

Current as Touch: Proprioceptive Contact Feedback for Compliant Dexterous Manipulation

Chenyang Ma,^{*} Yunchao Yao,^{*} Zhenyu Wei,^{*} Ruogu Li, Daniel Szafir,[†] Mingyu Ding[†]
University of North Carolina at Chapel Hill
{mach, yunchao, wzhenyu, ruogu, dszafir, md}@cs.unc.edu

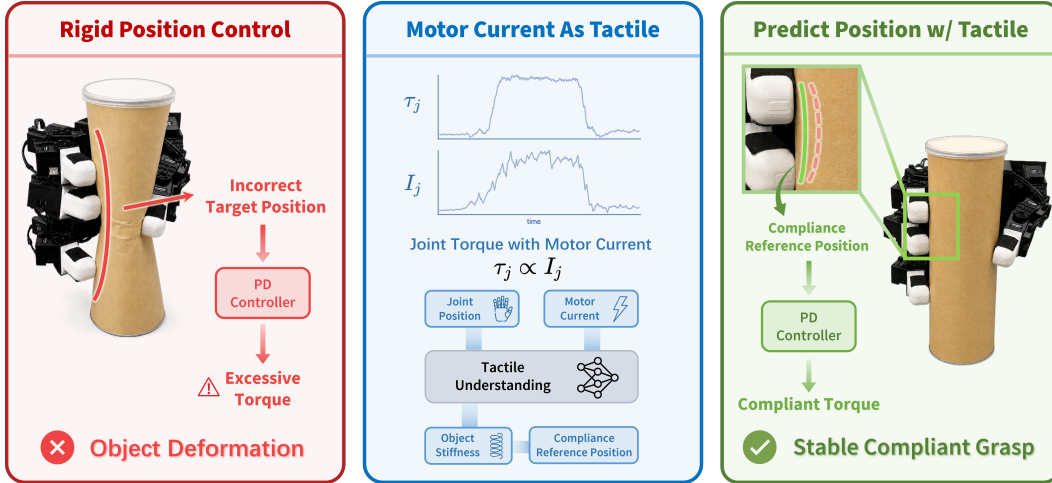


Figure 1: **Overview of the proposed proprioception-driven compliance framework.** **Left:** Rigid position control can generate excessive contact forces and deform the object. **Middle:** Our method uses motor current as tactile-like proprioceptive feedback to predict a compliance reference position (CRP) for standard PD control. **Right:** The predicted CRP induces compliant torque, enabling stable grasping without object damage or external tactile sensors.

Abstract: Compliance is essential for dexterous manipulation, yet existing solutions often rely on external tactile or force sensors that are costly, fragile, and difficult to deploy on low-cost robot hands. We propose a proprioception-driven framework that learns tactile-like contact feedback from motor current and joint states. Since motor current is closely related to actuator torque, it provides an intrinsic signal for perceiving contact force, object resistance, and grasp stability without additional sensing hardware. Rather than estimating external wrenches or commanding torque, our method predicts a compliance reference position: an ideal joint-position target for a standard PD controller whose induced position error generates appropriate grasping force. This position-based formulation is compatible with mainstream teleoperation and policy-learning pipelines, while enabling the robot to adapt interaction forces from real-time proprioceptive feedback. Thus, motor current serves not only as a force proxy but also as a learnable proprioceptive contact signal for compliance reference prediction. Experiments on multiple dexterous hands and contact-rich tasks, including fragile object handling, dynamic load adaptation, sustained surface contact, and thin-object retrieval, show stable compliant grasping, safer and more efficient teleoperation, and improved downstream policy learning without external tactile sensors. More details and demos could be found at our anonymous webpage: <https://cat.chenyangma.com/>

^{*}Equal contribution.

[†]Equal advising.

Keywords: Dexterous Manipulation, Compliance Control, Proprioceptive Sensing

1 Introduction

Dexterous manipulation in the real world requires more than accurate position tracking. When a robot hand interacts with fragile, deformable, or dynamically changing objects, it must continuously regulate contact forces while maintaining stable grasping [1, 2]. For example, stacking paper cups requires gentle contact to avoid deformation, pouring water requires adapting to changing load, wiping a board requires sustained contact, and retrieving a single card requires precise interaction with thin objects. These tasks are difficult for purely rigid position control, where an inaccurate target position can easily generate excessive force or cause slip.

A common way to achieve compliance is to equip robot hands with tactile sensors or force/torque sensors [3, 4, 5, 6, 7]. These sensors provide direct measurements of contact and enable force-aware control. However, they also introduce additional cost, fragility, calibration effort, and integration complexity. This limits their deployment on many low-cost dexterous hands, which are usually driven by joint-level position commands and low-level PD controllers. As a result, there remains a gap between the compliant interaction required by contact-rich manipulation and the hardware interfaces available in widely used robot hands.

This motivates us to ask whether tactile-like feedback can be obtained from signals intrinsic to the hand. In motor-driven dexterous hands, contact forces are generated through actuator torques, which are closely related to motor current. Motor current, together with joint states, therefore provides a built-in proprioceptive signal that reflects contact force and object resistance. As a motivating empirical observation, Fig. 2 illustrates that motor current and joint states vary consistently with measured contact force across different dexterous hands. A simple current-and-position regressor predicts measured normal force with RMSEs of 10.09g on Dex3 and 17.75g on LEAP Hand, with R^2 values of 0.99 and 0.95, respectively. This suggests that intrinsic motor signals contain useful contact information, motivating their use as proprioceptive feedback rather than explicit force estimates.

However, the goal of compliant manipulation is not merely to estimate contact force. Most teleoperation systems and learned policies for dexterous hands do not command torque directly; they output target joint positions that are tracked by low-level PD controllers. This creates a mismatch between tactile-like proprioceptive feedback, which reveals physical interaction, and the action space used by existing robot learning pipelines. To make such feedback practically useful, it must be converted into a position reference that can induce appropriate compliant torques through standard PD control.

In this work, we propose a proprioception-driven compliance framework that predicts a compliance reference position from motor current and joint states rather than direct torque commands. This reference is the ideal joint-position target sent to the PD controller: when tracked by the controller, the resulting position error generates the appropriate grasping force for stable contact without crushing the object. In this sense, compliance is represented as an adaptive position target: the robot still acts through the standard position-control interface, but the commanded target is informed by torque-

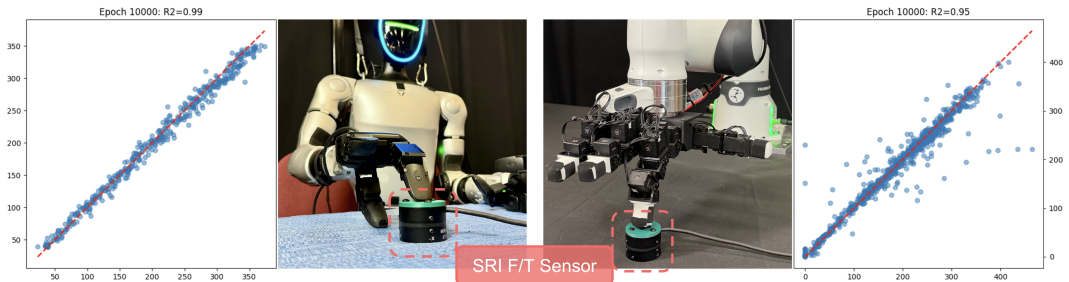


Figure 2: **Motivating measurement of motor current as proprioceptive contact feedback.** On Unitree Dex3 [8] and LEAP Hand [9], motor current and joint states vary consistently with contact force measured by an external force/torque sensor. This observation motivates using intrinsic actuator signals as contact feedback without treating them as explicit force estimates.

aware proprioceptive feedback. This formulation allows current-based contact feedback to be used as a general grasp representation for both teleoperation and policy learning.

We evaluate our framework on multiple dexterous hands and diverse contact-rich tasks, spanning fragile object handling, dynamic load adaptation, sustained surface contact, and thin-object retrieval. The results show that proprioceptive current feedback enables stable compliant grasping under varying contact conditions, improves teleoperation safety and efficiency, and benefits downstream policy learning across different hands and tasks.

In summary, the main contributions of this work are threefold:

- We learn torque-aware contact feedback from motor current and joint states, enabling tactile-free compliant manipulation using only intrinsic proprioceptive signals.
- We formulate compliant grasping as compliance-reference prediction, allowing proprioceptive contact feedback to induce compliant torques through standard PD controllers while remaining compatible with position-based teleoperation and policy-learning pipelines.
- We validate this proprioceptive compliance-reference interface on multiple hands and contact-rich tasks, improving teleoperation safety and efficiency while benefiting downstream policy learning.

2 Related Work

Tactile and Force-Torque Sensing for Contact-Rich Manipulation. Tactile and force sensing provide contact information for dexterous, contact-rich manipulation, where vision and position tracking often miss local geometry, slip, shear, and interaction forces. Prior work has developed high-resolution tactile fingertips such as GelSight [3], DIGIT [4], soft rounded tactile sensors [10], DTact [11], and 9DTact [12], as well as scalable tactile skins and arrays such as ReSkin [5], AnySkin [6], and flexible tactile arrays [13]. Recent learning systems incorporate tactile or visuo-tactile observations into manipulation policies, including 3D-ViTac [14], multi-modal tactile diffusion policies [15], and bimanual or reactive policies using tactile feedback [16, 17]. Force/torque sensing similarly provides direct wrench measurements, with compact or finger-mounted systems such as CoinFT [7] and UMI-FT [18] enabling force-aware manipulation learning. While effective, these approaches require additional tactile or force-torque hardware, calibration, and integration. In contrast, we study whether contact-aware signals can be recovered from motor current and joint states already available inside dexterous hands, avoiding external tactile or force/torque sensors.

Compliance Control with Proprioceptive Signals. Compliant manipulation has been extensively studied through hybrid position/force control, impedance control, and variable impedance control, which regulate robot motion based on physical interaction with the environment [19, 20, 21, 22, 23]. Although these methods enable stable and safe contact, they often require force/torque sensing, accurate dynamics, or torque-level control. Sensorless force estimation and current-based interaction control reduce this reliance by inferring external forces or joint torques from motor currents, joint states, disturbance observers, or actuator models [24, 25, 26]. Recent work has further explored compliant grasp planning and learned adaptive compliance for contact-rich manipulation [27, 28], and has begun to exploit motor-current signals for dexterous manipulation and compliance. For example, Zhao et al. [29] calibrate motor current into joint torque and combine it with dense tactile feedback in a sim-to-real RL framework for force-based dexterous grasping and manipulation. While this suggests that actuator-level signals capture useful physical interaction information, the method still relies on dense tactile observations, tactile simulation, and task-specific reinforcement learning. More closely related to our work, Minimalist Compliance Control [30] estimates external wrenches from motor current or voltage signals through motor models and robot Jacobians, and then updates position references with a task-space admittance controller. Although it demonstrates that actuator signals can substitute for external F/T sensors in compliance control, it still follows an explicit wrench-estimation and model-based admittance-control formulation. In contrast, our method learns compliance reference positions directly from motor current and joint states, avoiding

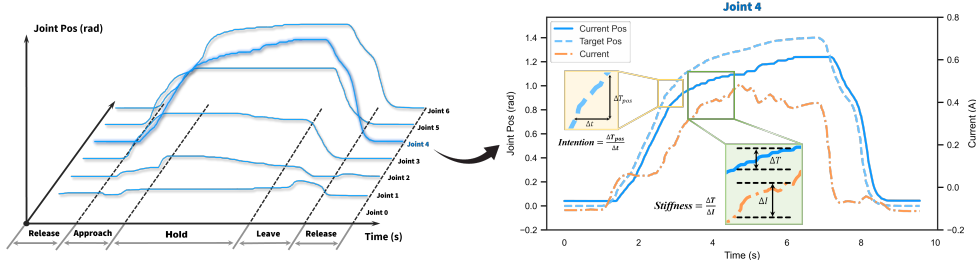


Figure 3: **Teleoperation data manifold for learning CRPs.** Demonstrations couple user intent with contact responses: command slopes encode grasp/release intent, while current changes relative to joint motion, e.g., $\Delta I / \Delta q$, provide stiffness-related contact cues. The model learns CRPs from this human-corrected manifold.

explicit wrench recovery while matching the joint-position command interface used by dexterous teleoperation and policy learning.

3 Methodology

Motivated by the empirical trend in Fig. 2, we use motor current, together with joint position, as proprioceptive contact feedback. Our goal is not to explicitly recover contact force or external wrench. Most dexterous hands expose a position-control interface, where a low-level PD controller tracks commanded joint references. Therefore, we formulate compliance as a reference-position prediction problem: given proprioceptive contact feedback, the model predicts a joint-position reference whose induced PD torque produces appropriate contact force.

3.1 Compliance Reference Position

Let q_t denote the measured joint position of the hand, \dot{q}_t the joint velocity, and $q_{\text{ref},t}^c$ the compliance reference position (CRP) predicted by our model. The low-level hand controller tracks this reference through a standard PD law:

$$\tau_t = K_p (q_{\text{ref},t}^c - q_t) - K_d \dot{q}_t, \quad (1)$$

where τ_t is the low-level controller torque, and K_p and K_d are fixed PD gains.

This formulation is particularly suitable for low-cost dexterous hands. If the reference is too far inside a rigid or fragile object, the PD controller produces excessive torque and may crush the object. If the reference is too close to the current position, the hand may fail to generate sufficient normal force and slip. The desired CRP therefore depends on the current hand state, contact condition, object response, and task intent. Our method learns this reference directly from data, using motor current as proprioceptive feedback about contact resistance.

3.2 Learning CRPs from Human-in-the-Loop Demonstrations

A key question is how we supervise q_{ref}^c . We use human teleoperation demonstrations as a human-in-the-loop closed-loop correction. During data collection, the operator does not command force directly. Instead, the operator observes the task outcome, such as object deformation, slip, or stable grasping, and continuously adjusts the target joint position q_{cmd} . As a result, the recorded command is not an arbitrary open-loop position target. It is a noisy but valid reference that has been corrected through visual feedback and whose induced PD torque achieves successful compliant interaction.

We therefore use the demonstrated command trajectory as supervision for the CRP:

$$q_{\text{ref},t}^{c,*} = q_{\text{cmd},t}. \quad (2)$$

These labels are not analytical optima; rather, they are noisy but task-valid CRP targets produced by human-in-the-loop closed-loop correction. Importantly, during inference, the model should not simply copy user command. Directly feeding $q_{\text{cmd},t}$ as input would leak the target into the observation

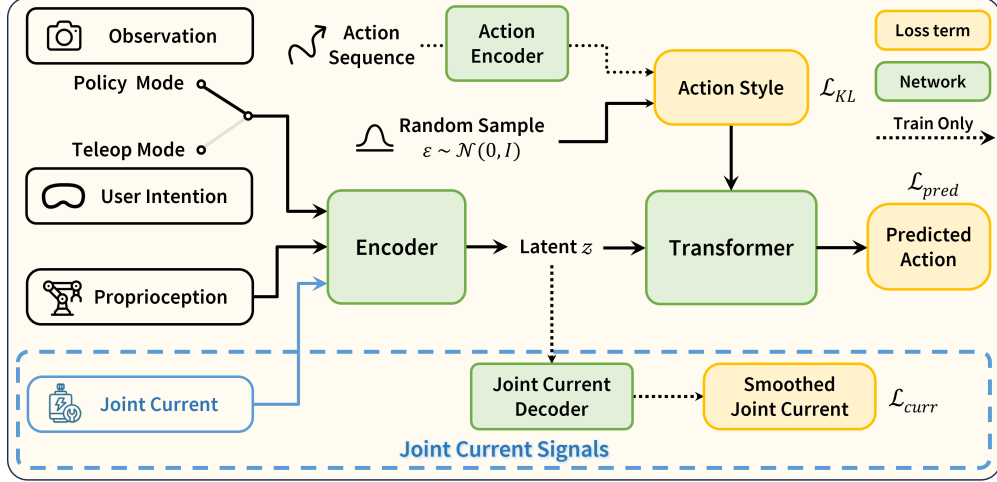


Figure 4: **Current-conditioned prediction pipeline.** The model predicts CRPs from proprioceptive history. In teleoperation, inputs include robot state, raw motor current, and user intent velocity; in policy mode, user intent is replaced by task-level object or goal pose.

and encourage shortcut learning. Instead, we represent user intent by the command velocity:

$$v_{\text{intent},t} = q_{\text{cmd},t} - q_{\text{cmd},t-1}. \quad (3)$$

This intent velocity captures whether the user is trying to grasp, release, or maintain contact, as well as how aggressively the user wants the hand to move. The model must combine this intent with the current hand state and motor-current feedback to predict the appropriate CRP.

Fig. 3 illustrates why such supervision contains learnable compliance signals. During free-space motion, position changes induce only small current variation, reflecting internal friction and actuator dynamics. When the hand contacts an object, small position changes can cause large current changes. The local relationship between Δq and ΔI provides contact stiffness information. Meanwhile, the slope of the user command trajectory provides grasping or releasing intent. Thus, the demonstration manifold contains both intent and contact-response information, allowing the model to learn how to map proprioceptive interaction signals to compliant reference positions.

3.3 Current-Conditioned Teleoperation and Policy Learning

Our framework supports two execution modes, as shown in Fig. 4. In teleoperation mode, the model assists a human operator by converting intent into a contact-aware CRP. Given an observation history of length T , the model receives

$$o_t^{\text{teleop}} = \{q_{t-T+1:t}, I_{t-T+1:t}, v_{\text{intent},t-T+1:t}\}, \quad (4)$$

where I_t is the raw motor current. The model predicts the next compliance reference position:

$$\hat{q}_{\text{ref},t}^c = f_\theta(o_t^{\text{teleop}}), \quad (5)$$

where f_θ denotes the learned CRP predictor parameterized by θ , q_t specifies the hand configuration, I_t provides contact-dependent feedback, and $v_{\text{intent},t}$ specifies the direction and speed of the user's intended motion. The predicted CRP then drives the hand through the standard PD controller.

In policy-learning mode, there is no online user command. The model instead conditions on task-level perception, such as the object or goal pose $g_t \in SE(3)$, together with proprioceptive feedback:

$$o_t^{\text{policy}} = \{q_{t-T+1:t}, I_{t-T+1:t}, g_t\}. \quad (6)$$

The demonstrated action is again used as the CRP supervision. In this setting, perception provides the task state, while motor current provides the contact state. This allows the policy to produce

different reference actions for the same visual state depending on whether the hand has established contact, is slipping, is overloaded, or has reached a stable grasp.

Both modes are trained with a reference prediction loss:

$$\mathcal{L}_{\text{ref}} = \left\| \hat{q}_{\text{ref},t}^c - q_{\text{ref},t}^{c,*} \right\|_2^2. \quad (7)$$

Our implementation follows ACT-style sequence modeling. As shown in Fig. 4, we retain the action encoder and KL regularization during training to capture variability in reference actions, while focusing on the current-conditioned observation encoder and auxiliary current supervision.

Although our network follows the sequence-modeling spirit of ACT, we mainly use temporal context on the observation side to capture recent current and motion trends. During real-robot execution, we apply a short exponential moving average over the most recent two predictions to reduce hardware jitter while preserving responsiveness to current changes.

3.4 Raw-Current Encoding with Smoothed-Current Auxiliary Supervision

Motor current is informative but noisy. Raw current often contains high-frequency PWM noise, communication spikes, and actuator disturbances. A straightforward solution would be to low-pass filter the current before feeding it into the policy, but filtering introduces phase delay, which weakens the response to sudden contact changes. We therefore feed raw, zero-delay current into the model.

Let \bar{I}_t denote an offline-smoothed current used only for supervision. Given the latent representation z_t from the observation encoder, an auxiliary decoder h_ϕ predicts the current and is trained with:

$$\hat{I}_t = h_\phi(z_t), \quad \mathcal{L}_{\text{cur}} = \left\| \hat{I}_t - \bar{I}_t \right\|_2^2. \quad (8)$$

The full training objective, with λ_{cur} balancing the auxiliary current-prediction loss, is:

$$\mathcal{L} = \mathcal{L}_{\text{ref}} + \lambda_{\text{cur}} \mathcal{L}_{\text{cur}}, \quad (9)$$

This auxiliary loss regularizes the latent representation to preserve slowly varying current patterns associated with contact and load, while the policy still uses raw current at inference. The auxiliary branch is training-only and adds no sensing or filtering delay at deployment.

4 Experiments

We evaluate four contact-rich tasks: teleoperated foam-cup stacking and whiteboard wiping, and policy-learning single-card picking and dynamic bottle holding. Teleoperation compares direct re-targeting, a current-free model, and our current-conditioned CRP predictor to test whether motor current is necessary for contact-conditioned reference prediction.

4.1 Experimental Setup

Experiments use the LEAP Hand on Franka Research 3 and Dex3 on Unitree G1. Robot state includes hand joints and arm end-effector pose. For teleoperation, *Retargeting* executes the retargeted operator command, *w/o Current* uses the same CRP predictor without current, and *w/ Current* is our full raw-current model. Demonstrations come from operators different from evaluation users, testing transfer rather than memorization of one correction style. For policy learning, the baseline is ACT-style behavior cloning with robot state and object pose; our method adds motor current. During execution, we average the first two predicted CRPs to reduce jitter while preserving responsiveness.

4.2 Teleoperation Tasks

Foam-cup stacking. The operator controls the LEAP Hand and Franka arm to stack four foam cups. We train on 550 trajectories: 50 grasping demonstrations for each of 10 objects with different

Table 1: Teleop foam-cup stacking results.

| Operator | Method | Time (s) ↓ | Deformed (%) ↓ | Grasp Failure (%) ↓ |
|----------|-------------|-------------|----------------|---------------------|
| Novice | Retargeting | 21.9 | 43.3% | 13.3% |
| | w/o Current | 31.1 | 15.0% | 76.7% |
| | w/ Current | 16.8 | 0.0% | 6.7% |
| Skilled | Retargeting | 20.7 | 25.0% | 15.0% |
| | w/o Current | 35.3 | 16.7% | 71.7% |
| | w/ Current | 16.1 | 0.0% | 3.3% |

Table 2: Teleop board-wiping results.

| Operator | Method | Success (%) ↑ | Time (s) ↓ |
|----------|-------------|---------------|------------|
| Novice | Retargeting | 40.0 | 9.3 |
| | w/o Current | 46.7 | 9.3 |
| | w/ Current | 100.0 | 6.9 |
| Skilled | Retargeting | 100.0 | 7.1 |
| | w/o Current | 60.0 | 8.3 |
| | w/ Current | 100.0 | 6.5 |

Table 3: Dynamic bottle-holding results.

| Water | Method | Stable (%) | Slipped (%) | Fell (%) |
|-------|-------------|--------------|-------------|----------|
| 0g | w/o Current | 100.0 | 0.0 | 0.0 |
| | w/ Current | 100.0 | 0.0 | 0.0 |
| 150g | w/o Current | 58.3 | 41.7 | 0.0 |
| | w/ Current | 83.3 | 16.7 | 0.0 |
| 250g | w/o Current | 16.7 | 58.3 | 25.0 |
| | w/ Current | 100.0 | 0.0 | 0.0 |
| 350g | w/o Current | 0.0 | 0.0 | 100.0 |
| | w/ Current | 41.7 | 58.3 | 0.0 |

Table 4: Single-card picking results. Strict success requires picking exactly one card, while tolerant success allows one or two cards.

| Method | Success Rate (%) | | Failure Mode (%) | |
|-------------|------------------|-------------|------------------|------------|
| | Strict | Tolerant | 2+ Cards | Missed |
| w/o Current | 55.8 | 65.4 | 7.7 | 26.9 |
| w/ Current | 76.9 | 90.4 | 0.0 | 9.6 |

stiffness, plus 50 free-space hand-motion trajectories. One novice user and one user with 10 minutes of practice each perform 15 trials.

Table 1 shows that our method achieves 100.0% success for both users and reduces completion time. The w/o Current model does not consistently improve over direct teleoperation because sustained wiping requires contact force regulation, not just an average motion prior. Current feedback indicates whether the eraser remains loaded against the board, allowing the CRP to maintain contact with less correction.

Board-wiping. The operator controls G1 with Dex3 to press an eraser against an inclined white-board and wipe it clean. We train on 100 demonstrations and evaluate the same two users over 15 trials each.

As shown in Table 2, our method achieves 100.0% success for both users and reduces completion time. The w/o Current model does not consistently improve over direct teleoperation, especially for the practiced user, because sustained wiping requires maintaining contact force against changing surface interaction rather than merely following an average motion prior. Current feedback helps the model detect whether the eraser remains properly loaded against the board, allowing the CRP to maintain contact with less manual correction.

4.3 Policy Learning Tasks

Dynamic bottle holding. A Franka-mounted LEAP Hand grasps a bottle while water is poured into it. We train on 100 demonstrations with approximately 250g poured at a near-constant rate. At test time, water is poured at random rates and evaluated at 0g, 150g, 250g, and 350g, where 350g is outside the training distribution. We conduct 12 trials per load.

This task highlights a subtle advantage of proprioceptive current feedback. Finger flexion currents mainly reflect grip closure rather than vertical load, but the LEAP Hand includes abduction/adduction motors at each finger base. During lateral grasping, increasing bottle weight raises the resistance and current on these joints, providing an implicit load signal.

Table 3 shows that both methods are stable at 0g, where little load adaptation is required. As the bottle becomes heavier, the baseline rapidly fails, whereas our method achieves 100% stability at 250g. Under the OOD 350g load, our method still prevents all drops, although some trials slip. These results suggest that motor current provides a load-sensitive signal, enabling partial generalization beyond the training load and avoiding catastrophic failure.

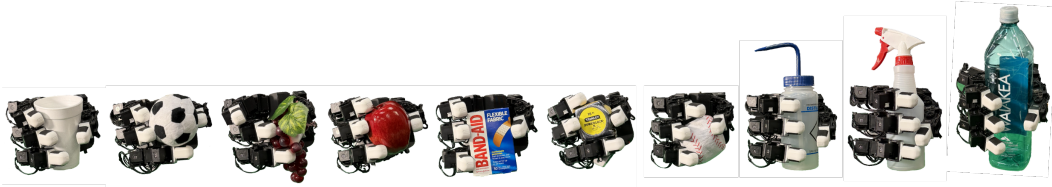


Figure 5: **Object set for teleoperation training and evaluation.** We collect grasping demonstrations on everyday objects with diverse shapes and stiffness, together with free-space hand-motion trajectories, to expose the model to both contact-induced and internal hand-motion currents.

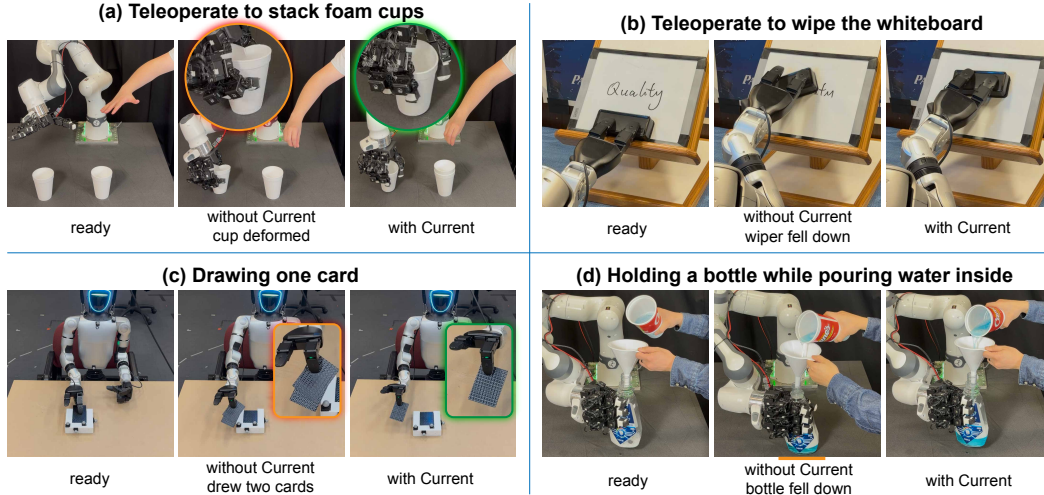


Figure 6: **Qualitative results across teleoperation and policy tasks.** Without current-conditioned CRP prediction, direct retargeting or behavior cloning can deform cups, lose wiping contact, draw multiple cards, or drop a dynamically loaded bottle. Motor-current feedback predicts contact-aware CRPs for compliant grasping, sustained contact, thin-object retrieval, and dynamic load adaptation.

Single-card picking. G1 with Dex3 must pick exactly one card from a deck. We train on 150 demonstrations with card SE(3) from VICON during training and inference, and evaluate 52 trials.

Table 4 shows that motor current improves strict success from 55.8% to 76.9% and tolerant success from 65.4% to 90.4%, while drawing-more-than-two-card failures drop from 7.7% to 0.0%. Current helps distinguish sufficient contact from excessive normal force. Without it, the policy infers contact only from position and object pose; with it, the policy receives proprioceptive evidence of contact formation and regulates the CRP more precisely.

5 Conclusion

We present a proprioception-driven compliance framework using motor current and joint states. Instead of external tactile/force sensors or explicit wrench estimation, our method predicts a compliance reference position (CRP): a joint-position reference whose PD error generates compliant grasping force. Human-in-the-loop teleoperation provides noisy but valid CRP supervision, while motor current supplies contact-dependent feedback for assisted teleoperation and policy learning. Across foam-cup stacking, whiteboard wiping, single-card picking, and dynamic bottle holding, experiments show safer teleoperation, better contact regulation, and stronger policy robustness.

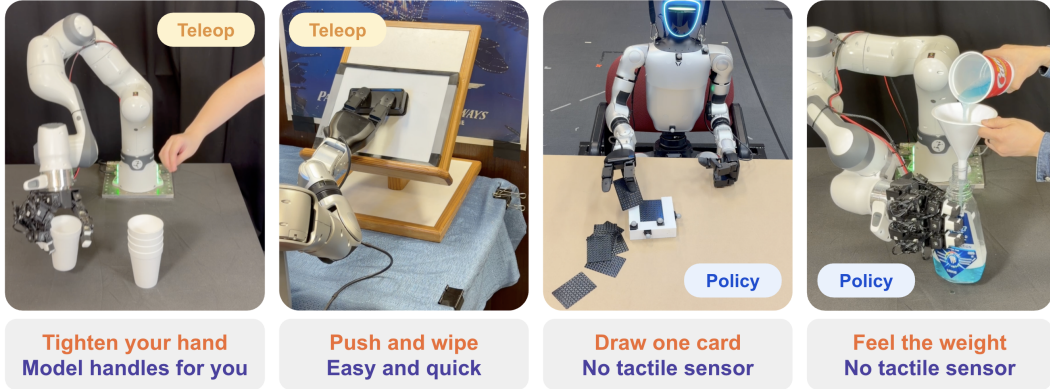
Limitations. Our method relies on motor current being informative about contact, which depends on hand hardware, transmission design, and current measurements. Dynamic load sensing, for example, benefits from the LEAP Hand’s finger abduction/adduction motors; hands without such joints may provide weaker shear or vertical-load signals. Demonstrations provide empirical CRP supervision rather than analytically optimal force labels, so performance may depend on demonstration quality and task coverage. Our policy experiments also use task-level object pose when needed and do not address perception or localization; integrating robust perception remains future work.

Our teleoperation experiments include a current-free CRP ablation, showing that kinematics and user intent alone are insufficient for reliable contact-conditioned references. We do not exhaustively ablate every component, such as the auxiliary current loss, execution smoothing, or heuristic current-threshold controllers; more detailed component studies remain future work.

References

- [1] A. Okamura, N. Smaby, and M. Cutkosky. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 255–262 vol.1, 2000. doi:10.1109/ROBOT.2000.844067.
- [2] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao. $\mathcal{D}(\mathcal{R}, \mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982–4988, 2025. doi:10.1109/ICRA55743.2025.11127754.
- [3] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.
- [4] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, et al. 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.
- [5] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*, 2021.
- [6] R. Bhirangi, V. Pattabiraman, E. Erciyas, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16563–16570. IEEE, 2025.
- [7] H. Choi, J. E. Low, T. M. Huh, S. Hong, G. A. Uribe, K. A. Hoffmann, J. Di, T. G. Chen, A. A. Stanley, and M. R. Cutkosky. Coinft: A coin-sized, capacitive 6-axis force torque sensor for robotic applications. *arXiv preprint arXiv:2503.19225*, 2025.
- [8] Unitree Robotics. Unitree dex3-1 power control dexterous hand. <https://www.unitree.com/Dex3-1>. Accessed: 2026-05-26.
- [9] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023.
- [10] B. Romero, F. Veiga, and E. Adelson. Soft, round, high resolution tactile fingertip sensors for dexterous robotic manipulation. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4802. IEEE, 2020.
- [11] C. Lin, Z. Lin, S. Wang, and H. Xu. Dtact: A vision-based tactile sensor that measures high-resolution 3d geometry directly from darkness. *arXiv preprint arXiv:2209.13916*, 2022.
- [12] C. Lin, H. Zhang, J. Xu, L. Wu, and H. Xu. 9dtact: A compact vision-based tactile sensor for accurate 3d shape reconstruction and generalizable 6d force estimation. *IEEE Robotics and Automation Letters*, 9(2):923–930, 2023.
- [13] Z. Zhao, J. Tang, J. Yuan, Y. Li, Y. Dai, J. Yao, Q. Zhang, S. Ding, T. Li, R. Zhang, et al. Large-scale integrated flexible tactile sensor array for sensitive smart robotic touch. *ACS nano*, 16(10):16784–16795, 2022.
- [14] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. *arXiv preprint arXiv:2410.24091*, 2024.

- [15] J. Zhao, N. Kuppuswamy, S. Feng, B. Burchfiel, and E. Adelson. Polytouch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 104–110. IEEE, 2025.
- [16] N. Gu, K. Kosuge, and M. Hayashibe. Tactilealoha: Learning bimanual manipulation with tactile sensing. *IEEE Robotics and Automation Letters*, 10(8):8348–8355, 2025. doi:10.1109/LRA.2025.3585396.
- [17] 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.
- [18] H. Choi, Y. Hou, C. Pan, S. Hong, A. Patel, X. Xu, M. R. Cutkosky, and S. Song. In-the-wild compliant manipulation with umi-ft. *arXiv preprint arXiv:2601.09988*, 2026.
- [19] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, June 1981. doi:10.1115/1.3139652.
- [20] N. Hogan. Impedance control: An approach to manipulation. In *1984 American Control Conference*, pages 304–313, 1984. doi:10.23919/ACC.1984.4788393.
- [21] R. Anderson and M. Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988. doi:10.1109/56.20440.
- [22] B. Siciliano and L. Villani. *Robot Force Control*, volume 540 of *The Springer International Series in Engineering and Computer Science*. Springer New York, New York, NY, 1 edition, 1999. ISBN 978-1-4615-4431-9. doi:10.1007/978-1-4615-4431-9.
- [23] F. J. Abu-Dakka and M. Saveriano. Variable impedance control and learning—a review. *Frontiers in Robotics and AI*, 7:590681, 2020.
- [24] A. Wahrburg, J. Bös, K. D. Listmann, F. Dai, B. Matthias, and H. Ding. Motor-current-based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control. *IEEE Transactions on Automation Science and Engineering*, 15(2):879–886, 2018. doi:10.1109/TASE.2017.2691136.
- [25] S. Liu, L. Wang, and X. V. Wang. Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation. *Robotics and Computer-Integrated Manufacturing*, 71:102168, Oct. 2021. doi:10.1016/j.rcim.2021.102168.
- [26] L. Han, J. Mao, P. Cao, Y. Gan, and S. Li. Toward sensorless interaction force estimation for industrial robots using high-order finite-time observers. *IEEE Transactions on Industrial Electronics*, 69(7):7275–7284, 2022. doi:10.1109/TIE.2021.3095820.
- [27] S. Chen, J. Bohg, and C. K. Liu. Springgrasp: Synthesizing compliant, dexterous grasps under shape uncertainty. *arXiv preprint arXiv:2404.13532*, 2024.
- [28] 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 *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4829–4836. IEEE, 2025.
- [29] Z. Zhao, H. Dong, Z. He, Y. Li, X. Yi, and Z. Li. Closing the reality gap: Zero-shot sim-to-real deployment for dexterous force-based grasping and manipulation. *arXiv preprint arXiv:2601.02778*, 2026.
- [30] H. Shi, S. Hu, Y. Hou, W. Wang, K. Liu, and S. Song. Minimalist compliance control. *arXiv preprint arXiv:2603.00913*, 2026.



Appendix

A Implementation Details

This appendix provides additional implementation details for the proposed current-conditioned compliance reference position (CRP) framework. The system uses proprioceptive histories, including joint positions and motor current, to predict position references that are executed by the standard low-level PD controller. Unless otherwise stated, the real-robot experiments use an observation horizon of 10 frames and predict a 10-frame future action chunk.

A.1 Dataset Details

Table 5 summarizes the datasets used in the four real-robot tasks. For all tasks, trajectories are split into 80% training and 20% evaluation sets.

Table 5: Dataset statistics for teleoperation and policy-learning tasks.

| Task | Hardware | Demonstrations | Average length / rate |
|--------------------------------|--------------------|----------------|-----------------------|
| Teleoperated object grasping | LEAP Hand + Franka | 550 | 10.2 s at 30.0 Hz |
| Teleoperated whiteboard wiping | Dex3 + Unitree G1 | 100 | 8.8 s at 28.6 Hz |
| Dynamic bottle holding | LEAP Hand + Franka | 100 | 14.9 s at 30.0 Hz |
| Single-card picking | Dex3 + Unitree G1 | 150 | 8.7 s at 27.9 Hz |

For teleoperated object grasping, we collect 50 grasping demonstrations for each of 10 objects with different stiffness: foam cup, toy football, grape, apple, band-aid, steel case ruler, toy baseball, water sprayer I, water sprayer II, and plastic water bottle. We additionally collect 50 free-space hand-motion demonstrations in the air. These free-space trajectories expose the model to motor-current patterns caused by internal hand motion rather than object contact, helping it separate contact-induced current changes from actuation currents. All object-grasping data are collected with the LEAP Hand mounted on a Franka arm.

For teleoperated whiteboard wiping, we collect 100 demonstrations on Dex3. For dynamic bottle holding, we collect 100 demonstrations with the LEAP Hand and Franka arm; each demonstration includes grasping an empty bottle and pouring 250 g of water into it. For single-card picking, we collect 150 demonstrations on Dex3 and G1, where each demonstration draws one card from the initialization pose.

A.2 Data Preprocessing

Raw signals. Each demonstration is stored as an HDF5 trajectory. For the dexterous hand, we record measured joint positions q_t , target joint positions q_t^{cmd} , and raw motor currents I_t . For

embodiments with an arm, we also record the arm joint positions and arm target positions. In the teleoperation setting, the operator target is converted into an intent velocity,

$$v_t^{\text{intent}} = q_t^{\text{cmd}} - q_{t-1}^{\text{cmd}}. \quad (10)$$

The first frame is assigned a zero intent velocity. This representation avoids feeding the target action directly to the model while preserving whether the operator intends to close, release, or hold the hand.

Current smoothing for labels and analysis. The model receives raw current at inference time, so it does not incur filtering delay. However, we use an offline-smoothed current as an auxiliary supervision target and for diagnostic plots. For each joint current trace, the preprocessing first applies a one-dimensional median filter and then a uniform moving average:

$$I_{t,j}^{\text{med}} = \text{MedianFilter}_{k_m}(I_{t,j}), \quad (11)$$

$$\bar{I}_{t,j} = \frac{1}{k_u} \sum_{s \in \mathcal{W}_{k_u}(t)} I_{s,j}^{\text{med}}. \quad (12)$$

The default implementation uses $k_m = 5$ and $k_u = 5$ with nearest-boundary padding. The median filter suppresses isolated communication or PWM spikes, while the moving average preserves the slower load-dependent trend used by the auxiliary current loss.

Chunk construction. For each valid time index t , we construct an observation window $o_{t-H+1:t}$ of length $H = 10$ and an action chunk $a_{t:t+K-1}$ of length $K = 10$. At the beginning of a trajectory, missing history frames are padded by repeating the first available frame. The teleoperation observation is

$$o_t^{\text{teleop}} = [v_t^{\text{intent}}, q_t^{\text{robot}}, I_t], \quad (13)$$

where q_t^{robot} concatenates the current hand and arm joint positions when an arm is used. For autonomous policy learning, the online user intent term is removed and replaced by task-level perception. In the single-card picking experiment, we use a VICON motion-capture system to measure the pose of the card deck relative to the robot base. In the dynamic bottle-holding experiment, the cup used for pouring is placed at a fixed position, and the corresponding precomputed $SE(3)$ pose is provided to the policy. The policy observation can therefore be written as

$$o_t^{\text{policy}} = [q_t^{\text{robot}}, I_t, g_t]. \quad (14)$$

where g_t denotes the task-level object, goal, or pouring pose. For the Dex3/G1 single-card policy preprocessing, the arm/object pose is represented as position plus a rotation representation, with the implementation using a 6D rotation representation by default.

Action space. The action label is the demonstrated target position after current-conditioned regularization by the human operator,

$$a_t = q_t^{\text{cmd}}, \quad (15)$$

including hand-joint references and, depending on the embodiment, either arm joint references or arm/end-effector pose references. This makes the learned output directly compatible with the position-control interface used by the corresponding hand-arm system. We evaluated both end-effector pose in $SE(3)$ and arm joint positions as arm-state inputs. Arm joint positions produced smoother and better tracked teleoperation actions, while policy-learning tasks showed no consistent performance difference between these two state choices.

Observation and action horizons. We tested observation horizons of 8, 10, 12, and 16 frames and did not observe a significant performance difference. We therefore use $H = 10$ as the default history length, which is long enough to capture recent current and motion trends while keeping inference lightweight. The action horizon is fixed to $K = 10$ frames. Since action chunking can produce multiple candidate commands for the same execution time, we study the number of recent predictions used in the exponential execution average in Section A.5.

Small-motion filtering for policy learning. Temporal imitation learning can be sensitive to nearly static segments because the intended future motion becomes ambiguous: many different future references can be consistent with almost identical observations. For runs in which small-motion filtering is enabled for policy-learning data, a candidate action chunk is kept only if its maximum target displacement exceeds a threshold,

$$\max_{s \in \{t, \dots, t+K-1\}} \|q_s^{\text{cmd}} - q_t^{\text{cmd}}\|_2 > \epsilon_{\text{move}}. \quad (16)$$

This removes idle clips and clips in which the operator is only holding a nearly constant command. We apply this filtering to the policy datasets, such as the single-card picking data, rather than to the teleoperation-assistance datasets. The appendix figure visualizes one trajectory and highlights which low-motion intervals are removed by this criterion.

Normalization. Observation, action, grasp-intention, and auxiliary current-label tensors are min-max normalized using statistics computed from the training split:

$$\tilde{x} = 2 \frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1. \quad (17)$$

Dimensions with range smaller than 10^{-6} are assigned unit range to avoid division by zero. The same normalization statistics are saved with each checkpoint and used during evaluation.

A.3 Model Architecture

The CRP predictor follows an ACT-style sequence model with a proprioceptive observation encoder, an action-style encoder used only during training, and a Transformer decoder over the future action chunk.

Observation encoder. The observation history is first transposed into channel-first form and passed through a temporal convolutional encoder:

$$z_t = E_{\theta}(o_{t-H+1:t}). \quad (18)$$

The implementation uses three one-dimensional convolution blocks with ReLU activations, temporal pooling, and a final projection with layer normalization. The default feature dimension is 64.

ACT-style action encoder and latent style. During training, the future action chunk is encoded together with the observation feature to produce a Gaussian latent style:

$$(\mu_t, \log \sigma_t^2) = A_{\psi}(a_{t:t+K-1}, z_t). \quad (19)$$

The latent is sampled with the reparameterization trick,

$$z_t^{\text{style}} = \mu_t + \sigma_t \odot \eta, \quad \eta \sim \mathcal{N}(0, I). \quad (20)$$

At inference time, no future action is available, so the style latent is set to zero. This preserves the ACT training regularization while keeping deployment causal.

Action chunk decoder. The observation feature and style latent are projected into a model dimension of 128 and added to learned positional embeddings over the $K = 10$ future action tokens. A four-layer Transformer encoder with four attention heads predicts the future CRP chunk:

$$\hat{a}_{t:t+K-1} = D_{\omega}(z_t, z_t^{\text{style}}). \quad (21)$$

The default dropout is 0.1.

Auxiliary current head. When auxiliary current supervision is enabled, a lightweight MLP predicts the offline-smoothed current \bar{I}_t from the observation feature z_t . This branch is used only for training and is removed from the control loop at deployment, so the deployed policy still conditions on raw current without phase delay.

Table 6: Default model and training hyperparameters.

| Hyperparameter | Value |
|---|--------------------|
| Observation horizon H | 10 frames |
| Action horizon K | 10 frames |
| Optimizer | AdamW |
| Learning rate | 1×10^{-4} |
| Batch size | 32 |
| Epochs | 300 |
| Observation feature dimension | 64 |
| Transformer model dimension | 128 |
| Transformer layers | 4 |
| Attention heads | 4 |
| Dropout | 0.1 |
| KL weight λ_{KL} | 1×10^{-5} |
| Auxiliary current weight λ_{cur} | 0.1 |
| KL annealing length | 100 epochs |
| Checkpoint interval | 2 epochs |
| Evaluation interval | 1 epoch |

A.4 Training Objective and Schedule

The main action loss is the mean-squared error between the predicted action chunk and the demonstrated CRP chunk:

$$\mathcal{L}_{\text{ref}} = \|\hat{a}_{t:t+K-1} - a_{t:t+K-1}\|_2^2. \quad (22)$$

The ACT-style latent is regularized with the KL divergence between the encoded posterior and a unit Gaussian prior:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \mathbb{E} \left[\sum_d (1 + \log \sigma_{t,d}^2 - \mu_{t,d}^2 - \sigma_{t,d}^2) \right]. \quad (23)$$

If auxiliary current prediction is enabled, the current head is trained with

$$\mathcal{L}_{\text{cur}} = \|\hat{I}_t - \bar{I}_t\|_2^2. \quad (24)$$

The total loss at epoch e is

$$\mathcal{L} = \mathcal{L}_{\text{ref}} + \lambda_{\text{KL}}(e) \mathcal{L}_{\text{KL}} + \lambda_{\text{cur}} \mathcal{L}_{\text{cur}}. \quad (25)$$

The KL weight is linearly annealed:

$$\lambda_{\text{KL}}(e) = \lambda_{\text{KL}}^{\max} \min \left(1, \frac{e+1}{E_{\text{anneal}}} \right), \quad (26)$$

where the default $\lambda_{\text{KL}}^{\max} = 10^{-5}$ and $E_{\text{anneal}} = 100$ epochs. This schedule prevents the latent from being over-regularized early in training while still discouraging uncontrolled action style variation later.

A.5 Action Chunking and Execution Aggregation

The model predicts a 10-frame sequence at each control step. During execution, multiple recent predictions may provide candidate CRPs for the current time index. We aggregate the most recent M candidates with an exponential average:

$$q_t^{\text{exec}} = \frac{\sum_{m=0}^{M-1} \alpha^m \hat{q}_t^{(t-m)}}{\sum_{m=0}^{M-1} \alpha^m}, \quad (27)$$

where $\hat{q}_t^{(t-m)}$ denotes the CRP for time t predicted m control steps earlier. We use a short aggregation window so that the command is smooth enough for hardware execution but still responsive to sudden current changes caused by contact.

Table 7: Execution aggregation ablation for dynamic bottle holding. Each setting is evaluated over 10 trials. Grasp success measures whether the robot first establishes a stable grasp on the empty bottle. Hold success measures whether the robot keeps the bottle from 0 g to 250 g of poured water after a successful grasp.

| Aggregated frames M | Grasp success | Hold success, 0 g to 250 g |
|-----------------------|---------------|----------------------------|
| 1 | 40% | 90% |
| 2 | 70% | 100% |
| 3 | 90% | 100% |
| 4 | 100% | 70% |

Table 7 shows the effect of the execution aggregation window. With $M = 1$, the command uses only the newest prediction. This makes the controller highly responsive, but the hand can enter small oscillations during grasp acquisition, reducing initial grasp success to 40%. Once a stable grasp is established, however, the current-conditioned CRP can still maintain the bottle in 90% of the pouring trials. Increasing the window to $M = 2$ improves grasp acquisition to 70% by damping frame-to-frame command jitter while preserving enough responsiveness to handle the increasing load. Using $M = 3$ further improves grasp success to 90% and still maintains 100% hold success from 0 g to 250 g. With $M = 4$, the initial grasp becomes very smooth and reaches 100% success, but the longer averaging window delays the response to dynamic load changes, reducing hold success to 70%. We therefore use a short aggregation window, with $M = 3$ providing the best balance between stable grasp acquisition and load-adaptive holding in these trials.

B KL Weight Ablations

We ablate the KL weight used by the ACT-style action encoder for teleoperation and bottle grasping. Each setting is evaluated over 10 real-robot trials.

Table 8: Teleoperation fist-closing success under different KL weights. Each setting is evaluated over 10 trials.

| $\lambda_{\text{KL}}^{\text{max}}$ | Successful trials / 10 | Success rate |
|------------------------------------|------------------------|--------------|
| 0 | 10/10 | 100% |
| 1×10^{-6} | 8/10 | 80% |
| 1×10^{-5} | 6/10 | 60% |
| 1×10^{-4} | 3/10 | 30% |

Table 9: Bottle-grasping success under different KL weights. Each setting is evaluated over 10 trials.

| $\lambda_{\text{KL}}^{\text{max}}$ | Successful trials / 10 | Success rate |
|------------------------------------|------------------------|--------------|
| 0 | 9/10 | 90% |
| 1×10^{-6} | 10/10 | 100% |
| 1×10^{-5} | 10/10 | 100% |
| 1×10^{-4} | 9/10 | 90% |

The teleoperation result shows a clear sensitivity to the KL weight. With $\lambda_{\text{KL}}^{\text{max}} = 0$, the model follows the user’s closing intent most directly and succeeds in all 10 trials. As the KL weight increases, the latent action style is pulled more strongly toward the prior, making the predicted CRPs more conservative and less able to follow the user’s online motion. This reduces fist-closing success from 100% to 30% as $\lambda_{\text{KL}}^{\text{max}}$ increases from 0 to 10^{-4} .

In contrast, bottle-grasping success is much less sensitive to the tested KL weights. All settings achieve either 90% or 100% success, suggesting that this policy task is dominated by the proprioceptive and task-state inputs rather than fine-grained user-following behavior. The default $\lambda_{\text{KL}}^{\text{max}} = 10^{-5}$ therefore provides stable bottle grasping while retaining the regularization benefits of the ACT-style latent for sequence prediction.

C Motor Current, Joint Torque, and Contact Force

Motor current is related to actuator torque, but the measured current is not a pure contact-force signal. This section makes explicit the main physical terms that appear in the measurement and explains why we learn a CRP directly rather than analytically converting current into contact force.

C.1 Current-to-Torque Model

For a motorized joint j , the electromagnetic torque can be approximated by

$$\tau_{j,t}^{\text{em}} = k_{\tau,j} (I_{j,t} - I_j^0), \quad (28)$$

where $k_{\tau,j}$ is the motor torque constant and I_j^0 is a current offset caused by electronics, calibration bias, and static preload. The torque available at the joint is affected by transmission and internal losses:

$$\tau_{j,t}^{\text{joint}} = \eta_j r_j \tau_{j,t}^{\text{em}} - \tau_{j,t}^{\text{fric}} - \tau_{j,t}^{\text{dyn}} - \tau_{j,t}^{\text{bias}} + \epsilon_{j,t}^{\tau}. \quad (29)$$

Here r_j is the transmission ratio, η_j is transmission efficiency, τ^{fric} includes Coulomb and viscous friction, τ^{dyn} includes inertial, Coriolis, and motor back-EMF effects, τ^{bias} captures gravity, cable tension, tendon preload, gear backlash, and unmodeled elastic elements, and ϵ^{τ} is stochastic measurement and actuation noise.

A commonly used decomposition is

$$\tau_{j,t}^{\text{fric}} = b_j \dot{q}_{j,t} + c_j \text{sgn}(\dot{q}_{j,t}) + \tau_{j,t}^{\text{strib}}, \quad (30)$$

where b_j is viscous friction, c_j is Coulomb friction, and τ^{strib} captures low-velocity Stribeck effects. Backlash and tendon compliance can be represented as history-dependent terms:

$$\tau_{j,t}^{\text{bias}} = g_j(q_t) + \tau_j^{\text{cable}}(q_t, h_t) + \tau_j^{\text{backlash}}(h_t), \quad (31)$$

where h_t denotes recent motion history. These terms are difficult to identify accurately for low-cost dexterous hands and may drift with temperature, wear, and cable routing.

C.2 From Joint Torque to Contact Force

If the hand kinematics and contact location were exactly known, contact wrench f_t could be related to joint torque through the contact Jacobian:

$$\tau_t^{\text{contact}} = J_c(q_t)^\top f_t. \quad (32)$$

In practice, this inverse problem is ill-conditioned for dexterous hands. The contact location may be unknown, multiple fingers may contact the object simultaneously, object compliance changes the local normal direction, and sliding or rolling changes the effective Jacobian. Therefore, the measured current contains a mixture of useful contact information and nuisance effects:

$$I_t = \Phi_{\text{contact}}(q_t, \dot{q}_t, f_t) + \Phi_{\text{internal}}(q_t, \dot{q}_t, h_t) + \epsilon_t^I. \quad (33)$$

The first term is the contact-dependent part we want to exploit. The second term includes internal friction, motor dynamics, cable effects, backlash, gravity, and thermal drift. The final term includes random sensor noise, quantization, communication spikes, and unmodeled disturbances.

This decomposition also clarifies what can and cannot be learned from data. Many hardware- and manufacturing-dependent effects are not analytically known but are repeatable for a fixed robot, such as joint-dependent current offsets, transmission efficiency, typical cable friction, or backlash patterns under similar motion histories. A sufficiently diverse dataset can let the model absorb these systematic terms into the learned mapping from $(q_{t-H+1:t}, I_{t-H+1:t})$ to CRPs. In contrast, truly random terms, such as isolated communication spikes, quantization noise, or non-repeatable impacts, cannot be predicted deterministically from the observation history. They can only be attenuated statistically through temporal context, smoothing used for auxiliary labels, robust training data, and conservative execution averaging. We include this distinction because it explains both the strength and the limitation of current-based tactile feedback: current provides useful contact-correlated information, but it is not a noise-free tactile force sensor.

C.3 Why Learn CRPs Instead of Explicit Force

Our method does not require a calibrated conversion from current to force. Instead, it learns a mapping

$$\hat{q}_{\text{ref},t}^c = f_{\theta}(q_{t-H+1:t}, I_{t-H+1:t}, u_t), \quad (34)$$

where u_t denotes user intent in teleoperation or task-level perception in policy learning. The learned mapping can use the repeatable part of the current signal, such as increased motor load during contact, while learning to ignore or average over nuisance terms that are not predictive of successful CRPs. The PD controller then converts the predicted position reference into interaction torque:

$$\tau_t = K_p (\hat{q}_{\text{ref},t}^c - q_t) - K_d \dot{q}_t. \quad (35)$$

This position-reference formulation matches the command interface of common dexterous hands and avoids relying on an explicit wrench estimator, contact Jacobian inversion, or torque-control interface.