 mm CS-MJE and 8.0 mm PA-MJE. On H2O, HaWoR performs better, reaching 72.5 mm CS-MJE and 6.6 mm PA-MJE, but EgoForce still outperforms it with 25.0 mm CS-MJE and 5.6 mm PA-MJE. These results indicate that direct monocular camera-space reconstruction is more robust than SLAM-dependent global pose recovery in close-range egocentric interaction settings.

Calibration-mismatch robustness. In real deployments, intrinsics may be noisy, shifted over time, or only approximately available rather than precisely measured for each device. We therefore evaluate sensitivity to calibration mismatch to test whether performance remains stable under realistic intrinsic errors. We evaluate calibration-mismatch sensitivity (see Fig. 18) by interpolating camera intrinsics between the default dataset calibration (0%) and AnyCalib [Tirado-Garín and Civera 2025] estimate adapted to HOT3D's camera model (100%), then extrapolating to 200%.

To quantify the mismatch, we define camera-geometry error that measures how much the perturbed camera changes viewing rays relative to the reference camera: for a grid of image pixels, we compute ray directions from the reference and perturbed camera models, measure their angular difference, and convert this into a metric displacement at multiple depths to account for near- and far-field hand positions in egocentric capture; the reported value is the mean displacement in millimeters, where larger values indicate stronger geometric inconsistency. Despite increasing camera-geometry error away from the dataset calibration, hand pose accuracy remains



(a) Relative change in CS-MJE (%) with increasing camera intrinsic noise level (%) on HOT3D.



(b) Effect of camera-geometry error on CS-MJE (mm) on HOT3D.

Fig. 18. **Robustness to calibration mismatch on HOT3D.** As camera-intrinsic perturbation increases, CS-MJE remains stable and even improves slightly under moderate mismatch, despite increasing camera-geometry error; performance degrades clearly only under large mismatches, indicating robustness to moderate calibration error.

stable over a broad range and is best at 50%, showing robustness to intrinsic mismatch and shows graceful degradation under extreme deviations (>150%). Interestingly, the best result is obtained at an intermediate interpolation between the dataset and AnyCalib intrinsics. However, selecting such an interpolation at deployment is not possible without ground-truth 3D hand poses, motivating future work on combining factory and software-estimated intrinsics for on-the-fly self-calibration without ground-truth.

Arm-based depth–scale stabilization. To better understand why arm information improves absolute hand reconstruction, we go beyond the aggregate CS-ACC and RS-MJE gains in Tab. 4 and analyze its effect on depth–scale ambiguity directly. Since monocular egocentric hand estimation is inherently affected by depth–scale ambiguity, we measure hand scale error, defined as the wrist-to-middle-finger MCP distance error, across different hand-to-camera distances. Arm-based depth–scale stabilization comes from: (1) forearm input to HALO, which provides additional context for improved hand mesh

Table 8. **Arm-based depth–scale stabilization.** Mean hand scale error on ARCTIC (mm), with changes shown in parentheses. Arm input improves scale estimation, especially in the near field, supporting its role in reducing depth–scale ambiguity.

Distance of hands from camera (mm)	Mean Hand Scale Error↓ (mm)	
	Without arm input	With arm input
200–300 (near-field)	4.7	2.7 (−2.0 ↓)
300–500	3.2	3.2 (0.0)
500–1000 (far-field)	3.3	3.0 (−0.3 ↓)

estimation, and (2) the parametric forearm mesh predicted by HALO (FARM shape and pose), which provides the arm 3D joints to RSS and anchors the hand–arm. Tab. 8 shows that arm context helps most in the near field, is neutral in the mid range, and still helps in the far field, supporting its role in mitigating depth–scale ambiguity rather than merely improving temporal smoothness or local mesh quality.

Hand-size metrics. To assess whether reconstruction depends on explicit hand-size calibration, we analyze both within-sequence scale stability and generalization across unseen hand sizes. Since our method does not use per-user hand-size calibration, we verify that the predicted scale remains stable across frames and is not strongly dependent on subject-specific size tuning. Even without calibration, frame-wise hand-scale variation remains small: around 4 mm on HOT3D and 2 mm on ARCTIC, corresponding to only 4% and 2% of the average hand size (HOT3D: 94 mm and ARCTIC: 90 mm), respectively. Notably, HOT3D and ARCTIC together cover 5 unseen hand sizes. Per-sequence calibration further reduces CS-MJE on HOT3D from 43.9 mm to 42.7 mm, while yielding no noticeable improvement on ARCTIC. These results indicate that EgoForce achieves stable scale reconstruction without explicit per-user calibration, is robust to hand-size variation across subjects, and remains stable within a sequence. This supports our goal of deployment on smart glasses while keeping additional calibration dependencies minimal.

9.2 Ablations

Camera-Space Lifting. Table 9 compares four ways for lifting root-relative predictions into camera space on H2O (pinhole) and HOT3D (fisheye). The naïve depth-based lifting, similar to HaMeR^{D}'s metric-depth formulation, fails under monocular scale ambiguity, yielding large translation errors (530.5/1851.9 mm) and correspondingly high acceleration errors (41.9/180.4 m/s^2). On H2O, our Ray Space Solver (RSS) and DGP from Valassakis and Garcia-Hernando [2024] achieve the same CS-MJE of 25 mm with nearly identical CS-ACC (7.4 m/s^2), indicating that both correctly exploit the pinhole projection constraints. On HOT3D, however, DGP reaches 115.6 mm CS-MJE with 25.0 m/s^2 CS-ACC, whereas RSS gives 45.8 mm and 23.5 m/s^2, respectively. Applying Kalman filtering on top of RSS ("RSS w/ KF") further stabilizes the lifted trajectory, yielding 43.9 mm CS-MJE and 14.0 m/s^2 CS-ACC, the best performance on both datasets. Please refer to Sup. Fig. 13 for qualitative illustrations of different camera-space lifting techniques, and to the supplementary video for the impact of Kalman filtering on temporal smoothness.

Fig. 19. **Qualitative camera-space results on egocentric datasets.** We compare our method against UmeTrack [Han et al. 2022] on three datasets with widely different camera intrinsics. Predicted left and right limb meshes are shown in red and blue, respectively, with ground truth highlighted in gray.

Table 9. **Ablation of Camera-Space Lifting Approaches.** We report CS-MJE (mm) and CS-ACC (m/s^2) on H2O (pinhole) and HOT3D (fisheye). "Est. Intri" = Estimated Intrinsics; "DGP" = Differential Global Positioning [Valassakis and Garcia-Hernando 2024]; "RSS" = Ray-Space Solver; "KF" = Kalman Filtering.

Method	H2O		HOT3D	
	CS-MJE ↓	CS-ACC ↓	CS-MJE ↓	CS-ACC ↓
HALO + Depth	530.5	41.9	1851.9	180.4
HALO + DGP	25.0	7.4	115.6	25.0
HALO + RSS (w/o KF)	25.0	7.4	45.8	23.5
HALO + RSS (w KF) (Ours)	**25.0**	**5.5**	**43.9**	**14.0**

Table 10. **Ablation of CIT.** Hand CS-MJE on HOT3D (mm), with the change shown in parentheses. Keeping undistortion fixed, CIT consistently improves performance.

Hand location by radial region (%)	With undistortion w/o CIT → w/ CIT	Without undistortion w/o CIT → w/ CIT
0–25	38.7 → 36.2 (−2.5 ↓)	67.4 → 42.8 (−24.6 ↓)
25–50	42.5 → 38.8 (−3.8 ↓)	95.7 → 56.8 (−38.9 ↓)
50–75	43.5 → 40.4 (−3.1 ↓)	141.5 → 82.1 (−59.4 ↓)
≥ 75	58.5 → 54.8 (−3.7 ↓)	152.8 → 99.3 (−53.5 ↓)

Isolating CIT from undistortion. In Tab. 10, we keep undistortion fixed and toggle only CIT. Across all radial regions, CIT consistently reduces hand CS-MJE. Importantly, these gains persist even after undistortion, confirming that CIT provides benefits beyond preprocessing. The improvements remain consistent across the full image, including peripheral regions where fisheye distortion is strongest, indicating that CIT improves robustness throughout the image, consistent with Tab. 3.

10 Implementation Details

Arm–Hand Crop Encoder. Given the hand and arm crops $\mathbf{I}_H \in \mathbb{R}^{224\times224\times3}$ and $\mathbf{I}_A \in \mathbb{R}^{112\times112\times3}$, patchification with patch size $p = 16$ produces $N_H = (224/p)^2 = 14^2 = 196$ hand tokens and $N_A = (112/p)^2 = 7^2 = 49$ arm tokens, for a total of $N = N_H + N_A = 245$ tokens. We use a ViT-H/16 backbone (pretrained ViTPose-H weights) with token and feature dimension $d = c = 1280$. This yields hand tokens $\mathbf{T}_H \in \mathbb{R}^{196\times1280}$ and arm tokens $\mathbf{T}_A \in \mathbb{R}^{49\times1280}$ and concatenating them produces the encoded visual tokens $\mathbf{X} \in \mathbb{R}^{245\times1280}$. The Crop Intrinsics Tokens have dimension $k = 128$ (Sec. 7.2) and are fused per patch by concatenation, a linear projection $\mathbb{R}^{d+k} \to \mathbb{R}^d$, and a residual addition (see Fig. 15).

Contextual Decoding of Hand–Arm Interactions. We instantiate four hand queries and three arm queries, $\mathbf{Q}_H \in \mathbb{R}^{4\times c}$ (2D joints, global pose, hand shape, hand pose) and $\mathbf{Q}_A \in \mathbb{R}^{3\times c}$ (2D joints, arm shape, arm pose), and stack them to form the target sequence $\mathbf{Q}_0 = [\mathbf{Q}_H; \mathbf{Q}_A] \in \mathbb{R}^{7\times c}$. A transformer decoder with $L_{\text{dec}} = 2$ layers and $h_{\text{dec}} = 8$ attention heads attends to the encoded patch tokens $\mathbf{X} \in \mathbb{R}^{N\times c}$ and outputs $\mathbf{Q}_L \in \mathbb{R}^{7\times r}$, where $r = 1280$. We split $\mathbf{Q}_L$ into hand and arm features, $\mathbf{f}_{\text{hand}} \in \mathbb{R}^{4\times1280}$ and $\mathbf{f}_{\text{arm}} \in \mathbb{R}^{3\times1280}$. The decoder self-attention enables information exchange between hand

and arm queries, while cross-attention grounds each query in the visual evidence provided by $\mathbf{X}$.

Plausible Arm Completion. When the forearm is not visible, we replace the missing arm query features using a hand-conditioned variational prior. Specifically, we predict $(\mu, \log \sigma^2) \in \mathbb{R}^{128}$ from the available hand features using two linear heads, sample a latent arm code $\mathbf{z}_{\text{arm}} \in \mathbb{R}^{128}$ via the reparameterization trick, and project it to the feature width $D = 1280$. We then decode this embedding using three residual MLP blocks (LayerNorm + ReLU). A final linear layer outputs an arm-query embedding $\mathbb{R}^{3\times1280}$ and is used to inpaint the missing arm query features.

2D Joint Decoder. We decode 56×56 heatmaps for $J_H = 21$ hand joints and $J_A = 3$ forearm joints, and obtain 2D coordinates via soft-argmax using $\text{softmax}(\tau\,\mathbf{H})$ with learnable temperature τ (initialized to 1). Per-joint confidence weights $w_j \in (0, 1)$ are predicted by an MLP on features bilinearly sampled at the predicted joint locations.

Ray Space Solver. We estimate the camera-space translation $\mathbf{t} \in \mathbb{R}^3$ from 24 2D–3D correspondences (21 hand + 3 forearm joints). Let $\tilde{\mathbf{u}}_i = (\tilde{u}_i, \tilde{v}_i)$ be the predicted 2D joint in crop coordinates, where the network input crop has size $(W_{\text{in}}, H_{\text{in}})$. We map crop coordinates back to full-image pixels by

$$u_i = x_0 + s_x \frac{\tilde{u}_i}{W_{\text{in}}}, \qquad v_i = y_0 + s_y \frac{\tilde{v}_i}{H_{\text{in}}},$$

where (x_0, y_0) denotes the top-left corner of the crop's bounding box in the full image, and (s_x, s_y) denotes its size in full-image pixels (width and height), *i.e.*,, $s_x = x_1 - x_0$ and $s_y = y_1 - y_0$ for the bottom-right corner (x_1, y_1). These full-image joint coordinates are normalized by calibrated intrinsics and unprojected through the native camera model to form unit bearing rays, after which we solve for $\mathbf{t}$ via a weighted point-to-ray least-squares system in closed form (see Sec. 7.3).

Kalman Filter. We temporally smooth the estimated camera-space translation $\mathbf{t}$ using a constant-velocity Kalman filter at freq = 30 Hz, with process-noise variances $q_{\text{pos}} = 0.001$ (position) and $q_{\text{vel}} = 10^{-5}$ (velocity), and measurement-noise variance $r_{\text{meas}} = 0.001$.

11 Limitations

Beyond the limitations discussed in the paper, we note several additional considerations.

(1) Although the forearm prior mitigates monocular depth–scale ambiguity, recovering a precisely metrically scaled MANO hand–arm configuration from a single monocular device remains underconstrained without user-specific scale cues (e.g., hand size or limb length).

(2) The arm modeling primarily serves as a geometric prior to stabilize and improve hand pose estimation. While the predicted arm meshes are often plausible, they are not yet optimized for tasks requiring precise hand–forearm localization.

(3) More generally, inferring the individual limb geometry remains difficult under severe occlusion, limited field of view, and fast motion. A more holistic model that jointly reasons about both the limbs, or even the upper body, could provide a stronger kinematic context and further stabilize hand and arm estimates.