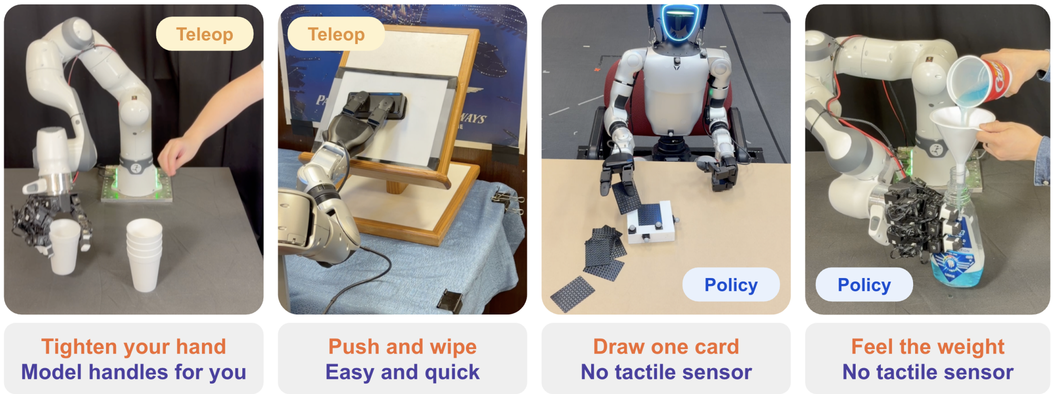

Appendix

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

Table 1: 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,

$$v_t^{\text{intent}} = q_t^{\text{cmd}} - q_{t-1}^{\text{cmd}}. \tag{1}$$

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:

$$I_{t,j}^{\text{med}} = \mathrm{MedianFilter}_{k_m}\left(I_{t,j}\right), \tag{2}$$

$$\bar{I}_{t,j} = \frac{1}{k_u} \sum_{s \in \mathcal{W}_{k_u}(t)} I_{s,j}^{\text{med}}. \tag{3}$$

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

$$o_t^{\text{teleop}} = \left[v_t^{\text{intent}},\ q_t^{\text{robot}},\ I_t\right], \tag{4}$$

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

$$o_t^{\text{policy}} = \left[q_t^{\text{robot}},\ I_t,\ g_t\right]. \tag{5}$$

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,

$$a_t = q_t^{\text{cmd}}, \tag{6}$$

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,

$$\max_{s \in \{t, \ldots, t+K-1\}} \left\| q_s^{\mathrm{cmd}} - q_t^{\mathrm{cmd}} \right\|_2 > \epsilon_{\mathrm{move}}. \tag{7}$$

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:

$$\tilde{x} = 2\frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1. \tag{8}$$

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:

$$z_t = E_\theta(o_{t-H+1:t}). \tag{9}$$

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:

$$(\mu_t, \log \sigma_t^2) = A_\psi(a_{t:t+K-1}, z_t). \tag{10}$$

The latent is sampled with the reparameterization trick,

$$z_t^{\mathrm{style}} = \mu_t + \sigma_t \odot \eta, \quad \eta \sim \mathcal{N}(0, I). \tag{11}$$

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:

$$\hat{a}_{t:t+K-1} = D_\omega(z_t, z_t^{\mathrm{style}}). \tag{12}$$

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 2: 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:

$$\mathcal{L}_{\mathrm{ref}} = \left\| \hat{a}_{t:t+K-1} - a_{t:t+K-1} \right\|_2^2 . \tag{13}$$

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

$$\mathcal{L}_{\mathrm{KL}} = -\frac{1}{2}\mathbb{E}\left[\sum_d \left(1 + \log \sigma_{t,d}^2 - \mu_{t,d}^2 - \sigma_{t,d}^2\right) \right] . \tag{14}$$

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

$$\mathcal{L}_{\mathrm{cur}} = \left\| \hat{\bar{I}}_t - \bar{I}_t \right\|_2^2 . \tag{15}$$

The total loss at epoch e is

$$\mathcal{L} = \mathcal{L}_{\mathrm{ref}} + \lambda_{\mathrm{KL}}(e)\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{cur}}\mathcal{L}_{\mathrm{cur}} . \tag{16}$$

The KL weight is linearly annealed:

$$\lambda_{\mathrm{KL}}(e) = \lambda_{\mathrm{KL}}^{\max} \min\left(1, \frac{e+1}{E_{\mathrm{anneal}}}\right) , \tag{17}$$

where the default $\lambda_{\mathrm{KL}}^{\max} = 10^{-5}$ and $E_{\mathrm{anneal}} = 100$ 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:

$$q_t^{\mathrm{exec}} = \frac{\sum_{m=0}^{M-1} \alpha^m \hat{q}_t^{(t-m)}}{\sum_{m=0}^{M-1} \alpha^m} , \tag{18}$$

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 3: 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 3 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 4: Teleoperation fist-closing success under different KL weights. Each setting is evaluated over 10 trials.

$\lambda_{\mathrm{KL}}^{\max}$	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 5: Bottle-grasping success under different KL weights. Each setting is evaluated over 10 trials.

$\lambda_{\mathrm{KL}}^{\max}$	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 $\lambda_{\mathrm{KL}}^{\max} = 0$, 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 $\lambda_{\mathrm{KL}}^{\max}$ 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 $\lambda_{\mathrm{KL}}^{\max} = 10^{-5}$ 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

$$\tau_{j,t}^{\mathrm{em}} = k_{\tau,j}\left(I_{j,t} - I_j^0\right), \tag{19}$$

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:

$$\tau_{j,t}^{\mathrm{joint}} = \eta_j r_j \tau_{j,t}^{\mathrm{em}} - \tau_{j,t}^{\mathrm{fric}} - \tau_{j,t}^{\mathrm{dyn}} - \tau_{j,t}^{\mathrm{bias}} + \epsilon_{j,t}^{\tau}. \tag{20}$$

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

$$\tau_{j,t}^{\mathrm{fric}} = b_j \dot{q}_{j,t} + c_j \,\mathrm{sgn}(\dot{q}_{j,t}) + \tau_{j,t}^{\mathrm{strib}}, \tag{21}$$

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:

$$\tau_{j,t}^{\mathrm{bias}} = g_j(q_t) + \tau_j^{\mathrm{cable}}(q_t, h_t) + \tau_j^{\mathrm{backlash}}(h_t), \tag{22}$$

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:

$$\tau_t^{\mathrm{contact}} = J_c(q_t)^{\top} f_t. \tag{23}$$

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:

$$I_t = \Phi_{\mathrm{contact}}(q_t, \dot{q}_t, f_t) + \Phi_{\mathrm{internal}}(q_t, \dot{q}_t, h_t) + \epsilon_t^{I}. \tag{24}$$

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

$$\hat{q}^{c}_{\mathrm{ref},t} = f_{\theta}(q_{t-H+1:t}, I_{t-H+1:t}, u_t), \tag{25}$$

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:

$$\tau_t = K_p \left(\hat{q}^{c}_{\mathrm{ref},t} - q_t\right) - K_d \dot{q}_t. \tag{26}$$

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.