Ego-Pi: VLA Fine-Tuning for Ego-Centric Human and Robot Data

Supplementary Material

7.1. Implementation Details

For the robot setup, we use the Galaxea R1 Pro from Galaxea Dynamics, equipped with a ZED mini mounted on the robot head. Each end-effector is fitted with a Tesollo hand featuring 20 joints per hand. For the tomato sorting task only, we use Inspire hands. Additionally, Ardurocam wrist cameras are mounted on each wrist, providing a 160° field-of-view. Note that only the robot uses wrist cameras and no wrist cameras are used when collecting human data.

For teleoperation, we utilize Quest controllers mounted on Manus gloves. The Quest controllers track the 6D pose of the operator's wrists relative to the headset. The robot reproduces the corresponding wrist poses relative to its head using inverse kinematics and PD control. Manus gloves capture the operator's finger joint angles, which are mapped to the robot's hand joints to enable dexterous control. Both the Manus gloves and Quest controllers record motion data at 100 Hz.

During inference, the policy is deployed on a server cluster using a single NVIDIA H200 GPU. The resulting inference rate is 13 Hz on a single H200 card.

7.2. Data Collection

To collect robot data, the operator wears the Quest 3 on the chest to visualize the robot workspace directly while standing near the robot. The system also supports first-person-view teleoperation, where the operator wears the Quest 3 headset to view the robot's ego-centric perspective through a ZED mini stereo camera mounted on the robot's head. However, due to the weight of the Quest 3 and discomfort during extended sessions, we primarily used the chest-mounted configuration during data collection.

To collect ego-centric human data, we use a ZED mini stereo camera mounted on a tripod. During data collection, synchronized stereo image pairs are recorded. To extract ground-truth labels for human hand actions, we process these stereo images using HaMeR [10], which reconstructs 21 MANO [13] hand keypoints for each hand. Corresponding keypoints between the stereo views are triangulated to obtain their 3D positions via stereo matching. For the tomato sorting task only, we used Manus gloves equipped with Quest controllers (identical to the teleoperation setup) to collect human data.

To ensure the validity of the collected ground-truth actions, we verify that the replayed trajectories produce physically feasible robot motions in open-loop. In a simple pick-and-place task, we observed that the reconstructed human hand trajectories successfully executed the cube pickup and closely matched the corresponding robot motions, confirming that the human actions can be reliably transferred to the robot domain in open-loop replay.

Also, the subtask instructions were labelled only for the tomato sorting task. Other tasks did not use subtask generation as they were shorter horizon tasks.

8. Joint Mapping between Human and Robot Hands

Let $q \in \mathbb{R}^{20}$ be the human hand joint angles measured by the Manus glove (Fig. 8), and let $q_{\text{robot}} \in \mathbb{R}^{20}$ be the corresponding robot joint angles (e.g., Tesollo hand).

We map each human joint angle to its robot counterpart using a per-joint offset δ_i and a scaling factor f_i:

$$q_{\text{robot},i} = \big(q_i + \delta_i\big)\, f_i, \qquad i \in \{1, \ldots, 20\}. \quad (6)$$

Here, δ_i is an additive offset and f_i is a multiplicative scaling factor. The specific per-joint values used in our implementation are listed below.

```
deg2rad = (pi/180)

q_robot[0] = (38.5 - q[1]) * 0.7 * deg2rad
```

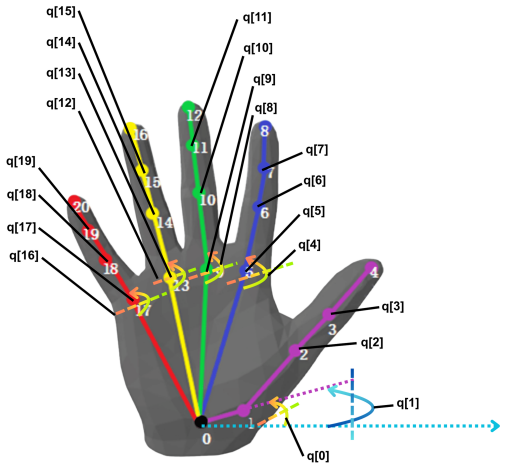

Figure 8. Hand labelled with joint angle locations

```
q_robot[1] = (q[0] + 36.0) * 1.2 * deg2rad
q_robot[2] = (q[2] + 10.0) * 1.2 * deg2rad
q_robot[3] = (q[3] +  5.0) * 1.2 * deg2rad

q_robot[4]  = q[4]  * 1.0 * deg2rad
q_robot[5]  = q[5]  * 1.1 * deg2rad
q_robot[6]  = q[6]  * 1.1 * deg2rad
q_robot[7]  = q[7]  * 1.0 * deg2rad

q_robot[8]  = q[8]  * 1.0 * deg2rad
q_robot[9]  = q[9]  * 1.0 * deg2rad
q_robot[10] = q[10] * 1.0 * deg2rad
q_robot[11] = q[11] * 1.0 * deg2rad

q_robot[12] = q[12] * 1.0 * deg2rad
q_robot[13] = q[13] * 1.0 * deg2rad
q_robot[14] = q[14] * 1.0 * deg2rad
q_robot[15] = q[15] * 1.0 * deg2rad

if q[17] > 55 and q[18] > 25:
    q_robot[16] = abs(q[16]) * 2.0  * 1.0
        * deg2rad
else:
    q_robot[16] = abs(q[16]) / 1.5 * 1.0 *
         deg2rad

q_robot[17] = q[16] * 1.0 * deg2rad
q_robot[18] = q[17] * 1.0 * deg2rad
q_robot[19] = q[19] * 1.0 * deg2rad
```

In order to map human hand joint angles provided by HaMeR to robot hand joints, we perform a similar projection with different offsets and scaling factors, as shown below:

```
q_robot[0]  = q[0]
q_robot[1]  = q[1]
q_robot[2]  = q[2]
q_robot[3]  = q[3]

q_robot[4]  = (q[4] - 0.12) * 1.0
q_robot[5]  = q[5]
q_robot[6]  = (q[6]  + 0.18) * 1.1
q_robot[7]  = (q[7]  + 0.18) * 1.1

q_robot[8]  = q[8]
q_robot[9]  = q[9]
q_robot[10] = (q[10] + 0.18) * 1.1
q_robot[11] = (q[11] + 0.18) * 1.1

q_robot[12] = (q[12] + 0.09) * 1.0
q_robot[13] = q[13]
q_robot[14] = (q[14]) * 0.9
q_robot[15] = q[15]

q_robot[16] = (q[16]) * -1.1
q_robot[17] = (q[17] + 0.24) * 1.0
q_robot[18] = (q[18]) * 1.2
q_robot[19] = (q[19]) * 1.3
```

9. Model Details

The hyperparameters for fine-tuning the $\pi_{0.5}$ policy are shown in Table 1.

Table 1. Hyperparameters for $\pi_{0.5}$ fine-tuning.

Hyperparameter	Value
Optimizer	AdamW
β_1	0.9
β_2	0.95
Weight Decay	0
Gradient Clip Norm	1.0
LR Schedule	Cosine
Warmup Ratio	0.001
Batch Size	128
Training Steps	5000