# Refinement of Accelerated Demonstrations via Incremental Iterative Reference Learning Control for Fast Contact-Rich Imitation Learning

Koki Yamane[1,2], Cristian C. Beltran-Hernandez[2], Steven Oh[2], Masashi Hamaya[2], and Sho Sakaino[1]

***Abstract*— Providing fast demonstrations for imitation learning (IL) in contact-rich manipulation is challenging: humans cannot demonstrate at high speed, and naively accelerating demonstrations alters contact dynamics and induces large tracking errors. We present a method that autonomously refines time-accelerated demonstrations by repurposing Iterative Reference Learning Control (IRLC) to iteratively update the accelerated reference from observed tracking errors. However, applying IRLC directly at high speed produces large early-iteration errors and unsafe transients, and starting from a distant initial trajectory can distort the path. To address these issues, we propose *Incremental Iterative Reference Learning Control (I2RLC)*, which gradually increases the speed while updating the reference, yielding fast, high-fidelity trajectories. We validate on real-robot arc-erasing tasks using a teleoperation setup with a compliance-controlled follower and a 3D-printed haptic leader. Both IRLC and I2RLC achieve up to 10x faster demonstrations with reduced tracking error; moreover, I2RLC improves spatial similarity to the original trajectories by 36.5% on average across two tasks and multiple speeds. We then use the refined trajectories to train IL policies; the resulting policies execute faster than the demonstrations. These results indicate that gradual speed scheduling coupled with reference adaptation provides a practical path to fast IL for contact-rich manipulation.**

## I. Introduction

Imitation learning (IL) acquires manipulation skills from human demonstrations without extensive manual programming [1]. Recent advances [2], [3] achieve dexterous manipulation from few demonstrations. Yet, *fast, contact-rich manipulation* remains challenging: robots must move quickly while keeping interaction forces within safe bounds. This study investigates how to obtain high-speed demonstrations for contact-rich tasks safely.

A promising strategy for contact-rich IL is to pair a force-controlled follower with a haptic teleoperated leader [4]. Recent studies have developed low-cost systems using bilateral teleoperation with identical manipulators [5], [6], [7] or an industrial collaborative follower with a 3D-printed haptic leader [8], [9]. Whereas prior systems typically rely on torque control, we employ a widely deployed position-controlled arm equipped with an external force/torque (F/T) sensor and a compliance controller as the follower, coupled with a 3D-printed haptic leader. This configuration enhances safety; however, system latency and the demands of precise contact control make it difficult for operators to provide high-speed demonstrations, resulting in datasets dominated by slow-motion trajectories.

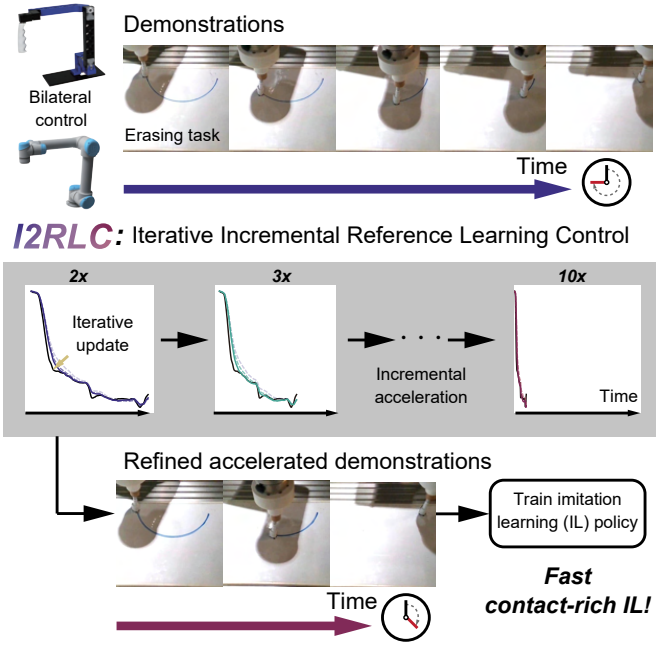


Fig. 1. Concept of the proposed method. I2RLC iteratively refines accelerated demonstrations while incrementally increasing the speed. The refined trajectories are then used for fast, contact-rich imitation learning.

One approach is to build a dataset of accelerated demonstrations by downsampling and replaying the control commands of the demonstrations at the same control frequency. However, such time-scaling alters contact dynamics and often induces large spatial deviations from the original trajectories. A more promising strategy is to execute the accelerated trajectories and refine them using observed tracking errors. Prior methods in this vein either require human intervention [10] or do not address contact-rich tasks [11].

This study proposes a method to generate time-accelerated trajectories for contact-rich manipulation without additional human intervention. Our key idea is to *repurpose* Iterative Reference Learning Control (IRLC) [12]—originally developed to reduce tracking error and improve insertion success—to refine accelerated demonstrations. However, applying IRLC directly at high speed often induces large errors in early iterations and unsafe behaviors, risking damage to

*This work was not supported by any organization.

[1] Intelligent and Mechanical Interaction Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. Email: yamane.koki.td@alumni.tsukuba.ac.jp, sakaino@iit.tsukuba.ac.jp.

[2] OMRON SINIC X Corporation, Nagase Hongo Building 3F, 5-24-5 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan. Email: cristian.beltran@sinicx.com, steven.oh@sinicx.com, masashi.hamaya@sinicx.com.

the environment or hardware and degrading spatial fidelity to the original demonstrations. To address these issues, we introduce *Incremental Iterative Reference Learning Control (I2RLC)*, which progressively increases execution speed while iteratively updating the reference, shown in Fig. 1. This incremental scheme safely reduces tracking error and yields faster trajectories suitable for contact-rich tasks.

We performed real-robot experiments on several erasing tasks and compared tracking performance on accelerated demonstrations under IRLC and I2RLC. Both methods refined the reference trajectories and reduced tracking error; however, I2RLC avoided large initial errors and preserved spatial similarity to the demonstrations. We then used the refined trajectories to train the Action Chunking Transformer (ACT) [3]. The resulting policies executed faster than the demonstrations.

**Contribution:** This study repurposes IRLC and extends it to I2RLC, a simple method that safely refines accelerated demonstrations to enable fast, contact-rich IL.

## II. Related Work

### A. Fast Contact-Rich Manipulation

Fast contact-rich manipulation has been achieved via control design, compliant hardware, and simulation-to-real training. A hybrid force–impedance controller with geometry-aware constraints enables torque-controlled robots to execute fast wiping [13]. High-speed contact manipulation has been demonstrated with highly backdrivable fingers [14] and passively compliant arms or grippers [15], [16], [17]. Simulation has also been used to train high-speed cutting policies via reinforcement learning [18]. These approaches typically rely on geometric priors, specialized hardware, or simulation. In contrast, this study targets fast contact-rich manipulation on widely deployed position-controlled arms via IL.

Recent IL methods, such as Force-aware ACT variants [6], [9], [19] and Diffusion Policy variants [20], [21], have demonstrated contact-rich skills including insertion, wiping, pivoting, and prying. Most work emphasizes robustness and generalization; here, we focus on faster execution while maintaining safe contacts. Our approach is complementary: it produces accelerated trajectories that can be used to train these IL backbones.

### B. Variable-Speed Imitation Learning

Several studies address variable speed. Dynamic Movement Primitives expose a time-scaling parameter to generate faster or slower trajectories [22]. Sakaino *et al.* proposed variable-speed IL by conditioning on a speed parameter, showing interpolation and extrapolation to unseen speeds [23]. Extensions adjust the policy's inference frequency to command higher speeds [10]. While these methods are powerful, providing sufficiently fast demonstrations and extrapolating to substantially higher speeds remains challenging. Downsampling-based data augmentation can expand coverage [24], but does not guarantee that the augmented trajectories remain executable when contact dynamics change. DemoSpeedup proposes an entropy-guided method, which enables safe demonstration acceleration [25], but does not focus on spatial similarity to the original demonstration. SAIL introduces latency-aware scheduling and adaptive speed modulation, achieving up to four times faster execution than demonstrations [26]. Yet, it relies on rigid, high-gain position control that is less suitable for sustained contact.

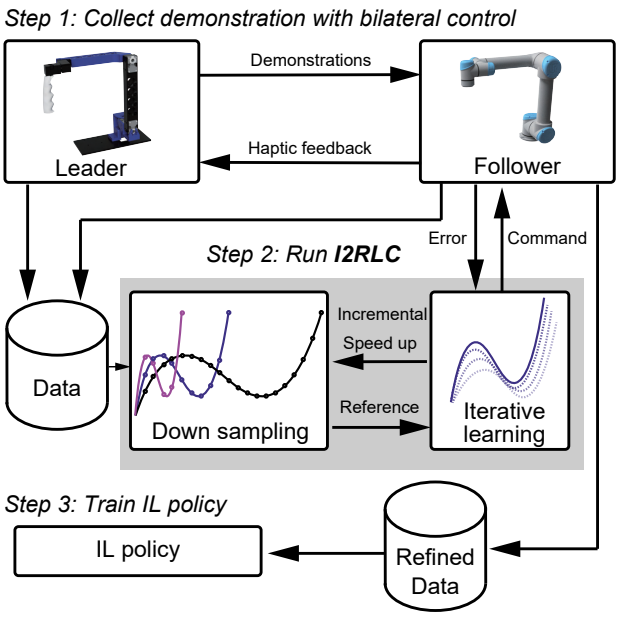


Fig. 2. The proposed pipeline comprises three stages: 1) collect demonstrations; 2) downsample to obtain accelerated demonstrations and refine them with I2RLC; and 3) train IL policies on the refined demonstrations.

### C. Online Trajectory Refinement

Several approaches can refine demonstrations online. Reinforcement learning can leverage imperfect demonstrations but typically requires substantial data collection [27], [28]. Operator-in-the-loop editing offers another route: *Motion Retouch* overwrites failure segments during time-accelerated execution via multilateral teleoperation [10], but it relies on human intervention. Iterative Learning Control (ILC) is a classical, data-efficient method for trajectory refinement [29]. Van den Berg *et al.* proposed an LQR-based ILC that gradually speeds up demonstrated trajectories [11]; however, it assumes linear dynamics and does not account for contact-rich tasks, whose interactions are inherently nonlinear.

We build on IRLC [12] and extend it to I2RLC. IRLC addresses limitations of conventional ILC for contact-rich insertion using an impedance controller. We show that the approach can also refine accelerated trajectories, yielding demonstrations suitable for contact-rich IL.

## III. Methodology

Our objective is to generate safe, accelerated demonstrations suitable for contact-rich IL. Fig. 2 illustrates the overview of our system. The setup comprises a position-controlled follower, a wrist-mounted F/T sensor at the arm's end effector, a 3D-printed haptic leader, and an RGB camera observing the workspace. An operator first provides

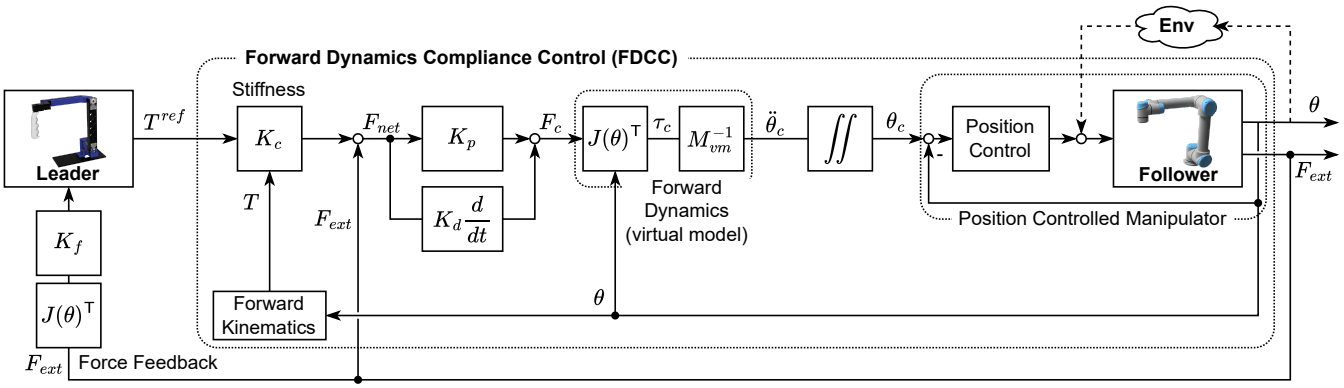


Fig. 3. Block diagram of Forward Dynamics Compliance Control (FDCC) [30] and force reflecting type bilateral teleoperation

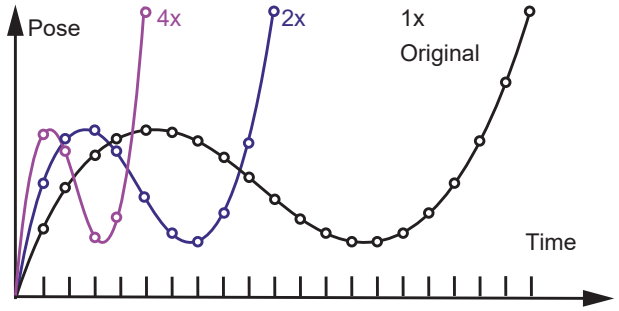


Fig. 4. We downsample desired trajectories and provide them to FDCC.

demonstrations via the bilateral leader–follower interface. Accelerated trajectories are produced by downsampling the reference motion. Next, I2RLC refines these accelerated trajectories. Finally, the refined demonstrations are used for IL policy training.

### A. Control System

Fig. 3 shows the block diagram of the control system.

**3D-printed haptic-feedback leader:** We employ a 3D-printed leader that provides haptic feedback. The leader builds on the General framework for low-cost and intuitive teleoperation systems (GELLO) [31] and its extensions [8], [9]. It has kinematics equivalent to that of the follower and uses low-cost servomotors with current control. We adopt force-feedback bilateral control: the follower's measured F/T $F_{\text{ext}}$ is mapped to the leader's joint torques via a feedback gain $K_f$. Gravity and friction compensation are also applied to the leader to reduce operator effort.

**Compliance controller:** To enable contact-rich manipulation, we use Forward Dynamics Compliance Control (FDCC) [30]. FDCC allows a position-controlled robot with an F/T sensor to move compliantly by combining impedance, admittance, and force control. A key component is a virtual forward-dynamics model that maps Cartesian wrench commands to joint accelerations.

Let $T^{ref}$ denote the follower's desired end-effector Cartesian pose computed from the leader's joint angles, and let $K_c$ denote the Cartesian stiffness. The commanded Cartesian force $F_c$ is determined by a desired net wrench $F_{\text{net}}$ and PD gains $K_p, K_d$. The desired joint acceleration $\ddot{\theta}$ is then obtained from the virtual forward-dynamics model given $F_c$, the current joint angles $\theta$, and a simplified inertia matrix $M_{vm}$.

### B. Data Collection and Demonstration Acceleration

We collect demonstrations of length $T$ time steps, forming $\mathcal{D} = \{(o_t, \boldsymbol{T}_t^{ref})\}_{t=1}^{T}$ , where $o_t$ comprises the F/T readings $F_{ext}$, an RGB image, a homogeneous transformation matrix of the measured follower's Cartesian pose $\boldsymbol{T}_t$, and the follower's joint angles $\theta$. To obtain accelerated demonstrations, we downsample the reference trajectory by an $n$-fold speedup ($n \in \{1, 2, ..., N\}$), resulting in $\hat{\boldsymbol{\tau}}_n = \{\boldsymbol{T}_{nt}\}_{t=1}^{\lfloor T/n \rfloor}$ and $\hat{\boldsymbol{\tau}}_n^{ref} = \{\boldsymbol{T}_{nt}^{ref}\}_{t=1}^{\lfloor T/n \rfloor}$ and (see Fig. 4), where $\lfloor \cdot \rfloor$ denotes the floor function. We then command the FDCC with setpoints $\hat{\boldsymbol{T}}_n^{ref}$ at a fixed control rate and use $\hat{\boldsymbol{T}}_n$ for IRLC described in the later section.

### C. Incremental Iterative Reference Learning Control

Classical iterative learning control suppresses modeling errors and disturbances by augmenting the torque command with a term learned from the previous trial's tracking error. It is used primarily for high-speed, high-precision position control and has typically been applied to non-contact tasks.

By contrast, recent work proposes Iterative Reference Learning Control (IRLC) [12], which places an impedance controller at the low level and updates its reference pose rather than directly correcting torques. This design preserves the intrinsic compliance of impedance control while enabling iterative refinement, making it suitable for contact-rich tasks.

We aim to refine the accelerated demonstrations using IRLC. The update rule can be expressed as follows:

$$
{}^{i}\bar{\boldsymbol{\tau}}_n^{ref} = \begin{cases} \left\{ {}^{(i-1)}\bar{\boldsymbol{T}}_t^{ref} \exp\left[ l \log\left\{ {}^{(i-1)}\boldsymbol{T}_t^{-1} \hat{\boldsymbol{T}}_{nt} \right\} \right] \right\}_{t=1}^{\lfloor T/n \rfloor} & (i \geqq 2), \\ \hat{\boldsymbol{\tau}}_n^{ref} & (i = 1), \end{cases} \tag{1}
$$

where ${}^{i}\bar{\boldsymbol{\tau}}^{ref}$ is an the updated reference trajectory at iteration $i \in \{1, 2, \ldots, I\}$, then is inputted to FDCC. $l$ is a learning gain. Here, exp and log denote the matrix exponential and

**Algorithm 1** Incremental Iterative Reference Learning Control (I2RLC)

**Require:** Demonstration data $\mathcal{D}$; max speed $N$ ; max updates per speed $I$; learning gain $l$

**Precompute** accelerated references from the demo: $\{\hat{\boldsymbol{\tau}}_{(n)}, \hat{\boldsymbol{\tau}}^{ref}_{(n)} = \textsc{Downsample}(\mathcal{D}, n)\}_{n=1}^{N}$
$\bar{\boldsymbol{\tau}}^{ref}_{(1)} \leftarrow \hat{\boldsymbol{\tau}}^{ref}_{(1)}$ ▷ initial reference at $1\times$
**for** $n = 2$ **to** $N$ **do** ▷ incrementally increase speed
  $\bar{\boldsymbol{\tau}}^{ref}_{(n)} \leftarrow \bar{\boldsymbol{\tau}}^{ref}_{(n-1)}$ ▷ warm start from previous speed
  **for** $i = 1$ **to** $I$ **do** ▷ IRLC updates at $n$-fold speedup
    $\bar{\boldsymbol{\tau}}^{ref}_{(n)} \leftarrow \textsc{Update}\big(\bar{\boldsymbol{\tau}}^{ref}_{(n)}, \boldsymbol{\tau}, \hat{\boldsymbol{\tau}}_{(n)}, l\big)$ ▷ Eq. (2)
    $\boldsymbol{\tau} \leftarrow \textsc{Playback}(\bar{\boldsymbol{\tau}}^{ref}_{(n)})$
  **end for**
**end for**
**return** $\{\bar{\boldsymbol{\tau}}_{(n)}\}_{n=1}^{N}$ ▷ refined accelerated demonstrations

logarithm, respectively, and these are used to handle 3D rotation.

We apply this approach to replay recorded demonstrations at high speed. Specifically, we downsample the demonstration to obtain an accelerated reference, execute it, measure the tracking error compared to the original, and update the reference. We repeat this procedure for several iterations. However, larger $n$-fold speedups can cause the uncorrected motion to deviate substantially, especially in the early iterations, leading to serious failures in contact-rich tasks.

To address this, we propose *Incremental Iterative Reference Learning Control (I2RLC)*, which increases the execution speed gradually across iterations while updating the reference from observed tracking errors. This incremental scheme enables iterative playback that remains close to the original trajectory even under substantial speedups. The update rule is:

$$ {}^{i}\bar{\boldsymbol{\tau}}^{ref}_{n} = \begin{cases} \left\{ {}^{(i-1)}\bar{\boldsymbol{T}}^{ref}_{t} \exp\left[ l \log\left\{ {}^{(i-1)}\boldsymbol{T}_{t}^{-1} \hat{\boldsymbol{T}}_{nt} \right\} \right] \right\}_{t=1}^{\lfloor T/n \rfloor} & (i \geq 2), \\ {}^{I}\bar{\boldsymbol{\tau}}^{ref}_{n-1} & (i = 1), \end{cases} \tag{2} $$

If $i \geq 2$, the update rule is identical to IRLC. For $i = 1$, we warm-start with ${}^{I}\bar{\boldsymbol{\tau}}^{ref}_{n-1}$ , the reference obtained after $I$ iterations at the $(n-1)$-fold speed stage. After completing the iterations at stage $n$, advance to the next stage by setting $n \leftarrow n + 1$. This warm start mitigates large tracking errors at the initial iteration. Algorithm 1 summarized I2RLC's procedures.

### D. IL Training

Given the refined accelerated demonstrations, we train IL policies. While I2RLC can be used for arbitrary IL algorithms, this study adopts ACT [3] because of its sample efficiency. ACT has a Transformer architecture and predicts sequences of future actions. These actions are stored and temporally ensembled for smoother action generations. The ACT input comprises $F_{ext}$, an RGB image, the follower's pose $\boldsymbol{T}$, and the output is the sequence of $\boldsymbol{T}^{ref}$. We used vector form for 6D pose, which consists of the 3D translation vector and the 3D rotation vector.

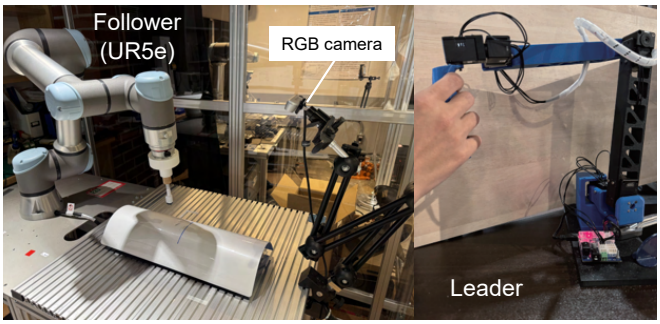


Fig. 5. Experimental setup.

TABLE I
CONTROLLER PARAMETERS

| | x | y | z | rx | ry | rz |
|---|---|---|---|---|---|---|
| $K_c$ | 300 | 300 | 300 | 100 | 100 | 100 |
| $K_p$ | 0.0252 | 0.0252 | 0.0252 | 0.36 | 0.36 | 0.36 |
| $K_d$ | 0.0072 | 0.0072 | 0.0072 | 0.0072 | 0.0072 | 0.0072 |

## IV. EXPERIMENTS

We validate our method on a real robot. Our evaluation addresses two questions: 1) Can I2RLC refine accelerated demonstrations, and 2) Can the refined trajectories be used effectively for IL? To this end, we conduct contact-rich erasing tasks and compare tracking performance, then train IL policies using the refined demonstrations.

### A. Robot System

Fig. 5 shows our experimental setup. We used a Universal Robots UR5e with a built-in F/T sensor as the follower, and an Intel RealSense D435 camera positioned at a fixed viewpoint. An eraser holder attached to the end effector incorporates springs to promote stable, compliant contact. The leader employed six servomotors: three Dynamixel XM430-W350-T units for the shoulder and elbow joints, and three Dynamixel XC330-T288-T units for the wrist joints.

The control frequency of the FDCC and the leader was 500 Hz, and the communication frequency between the leader and the follower was 50 Hz. The stiffness parameter and PD gain for FDCC in Fig. 3 were shown in Table I. The control System was implemented on ROS1 Noetic.

We used a desktop PC powered by an AMD Ryzen Threadripper 7960X CPU and two NVIDIA RTX 4000 Ada GPUs for robot control and train IL.

### B. Erasing Tasks

**Flat-Surface Arc Erasing:** We evaluated a contact-rich, trajectory-sensitive task in which the robot erases an arc drawn on a whiteboard using an end-effector-mounted eraser. Because the target is an arc rather than a straight line, a naively time-downsampled replay causes the end-effector path to deviate from the demonstration and may fail.

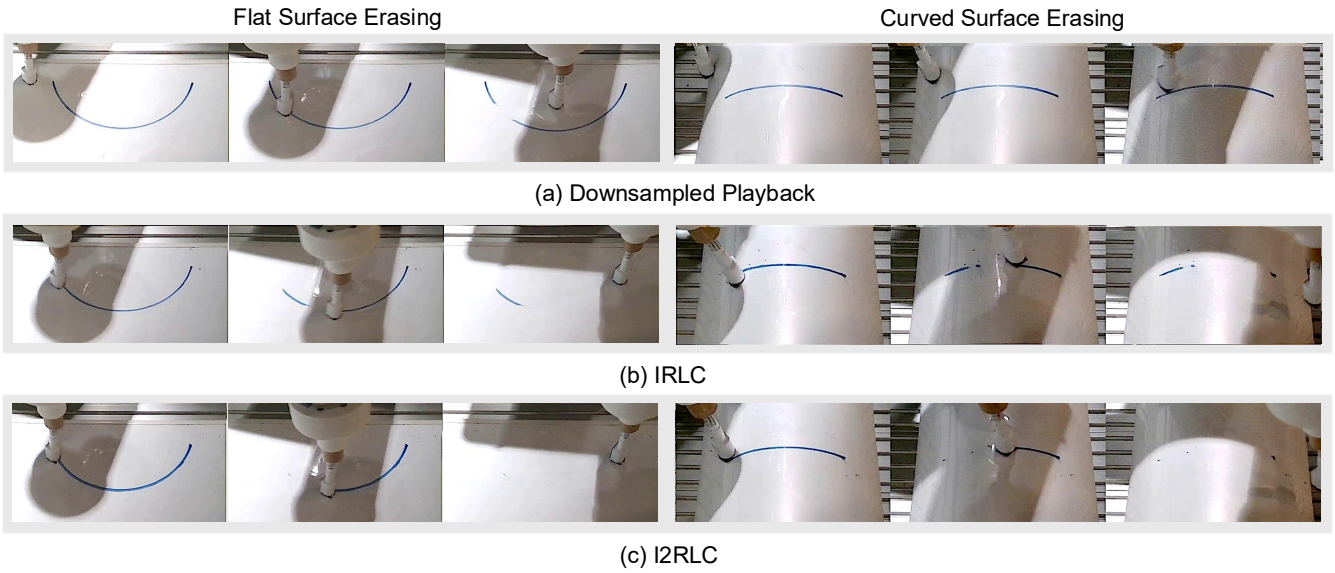


Fig. 6. Snapshots of erasing tasks in flat and curved surfaces (10x speed). I2RLC successfully erased the blue lines.

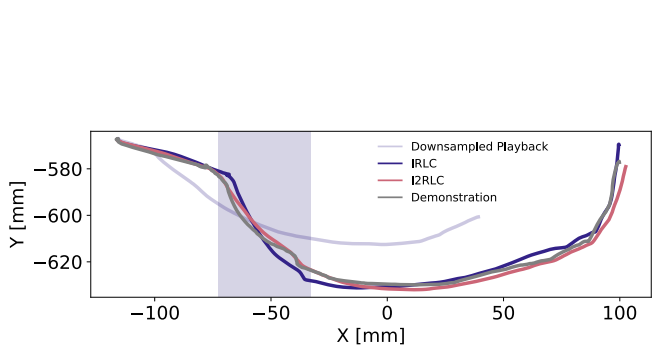


Fig. 7. Trajectory comparison for flat surface arc erasing at 10x speed. Shaded regions indicate segments where IRLC incurs larger errors, corresponding to the residual line segments in Fig. 6(b).

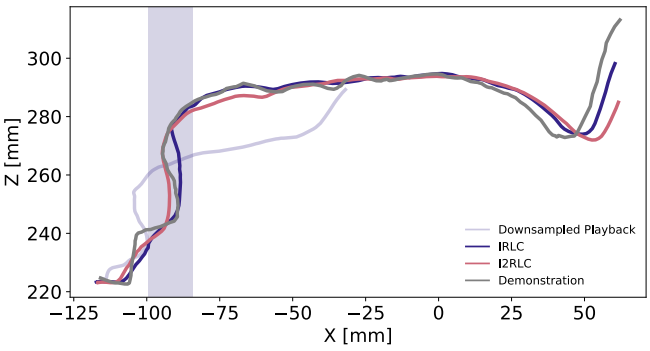


Fig. 8. Trajectory comparison for curved surface arc erasing at 10x speed.

TABLE II
IRLC AND I2RLC TRAJECTORY ERROR COMPARISON

| | | Speed | 3x | 4x | 5x | 6x | 7x | 8x | 9x | 10x |
|---|---|---|---|---|---|---|---|---|---|---|
| 6*Flat | MAE ↓ [mm] | Playback | 20.65 | 27.19 | 32.64 | 37.68 | 42.58 | 46.40 | 50.47 | 53.68 |
| | | IRLC | **3.43** | **3.65** | **3.88** | **4.00** | **4.16** | **4.36** | **4.61** | **4.78** |
| | | I2RLC | 5.13 | 5.68 | 6.39 | 7.20 | 7.44 | 7.91 | 8.37 | 8.57 |
| | DTW ↓ | Playback | 4.265 | 4.890 | 5.104 | 5.344 | 5.417 | 5.486 | 5.592 | 5.534 |
| | | IRLC | 0.998 | 1.213 | 0.964 | 0.816 | 0.662 | 0.625 | 0.587 | 0.598 |
| | | I2RLC | **0.545** | **0.488** | **0.462** | **0.524** | **0.474** | **0.532** | **0.442** | **0.414** |
| 6*Curved | MAE ↓ [mm] | Playback | 27.96 | 36.96 | 42.11 | 51.77 | 53.02 | 58.23 | 64.68 | 63.59 |
| | | IRLC | **4.54** | **4.93** | **5.08** | 8.23 | **6.08** | **6.77** | **7.70** | **7.52** |
| | | I2RLC | 5.57 | 6.03 | 6.50 | **7.62** | 7.57 | 8.30 | 8.71 | 8.93 |
| | DTW ↓ | Playback | 3.245 | 3.468 | 3.515 | 5.217 | 4.582 | 5.056 | 5.807 | 5.189 |
| | | IRLC | 1.094 | 1.050 | 0.902 | 1.280 | 0.897 | 0.837 | 0.975 | 0.836 |
| | | I2RLC | **1.061** | **0.864** | **0.744** | **0.611** | **0.531** | **0.500** | **0.438** | **0.472** |

**Curved-Surface Arc Erasing:** As a more challenging contact-rich task, the robot traces and erases an arc on a convex cylindrical surface. High-speed execution is difficult due to the increased likelihood of the eraser lifting off or applying excessive contact force.

### C. Incremental Iterative Reference Learning Control

*1) Setup:* We compared trajectory error for IRLC at a fixed target speed against I2RLC, which increased speed incrementally up to 10x. At each speed, we performed three update iterations ($I = 3$). The learning rate $l$ was

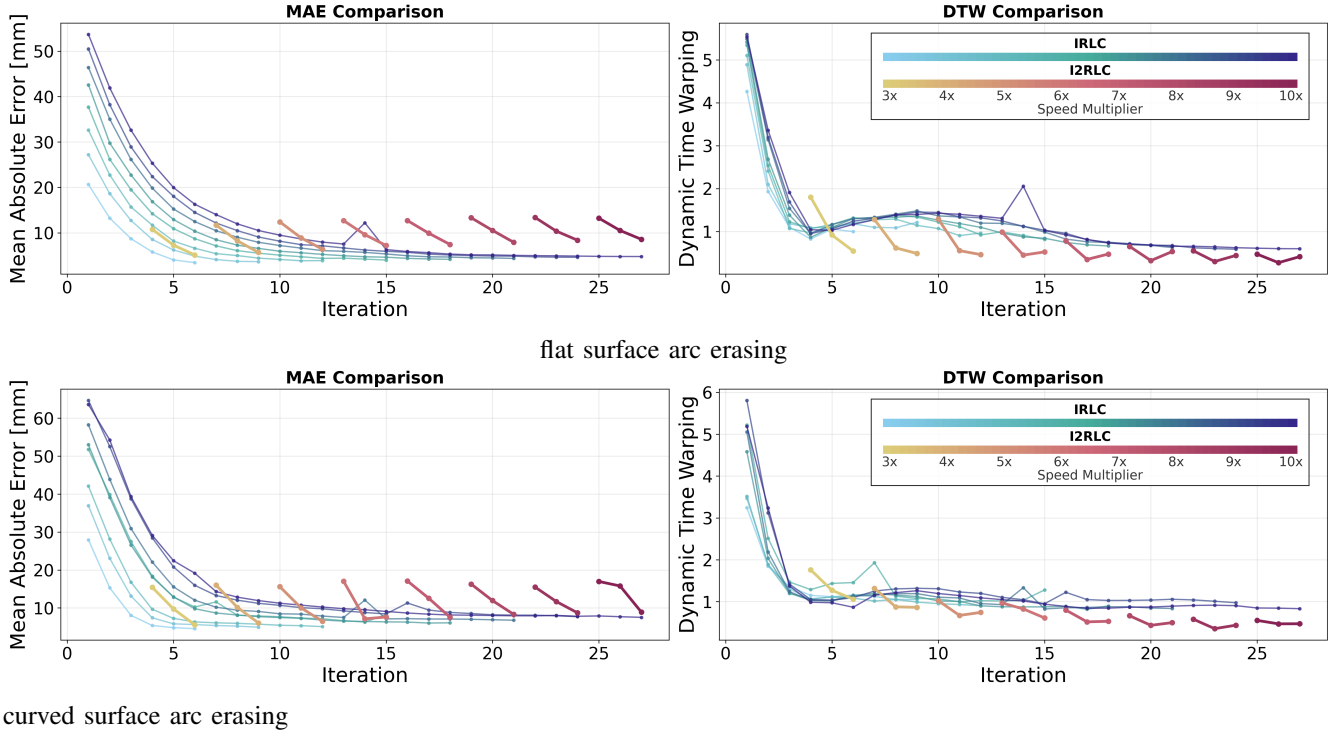


Fig. 9. Learning curves (MAE, left; DTW, right) for IRLC and I2RLC on the arc-erasing task. Line shade encodes speed (darker = higher speed multiplier). For fairness, the total number of update iterations is matched across methods. I2RLC exhibits smaller initial-iteration errors and lower DTW.

0.4. For fairness, we matched the total number of update iterations across methods; in I2RLC, the updated trajectory at one speed was carried forward to the next. We report two trajectory-tracking metrics: mean absolute error (MAE) and dynamic time warping (DTW). MAE is the time-averaged absolute pointwise position error—the same feedback signal used by IRLC. DTW quantifies spatial discrepancy between trajectories by allowing nonuniform temporal alignment.

*2) Results:* To visualize the effect of the proposed method, Fig. 6 presents snapshots from the flat and curved surface erasing tasks executed at the final iteration of 10x speed. A naively downsampled replay failed to erase most of the blue line (Fig. 6a). IRLC removed the majority of the line but left some residual segments (Fig. 6b). By contrast, I2RLC erased the blue line completely (Fig. 6c). Fig. 7 and 8 present 2D plots of the executed trajectories at the final iteration of the 10x speed stage. While IRLC and I2RLC follow similar paths, IRLC exhibits larger errors within the shaded regions, which correspond to the residual line segments visible in Fig. 6b.

For quantitative analysis, Table III reports tracking errors between the executed and downsampled trajectories at the final iteration of each speed. The naive playback baseline yielded the largest MAE and DTW. IRLC achieved the lowest MAE, whereas I2RLC achieved the lowest DTW, indicating better spatial alignment.

Sect. IV show learning curves for IRLC and I2RLC. Both methods reduce MAE and DTW over iterations. Under IRLC, the trajectory often deviated significantly in the initial iterations, producing large errors in both MAE and DTW. Although the task succeeded in our setting, such transients may saturate actuators or lead to failure in other tasks. By contrast, I2RLC maintained both errors within a narrower range throughout, including early iterations, substantially reducing the risk of unstable behavior before convergence.

At the final iteration, IRLC yielded a lower MAE. This is because I2RLC followed a slower target in the initial iteration, which introduces a temporal lag that inflates MAE. Conversely, I2RLC achieves a lower DTW, indicating better spatial fidelity to the demonstrated path. Starting from more distant initial references, IRLC can distort the trajectory; in the erasing tasks, this led to occasional departures from the target curve and residual marks.

In summary, both IRLC and I2RLC effectively refine accelerated trajectories, with I2RLC offering smaller errors in early iterations and better preservation of spatial similarity.

### *D. Imitation Learning*

*1) Setup:* For the flat surface arc erasing task, we collected two demonstrations for each of three arc patterns (six demonstrations total). We then generated accelerated playbacks up to 6x using I2RLC, increasing the speed incrementally and running three refinement iterations at each speed stage. Finally, we executed 10 playback rollouts per speed with Gaussian disturbances, yielding a 60-episode dataset. We trained ACT for 15000 epochs with a chunk size of 50 and an inference rate of 50 Hz.

At inference time, we evaluated the trained policy on the flat surface task at both seen and unseen arc positions. The arc position patterns are shown in Fig 10. We define the


Fig. 10. Line patterns of flat surface arc erasing task

TABLE III
ERASED-AREA RATE WITH IMITATION LEARNING↑[%].

| line pattern | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Trained (T) or Unseen (U) | U | T | U | T | U | T | U |
| ACT | 53±9 | 62±5 | 56±6 | 85±14 | 67±9 | 67±17 | 68±9 |

erased-area rate as the fraction of the initial blue-line area that is removed. From pre- and post-execution images, we segment the blue line to obtain binary masks; the removed area is computed from the mask difference and normalized by the initial mask area.

*2) Results:* Table III reports the erased-area rates measured from images. The learned policy executed faster than the original demonstration and erased over $50\%$ of the blue-line segments, but it left slightly more residual line area than direct playback. Two factors may contribute to this gap: 1) the output post-processing (Temporal Ensemble) used in our IL pipeline, which can introduce latency and damp rapid corrections; and 2) occlusions late in the task, when the end effector obscures the target line.

In summary, ACT with faster demonstrations achieved an erasure of over 50% of the blue-line segments. However, designing IL methods for high-speed manipulation remains an important direction for future work.

## V. CONCLUSION

We introduced *Incremental Iterative Reference Learning Control (I2RLC)*, a method for reliable high-speed replay of demonstrations in contact-rich imitation learning (IL) without additional human intervention. I2RLC incrementally increases execution speed while iteratively correcting the reference from observed tracking errors, enabling substantial speedups up to 10x with small tracking errors and preserved spatial fidelity. Combined with low-level compliance control, this approach applies naturally to contact-rich manipulation. The resulting refined trajectories are compatible with standard IL algorithms. A current limitation is the iteration cost required to achieve large speedups. Future work includes optimizing learning gains and speed-scheduling policies, as well as developing faster imitation learning algorithms using the refined accelerated demonstrations.

## REFERENCES


[1] S. An, Z. Meng, C. Tang, Y. Zhou, T. Liu, F. Ding, S. Zhang, Y. Mu, R. Song, W. Zhang *et al.*, "Dexterous manipulation through imitation learning: A survey," *arXiv preprint arXiv:2504.03515*, 2025.

[2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[3] T. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *Robotics: Science and Systems*, 2023.

[4] L. Peternel, T. Petrič, and J. Babič, "Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 1497–1502.

[5] K. Yamane, Y. Saigusa, S. Sakaino, and T. Tsuji, "Soft and rigid object grasping with cross-structure hand using bilateral control-based imitation learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1198–1205, 2023.

[6] M. Kobayashi, T. Buamanee, and T. Kobayashi, "Alpha-$\alpha$ and bi-act are all you need: Importance of position and force information/control for imitation learning of unimanual and bimanual robotic manipulation with low-cost system," *IEEE Access*, 2025.

[7] Y. Kanai, A. Kanazawa, H. Ichiwara, H. Ito, N. Noguchi, and T. Ogata, "Input-gated bilateral teleoperation: An easy-to-implement force feedback teleoperation method for low-cost hardware," *arXiv preprint arXiv:2509.08226*, 2025.

[8] S. Sujit, L. Nunziante, D. O. Lillrank, R. F. J. Dossa, and K. Arulkumaran, "Improving low-cost teleoperation: Augmenting gello with force," in *IEEE/SICE International Symposium on System Integration*, 2025, pp. 747–752.

[9] J. J. Liu, Y. Li, K. Shaw, T. Tao, R. Salakhutdinov, and D. Pathak, "Factr: Force-attending curriculum training for contact-rich policy learning," *arXiv preprint arXiv:2502.17432*, 2025.

[10] K. Inami, S. Sakaino, and T. Tsuji, "Motion retouch: Motion modification using four-channel bilateral control," in *IEEE International Conference on Mechatronics*, 2025, pp. 1–6.

[11] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 2074–2081.

[12] J. M. S. Ducaju, B. Olofsson, and R. Johansson, "Iterative reference learning for cartesian impedance control of robot manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 11 171–11 178.

[13] M. Iskandar, C. Ott, A. Albu-Schäffer, B. Siciliano, and A. Dietrich, "Hybrid force-impedance control for fast end-effector motions," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3931–3938, 2023.

[14] Y. Karako, S. Kawakami, K. Koyama, M. Shimojo, T. Senoo, and M. Ishikawa, "High-speed ring insertion by dynamic observable contact hand," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 2744–2750.

[15] K. Tanaka and M. Hamaya, "Twist snake: Plastic table-top cable-driven robotic arm with all motors located at the base link." in *IEEE International Conference on Robotics and Automation*, 2023, pp. 7345–7351.

[16] R. M. Hartisch and K. Haninger, "High-speed electrical connector assembly by structured compliance in a finray-effect gripper," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 2, pp. 810–819, 2023.

[17] Y. Yao, U. Yoo, J. Oh, C. G. Atkeson, and J. Ichnowski, "Soft robotic dynamic in-hand pen spinning," in *IEEE International Conference on Robotics and Automation*, 2025, pp. 1–7.

[18] C. C. Beltran-Hernandez, N. Erbetti, and M. Hamaya, "Sliceit!-a dual simulator framework for learning robot food slicing," in *IEEE International Conference on Robotics and Automation*, 2024, pp. 4296–4302.

[19] T. Kamijo, C. C. Beltran-Hernandez, and M. Hamaya, "Learning variable compliance control from a few demonstrations for bimanual robot with haptic feedback teleoperation system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 12 663–12 670.

[20] 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*, 2025, pp. 4829–4836.

[21] J. H. Kang, S. Joshi, R. Huang, and S. K. Gupta, "Robotic compliant object prying using diffusion policy guided by vision and force observations," *IEEE Robotics and Automation Letters*, 2025.

[22] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023.

[23] S. Sakaino, K. Fujimoto, Y. Saigusa, and T. Tsuji, "Imitation learning for variable speed contact motion for operation up to control bandwidth," *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, pp. 116–127, 2022.

[24] K. Yamamoto, H. Ito, H. Ichiwara, H. Mori, and T. Ogata, "Real-time motion generation and data augmentation for grasping moving objects with dynamic speed and position changes," in *IEEE/SICE International Symposium on System Integration*, 2024, pp. 390–397.

[25] L. Guo, Z. Xue, Z. Xu, and H. Xu, "Demospeedup: Accelerating visuomotor policies via entropy-guided demonstration acceleration," *arXiv preprint arXiv:2506.05064*, 2025.

[26] N. R. Arachchige, Z. Chen, W. Jung, W. C. Shin, R. Bansal, P. Barroso, Y. H. He, Y. C. Lin, B. Joffe, S. Kousik *et al.*, "Sail: Faster-than-demonstration execution of imitation learning policies," *arXiv preprint arXiv:2506.11948*, 2025.

[27] B. Kang, Z. Jie, and J. Feng, "Policy optimization with demonstrations," in *International Conference on Machine Learning*, 2018, pp. 2469–2478.

[28] L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal, "From imitation to refinement-residual rl for precise assembly," in *IEEE International Conference on Robotics and Automation*, 2025, pp. 01–08.

[29] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[30] S. Scherzinger, A. Roennau, and R. Dillmann, "Forward dynamics compliance control (FDCC): A new approach to cartesian compliance for robotic manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4568–4575.

[31] 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*, 2024, pp. 12 156–12 163.