Hoi! - A Multimodal Dataset for Force-Grounded, Cross-View Articulated Manipulation

Supplementary Material

Contents

A. Hoi! Dataset Details & Gripper Calibration Details

In the following we give a detailed description of the dataset details & calibrations done for the Hoi! gripper depicted in Supplementary Fig. 2.

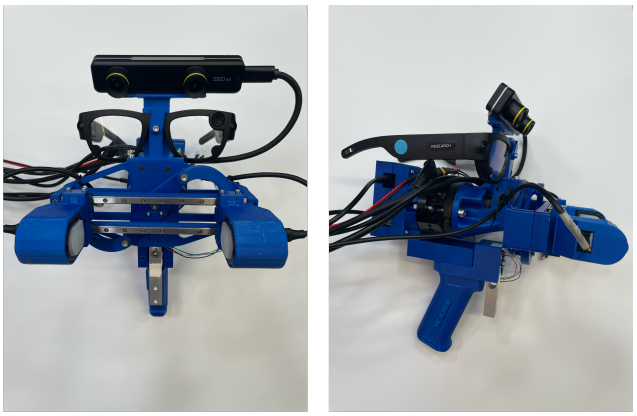
A.1. Recording Device Specs

We show the sensor specifications of the recording devices in Tab. 1.

Plat.Specs	**Cam/FOV**	**Res.**	**Rate**
Aria	2 / 70–150°	1408^2	30Hz
ZED	2 / 90°	1280x720	20Hz
Digit	1 / –	VGA	20Hz
F/T	–	6-axis	100 Hz
iPhone	1 / 77°	4K/LiDAR	30Hz
Spot	5 / 360°	Gray/RGB,D	15Hz
GoPro	1 / 90-122°	4K	60Hz

Supplementary Table 1. Key sensor specifications in the dataset.

Regarding the accuracy of the force-torque measurements we refer to the datasheet. Key specs include an accuracy across all axes of $< 2\%$ and a noise-free resolution @ 100 Hz of 70-100 mN / 0.6-2.1 mNm.



Supplementary Figure 2. **Hoi! Gripper.** The 2-finger parallel gripper is operated through the load cell, where the measured load is translated into gripping force. Interaction force and tactile contact pressure are measured through the Digit and Force-Torque sensors respectively. Aria Glasses and a stereo camera provide pose estimation and wrist-view observations. We will release the design as open source.

A.2. Motor Calibration

The Hoi! gripper's gripping force is modeled as

$$F_i^{(\text{grip})} = g(J(q),\, \eta(I),\, I)$$

where I denotes the motor current, $J(q)$ the gripper Jacobian as a function of the motor position q, and $\eta(I)$ a load-dependent efficiency factor. First, the motor torque is expressed as a proportional function of the current

$$\tau(I_{\text{mA}}) = k_1 I + k_2.$$

with $k_1 = 1.769$ and $k_2 = -0.2214$, as stated in the data sheet. Second, the gripping force is derived via the jacobian of the kinematic relationship $F = J(q)\,\tau$, where τ is calculated as a function of the lever angle q:

$$J(q) = 2\left(-L_1 \sin q - \frac{L_1^2 \sin q \cos q}{\sqrt{L_2^2 - \left(L_1 \sin q\right)^2}}\right).$$

To account for efficiency variations due to load-dependent motor performance and friction, we introduce a load-dependent calibration factor. This factor

is obtained empirically by gripping a force sensor and recording pairs of measured gripping forces and corresponding motor currents. Using least-squares estimation, we determine $\eta(I)$ across multiple load regimes.

A.3. Inter-Sensor Calibration

We use Kalibr [14] to calibrate the gripper in the following order: We first calibrate the intrinisics of the ZED camera, and then the visual-inertial extrinsics to the Zed's IMU. We then use visual-inertial calibration between the ZED images and the IMU in the force-torque sensor to find the extrinsics between stereo camera and FT sensor. The Aria device runs its own intrinsic calibration, and we find the extrinsics through stereo calibration between one Aria and one Zed camera in their overlapping field-of-view

A.4. Gripper Gravity Compensation

To measure the isolated interaction forces between gripper and furniture, we need to compensate for gravitational forces acting on the endeffector, as well as internal biases of the FT sensor. The governing equation is:

$$\mathbf{f}_S^{\text{meas}} = \mathbf{f}_S^{\text{ext}} + \mathbf{f}_S^{g} + \mathbf{b}_f,$$
$$\boldsymbol{\tau}_S^{\text{meas}} = \boldsymbol{\tau}_S^{\text{ext}} + \boldsymbol{\tau}_S^{g} + \mathbf{b}_\tau,$$

where

$$\mathbf{f}_S^{g} = \mathbf{R}_{S\leftarrow W}\, m\, \mathbf{g}_W,$$
$$\boldsymbol{\tau}_S^{g} = \mathbf{r}_{C\leftarrow S} \times \mathbf{f}_S^{g},$$
$$\boldsymbol{\tau}_S^{\text{ext}} = \mathbf{r}_{P\leftarrow S} \times \mathbf{f}_S^{\text{ext}},$$

where $\mathbf{r}_{SP}$ is the vector from the sensor origin to the contact point. Here, $\mathbf{f}_S^{\text{meas}}$ and $\boldsymbol{\tau}_S^{\text{meas}}$ are the measured forces and torques in the sensor frame S, $\mathbf{f}_S^{\text{ext}}$ and $\boldsymbol{\tau}_S^{\text{ext}}$ are the external forces and torques acting on the sensor, $\mathbf{f}_S^{g}$ and $\boldsymbol{\tau}_S^{g}$ are the gravitational forces and torques acting on the sensor, $\mathbf{b}_f$ and $\mathbf{b}_\tau$ are the internal biases of the force-torque sensor, $\mathbf{R}_{S\leftarrow W}$ is the rotation matrix from the world frame W to the sensor frame S, m is the mass of the endeffector assembly, $\mathbf{g}_W$ is the gravity vector in the world frame, and $\mathbf{r}_{C\leftarrow S}$ is the vector from the sensor origin to the center of mass of the endeffector assembly. We measure the mass of the endeffector using a scale, while the center of mass is estimated from the CAD model of the endeffector. $\mathbf{R}_{S\leftarrow W}$ is taken from the Aria SLAM and extrinsic calibration. Internal biases are estimated during no-load conditions, where external forces and torques are zero. We estimate no load measurement windows by subtracting the median filtered force magnitude from the raw force magnitude and thresholding the result. We then estimate the biases by solving the following least-squares problem:

$$\min_{\mathbf{b}_f,\mathbf{b}_\tau} \sum_{k=1}^{N} \left\| \begin{bmatrix} \mathbf{f}_{S,k}^{\text{meas}} - \mathbf{R}_{WS,k}^{\top}\, m\, \mathbf{g}_W - \mathbf{b}_f \\ \boldsymbol{\tau}_{S,k}^{\text{meas}} - \mathbf{r}_{SC} \times (\mathbf{R}_{WS,k}^{\top}\, m\, \mathbf{g}_W) - \mathbf{b}_\tau \end{bmatrix} \right\|^2,$$

where N is the number of no-load samples. Given the estimated biases and the known mass and center of mass, we can now compute the external forces and torques during interaction as

$$\mathbf{f}_S^{\text{ext}} = \mathbf{f}_S^{\text{meas}} - \mathbf{R}_{WS}^{\top}\, m\, \mathbf{g}_W - \mathbf{b}_f,$$
$$\boldsymbol{\tau}_S^{\text{ext}} = \boldsymbol{\tau}_S^{\text{meas}} - \mathbf{r}_{SC} \times (\mathbf{R}_{WS}^{\top}\, m\, \mathbf{g}_W) - \mathbf{b}_\tau.$$

. The results of gravity compensation are depicted in Supplementary Fig. 2. We also apply a Butterworth filter of degree 4 to filter noise.

B. Alignment of Sensors in the Hoi! Dataset Recordings

We give a more in-depth explantion of both temporal and spatial alignment of the multiple sensor streams in the Hoi! dataset. This process allows us to capture interactions over time and across multiple perspectives, as depicted in Fig. 4.

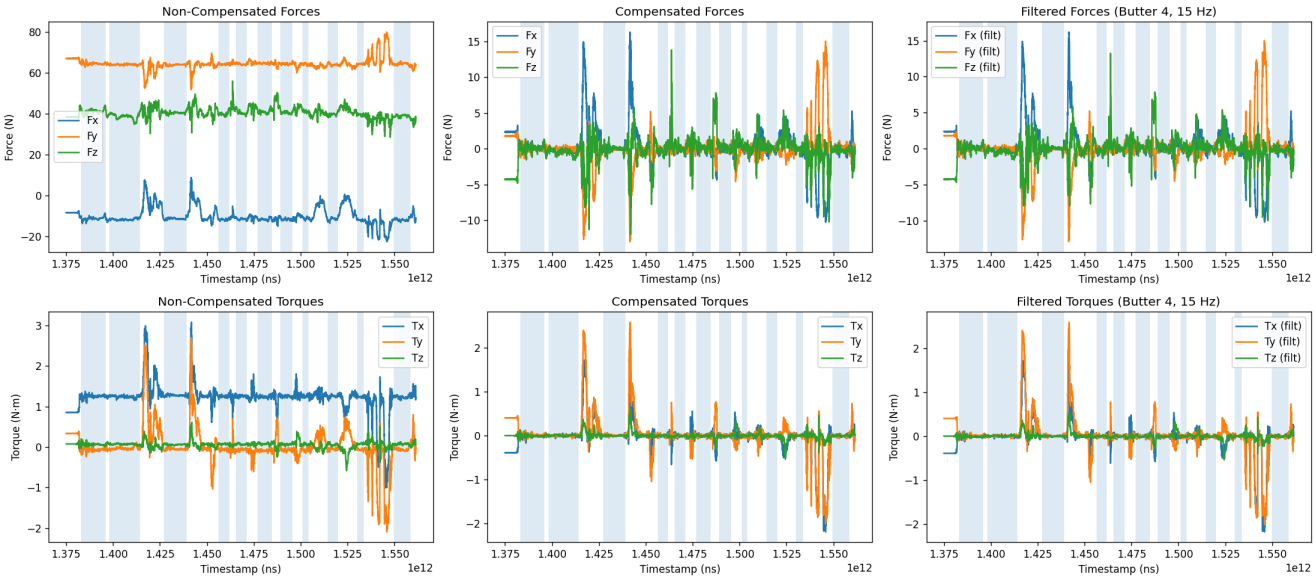
B.1. Time Alignment

As the different recording modules run independently of each other on different internal clocks, we need to align the time frames of the recordings in post processing. During recording, we display a QR code encoding the current Unix timestamp at 25 Hz into each camera stream i. We detect and decode the first QR code, yielding a time pair $(t_{internal}, t_{\text{ref}})$, where $t_{internal}$ is the internal timestamp of the video stream i at which the QR code was captured, and t_{ref} is the corresponding reference Unix timestamp encoded in the QR code. Detecting and decoding this QR code yields a time offset for each video stream with respect to a common reference time, in our case the Aria egocentric camera stream, as

$$\Delta_i = (t_{\text{ref},i} - t_{\text{internal},i}) - (t_{\text{ref,Aria}} - t_{\text{internal,Aria}}),$$

where Δ_i denotes the relative clock offset of stream i with respect to the Aria reference. The uncertainty of this single-shot estimate is dominated by frame-timing quantization: assuming the QR update occurs uniformly $\delta_i \sim \text{Uniform}(0, T_i), i \in \{\text{ArEgo, Sensor}\}$, within the exposure of the first detected frame, the standard deviation of the alignment error is

$$\sigma_{\Delta_i} = \sqrt{\frac{T_i^2}{12} + \frac{T_{\text{Aria}}^2}{12}},$$



Supplementary Figure 3. **Gravity Compensation.** We depict the uncompensated (left), compensated (middle) and filtered compensated (right) forces (upper row) and torques (lower row) of an examplary gripper recording (bathroom_2). The no-load windows are stylized in blue.

Supplementary Figure 4. **Spatial and Temporal Alignment** We both temporally and spatially align all recording modules. This allows us to capture interactions over time across multiple viewpoints.

yielding a 95% confidence interval of approximately $\pm 1.96\,\sigma_{\Delta_i}$. For typical frame rates of 30–60 Hz, this corresponds to a temporal alignment accuracy of roughly 10–25 ms per stream. In a representative

Supplementary Figure 5. **Time Alignment.** The recording is started for all recording modules of a single recording and a QR code encoding the current time is shown to all video streams, so that the individual clocks can be aligned in post processing.

example, when the iPhone records at 60 Hz ($T_i \approx 16.7$ ms) and the Aria at 30 Hz ($T_{\text{Aria}} \approx 33.3$ ms), the resulting uncertainty is $\sigma_{\Delta_i} \approx 10.8$ ms,

B.2. Spatial Alignment

We spatially align all recording devices into a common reference frame using visual localization against the high-resolution 3D scans. We first construct a reference set of dense 2D-3D correspondences from the point cloud and corresponding panoramic images captured with the scanner. We rectify the panoramic images into multiple perspective images with virtual camera parameters $\mathbf{K}, \mathbf{R}, \mathbf{t}$. We further convert the point cloud into a mesh using the Leica proprietary software and render depth maps from the same virtual camera poses. The rendered depth maps are then back-projected into 3D space using the known virtual camera intrinsics $\mathbf{K}$ and extrinsics $(\mathbf{R}, \mathbf{t})$ as

$$\mathbf{P} = \mathbf{R}^{-1}\mathbf{K}^{-1} \begin{bmatrix} u\,d \\ v\,d \\ d \end{bmatrix} - \mathbf{R}^{-1}\mathbf{t},$$

where (u, v) are pixel coordinates in the image plane and d is the corresponding depth value at that pixel. This operation transforms each depth pixel into a 3D point $\mathbf{P}$ in the global coordinate frame, thereby establishing dense 2D-3D correspondences between the rendered depth maps and the rectified panoramic images. We assume zero drift and global consistency of the per-device SLAM, which reduces the global registration problem to a single rigid 3D-3D alignment. While in principle a single localized frame would suffice for this alignment, we automatically select a set of high-quality keyframes for robust registration. To obtain these keyframes, we filter all frames of a trajectory according to feature density, sharpness, and scene depth. Specifically, we retain frames with a high number of ORB [15] keypoints, a high variance of the Laplacian (indicating low motion blur) and a high mean estimated depth. With the latter, we filter out frames where the device is very close to the furniture, where usually no good references are in view. For Aria and GoPro cameras, which do not provide depth maps, we use DepthAnything v2 [18]. On this filtered subset, we extract DINOv2 [11] features and perform farthest-point sampling [6, 13] in feature space to select N diverse and representative keyframes per trajectory. We estimate the 6-DoF pose of each keyframe through hloc [16] with the 2D-3D database constructed from the Leica scan and robustly estimate a single rigid transformation $\mathbf{T}^{\text{query}}_{\text{world}}$ between each sensor trajectory and the shared world frame. Since we also have the corresponding query poses $\mathbf{T}^{\text{cam}_i}_{\text{query}}$, with query $\in \{\text{iPhone}, \text{Aria}, \text{UMI}\}$, we can estimate a single rigid transformation $\mathbf{T}^{\text{query}}_{\text{world}}$ aligning each query trajectory to the common world frame by

$$\mathbf{T}^{\text{query}}_{\text{world}} = \mathbf{T}^{\text{cam}_i}_{\text{world}}\, \mathbf{T}^{\text{cam}_i}_{\text{query}}{}^{-1}.$$

After localizing each keyframe, we compute the Frobenius distances between all pairwise combinations of the N estimated transformations $\mathbf{T}^{\text{query}}_{\text{world}}$ and reject outliers based on a threshold relative to the median distance. We then average the remaining inlier transformations to obtain a final robust estimate of $\mathbf{T}^{\text{device}}_{\text{world}}$. This simple outlier rejection strategy is sufficient, as Hierarchical Localization already performs RANSAC [5]-based PnP pose estimation internally. Finally, we transform

Supplementary Figure 6. **Spot Teleoperation** The Spot Robot is teleoperated using a Meta Quest 3. We retarget the remote's 6-DoF pose to the gripper and control the base using the remote mounted Joystick.

all Aria poses and hand poses into the common world frame using the estimated transformation $\mathbf{T}_{\text{world}}^{\text{Aria}}$, yielding globally aligned 6-DoF trajectories for the Aria head, hands, and Hej gripper. The iPhone cameras are statically mounted, and we therefore directly measure their poses in the world frame during spatial calibration. The UMI gripper poses are transformed into the world frame using the estimated transformation $\mathbf{T}_{\text{world}}^{\text{UMI}}$.

C. Spot Recordings

As mentioned in the main paper, we collect robotic data for a subset of interactions using a Boston Dynamics Spot Robot. Here, we record the robot's joint states, surrounding RGB-D cameras, the gripper RGB-D camera as well as Aria data using a wrist-mounted Aria, imitating the wrist viewpoint of gripper and wrist recordings. As shown in Fig. 6, the robot is teleoperated using a Meta Quest 3. Here the operator controls the robot base using the joystick on the Quest remote, while the remote's 6-DoF pose is retargeted to the Spot gripper. The opening angle is also controlled via button on the remote.

D. Evaluations

In the following we give more in-depth description of how we created the evaluation groundtruth used in our evaluations from the Hoi! dataset.

D.1. In-The-Wild Articulation Estimation

In this section, we evaluate a set of different approaches to articulation estimation. This task concerns recovering articulation parameters (axis and position) as well as the articulation type from visual observations. We first evaluate ArtiPoint [17], a recent method for articulation estimation in the wild. We also include the Gaussian-Splatting-based approach ArtGS [10]. Furthermore, we investigate how GPT-5, as a state-of-the-art vision-language model (VLM), can infer articulation types from both egocentric and exocentric observations. For this analysis, we make use of our dataset's posed RGB frames (both egocentric and exocentric), articulation annotations, and provided 3D ground-truth.As the Aria does not provide dense depth, we generate dense RGB-D data, we employ MapAnything [9] to predict dense depth for each posed frame in our interaction sequences. We convert these depth estimates into metric scale by rendering depth maps from our 3D ground-truth meshes under the corresponding camera pose. We then robustly compute a global scale factor by forming a per-pixel scale histogram and selecting the scale as the mode

$$s = \arg\max_{s} h(s)$$

, where $h(s)$ denotes the histogram of per-pixel ratios between predicted and rendered depths. This is necessary because certain regions in the predicted depth differ from the rendered depth (*e.g.*, the operator's hand or articulated parts present in the predicted depth but absent in the mesh rendering). We finally apply the scale factor to obtain dense, metrically accurate depth for each frame. The actual ground truth articulation parameters are provided using a light-weight manual annotation tool presented in [17].

D.2. Tactile Force Estimation

We evaluate the utility of our dataset for the task tactile force estimation from gel-based tactile sensors. This task aims to estimate contact forces acting on the sensor's surface solely from the tactile images captured by the sensor during contact. We specifically focus on estimating the normal and tangential forces, as these are most relevant for manipulation tasks. We combine the tafigctile images provided by the Hoi! gripper's Gelsight Digit sensors with the corresponding forces provided by the gripper's force-torque sensor and the gripper's gripping forces into ground-truth labels. We evaluate two versions the Sparsh model [8], the SOTA method for self-supervised tactile representations. We evaluate both the DINO [1] and DINOv2 [11] decoder, with a fine-tuned force estimation decoder (referred to as 'Task 1' in the original Sparsh paper). The interaction forces during grasping can be decomposed into two components: a normal preload $F_i^{(\text{grip})}$ resulting from the motor-torque–induced gripping force, and an external reactive force $F_i^{(\text{ext})}$ exerted by the environment.

The external force is measured by the force–torque (FT) sensor as $F_s^{(\text{ext})}$ in the sensor's local coordinate frame. The preload $F_i^{(\text{grip})}$, while sensed by the tactile sensors (Digits), represents an internal force and is therefore not captured by the FT sensor. To express all forces in a consistent coordinate system, we define an interaction frame i, whose axes are aligned with the Digit frames. The interaction frame is defined to be coaxial with the Digit frames; however, since the two Digits face each other, corresponding axes have opposing directions ($x_{\text{L}} = -x_{\text{R}}$, $z_{\text{L}} = -z_{\text{R}}$, $y_{\text{L}} = y_{\text{R}}$). This does not affect our analysis, as we only consider the sum of absolute force magnitudes. The FT-sensor measurements are rotated into this frame as $F_i^{(\text{ext})} = \mathbf{R}_{i \leftarrow s}\, F_s^{(\text{ext})}$, where $\mathbf{R}_{i \leftarrow s}$ denotes the rotation from the sensor frame s to the interaction frame i. All subsequent equations and force components are defined in this frame. Considering the hardware configuration of the *Hoi* gripper, the combined absolute forces are computed as

$$
\begin{aligned}
F_i^{(\text{tang})} &= \Big\| \sum_{k \in \{\text{L},\,\text{R}\}} |F_{i,\,k,\{x,y\}}^{(\text{ext})}| \Big\|_2, \\
F_i^{(\text{norm})} &= \sum_{k \in \{\text{L},\,\text{R}\}} |F_{i,\,k,z}^{(\text{ext})}|, \\
F_i^{(\text{comb})} &= \sqrt{\big(F_i^{(\text{tang})}\big)^2 + \big(F_i^{(\text{norm})}\big)^2}.
\end{aligned}
\tag{1}
$$

The gripping force is estimated from the gripper's torque–current relationship, its Jacobian, and a load-dependent calibration factor as $F_i^{(\text{grip})} = g(J(q),\, \eta(I),\, I)$. Since the Sparsh models are trained on force data within the range of $[4, 4, 5]\,\text{N}$ along the x, y, and z axes, respectively, we clip the combined ground-truth magnitudes to $F_{\text{max}}^{(\text{norm})} = 10\,\text{N}$ and $F_{\text{max}}^{(\text{tang})} = \sqrt{32}\,\text{N}$ to ensure consistency between training and evaluation distributions. This avoids extrapolation to unseen force magnitudes and provides a fair comparison of model performance. The gripping force is estimated using the gripper's torque-current relationship, its Jacobian and a load dependent calibration factor $F_{\text{grip}} = g(J(q),\, \eta(I),\, I)$. As the Sparsh models are trained on force data within therange of $[4, 4, 5]\,\text{N}$ for x, y, and z, respectively, we clip our combined ground-truth magnitudes to $F_{\text{max,normal}} = 2 \times 5 = 10\,\text{N}$ and $F_{\text{max,tangential}} = \sqrt{4^2 + 4^2} = \sqrt{32}\,\text{N}$ to ensure consistency between training and testing distributions. This avoids extrapolation to unseen force magnitudes and provides a fair assessment of model performance. The extended evaluation results are depicted in Tab. 2.

As depicted in Fig. 7, we observe that the force predictions generally under-predict the tactile forces, even though the GT is clipped to be within the training range.

D.3. Visual Force Estimation

We evaluate the utility of our dataset to the task of visual force estimation. In this task a 3D interaction force is estimated alongside an affordance (interaction type) in order to complete a given manipulation goal. In our context the input might be an RGB-D image of a drawer alongside the prompt "open the drawer" and the model would predict where to interact and what force to apply. We evaluate the ForceSight model [3], a model that aims to predict forces as part of visual-force goals for robotic manipulation, demonstrating that force goals can significantly increase robotic manipulation performance. The ForceSight dataset consists of interaction sequences that include posed RGB-D observations, per-frame force–torque (FT) readings, and gripping forces. Each sequence is paired with an open-language goal derived from our interaction annotations. Using the Hoi! gripper's FT sensor, we generate per-image force–torque labels and leverage the 3D ground-truth trajectories provided by our dataset. We evaluate the model across a diverse subset of six environments. Because raw force-torque signals may include operator-induced forces in directions unrelated to the articulation (*e.g.*, internal stresses not required for the intended motion), we report results on both the raw data and a motion-aligned variant. For fair comparison, we project the measured force vector $\mathbf{f}$ onto the gripper's linear velocity $\mathbf{v}$ using $f_{\parallel} = \mathbf{f} \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|}$, and similarly project the torque vector $\boldsymbol{\tau}$ onto the rotational velocity $\boldsymbol{\omega}$ as $\tau_{\parallel} = \boldsymbol{\tau} \cdot \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$. This removes force and torque components that do not contribute to the articulated interaction, resulting in a fairer evaluation signal.

D.4. Multimodal Learning for Real-World Robotics

We conduct a force-centric experiment to demonstrate **Hoi!**'s utility for multimodal skill learning. Specifically, we train a visual force predictor on **Hoi!** gripper data and integrate the learned force prior into Spot's position-force controller. Given an RGB-D observation and the articulated-part class label, the model predicts a feed-forward interaction force $\mathbf{f}_{ff}$.

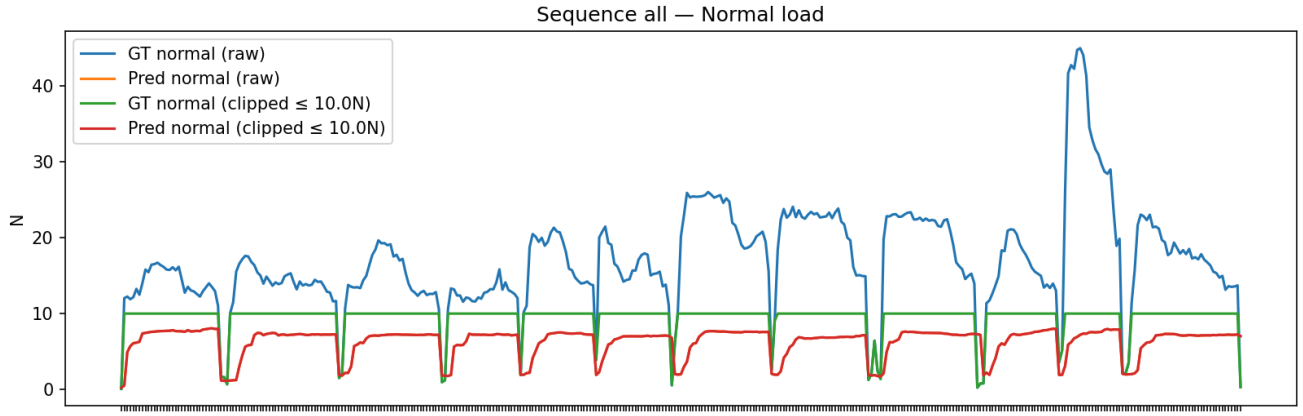
Our predictor uses frozen DINOv2 image features and a small MLP head to estimate a residual on top of a class-specific force prior. More formally, given an observation $o = (I, d, c)$ with RGB image I, depth-derived features d, and class label c, the model predicts

$$
\hat{\mu} = \mu_c + \alpha\, h_\theta(\phi_{\text{DINO}}(I)),
\tag{2}
$$

where μ_c denotes the class prior in log-force space,

Supplementary Table 2. **Tactile Force Estimation.** We show the evaluation results for our tactile force estimation evaluation per location and split into tangential, normal and combined forces over 2 digit images.

	Tangential		Normal		Combined	
Location	DINO	DINOv2	DINO	DINOv2	DINO	DINOv2
bathroom_2	2.07 [1.88, 2.25]	2.02 [1.81, 2.23]	4.81 [4.54, 5.09]	4.97 [4.69, 5.26]	4.92 [4.64, 5.21]	5.17 [4.88, 5.47]
bedroom_4	3.23 [3.09, 3.37]	2.77 [2.63, 2.92]	3.22 [3.07, 3.38]	3.72 [3.60, 3.85]	3.41 [3.22, 3.60]	3.96 [3.81, 4.11]
bedroom_6	3.39 [3.19, 3.59]	3.77 [3.52, 4.00]	3.48 [3.23, 3.71]	3.43 [3.22, 3.63]	4.31 [4.04, 4.57]	4.45 [4.24, 4.66]
kitchen_7	3.05 [2.82, 3.27]	3.83 [3.67, 4.00]	2.76 [2.60, 2.92]	3.50 [3.39, 3.61]	3.46 [3.28, 3.65]	3.19 [2.97, 3.41]
office_1	3.63 [3.41, 3.86]	3.76 [3.52, 4.01]	3.88 [3.62, 4.14]	4.07 [3.86, 4.27]	4.61 [4.30, 4.92]	4.91 [4.67, 5.17]
livingroom_1	2.54 [2.31, 2.77]	2.61 [2.40, 2.82]	3.63 [3.34, 3.91]	3.71 [3.46, 3.95]	3.89 [3.56, 4.21]	3.90 [3.61, 4.19]
Overall	**3.07** [2.87, 3.26]	**3.18** [2.99, 3.38]	**3.45** [3.24, 3.66]	**3.79** [3.61, 3.96]	**3.86** [3.62, 4.11]	**4.11** [3.90, 4.33]



Supplementary Figure 7. **Tactile force Estimation.** We depict the GT forces, clipped GT forces and the estiamted forces (DINOv2) for an examplary recording.

ϕ_{DINO} is the frozen visual backbone, h_θ is the learned regression head, and α scales the visual residual. We train the network to regress the log-transformed peak interaction force using an ℓ_2 loss,

$$\mathcal{L} = \|\hat{\mu} - \log(1 + F)\|_2^2 . \tag{3}$$

At test time, the predicted force prior is mapped back to force space and injected as a feed-forward term into the controller.

As a baseline, we use standard impedance control,

$$\mathbf{f}_{cmd} = \mathbf{K}\,\Delta\mathbf{r} + \mathbf{D}\,\Delta\mathbf{v}, \tag{4}$$

where $\Delta\mathbf{r}$ and $\Delta\mathbf{v}$ denote pose and velocity errors, respectively. In the position-force variant, we augment this controller with the predicted feed-forward force term:

$$\mathbf{f}_{cmd} = \mathbf{K}\,\Delta\mathbf{r} + \mathbf{D}\,\Delta\mathbf{v} + \mathbf{f}_{ff}. \tag{5}$$

We evaluate on articulated objects in our lab with increasing mechanical stiffness. For each trial, the robot is initialized in front of the object, and a simple geometric heuristic selects the articulation type (eccentric handle → revolute, centric handle → prismatic), such that failures are predominantly attributable to interaction force selection rather than articulation misclassification. Each articulated part is operated five times. As shown below, the learned force prior substantially improves real-world success rate (SR), highlighting **Hoi!**'s potential for multimodal policy learning in real-world manipulation settings.

Control Type	**Drawer**	**Oven**	**Dishwasher**
Position (SR [%])	100	0	0
Position-Force (SR [%])	100	80	60
Avg. Force Pred. [N]	16,80	46,38	53,29

D.5. Hand Pose Estimation

As our dataset captures not only gripper interactions but also hand interactions, we additionally explore the performance of hand-pose estimation across these viewpoints. Because the Aria MPS hand keypoints are automatically generated-derived from stereo and globally optimized trajectories but not from manual annotations or motion-capture systems, we treat this evaluation as an exploratory analysis rather than a definitive benchmark.

We employ the method of Pavlakos et al. [12], which

Supplementary Table 3. **Hand Pose Estimation** Average PCK@0.15 on our evaluation locations and compared to the Hamer baseline performace on New Days, VISOR and Ego4D datasets.

Location / Dataset	**PCK**
bathroom_2	0.757
bedroom_4	0.764
bedroom_6	0.708
kitchen_7	0.535
office_1	0.732
livingroom_1	0.748
Overall	**0.699**
New Days [2]	0.888
VISOR [4]	0.893
Ego4D [7]	0.844

has shown strong in-the-wild performance. Using the Aria MPS trajectories, we project hand keypoints into egocentric frames and compute the commonly used PCK metric [19].

While the results (Tab. 3) show a noticeable performance gap relative to controlled benchmarks, this is expected given the challenging characteristics of our real-world setting - fast hand motions, natural manipulation behaviors, and lower-resolution egocentric imagery. Rather than indicating deficiencies, these findings highlight the difficulty of egocentric manipulation scenes and underline the opportunity for future methods to better leverage the rich multimodal signals present in our dataset.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021.

[2] Tianyi Cheng, Dandan Shan, Ayda Sultan, Richard E. L. Higgins, and David F. Fouhey. Towards a richer 2d understanding of hands at scale. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2023. Curran Associates Inc.

[3] Jeremy A. Collins, Cody Houff, You Liang Tan, and Charles C. Kemp. Forcesight: Text-guided mobile manipulation with visual-force goals, 2023.

[4] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations, 2022.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[6] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[7] 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*, pages 18995–19012, 2022.

[8] Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *8th Annual Conference on Robot Learning*, 2024.

[9] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. Mapanything: Universal feed-forward metric 3d reconstruction, 2025.

[10] Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Artgs: Building interactable replicas of complex articulated objects via gaussian splatting, 2025.

[11] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[12] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

[13] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

[14] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. pages 4304–4311, 2016.

[15] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[16] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[17] Abdelrhman Werby, Martin Büchner, Adrian Röfer, Chenguang Huang, Wolfram Burgard, and Abhinav

Valada. Articulated object estimation in the wild, 2025.

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2, 2024.

[19] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013.