Technical Report: Zero-shot dexterity for low-cost, tendon-driven robotic hands.

0.1 Introduction

Motivation. Toward learning dexterous manipulation from human demonstrations, the widespread use of low-degree-of-freedom grippers as robotic end-effectors introduces fundamental limitations. Grippers are inherently incapable of expressing many fine-grained manipulation behaviors due to their restricted action space, which prevents faithful reproduction of human dexterity. Moreover, when human demonstrations are collected with gripper-based robots in mind, operators are often required to artificially constrain their hand motions into gripper-like configurations. This mismatch not only deviates from natural human manipulation but also limits the scalability and morphological completeness of the collected data.

To enable zero-shot transfer from human demonstrations to robotic systems, we argue that it is necessary to move beyond gripper-based end-effectors and toward human-like dexterous hands that more closely align with the structure and capabilities of human manipulation.

System Setup. To study this problem, we employ a single Kinova robotic arm equipped with a RUKA dexterous hand, configured to mimic a human arm in a tabletop manipulation setting. Human demonstrations are collected using Project Aria Gen1 smart glasses as a multimodal egocentric sensing device. While this setup enables zero-shot transfer from human data to robotic execution, it also introduces several key challenges.

Challenges. First, there exists a significant visual domain gap between training and deployment. Policies are trained exclusively on egocentric human data; however, at inference time, the robot arm and hand appear in the visual observations, resulting in out-of-distribution inputs for vision-based policies. Second, there is a morphology gap between human hands and robotic end-effectors. Although modern dexterous hands approach human-like mechanical design, differences in actuation, kinematic constraints, and the diversity of human hand anatomies present in the dataset prevent direct reuse of human hand trajectories for robotic execution. Third, to learn general and reactive manipulation policies, the system must operate in closed loop, with actions conditioned on a continuously changing environment. However, because the data are collected from humans wearing smart glasses and performing tasks naturally, frequent head and torso movements lead to unstable egocentric observations. In the absence of explicit 3D scene reconstruction, this results in temporally inconsistent environmental representations.

0.2 Method

Unified action space. To address these challenges, we adopt a representation strategy inspired by prior work such as EgoZero and Point-Policy, modeling the human hand as a unified end-effector in a morphology-agnostic point space. We posit that fingertips dominate physical contact and collision during manipulation. Accordingly, we represent each hand state as a compact set of nine 3D keypoints, consisting of five fingertip positions and four points encoding the wrist pose and orientation. This abstraction preserves essential contact geometry while reducing variance between training and deployment.

Using the built-in hand tracking model of the Project Aria smart glasses, we obtain per-frame image–hand pose pairs expressed in the camera coordinate frame. For each timestep t, the raw hand landmarks are preprocessed into a unified end-effector representation

$$\mathcal{H}_t = \left\{ \mathbf{p}_t^{(i)} \right\}_{i=1}^{9}, \quad \mathbf{p}_t^{(i)} \in \mathbb{R}^3.$$

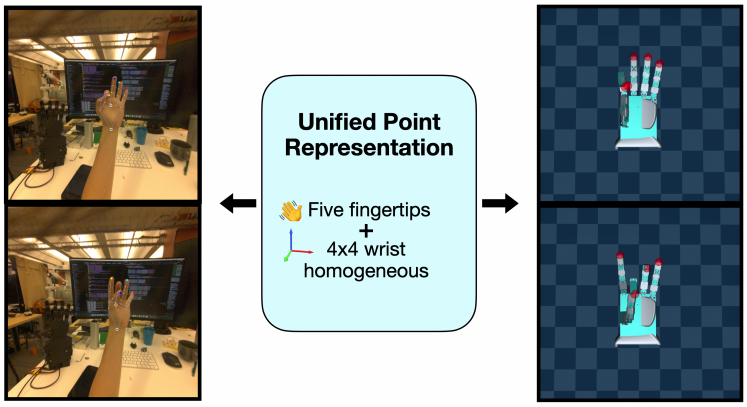

Figure 1: Aligning the human hand and robot hand action representation

where $\{\mathbf{p}_t^{(i)}\}_{i=1}^{5}$ correspond to fingertip positions and the remaining points encode the wrist transformation. These representations are stored and used as action supervision during training.

At test time, given the calibrated camera-to-robot transformation $\mathbf{T}_{\text{cam}\leftarrow\text{base}}$ and the real-time joint configuration $\mathbf{q}_t$ of the Kinova arm, we compute the forward kinematics of the robotic hand to recover fingertip positions and wrist orientation in the camera frame:

$$\hat{\mathbf{h}}_t = \text{FK}(\mathbf{q}_t; \mathbf{T}_{\text{cam}\leftarrow\text{base}}).$$

The same preprocessing pipeline is applied to $\hat{\mathbf{h}}_t$ to obtain a nine-point end-effector representation aligned with the human hand representation used during training. This shared representation enables consistent zero-shot transfer from egocentric human demonstrations to robotic execution.

Hand Retargeting. Human hands and robotic dexterous hands differ substantially in kinematic structure and degrees of freedom. In particular, the human hand exhibits approximately 21 degrees of freedom, whereas the RUKA hand has only 13 degrees of freedom. As a result, fingertip configurations observed in human demonstrations are often kinematically infeasible for the robot. A direct inverse kinematics (IK) mapping from human fingertip targets to the robot hand therefore yields the closest reachable configuration, but may significantly distort the intended hand pose. Since object interaction is highly sensitive to the full dexterous hand configuration, such naive IK-based retargeting can lead to contact patterns that deviate substantially from the original human demonstration.

To address this mismatch, we propose a point-space retargeting formulation that aligns human and robot finger configuration spaces prior to policy learning. Specifically, for each robotic finger, we construct a fingertip configuration point cloud by randomly actuating the finger joint angles in simulation Let $\boldsymbol{\theta} \in \mathbb{R}^{d_r}$ denote the robot finger joint angles. Using forward kinematics $\mathcal{F}_r(\cdot)$, we obtain the corresponding fingertip position

$$\mathbf{p}_r = \mathcal{F}_r(\boldsymbol{\theta}) \in \mathbb{R}^3. \tag{1}$$

By uniformly sampling joint configurations over valid joint limits and accumulating the resulting fingertip positions, we obtain a robot fingertip point cloud $\mathcal{P}_r = \{\mathbf{p}_r^{(i)}\}_{i=1}^{N_r}$.

In parallel, we collect egocentric human hand motion data by having multiple people wear smart glasses and perform free-form hand motions for approximately two minutes. Using the built-in hand tracking model, we extract per-frame human fingertip positions and aggregate them into a human fingertip point cloud $\mathcal{P}_h = \{\mathbf{p}_h^{(j)}\}_{j=1}^{N_h}$ for each finger.

We then align the human and robot fingertip point clouds using Iterative Closest Point (ICP). ICP alternates between establishing point correspondences and optimizing a rigid transformation $(\mathbf{R}, \mathbf{t})$ that minimizes the alignment error:

$$\min_{\mathbf{R}\in SO(3), \mathbf{t}\in\mathbb{R}^3} \sum_j \left\| \mathbf{p}_r^{(\pi(j))} - (\mathbf{R}\mathbf{p}_h^{(j)} + \mathbf{t}) \right\|^2, \tag{2}$$

where $\pi(j)$ denotes the nearest neighbor correspondence from the human point cloud to the robot point cloud. This optimization is solved iteratively by alternating between correspondence assignment and closed-form rigid alignment.

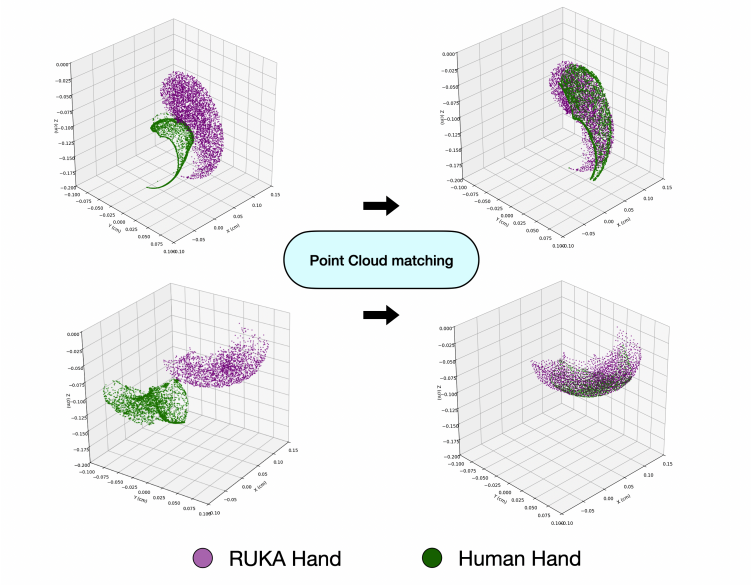

Figure 2: Aligning the human hand and robot hand action representation

The resulting transformation $(\mathbf{R}, \mathbf{t})$ is treated as a finger-specific retargeting map that projects human fingertip positions into the robot's reachable configuration space. We apply this transformation as a preprocessing step to all training demonstrations:

$$\tilde{\mathbf{p}}_h = \mathbf{R}\mathbf{p}_h + \mathbf{t}, \tag{3}$$

ensuring that all policy supervision targets lie within the robot's kinematic feasibility region.

By retargeting demonstrations in point space rather than joint space, our method preserves the geometric structure of human hand-object interactions.

Unified State Representation. To align state representations across demonstrations and deployment, we represent objects using a set of semantic keypoints. Ensuring consistent object semantics across all demonstrations, timesteps, and inference episodes is nontrivial due to viewpoint changes and appearance variation. To address this, we design an interactive annotation interface that allows a user to manually specify a set of semantic keypoints on the object of interest. Given the annotated keypoints in an initial reference frame, we employ a feature-level correspondence model

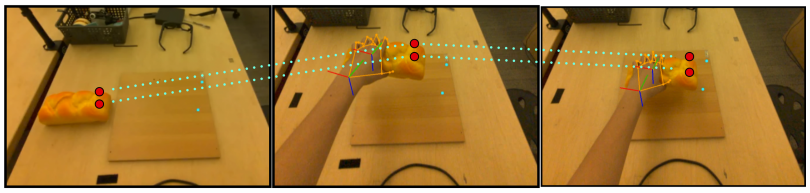

Figure 3: Aligning the human hand and robot hand action representation

(DIFT) to automatically identify the corresponding keypoints in the first frame of each demonstration, as well as in the initial frame at inference time. Formally, for an object keypoint $\mathbf{o}^{(k)} \in \mathbb{R}^3$, we establish correspondences

$$\mathbf{o}_{t=0}^{(k)} = \text{DIFT}(\mathbf{o}_{\text{ref}}^{(k)}),$$

ensuring semantic consistency across sequences.

Once initialized, we apply a point tracking model (AllTracker) to propagate each keypoint temporally across all frames of a demonstration:

$$\{\mathbf{o}_t^{(k)}\}_{t=0}^{T} = \text{AllTracker}(\mathbf{o}_0^{(k)}),$$

yielding stable and temporally consistent object keypoint trajectories. As shown in Figure 3, this pipeline produces robust semantic correspondences and stable point tracking across diverse demonstrations and deployment scenarios.

Data Processing And Augmentation We unify all environments into a single *egocentric point space*, aligning robot perception between training and test time. Policies are trained using a deterministic Transformer architecture operating on this unified representation. At time step t, the policy takes as input the current robot hand state $\mathbf{s}_t \in \mathbb{R}^9$ together with a set of egocentric 2D object points $\mathcal{P}_t$, both expressed in the current camera frame. The policy predicts a future chunk of robot actions over a fixed horizon H, mapping the current observation to a sequence of future actions. During data collection, natural head and torso motion causes the egocentric camera frame to change over time, so future robot states and actions are observed in different camera frames. To maintain frame consistency, we leverage SLAM to obtain relative camera transformations between frames. Using these transformations, we reproject all future robot states and action targets back into the initial camera frame at time t. As a result, each training example follows the form

$$(\mathbf{s}_t, \mathcal{P}_t) \longrightarrow \tilde{\mathcal{A}}_{t:t+H-1},$$

where the action sequence $\tilde{\mathcal{A}}_{t:t+H-1}$ is fully expressed in a single, fixed egocentric frame. This design removes camera motion as a confounding factor and enables the policy to learn object-centric manipulation dynamics directly in point space. By operating in a unified point-space representation, the observation variance is significantly reduced compared

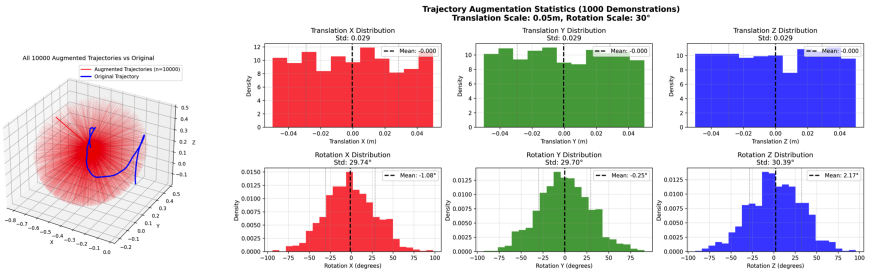

Figure 4: Aligning the human hand and robot hand action representation

to image-based policies. To further encourage the policy to focus on the current object state rather than the absolute robot configuration, we apply data augmentation directly in point space. Given the initial robot state $\mathbf{s}_t$, we sample a synthetic rigid transformation $\mathbf{T}_{\text{aug}} \in SE(3)$ within a bounded range of translations and rotations. We then linearly interpolate a robot trajectory between the augmented initial state and the original initial state, producing a smooth transition back to the demonstration trajectory. The interpolated segment is concatenated with the original trajectory to form an augmented sequence. Formally, the augmented training data can be written as

$$(\mathbf{s}_t^{\text{aug}}, \mathcal{P}_t) \longrightarrow \tilde{\mathcal{A}}_{t:t+H-1}, \quad \mathbf{s}_t^{\text{aug}} = \mathbf{T}_{\text{aug}}\, \mathbf{s}_t.$$

This augmentation exposes the policy to a diverse set of initial robot configurations while keeping the object points fixed, biasing learning toward object-centric reasoning. As a result, at inference time the policy generalizes to arbitrary initial hand poses without requiring explicit re-alignment.

Train Human Policy. We train a deterministic human policy using an action-chunking Transformer. At each timestep, the current hand and object states are represented in point space and embedded using an MLP-based point encoder. The resulting latent tokens are processed by a Transformer encoder, which outputs a fixed-length chunk of future actions conditioned on the current state. The policy is optimized with a regression objective over the action horizon:

$$\theta^\star = \arg\min_\theta \; \mathbb{E}_t \left[\left\| \pi_\theta(\mathbf{s}_t) - \mathbf{a}_{t:t+H-1} \right\|_2^2 \right].$$

This chunked prediction formulation captures short-horizon temporal structure while reducing compounding errors during rollout.

0.3 Experiments

We evaluate our framework on three tabletop manipulation tasks collected fully *in-the-wild* and compare against two baselines. Our experiments are designed to probe three practical questions: (i) can a policy trained purely from

untethered, egocentric demonstrations generalize to novel object poses and viewpoints at test time; (ii) is a compact *2D point* representation sufficient in this setting, or is explicit *3D* geometry necessary; and (iii) how much does our hand-to-robot retargeting pipeline contribute to downstream task success.

Task description. We consider three tasks spanning increasing contact complexity and dexterity requirements:

- **Pick-and-place bread.** The robot picks a bread piece from the table and places it onto a plate. While this task requires limited dexterity, it provides a clean test of spatial generalization under in-the-wild camera motion and varying object placements.
- **Rotate cap.** The robot rotates a small cap using repeated finger actions while coordinating wrist rotation. This task stresses sustained contact, cyclic motion, and precise orientation control, making it more sensitive to state representation and retargeting quality.
- **Slide card from box.** The robot slides a thin card out of a box. Because the card is slender and prone to sticking or slipping, success depends on fine-grained contact timing and stable fingertip-object interaction, serving as a challenging dexterous benchmark.

Table 1: Overall task success rates for our method.

Task	**Success rate (%)**
Pick-and-place bread	9/10
Rotate cap	6/10
Slide card from box	8/10

Baselines and ablations. To isolate the contribution of each design choice in our pipeline, we evaluate three baselines and ablations that progressively remove key components. *3D object keypoints (AruCo).* As an upper-bound reference, we attach ArUco markers to the manipulated objects and recover explicit 3D object keypoints via pose estimation. We then train the same point-embedding transformer policy using these 3D points as input. Comparing this setting against our default 2D point representation allows us to assess whether 2D keypoints, when paired with egocentric closed-loop feedback, provide sufficient spatial information relative to explicit 3D geometry. *IK retargeting.* To evaluate the role of our point-cloud-based hand retargeting, we disable the retargeting module and instead apply a naive inverse kinematics (IK) solver that directly maps human fingertip targets to robot joint configurations. This ablation examines how much precise contact preservation and finger-level consistency contribute to task success, relative to a straightforward kinematic mapping. *Open-loop policy.* Finally, we consider an open-loop variant of our policy that conditions on a single initial state and predicts the entire action chunk without intermediate feedback. This setting removes closed-loop correction during execution, enabling us to quantify the importance of feedback for recovery from small errors, handling distribution shift, and achieving higher overall success rates in in-the-wild scenarios.

0.4 Conclusion

We presented a framework for learning dexterous manipulation policies from egocentric human demonstrations by unifying both hand actions and object state into a compact point-space representation. Across three tabletop tasks collected *in-the-wild*, our closed-loop policy achieves strong success rates (Table 1), demonstrating that a lightweight 2D point representation can be sufficient for robust manipulation when paired with reactive feedback. In addition, our ablations highlight the contribution of individual design choices: the 2D point policy performs comparably to an ArUco-based 3D keypoint variant on pick-and-place (Table 2), closed-loop execution improves performance over an open-loop alternative on cap rotation (Table 3), and point-space hand retargeting substantially improves performance over a no-retargeting baseline on card sliding (Table 4).

Limitations and future work. Despite these results, several limitations remain. *Scalability* is constrained by the need to define object semantics (e.g., selecting keypoints) and by the per-task data collection required to cover diverse interaction modes; scaling to many objects and tasks will likely require more automated keypoint discovery, broader data coverage, and more efficient fine-tuning or pretraining. *Generalization* is evaluated only in a tabletop setting with a single robot platform; transferring to different cameras, hands, and workspaces may require additional calibration

Table 2: Comparison between 2D and 3D object points.

Task	**Success rate (ours)**	**Success rate (AruCo)**
Pick-and-place bread	9/10	9/10

Table 3: Comparison between closed-loop and open-loop control.

Task	**Success rate (ours)**	**Success rate (open-loop)**
Rotate cap	6/10	5/10

Table 4: Comparison between retargeting and no retargeting.

Task	**Success rate (ours)**	**Success rate (no retargeting)**
Slide card from box	8/10	2/10

procedures and more diverse training data. *Perception robustness* is limited by the reliability of 2D point tracking under severe occlusions, fast head motion, or poor lighting, which can degrade closed-loop feedback. Finally, *retargeting* is learned from geometric alignment in point space and does not explicitly optimize for contact forces or object-level outcomes; incorporating physics-aware constraints or tactile feedback is a promising direction.