

TouchBridge: Tactile-Conditioned Human-to-Robot Transfer for Dexterous Manipulation

Haozhe Zhang^{1*}, Gu Zhang^{1*†}, Zirui Zou^{2*}, Yuzheng Chen³, Jinlong Wu⁴,
Zixuan Hu⁵, Jianke Zhang¹, Yucheng Hu¹, Zhecheng Yuan¹, Huazhe Xu¹

¹ Tsinghua University ² Zhejiang University ³ Nanyang Technological University ⁴ Hunan University ⁵ Peking University

* Equal Contribution † Project Lead

Project Website: <https://touchbridge2026.github.io/>

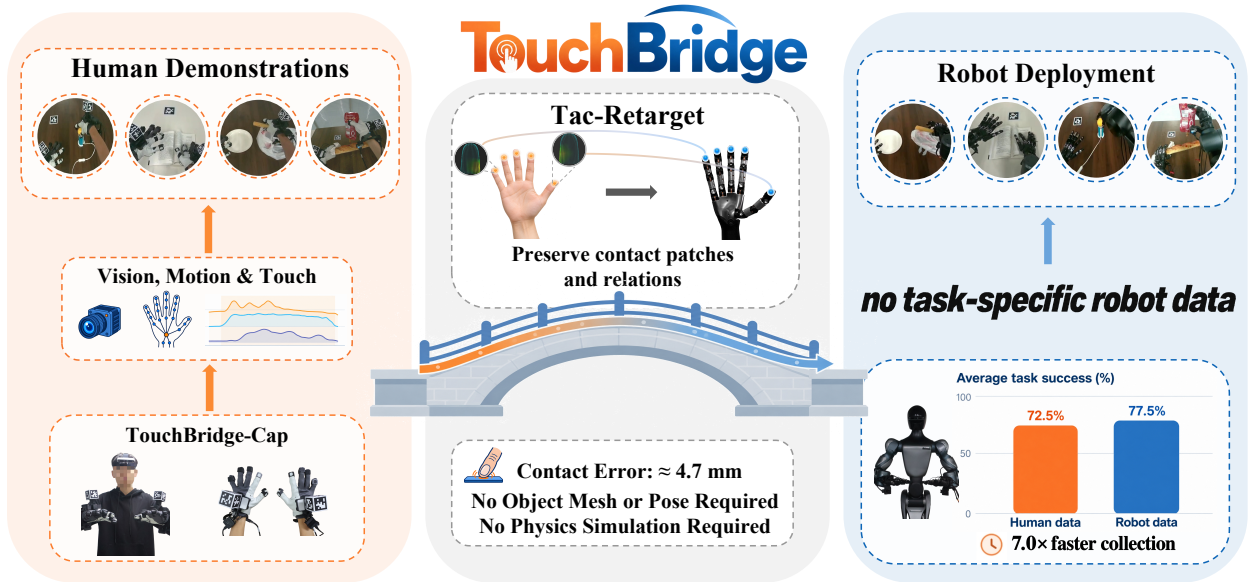


Fig. 1: We introduce TouchBridge, a tactile-conditioned human-to-robot transformation framework. TouchBridge-Cap records human demonstrations with sensing configurations aligned to the robot interface. Building on this correspondence, Tac-Retarget converts human demonstrations into robot-domain training data while preserving the demonstrated contact locations and spatial relations. Policies fine-tuned on these data achieve success rates close to those achieved by policies trained on teleoperated robot demonstrations, while human demonstrations can be collected substantially faster.

Abstract—Human demonstrations offer an efficient and scalable source of data for dexterous robot manipulation, but differences in sensing and hand morphology complicate the transfer of contact-rich interactions. We present TouchBridge, a hardware–software co-designed framework for human-to-robot data transformation. A wearable human demonstration interface and a corresponding robot interface establish cross-embodiment correspondence in visual and fingertip tactile sensing. We introduce Tac-Retarget, a tactile-conditioned re-targeting method that uses measured contact to align fingertip contact patches and preserve their spatial relations within and across hands. Unlike object-aware or simulation-based transfer pipelines, Tac-Retarget requires no object mesh, object-pose estimate, or physics simulation for re-targeting, enabling direct and scalable conversion of human demonstrations into robot-domain training data. Combined with visual synthesis and camera-anchored action conversion, TouchBridge produces robot-domain demonstrations for fine-tuning a pretrained vision-language-action (VLA) model. Across four real-world contact-rich tasks, policies fine-tuned solely on TouchBridge-converted human demonstrations achieve a 72.5% average success rate, compared with 77.5% using an equal number of

teleoperated robot demonstrations, while human demonstration collection is 7.0× faster. Under the same policy and training setup, Tac-Retarget outperforms DexPilot and UniDex by 25.0 and 42.5 percentage points, respectively.

I. INTRODUCTION

Building foundation models for contact-rich dexterous manipulation requires large-scale demonstrations that capture not only hand motion, but also physical interaction with the environment. Collecting such data directly on dexterous robots, however, is slow and demands substantial operator effort. Human demonstrations provide a natural and potentially scalable source of manipulation experience, and have increasingly been used for large-scale pretraining [1], [2] and direct human-to-robot transfer [3], [4]. Realizing this potential, however, requires transforming human experience into robot-executable training data both faithfully and efficiently.

This transformation faces two key challenges. First, *kinematic correspondence alone does not preserve physical interaction*. Conventional re-targeting methods typically match

hand poses, fingertip positions, or hand-centric vectors [1], [5], [6]. While effective for free-space motion, these objectives do not explicitly capture contact. As a result, fixed human-derived fingertip targets can displace a contacting finger from its support region, distort relations among simultaneous contacts, or break contact altogether. For contact-rich manipulation, kinematically plausible motion may therefore still be physically ineffective.

Second, *preserving physical interaction should not come at the cost of data scalability*. Object-aware approaches incorporate hand-object geometry or physics-based constraints [7]–[10], while other human-to-robot pipelines refine transferred trajectories or policies through simulation and reinforcement learning [11], [12]. Such methods can improve physical consistency, but require object meshes, object-pose estimation, environment reconstruction, or simulation-based rollout and refinement. These object- and task-specific requirements introduce substantial modeling and engineering overhead, making them difficult to scale with increasingly large and diverse human datasets.

Our key observation is that *physical contact itself provides a natural correspondence across embodiments*. Tactile sensing directly reveals which fingertips are in contact, where contact occurs, and how strongly different regions are loaded. Because these signals are measured directly during human demonstrations, they provide interaction constraints without requiring explicit object reconstruction or simulation-based refinement. Tactile sensing therefore addresses both challenges simultaneously: it augments kinematic retargeting with physical interaction information while preserving a direct human-to-robot data conversion pathway.

Building on this idea, we present **TouchBridge**, a hardware–software co-designed framework for scalable conversion of human demonstrations of contact-rich dexterous manipulation into robot-domain data. At the hardware level, we develop **TouchBridge-Cap**, a wearable capture interface that records RGB observations, hand motion, and fingertip tactile measurements. The corresponding robot interface approximately matches the wrist-camera-to-hand geometry, fingertip sensor placement, and tactile sensing layout of TouchBridge-Cap, establishing cross-embodiment correspondence in visual and tactile observations. At the software level, TouchBridge transforms these human demonstrations into robot-domain observations and actions for policy learning without task-specific robot demonstrations or object-specific environment modeling.

At the core of TouchBridge is **Tac-Retarget**, a tactile-conditioned retargeting method that uses measured contact to guide motion transfer across embodiments. Tac-Retarget combines *contact-patch alignment*, which aligns tactile contact regions between human and robot fingertips, with a *contact-activated graph*, which preserves spatial relations among simultaneous contacts within and across hands. Together, these objectives transfer not only fingertip motion but also the geometry and spatial structure of demonstrated contacts. The optimization operates directly on measured human kinematics and tactile responses, requiring no object

mesh, object-pose estimate, or physics simulation. Combined with robot-domain visual synthesis and camera-anchored action construction, the resulting pipeline converts raw human demonstrations into robot-domain training sequences.

We evaluate TouchBridge on four contact-rich real-world manipulation tasks: bulb screwing, sauce application, fragile-object grasping, and page flipping. Under the same policy and training setup, demonstrations converted using Tac-Retarget yield an average real-robot policy success rate of **72.5%**, outperforming DexPilot [5] and UniDex [1] by 25.0 and 42.5 percentage points, respectively, while also achieving lower contact error. Policies fine-tuned solely on TouchBridge-converted human demonstrations achieve a 72.5% average success rate, close to the 77.5% obtained using an equal number of teleoperated robot demonstrations, while human demonstrations are collected **7.0 $\times$** faster. These results show that TouchBridge retains much of the utility of robot-collected demonstrations while substantially reducing the cost of acquiring training data, providing a practical path toward scaling robot learning from human experience.

Our contributions are summarized as follows:

- We present **TouchBridge**, a hardware–software co-designed framework that transforms contact-rich human demonstrations into robot-domain training sequences. Through aligned visual and fingertip tactile sensing across human and robot interfaces, TouchBridge enables policy learning for contact-rich dexterous manipulation without task-specific teleoperated robot demonstrations.
- We introduce **Tac-Retarget**, a tactile-conditioned retargeting method that uses measured contact-patch alignment and a contact-activated graph to preserve physical interaction across embodiments. Compared with kinematic retargeting, Tac-Retarget substantially improves both contact preservation and downstream real-robot policy success.
- TouchBridge converts human demonstrations without object meshes, object-pose estimates, or simulation-based rollout refinement. Together, the **7.0 $\times$** faster demonstration collection and policy performance close to that obtained from robot-collected data make the framework well suited for scaling robot learning from human demonstrations.

II. RELATED WORK

A. Learning from Human Demonstrations

Human demonstrations provide a scalable alternative to collecting task data on physical robots. UniDex [1] and EgoScale [2] use large-scale human-derived data to pre-train transferable dexterous policies. These works primarily emphasize scaling human-derived data, with less emphasis on the fidelity of the converted demonstrations, particularly the preservation of contact interactions across embodiments. EgoMimic [13] and MotionTrans [14] co-train manipulation policies on human and robot demonstrations. Many other methods use simulation to ground human demonstrations in the robot embodiment. ManipTrans [11] refines retar-

TABLE I: **Comparison of human-to-robot transfer frameworks.** High-DoF denotes hands with at least 20 DoF. Robot-Free denotes fine-tuning without robot-collected demonstrations. ✗ indicates that the capability has not been experimentally validated.

Method	High-DoF Hands	Sim.-Free	No Object Pose/Geom.	Robot-Free Fine-Tuning
TouchBridge	✓	✓	✓	✓
AINA [4]	✗	✓	✗	✓
HumanEgo [3]	✗	✓	✗	✓
DexUMI [15]	✗	✓	✓	✓
ManipTrans [11]	✓	✗	✗	✓
TopoRetarget [9]	✓	✗	✗	✓
RoboPaint [10]	✗	✗	✗	✓
SPIDER [16]	✓	✗	✗	✓
C2Dex [17]	✗	✗	✗	✗
MotionTrans [14]	✗	✓	✓	✗
EgoMimic [13]	✗	✓	✓	✗

geted hand-object trajectories through residual reinforcement learning. SimToolReal [12] trains an object-centric policy in simulation and uses object trajectories from human videos as deployment goals. Both involve additional effort in environment construction and robot-object interaction modeling. A further line of work develops interfaces and transformation pipelines for learning directly from human demonstrations. DexUMI [15] combines wearable exoskeletons and visual conversion to bridge the human-robot embodiment gap, with the exoskeleton mechanically designed to match the target hand’s kinematics. Adapting to new robot hands generally requires hardware redesign, which is particularly challenging for high-DoF hands. Moreover, the wearable mechanism is typically tailored to a limited range of human hand sizes, reducing user adaptability and potentially compromising wearer comfort during demonstration collection. HumanEgo [3] enables task-level transfer without robot training data, but has only been demonstrated with parallel-jaw grippers. AINA [4] extends robot-data-free transfer to multi-fingered hands using visually estimated hand-object geometry and fingertip motion, without explicit tactile contact constraints. Our method instead uses measured tactile signals to preserve contact patches and their spatial relations during motion retargeting without simulation refinement, while using a glove-based capture interface that requires no mechanical correspondence with the target hand’s joint structure. This enables policies fine-tuned solely on converted human demonstrations to be deployed on the real robot without task-specific robot demonstrations.

B. Dexterous Retargeting

Dexterous retargeting maps human hand motion to robot hands with different kinematic structures. DexPilot [5] and AnyTeleop [6] optimize hand-centric task-space or keypoint-vector correspondences. UniDex [1] combines fingertip inverse kinematics with manual adjustment of the hand-base offset. These approaches are effective for pose transfer but do not condition the mapping on tactile measurements. FunGrasp [7] transfers object-relative functional contacts and uses simulation-trained control for execution. DexFlow [8] refines retargeted poses using contact maps inferred from hand-object geometry. TopoRetarget [9] preserves local in-

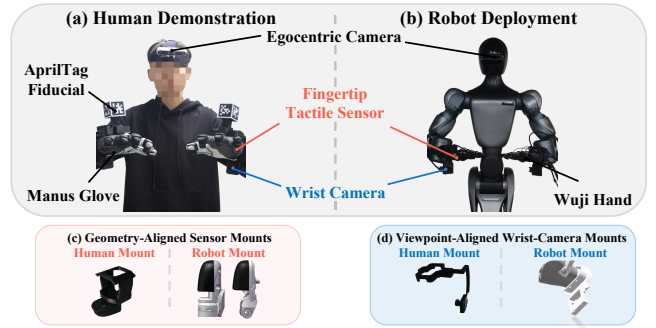


Fig. 2: **Human-Robot Hardware Co-design.** (a,b) Human demonstration and robot deployment interfaces. (c,d) Corresponding fingertip tactile-sensor placements and wrist-camera configurations reduce mismatch across embodiments.

teraction topology using an object mesh and pose trajectory. C2Dex [17] reconstructs stable object-side contacts from monocular video and further refines the trajectory in simulation. SPIDER [16] uses physics-based sampling and virtual contact guidance to generate dynamically feasible robot trajectories from human demonstrations. RoboPaint [10] uses force-weighted tactile correspondences within a Real-Sim-Real pipeline. Our method instead directly converts human demonstrations with visual and tactile observations into robot-domain trajectories while preserving measured contact, without relying on physical simulation, object meshes, or object-pose estimates.

As shown in Tab. I, compared with prior human-to-robot transfer frameworks, TouchBridge supports high-DoF dexterous hands (20 DoF) with simulation-free and object-model-free data conversion, enabling task learning directly from converted human demonstrations.

III. METHOD

A. Human-Robot Vision and Tactile System

We align the wrist-camera and fingertip tactile-sensor configurations across the human demonstration and robot deployment interfaces. As shown in Fig. 2, both interfaces use similarly mounted wrist cameras. Their fingertip tactile sensors have corresponding placements and matched sensing layouts. Differences in hand morphology and egocentric appearance are handled by the human-to-robot transformation pipeline described in Sec. III-C.

1) *Human Demonstration Interface:* We develop **TouchBridge-Cap**, a wearable human demonstration interface that records egocentric RGB observations using an Intel RealSense D455, together with bilateral wrist-view RGB observations from two Hikrobot cameras. A pair of Manus gloves captures articulated hand kinematics, while a PaXini Hall-effect tactile array mounted on each fingertip records spatially distributed contact responses. Multi-face AprilTag fiducials mounted near the wrists are localized offline in the egocentric video. Using the detected fiducial poses and calibrated fiducial-to-glove transformations, we transform the glove-estimated fingertip poses into the D455 camera frame.

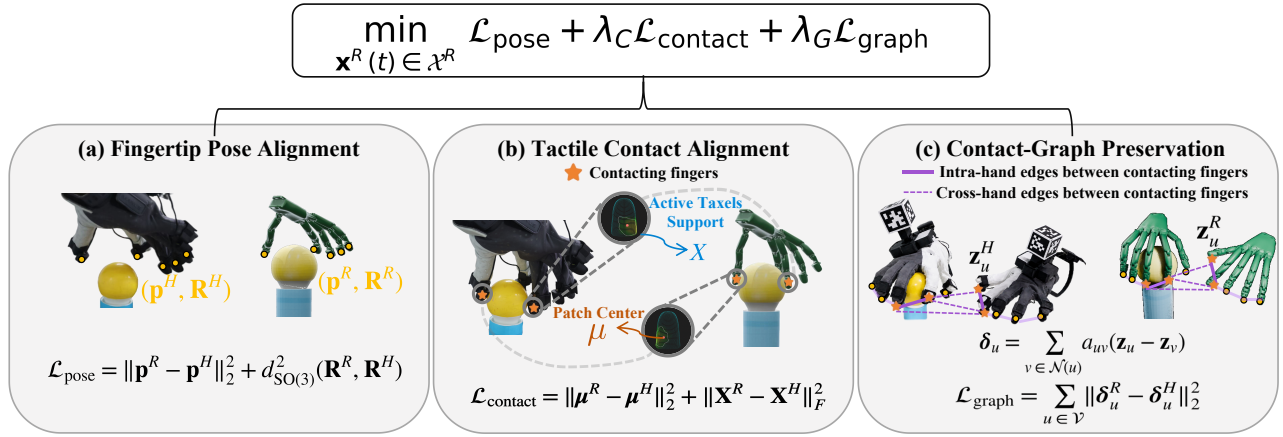


Fig. 3: **Tac-Retarget**. (a) Fingertip pose alignment tracks the demonstrated fingertip positions and orientations in the camera frame. (b) Contact alignment matches the patch center and active taxels between the human and robot fingertips. (c) A tactile-activated contact graph preserves the demonstrated spatial relations among active contacts within and across the two hands. Weighting terms and regularization are omitted for clarity.

2) *Robot Deployment Interface*: The robot is equipped with an egocentric camera, bilateral wrist cameras, and five PaXini tactile sensors on each dexterous hand. Custom mounts position the wrist cameras similarly to those on **TouchBridge-Cap** and place tactile sensors at corresponding fingertip locations. Together with the matched tactile sensing layouts, this design provides aligned wrist-view observations and fingertip-local tactile representations across the human and robot embodiments.

B. Tac-Retarget

1) *Problem Setup*: At time t , the human input consists of bimanual camera-frame fingertip poses $\mathbf{P}^H(t) = (\mathbf{p}^H(t), \mathbf{R}^H(t))$ and synchronized distributed tactile measurements $\mathbf{S}^H(t)$, where $\mathbf{p}^H(t)$ and $\mathbf{R}^H(t)$ collect the fingertip positions and orientations, respectively. The robot state is $\mathbf{x}^R(t) = (\mathbf{T}^R(t), \mathbf{q}^R(t))$, where $\mathbf{T}^R(t) \in \text{SE}(3)^2$ contains the two hand-base poses and $\mathbf{q}^R(t)$ stacks the joint configurations of both hands. Forward kinematics maps this state to the robot fingertip poses $\mathbf{P}^R(t) = (\mathbf{p}^R(t), \mathbf{R}^R(t))$.

We solve

$$\mathbf{x}^{R*}(t) = \arg \min_{\mathbf{x}^R(t) \in \mathcal{X}^R} \mathcal{L}_{\text{pose}} + \lambda_C \mathcal{L}_{\text{contact}} + \lambda_G \mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{reg}}, \quad (1)$$

where $\mathcal{X}^R$ is the feasible bimanual robot state space. The first three terms correspond to the fingertip pose alignment, tactile contact alignment, and contact graph shown in Fig. 3, while $\mathcal{L}_{\text{reg}}$ combines hand-base regularization, feasibility penalties and temporal regularization terms. No object mesh or pose estimate enters the optimization.

2) *Fingertip Pose Alignment*: We align the robot fingertip poses with the demonstrated human fingertip poses in the camera frame. We use the calibrated tactile-sensor frames to define fingertip poses for both motion tracking and contact alignment. The pose objective is

$$\mathcal{L}_{\text{pose}} = \sum_i w_i(t) \left[\|\mathbf{p}_i^R - \mathbf{p}_i^H\|_2^2 + \lambda_{\text{rot}} d_{\text{SO}(3)}^2(\mathbf{R}_i^R, \mathbf{R}_i^H) \right], \quad (2)$$

where i indexes the fingertips of both hands, $w_i(t)$ denotes the contact-conditioned tracking weight, and $d_{\text{SO}(3)}$ is the geodesic distance between rotations. The position and orientation terms establish global motion correspondence across embodiments. Both robot hand-base poses remain optimization variables rather than tracking targets derived from the measured human wrists.

3) *Tactile Contact Alignment*: For each fingertip, contact is detected from its total tactile response and number of active taxels. For a contacting fingertip i , let $\mathbf{r}_{im}$ denote the calibrated local position of taxel m and $s_{im}(t)$ its tactile response magnitude. The response-weighted contact center in the fingertip sensor frame is

$$\boldsymbol{\mu}_{i,\text{loc}}(t) = \frac{\sum_m s_{im}(t) \mathbf{r}_{im}}{\sum_m s_{im}(t) + \epsilon}. \quad (3)$$

With matched calibrated taxel layouts, we use the same human-derived local contact center for both human and robot fingertips. We transform this center into the camera frame using the measured human sensor pose and the robot sensor pose from forward kinematics, obtaining $\boldsymbol{\mu}_i^H$ and $\boldsymbol{\mu}_i^R$, respectively. We stack these camera-frame centers over contacting fingertips as $\boldsymbol{\mu}^H(t)$ and $\boldsymbol{\mu}^R(t)$. Similarly, $\mathbf{X}^H(t)$ and $\mathbf{X}^R(t)$ collect corresponding active-taxel positions transformed into the camera frame. The contact objective is

$$\mathcal{L}_{\text{contact}} = \|\boldsymbol{\mu}^R(t) - \boldsymbol{\mu}^H(t)\|_2^2 + \lambda_x \|\mathbf{X}^R(t) - \mathbf{X}^H(t)\|_{\Omega(t)}^2, \quad (4)$$

where $\Omega(t)$ contains normalized active-taxel response weights, assigning greater importance to taxels with stronger responses. The first term aligns the response-weighted patch centers, while the second preserves the spatial distribution of the active taxels.

4) *Contact Graph*: To preserve the spatial relations among simultaneous contacts, we construct corresponding human and robot contact graphs. We follow the distance-weighted Laplacian matching formulation of TopoRetarget [9] and adapt it to a tactile-activated graph. Within each hand, intra-hand edges connect pairs of contacting

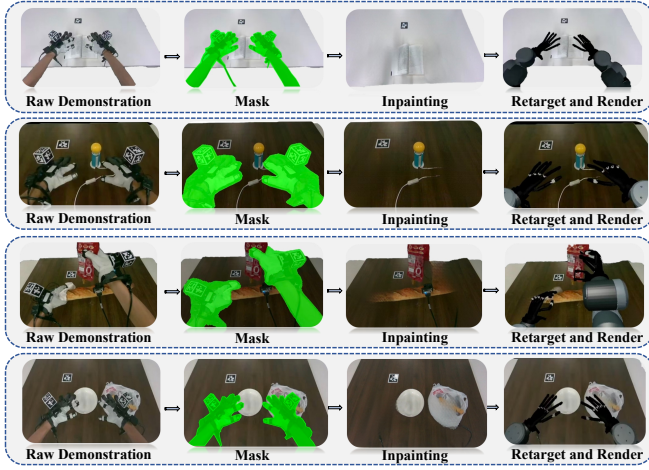


Fig. 4: **Human-to-robot data generation.** From left to right: raw human demonstrations, removal masks, inpainted scenes, and synthesized robot-domain observations.

fingertips, while cross-hand edges connect active contacts between the two hands. Tactile activation determines this contact-dependent topology, which is shared by the human and robot graphs.

Let $\mathbf{z}_u^H$ and $\mathbf{z}_u^R$ denote the corresponding human and robot positions of node u . The edge weights are computed and normalized as

$$\begin{aligned} \tilde{a}_{uv} &= \exp(-\kappa \|\mathbf{z}_u^H - \mathbf{z}_v^H\|_2) [1 + \gamma(\phi_u + \phi_v)], \\ a_{uv} &= \frac{\tilde{a}_{uv}}{\sum_{k \in \mathcal{N}(u)} \tilde{a}_{uk} + \epsilon}, \end{aligned} \quad (5)$$

where ϕ_u is the tactile response associated with node u . The graph topology and normalized weights are computed from the human demonstration and applied unchanged to the robot graph. For each node, we summarize its local spatial relations by the weighted relative-offset vector

$$\delta_u = \sum_{v \in \mathcal{N}(u)} a_{uv} (\mathbf{z}_u - \mathbf{z}_v), \mathcal{L}_{\text{graph}} = \sum_u \left\| \delta_u^R - \delta_u^H \right\|_2^2. \quad (6)$$

Contact determines which interaction edges are active, while tactile magnitude controls their relative importance. Matching the weighted relative offsets preserves the demonstrated spatial relations among active contacts within and across the two hands.

C. Human-to-Robot Data Transformation Pipeline

1) *Robot-Domain Observation Synthesis:* As shown in Fig. 4, we synthesize robot-domain visual observations from human demonstrations. Building on the hardware design described in Sec. III-A, we retain the wrist-view images while converting the egocentric view to the robot domain. SAM 2 [18] segments the human hands and wearable hardware, and ProPainter [19] inpaints the masked regions of the egocentric video. Our tactile-conditioned retargeting method (Sec. III-B) maps the camera-frame fingertip trajectories and tactile contact observations to robot hand-base poses and hand joint configurations. We then render the robot meshes

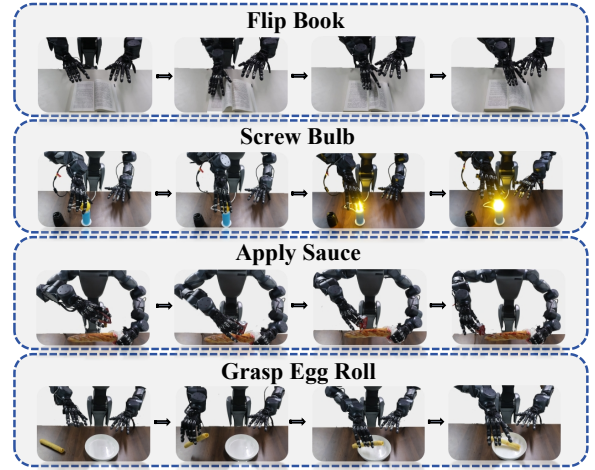


Fig. 5: Four contact-rich tasks evaluated in our experiments.

onto the inpainted egocentric frames using calibrated camera intrinsics and depth-aware occlusion handling.

2) *Camera-anchored Action Construction:* Following UMI [20], we represent arm actions as end-effector poses relative to the pose at the start of each action chunk. Hand actions use absolute joint targets obtained from retargeting.

We estimate the motion of the head-mounted camera using an AprilTag fixed in the workspace, separate from the wrist-mounted fiducials used for hand tracking. For a chunk anchored at observation time τ , we transform each retargeted end-effector pose from the camera frame C_t at time t to the anchor frame C_τ :

$${}^{C_\tau} \tilde{\mathbf{T}}_{E_b,t} = {}^{C_\tau} \mathbf{T}_A ({}^{C_t} \mathbf{T}_A)^{-1} {}^{C_t} \mathbf{T}_{E_b,t}, \quad (7)$$

where A denotes the stationary scene tag and $b \in \{L, R\}$ indexes the two arms. These aligned poses are then used to compute the relative arm actions.

IV. EXPERIMENTS

Overview: We evaluate TouchBridge through data analyses and real-robot experiments to address four questions:

- 1) How closely do our human-derived robot demonstrations resemble real-robot demonstrations in visual observations and actions?
- 2) Can policies fine-tuned solely on our human-derived demonstrations achieve task performance comparable to that of policies fine-tuned on an equal number of teleoperated robot demonstrations?
- 3) How effective is Tac-Retarget compared with existing simulation-free retargeting methods, and how do its tactile components contribute to its performance?
- 4) How does policy pretraining affect learning from our human-derived robot demonstrations?

A. Experimental Setup

1) *Robot Platform and Policy Execution:* All real-robot evaluations are conducted on a Moz1 bimanual robot equipped with two Wuji Hands. The visual and tactile sensing configuration follows the deployment interface described

in Sec. III-A.2. All policies use the same observation acquisition procedures, low-level controllers, and action-execution rate.

2) *Task Descriptions*: We evaluate all methods on four contact-rich tasks: Flip Book, Screw Bulb, Apply Sauce, and Grasp Egg Roll. These tasks involve diverse contact interactions, from precise fingertip manipulation to force-sensitive object handling. Representative rollouts are shown in Fig. 5.

Flip Book. The robot turns a single page of a book using the right index finger. The motion requires sustained fingertip contact throughout the page turn while avoiding displacement of the book.

Screw Bulb. The robot screws a light bulb into a socket until the bulb lights up. The task requires stable multi-finger contact and accurate rotational motion.

Apply Sauce. The robot holds a baguette with its left hand and squeezes ketchup onto it from left to right with its right hand. The task requires precise fingertip force control to produce the desired amount of ketchup.

Grasp Egg Roll. The robot picks up a fragile egg roll with the right hand and places it onto a plate. The robot must apply sufficient grasp force to prevent slippage without causing visible deformation or breakage.

3) *Data Collection*: We construct separate human-derived and teleoperated robot datasets for the four evaluation tasks. The human demonstrations are converted into robot training sequences using the pipeline in Sec. III-C. Tactile measurements are used only for retargeting, not as policy inputs.

For each task, we collect 60 human demonstrations and 60 teleoperated robot demonstrations. The teleoperated demonstrations serve as a reference for data-quality analysis and are used to train separate policies for the data-source comparison. They are not mixed into the training data of our method or the human-data baselines. Within each human demonstration, the operator is instructed to minimize head motion to reduce frame-to-frame viewpoint changes. Across demonstrations, we vary the camera position and viewing angle to increase the diversity of camera-object configurations.

4) *Evaluation Protocol*: **Policy Success Rate.** We evaluate each policy over 20 real-world trials per task. Success rate is the percentage of trials satisfying the task-specific success criterion. **Replay Success Rate.** We replay 20 human-derived converted demonstrations per task and report the percentage of replays satisfying the same task-specific success criteria used for policy evaluation. **Data Collection Time.** For each task and collection method, we report the average collection time per successful demonstration by dividing the total elapsed collection time by the 60 successful demonstrations.

B. Evaluation of Human-Derived Robot Demonstrations

We assess our human-derived robot demonstrations from two perspectives: visual similarity to real-robot observations and the ability of the converted actions to reproduce successful task execution.

TABLE II: Contact error of retargeting methods and tactile ablations.

Method	Contact error (mm)
Tac-Retarget (Ours)	4.667
DexPilot	6.354
UniDex	7.715
Tac-Retarget (w/o contact graph)	6.659
Tac-Retarget (w/o contact alignment)	7.071
Tac-Retarget (kinematic only)	7.540

1) *Visual Similarity*: We feed each multi-view observation, together with a shared task instruction, into the vision-language model (VLM) backbone of GR00T N1.7 [21]. We then extract the output of the final vision-language fusion layer before the action expert as the observation representation. As shown in Fig. 7(b), the projected features of human-derived robot observations substantially overlap with those of real-robot observations, whereas raw human observations exhibit greater separation from the real-robot data. This pattern suggests that the transformation reduces the visual gap. Fig. 7(a) provides a qualitative comparison of human-derived and real-robot observations in egocentric and wrist views.

2) *Action Replay*: We replay the actions from our human-derived robot demonstrations on the robot, keeping the initial scene configuration matched to that used during demonstration collection. Following the replay evaluation protocol in Sec. IV-A.4, the converted actions achieve an average replay success rate of 87.5% across the four tasks (Fig. 6(a)).

Together, these results suggest that our pipeline reduces the visual gap with real-robot data while retaining effective actions for task execution.

C. Policy Learning from Human-Derived vs. Robot Demonstrations

We evaluate whether policies fine-tuned on our human-derived demonstrations perform comparably to those fine-tuned on teleoperated robot demonstrations. For each task, we fine-tune GR00T N1.7 [21] on either 60 human-derived demonstrations or 60 teleoperated robot demonstrations, using identical training settings.

As shown in Fig. 6(b), policies trained on human-derived data achieve an average success rate of 72.5%, compared with 77.5% for those trained on teleoperated robot data. They achieve the same success rate on Flip Book and remain within ten percentage points on each of the other three tasks. These results show that policies fine-tuned solely on our human-derived demonstrations achieve task success rates close to those of policies fine-tuned on an equal number of teleoperated robot demonstrations.

Human demonstration collection is also substantially faster than robot teleoperation. As shown in Fig. 6(c), collecting a successful demonstration takes 11.7 seconds on average with **TouchBridge-Cap**, compared with 81.7 seconds through teleoperation, corresponding to a 7.0 $\times$ collection speedup. Thus, faster demonstration collection and comparable policy success rates suggest TouchBridge’s potential for scaling robot learning from human demonstrations.

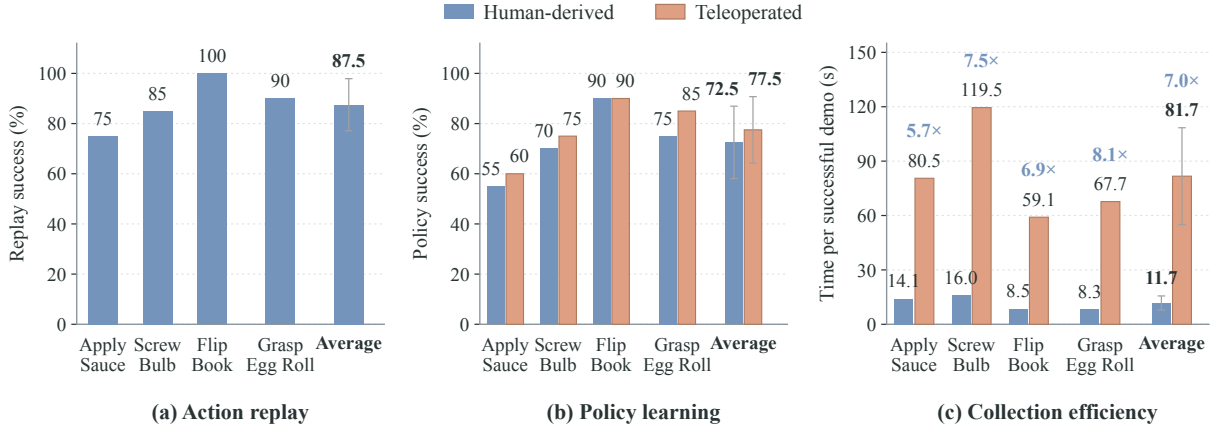


Fig. 6: (a) Action replay success rates across four tasks, with 20 converted human demonstrations replayed per task under matched initial scene conditions. (b) Real-robot success rates of GR00T N1.7 [21] policies fine-tuned on 60 demonstrations per task from each data source. (c) Average collection time per successful demonstration, including failed attempts and scene resets. Multipliers indicate collection speedups. Error bars show standard deviation over the four tasks.

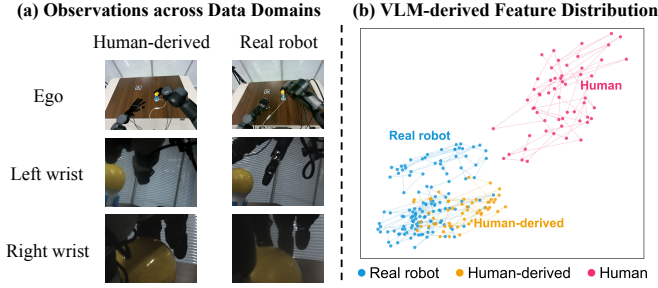


Fig. 7: Observation comparison across data domains. (a) Multi-view observations from real-robot and human-derived data. (b) VLM feature distributions before the action expert.

D. Evaluation of Tac-Retarget

We evaluate Tac-Retarget, its tactile ablations and baselines through contact preservation during trajectory replay in simulation and downstream policy performance on the robot.

1) *Contact Preservation in Simulation*: We use Grasp Egg Roll for this analysis because it provides a more tractable setting for reliable contact simulation. We load the accurately reconstructed object mesh into Isaac Sim [22] for evaluation.

We apply each retargeting method to the same 20 human demonstrations, replay the resulting trajectories in the simulation environment and report the contact error of each method in Tab. II. Contact error measures the Euclidean distance between the response-weighted human tactile contact center and the simulated contact center in corresponding fingertip-local frames.

As shown in Tab. II, Tac-Retarget achieves the lowest mean contact error of 4.667 mm, outperforming DexPilot [5] and UniDex [1]. Removing either contact alignment or the contact graph increases the error, with the kinematic-only variant performing worst among the ablations. These results show that both components improve contact-location preservation. Fig. 8 illustrates common retargeting failure modes across four tasks, including missing contacts, misplaced contacts, and object penetration. Tac-Retarget preserves the intended fingertip contacts without obvious object penetration in the examples shown.

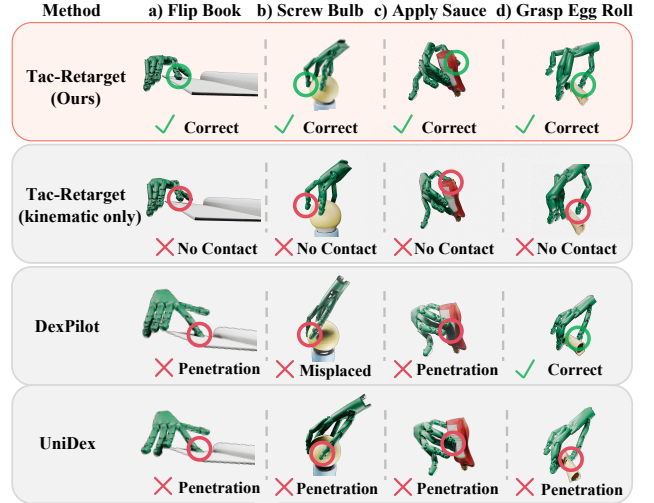


Fig. 8: Qualitative comparison of Tac-Retarget, its kinematic-only variant, and retargeting baselines.

2) *Downstream Policy Performance*: For each retargeting method, we convert the same 60 human demonstrations per task and fine-tune GR00T N1.7 [21] on the resulting data under identical training settings. Evaluation follows Sec. IV-A.4. As shown in Fig. 9, Tac-Retarget performs best on all four tasks, averaging 72.5% success and exceeding DexPilot [5] and UniDex [1] by 25.0 and 42.5 percentage points, respectively. Removing the contact graph or contact alignment reduces average success, with the largest drop when both are removed, supporting the contribution of both tactile components to policy learning.

The simulation and real-robot policy results show that Tac-Retarget improves contact preservation and supports more successful policy learning for contact-rich manipulation.

E. Effect of Policy Pretraining

We examine how policy pretraining affects learning from our human-derived robot demonstrations using the same 60 converted demonstrations per task. As shown in Fig. 10, pre-trained GR00T N1.7 [21] and $\pi_{0.5}$ [23] achieve average suc-

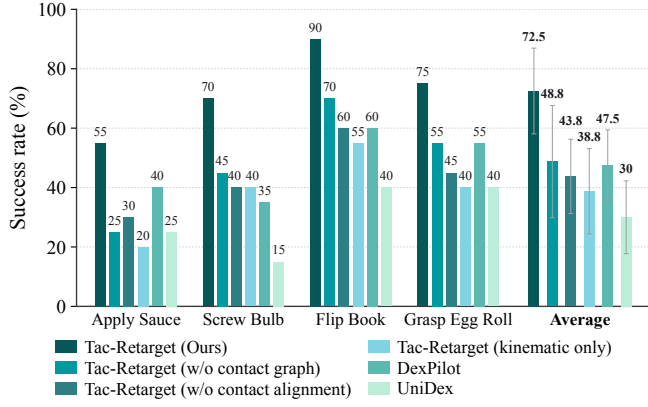


Fig. 9: Real-robot policy success rates comparing Tac-Retarget, its tactile ablations and existing retargeting methods. Error bars show standard deviation over the four tasks.

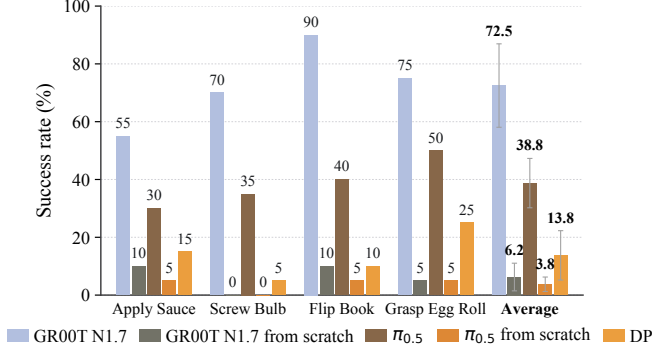


Fig. 10: Real-robot policy success rates under different policy and pretraining settings. Error bars show standard deviation over the four tasks.

cess rates of 72.5% and 38.8%, respectively, compared with 6.2% and 3.8% for their respective from-scratch variants. Diffusion Policy [24], included as an additional baseline, achieves 13.8%. All policies are evaluated following Sec. IV-A.4, without task-specific robot demonstrations for training. These results demonstrate the benefit of pretraining when learning from limited human-derived demonstrations, with pretrained GR00T N1.7 [21] achieving the best performance among the evaluated policies.

V. CONCLUSION

We presented TouchBridge, a hardware–software co-designed framework for learning contact-rich dexterous manipulation without task-specific robot demonstrations. It combines TouchBridge-Cap for efficient human demonstration collection with Tac-Retarget for contact-preserving motion transfer, producing robot-domain data without object meshes, object-pose estimates, or simulation-based refinement. With faster demonstration collection and policy success rates close to those obtained from robot-collected data, we believe TouchBridge can serve as a practical framework for the community to scale robot learning from human demonstrations. Our current study uses a modest demonstration set; extending the framework to larger and more diverse human datasets remains an important direction for future work.

TABLE III: Collecting rates of the human demonstration streams.

Data stream	Collecting rate
Egocentric RGB	30 Hz
Left/right wrist-view RGB	30 Hz
Bilateral Manus hand poses	130 Hz
Fingertip tactile stream	90 Hz

APPENDIX

A. Multimodal Data Collection and Temporal Alignment

1) *Asynchronous Multimodal Collection*: Human demonstrations are recorded using independent acquisition threads for egocentric RGB-D observations, bilateral wrist-view images, articulated hand poses, and bilateral fingertip tactile measurements. All samples are timestamped using a shared host clock. Table III summarizes the collecting sampling rates of the multimodal streams.

2) *Camera-Frame Fingertip Reconstruction*: As shown in Fig. 11, we first estimate the AprilTag fiducial-cube pose ${}^C\mathbf{T}_Q$ in the egocentric camera frame. Given the Manus fingertip pose ${}^W\mathbf{T}_{F_i}$ relative to the wrist, its camera-frame pose is reconstructed as

$${}^C\mathbf{T}_{F_i} = {}^C\mathbf{T}_Q {}^Q\mathbf{T}_W {}^W\mathbf{T}_{F_i}, \quad (8)$$

where C , Q , W , and F_i denote the camera, fiducial-cube, wrist, and i -th fingertip frames, respectively.

Because the tactile sensor is mounted with a fixed offset from the Manus fingertip frame, we further compute its pose as

$${}^C\mathbf{T}_{S_i} = {}^C\mathbf{T}_{F_i} {}^{F_i}\mathbf{T}_{S_i}, \quad (9)$$

where S_i denotes the corresponding tactile-sensor frame. The resulting sensor origin and orientation are used as the human fingertip target in Tac-Retarget, such that the kinematic target is expressed in the same coordinate frame as the tactile taxel layout.

For the hardware configuration shown in Fig. 11, the calibrated wrist-to-cube transform is

$${}^Q\mathbf{T}_W = \begin{bmatrix} 1 & 0 & 0 & 0.000 \\ 0 & -1 & 0 & -0.065 \\ 0 & 0 & -1 & -0.015 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

The sensor-to-fingertip mounting transform, shared by all fingertips, is

$${}^{F_i}\mathbf{T}_{S_i} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -0.275637 & 0.961262 & 0.02477 \\ 0 & 0.961262 & 0.275637 & 0.00186 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

All translation entries are expressed in meters.

B. Implementation Details of Tac-Retarget

This section supplements Sec. III-B with tactile preprocessing, contact-dependent weighting, graph construction, and compliance compensation.

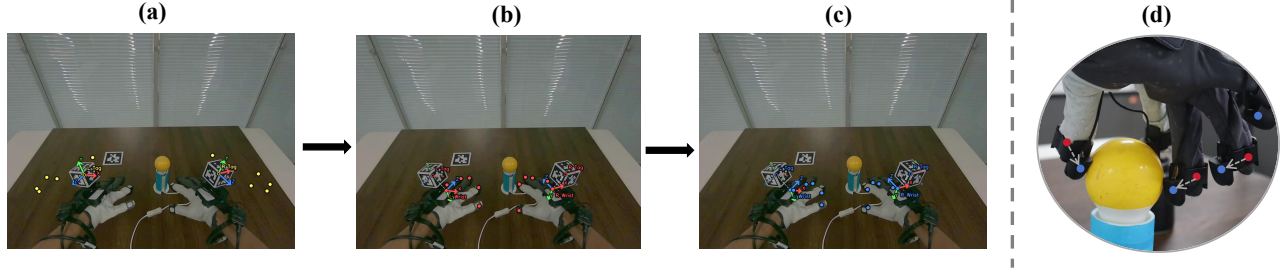


Fig. 11: **Reconstruction of camera-frame tactile fingertip poses.** (a) Wrist-mounted AprilTag fiducial frames detected in the egocentric image. (b) Manus fingertip positions (red) reconstructed using the calibrated wrist-to-cube transforms. (c) Tactile-sensor origins (blue) obtained by applying the fixed sensor-to-fingertip mounting transform. (d) Close-up of the fixed spatial offsets between the Manus fingertips (red) and tactile-sensor origins (blue).

1) *Tactile Preprocessing and Contact Activation:* We align the tactile measurements with the visual timestamps by linearly interpolating the three-axis taxel readings. For taxel m on fingertip i , its response magnitude is $s_{im} = \|\mathbf{f}_{im}\|_2$, and the total fingertip response is $F_i = \sum_m s_{im}$. The active-taxel set is $\mathcal{A}_i = \{m : s_{im} > \tau_s\}$. A contact label $c_i \in \{0, 1\}$ is set to one when $F_i > \tau_F$ and $|\mathcal{A}_i| \geq n_{\min}$. The thresholds are expressed in the input tactile-response scale.

The contact center in Eq. (3) uses responses from all taxels, whereas the taxel-cloud term uses only $\mathcal{A}_i$. For each active taxel, its weight is

$$\omega_{im} = \frac{1}{|\mathcal{A}_i|} \frac{s_{im}}{\max(\max_{k \in \mathcal{A}_i} s_{ik}, \epsilon)}. \quad (12)$$

This weighting emphasizes stronger responses while averaging the contribution over the active taxels. We use the same calibrated local taxel layout for the human and robot hands. The contact center, represented in this common local sensor frame, is transformed into the camera frame using the corresponding human or robot sensor pose.

2) *Contact-Conditioned Weighting and Graph Construction:* Let $c_i \in \{0, 1\}$ indicate whether the i -th fingertip is in contact. We use the contact-conditioned pose-tracking weight for fingertips:

$$w_i = \lambda_p(\beta_0 + \beta_c c_i) \eta^{c_i}, \quad (13)$$

where η reduces pose tracking during detected contact. The tactile patch, taxel-cloud, and compliance objectives are activated upon contact and weighted by their respective global coefficients.

The rotation coefficient is configured independently as λ_R , corresponding to $\lambda_{\text{rot}} = \lambda_R / \lambda_p$ in Eq. (2). With $\beta_0 = 1$, $\beta_c = 2$, and $\eta = 0.05$, the pose-tracking weight during contact is 0.15 times its non-contact value. This reduction allows the tactile objectives to influence the solution while preserving fingertip motion tracking.

The response ϕ_u in Eq. (5) is the associated fingertip total response for fingertip and contact-center nodes. Human-derived edge weights are reused for the robot graph. In addition to Laplacian matching, the implementation penalizes discrepancies in Euclidean distances between all pairs of

contacting fingertips, with an additional coefficient for cross-hand pairs. These auxiliary terms preserve contact spacing within and across hands.

3) *Response-Dependent Compliance Compensation:* Contact-patch alignment preserves the demonstrated contact geometry, but does not explicitly represent local deformation during contact. We therefore include a response-dependent geometric compensation term that shifts the target contact center. For an active contact, the offset magnitude is

$$d_i = \min\left(d_{\max}, \frac{\max(0, F_i - F_{\min})}{k}\right), \quad (14)$$

where k controls the response-to-displacement conversion and $d_{\max}$ bounds the target offset.

Using the human sensor origin $\mathbf{p}_i^H$ and contact center $\boldsymbol{\mu}_i^H$, we define

$$\mathbf{u}_i = \frac{\boldsymbol{\mu}_i^H - \mathbf{p}_i^H}{\|\boldsymbol{\mu}_i^H - \mathbf{p}_i^H\|_2}, \quad (15)$$

$$\boldsymbol{\mu}_i^{H, \text{comp}} = \boldsymbol{\mu}_i^H + d_i \mathbf{u}_i,$$

where $\mathbf{u}_i$ points from the sensor toward the contacted object, so that the compliance target lies slightly beyond the measured contact surface. Intuitively, once contact is established, the fingertip is encouraged to press slightly deeper, as if the compliant fingertip pad were compressed under load. The displacement increases with the tactile response up to a prescribed maximum depth, encouraging a firmer and more stable grasp rather than merely touching the object surface.

The implementation augments the objective in Eq. (1) with

$$\lambda_{\text{comp}} \mathcal{L}_{\text{comp}} = \lambda_{\text{comp}} \sum_{i: c_i=1} \left\| \boldsymbol{\mu}_i^R - \boldsymbol{\mu}_i^{H, \text{comp}} \right\|_2^2. \quad (16)$$

4) *Hyperparameters:* Table IV summarizes the selected parameters that directly correspond to the objectives described above. The table reports separate absolute weights for patch-center and active-taxel alignment. In the factorized notation of Eq. (4), these correspond to $\lambda_C = \lambda_{\text{patch}}$ and $\lambda_x = \lambda_{\text{taxel}} / \lambda_{\text{patch}}$.

ACKNOWLEDGMENT

This work is supported by Tsinghua University - Keystone Electrical (Zhejiang) Co., Ltd Joint Research Center for Embodied Multimodal Artificial Intelligence (JCEMAI).

TABLE IV: Selected hyperparameters for Tac-Retarget.

Parameter	Value	Description
τ_F	0.02	Contact threshold on the total fingertip response.
τ_s	0.01	Response threshold for selecting active taxels.
λ_p	100	Camera-frame fingertip-position tracking weight.
λ_R	1	Fingertip-orientation tracking weight.
(β_0, β_c)	(1, 2)	Baseline tracking multiplier and contact-dependent increment.
η	0.05	Pose-tracking scale applied during detected contact.
λ_{patch}	20	Contact-patch center alignment weight.
λ_{taxel}	4	Active-taxel cloud alignment weight.
λ_{comp}	16	Response-dependent compliance-target weight.
k	20	Response-to-displacement conversion factor.
d_{max}	0.01 m	Maximum inward compliance displacement.
λ_G	1	Tactile contact-graph Laplacian weight.
κ	20 m ⁻¹	Distance-decay factor for graph-edge weights.
γ	1	Tactile-response modulation of graph-edge weights.
$(\lambda_{\text{cross}}, \lambda_{\text{pair}})$	(0.4, 0.25)	Cross-hand and pairwise contact-distance weights.
$\lambda_{q, \text{smooth}}$	0.45	First-order temporal smoothness weight for joint angles.
$(\lambda_{b, p}, \lambda_{b, R})$	(10, 5)	First-order smoothness weights for base translation and rotation.
λ_{acc}	0.08	Second-order smoothness weight for bases and joint angles.

REFERENCES

- [1] G. Zhang *et al.*, “UniDex: A robot foundation suite for universal dexterous hand control from egocentric human videos,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026, pp. 1841–1852.
- [2] R. Zheng *et al.*, “EgoScale: Scaling dexterous manipulation with diverse egocentric human data,” *arXiv preprint arXiv:2602.16710*, 2026.
- [3] Z. Wang, B. He, K. Yu, S. Lee, R. Gao, F. Huang, and Y. Aloimonos, “HumanEgo: Zero-shot robot learning from minutes of human egocentric videos,” *arXiv preprint arXiv:2605.24934*, 2026.
- [4] I. Guzey, H. Qi, J. Urain, C. Wang, J. Yin, K. Bodduluri, M. Lambeta, L. Pinto, A. Rai, J. Malik, T. Wu, A. Sharma, and H. Bharadhwaj, “Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2026.
- [5] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. D. Ratliff, and D. Fox, “DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 9164–9170.
- [6] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, “AnyTeleop: A general vision-based dexterous robot arm-hand teleoperation system,” in *Proc. Robot.: Sci. Syst. (RSS)*, 2023, doi: 10.15607/RSS.2023.XIX.015.
- [7] L. Huang, H. Zhang, Z. Wu, S. Christen, and J. Song, “Fun-Grasp: Functional grasping for diverse dexterous hands,” *IEEE Robot. Autom. Lett.*, vol. 10, no. 6, pp. 6175–6182, Jun. 2025, doi: 10.1109/LRA.2025.3561573.
- [8] X. Lin, K. Yao, L. Xu, X. Wang, X. Li, Y. Wang, and M. Li, “DexFlow: A unified approach for dexterous hand pose retargeting and interaction,” *arXiv preprint arXiv:2505.01083*, 2025.
- [9] J. Wu, S. Yao, G. He, X. Liu, Z. Zeng, X. Jiang, H. Yang, W. Zhang, and H. Zhao, “TopoRetarget: Interaction-preserving retargeting for dexterous manipulation,” *arXiv preprint arXiv:2606.16272*, 2026.
- [10] J. Fan, Z. Zhao, Y. Zhang, C. Chen, P. Wang, H. Zhang, and Z. Cheng, “RoboPaint: From human demonstration to any robot and any view,” *arXiv preprint arXiv:2602.05325*, 2026.
- [11] K. Li, P. Li, T. Liu, Y. Li, and S. Huang, “ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025, pp. 6991–7003.
- [12] K. Kedia, T. G. W. Lum, J. Bohg, and C. K. Liu, “SimToolReal: An object-centric policy for zero-shot dexterous tool manipulation,” *arXiv preprint arXiv:2602.16863*, 2026.
- [13] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu, “EgoMimic: Scaling imitation learning via egocentric video,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025, pp. 13226–13233.
- [14] C. Yuan, R. Zhou, M. Liu, Y. Hu, S. Wang, L. Yi, C. Wen, S. Zhang, and Y. Gao, “MotionTrans: Human VR data enable motion-level learning for robotic manipulation policies,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2026.
- [15] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song, “DexUMI: Using human hand as the universal manipulation interface for dexterous manipulation,” in *Proc. Conf. Robot Learn. (CoRL)*, 2025, pp. 437–459.
- [16] C. Pan, C. Wang, H. Qi, J. Urain, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan, “SPIDER: Scalable physics-informed dexterous retargeting,” *arXiv preprint arXiv:2511.09484*, 2025.
- [17] J. Ren, Z. Jiang, Y. Yang, H. Jia, H. Jiang, B. Li, Y. Yao, C. Lin, Q. Shen, Z. Bing, X.-X. Long, and X. Cao, “C2Dex: Contact-consistent reconstruction and retargeting for dexterous manipulation from monocular video,” *arXiv preprint arXiv:2608.07045*, 2026.
- [18] N. Ravi *et al.*, “SAM 2: Segment Anything in Images and Videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [19] S. Zhou, C. Li, K. C. K. Chan, and C. C. Loy, “ProPainter: Improving propagation and transformer for video inpainting,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 10477–10486.
- [20] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, “Universal Manipulation Interface: In-the-wild robot teaching without in-the-wild robots,” in *Proc. Robot.: Sci. Syst. (RSS)*, 2024.
- [21] NVIDIA *et al.*, “GR00T N1: An open foundation model for generalist humanoid robots,” *arXiv preprint arXiv:2503.14734*, 2025.
- [22] S. Gao, M. Pagnucco, T. Bednarz, and Y. Song, “NVIDIA Isaac Sim: Enabling scalable, GPU-accelerated simulation for robotics,” *arXiv preprint arXiv:2606.03551*, 2026.
- [23] K. Black *et al.*, “ $\pi_{0.5}$: A vision-language-action model with open-world generalization,” in *Proc. Conf. Robot Learn. (CoRL)*, 2025, pp. 17–40.
- [24] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion Policy: Visuomotor policy learning via action diffusion,” in *Proc. Robot.: Sci. Syst. (RSS)*, 2023, doi: 10.15607/RSS.2023.XIX.026.