RLHND: Video Foundation Models as Physically Grounded Hand Trackers for Robot Learning

Seungjun Moon[1,2] **Subin Jeon**[3] **Sangwoo Kim**[1] **Hanbyul Joo**[3] **Jinwoo Shin**[1,2]
[1]RLWRLD [2]KAIST [3]Seoul National University
seungjun.moon@rlwrld.ai

arXiv:2610.09455v1 [cs.CV] 7 Oct 2026

Abstract

Recently, approaches that leverage human video datasets for robot policy training have become increasingly prevalent. However, most existing hand trackers regress pose from cropped frames with limited priors on hand motion and object interaction, resulting in inaccurate and physically inconsistent estimates. Moreover, the lack of physical cues, *e.g.*, contact and force, limits the use of human videos for robot policy training. To this end, we propose RLHND, a video foundation model-based hand tracking model that jointly estimates hand pose and realistic tactile information from monocular egocentric videos. RLHND turns the pre-trained Cosmos 3 video diffusion backbone into a deterministic clip-level feature extractor via clean-latent conditioning, carrying its learned priors on hand motion and hand-object interaction into tracking. For pose estimation, RLHND (i) predicts hand poses with anatomically plausible joint angles and (ii) enables optional conditioning on the shape parameter to maintain consistent hand shape within the same video and even across videos recorded by the same actor. For tactile estimation, a separate tactile expert stream, trained with the pose stream frozen, predicts dense contact and force over the hand surface. We further adopt LBS-based feature spreading to enable vertex-wise feature extraction without costly per-vertex attention. RLHND achieves state-of-the-art performance across various benchmark datasets for pose estimation, while also achieving state-of-the-art performance in contact and force estimation. Moreover, we demonstrate the utility of RLHND for robot learning through retargeting results and real-world robot experiments. The code will be publicly available at https://seungjun-moon.github.io/rlhnd/.

1 Introduction

Along with the rapid advancement of vision-language models (OpenAI, 2023; Liu et al., 2023; Gemini Team, 2023; Bai et al., 2025), robotics foundation models have also made significant progress in recent years (Brohan et al., 2023; Octo Model Team et al., 2024; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025; Bjorck et al., 2025; Kim et al., 2026b). However, robotics datasets remain substantially more limited in scale than vision-language datasets and are considerably more labor-intensive to collect. Recently, researchers have recognized the potential of human egocentric videos, which are relatively easy to collect and can capture a wide range of tasks that are difficult to demonstrate through teleoperation. Consequently, human datasets (Grauman et al., 2022; Banerjee et al., 2024; Hoque et al., 2026) have been increasingly leveraged to train robotics foundation models (Ye et al., 2025a; Kareer et al., 2024; Yang et al., 2025; Luo et al., 2025; Kim et al., 2026b).

However, utilizing human videos for robot training requires bridging two major gaps between human videos and real-world robot episodes. The first is the *action-space discrepancy* between humans and robots. Existing works address this discrepancy by retargeting human hand motion to a specific robot embodiment (Handa et al., 2020; Qin et al., 2023; Shaw et al., 2022; Lepert et al., 2025), defining a co-action space that encompasses both human and robot joint actions (Kareer et al., 2024; Luo et al., 2025; Li et al., 2025), or adopting an end-effector-based action space consisting of wrist poses and keypoints (Haldar & Pinto, 2025; Liu et al., 2025; Kim et al., 2026a; Cai et al., 2025).

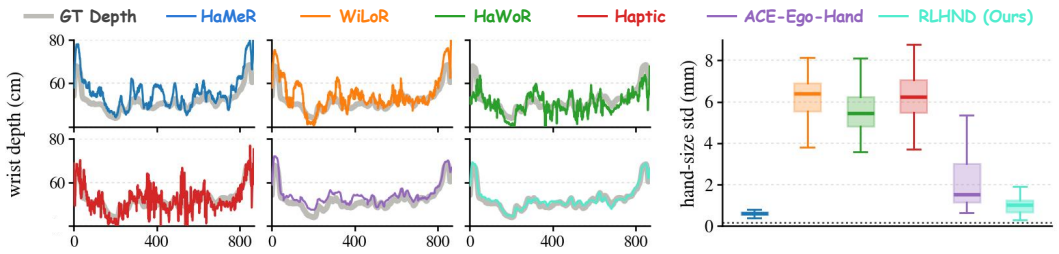

Figure 1: **3D instability of hand motion reconstruction methods.** *Left:* Wrist depth over 850 frames from an ARCTIC test-set clip. Baselines exhibit severe oscillation or drift relative to the ground-truth depth. *Right:* Per-video standard deviation of the estimated hand size. Even within a single video, existing hand motion estimators exhibit ∼8 mm variation in estimated hand size.

Regardless of the choice of action space, these approaches require human action labels that provide 3D hand joint locations. However, existing hand motion reconstruction methods are often optimized primarily for 2D reprojection or image-space alignment, which can make their 3D predictions unreliable for robot action labeling. As shown in Figure 1, existing hand reconstructions exhibit severe jitter along the depth axis (left), while the estimated size of the same hand can fluctuate substantially across frames (right). The latter arises from the inherent scale-depth ambiguity under perspective projection, where depth errors can be absorbed into the estimated hand shape. Such temporally inconsistent and physically implausible 3D trajectories are problematic for robot learning, where small errors in motion estimation can lead to substantially different or infeasible actions.

The second gap is *missing physical information*. Recent vision-language-action models increasingly incorporate tactile input alongside vision (Zhang et al., 2026a; Niu et al., 2026; Chen et al., 2026; Zhang et al., 2026b), as tactile information can distinguish a successful grasp from merely hovering over an object. However, human datasets generally cannot provide tactile information, except for those collected with pressure boards (Grady et al., 2022) or wearable tactile gloves (Song et al., 2025). To this end, several vision-based contact and force estimation models (Jeon et al., 2026; Jung & Lee, 2025) have emerged, but these remain underexplored compared to hand pose estimation.

In this paper, we present RLHND, which produces both metrically accurate, anatomically realistic 3D bimanual hand motion and dense contact and force estimates over the hand surface from a monocular human video. Since recent works (Wang et al., 2026; Liu et al., 2026) show that pre-trained video diffusion models internalize strong priors on hand-object interactions, RLHND uses Cosmos 3 Nano (NVIDIA, 2026) as its video feature encoder. Unlike Liu et al. (2026), which feeds clean videos directly into a denoising backbone, we instead route them through the conditioning-frame interface of Cosmos 3, which is designed to receive clean context frames during pre-training. This design keeps our fine-tuning aligned with the pre-trained interface of the backbone.

On top of the Cosmos 3 encoder, we make three design choices motivated by real-world deployment. *(i) Shape caching.* RLHND caches the shape parameter and uses a single value throughout the entire video. This not only eliminates the hand-size fluctuations shown in Figure 1, but also allows direct injection of a pre-calibrated $\boldsymbol{\beta}$, which is particularly useful in real-world data collection settings where the hand shape of the operator is known. *(ii) Anatomical pose parameterization.* Rather than regressing the 45 rotational DoF of MANO, we align each joint with its anatomical axes and freeze the axes about which a human finger cannot rotate, regressing 29 DoF. *(iii) A decoupled tactile expert.* Contact and force are predicted by a second expert stream that is trained with the pose stream frozen, so tactile supervision never perturbs the pose. To obtain vertex-wise features, we introduce LBS-based feature spreading, which uses the linear blend skinning weights of MANO to spread bone-level features to the 778 MANO vertices, avoiding the need for costly vertex-wise attention.

For motion reconstruction, we evaluate RLHND on HOT3D and ARCTIC, as well as EgoDex, which is held out during training. RLHND consistently outperforms all baselines across the three benchmark datasets. For tactile estimation, RLHND achieves the best performance in both contact and force estimation among various benchmarks. Finally, we demonstrate the utility of RLHND for robot learning by retargeting its human demonstrations to five dexterous hands and training Dexterous Point Policy (Kim et al., 2026a). RLHND produces more accurate and less jittery joint actions than the baselines, and training with keypoints from RLHND improves Dexterous Point Policy notably.

Contributions. We highlight the main contributions of our work as follows:

- We extend video foundation model-based hand tracking beyond pose estimation: RLHND leverages the clean-latent conditioning interface of Cosmos 3 to extract rich spatiotemporal representations from monocular video and jointly estimates metrically accurate bimanual motion with dense contact and force over the hand surface.
- We achieve state-of-the-art performance in both hand motion reconstruction and tactile estimation, enabled by physically grounded design choices including cached hand shape, anatomically constrained pose, and a decoupled tactile expert with LBS-based feature spreading.
- We demonstrate the utility of RLHND for robot learning: its outputs yield accurate and smooth retargeted commands across five dexterous hands and enable training Dexterous Point Policy from human demonstrations with automatically labeled contact.

2 Related Work

Hand motion reconstruction. Hand motion reconstruction methods have been developed to improve both pose accuracy and temporal consistency. The primary approach uses image-based methods (Pavlakos et al., 2024; Potamias et al., 2024), which regress MANO (Romero et al., 2017) parameters from a cropped image input by a hand detection model (Potamias et al., 2024). Since they treat every frame independently, they cannot consider any temporal consistency among frames. In contrast, video-based methods (Zhang et al., 2025; Ye et al., 2025b; Xu et al., 2026) add temporal context either by fusing crop features across frames (Ye et al., 2025b) or by tracking the hand and the camera in a world frame (Zhang et al., 2025; Xu et al., 2026). Despite this, their reliance on per-frame crops causes them to inherit detector box jitter and remain only weakly consistent over time. Most recently, models built on video foundation backbones take the entire clip as input and infer over both hands jointly (Wang et al., 2026; Liu et al., 2026), which notably improves temporal coherence. Even these models, however, regress a hand shape per window, leading to temporal variation in the estimated hand size, while the scale-depth ambiguity of perspective projection allows depth errors to manifest as variations in the estimated shape. RLHND keeps the clip-level formulation but fixes a single shape per video and constrains the pose to the anatomical degrees of freedom, which removes both failure modes.

Hand contact and force estimation. Unlike motion reconstruction, force labels require specialized hardware, *e.g.*, pressure boards (Grady et al., 2022) or tactile gloves (Song et al., 2025). Pressure-VisionDB (Grady et al., 2022) records RGB video of a bare hand pressing a planar capacitive pad, providing dense and finely resolved pressure maps that are spatially limited to the palm-pad contact region. OpenTouch (Song et al., 2025) instead moves the sensor onto the hand, pairing in-the-wild egocentric video with a 16×16 taxel grid on the palmar side of a wearable glove, enabling force measurements during natural manipulation of diverse objects. EgoPressure (Zhao et al., 2025) extends pressure sensing to egocentric video with multi-view MANO fits, while EgoTactile (Zeng et al., 2026) provides full-hand glove pressure on 63 everyday objects and a bare-hand subset. However, differences in sensor calibration and hand-surface correspondence make it difficult to pool these datasets without converting their labels to a common representation. For contact estimation, tactile hardware is not required: HACO (Jung & Lee, 2025) generates hand contact labels from the distances between hand and object meshes in HOI datasets and uses them to train contact estimation models. For force estimation, however, such geometric supervision is insufficient, requiring datasets with physical force measurements. HOPE (Jeon et al., 2026) unifies contact and force labels from pressure boards and tactile gloves on the MANO surface, enabling joint learning of contact and force estimation. RLHND follows the unified label format of HOPE while introducing a decoupled tactile expert, achieving improved tactile estimation performance.

3 Method

Given a monocular video, RLHND estimates both hand pose and tactile signals. The video is processed as a sequence of W clips $\mathcal{C}_1, \ldots, \mathcal{C}_W$, each holding T consecutive frames $\mathcal{C}_w = (\mathbf{I}_t)_{t=1}^{T}$. The pose output consists of MANO (Romero et al., 2017) parameters, *i.e.*, shape $\hat{\boldsymbol{\beta}}^w \in \mathbb{R}^{10}$ and

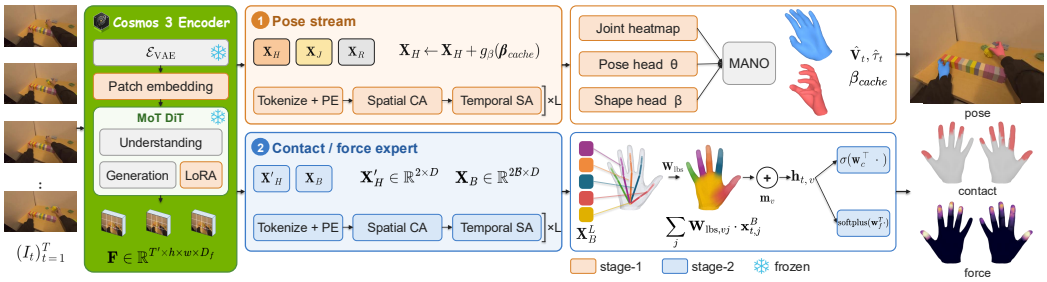

Figure 2: **RLHND overview.** A pre-trained Cosmos 3 backbone receives the clip through its conditioning-frame interface and is adapted with a trainable patch embedding and LoRA adapters. The pose expert stream, trained in stage-1, yields MANO parameters and a metric camera-space translation while the tactile expert stream is trained in stage-2 with the pose stream frozen.

anatomically constrained articulation $\hat{\boldsymbol{\theta}}_t$ with the global wrist rotation, along with metric camera-space translation $\hat{\boldsymbol{\tau}}_t \in \mathbb{R}^3$ and per-frame hand presence and visibility. The tactile output consists of per-vertex tactile signals: contact probability $\hat{\mathbf{c}}_t \in [0,1]^{|V|}$ and force $\hat{\mathbf{f}}_t \in \mathbb{R}_{\geq 0}^{|V|}$, where $|V| = 778$ is the number of MANO vertices. The RLHND architecture consists of three parts: the *clean-latent encoder* converts a pre-trained video-diffusion backbone into a deterministic video feature extractor, the *pose expert stream* predicts pose, shape, and translation in MANO parameter form from the video features, and the *tactile expert stream* estimates contact and force from the same video features with its own weights, trained with the pose stream frozen, thereby decoupling tactile prediction from pose prediction. Figure 2 illustrates the overall process of the proposed method.

3.1 Clean-Latent Video Diffusion Encoder

Since video diffusion models are known to internalize strong priors over hand motion and object interactions (Wang et al., 2026), we use the pre-trained encoder of Cosmos 3 (NVIDIA, 2026). Its video tokenizer, *i.e.*, Wan2.2 VAE (Wan Team, 2025), encodes each clip into a clean latent $\mathbf{z} = \mathcal{E}_{\text{VAE}}(\mathcal{C}_w)$. We fine-tune the patch embedding and generation pathway with LoRA (Hu et al., 2022), yielding a deterministic spatiotemporal feature grid $\mathbf{F} \in \mathbb{R}^{T' \times h \times w \times D_f}$.

Our use of the pre-trained VAE differs from prior work (Wang et al., 2026; Liu et al., 2026), which feeds clean videos to Wan2.2 through its denoising interface by setting the out-of-distribution noise level $\sigma = 0$. Instead, we use the conditioning interface of Cosmos 3 for clean context frames, keeping our input scheme aligned with pre-training. We also feed an empty prompt, since the video stream is already conditioned on the clean frames, whereas Liu et al. (2026) uses a fixed caption.

3.2 Pose Expert Stream

Following ACE-Ego-Hand (Liu et al., 2026), we tokenize the $\mathbf{F}$ with spatial positional and Fourier-encoded ray embeddings, and decode the resulting tokens with alternating spatial cross-attention and bidirectional temporal self-attention. We depart from this design in two aspects: how the two hand tokens $\mathbf{X}_H$ are formed with the shape conditioning, and how hand articulation is parameterized.

Shape conditioning. We construct the two hand tokens $\mathbf{X}_H$ from the cached MANO shape parameter $\boldsymbol{\beta}_{\text{cache}}$, which enforces consistent hand geometry throughout the video and can also be obtained from a one-time calibration of the actor's hand. Unlike ACE-Ego-Hand (Liu et al., 2026), which initializes its hand tokens from fixed learnable parameters, our model can directly incorporate this pre-measured shape. Let v^w denote the fraction of frames in $\mathcal{C}_w$ in which the hand is predicted present. The hand tokens are built from a learnable query $\mathbf{q}_H$ and a zero-initialized MLP g_β:

$$\mathbf{X}_H^w = \mathbf{q}_H + g_\beta(\boldsymbol{\beta}_{\text{cache}}), \qquad \boldsymbol{\beta}_{\text{cache}} = \begin{cases} \hat{\boldsymbol{\beta}}^{w^*}, & w^* = \min\{\, w' : v^{w'} > T_v \,\} \text{ exists}, \\ \hat{\boldsymbol{\beta}}^{w}, & \text{otherwise}, \end{cases} \tag{1}$$

The shape estimated from the first sufficiently visible clip is shared across the video, and earlier clips are decoded again once it becomes available. The zero initialization of g_β gradually introduces

the cached shape during training. We use the ground-truth $\boldsymbol{\beta}$ with probability 0.5 and disable the shape-head loss for these samples. Since the public benchmarks do not provide pre-calibrated hand shapes, we estimate $\boldsymbol{\beta}$ from the first sufficiently visible clip and reuse it throughout each video.

Constrained pose parameterization. For more faithful tracking of the ground-truth hand articulation, we anatomically constrain the DoF of MANO (Yang et al., 2021). While MANO allows 3-axis rotation at each of its 15 joints, resulting in 45 DoF, the human hand has only 21 DoF. We therefore align the rotation axes with the anatomical structure and disable infeasible axes. Following the anatomy-aware kinematics of Yang et al. (2021), the rotation $\mathrm{R}_j \in SO(3)$ of joint j is factorized about its anatomically defined twist, spread, and bend axes $(\mathbf{a}^{\text{twist}}j, \mathbf{a}_j^{\text{spread}}, \mathbf{a}_j^{\text{bend}})$:

$$\mathrm{R}_j \;=\; \exp\!\big(\varphi_j^{\text{twist}}\,[\mathbf{a}_j^{\text{twist}}]_\times\big)\;\exp\!\big(\varphi_j^{\text{spread}}\,[\mathbf{a}_j^{\text{spread}}]_\times\big)\;\exp\!\big(\varphi_j^{\text{bend}}\,[\mathbf{a}_j^{\text{bend}}]_\times\big), \tag{2}$$

where $[\cdot]_\times$ is the skew-symmetric operator. Then, the feasible set $\mathcal{C}$ is defined as below:

$$\mathcal{C} \;=\; \big\{\boldsymbol{\Theta} \;:\; \varphi_j^{\text{twist}} = \varphi_j^{\text{spread}} = 0 \;\; \forall j \in \text{PIP} \cup \text{DIP}\big\}. \tag{3}$$

We leave the thumb unconstrained because its carpometacarpal joint is a saddle joint with non-orthogonal, non-fixed axes, for which the twist-spread-bend decomposition does not isolate an infeasible axis. This leaves 29 DoF after constraining the remaining joints. To train the model with this representation, we obtain constrained 29-DoF labels by fitting the existing 45-DoF MANO labels to the feasible set $\mathcal{C}$ via inverse kinematics (Zhou et al., 2020; Li et al., 2021):

$$\widehat{\boldsymbol{\Theta}}_c \;=\; \underset{\boldsymbol{\Theta}_c \in \mathcal{C}}{\arg\min}\; \big\|\mathrm{J}(\boldsymbol{\Theta}_c;\boldsymbol{\beta}) - \mathrm{J}(\boldsymbol{\Theta};\boldsymbol{\beta})\big\|_2^2, \tag{4}$$

where $\mathrm{J}(\cdot\,;\boldsymbol{\beta})$ returns the forward-kinematics joint positions under the shape $\boldsymbol{\beta}$ carried by the label. We solve Eq. (4) in the anatomy-aligned Euler coordinates using damped Gauss-Newton steps:

$$\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + \big(\mathbf{G}^\top\mathbf{G} + \lambda\mathbf{I}\big)^{-1}\mathbf{G}^\top\boldsymbol{\epsilon}, \tag{5}$$

where $\boldsymbol{\varphi}$ contains the optimized finger DoF, $\mathbf{G}$ is the Jacobian of the joint positions with respect to $\boldsymbol{\varphi}$, and $\boldsymbol{\epsilon}$ stacks the joint and fingertip residuals. Further details are provided in Appendix A.2.

Output Heads. The readout heads decode the hand tokens into pose, shape, depth, and presence predictions. Metric camera-space translation is then recovered using the ray-based mixed-PnP solver of ACE-Ego-Hand (Liu et al., 2026). We refer to Appendix A.3 for details.

3.3 Tactile Expert Stream

Tactile supervision is far scarcer than pose supervision, as pressure-labeled datasets often provide constrained or no pose labels (Grady et al., 2022; 2024; Song et al., 2025). We therefore decouple tactile estimation from pose estimation, preventing such training from affecting pose prediction.

Bone tokens. Attending over all $|V|$ vertices per frame is computationally prohibitive, so we perform tactile estimation at the hand-skeleton level. Let $\mathcal{B}$ denote the $|\mathcal{B}| = 16$ joints of the MANO kinematic tree. Each joint drives one bone of the skeleton through the linear-blend-skinning weights, so we attach one *bone token* per joint and hand, yielding $\mathbf{X}_B \in \mathbb{R}^{2|\mathcal{B}|\times D}$ with $|\mathcal{B}| \ll |V|$. Each bone token represents the features of the vertices skinned to its bone.

LBS-based vertex readout. We keep temporal reasoning at the bone level and expand the features to vertices only at the output using the MANO Linear-Blend-Skinning (LBS) weight $\mathbf{W}_{\mathbf{lbs}} \in \mathbb{R}^{|V|\times|\mathcal{B}|}$, the fixed matrix from MANO that maps features to vertices. Combined with a learned vertex embedding $\mathbf{m}_v$, the per-vertex tactile feature is computed as follows:

$$\mathbf{h}_{t,v} \;=\; \sum_{j\in\mathcal{B}} \mathbf{W}_{\mathbf{lbs},vj}\,\big[\mathbf{X}_B^L\big]_{t,j} \;+\; \mathbf{m}_v, \tag{6}$$

where $[\mathbf{X}_B^L]_{t,j}$ is the token of node j at frame t in $\mathbf{X}_B^L$, obtained by interpolation from T' to T.

Two small MLP heads read the per-vertex contact logit and the force-distribution logit from $\mathbf{h}_{t,v}$, and the per-hand total force $\hat{F}_t$ is read from the expert's hand token through a softplus; the per-vertex force is the total distributed over the vertices, $\hat{f}_{t,v} = \hat{F}_t\,\mathrm{softmax}_v(\cdot)$.

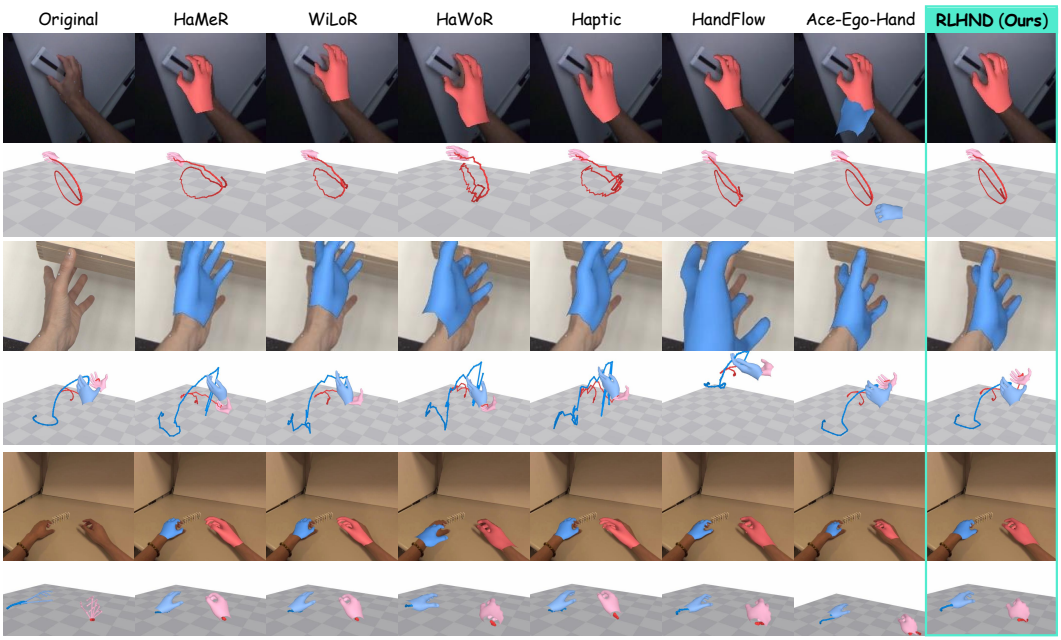

Figure 3: **Qualitative comparison of hand motion reconstruction.** We visualize the reconstruction results from HOT3D (top), ARCTIC (middle), and EgoDex (bottom), with mesh overlays and trajectory visualizations. Extensive visualization results can be found on the project page.

3.4 Training

At stage-1 training, we only train the pose expert stream with the following:

$$\mathcal{L}_{\text{pose}} = \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\beta}\mathcal{L}_{\beta} + \lambda_{\text{3D}}\mathcal{L}_{\text{3D}} + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\tau}\mathcal{L}_{\tau} + \lambda_{\text{pres}}\mathcal{L}_{\text{pres}} + \lambda_{\text{tmp}}\mathcal{L}_{\text{tmp}} + \lambda_{\text{ray}}\mathcal{L}_{\text{ray}}. \quad (7)$$

At stage-2 training, the pose expert stream is frozen, and only the tactile expert stream is trained:

$$\mathcal{L}_{\text{tactile}} = \lambda_c \,\text{BCE}_w(\hat{c}_{t,v},\, c_{t,v}) + \lambda_F \,\frac{1}{|V|}\Big|\sum_v \hat{f}_{t,v} - \sum_v f_{t,v}\Big| + \lambda_\pi \,\text{CE}\Big(\hat{\mathbf{f}}_t / \sum_v \hat{f}_{t,v},\; \mathbf{f}_t / \sum_v f_{t,v}\Big). \quad (8)$$

Since contact vertices are far rarer than non-contact vertices, we use a positively re-weighted BCE, *i.e.*, BCE_w. The force is supervised through its per-hand total, normalized by the number of vertices, and through its distribution over the vertices, weighted by λ_F and λ_π, respectively. We elucidate the details of each loss term in Appendix A.4.

4 Experiments

Setup. Stage-1 is trained on a weighted mixture of egocentric and exocentric hand-video datasets, and stage-2 on force-labeled glove and pressure-pad recordings together with contact-only labels derived from hand–object meshes, which supervise only the contact head. The datasets and label generation are described in Appendix A.2, the schedules and hyperparameters in Appendix A.5, and the metrics in Appendix A.4. Motion reconstruction is evaluated on the held-out HOT3D and ARCTIC egocentric splits and, zero-shot for our method and every baseline, on EgoDex; tactile estimation on the OpenTouch and PressureVisionDB (PVDB) test splits of HOPE (Jeon et al., 2026) and, for contact only, on HOT3D, DexYCB, and ARCTIC with mesh-derived contact labels. All baselines use their official weights and inference code.

4.1 Hand Pose Estimation

In Figure 3, crop-based models exhibit noisy hand trajectories and incorrect mesh overlays, while ACE-Ego-Hand performs better but still shows erroneous hand detection (the left hand in the first

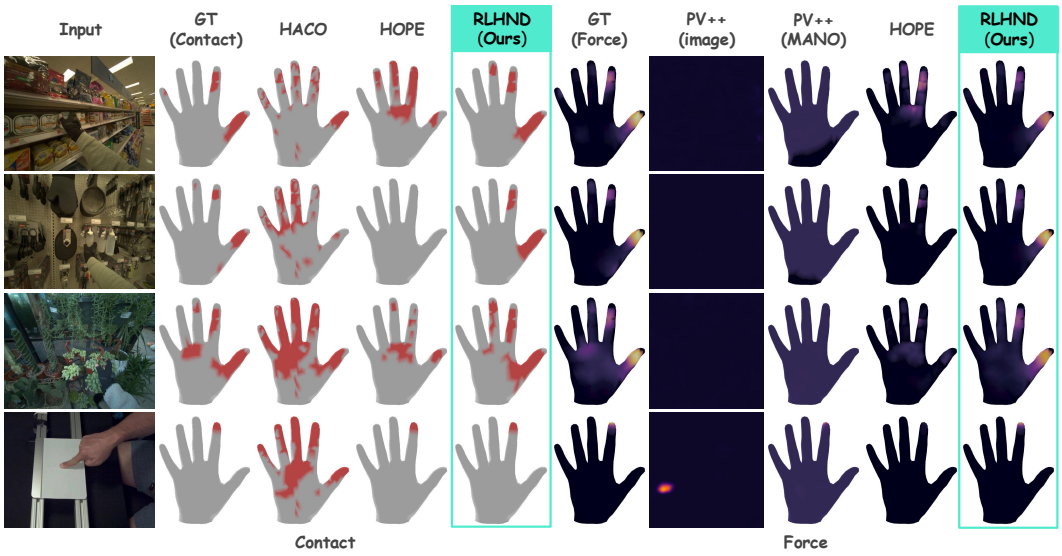

Figure 4: **Qualitative comparison of contact and force estimation.** We visualize the *contact* (left) and *force* (right) estimation from OpenTouch (first to third row) and PVDB (fourth row).

Test set	Method	F_{Acc} ↑	MPJPE-p ↓	PA-p ↓	MPJPE^{+OOS} ↓	EPE2D-p ↓	Jitter ↓	σ_{shape} ↓
HOT3D	HaMeR (Pavlakos et al., 2024)	0.743	65.53	11.42	66.85	130.33	25.40	1.17
	WiLoR (Potamias et al., 2024)	0.743	44.89	9.71	46.23	134.89	22.17	1.90
	HaWoR (Zhang et al., 2025)	0.743	41.53	10.15	42.99	126.10	26.13	7.37
	HaPTIC (Ye et al., 2025b)	0.743	62.42	11.26	63.70	130.40	38.27	3.20
	HandFlow (Xu et al., 2026)	0.729	39.90	9.50	24.87	127.85	16.99	3.68
	ACE-Ego-Hand (Liu et al., 2026)	0.928	24.41	8.60	20.30	38.09	6.48	1.22
	RLHND (Ours)	**0.996**	13.01	6.72	13.08	**6.84**	**4.38**	1.30
	RLHND (Ours) (β-cache)	**0.996**	**13.00**	**6.68**	**13.07**	6.89	4.39	**0.00**
ARCTIC ego	HaMeR (Pavlakos et al., 2024)	0.887	31.76	9.72	52.99	86.30	21.73	0.52
	WiLoR (Potamias et al., 2024)	0.887	24.05	7.49	46.33	87.66	25.92	6.15
	HaWoR (Zhang et al., 2025)	0.886	31.72	9.55	52.88	82.72	29.34	5.44
	HaPTIC (Ye et al., 2025b)	0.886	24.72	7.92	46.88	84.05	51.09	6.19
	HandFlow (Xu et al., 2026)	0.918	40.03	9.91	47.47	74.09	13.47	5.12
	ACE-Ego-Hand (Liu et al., 2026)	**0.994**	14.86	7.50	16.86	**11.76**	**3.52**	1.91
	RLHND (Ours)	0.993	**13.36**	**6.42**	**15.40**	12.19	4.19	0.96
	RLHND (Ours) (β-cache)	0.993	13.44	6.44	15.46	12.28	4.17	**0.00**
EgoDex	HaMeR (Pavlakos et al., 2024)	0.970	45.97	11.79	45.90	66.31	15.48	0.43
	WiLoR (Potamias et al., 2024)	0.970	42.57	10.87	42.50	67.68	13.05	0.22
	HaWoR (Zhang et al., 2025)	0.970	27.76	11.22	27.73	54.20	16.94	4.58
	HaPTIC (Ye et al., 2025b)	0.970	40.92	11.29	40.85	65.38	25.81	2.33
	HandFlow (Xu et al., 2026)	0.978	24.60	11.45	22.34	41.44	10.89	3.57
	ACE-Ego-Hand (Liu et al., 2026)	0.999	20.95	10.99	20.85	22.26	3.01	0.59
	RLHND (Ours)	**1.000**	**19.90**	**10.12**	**19.89**	**20.72**	**2.93**	0.71
	RLHND (Ours) (β-cache)	**1.000**	19.93	10.14	19.91	**20.72**	2.94	**0.00**

Table 1: **Quantitative comparison of motion reconstruction.** RLHND outperforms baselines on nearly every metric on every benchmark consistently. We denote **best** and second best values.

row), unrealistic motion (twisted fingers in the second row), and inaccurate hand scale (the third-row overlay). In contrast, ours produces robust mesh overlays and smooth hand trajectories.

Table 1 reports quantitative comparisons with per-window β estimation and β-caching, which estimates β from the first output window and reuses it throughout the video. Ours consistently outperforms across all test sets, including zero-shot EgoDex, with higher F_{Acc} than the external hand detectors used by HaMeR, WiLoR, HaWoR, and HaPTIC, as well as strong 3D and 2D projection metrics. Although β-caching removes σ_{shape}, it does not improve jitter, as analyzed in Appendix B.2.

Method	Contact (vertex-level)								Force (kPa)			
	OpenTouch		DexYCB		HOT3D		ARCTIC		OpenTouch		PVDB	
	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
PressureVision (Grady et al., 2022)	–	–	–	–	–	–	–	–	1.930	6.190	0.471	3.726
PressureVision++ (Grady et al., 2024)	–	–	–	–	–	–	–	–	1.920	6.200	0.248	3.025
HACO (Jung & Lee, 2025)	0.373	0.541	0.543	0.883	0.244	0.749	0.580	0.907	–	–	–	–
HOPE (Jeon et al., 2026)	0.663	0.894	0.506	0.868	0.197	0.762	0.591	0.914	1.781	5.388	**0.236**	2.091
RLHND (Ours)	**0.696**	**0.980**	**0.572**	**0.915**	**0.589**	**0.959**	**0.602**	**0.935**	**0.489**	**2.508**	0.274	**2.066**

Table 2: **Quantitative comparison of contact and force estimation.** RLHND outperforms baselines on nearly every benchmark consistently. We denote **best** and second best values.

Method	Sharpa Wave (22 DoF)		WUJI v2 (20 DoF)		Shadow (24 DoF)		Inspire RH56 (12 DoF)		ALLEX (15 DoF)	
	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓
Oracle (GT joints)	0.0	0.34	0.0	0.33	0.0	0.35	0.0	0.18	0.0	0.36
HaMeR (Pavlakos et al., 2024)	16.6	1.45	19.4	1.31	12.6	1.31	8.1	0.90	14.1	1.19
WiLoR (Potamias et al., 2024)	14.2	1.48	17.0	1.34	10.2	1.20	6.3	0.90	11.2	1.18
HaWoR (Zhang et al., 2025)	15.0	0.61	18.0	0.47	12.2	0.50	8.9	0.39	14.4	0.50
HaPTIC (Ye et al., 2025b)	14.9	1.39	17.6	1.27	11.9	1.10	7.0	0.87	12.7	1.21
HandFlow (Xu et al., 2026)	18.7	0.58	21.8	0.55	14.2	0.52	9.9	0.49	17.8	0.53
ACE-Ego-Hand (Liu et al., 2026)	14.0	0.20	17.4	0.21	10.8	0.21	7.4	0.11	12.0	0.20
RLHND (Ours)	**12.7**	0.18	**16.1**	**0.16**	9.6	**0.19**	**6.0**	**0.07**	**10.2**	**0.17**
RLHND (Ours) (β-cache)	12.8	**0.17**	16.3	0.17	**9.5**	0.20	**6.0**	**0.07**	10.3	**0.17**

Table 3: Retargeting quality on five dexterous right hands (DexPilot, wrist-relative targets), averaged over the HOT3D, ARCTIC ego and EgoDex test sets; per-dataset results are in Appendix B.4. Q-err is the mean absolute deviation (°) of the joint command from the one the same solver produces on the ground-truth joints, and Jerk the median frame-to-frame command noise (°/frame^2); see Appendix A.4. The grey Oracle row retargets the ground-truth joints.

4.2 Tactile Estimation

Figure 4 compares predicted contact and pressure maps on OpenTouch and PressureVisionDB. RLHND produces localized contact regions and accurate pressure patterns on the hand surface, whereas HACO over-predicts contact and HOPE produces fragmented contact regions with underestimated pressure. PressureVision++ predicts pressure only in its native planar-sensor representation.

Table 2 quantifies contact (F1 and AUROC) on OpenTouch, HOT3D, DexYCB, and ARCTIC and force on OpenTouch and PVDB; the PressureVision family (Grady et al., 2022; 2024) predicts pressure on the image plane rather than the hand surface, so its contact metrics are not reported. RLHND achieves the best contact scores across all four datasets in both metrics, with competitive force estimation.

4.3 Effectiveness on Robot Learning

Real-robot experiments. Dexterous Point Policy (Kim et al., 2026a), *i.e.*, DPP, trains robot policies from human demonstrations using keypoint locations and fingertip contact labels for force application, with the latter *manually* annotated for each demonstration. RLHND provides both *automatically* from a single model, with fingertip keypoints from its pose stream and contact labels from its contact head. Table 4 compares rollout results with DPP + RLHND; since the original tracker predicts no contact, both rows use RLHND's contact labels and differ only in the hand keypoints. DPP + RLHND improves performance across the evaluated tasks, with larger gains on the complex bimanual tasks.

Retargeting to robot hands. We evaluate how well tracker outputs serve as robot action spaces in Table 3. We retarget per-frame predictions to five dexterous right hands using the same DexPilot (Handa et al., 2020) solver with wrist-relative targets, and report Q-err and joint jerk averaged over HOT3D, ARCTIC ego, and EgoDex. RLHND, with or without the β-cache, achieves the lowest mean Q-err across all five embodiments and produces among the smoothest commands.

Method	Human : Robot episodes	Pick and Place					Bimanual			All ↑
		Bottle	Box	Ball	Bird	Avg. ↑	Plastic bag	Assemble tissue	Avg. ↑	
DPP (Kim et al., 2026a)	100 : 100	**93.8**	**100.0**	81.3	81.3	89.1	53.1	75.0	64.1	80.8
DPP + RLHND (Ours)	100 : 100	**93.8**	**100.0**	**87.5**	**93.8**	**93.8**	**59.4**	**90.6**	**75.0**	**87.5**

Table 4: **Real-robot success rate of Dexterous Point Policy.** DPP is trained on human demonstrations labeled by its original tracker, and DPP + RLHND on the same demonstrations labeled by RLHND.

Pose variant	ARCTIC ego				EgoDex			
	MPJPE-p ↓	PA-p ↓	EPE2D-p ↓	Q-err ↓	MPJPE-p ↓	PA-p ↓	EPE2D-p ↓	Q-err ↓
A0: w/o Cosmos 3	16.62	7.55	13.40	12.48	21.57	10.63	22.45	15.64
A1: w/o constrained pose	14.26	6.52	12.33	10.85	19.91	10.38	20.76	15.31
A2: w/o β-conditioning	14.87	6.72	12.27	10.48	20.06	10.33	20.95	15.29
A3: Full model	13.36	**6.42**	**12.19**	10.12	**19.90**	**10.12**	**20.72**	**15.22**
A4: Full model + GT β	**13.15**	6.45	12.21	**9.69**	–	–	–	–

Tactile variant	DexYCB		HOT3D		ARCTIC		OpenTouch force (kPa)	
	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	MAE ↓	RMSE ↓
B0: w/o Cosmos 3	0.549	0.900	0.548	0.940	0.557	0.919	0.490	2.686
B1: w/o LBS spread	0.532	0.892	0.569	0.948	0.557	0.913	0.537	2.623
B2: w/o contact-only data	0.047	0.683	0.145	0.616	0.117	0.755	0.500	2.524
B3: Full tactile expert	**0.572**	**0.915**	**0.589**	**0.959**	**0.602**	**0.935**	**0.489**	**2.508**

Table 5: **Leave-one-out ablations of RLHND.** *Top (pose):* variants A0-A4, evaluated on ARCTIC ego and EgoDex. *Bottom (tactile):* variants B0–B3, evaluated on contact benchmarks and OpenTouch.

4.4 Ablation Studies

We ablate the three components that distinguish RLHND from its base tracker by removing each from the full model and retraining under the same 20k-step schedule and data mixture (Table 5). Variant (A0) replaces the Cosmos 3 Nano encoder with the Wan2.2 backbone following ACE-Ego-Hand, (A1) replaces the anatomically constrained parameterization of Section 3.2 with 16 free 6D rotations supervised on the raw 45-DoF MANO labels, and (A2) disables external-β injection and teacher forcing, so shape is predicted by the shape head alone, without the one-time hand calibration of Section 3.2, as in the full model (A3).

The encoder is the dominant factor, with A0 degrading performance across all metrics and test sets, confirming that a strong video backbone provides substantial gains. Removing either the anatomically constrained pose representation (A1) or shape caching (A2) also degrades MPJPE in ARCTIC, while their combination yields a larger improvement in the full model. The full model with ground-truth β (A4) gains nothing in joint-level metrics but improves Q-err, suggesting that fixed hand shape may stabilize retargeting.

Each tactile variant retrains the stage-2 expert on the frozen full pose model: B0 uses the Wan2.2 stage-1 model of A0, B1 replaces the fixed MANO skinning weights $\mathbf{W}_{\mathbf{lbs}}$ of Eq. (6) with a learnable 778×16 spread matrix, and B2 trains on OpenTouch and PVDB alone. B0 and B1 both degrade performance, while removing the contact-only data (B2) causes a far larger contact drop with no notable change in force.

5 Conclusion

We presented RLHND, a physically grounded hand tracker built on a pre-trained video foundation model for robot learning. It achieves state-of-the-art motion reconstruction and tactile estimation, produces smooth retargeted commands across five dexterous hands, and lets Dexterous Point Policy outperform its original tracker without manual annotation.

REPRODUCIBILITY STATEMENT

The architecture and training procedure are specified in Section 3.1–3.4, with implementation details, label construction, losses, evaluation metrics, training schedules, and baseline protocols provided in Appendix A. All training and evaluation data come from publicly available datasets, and the retargeting solver and its settings are described in Section 4.3. We will release the implementation of RLHND, along with the pre-trained weights.

ETHICS STATEMENT

RLHND uses only publicly released human hand-object datasets under their respective licenses and consent procedures. We collect no new human data and release no raw video. Since the method can recover fine-grained hand activity from ordinary footage, including egocentric recordings containing bystanders, we intend it for robot learning from consenting demonstrators and discourage surveillance use. Contact and force are inferred rather than measured, so we recommend joint-torque or current limits and human supervision when deploying retargeted commands. Finally, the limited diversity of public datasets may lead to performance variation across subjects, hand sizes, and skin tones that is not captured by our evaluation sets. The authors declare no conflicts of interest.

AI USE STATEMENT

We used generative AI tools to assist with figure and table editing, and routine coding. AI-assisted code was reviewed and tested by the authors, and all reported results were independently produced and verified by the authors. All AI-assisted text and figures were reviewed against the underlying experiments and data. The authors take full responsibility for the content of this paper.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. Introducing HOT3D: An egocentric dataset for 3D hand and object tracking. *arXiv preprint arXiv:2406.09598*, 2024. 1, 17, 18, 21

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 1

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 1

Xiongyi Cai, Ri-Zhao Qiu, Geng Chen, Lai Wei, Isabella Liu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. In-N-On: Scaling egocentric manipulation with in-the-wild and on-task data. *arXiv preprint arXiv:2511.15704*, 2025. 1

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. 23, 26

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. DexYCB: A benchmark for capturing

hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9044–9053, 2021. 16, 17, 18, 21

Baijun Chen, Weijie Wan, Tianxing Chen, Xianda Guo, Congsheng Xu, Yuanyang Qi, Haojie Zhang, Longyan Wu, Tianling Xu, Zixuan Li, Yizhe Wu, Rui Li, Xiaokang Yang, Ping Luo, Wei Sui, and Yao Mu. UniVTAC: A unified simulation platform for visuo-tactile manipulation data generation, learning, and benchmarking. *arXiv preprint arXiv:2602.10093*, 2026. 2

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12943–12954, 2023. 16, 17, 18, 21

Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D Twigg, Chengde Wan, James Hays, and Charles C Kemp. PressureVision: Estimating hand pressure from a single RGB image. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 5, 8, 19

Patrick Grady, Jeremy A Zhao, Chengcheng Tang, Aditya Kumar, Christopher D Twigg, Yafei Ye, Etienne Vouga, and Charles C Kemp. PressureVision++: Estimating fingertip pressure from diverse RGB images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 8698–8708, 2024. 5, 8

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

Siddhant Haldar and Lerrel Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. In *Conference on Robot Learning*, 2025. 1

Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3D annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3196–3206, 2020. 17, 18

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *IEEE International Conference on Robotics and Automation*, 2020. 1, 8, 23

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations*, 2026. 1, 17

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4

Subin Jeon, Byungjun Kim, and Hanbyul Joo. HOPE: Hand-object pressure estimation from monocular videos. *arXiv preprint arXiv:2608.06192*, 2026. 2, 3, 6, 8, 18, 22, 23, 33

Daniel Jung and Kyoung Mu Lee. Learning dense hand contact estimation from imbalanced data. *Advances in Neural Information Processing Systems*, 38:120351–120384, 2025. 2, 3, 8, 18, 22, 23

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. *arXiv preprint arXiv:2410.24221*, 2024. 1

Beomjun Kim, Seong Hyeon Park, Seunghoon Sim, Seungjun Moon, Sanghyeok Lee, and Jinwoo Shin. Dexterous point policy: Learning point-based dexterous hand policies from human demonstrations. *arXiv preprint arXiv:2606.10614*, 2026a. 1, 2, 8, 9

Dongyoung Kim, Huiwon Jang, Myungkyu Koo, et al. RLDX-1 technical report. *arXiv preprint arXiv:2605.03269*, 2026b. 1

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10138–10148, 2021. 17

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. In *Conference on Robot Learning*, 2025. 1

Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3383–3393, 2021. 5

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, Yizhong Zhang, Xi Chen, Hao Chen, Lily Sun, Dong Chen, Jiaolong Yang, and Baining Guo. VITRA: Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025. 1

Jongbin Lim, Taeyun Ha, Mingi Choi, Jisoo Kim, Byungjun Kim, Subin Jeon, and Hanbyul Joo. HRDexDB: A paired human-robot dataset for cross-embodiment dexterous grasping. *arXiv preprint arXiv:2604.14944*, 2026. 17, 18

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023. 1

Vincent Liu, Ademi Adeniji, Haotian Zhan, Siddhant Haldar, Raunaq Bhirangi, Pieter Abbeel, and Lerrel Pinto. EgoZero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025. 1

Yufei Liu, Xixi Wang, Hao Li, Ganlong Zhao, Kaitong Cai, Chengkai Jin, Chunxiao Liu, Jianbo Liu, Siyuan Huang, Xingang Pan, and Hongsheng Li. ACE-Ego-Hand: Repurposing video diffusion models for occlusion-robust egocentric 3d hand motion recovery, 2026. URL https://arxiv.org/abs/2608.20308. 2, 3, 4, 5, 7, 8, 19, 20, 23, 33

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025. 1

Dantong Niu, Zhuoyang Liu, Zekai Wang, Boning Shao, Zhao-Heng Yin, Anirudh Pai, Yuvan Sharma, Stefano Saravalle, Ruijie Zheng, Jing Wang, Ryan Punamiya, Mengda Xu, Yuqi Xie, Yunfan Jiang, Letian Fu, Konstantinos Kallidromitis, Matteo Gioia, Junyi Zhang, Jiaxin Ge, Haiwen Feng, Fabio Galasso, Wei Zhan, David M. Chan, Yutong Bai, Roei Herzig, Jiahui Lei, Li Fei-Fei, Ken Goldberg, Jitendra Malik, Pieter Abbeel, Yuke Zhu, Danfei Xu, Linxi Fan, and Trevor Darrell. T-Rex: Tactile-reactive dexterous manipulation. *arXiv preprint arXiv:2606.17055*, 2026. 2

NVIDIA. Cosmos 3: A mixture-of-transformers omni foundation model for physical ai. *arXiv preprint arXiv:2606.02800*, 2026. 2, 4

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 1

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9826–9836, 2024. 3, 7, 8, 23, 33

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 1

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild, 2024. 3, 7, 8, 23, 33

Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. AnyTeleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems*, 2023. 1

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):1–17, 2017. 3

Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. VideoDex: Learning dexterity from internet videos. In *Conference on Robot Learning*, 2022. 1

Yuxin Ray Song, Jinzhou Li, Rao Fu, Devin Murphy, Kaichen Zhou, Rishi Shiv, Yaqi Li, Haoyu Xiong, Crystal Elaine Owens, Yilun Du, et al. OpenTouch: Bringing full-hand touch to real-world interaction. *arXiv preprint arXiv:2512.16842*, 2025. 2, 3, 5, 18, 21

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 4

Yuxi Wang, Chengkai Jin, Yufei Liu, Wenqi Ouyang, Tianyi Wei, Zhiwei Zeng, Siyuan Huang, Zhiqi Shen, and Xingang Pan. The surprising effectiveness of video diffusion models for hand motion reconstruction. *arXiv preprint arXiv:2606.30308*, 2026. 2, 3, 4

Bowen Wen, Shaurya Dewan, and Stan Birchfield. Fast-foundationstereo: Real-time zero-shot stereo matching. *arXiv preprint arXiv:2512.11130*, 2025a. 23, 26

Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025b. 26

Mingxin Xu et al. Handflow: Fully generative 4d hand recovery with flow matching. https://github.com/mxxu00/HandFlow, 2026. V1 open-source release. 3, 7, 8, 23, 33

Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11097–11106, 2021. 5

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Hongxu Yin, Sifei Liu, et al. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025. 1

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025a. 1

Yufei Ye, Yao Feng, Omid Taheri, Haiwen Feng, Shubham Tulsiani, and Michael J. Black. Predicting 4D hand trajectory from monocular videos, 2025b. 3, 7, 8, 23, 33

Yuan Zeng, Yujia Shi, Tiao Tan, Xingting Li, Yaqi Qin, Zongqing Lu, Wenming Yang, Jing-Hao Xue, and Qingmin Liao. EgoTactile: Learning grasp pressure for everyday objects from egocentric video. In *International Conference on Machine Learning*, 2026. 3

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OakInk2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 445–456, 2024. 17, 18

Chi Zhang, Penglin Cai, Ziheng Xi, Haoqi Yuan, Hao Luo, Wanpeng Zhang, Sipeng Zheng, Chaoyi Xu, and Zongqing Lu. Human-centric transferable tactile pre-training for dexterous robotic manipulation. *arXiv preprint arXiv:2607.01067*, 2026a. 2, 17

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1805–1815, 2025. 3, 7, 8, 23, 33

Xidong Zhang, Yichi Zhang, Jiaxin Shi, Fucai Zhu, Siyu Zhu, Michael Yu Wang, Xiaojun Wu, and Weihao Yuan. UniTacVLA: Unified tactile understanding and prediction in vision language action models. *arXiv preprint arXiv:2606.31723*, 2026b. 2

Yiming Zhao, Taein Kwon, Paul Streli, Marc Pollefeys, and Christian Holz. EgoPressure: A dataset for hand pressure and pose estimation in egocentric vision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 3, 34

Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5346–5355, 2020. 5

APPENDIX CONTENTS

A Implementation Details **16**

A.1 β-cache Inference Procedure 16

A.2 Pseudo-Label Generation 16

A.3 Ray-Based Camera Solver 19

A.4 Objective Function and Evaluation Metrics 19

A.5 Training and Evaluation 22

B Additional Experimental Results **27**

B.1 Real-Robot Experiments with DPP 27

B.2 Effect of the β-Cache on Jitter 28

B.3 Inference Cost 29

B.4 Per-Dataset Retargeting Results 29

B.5 Additional Qualitative Results 30

B.6 Additional Ablation Results 30

C Limitations **34**

A Implementation Details

A.1 $\boldsymbol{\beta}$-cache Inference Procedure

Algorithm 1 $\boldsymbol{\beta}$-cache inference for one video, run per hand

Require: clips $\mathcal{C}_1, \ldots, \mathcal{C}_W$; visibility threshold T_v; decoder $\mathcal{D}$

$\boldsymbol{\beta}_{\text{cache}} \leftarrow \emptyset,\ \mathcal{P} \leftarrow [\,]$ ▷ no shape yet; no clips held back
for $w = 1$ **to** W **do**
  if $\boldsymbol{\beta}_{\text{cache}} \neq \emptyset$ **then**
    output $\mathcal{D}(w \mid \boldsymbol{\beta}_{\text{cache}})$ ▷ shape known: decode once
  else
    $(\hat{\boldsymbol{\beta}}^w, v^w) \leftarrow \mathcal{D}(w)$ ▷ probe: shape-head output and visible fraction
    if $v^w > T_v$ **then**
      $\boldsymbol{\beta}_{\text{cache}} \leftarrow \hat{\boldsymbol{\beta}}^w$ ▷ $\mathcal{C}_w$ is the first well-visible clip, *i.e.*, w^*
      for all $u \in \mathcal{P} \cup \{w\}$ **do**
        output $\mathcal{D}(u \mid \boldsymbol{\beta}_{\text{cache}})$ ▷ re-decode the held-back clips
      end for
      $\mathcal{P} \leftarrow [\,]$
    else
      append w to $\mathcal{P}$ ▷ hold back until the shape is known
    end if
  end if
end for
for all $u \in \mathcal{P}$ **do**
  output $\mathcal{D}(u \mid \hat{\boldsymbol{\beta}}^u)$ ▷ no clip qualified: fall back to per-clip shapes
end for

Algorithm 1 details the inference-time shape protocol of Section 3.2. Videos are decoded clip by clip until the first sufficiently visible clip $\mathcal{C}_{w^*}$ is found, whose shape-head output becomes $\boldsymbol{\beta}_{\text{cache}}$ and is reused for all clips; previously probed clips are then decoded again with the cached shape. If no clip qualifies, each clip retains its own shape estimate, corresponding to the per-clip setting in Table 1. With an externally calibrated shape, the algorithm starts with $\boldsymbol{\beta}_{\text{cache}}$ initialized and decodes every clip once without probing. Figure 5 illustrates this procedure on a test recording, where the first well-visible window supplies the shape shared by the entire video.

A.2 Pseudo-Label Generation

Pose label conversion. Table 6 compares solvers for converting the original 45-DoF MANO labels into the constrained set $\mathcal{C}$ of Eq. (3) by minimizing the fitting objective in Eq. (4). All solvers start from the same projection onto $\mathcal{C}$ and optimize the same 20 active finger DoF, *i.e.*, the three axes of each non-thumb MCP joint and the bend of each non-thumb PIP and DIP joint. We evaluate the mean per-joint distance between the converted pose and the original label over the 15 hand joints and 5 fingertips, using the label's shape $\boldsymbol{\beta}$; frames come from DexYCB (Chao et al., 2021) and ARCTIC (Fan et al., 2023), and the cost column reports wall-clock time relative to ours on the same frames.

Adam with many steps is the most accurate solver, *e.g.*, 200 steps reduce the median error to 0.34 mm on DexYCB and 0.65 mm on ARCTIC, but at 283× our cost. Since the conversion runs over every frame of every training set, we exclude it because such preprocessing cost would hinder further scaling of the training corpus. Among the remaining solvers, we first discard those with unacceptable worst-case errors, *i.e.*, the undamped Gauss–Newton step, whose near-singular Jacobian at extended fingers raises the 99th percentile on ARCTIC to 13.2 mm and the worst-frame error to 25.0 mm. We then discard cyclic coordinate descent and the root-to-tip sequential solve, which match our error within 0.05 mm but cost 3× and 8× as much. This leaves the damped Gauss–Newton step of Eq. (5), which we run for three iterations at $\lambda{=}10^{-3}$ with a finite-difference Jacobian. The fit is insensitive to the damping value between 10^{-3} and 10^{-5}, and retaining the fingertips in the residual reduces errors at contact-relevant vertices, *e.g.*, from 0.66 to 0.47 mm on DexYCB. Each converted label stores its residual, which is 0.5-1 mm across the corpus.

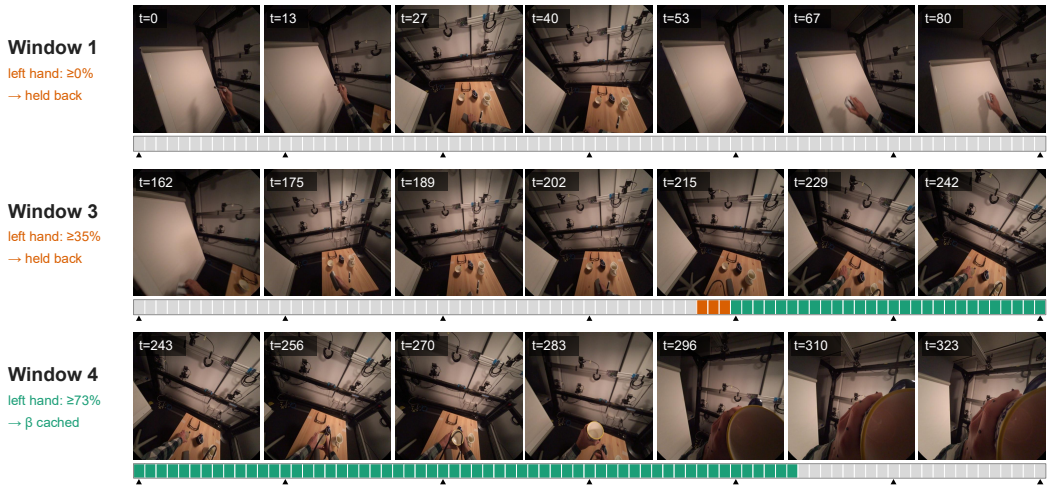

Figure 5: **β-cache inference on a test recording.** Left hand of a HOT3D Aria test recording decoded in 81-frame windows. Seven evenly spaced frames are shown per window, with the strip below marking frame-level presence or absence and triangles indicating the displayed frames. Windows whose visibility stays under T_v are held back. The left hand first exceeds T_v in window 4, so its shape becomes $\boldsymbol{\beta}_{\text{cache}}$ and windows 1–3 are decoded again; the right hand is cached from window 1.

Figure 6: **Contact label differences between distance threshold and signed distance.** Contact labels from the conventional unsigned distance (middle) and our signed distance (right) on two DexYCB grasps. We denote the contacted vertices with the red mark.

We convert only the MANO parameters and leave all 2D and 3D keypoint labels unchanged. Most of our sources obtain 3D keypoints first, *e.g.*, from marker-based motion capture, multi-view triangulation, or a head-mounted device, and fit MANO to these keypoints afterwards. Thus, the anatomically infeasible twist and spread angles in the original MANO labels arise from the unconstrained MANO fitting rather than from the keypoint annotations themselves, as the fitting process can use the extra DoF of MANO to absorb residual keypoint error. Refitting the keypoints to the constrained MANO would moreover alter the ground truth against which every method is evaluated, reducing benchmark reproducibility and potentially tailoring the evaluation to our parameterization. Keeping the original keypoints therefore ensures that all comparisons in Section 4 remain against the datasets as released.

The conversion is applied to every stage-1 training source, *i.e.*, ARCTIC (Fan et al., 2023), HOT3D (Banerjee et al., 2024), H2O (Kwon et al., 2021), OakInk2 (Zhan et al., 2024), DexYCB (Chao et al., 2021), HO3D (Hampali et al., 2020), and HRDexDB (Lim et al., 2026). Motion reconstruction is evaluated against the released keypoints on 210k hand frames from the HOT3D (Banerjee et al., 2024) test split, 53k from the ARCTIC (Fan et al., 2023) egocentric validation split, and 174k from the EgoDex (Hoque et al., 2026) test split; EgoDex is unseen during training by our method and by every baseline.

Contact label extraction. We derive dense per-vertex contact labels from hand-object interaction datasets that provide ground-truth hand and object meshes. Let $\mathbf{x}_{t,v} \in \mathbb{R}^3$ denote the world-space position of MANO vertex $v \in V$ at frame t, and let $\mathcal{S}_t^{\text{obj}}$ denote the posed object surface. A natural choice is the *unsigned* point-to-surface distance used by distance-based contact annotation (Zhang

Step rule	λ	Iters	DexYCB Median ↓	DexYCB P99 ↓	DexYCB Max ↓	ARCTIC Median ↓	ARCTIC P99 ↓	ARCTIC Max ↓	Cost ↓
None (projection only)	–	–	1.03	2.34	3.23	1.98	3.96	7.85	0.1×
First-order descent	–	3	1.01	2.32	3.20	1.95	3.91	7.80	1×
Cyclic coordinate descent	–	3	0.42	**0.77**	1.43	0.91	1.54	4.41	3×
Root-to-tip sequential	10^{-3}	3	0.47	0.80	1.43	0.93	1.46	**4.34**	8×
Gauss–Newton	0	3	**0.34**	1.27	9.88	**0.72**	13.21	24.96	1×
Ours (damped Gauss–Newton)	10^{-3}	3	0.47	0.80	**1.41**	0.92	**1.44**	5.07	1×
Adam	–	50	0.37	0.70	1.46	0.83	1.36	4.62	69×
Adam	–	200	0.34	0.63	1.15	0.65	1.16	2.08	283×

Table 6: **Constrained-label fitting.** Label error (mm), *i.e.*, the mean per-joint distance between the constrained pose and the original label over the 15 hand joints and the 5 fingertips, on 4057 DexYCB and 4089 ARCTIC frames. Every refinement starts from the same projection and runs the same number of iterations, so the rows differ only in the step rule. The undamped solve attains a competitive median but its worst frames degrade by an order of magnitude, which is the failure mode that matters when the output is a training label; damping bounds it. Cost is wall-clock time relative to ours on the same frames and hardware. We denote **best** and second best values.

et al., 2026a; Jung & Lee, 2025): a vertex is considered in contact when it lies within δ_c of the object surface. However, mocap-fitted hand meshes frequently interpenetrate the object. A vertex that penetrates deeper than δ_c can therefore be *farther* than δ_c from the object surface, causing the unsigned rule to incorrectly label it as non-contact. As Figure 6 shows for two DexYCB grasps, this can erase fingertips where the grip is firmest and fragment the contact region; in the right example, the unsigned rule misses 253 of the 364 ground-truth contact vertices. We therefore use the *signed* distance

$$d_{t,v} \;=\; \pm\,\mathrm{dist}\Big(\mathbf{x}_{t,v},\, \mathcal{S}_t^{\mathrm{obj}}\Big)\,, \tag{9}$$

where the sign is negative if $\mathbf{x}_{t,v}$ lies inside the object, *i.e.*, if a pseudo-normal inside-outside test marks it as interior, and label

$$c_{t,v} \;=\; \mathbf{1}[\,d_{t,v} \le \delta_c\,]\,, \qquad \delta_c = 5\,\mathrm{mm}. \tag{10}$$

A penetrating vertex satisfies $d_{t,v} < 0 \le \delta_c$ and is therefore labeled as contact regardless of penetration depth, recovering the clean, complete contact regions shown on the right of Figure 6. We store the continuous signed distances along with $d_{t,v}$, which enables δ_c to be tuned at training time.

These geometric labels supervise only the contact head in stage-2 and come from the sources that release object meshes, *i.e.*, ARCTIC (Fan et al., 2023), HOT3D (Banerjee et al., 2024), DexYCB (Chao et al., 2021), HO3D (Hampali et al., 2020), HRDexDB (Lim et al., 2026), and OakInk2 (Zhan et al., 2024); they also serve as the contact ground truth of the DexYCB test, HOT3D test, and ARCTIC egocentric validation splits, which carry no force.

Force label extraction. For the two force-labeled sources, we follow the label revision of HOPE (Jeon et al., 2026), so that our training targets and the test ground truth in Table 2 use a consistent convention. On OpenTouch (Song et al., 2025), the glove provides a 16×16 taxel grid with 169 active taxels. We lift the readings onto the MANO surface using the calibrated taxel-to-vertex correspondence $\mathcal{T}(v)$: each taxel value is assigned to every vertex it covers, while vertices covered by multiple taxels take their mean,

$$\bar{p}_{t,v} \;=\; \frac{1}{|\mathcal{T}(v)|} \sum_{i \in \mathcal{T}(v)} \pi_{t,i}, \tag{11}$$

which covers 205 of the 778 vertices. The remaining vertices are unsensed and are excluded from both the contact and force losses using a per-vertex mask.

Since the glove signal drifts and does not return to zero between grasps, HOPE gates it using a per-clip frame state. Specifically, the total pressure is tracked against an adaptive floor with hysteresis and morphological cleaning; frames below the floor are labeled as no-contact and assigned zero

force, while no-contact frames within three frames of a transition are excluded from the contact loss as ambiguous. Training uses the 1469 clips in the authors' allowlist, while the test split uses their human-verified frame states on the raw glove pressure, *i.e.*, the ground truth used by their evaluation code.

On PressureVisionDB (Grady et al., 2022), we use the vertex-level pressure targets derived by the HOPE authors from the pressure pad and their MANO fits, restricted to the 421 palm-side vertices that can contact the pad; frames without a target are left unsupervised. Both sources use kPa as the force unit, and force-derived contact is defined as $c_{t,v} = \mathbf{1}[\,\bar{p}_{t,v} > 1\,\mathrm{kPa}\,]$ within the sensed vertex set.

OpenTouch and PressureVisionDB are the two force-labeled sources of stage-2, used together with the contact-only sources above, and tactile estimation is evaluated on their test splits as defined by HOPE.

A.3 Ray-Based Camera Solver

A zero-initialized linear *ray head* on the tapped feature grid $\mathbf{F}$ predicts a unit viewing ray per cell, which is averaged over the T' latent frames to obtain one ray field $\hat{\mathbf{r}} \in \mathbb{R}^{h\times w\times 3}$ per clip, since the intrinsics are constant within a clip. It is supervised by $\mathcal{L}_{\mathrm{ray}}$, the cosine distance to the calibrated pixel rays $\mathbf{r}$ defined in Appendix A.4, and its Fourier encoding serves as the ray positional embedding of Section 3.2.

The translation $\hat{\boldsymbol{\tau}}_t = (\hat{\tau}_{x,t}, \hat{\tau}_{y,t}, \hat{\tau}_{z,t})$ is recovered by the mixed PnP scheme of ACE-Ego-Hand (Liu et al., 2026). The depth head of the hand token directly predicts $\hat{\tau}_{z,t}$, thereby fixing the depth $z_{t,j}$ of each posed canonical joint $\mathbf{J}^{\mathrm{can}}_{t,j}$. The in-plane components are then obtained by the closed-form least-squares solution

$$
\begin{aligned}
(\hat{\tau}_{x,t}, \hat{\tau}_{y,t}) &= \arg\min_{\tau_x,\tau_y} \sum_j w_{t,j} \left\| \frac{\mathbf{J}^{\mathrm{can}}_{t,j,xy} + (\tau_x, \tau_y)}{z_{t,j}} - \mathbf{b}_{t,j} \right\|_2^2, \\
\mathbf{b}_{t,j} &= \Pi_{\mathbf{K}}^{-1}(\hat{\mathbf{a}}_{t,j}),
\end{aligned}
\tag{12}
$$

where $\mathbf{b}_{t,j}$ denotes the bearing, *i.e.*, the normalized image coordinate obtained by back-projecting the predicted 2D anchor $\hat{\mathbf{a}}_{t,j}$ of Appendix A.4 through the calibrated intrinsics $\mathbf{K}$. The weight is $w_{t,j} = z_{t,j}^{-2}$, and the solution uses only joints whose anchors lie inside the frame and whose depths are positive. When fewer than a minimum number of joints vote, or the refit residual exceeds a threshold, the solver falls back to placing the wrist on its own bearing at depth $\hat{\tau}_{z,t}$. The camera-frame joints are then $\widehat{\mathbf{J}}^{\mathrm{cam}}_t = \mathbf{J}^{\mathrm{can}}_t + \hat{\boldsymbol{\tau}}_t$, which are used in the 3D, 2D, and temporal losses.

All reported models use the calibrated $\mathbf{K}$ of each dataset; the ray head is used only for the positional embedding and auxiliary ray supervision.

A.4 Objective Function and Evaluation Metrics

This section gives the closed-form definitions of all training losses in Section 3.4 and evaluation metrics in Section 4. Throughout, $\langle\cdot\rangle$ denotes the mean over the valid entries of a mask. For training losses, $m_{t,s} \in \{0,1\}$ indicates the availability of the corresponding supervision type, *e.g.*, MANO parameters, 3D keypoints, or 2D keypoints, for frame t and hand s, allowing heterogeneous sources to be trained jointly in a single mixture. Here, t indexes frames, $s \in \{\mathrm{L},\mathrm{R}\}$ indexes hands, and j indexes the J=21 joints; hats denote predictions.

Rotations and shape. Let $\widehat{\mathrm{R}}_{t,j}$ be the assembled rotation stack of Section 3.2, *i.e.*, the global orientation and the 15 joint rotations, and $\mathrm{R}_{t,j}$ the ground truth. The rotation and shape losses are:

$$
\mathcal{L}_{\mathrm{rot}} = \left\langle \arccos \tfrac{\mathrm{tr}(\widehat{\mathrm{R}}^\top \mathrm{R})-1}{2} \right\rangle + \left\langle \|\widehat{\mathrm{R}} - \mathrm{R}\|_F^2 \right\rangle, \qquad \mathcal{L}_\beta = \left\langle \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_1 \right\rangle, \tag{13}
$$

where the shape loss is applied only to hands with ground-truth $\boldsymbol{\beta}$ that are not teacher-forced in that step, as described in Section 3.2, ensuring that the shape head is supervised under the same unconditioned setting used at test time.

3D joints. The wrist-relative term supervises articulation for both the MANO-posed joints and the joints directly regressed by the joint tokens, while the camera-frame and wrist terms supervise absolute placement through the recovered translation:

$$\mathcal{L}_{3D} = \lambda_{\text{rel}}\big\langle\|\widehat{\mathbf{J}}^{\text{rel}} - \mathbf{J}^{\text{rel}}\|_1\big\rangle + \lambda_{\text{cam}}\big\langle\|\widehat{\mathbf{J}}^{\text{cam}} - \mathbf{J}^{\text{cam}}\|_1\big\rangle + \lambda_{\text{w}}\big\langle\|\widehat{\mathbf{J}}^{\text{cam}}_{\text{wrist}} - \mathbf{J}^{\text{cam}}_{\text{wrist}}\|_1\big\rangle. \tag{14}$$

2D terms. The soft-argmax anchors and the reprojection of the posed 3D joints are supervised in normalized image coordinates:

$$\mathcal{L}_{2D} = \big\langle\|\hat{\mathbf{a}} - \mathbf{u}\|_1\big\rangle_{\text{in}} + \big\langle\|\Pi(\widehat{\mathbf{J}}^{\text{cam}}) - \mathbf{u}\|_1\big\rangle_{\text{in,front}} + \tfrac{1}{2}\big\langle\|\Pi(\widehat{\mathbf{J}}^{\text{cam}}_{\text{wrist}}) - \mathbf{u}_{\text{wrist}}\|_1\big\rangle_{\text{in,front}}, \tag{15}$$

where $\langle\cdot\rangle_{\text{in}}$ masks to confidence-weighted targets whose projections lie inside the frame, *i.e.*, the valid targets for the soft-argmax, and $\langle\cdot\rangle_{\text{front}}$ additionally requires the predicted joint to lie in front of the camera.

Translation, presence, smoothness, rays. The translation target is the offset between the ground-truth wrist and the wrist of the ground-truth-posed canonical MANO, $\boldsymbol{\tau}^{\text{gt}}_t = \mathbf{J}^{\text{cam}}_{\text{wrist},t} - \mathbf{J}^{\text{can}}_{\text{wrist},t}$, supervised with an $\mathcal{L}_1$ loss; Existence and visibility logits are supervised with binary cross-entropy on frames with known presence; $\mathcal{L}_{\text{tmp}}$ penalizes the second temporal difference, *i.e.*, the acceleration, of the camera-frame joints over valid frames; Finally, $\mathcal{L}_{\text{ray}} = \langle 1 - \hat{\mathbf{r}}^\top\mathbf{r}\rangle$ is the mean cosine distance between the predicted and calibrated pixel rays over the token grid.

Detection gate. All "-p" pose metrics follow the detection-gated protocol of ACE-Ego-Hand (Liu et al., 2026). Let $o_{t,s} \in \{0, 1\}$ denote whether the ground-truth hand is on screen, $\hat{p}_{t,s}$ the predicted presence probability, $\mathcal{B}(\cdot)$ the axis-aligned bounding box of a set of image points, and $\mathcal{B}_{1.1}(\mathbf{u}_{t,s})$ the ground-truth keypoint box dilated by 10%. A prediction counts as a detection when all three conditions below hold:

$$\begin{aligned} d_{t,s} = {} & \mathbb{1}\big[\hat{p}_{t,s} > 0.5\big]\, \mathbb{1}\big[\Pi(\widehat{\mathbf{J}}^{\text{cam}}_{t,s}) \text{ has a joint inside the frame with } \hat{z} > 0\big] \\ & \cdot\, \mathbb{1}\big[\mathcal{B}(\Pi(\widehat{\mathbf{J}}^{\text{cam}}_{t,s})) \cap \mathcal{B}_{1.1}(\mathbf{u}_{t,s}) \neq \emptyset\big]. \end{aligned} \tag{16}$$

Matched frames, *i.e.*, $d_{t,s} = 1$, contribute their actual errors, while a missed on-screen hand incurs a fixed *canonical-MANO penalty*: The error of the wrist-aligned rest pose with the mean shape for the 3D terms and the image diagonal for the 2D term. This prevents the metrics from being improved by dropping difficult frames.

F_{Acc}. Frame accuracy is the fraction of frames with neither a missed on-screen hand nor a spurious detection:

$$\mathrm{F}_{\text{Acc}} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\Big[\sum_s o_{t,s}\,(1 - d_{t,s}) + (1 - o_{t,s})\, f_{t,s} = 0\Big], \tag{17}$$

where $f_{t,s} = 1$ indicates a hand predicted to be present inside the frame despite a negative annotation. On bimanual datasets without explicit absence labels, only the first term is active.

MPJPE-p and PA-p. Let $\bar{\mathbf{J}}_{t,s} = \mathbf{J}_{t,s} - \mathbf{J}_{t,s,\text{wrist}}$ denote the wrist-aligned joints and $e^{\text{can}}_{t,s}$ the canonical-MANO penalty of a missed hand. MPJPE-p is the mean per-joint error in mm over the on-screen ground-truth hands, where a detected hand contributes its wrist-aligned error and a missed hand contributes the penalty:

$$\text{MPJPE-p} = \Big\langle d_{t,s}\, \tfrac{1}{J}\textstyle\sum_j \big\|\bar{\widehat{\mathbf{J}}}_{t,s,j} - \bar{\mathbf{J}}_{t,s,j}\big\|_2 + (1 - d_{t,s})\, e^{\text{can}}_{t,s}\Big\rangle_{o_{t,s}=1}. \tag{18}$$

PA-p is defined in the same way after aligning $\bar{\widehat{\mathbf{J}}}_{t,s}$ to $\bar{\mathbf{J}}_{t,s}$ with a Procrustes fit, *i.e.*, the optimal rotation, translation, and scale. The penalty of a missed hand is aligned in the same way.

EPE2D-p. Let $\mathcal{V}_{t,s}$ be the ground-truth joints inside the frame and (W, H) the image size. EPE2D-p is the mean 2D joint error in pixels of the evaluation resolution over the on-screen ground-truth hands, where a missed hand contributes the image diagonal:

$$\text{EPE2D-p} = \Big\langle d_{t,s}\, \tfrac{1}{|\mathcal{V}_{t,s}|}\textstyle\sum_{j\in\mathcal{V}_{t,s}} \big\|\Pi(\widehat{\mathbf{J}}^{\text{cam}}_{t,s,j}) - \mathbf{u}_{t,s,j}\big\|_2 + (1 - d_{t,s})\sqrt{W^2 + H^2}\Big\rangle_{o_{t,s}=1}. \tag{19}$$

MPJPE$^{+\mathrm{OOS}}$. This metric bypasses the detection gate and measures the wrist-aligned error over *all* frames with a ground-truth 3D hand $g_{t,s} = 1$, including out-of-sight ones:

$$\mathrm{MPJPE}^{+\mathrm{OOS}} = \left\langle \tfrac{1}{J} \textstyle\sum_j \left\| \hat{\bar{\mathbf{J}}}_{t,s,j} - \bar{\mathbf{J}}_{t,s,j} \right\|_2 \right\rangle_{g_{t,s}=1}. \tag{20}$$

This metric therefore evaluates temporal extrapolation across periods of missing observations.

Jitter. Jitter measures temporal consistency rather than accuracy. Let $\mathcal{R}$ be the set of maximal runs of at least three consecutive frames in which the hand is predicted present and is on screen in the ground truth. Jitter is the joint-averaged norm of the second temporal difference of the camera-frame joints in mm/frame^2, averaged first within each run and then across runs:

$$\mathrm{Jitter} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{1}{|r| - 2} \sum_{t \in r} \frac{1}{J} \sum_j \left\| \widehat{\mathbf{J}}^{\mathrm{cam}}_{t+1,j} - 2\widehat{\mathbf{J}}^{\mathrm{cam}}_{t,j} + \widehat{\mathbf{J}}^{\mathrm{cam}}_{t-1,j} \right\|_2. \tag{21}$$

The 2D detection gate used for the accuracy metrics is not applied, so that each tracker is evaluated on all of its predictions rather than only on well-localized ones. Since the evaluated frame sets differ across methods, we additionally verified that restricting all methods to the frames detected by every method leaves the ranking unchanged on all three test sets.

Shape jitter. σ_{shape} quantifies the within-recording drift in predicted hand *size*, despite the subject's hand remaining consistent. Let $\ell(\boldsymbol{\beta})$ be the length of the middle-finger bone chain, *i.e.*, wrist-MCP-PIP-DIP-tip, of the MANO template posed with shape $\boldsymbol{\beta}$ at the identity pose $\boldsymbol{\theta}_0$. The bone length and the resulting shape jitter are:

$$\ell(\boldsymbol{\beta}) = \sum_{k=1}^{4} \left\| \mathbf{J}_k(\boldsymbol{\beta}, \boldsymbol{\theta}_0) - \mathbf{J}_{k-1}(\boldsymbol{\beta}, \boldsymbol{\theta}_0) \right\|_2, \qquad \sigma_{\mathrm{shape}} = \left\langle \mathrm{std}_{t \in \mathcal{T}_{c,s}} \ \ell(\hat{\boldsymbol{\beta}}_{t,s}) \right\rangle_{(c,s)}, \tag{22}$$

where $\mathcal{T}_{c,s}$ is the set of frames of recording c and hand s for which the method predicts the hand as present while the ground-truth hand is on screen, and the outer mean, in mm, is taken over all recordings and hands with $|\mathcal{T}_{c,s}| \geq 30$. Because ℓ is evaluated at a fixed pose, it depends only on $\hat{\boldsymbol{\beta}}$, so a method that uses a single shape per recording has $\sigma_{\mathrm{shape}} = 0$. Per-frame trackers re-estimate $\hat{\boldsymbol{\beta}}$ for every crop, causing hand size to fluctuate by several millimeters within a recording, while clip-level models re-estimate it once per window. The $\boldsymbol{\beta}$-cache protocol applies the shape cached from the first well-visible window to every window in the recording, making $\sigma_{\mathrm{shape}} = 0$ by construction.

Contact F1 and AUROC. Let $c_{t,v} \in \{0, 1\}$ be the ground-truth contact of MANO vertex v in frame t and $\hat{c}_{t,v} \in [0, 1]$ the predicted probability. Both scores are micro-averaged over all $|V|$ vertices of every evaluated frame:

$$\begin{aligned} \mathrm{F1} &= \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \quad \text{with} \quad \hat{c}_{t,v} > \tau_c, \\ \mathrm{AUROC} &= \Pr\left[\hat{c}_{t,v} > \hat{c}_{t',v'} \;\middle|\; c_{t,v} = 1,\ c_{t',v'} = 0 \right], \end{aligned} \tag{23}$$

where F1 is reported at the fixed operating point $\tau_c = 0.8$, selected on the validation splits, whereas AUROC evaluates the predicted ranking independently of this threshold. AUROC is computed from per-bin score histograms with 20k bins, agreeing with the exact rank-based statistic to within 10^{-6} without materializing the full score vector.

Force MAE and RMSE. Let $f_{t,v}$ be the measured and $\hat{f}_{t,v}$ the predicted per-vertex pressure in kPa. Over all vertices of frames with force ground truth, the two force metrics are:

$$\mathrm{MAE} = \langle |\hat{f}_{t,v} - f_{t,v}| \rangle, \qquad \mathrm{RMSE} = \sqrt{\langle (\hat{f}_{t,v} - f_{t,v})^2 \rangle}. \tag{24}$$

Tactile ground truth and evaluation protocol. Contact ground truth differs by source, which matters when comparing the columns of Table 2. OpenTouch (Song et al., 2025) labels a vertex from measured glove pressure above 1 kPa, whereas DexYCB (Chao et al., 2021), HOT3D (Banerjee et al., 2024), and ARCTIC (Fan et al., 2023) derive contact geometrically as a MANO vertex lying

Term of Eq. (7)	Component	Weight	Value
$\mathcal{L}_{\text{rot}}$	geodesic rotation error	λ_{geo}	1
	Frobenius rotation error	λ_{mse}	1
$\mathcal{L}_{\beta}$	shape	λ_{β}	0.1
$\mathcal{L}_{\text{3D}}$	wrist-relative joints	λ_{rel}	10
	camera-frame joints	λ_{cam}	5
	camera-frame wrist	λ_{w}	2
$\mathcal{L}_{\text{2D}}$	soft-argmax anchors	λ_{anch}	1
	joint reprojection	λ_{reproj}	1
	wrist reprojection	$\lambda_{\text{reproj,w}}$	0.5
$\mathcal{L}_{\tau}$	translation	λ_{τ}	1
$\mathcal{L}_{\text{pres}}$	existence	λ_{exist}	0.5
	visibility	λ_{vis}	0.25
$\mathcal{L}_{\text{tmp}}$	temporal smoothness	λ_{tmp}	0.5
$\mathcal{L}_{\text{ray}}$	pixel rays	λ_{ray}	1

Table 7: Stage-1 loss weights. Each row is one component of the corresponding term of Eq. (7).

within 5 mm of the object mesh. The latter definition is also used to train HACO (Jung & Lee, 2025), making these three splits directly aligned with its training labels. For HOT3D, the Aria and Quest recordings are evaluated jointly by summing confusion counts for F1 and score histograms for AUROC, rather than averaging the per-rig metrics. ARCTIC provides no test-split ground truth, so we report its validation split. We re-run HOPE (Jeon et al., 2026) and HACO from their official code and checkpoints as described in Appendix A.5, using exactly the frames evaluated in our results, rather than quoting published numbers.

Q-err. Let $\mathbf{q}_t \in \mathbb{R}^D$ be the joint command produced by a retargeting solver at frame t from the predicted hand, and let $\mathbf{q}_t^{\text{oracle}}$ denote the command produced by the *same* solver at the same frame using the *ground-truth human joints*. All retargeting metrics are computed over contiguous runs of at least three ground-truth-labeled frames, ensuring that temporal differences do not span label gaps. Q-err is the mean absolute joint-angle deviation in degrees between the two commands:

$$\text{Q-err} = \left\langle \tfrac{1}{D} \left\| \mathbf{q}_t - \mathbf{q}_t^{\text{oracle}} \right\|_1 \right\rangle_t . \tag{25}$$

Jerk. Jerk is the joint-averaged absolute second temporal difference of the command, reported in $^\circ$/frame^2 as its *median* over frames:

$$\text{Jerk} = \text{median}_t \ \tfrac{1}{D} \left\| \mathbf{q}_{t+1} - 2\mathbf{q}_t + \mathbf{q}_{t-1} \right\|_1 . \tag{26}$$

Since the ground-truth annotations contain sparse frame-to-frame steps, we use the median instead of the mean. The mean is dominated by these steps in the Oracle row, whereas the median better reflects the frame-to-frame noise introduced by the tracker. We disable low-pass filter of DexPilot, so Jerk reflects the tracker rather than the filter.

A.5 Training and Evaluation

Training. Training videos are sampled as clips of T=81 frames at an input height of 480. The RLHND decoder uses width D=384, 4 layers, 8 heads, and 4 register tokens, with LoRA at rank 64. Both stages use AdamW with β_1=0.9, β_2=0.95, weight decay 0.01, 200 warmup steps followed by cosine decay, gradient clipping at 1.0, and bf16 mixed precision. Learning rates are 2×10^{-4} for the decoder, ray head, and tactile expert, 1×10^{-4} for LoRA, and 2×10^{-5} for the patch embedding. Stage-1 trains for 20k steps on 8 H100 GPUs with 2 clips per GPU and 2-step gradient accumulation, giving an effective batch size of 32. Stage-2 trains the tactile expert for 20k steps on a single A100 with a batch size of 2, while freezing the entire stage-1 model so that only the expert and vertex readout receive gradients. The stage-1 loss weights in Eq. (7) are listed in Table 7. Stage-2 uses $\lambda_c = \lambda_F = \lambda_\pi = 1$ with a positive-class weight of 10 in BCE_w.

Evaluation. We run every baseline with its official code and released weights on exactly the frames we score, rather than quoting published numbers. This ensures that all methods are evaluated on the same frames. The crop-based per-frame trackers HaMeR (Pavlakos et al., 2024), WiLoR (Potamias et al., 2024), HaWoR (Zhang et al., 2025), and HaPTIC (Ye et al., 2025b) share the hand detector released with WiLoR and the detection logic introduced in HaWoR. The detector selects the highest-confidence box per side and mirrors left hands to the right-hand model. Their F_{Acc} therefore coincides, and their "-p" metrics use the same detection front-end. HandFlow (Xu et al., 2026) is run through its released pipeline. Since it has no 2D head, EPE2D-p uses its reprojected joints, and camera translation is obtained from its crop-based weak-perspective lift. ACE-Ego-Hand (Liu et al., 2026) is run with its official inference code and checkpoint. We use one forward pass per test window at the protocol resolution with the released MANO decode and camera solver.

For tactile estimation, HACO (Jung & Lee, 2025) is run from its public release. HOPE (Jeon et al., 2026) uses the official code and checkpoint shared by its authors on request. Retargeting uses the same DexPilot (Handa et al., 2020) solver for every method, as in Section 4.3.

The two RLHND rows in Table 1 share one checkpoint and differ only in the shape protocol. One uses per-window prediction, while the other uses the β-cache of Section 3.2, which applies the shape cached from the first well-visible window to every window of the recording.

Training DPP. We explain the detailed settings for training DPP as following:

Data. The 200 human demonstrations comprise 25 demonstrations for each pick-and-place object, *e.g.*, ball, bottle, box, and bird, together with 50 plastic-bag flips and 50 tissue assemblies. They are recorded at 1280×720 and 30 fps using the ego rig. The 200 teleoperated robot episodes follow the same task split. The $100:100$ mixture in Table 4 uses all 200 human and all 200 robot episodes. Without rollout, we additionally train $100:10$ and $100:50$ mixtures, which keep all human episodes and sample 20 or 100 robot episodes using a task-balanced even stride. Figure 7 shows example frames from both data sources.

Human hand labels. RLHND is run on the right ego view at the native resolution using the rectified ZED Mini intrinsics, closely following the original DPP setting for a fair comparison. It returns the 21 keypoints of both hands in the camera frame and in the MANO order expected by DPP, together with a per-frame validity flag from its presence head. The original DPP labels are based on HaWoR keypoints, rescaled per frame so that the wrist aligns with the stereo depth using a $+2$ cm offset and a temporal median over 5 frames, with the scale clipped to $[0.5, 2]$. The resulting keypoints are cleaned to remove duplicate slots, spikes, and track discontinuities. The RLHND keypoints undergo the same alignment and cleaning before packing, ensuring that the comparison focuses on the tracker rather than the depth scale.

Contact labels. Since HaWoR has no contact head, both rows of Table 4 use the contact labels from contact head of RLHND. Specifically, its per-vertex contact probabilities on the $|V|$ MANO vertices are converted by the DPP packer into 21 per-keypoint labels using a fixed vertex-to-keypoint weight map. Figure 8 shows these automatic labels on example demonstrations. In short, The two rows differ only in the hand keypoints.

Packing and objective. Episodes are resampled to 20 fps, and each observation is represented in the camera frame of its anchor step relative to a per-episode origin. The observation consists of 18 tokens of width 512, comprising up to four objects represented by 512 stereo points each, together with MiniLM embeddings of the instruction and object names, the 42 hand keypoints, and the 9-D camera pose. Object points are obtained from Fast-FoundationStereo (Wen et al., 2025a) depth and SAM 3 (Carion et al., 2025) masks. The action is a 16-step chunk, *i.e.*, 0.8s, of keypoint displacements, normalized per step and channel using the statistics of the DPP pre-training corpus and clipped to ± 5. The objective is rectified flow, implemented as an MSE on the velocity between Gaussian noise and the target, masked when a hand is not tracked, together with a BCE loss on the 42 contact logits computed from detached features so that the contact loss does not backpropagate into the trunk.

Optimization. Every policy starts from the same DiT-B checkpoint pretrained for 100k steps and is fine-tuned for 60k steps with AdamW using a learning rate of 10^{-4}, weight decay of 10^{-4}, 1k warm-up steps followed by a constant learning rate, gradient clipping at 1.0, batch size 128, dropout 0.1, and bf16 autocast on a single A100, which takes 60-90 minutes per training each policy. Snapshots

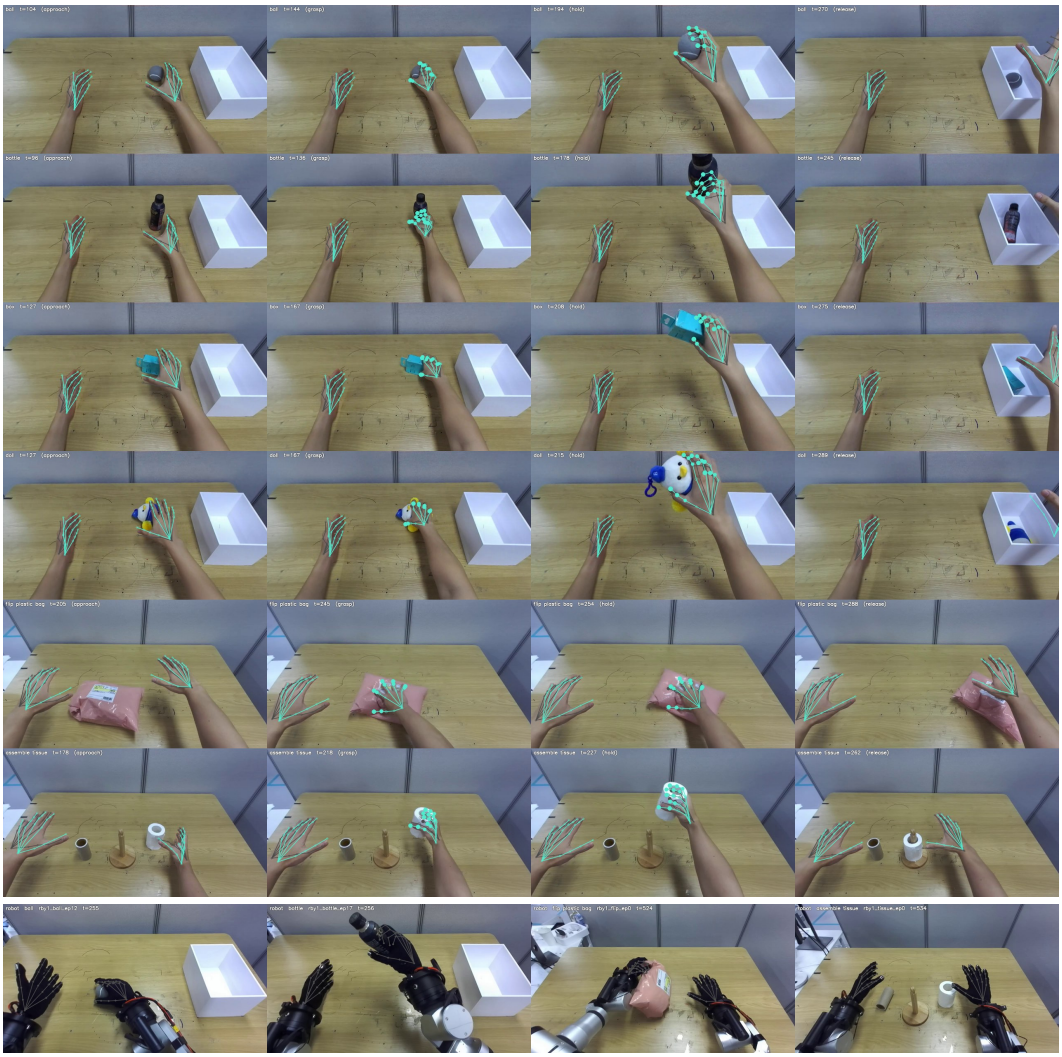

Figure 7: **Examples of the DPP training data.** *Top*: human demonstrations for the six tasks, *e.g.*, ball, bottle, box, bird, plastic bag, and tissue, across approach, grasp, hold, and release phases, viewed from the shared ego camera. Gray and teal denote the original DPP, *i.e.*, HaWoR, and RLHND hand keypoints, respectively. Marker size and fill indicate the corresponding contact labels. *Bottom*: teleoperated robot episodes with forward-kinematics keypoints in gray.

are saved every 10k steps, and the 60k snapshot is used for deployment with 6 Euler steps from Gaussian noise.

Evaluation on real-robot. Figure 9 shows the hardware platform. We use an RB-Y1 bimanual robot with a WUJI Hand 2 on each arm, *i.e.*, a 7-DoF arm and a 20-DoF hand per side, and the policy observes the scene through the head-mounted ZED Mini stereo camera at 1280×720 and 30 fps. Human demonstrations are recorded with an identical camera rig aimed to reproduce the ego view of the robot, therefore human and robot episodes share the same image format, intrinsics, and viewpoint. We collect 200 human demonstrations and 200 teleoperated robot episodes across six tasks, four single-handed pick-and-place tasks and two bimanual tasks:

- **Pick and place (bottle, box, ball, bird).** The right hand picks up the object from the table and places it in the white container. The four tasks share the same procedure and differ only in the object, including a rigid bottle and box, a small ball, and a soft plush bird.

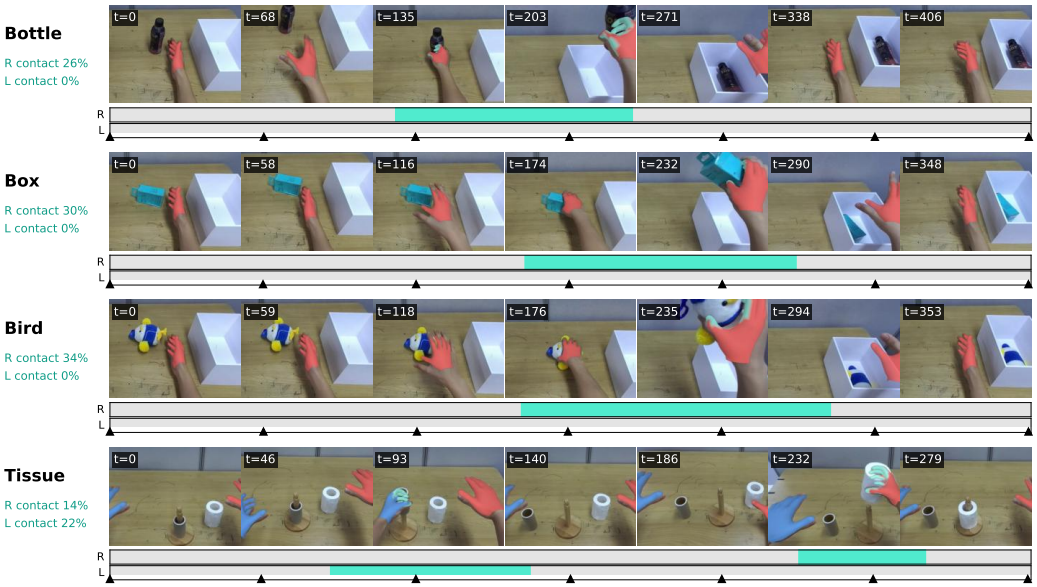

Figure 8: **Automatic contact labels from RLHND.** We denote vertices whose predicted contact probability exceeds 0.5 as teal. The strips below each row show the frame-level contact labels for the right (R) and left (L) hands, obtained by thresholding the mean contact probability over the fingertip vertices at 0.5. Triangles mark the displayed frames and percentages indicate the fraction of frames labeled as contact.

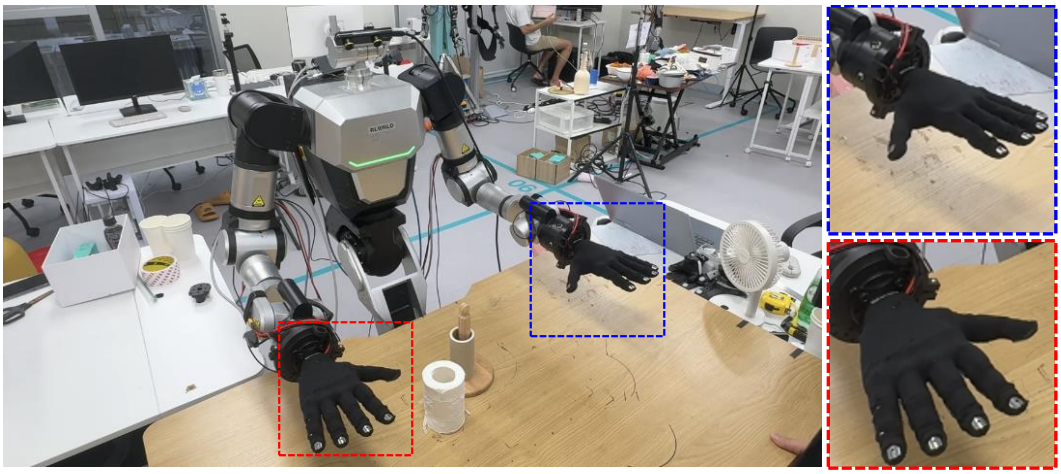

Figure 9: **Real-robot platform.** An RB-Y1 bimanual robot with a WUJI dexterous hand on each arm in the tabletop workspace used for the real-robot experiments. The insets on the right show close-ups of the two hands, with the robot's left and right hands outlined in blue and red, respectively.

- **Plastic bag.** The left hand picks up the plastic bag, flips it so that the barcode faces up, and hands it to the right hand, which then passes it to the other rail. This task models a conveyor sorting scenario in a logistics line, where packages must be oriented for scanning and routed to the appropriate lane.
- **Tissue assembly.** The left hand removes the used tissue core from the holder, while the right hand picks up a new tissue roll and places it on the holder.

Both rows of the table use the same policy and training recipe, *i.e.*, a Dexterous Point Policy with DiT structure with 12 layers of width 768, a 16-step action horizon, and up to four objects of 512

Figure 10: **Real-robot rollouts of DPP + RLHND.** Five snapshots of one rollout per task of the DPP + RLHND policy on the RB-Y1 with WUJI hands, from a third-person view. The four pick-and-place tasks (bottle, box, ball, bird) have the right hand grasp the object and drop it into the white container, and the two bimanual tasks are the ones where the left hand picks up and flips the plastic bag before the right hand passes it on, and the left hand removes the used tissue core before the right hand places the new roll on the holder.

points each, initialized from a checkpoint pre-trained for 100k steps on the DPP human-video corpus and fine-tuned for 60k steps with batch size 128 on the mixed set. The only difference between the rows is the *hand keypoint labels* of the 200 human episodes. The original tracker, HaWoR, provides the keypoints for DPP, while RLHND provides them for DPP + RLHND; both rows use the contact labels of RLHND, since HaWoR does not predict contact. The robot episodes, objects, instructions, and hyperparameters are identical.

At the inference, object points are extracted online from the stereo pair using Fast-FoundationStereo (Wen et al., 2025a;b) depth and SAM 3 (Carion et al., 2025) masks at 5 Hz, and the policy runs with 6 flow steps to predict a 16-step chunk of keypoint displacements and per-keypoint contact probabilities for both hands. The predicted keypoints are retargeted to the robot by inverse kinematics over the arm and hand joints with the finger abduction joints locked. Keypoints whose contact probability exceeds 0.05 are pushed 15 mm along the outward normal of their link to inject contact force. Each task is evaluated over 16 trials per condition with randomized object placement.

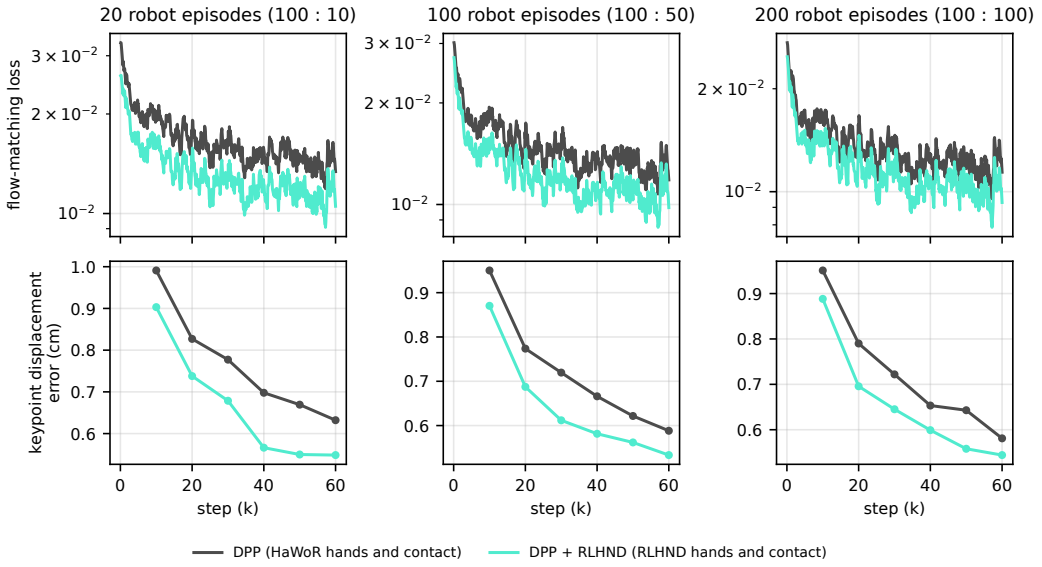

Figure 11: **Fine-tuning of DPP and DPP + RLHND.** *Top*: action flow-matching loss over 60k fine-tuning steps. *Bottom*: keypoint displacement error of sampled actions at every 10k-step snapshot, measured as the mean L2 distance in cm between predicted and recorded 16-step displacements of the 21 keypoints. Results are shown for three human : robot ratios; only the 100 : 100 policies are deployed in Table 4. Gray denotes DPP with its original human labels and teal denotes DPP + RLHND.

B Additional Experimental Results

B.1 Real-Robot Experiments with DPP

Rollouts. Figure 10 shows one rollout of the DPP + RLHND policy per task of Table 4. The pick-and-place rollouts share the same approach, grasp, lift, carry and release structure and differ only in the object; in the plastic-bag task the left hand picks up and flips the bag before handing it to the right hand, and in the tissue task the left hand removes the used core before the right hand places the new roll.

Depth of the human hand labels. Figure 12 compares the wrist depth of the two trackers with the stereo depth that the DPP pipeline aligns them to. Even before the alignment, RLHND already follows the stereo trajectory closely. Over the 200 demonstrations, its right-wrist depth deviates from the stereo target by a median of 2.3 cm, which is *17.8%* lower than HaWoR's. In both cases the deviation is dominated by a constant offset, with a median bias of $+2.2$ cm for RLHND and $+2.6$ cm for HaWoR, whereas the frame-to-frame errors are visibly smaller for RLHND during fast reaching motions. Since the alignment only rescales each frame and leaves the temporal noise of the tracker in place, the residual error is also smaller for RLHND after the per-frame alignment, *e.g.*, at 0.1 cm against 0.2 cm. The remaining offset is largely a scale-depth ambiguity of the monocular hand, so it can be reduced further by measuring the actor's hand once and running RLHND with a pre-defined β_{cache}, which fixes the hand size and therefore the depth. We nevertheless apply the same alignment to both sources, so that Table 4 compares the trackers rather than their depth scale.

Fine-tuning loss. Figure 11 compares DPP fine-tuned with its original human labels and with RLHND labels, using both the action flow-matching loss and the keypoint displacement error of sampled actions on training windows. Only the 100 : 100 policies of Table 4 were deployed on the robot, but we fine-tuned both label sources for three human : robot ratios (100 : 10, 100 : 50, and 100 : 100, *i.e.*, 20, 100, and 200 robot episodes). DPP + RLHND consistently achieves a lower flow-matching loss across all ratios, and its sampled keypoint displacements remain closer to the recorded trajectories throughout training, reaching 0.55, 0.53, and 0.54 cm versus 0.63, 0.59, and

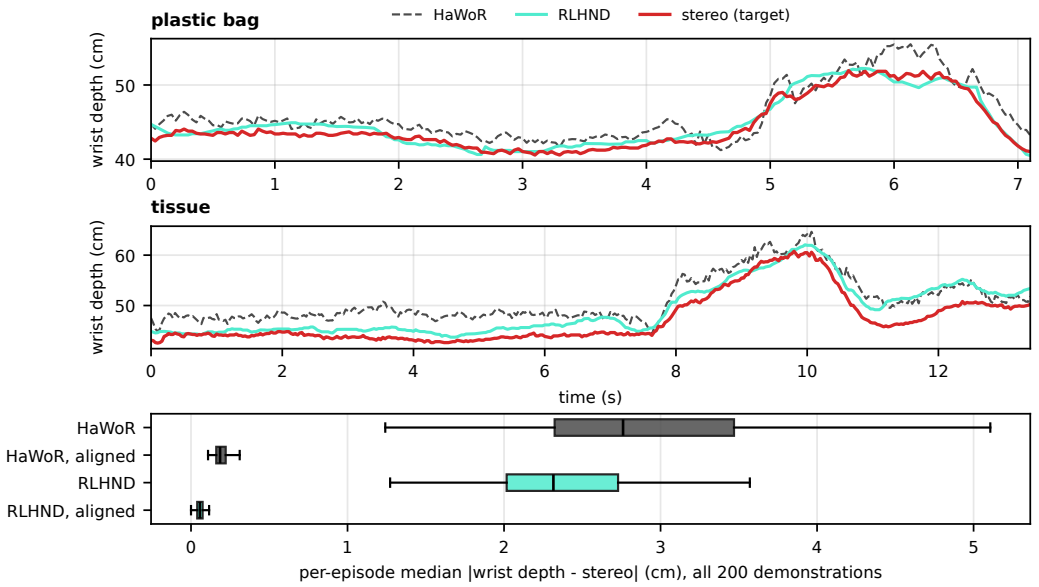

Figure 12: **Wrist depth of the human hand labels against the stereo depth.** The top two rows show the right-wrist depth over one demonstration each of the plastic-bag and tissue tasks, for the stereo target, raw HaWoR, and raw RLHND, before the per-frame alignment. The bottom row shows the per-episode median absolute deviation from the stereo target over all 200 demonstrations, for both trackers before and after the alignment.

	HOT3D			ARCTIC ego			EgoDex		
Method	Jitter	Jitter$_z$	Jitter$_{xy}$	Jitter	Jitter$_z$	Jitter$_{xy}$	Jitter	Jitter$_z$	Jitter$_{xy}$
HaMeR	25.40	15.84	17.51	21.73	18.21	9.56	15.48	11.88	8.53
WiLoR	22.17	13.45	15.50	25.92	22.84	9.53	13.05	9.71	7.43
HaWoR	26.13	16.96	17.42	29.34	26.12	10.56	16.94	13.83	8.35
HaPTIC	38.27	27.02	23.97	51.09	47.28	16.40	25.81	21.88	12.06
HandFlow	16.99	8.41	13.37	13.47	7.88	9.49	10.89	5.69	8.32
RLHND (Ours)	**4.38**	**1.55**	**3.78**	4.19	**1.48**	3.55	**2.93**	**1.58**	**2.20**
RLHND (Ours) (β-cache)	4.39	**1.55**	3.79	**4.18**	**1.48**	**3.53**	2.94	**1.58**	**2.20**

Table 8: **Jitter decomposition.** Jitter (mm/frame^2) is split into its depth component Jitter$_z$ and image-plane component Jitter$_{xy}$ under the protocol of Table 1.

0.58 cm for DPP at 100 : 10, 100 : 50, and 100 : 100, respectively. The deployed DPP + RLHND policy also achieves a higher rollout success rate in Table 4.

B.2 Effect of the β-Cache on Jitter

Table 1 shows that the β-cache removes within-recording shape drift, *e.g.*, σ_{shape} drops to zero, but leaves joint jitter essentially unchanged. Table 8 explains why by decomposing jitter into its depth component Jitter$_z$, the second temporal difference of the camera-frame z coordinate, and its image-plane component Jitter$_{xy}$.

Three observations follow. First, the residual jitter of clip-level trackers is dominated by the image plane. For RLHND, Jitter$_{xy}$ accounts for at least three quarters of the total on every test set and is almost entirely a wrist-level effect. The wrist alone has 3.3 / 3.2 / 2.5 mm/frame^2 on HOT3D / ARCTIC / EgoDex, indicating that the 2D anchors driving mixed-PnP placement move slightly from frame to frame while the articulation remains stable.

Method	Params (M)	Detector (ms)	Model (ms)	Total (ms/frame)	FPS	MPJPE-p (HOT3D)
HaMeR	699	2.7	11.7	14.4	69.4	65.53
WiLoR	668	2.7	11.9	14.6	68.4	44.89
HaWoR	720	2.7	12.1	14.8	67.7	41.53
HaPTIC	1,357	2.7	23.8	26.5	37.7	62.42
HandFlow	866	–	17.4	17.4	57.4	39.90
ACE-Ego-Hand	3,044	–	1.8	1.8	568.6	24.41
RLHND (Ours)	7,849	–	6.9	6.9	144.5	13.01

Table 9: **Inference cost.** Inference time measured on one NVIDIA H100 (80 GB) over the same HOT3D Aria test windows, using GPU-synchronized model-forward time per video frame with both hands. Video decoding is excluded and the first forward pass is discarded as warm-up. For crop-based trackers, the shared hand-detector cost is reported separately and included in the total. MPJPE-p is from Table 1.

Second, the $\boldsymbol{\beta}$-cache cannot affect either component by construction. Depth is predicted directly by the per-frame depth head of the translation solver in Section 3.2 and does not depend on $\boldsymbol{\beta}$. The cached shape only rescales the canonical joints, shifting the solved translation by a constant, sub-millimeter amount. On HOT3D, the two RLHND rows differ by a median wrist displacement of 0.2 mm in z and 0.4 mm in xy. Because a constant shift within a recording has zero second temporal difference, Jitter$_z$ and Jitter$_{xy}$ coincide between the two rows to two decimals. The benefit of the cache is therefore confined to size consistency measured by σ_{shape}, while temporal stability comes from the clip-level decoding shared by both rows.

Third, the decomposition sharpens the comparison with crop-based trackers. Their jitter is dominated by depth, reaching 18-47 mm/frame^2 on ARCTIC versus 1.5 for RLHND, a 12-32$\times$ gap, whereas their image-plane jitter is only 2-5$\times$ higher. Per-frame crops therefore stabilize apparent hand size but re-estimate metric distance independently at every frame, which directly affects the robot action space.

B.3 Inference Cost

Table 9 and Figure 13 report the inference cost of every tracker in Table 1 under a common protocol: the same H100, the same HOT3D Aria test windows, GPU-synchronized model-forward time per video frame with both hands, and, for crop-based trackers, the shared hand detector included in the measurement. RLHND runs its 7.8B-parameter Cosmos 3 Nano encoder once per 81-frame clip, so despite being the largest model, it has a per-frame cost comparable to a single ViT-H hand crop and lower cost than the crop-based trackers once the second hand and detector are included. ACE-Ego-Hand, whose Wan2.2 encoder is truncated at a shallower depth, is the cheapest clip-level tracker at $4\times$ our throughput, but its HOT3D error is nearly twice ours.

B.4 Per-Dataset Retargeting Results

Table 10 breaks down the retargeting comparison of Table 3 by test set, *i.e.*, HOT3D, ARCTIC ego, and EgoDex, using the same DexPilot solver, wrist-relative targets, and metrics as in the main text. On HOT3D, RLHND has the lowest or second-lowest Q-err on every embodiment; on ARCTIC ego it has the lowest Q-err on Sharpa, Shadow, and ALLEX, while HaPTIC and WiLoR are closer on WUJI and Inspire. On both sets RLHND, together with ACE-Ego-Hand, produces the smoothest commands. On EgoDex, Q-err differences are smaller, with all trackers within 3° of each other on WUJI and Sharpa, likely due to noise inherent in the ARKit ground-truth joints themselves. The Oracle row has jerk of 0.13-0.27 deg/frame^2 on HOT3D and 0.17-0.43 deg/frame^2 on EgoDex, showing that the reference commands used for Q-err already contain frame-to-frame noise.

RLHND remains within 2° of the best method on every embodiment on EgoDex and achieves the lowest Q-err on Shadow. It also ties ACE-Ego-Hand for the lowest jerk, with each method achieving the lowest jerk on some of the five hands. Across all three test sets, the crop-based per-frame trackers exhibit substantially higher jerk than the clip-level trackers HandFlow, ACE-Ego-Hand, and RLHND, with differences of the same order of magnitude.

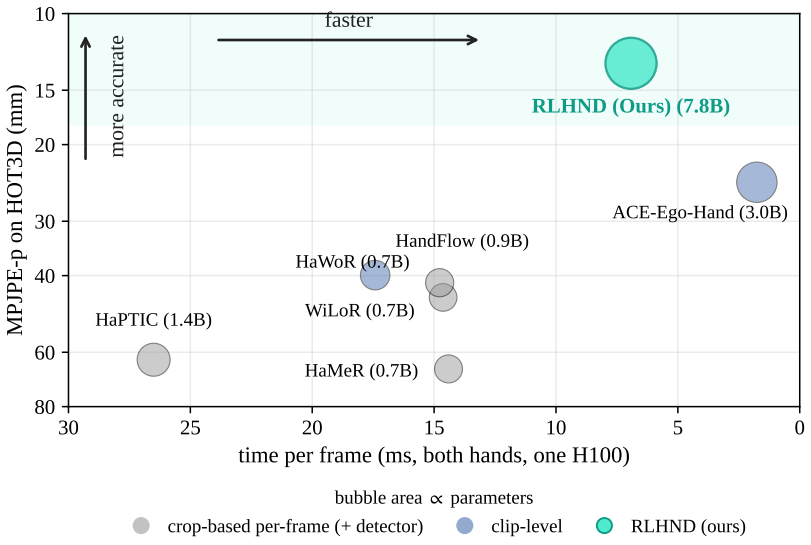

Figure 13: **Throughput against accuracy.** Time per frame with both hands on one H100 (ms, linear, axis reversed so that faster is to the right) against MPJPE-p on HOT3D (log scale, lower is better, so better trackers lie further from the origin); bubble area is proportional to the number of parameters used at inference (labeled in billions). Crop-based per-frame trackers include the shared hand detector. Numbers in Table 9.

B.5 Additional Qualitative Results

Figure 14 extends Figure 3 with additional EgoDex windows from a test set unseen during training. The mesh overlays are similar across methods, as crop-based trackers are optimized primarily for per-frame reprojection. Their trajectories, however, show clearer differences. The crop-based trackers, *i.e.*, HaMeR, WiLoR, HaWoR, and HaPTIC, estimate each frame independently and exhibit jagged wrist trajectories, consistent with the depth instability quantified in Figure 1 and reflected in the jerk metric of Table 10. ACE-Ego-Hand produces smoother trajectories but misses a hand in several windows, resulting in missing or displaced meshes, whereas RLHND tracks both hands more consistently and more closely follows the ground-truth trajectories.

Figure 15 extends Figure 4 with additional OpenTouch and PVDB test frames. The same qualitative differences appear across scenes: HACO predicts contact over broader regions of the palm and fingers, HOPE produces fragmented contact regions with lower pressure estimates, and PressureVision++ produces little output when the hand is outside its planar-sensor representation. In contrast, RLHND places contact and pressure in regions corresponding to the glove or pressure pad measurements, including the single-fingertip presses in the PVDB examples.

B.6 Additional Ablation Results

Per-vertex image sampling. We additionally evaluate per-vertex image features to test whether local visual evidence improves tactile estimation. We augment Eq. (6) with a zero-initialized MLP applied to decoder features bilinearly sampled at each vertex's projected image location, together with a visibility cue given by the cosine between the vertex normal and the viewing ray. This variant achieves vertex-level contact F1 of 0.709 / 0.543 / 0.570 on OpenTouch / DexYCB / HOT3D and an OpenTouch force MAE of 1.256 kPa, compared with 0.696 / 0.572 / 0.589 and 0.489 kPa for the default readout in Table 5. Replacing the nearest-latent lookup with temporal interpolation or removing the visibility cue changes every metric by less than 0.01. These results suggest that per-vertex image evidence provides little additional information beyond the pose-conditioned features and label supervision, so we omit the image-sampling branch from the final design.

Figure 14: **Extensive qualitative comparison of 2D and 3D hand motion reconstruction.**

Figure 15: **Extensive qualitative comparison of per-vertex contact and force estimation.**

Test set	Method	Sharpa Wave (22 DoF)		WUJI v2 (20 DoF)		Shadow (24 DoF)		Inspire RH56 (12 DoF)		ALLEX (15 DoF)	
		Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓	Q-err ↓	Jerk ↓
HOT3D	Oracle (GT joints)	0.0	0.27	0.0	0.22	0.0	0.27	0.0	0.13	0.0	0.25
	HaMeR (Pavlakos et al., 2024)	15.0	1.50	17.8	1.46	13.1	1.30	7.1	0.99	13.3	1.36
	WiLoR (Potamias et al., 2024)	11.8	1.25	14.6	1.24	9.4	1.03	4.9	1.00	9.6	1.11
	HaWoR (Zhang et al., 2025)	13.6	0.37	15.8	0.32	10.7	0.36	7.2	0.25	11.2	0.32
	HaPTIC (Ye et al., 2025b)	15.3	1.60	18.2	1.52	12.7	1.09	6.8	1.10	13.7	1.66
	HandFlow (Xu et al., 2026)	14.8	0.56	16.5	0.54	14.5	0.46	10.4	0.38	15.5	0.59
	ACE-Ego-Hand (Liu et al., 2026)	10.0	0.24	12.4	0.25	8.5	0.26	4.9	0.12	8.8	0.23
	RLHND (Ours)	**8.9**	**0.22**	**10.2**	**0.17**	**7.4**	**0.17**	4.0	0.08	**6.9**	0.19
	RLHND (Ours) (β-cache)	9.0	**0.22**	11.2	0.18	**7.4**	0.18	**3.9**	**0.07**	7.1	**0.17**
ARCTIC ego	Oracle (GT joints)	0.0	0.34	0.0	0.38	0.0	0.36	0.0	0.23	0.0	0.44
	HaMeR (Pavlakos et al., 2024)	14.0	1.59	17.0	1.43	11.6	1.50	8.6	1.25	14.1	1.56
	WiLoR (Potamias et al., 2024)	10.7	1.57	14.1	1.40	9.7	1.12	6.4	1.00	10.3	1.49
	HaWoR (Zhang et al., 2025)	13.2	0.95	16.8	0.74	11.8	0.72	11.0	0.71	14.3	0.77
	HaPTIC (Ye et al., 2025b)	10.2	1.47	**13.2**	1.37	10.3	1.36	**6.2**	1.20	10.1	1.32
	HandFlow (Xu et al., 2026)	23.1	0.57	26.9	0.63	16.3	0.59	12.6	0.81	22.1	0.63
	ACE-Ego-Hand (Liu et al., 2026)	12.0	0.21	17.2	0.24	12.1	**0.21**	9.1	0.16	12.4	0.25
	RLHND (Ours)	**9.3**	0.18	15.4	**0.18**	9.7	0.23	6.8	**0.08**	**9.4**	**0.20**
	RLHND (Ours) (β-cache)	**9.3**	**0.15**	15.2	0.20	**9.6**	0.23	6.8	0.10	**9.4**	**0.20**
EgoDex	Oracle (GT joints)	0.0	0.42	0.0	0.40	0.0	0.43	0.0	0.17	0.0	0.39
	HaMeR (Pavlakos et al., 2024)	21.0	1.27	23.5	1.05	13.2	1.14	8.5	0.47	15.0	0.66
	WiLoR (Potamias et al., 2024)	20.0	1.62	22.2	1.37	**11.6**	1.46	7.7	0.71	**13.8**	0.95
	HaWoR (Zhang et al., 2025)	**18.3**	0.52	21.6	0.35	14.2	0.42	8.6	0.22	17.5	0.40
	HaPTIC (Ye et al., 2025b)	19.2	1.10	**21.3**	0.91	12.5	0.85	8.1	0.31	14.3	0.66
	HandFlow (Xu et al., 2026)	**18.3**	0.62	22.0	0.48	11.8	0.49	**6.8**	0.29	15.8	0.38
	ACE-Ego-Hand (Liu et al., 2026)	20.1	0.15	22.4	**0.14**	11.7	**0.16**	8.4	0.05	14.8	**0.12**
	RLHND (Ours)	20.0	**0.14**	22.8	**0.14**	**11.6**	0.18	7.4	**0.03**	14.4	0.13
	RLHND (Ours) (β-cache)	20.0	0.15	22.6	**0.14**	11.7	0.17	7.4	0.04	14.5	0.14

Table 10: Per-dataset retargeting quality on five dexterous right hands (DexPilot, wrist-relative targets); Table 3 in the main text reports the mean of the three blocks. Q-err is the mean absolute deviation (°) of the joint command from the one the same solver produces on the ground-truth joints; Jerk is the median over frames of the joint-averaged second temporal difference of the command (°/frame^2), i.e. the frame-to-frame noise on typical frames (Appendix A.4). The grey Oracle row retargets the ground-truth joints (Q-err $= 0$ by construction). We denote **best** and second best values with shade and bold.

Conditioning the expert on the pose stream. HOPE (Jeon et al., 2026) ablates the inputs of its vertex transformer and reports that combining visual features with hand pose outperforms either alone, suggesting that the two provide complementary information for contact and pressure estimation. Our expert already receives pose information indirectly, as its bone tokens are initialized from the pose decoder and spread to vertices through the MANO skinning weights. We therefore test whether directly attending to the pose tokens provides additional information. We implement this as *one-way* attention, where the expert's bone tokens take the pose tokens as additional keys and values, while the pose tokens never attend back. Thus, the frozen stage-1 stream remains unchanged by the tactile objective and the pose predictions are bit-identical. Table 11 compares the two variants using the same checkpoint, data, and training recipe.

The one-way variant performs worse on OpenTouch (0.635 vs. 0.696 F1) and DexYCB (0.565 vs. 0.572), improves by 0.004 on HOT3D, and has higher force MAE on OpenTouch (0.573 vs. 0.489 kPa). AUROC is nearly unchanged across all three splits (0.979/0.910/0.955 vs. 0.980/0.915/0.959), indicating that the additional pathway does not materially change the ranking of vertices and mainly shifts the operating point without a consistent direction. These results suggest that the pose information available through the existing bone tokens is already sufficient, and that directly attending to the pose tokens provides little additional signal. We therefore omit the additional attention pathway from the final expert.

Variant	OpenTouch		DexYCB		HOT3D		OpenTouch force (kPa)	
	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	F1 ↑	AUROC ↑	MAE ↓	RMSE ↓
w/ one-way attention	0.635	0.979	0.565	0.910	**0.593**	0.955	0.573	2.574
Ours (no pose conditioning)	**0.696**	**0.980**	**0.572**	**0.915**	0.589	**0.959**	**0.489**	**2.508**

Table 11: **One-way attention ablation.** Comparison between the proposed tactile expert and a variant in which the bone tokens additionally attend to the pose stream. **Best** values are shaded.

C LIMITATIONS

Shape calibration on public benchmarks. The β-cache is designed around a hand shape measured once per operator, which is natural in robot data collection but unavailable on public benchmarks without per-subject calibration. We therefore initialize the cache from the first well-visible window on these benchmarks, and can demonstrate the benefit of external calibration only using the ground-truth β available in ARCTIC (A4 in Table 5). The benefit of external calibration in unconstrained settings, and the robustness of the cache when the first well-visible window provides a poor view of the hand, remain to be evaluated.

Scarce tactile supervision. Force supervision in our training data comes from a single tactile glove (OpenTouch) and a single planar pressure pad (PressureVisionDB), limiting the diversity of pressure magnitudes and contact surfaces seen during training. Newer datasets such as EgoPressure (Zhao et al., 2025) provide additional egocentric pressure data, but incorporating them requires reconciling their sensor calibration, surface parameterization, and contact definitions with those of existing datasets. Until tactile datasets adopt more consistent surface representations and pressure units, scaling force supervision across heterogeneous sensors remains challenging.

From hand labels to robot actions. Accurate hand motion and contact labels do not directly determine robot actions. Our real-robot experiments use a fixed inverse-kinematics retargeting procedure with locked abduction joints and a heuristic contact offset, while the retargeting results in Table 3 measure fidelity to a solver applied to the ground-truth hand rather than task success. Mapping human hand motion and contact patterns to dexterous hands with different kinematics, contact geometry, and compliance, as well as exploiting predicted force beyond binary contact, remains outside the scope of this work. Finally, RLHND processes clips offline rather than in real time, and its backbone is substantially larger than crop-based regressors.