REFORCE: Learning Force-aware Retargeting for Dexterous Manipulation

Yuhang Wu, Lingqi Zeng, Changwei Jing, Jianglong Ye†, Xiaolong Wang†
UC San Diego
https://wuyuhang-eai.github.io/reforce/
† Equal advising.

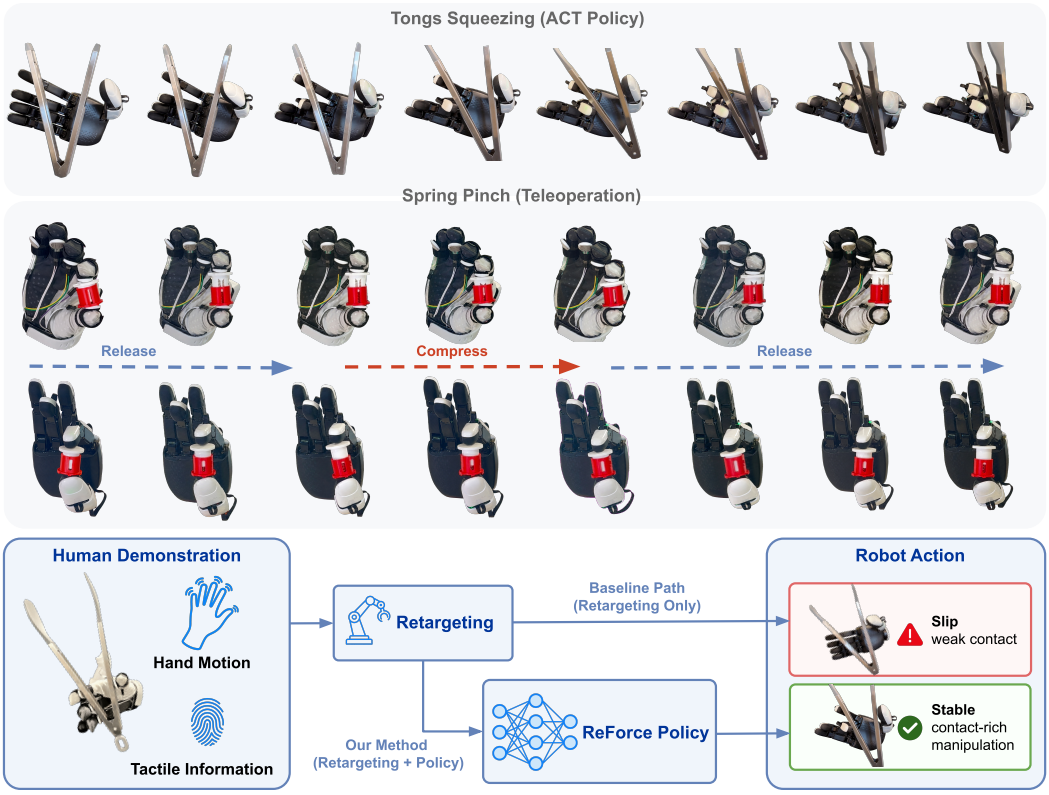


Figure 1: **Force-aware retargeting overview.** Human-guided motion and contact references specify the intended interaction, while REFORCE uses robot-side fingertip force feedback to adapt the reference into dexterous-hand commands during contact-rich execution.

Abstract: Human demonstrations offer a scalable data source for dexterous manipulation, but transferring them to robot actions remains challenging due to the embodiment gap. Today's retargeting is mostly kinematic, yet manipulation is decided by *force*, which governs how the hand interacts with the object and how the object moves. In this paper, we present REFORCE, a **Force**-aware **Re**targeting method that turns human motion and forces into robot actions that reproduce the intended contact. REFORCE predicts a residual on the kinematically retargeted action to reach the desired force, using a general force tracker trained on large-scale simulation interactions. It supports both online force-aware teleoperation and offline data translation. In simulation and on real hardware, REFORCE achieves lower force-tracking error and stronger multi-finger contact engagement on contact-rich tasks such as paper-cup grasping and tongs manipulation.

Keywords: Dexterous Manipulation, Dexterous Hand Retargeting, Imitation Learning, Tactile Feedback, Learning from Human

1 Introduction

Human data is a rich source for learning dexterous manipulation [1, 2, 3, 4, 5, 6, 7, 8], but transferring it to robot actions suffers from the embodiment gap. How to better retarget human demonstrations to deployable robot actions remains an open question. Most of today's retargeting algorithms [9, 10] are kinematic only: an optimizer aligns human joint positions to their robot counterparts. Such objectives ignore geometry and contact dynamics, so their output is a coarse motion match; in practice it serves only to pre-train or co-train policies that still rely on robot data [3, 8, 4], or runs with a human teleoperator closing the gap. Manipulation, however, is decided by *force*: force determines how the hand interacts with the object, and whether a grasp holds, crushes, or slips [11, 12, 13, 14].

Recent methods [6, 15, 16, 17] therefore make retargeting force-aware: given a tracked hand-object demonstration, they adopt reinforcement learning or sampling in simulation to search for robot trajectories that reproduce its contact forces. These methods share two limitations: (i) each run needs a digital twin, a simulation-ready object model with accurate hand and object pose, unavailable for an arbitrary object; and (ii) each trajectory is optimized against a recorded dataset, so they cannot run during online teleoperation, which is still the main source of high-quality data.

We present REFORCE, a **Force**-aware **Re**targeting method that turns human demonstrations into robot actions which reproduce the intended contact. Given motion and forces from a human demonstrator, REFORCE produces robot actions that reproduce the intended contact, and transfers zero-shot to force-sensitive tasks such as grasping a paper cup or using tongs. To build it, we collect large-scale hand-object trajectories in simulation, recording both joint positions and contact forces, and train a closed-loop force tracker on them. The tracker predicts a residual on top of the kinematically retargeted action so that the robot reaches the desired force, in both online teleoperation and offline data translation.

We evaluate REFORCE in both simulation and the real world, on force-sensitive tasks such as paper-cup grasping and tongs manipulation, under online teleoperation and offline translation settings. It achieves lower force-tracking error, while reducing severe missing-contact failures.

In summary, our contributions are threefold:

- We propose general force tracking, a retargeting paradigm that adapts kinematic retargeting using online force feedback and requires no digital twin.
- We collect a large-scale simulated dataset of hand-object trajectories with paired joint positions and contact forces, and train a single force tracker on it.
- We evaluate in simulation and on real hardware, showing improved force tracking under both online teleoperation and offline references.

2 Related Work

Learning from human demonstration. Human demonstrations provide task intent and temporal structure for robot manipulation, and recent imitation-learning methods and robot datasets have made them effective sources for trajectory generation [18, 19, 20, 21, 22, 23, 24, 25, 26]. For contact-rich manipulation, force-centered imitation work further shows that demonstrations encode contact skills, not only motion [11, 27, 28, 29, 30]. REFORCE builds on this view by adapting human-guided trajectories when robot-side fingertip force disagrees with the intended contact.

Dexterous retargeting. Dexterous hand retargeting seeks to transfer human hand motion to robot hands despite differences in morphology, actuation, sensing, and compliance. Teleoperation, human-video, mixed-reality, and portable motion-capture systems improve demonstration collection, while reinforcement learning, residual refinement, and human-like hardware reduce embodiment mismatch [9, 31, 32, 33, 10, 34, 35, 36, 37, 38, 39, 30]. These methods primarily obtain, map, or refine human motion; REFORCE instead studies how a robot hand should adapt a human-guided motion and force reference under real-time contact feedback.

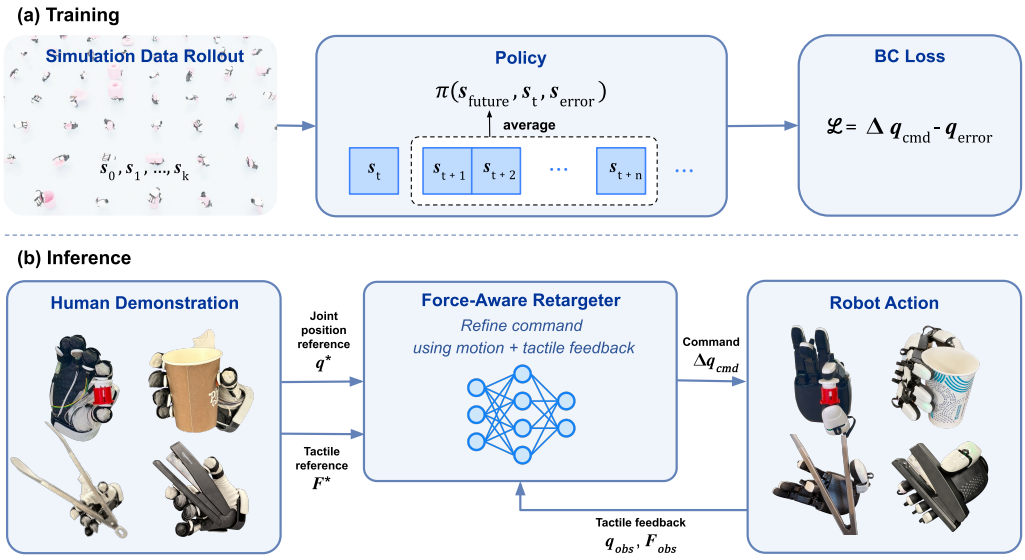


Figure 2: **ReForce training and inference pipeline.** During training, randomized and augmented simulation interactions provide joint and force trajectories from which future-window motion and contact targets are constructed. During deployment, REFORCE adapts references using online fingertip force feedback.

Force and tactile feedback for contact-rich manipulation. Hybrid position/force and impedance control provide reactive contact regulation, but require task-specific gains and hand-designed force-to-motion mappings [40, 41, 42]. Learning-based tactile manipulation and recent visual-tactile policies show that contact feedback improves grasp adjustment, in-hand dexterity, and contact-rich policies when vision and proprioception are insufficient [43, 44, 45, 46, 47, 48, 49, 50, 51, 13]. REFORCE keeps this fast local feedback role, but learns a retargeting controller that can be composed with multiple human-guided trajectory sources.

3 Method

3.1 Problem Formulation

The REFORCE policy is designed as a closed-loop force-aware retargeting controller that adapts an upstream motion-and-contact reference using the current hand state and fingertip force feedback. At control step t, let $q_t^{\text{obs}} \in \mathbb{R}^{D_q}$ denote the observed hand joint configuration and $F_t \in \mathbb{R}^{D_F}$ the measured fingertip normal forces, where D_q is degrees of freedom and $D_F = 5$ corresponds to the thumb, index, middle, ring, and pinky. The measured state, upstream target, and target–measurement error are

$$s_t = \left(q_t^{\text{obs}}, F_t\right), \qquad s_t^\star = \left(q_t^\star, F_t^\star\right), \qquad e_t = \left(q_t^\star - q_t^{\text{obs}}, F_t^\star - F_t\right). \tag{1}$$

Here, $s_t^\star$ is the reference motion-and-contact state supplied by an upstream reference source, where $q_t^\star$ denotes the target joint configuration and $F_t^\star$ denotes the target contact force. Before being passed to the network, all input quantities are standardized. The standardization procedure is described in Appendix A.3. The REFORCE policy, parameterized by θ, predicts an bounded joint-command update.

$$\Delta q_t^{\text{cmd}} = \pi_\theta\left(s_t, s_t^\star, e_t\right). \tag{2}$$

Let q_t^{cmd} denote the accumulated joint command at control step t. The bounded update is accumulated and clipped to the hardware joint limits:

$$q_{t+1}^{\text{cmd}} = \text{clip}\left(q_t^{\text{cmd}} + \Delta q_t^{\text{cmd}}, q_{\min}, q_{\max}\right). \tag{3}$$

Here, $q_{\min}, q_{\max} \in \mathbb{R}^{D_q}$ are the elementwise lower and upper joint limits, respectively.

3.2 Force-Aware Policy Learning and Execution

Post-hoc motion-and-contact targets. The policy is trained from simulated hand–object interaction trajectories $\{(q_t, F_t)\}_{t=1}^{T}$ collected under randomized hand configurations, motion trajectories, and object contact properties. The post-hoc motion and force targets are

$$q_t^\star = \frac{1}{K_t}\sum_{k=1}^{K_t} q_{t+k}, \qquad F_t^\star = \frac{1}{K_t}\sum_{k=1}^{K_t} F_{t+k}, \tag{4}$$

where K_t is the number of available future steps in the target windows. These targets summarize a nearby motion-and-contact state reached later in the rollout. The supervised action is instead the demonstrated one-step joint update, $\Delta q_t^{\mathrm{demo}} = q_{t+1} - q_t$, so the post-hoc target provides future motion-and-contact context rather than serving directly as the action label.

Behavior-cloning objective. Let $i \in \{1, \ldots, 5\}$ index the finger groups and let D_i be the number of joints associated with finger i. Let $\mathcal{D}$ denote the training-sample distribution. REFORCE is trained using mean-squared error, first averaged over the joints within each finger and then averaged equally across fingers:

$$\mathcal{L}_{\mathrm{ReForce}} = \mathbb{E}_{t\sim\mathcal{D}}\left[\frac{1}{5}\sum_{i=1}^{5}\frac{1}{D_i}\left\|\Delta q_t^{\mathrm{demo},(i)} - \Delta q_t^{\mathrm{cmd},(i)}\right\|_2^2\right]. \tag{5}$$

The training data include variations in contact timing, force response, missing contact, and residual force, exposing the controller to contact deviations that may also occur during real-world deployment.

3.3 Learning from Human Demonstration

Human demonstration retargeting. Human hand motion is retargeted into robot-hand joint configurations, while tactile measurements are calibrated and mapped to the corresponding robot fingertips. This produces the paired reference trajectory $\tau^{\mathrm{ref}} = \left\{\left(q_t^{\mathrm{ref}}, F_t^{\mathrm{ref}}\right)\right\}_{t=1}^{T}$, where T is the number of demonstration frames, q_t^{ref} is the retargeted robot configuration, and F_t^{ref} is the corresponding per-finger force reference.

Action-chunking policy. We train an ACT policy [18], parameterized by η, to predict a short chunk of future robot-space motion-and-force references. Its configuration-history input is $\mathbf{Q}_t = (q_{t-H+1}, \ldots, q_t)$, where H is the history length. During training, this history is drawn from the retargeted demonstration trajectory; during deployment, it is updated from the configured runtime history source.

The policy contains separate motion and force prediction branches. The motion branch conditions on the configuration history and task phase ϕ_t, whereas the force branch conditions on the task phase alone:

$$\left\{\hat{q}_{t+k|t}^{\mathrm{ref}}\right\}_{k=0}^{C-1} = \pi_\eta^q\left(\mathbf{Q}_t, \phi_t\right), \qquad \left\{\hat{F}_{t+k|t}^{\mathrm{ref}}\right\}_{k=0}^{C-1} = \pi_\eta^F\left(\phi_t\right). \tag{6}$$

Here, C is the chunk length, k is the offset within the chunk, and $\hat{q}_{t+k|t}^{\mathrm{ref}}$ denotes the reference for time $t+k$ predicted from the policy input at time t. The model is deterministic and does not take the measured fingertip force F_t as input. Consequently, the predicted force chunk represents phase-dependent contact intent learned from the demonstrations, while REFORCE uses online force feedback to adapt the robot command.

Training objective. Using uniform temporal weighting, the motion and force prediction losses for a complete chunk are

$$\mathcal{L}_q^{\mathrm{MSE}} = \frac{1}{CD_q}\sum_{k=0}^{C-1}\left\|\hat{q}_{t+k|t}^{\mathrm{ref}} - q_{t+k}^{\mathrm{ref}}\right\|_2^2, \qquad \mathcal{L}_F^{\mathrm{MSE}} = \frac{1}{CD_F}\sum_{k=0}^{C-1}\left\|\hat{F}_{t+k|t}^{\mathrm{ref}} - F_{t+k}^{\mathrm{ref}}\right\|_2^2. \tag{7}$$

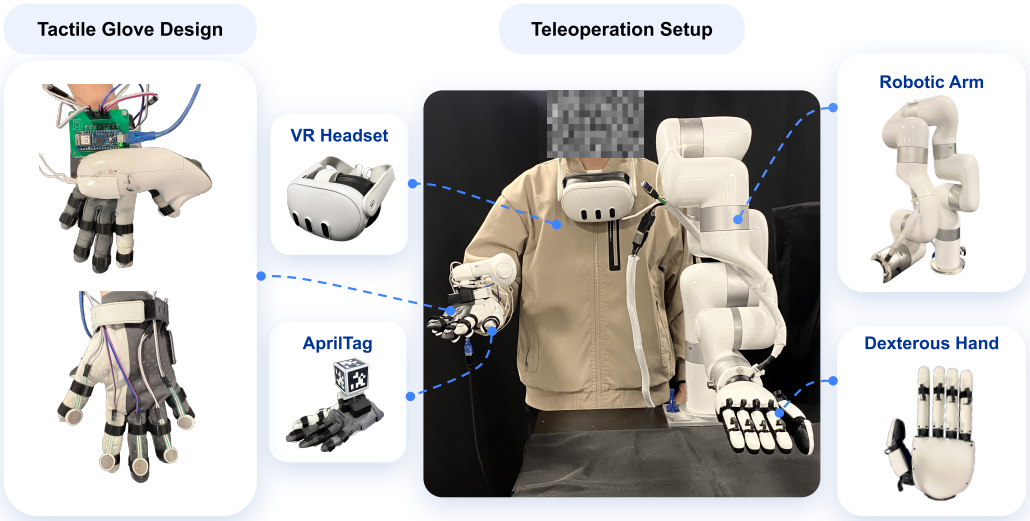


Figure 3: **Human demonstration and robot execution setup.** Teleoperation uses a Quest controller for wrist pose, a Manus glove for hand pose, and calibrated fingertip FSR sensors for human contact force. For ACT demonstration collection, a RealSense L515 camera tracks an AprilTag attached near the wrist, while the glove and FSR setup remains unchanged. A UFACTORY xArm equipped with an XHand executes the robot-side motion and provides fingertip tactile feedback.

The total training objective is

$$\mathcal{L}_{\mathrm{ACT}} = \mathcal{L}_q^{\mathrm{MSE}} + \lambda_F \mathcal{L}_F^{\mathrm{MSE}}, \tag{8}$$

where $\lambda_F \geq 0$ controls the relative weight of the force prediction loss. Invalid terminal-padding positions are masked out before averaging.

Composition with REFORCE. At deployment, the action-chunking policy is queried periodically to produce motion-and-force reference chunks. We apply standard temporal ensembling over overlapping chunks: at each control step, predictions from all active chunks that correspond to the current time are combined, with slightly larger weights assigned to more recent policy queries. The resulting motion reference $q_t^\star$ and force reference $F_t^\star$ are supplied directly to REFORCE. The measured fingertip force is provided separately to REFORCE as online feedback, allowing it to adapt the robot command while tracking the predicted references.

4 Experimental Evaluation

We evaluate REFORCE along three questions: (1) whether it improves real-world force tracking over direct replay and admittance control under the same reference trajectory; (2) whether it improves contact-aware execution when composed with a learned reference policy; and (3) how its training-data composition and input representation affect performance in simulation.

4.1 Real-World Evaluation

Hardware and human demonstrations. Figure 3 summarizes the sensing and execution setup. The robot platform consists of a UFACTORY xArm equipped with a 12-DoF XHand, whose fingertip tactile sensors provide the measured forces F_t. For teleoperation, a Quest controller provides the wrist pose, a Manus glove captures the human hand pose, and five fingertip FSR sensors provide the force reference $F_t^\star$. The human-side FSR measurements and robot-side tactile measurements are calibrated in Newtons so that $F_t^\star$ and F_t share a common physical scale.

For demonstrations used to train the action-chunking reference policy, an AprilTag and RealSense L515 RGB-D camera track the wrist alongside the glove and fingertip-force measurements. The policy used here is trained on the retargeted XHand joint trajectory and the calibrated per-finger forces; the wrist trajectory is recorded separately.

Tasks and evaluation criteria. The real-world experiments include paper-cup grasping and tongs manipulation. The replay comparison uses side and top paper-cup grasps, while the learned-reference evaluation includes both tasks. These tasks require sufficient multi-finger contact while avoiding forces that deform the manipulated object or exceed a safe operating range.

For all experiments, we measure force-tracking performance using the mean absolute force error over time and fingertips. For trajectory e with T_e logged time steps, the error is

$$\mathcal{E}_F^{(e)} = \frac{1}{D_F T_e} \sum_{t=1}^{T_e} \sum_{i=1}^{D_F} \left| F_{e,t,i} - F_{e,t,i}^{\mathrm{ref}} \right|, \tag{9}$$

where $F_{e,t,i}$ and $F_{e,t,i}^{\mathrm{ref}}$ are the measured and reference forces, respectively, for fingertip i at time step t in trajectory e. Errors are first computed independently for each trajectory and are then averaged equally across trials or episodes. This normalization makes the metric independent of trajectory duration and reports force-tracking error in Newtons.

For learned-reference experiments, we additionally report force-safe success, the number of over-force trials, severe missing-contact trials, and the mean number of active fingers. For paper-cup grasping, the force threshold is 1.0 N, while for tongs manipulation it is 3.0 N. These thresholds are empirically chosen based on repeated real-world trials to reflect task-specific force levels that avoid visible deformation or unsafe contact. Force-safe success requires task completion, forces below the task-specific threshold, and no severe missing-contact failure. A severe missing-contact failure is recorded when three or more fingers never establish contact during the trial. One paper-cup REFORCE trial with missing tactile data is excluded from force-based statistics.

Baselines. The *replay* baseline directly executes the nominal joint reference without force-dependent correction. The *admittance* baseline uses the same force reference and tactile observations as REFORCE, but maps force error to joint correction through a hand-designed task-space controller. For finger i, it computes

$$e_{i,t}^F = F_{i,t}^\star - F_{i,t}, \qquad u_{i,t} = k_p e_{i,t}^F + k_d \dot{e}_{i,t}^F. \tag{10}$$

Here, $e_{i,t}^F$ and $u_{i,t}$ are the scalar force error and force command, k_p and k_d are the corresponding gains, and $\dot{e}_{i,t}^F$ is the force-error derivative. A virtual task-space admittance model converts the force command into a fingertip displacement:

$$M_i \ddot{x}_{i,t} + B_i \dot{x}_{i,t} + K_i \left(x_{i,t} - x_{i,t}^{\mathrm{ref}} \right) = u_{i,t} \hat{n}_{i,t}. \tag{11}$$

Here, $x_{i,t}$ and $x_{i,t}^{\mathrm{ref}}$ are the virtual and reference fingertip positions, $\hat{n}_{i,t}$ is the unit contact normal, and M_i, B_i, and K_i are the virtual mass, damping, and stiffness. With $\Delta x_{i,t} = x_{i,t} - x_{i,t}^{\mathrm{ref}}$, the joint correction is

$$\Delta q_{i,t}^{\mathrm{adm}} = J_{i,t}^\top \left(J_{i,t} J_{i,t}^\top + \lambda_J^2 I_3 \right)^{-1} \Delta x_{i,t}. \tag{12}$$

Here, $J_{i,t} \in \mathbb{R}^{3 \times D_i}$ is the fingertip Jacobian, $\lambda_J > 0$ is the damping coefficient, I_3 is the identity matrix, and $\Delta q_{i,t}^{\mathrm{adm}} \in \mathbb{R}^{D_i}$ is the finger-joint correction. REFORCE replaces this hand-designed force-to-motion mapping with the learned force-aware correction policy described in Section 3.2.

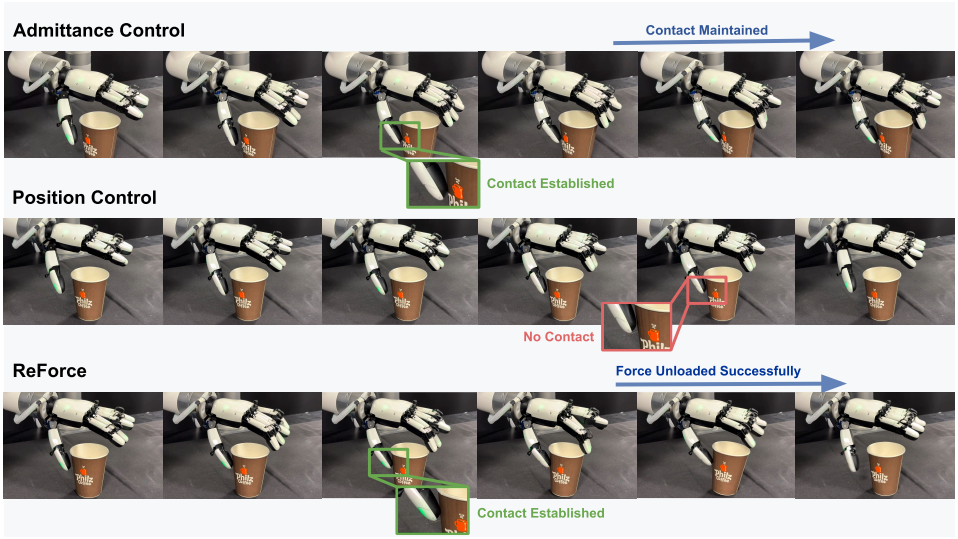


Figure 4: **Qualitative comparison of execution behavior.** Direct replay does not react to contact error, while admittance control uses a fixed force-to-motion mapping. REFORCE learns contact-dependent command corrections from interaction data.

Table 2: **Real-world execution with learned references.** Force-safe success requires task completion, forces below the task-specific threshold, and no severe missing-contact failure. Tracking error is the mean absolute force error over time and fingertips, reported in Newtons.

Task	Method	Force-safe success ↑	Tracking error ↓	Over-force ↓	Severe missing contact ↓	Active fingers ↑
Paper cup	Reference policy	30.0%	0.812	0/10	7/10	0.17
Paper cup	Reference policy + Admittance	60.0%	0.683	4/10	1/10	1.26
Paper cup	Reference policy + REFORCE	**70%**	**0.124**	2/10	**0/10**	**2.61**
Tongs	Reference policy	50.0%	0.454	0/10	5/10	0.94
Tongs	Reference policy + Admittance	**90.0%**	**0.419**	0/10	1/10	1.34
Tongs	Reference policy + REFORCE	54.5%	0.441	4/11	1/11	**1.62**

4.1.1 Force Tracking with Replayed References

Table 1 compares direct replay, admittance control, and REFORCE, with each method driven by the same recorded source trajectory for a given paper-cup grasp. REFORCE achieves the lowest force-tracking error for both side and top grasps.

Table 1: **Real-world paper cup grasping replay.** Force-tracking error in Newtons, reported as mean ± standard deviation over five trials. Lower is better.

Grasp	Replay	Admittance	REFORCE
Side	0.309 ± 0.006	0.280 ± 0.013	$\mathbf{0.247 \pm 0.036}$
Top	0.736 ± 0.000	0.619 ± 0.048	$\mathbf{0.474 \pm 0.049}$

Figure 4 summarizes the qualitative differences among the three execution methods. Detailed force trajectories for both grasp configurations are provided in Appendix B, Figure 6. Direct replay may fail to reproduce the demonstrated contact despite following the recorded motion, while admittance control can retain residual force after the reference decreases. REFORCE better balances contact establishment and force release by adapting the command from the measured interaction state.

4.1.2 Execution with Learned Motion-and-Force References

We next evaluate REFORCE when the reference is generated by the action-chunking policy described in Section 3.3. We compare the reference policy alone, with admittance control, and with REFORCE. Because low measured force can indicate either safe execution or missing contact, we report both force-safety and contact-engagement metrics.

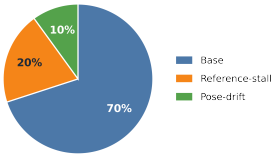


Figure 5: **Composition of the training mixture.**

Table 3: **Simulation ablation of training-data composition and policy inputs.** Force-tracking error is reported in Newtons as mean ± sample standard deviation over three seeds. Lower is better.

Training distribution	State + force target	+ pose reference	+ error features
Base	0.0601 ± 0.0002	0.0400 ± 0.0013	0.0388 ± 0.0015
Base + Reference-stall aug.	0.0592 ± 0.0012	0.0401 ± 0.0025	0.0381 ± 0.0005
Base + Pose-drift aug.	0.0609 ± 0.0006	0.0396 ± 0.0010	0.0385 ± 0.0013
Base + Both aug.	0.0590 ± 0.0012	0.0396 ± 0.0030	$\mathbf{0.0379 \pm 0.0006}$

For paper-cup grasping, REFORCE attains the highest force-safe success, the lowest force-tracking error, no severe missing-contact failures, and the largest mean number of active fingers. For tongs manipulation, admittance control achieves the lowest force-tracking error and the highest force-safe success, while REFORCE achieves the largest mean number of active fingers and substantially reduces severe missing-contact failures relative to the reference policy. Although REFORCE slightly improves tracking error over the reference policy on tongs, its four over-force trials reduce force-safe success. These results indicate that REFORCE consistently improves contact engagement, while its force-tracking advantage over admittance control is task-dependent.

4.2 Simulation Ablation of Training Data and Policy Inputs

We conduct a simulation ablation to study how training-data construction and policy input representation affect force tracking. Each configuration is trained with three random seeds and evaluated on the same held-out set of $N = 14{,}605$ simulated episodes. We use the force-tracking metric defined in Eq. 9, computing the error independently for each episode and then averaging equally across all episodes. For this evaluation, $F^{\mathrm{ref}}_{e,t,i}$ is the instantaneous force $F^{\mathrm{demo}}_{e,t,i}$ from the corresponding simulated demonstration trajectory, which is distinct from the future-window target $F^{\star}_{t}$ supplied to the policy.

Training-data construction. The object assets and target grasps used to seed these demonstrations come from Dex1B [52]. The *Base* distribution contains nominal simulated demonstrations with the standard observation perturbations used during training: Gaussian joint noise, force noise, and per-finger force dropout. It contains no additional structured recovery examples. We compare it with two targeted data constructions.

Reference-stall augmentation replaces 20% of the Base samples with examples that partially stall the pose reference by holding selected non-exempt joint-reference components fixed for force-active fingers ($F^{\star}_{t,i} \geq 0.1$ N), while retaining the demonstrated one-step action. This exposes the policy to stale motion references without changing the action supervision.

Pose-drift augmentation constructs synthetic recovery samples from low-force states. For each finger i satisfying $F^{\star}_{t,i} < 0.1$ N, the observed joints associated with that finger are perturbed toward joint-limit configurations. Let $\mathcal{J}_i$ denote the joint-index set of finger i and let $\tilde{q}^{\mathrm{obs}}_{t}$ denote the perturbed observation; q^{demo}_{t+1} is the next configuration in the corresponding simulated demonstration. Because this transformation changes the required action, its label is recomputed elementwise as

$$\Delta q^{\mathrm{recovery}}_{t,\mathcal{J}_i} = \mathrm{clip}\left(q^{\mathrm{demo}}_{t+1,\mathcal{J}_i} - \tilde{q}^{\mathrm{obs}}_{t,\mathcal{J}_i}, -5^{\circ}, 5^{\circ}\right). \tag{13}$$

Only eligible fingers receive the recomputed labels; the other fingers retain their demonstrated one-step updates. If an episode contains no eligible low-force state, the Base sampling procedure is used.

We evaluate four training mixtures: Base alone; Base with 20% reference-stall samples; Base with 10% pose-drift samples; and a combined mixture containing 70% Base, 20% reference-stall, and 10% pose-drift samples 5.

Policy inputs. We also study how the choice of policy inputs affects force-tracking performance by comparing three progressively richer input representations:

- **State and force target:** $\left[q_t^{\mathrm{obs}}, F_t, F_t^\star\right]$;
- **With pose reference:** $\left[q_t^{\mathrm{obs}}, F_t, q_t^\star, F_t^\star\right]$;
- **With error features:** $\left[q_t^{\mathrm{obs}}, F_t, q_t^\star, F_t^\star, q_t^\star - q_t^{\mathrm{obs}}, F_t^\star - F_t\right]$.

Adding the pose reference reduces force-tracking error across all training distributions by approximately 32–35% relative to the state-and-force-target input, and explicit motion and force errors provide a further improvement. Both structured augmentations improve the error-feature policy individually, while their combination performs best, attaining the lowest overall error of 0.0379 ± 0.0006 N.

5 Conclusion and Limitations

We presented REFORCE, a force-aware retargeting method that adapts human-guided motion and contact references using online fingertip force feedback. In real-world experiments, REFORCE improves force tracking and multi-finger contact engagement on contact-sensitive manipulation tasks, supporting its role as a force-aware execution layer between kinematic reference and robot control. The current system has two main limitations. First, the ACT-style reference policy relies only on joint-state history and task phase, limiting its ability to adapt references to the observed task state; incorporating visual observations could enable more task-aware motion-and-force reference generation. Second, REFORCE currently uses only normal fingertip force, while richer tactile signals such as shear force, slip, and contact distribution could provide a more complete representation of physical interaction.

References

[1] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022.

[2] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Computer Vision and Pattern Recognition (CVPR)*, 2024.

[3] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, G. Yang, J. Zhang, S. Yi, G. Shi, and X. Wang. Humanoid policy $\sim$ human policy. In *Conference on Robot Learning (CoRL)*, 2025.

[4] R. Zheng, D. Niu, Y. Xie, J. Wang, M. Xu, Y. Jiang, F. Castañeda, F. Hu, Y. L. Tan, L. Fu, et al. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

[5] R. Punamiya, S. Kareer, Z. Liu, J. Citron, R.-Z. Qiu, X. Cai, A. Gavryushin, J. Chen, D. Liconti, L. Y. Zhu, et al. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

[6] K. Li, P. Li, T. Liu, Y. Li, and S. Huang. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2025.

[7] S. Kareer, K. Pertsch, J. Darpinian, J. Hoffman, D. Xu, S. Levine, C. Finn, and S. Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025.

[8] H. Luo, Y. Feng, W. Zhang, S. Zheng, Y. Wang, H. Yuan, J. Liu, C. Xu, Q. Jin, and Z. Lu. Being-H0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[9] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170, 2020.

[10] 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 *Robotics: Science and Systems (RSS)*, 2023.

[11] W. Xie, S. Caldararu, and N. Correll. Just add force for contact-rich robot policies. *arXiv preprint arXiv:2410.13124*, 2024.

[12] Y. Hou, Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, S. Feng, B. Burchfiel, and S. Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4829–4836, 2025.

[13] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025.

[14] A. Adeniji, Z. Chen, V. Liu, V. Pattabiraman, R. Bhirangi, S. Haldar, P. Abbeel, and L. Pinto. Feel the force: Contact-driven learning from humans. *arXiv preprint arXiv:2506.01944*, 2025.

[15] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song. DexMachina: Functional retargeting for bimanual dexterous manipulation. In *International Conference on Machine Learning (ICML)*, 2026.

[16] C. Pan, C. Wang, H. Qi, 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] X. Zhu, Z. Liu, S. Jain, C. Li, M. Noori, H. Zhao, J. Welsh, M. A. Lin, W. Liu, T. Wang, X. Da, Z. Luo, V. Kulkarni, N. Bhatti, Y. Zhu, L. Fan, B. Wen, D. Xu, S. Pouya, and Y. Chang. Learning dexterous manipulation using contact wrench guidance from human demonstration. *arXiv preprint arXiv:2607.00033*, 2026.

[18] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[19] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 22955–22968, 2022.

[20] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[21] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[22] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning (CoRL)*, pages 1820–1864, 2023.

[23] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. DexMimicGen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[24] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163, 2024.

[25] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[26] Open X-Embodiment Collaboration et al. Open X-embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

[27] K. Yu, Y. Han, Q. Wang, V. Saxena, D. Xu, and Y. Zhao. MimicTouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[28] W. Liu, J. Wang, Y. Wang, W. Wang, and C. Lu. ForceMimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. *arXiv preprint arXiv:2410.07554*, 2025.

[29] D. Zhang, C. Yuan, C. Wen, H. Zhang, J. Zhao, and Y. Gao. KineDex: Learning tactile-informed visuomotor policies via kinesthetic teaching for dexterous manipulation. In *Conference on Robot Learning (CoRL)*, pages 4123–4138, 2025.

[30] H. Liu, Y. Jiang, H. Park, Y. Xue, and Z. Wang. DexTeleop-0: Force-aware bimanual dexterous teleoperation with ego-centric perception towards shared autonomy. *arXiv preprint arXiv:2606.23431*, 2026.

[31] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. DexMV: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision (ECCV)*, 2022.

[32] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *IEEE Robotics and Automation Letters*, 8(5): 2882–2889, 2023.

[33] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. In *Robotics: Science and Systems (RSS)*, 2022.

[34] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto. Holo-Dex: Teaching dexterity with immersive mixed reality. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969, 2023.

[35] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. DexCap: Scalable and portable MoCap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[36] S. Zhao, X. Zhu, Y. Chen, C. Li, X. Zhang, M. Ding, and M. Tomizuka. DexH2R: Task-oriented dexterous manipulation from human to robots. *arXiv preprint arXiv:2411.04428*, 2024.

[37] X. Liu, J. Adalibieke, Q. Han, Y. Qin, and L. Yi. DexTrack: Towards generalizable neural tracking control for dexterous manipulation from human references. *arXiv preprint arXiv:2502.09614*, 2025.

[38] C. Xin, M. Yu, Y. Jiang, Z. Zhang, and X. Li. Analyzing key objectives in human-to-robot retargeting for dexterous manipulation. *arXiv preprint arXiv:2506.09384*, 2025.

[39] J. Ye, L. Wei, G. Jiang, C. Jing, X. Zou, and X. Wang. From power to precision: Learning fine-grained dexterity for multi-fingered robotic hands. *arXiv preprint arXiv:2511.13710*, 2025.

[40] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[41] N. Hogan. Impedance control: An approach to manipulation: Part i—theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–7, 1985.

[42] J. Liang, X. Cheng, and O. Kroemer. Learning preconditions of hybrid force-velocity controllers for contact-rich manipulation. In *Conference on Robot Learning (CoRL)*, pages 679–689, 2022.

[43] Z. Kappassov, J.-A. Corrales, and V. Perdereau. Tactile sensing in dexterous robot hands—review. *Robotics and Autonomous Systems*, 74:195–220, 2015.

[44] W. Yuan, S. Dong, and E. H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[45] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

[46] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.

[47] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. ReSkin: Versatile, replaceable, lasting tactile skins. In *Conference on Robot Learning (CoRL)*, 2021.

[48] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang. Rotating without seeing: Towards in-hand dexterity through touch. In *Robotics: Science and Systems (RSS)*, 2023.

[49] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning (CoRL)*, pages 2549–2564, 2023.

[50] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. *arXiv preprint arXiv:2312.01853*, 2023.

[51] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, and M. Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *Conference on Robot Learning (CoRL)*, 2024.

[52] J. Ye, K. Wang, C. Yuan, R. Yang, Y. Li, J. Zhu, Y. Qin, X. Zou, and X. Wang. Dex1B: Learning with 1b demonstrations for dexterous manipulation. *arXiv preprint arXiv:2506.17198*, 2025.

Appendix

A Training Configuration

A.1 REFORCE Training Configuration

Table 4 summarizes the main training and architecture settings used for the REFORCE policy in our experiments.

Table 4: **REFORCE training configuration.** Main hyperparameters used for the reported experiments.

Item	Value
Future-target horizon	Up to 16 future frames; joint and force targets are averaged over the available future window
Policy architecture	Per-finger MLP policy; the four non-thumb fingers use a shared $[128, 128]$ trunk with $[64]$ finger-specific heads, while the thumb uses a separate $[128, 128]$ trunk and $[64]$ head; ReLU activations
Joint-update limit	0.088 rad per controlled joint, approximately 5°
Observation perturbations	Gaussian joint noise, Gaussian force noise with standard deviation 0.1 N, and per-finger force dropout with probability 0.30
Optimization	AdamW with learning rate 10^{-3}, weight decay 10^{-5}, batch size 1024, 20 epochs, and 500 training steps per epoch

A.2 ACT Reference-Policy Configuration

A separate ACT-style reference policy is trained for each task using approximately 30–40 human demonstrations. All reported task configurations use no visual input; the policy therefore takes only the configuration history $\mathbf{Q}_t$ and task phase ϕ_t as input.

Table 5: **ACT reference-policy training configuration.** Main parameters used for the learned motion-and-force reference policies.

Item	Value
Training data	Separate policy per task; approximately 30–40 human demonstrations per task
Temporal context	History length $H = 9$ and prediction chunk length $C = 32$
Sampling frequency	Raw demonstrations at 83 Hz; training anchors sampled at 10 Hz
Phase representation	$D_\phi = 3$, with $\phi_t = [\tau_t, \sin(2\pi\tau_t), \cos(2\pi\tau_t)]$, where $\tau_t = (t-1)/(T-1)$
Force-loss weight	$\lambda_F = 1.0$
Network architecture	Context MLP with widths $[128, 128, 256]$, followed by separate 2-layer motion and force Transformer decoders with width 256, 4 attention heads, feed-forward width 1024, and dropout 0.1

A.3 Policy Input Standardization

Joint configurations and forces are normalized before being used by the reference policy. Joint configurations are standardized directly using their training-set statistics. Because the force distribution is strongly concentrated near zero and has a long positive tail, force values are first log-transformed and then standardized. Elementwise, the transformations are

$$\bar{q} = \frac{q - \mu_q}{\sigma_q}, \qquad \bar{F} = \frac{\log\left(1 + \max(F, 0)/s_F\right) - \mu_F}{\sigma_F}, \tag{14}$$

where μ_q and σ_q are the training-set statistics of the joint configurations, μ_F and σ_F are the statistics of the log-transformed forces, and $s_F = 1.0$ N is the force log-scale parameter.

B Real-Robot Force-Tracking Curves

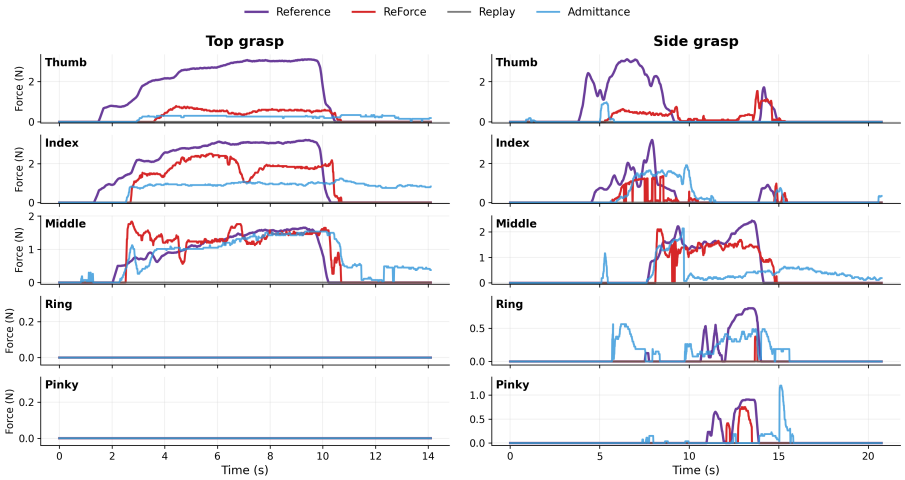
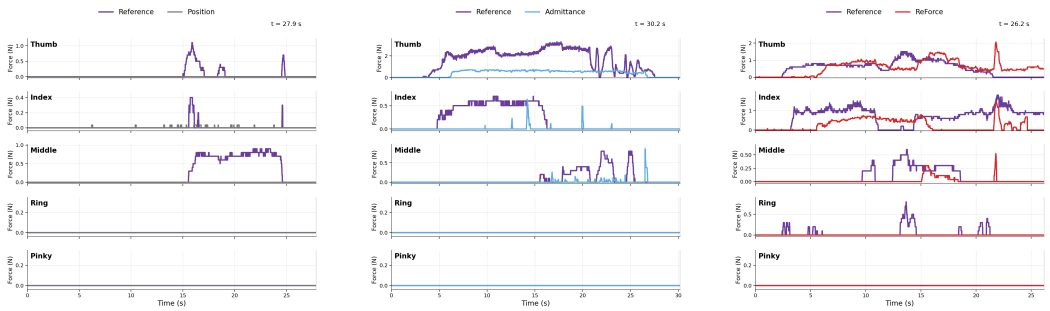


Figure 6: **Force tracking under replayed references.** Real-robot force trajectories for side and top paper-cup grasps using direct replay, admittance control, and REFORCE. These curves correspond to the results in Table 1.



(a) Position replay **(b) Admittance control** **(c) ReForce**

Figure 7: **Representative paper-cup force-tracking trials.** Force trajectories produced by position replay, admittance control, and REFORCE.