Compliant Residual DAgger: Improving Real-World Contact-Rich Manipulation with Human Corrections

Xiaomeng Xu* **Yifan Hou*** **Zeyi Liu** **Shuran Song**

Stanford University

Abstract

We address key challenges in Dataset Aggregation (DAgger) for real-world contact-rich manipulation: how to collect informative human correction data and how to effectively update policies with this new data. We introduce Compliant Residual DAgger (CR-DAgger), which contains two novel components: 1) a Compliant Intervention Interface that leverages compliance control, allowing humans to provide gentle, accurate delta action corrections without interrupting the ongoing robot policy execution; and 2) a Compliant Residual Policy formulation that learns from human corrections while incorporating force feedback and force control. Our system significantly enhances performance on precise contact-rich manipulation tasks using minimal correction data, improving base policy success rates by over 60% on two challenging tasks (book flipping and belt assembly) while outperforming both retraining-from-scratch and finetuning approaches. Through extensive real-world experiments, we provide practical guidance for implementing effective DAgger in real-world robot learning tasks. Result videos are available at: https://compliant-residual-dagger.github.io/

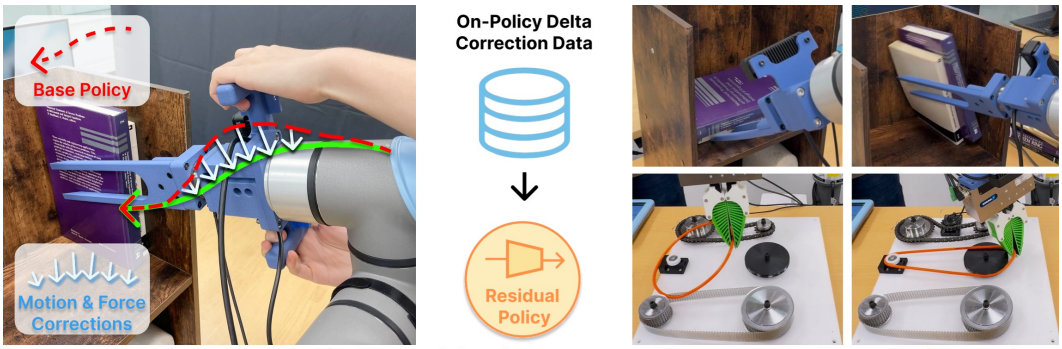

Figure 1: **CR-DAgger.** To improve a robot manipulation policy, we propose a compliant intervention interface (a) for collecting human correction data, and use this data to update a compliant residual policy (b), and thoroughly study their effects by deploying the updated policy on two contact-rich manipulation tasks in the real world (c).

1 Introduction

Learning from human demonstrations has seen many recent successes in real-world robotic tasks [9, 21, 41, 48, 42]. However, to obtain a successful policy, human demonstrators often have to repeatedly deploy a policy and observe its failure cases, then collect more data to update the policy until it

*Equal Contributions

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

succeeds. This process is broadly referred to as Dataset Aggregation (DAgger) [36, 23]. However, doing DAgger effectively for real-world robotic problems still faces the following challenges:

How to collect informative human correction data? DAgger is most effective when the correction data is within the original policy's induced state-action distribution [36]. In practice, the common approach is either (1) collecting offline demonstrations that cover the policy's typical failure scenarios [8], or (2) human taking over robot control during policy deployment [37, 32]. However, in both cases, the human demonstrator has no access to the original policy's behavior and may deviate excessively from it. Human taking over additionally introduces force discontinuity when they do not instantly reproduce the exact same robot force. This is partially due to the lack of effective correction interfaces that support precise and instantaneous intervention.

How to effectively update the policy with new data? Prior methods for improving a pretrained policy with additional data include (1) retraining the policy from scratch with the aggregated dataset [23], which can be computationally expensive; (2) finetuning the policy with only the additional data [41, 17, 7], which is sensitive to the quality of the new data [51], and (3) training a residual policy separately on top of the pretrained policy, which is typically done with Reinforcement Learning [2, 51] or Imitation Learning [5], both require a large number of samples.

In this work, we address these questions by proposing an improved system **Compliant Residual DAgger (CR-DAgger)** consisting of two critical components:

- **Compliant Intervention Interface.** We propose an on-policy correction system based on kinesthetic teaching to collect delta action *without interrupting the current robot policy*. Leveraging compliance control, the interface lets humans directly apply force to the robot and feel the magnitude of their instantaneous correction. Unlike take-over corrections, our design allows smooth transition between correction/no correction mode, while providing direct control of correction magnitudes.
- **Compliant Residual Policy.** Leveraging the force feedback from our Compliant Intervention Interface, we propose a residual policy formulation that takes in an *extra force modality* and predicts both residual motions and *target forces*, which can fully describe the human correction behavior. The Compliant Residual Policy is force-aware, even when the base policy is position-only. We show that our residual policy formulation *learns effective correction strategies* using the data collected from our Compliant Intervention Interface.

Together, our system significantly improves the success rate of precise contact-rich robot manipulation tasks using a small amount of additional data. We demonstrate the efficacy of our method on two challenging tasks involving long horizons and sequences of contacts: book flipping and belt assembly. We improve over the base policy success rate by over 60%, while also outperforming retrain-from-scratch and finetuning under the same data budgets. In summary, our contributions are:

- A **Compliant Intervention Interface**, a system that allows humans to provide accurate, gentle, and smooth corrections in both position and force to a running robot policy without interrupting it.
- A **Compliant Residual Policy**, a policy formulation that seamlessly integrates additional force modality inputs and predicts residual motions and forces.
- A **practical guide** for efficient DAgger based on extensive real-world studies for critical but often overlooked design choices, such as batch size and sampling strategy. Our hardware design, training data, and policy code can be found here.

2 Related Work

Human-in-the-Loop Corrections for Robot Policy Learning. The original DAgger work [36] requires the demonstrator to directly label actions generated by the policy. In robotics, a practical variation is to let the human *take over* the robot control and provide correct action directly [23]. Such human correction motion can be recorded with spacemouse [7, 29], joystick [37], smartphone [32], or arm-based teleoperation system [18, 19, 41]. We instead proposes a novel kinesthetic teaching system with compliance controller that allows the demonstrator to apply delta corrections while the robot policy is still running, and additionally records force feedback. Our results show that both the delta correction data and the force data are crucial to the success of the learned policy.

Improving Pretrained Robot Policies with New Data. The most direct approach to improving a pretrained policy with new correction data is to retrain the policy on the aggregated dataset, combining

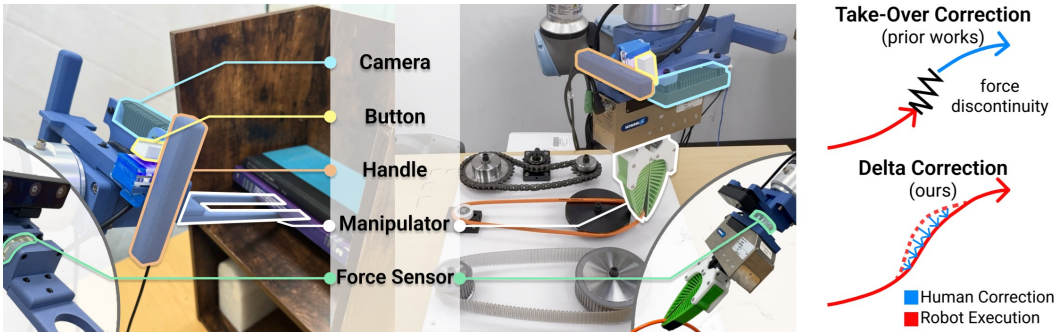

Figure 2: **Compliant Intervention Interface** characterized by a kinesthetic correction hardware setup where humans hold on the handle and apply forces to correct robot execution, providing on-policy delta corrections.

prior demonstrations with new feedback [32, 37]. Alternatively, Reinforcement Learning (RL) offers a framework to incorporate both offline and online data, either by warm-starting replay buffers [31, 3] or by using offline data to guide online fine-tuning [49, 43]. When policies are trained on large-scale human demonstration datasets [6, 38, 25, 34, 24, 44, 45], retraining becomes impractical, especially when the original data is inaccessible. In such cases, fine-tuning with only the new data is a common solution, using either imitation learning [29, 17, 41] or RL [33, 7]. Another line of work introduces an additional residual model on top of the original policy. These residual policies can be trained with RL in simulation [51, 2, 15], but suffers from sim-to-real challenges. Training residual policy in the real world usually requires a large number of samples [5, 22], intermediate scene representation [14], or consistent visual observations between training and testing [13, 16], making the approach hard to adopt in practice. In this work, we introduce a practical data collection system and an efficient residual policy learning algorithm for long-horizon, contact-rich manipulation tasks. Our approach requires only a small amount of real-world correction data and supports integration of additional sensory modalities not present in the original model, leading to improved policy performance.

3 CR-DAgger Method

Our goal is to improve a pretrained robot policy with a small amount of human correction data. To achieve this, we propose a Compliant Intervention Interface (§ 3.1) that enables precise and intuitive on-policy human correction data collection, and a Compliant Residual Policy (§ 3.2) that efficiently learns the correction behaviors to be used on top of the pretrained policy. Throughout the paper, we use the term *base policy* to refer to the pretrained policy without online improvements.

3.1 Compliant Intervention Interface

Correction data is collected by human demonstrators to rectify policy failures. Unlike initial demonstrations that establish baseline behaviors, correction data specifically targets failure modes observed during policy deployment. Correction data is most effective when it corrects failures in policy-induced state distributions [36]. The interface through which these corrections are collected significantly impacts the quality of correction data, which should be intuitive for demonstrators, capture critical corrective information at precise moments of failures, and facilitates collecting data close to the base policy state-action distribution.

There are two types of correction collection methods: *Off-policy correction* is when humans observe failures of the base policy during deployment, and then recollect extra offline demonstrations to address failure cases. This approach is most commonly used for improving Behavior Cloning policy performance due to its simplicity - it requires *no additional infrastructure* beyond the original data collection setup. However, the resulting demonstrations may fail to cover all the failure cases or deviate from the original state-action distribution. We focus on *on-policy correction* instead, where humans can monitor policy execution and intervene on the spot when failures occur or are anticipated. This approach allows humans to provide corrections more targeted to the base policy's failure cases. However, challenges still exist for an intervention system:

- **Non-smooth transitions**. Intervention in robotics is typically implemented by *take-over* correction: letting human take complete control and overwrite robot policy. As the underlying control abruptly switches between robot policy and human intention, disturbances are introduced due to policy

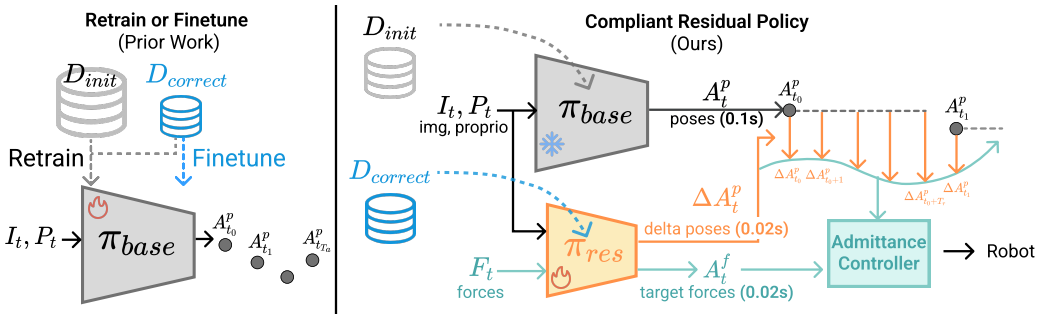

Figure 3: **Policy Update Methods.** Left: Common policy update methods - retraining and finetuning. Right: Ours. The base policy runs at 1 Hz. It takes in images I_t and proprioceptions P_t and predicts 32 frames of end-effector poses $A_t^p = \{A_{t_0}^p, A_{t_1}^p, ..., A_{t_{T_a}}^p\}$ spaced 0.1 Seconds apart. The Compliant Residual Policy runs at 50 Hz. It takes in additional force inputs F_t and predicts 5 frames of delta poses $\Delta A_t^p = \{\Delta A_{t_0}^p, \Delta A_{t_0+1}^p, ..., \Delta A_{t_0+T_r}^p\}$ and target forces A_t^f spaced 0.02 Seconds apart. The combined poses of A_t^p and ΔA_t^p, and target forces A_t^f are taken by an admittance controller to command the robot.

inference and human response latency, especially when the robot is withholding external forces. The recorded data thus may include undesired actions that do not reflect the human's intention.

- **Distribution shift.** The human-intervened state-action distribution may deviate significantly from the original distribution. Additionally, the non-smooth transition above could bring in disturbances and add to the distribution shift.
- **Indirect correction brings errors.** Correction is commonly implemented via teleoperation interfaces such as spacemouse or joysticks [17, 7]. With spatial mismatch and teleoperation latency, it is hard for the demonstrator to instantly provide accurate corrections upon intervention starts without going through a short adjustment period.
- **Missing information.** The recorded correction data need to fully describe the human's intended action. Simply recording the robot's position is not sufficient, since it may be under the influence of human correction force and will cause different result when testing without human.

We propose a *Compliant Intervention Interface* with the following designs to solve those challenges:

- **Delta correction instead of take-over correction.** Unlike take-over correction, where the demonstrator has no idea of the policy's original intention once taking over, we propose a novel on-policy delta correction method: we let the robot policy executes continuously while the human applies forces to the robot with a handle mounted on the end effector, resulting in delta actions on top of the policy action. The human demonstrator can always sense the policy's intention through haptic feedback, and easily control the magnitude of intervention by the amount of force applied to the handle. As a result, delta correction ensures smooth intervention data and limits the human from providing very large corrections. The approach is also intuitive as the human can directly move the robot towards desired correction directions.
- **Correction interface with compliance control.** In order to apply delta correction over a running policy, we provide a compliant interface that allows humans to safely intervene and apply force to the robot to affect its behaviors at any time, as shown in Fig. 2. We design a kinesthetic correction hardware setup with a detachable handle for human to hold when correcting, and allows easy tool-swapping for different tasks. We run a compliance controller (specifically admittance control) in the background to respond to both contact forces and human correction forces, allowing the human to influence but not completely override the policy execution. The admittance controller uses a constant stiffness $\sim$1000 N/m to allow easy human intervention and ensure accurate tracking.
- **Correction recording with buttons and force sensor.** Our interface additionally includes an ATI 6-D force sensor to directly measure contact forces, and a single-key keyboard placed on the handle to record the exact timings of correction starts/ends. Both the policy's original commands and the human's delta corrections are recorded, along with force sensor readings during the interaction.

3.2 Compliant Residual Policy

Given the correction data, there are multiple ways to update the policy. Common practices include *retraining* the base policy from scratch with both initial data and correction data, and *finetuning* the

base policy with only the correction data. However, *retraining* is costly as it requires updating the entire base policy network from scratch with all the available data. It also requires access to the base policy's initial training data, which might not be accessible for many open source pretrained models. The amount of correction data is significantly smaller than the initial training data, thus simply mixing them together makes the policy hard to gain effective corrective behaviors. While *finetuning* allows updating partial policy network parameters with new data only, its training stability can be easily affected by the distribution mismatch between the correction data and initial training data. Moreover, both retraining and finetuning can only update the policy with its fixed network architecture while being unable to incorporate new inputs and outputs. We propose a compliant residual policy trained only on the correction data to refine base policy's position actions and predict additional force actions.

Compliant residual policy formulation. Our policy directly learns corrective behavior from the human delta correction data, as shown in Fig. 3. It takes as input the same visual and proprioceptive feedback as the base policy but with a shorter horizon. It also takes in an extra force modality, which is available using our Compliant Intervention Interface. The policy outputs five frames of actions at a time, corresponding to 0.1 s of execution time when running at 50 Hz. The action space is 15-dimensional: the first nine dimensions represent the SE3 delta pose from the base policy action to the robot pose command [8], while the later six dimensions represent the expected wrench (force and torque) the robot should feel from external contacts. Both the robot pose command and the expected wrench are sent to a standard admittance controller for execution with compliance.

The residual policy directly uses the base policy's frozen image encoder [35, 1, 46] to extract an image embedding, a temporal convolution network [39] to encode the force vectors, followed by fully-connected layers to decode actions.

Advantages. This formulation provides the following advantages:

- *Sample-efficient learning.* The residual policy's network is light-weight ($\sim$2MB trainable weights) and only requires a small amount of correction data to train (50$\sim$100 demonstrations).
- *Incorporating new sensor modality.* Compared to retraining and finetuning methods that are limited to the base policy's network architecture, residual policy can incorporate new sensor modality. This allows taking any position-based pretrained policy and turning it force-aware simply by collecting a small amount of correction data with force modality.
- *High-frequency inference.* The light-weight residual policy runs at a higher frequency than the base policy, incorporating high-frequency force feedback and enabling reactive corrective behaviors. This reactivity is particular important for error correction during contact events.

Training strategy. In prior work, a residual policy is trained either in simulation with RL [2, 51] to give it sufficient coverage of the state distribution, or in the real world with pre-collected behavior cloning data [31]. In this work, we train the Compliant Residual Policy completely on the small amount of new real-world correction data with the following strategies:

- *Ensure sufficient coverage of in-distribution data.* Human correction tends to be frequent around a few key moments of the task. A residual trained on correction data alone can extrapolate badly around states where no correction is provided. To help the residual policy understand when *not* to provide corrections, we: (1) include the no correction data for training but label it as zero residual actions; (2) collect a few trajectories where the demonstrator always holds the handle and marks the whole trajectory as correction even when the correction is small or zero. Details are in § A.3.
- *Prioritize correction data over no-correction (zero residual action) data.* Similar to [29], we alter the data sample frequency during training based on whether they have human correction or not. Specifically, since the moment of correction start indicates where the current policy performs badly followed by immediate action to fix it, we sample data more frequently for a short period immediately after correction starts. Our real-world ablations (§ 4.5) demonstrate that our training strategies improve the quality of the residual policy and the overall success rate.

4 Evaluation

For each task, we train a diffusion policy [8] with 150$\sim$400 demonstrations as the base policy. We first deploy the base policy and observe its performance and failure modes. Next, from the same base policy, we collect 50$\sim$100 correction episodes with each data collection method. Then, we update the policy using each network updating method and training procedure. Finally, we deploy the updated policies and evaluate their performance under the same test cases. Details of tasks and comparisons are described below.

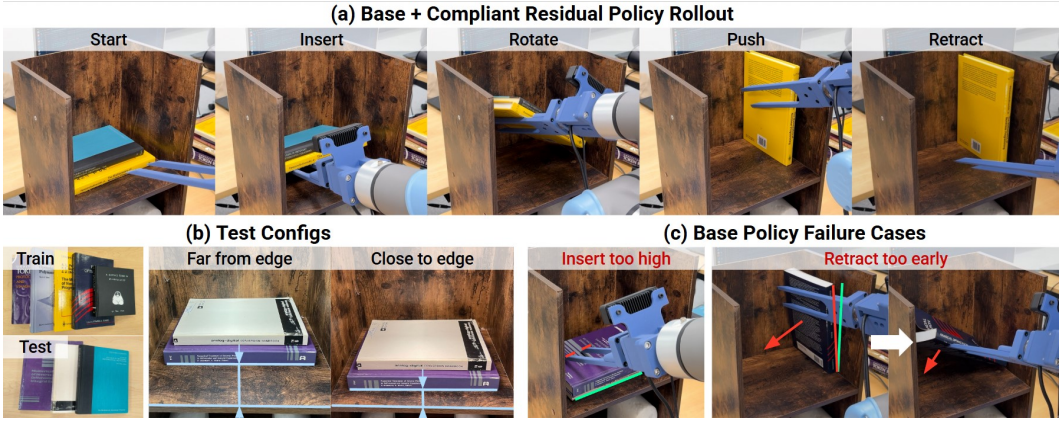

Figure 4: **Book Flipping Task.** (a) Policy rollout of [Compliant Residual] policy trained with [On-Policy Delta] data, demonstrating accurate insertion motions and forceful pushing strategy. (b) Different test scenarios. (c) Typical failure cases of the base policy: inserting too high above the book and missing the gap; retracting the fingers before the books can steadily stand.

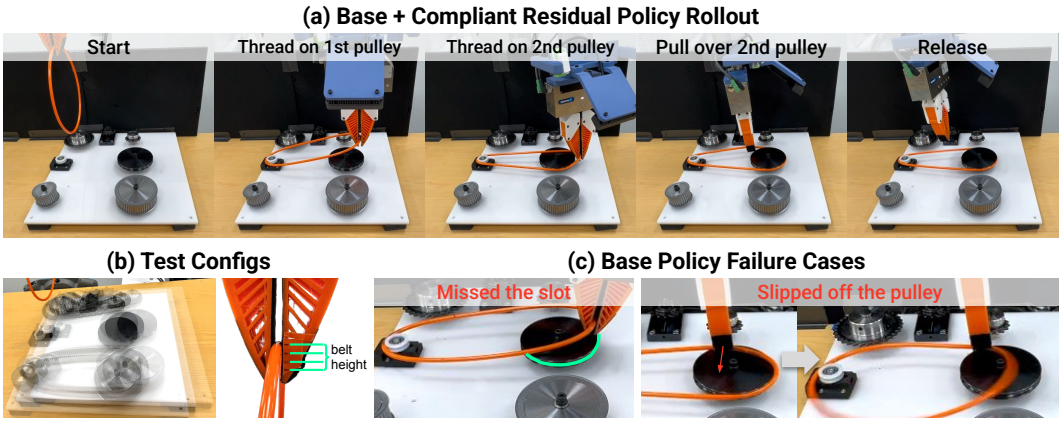

Figure 5: **Belt Assembly Task.** (a) Policy rollout of [Compliant Residual] policy trained with [On-Policy Delta] data, demonstrating accurate force-position coordination and adaptation. (b) Different test scenarios. (c) Typical failure cases of the base policy: missing the slot by going too high above the pulley; tilting the belt and causing it to slip off the pulley.

4.1 Contact-Rich Manipulation Tasks

Book Flipping: As shown in Fig. 4 (a), this task is to flip up books on a shelf. Starting with one or more books lying flat on the shelf, the robot first insert fingers below the book, then rotate the book up and push them firmly against the shelf to let them stand on their own. The base policy is trained with 150 demonstrations.

This task is challenging as it involves rich use of physical contacts and forceful strategies [20]. A position-only strategy always fails immediately by triggering large forces, so we execute all policies through the same admittance controller. The task success requires high precision in both motion and force to accurately align the fingers with the gap upon insertion, and to provide enough force to rotate heavy books and make the books stand firmly.

Each evaluation includes 20 rollouts on 4 test cases (5 rollouts each), as shown in Fig. 4 (b): 1) flipping a single book (three seen and two unseen books), initially far from the shelf edge; 2) flipping a single book close to the shelf edge; 3) flipping two books together (combinations of three seen and three unseen books), initially far from the shelf edge; 4) flipping two books close to the shelf edge. We use the same initial configurations for all evaluations.

Belt Assembly: As shown in Fig. 5 (a), this task is to assemble a thin belt onto two pulleys, which is part of the NIST board assembly challenge [26]. Starting with the belt grasped by the gripper, the robot needs to first thread the belt over the small pulley, next move down while stretching the belt to

thread its other side on the big pulley, then rotate 180° around the big pulley to tighten the belt, and finally pull up to release the belt from the gripper. The base policy is trained with 405 demonstrations.

The task is challenging as it requires both position and force accuracy throughout the process. Specifically, the belt is thin and soft so the initial alignments onto the pulleys are visually ambiguous. Also, since the belt is not stretchable, there is more resistance force and less position tolerance as the belt approaches the second pulley, requiring a policy with good force-position coordination and adaptation. We ran 32 rollouts across four different initial board positions and four grasp locations (Fig. 5 (b)).

4.2 Base Policy and its Failure Modes

We trained a diffusion policy [8] that takes in past images from a wrist-mounted camera and robot proprioception observations, and predicts a future position-based action trajectory. To isolate the contribution of force inputs versus human corrections, we trained diffusion policies with and without force inputs as baselines for the belt assembly task.

The book flipping base policy achieves a 40% success rate with the following common failure cases (Fig. 4 (c)): (1) Missed insertion. The fingers initially go too high above the book or aims for the gap between the two books, failing to properly insert beneath the books. (2) Incomplete flipping. At the last stage, the policy retracts the blade before the book can stand stably, causing it to fall back.

The belt assembly base policy achieves a 15.6% success rate. Adding force input increases the base policy success rate to 43.8%. Common failure cases include (Fig. 5 (c)): (1) Missed slotting: the fingertip goes too high or too low, causing the belt to miss the slot on the big pulley. (2) Belt slippage: the fingers pull the belt in the wrong direction, causing the belt to tilt and slip off the pulley.

4.3 Comparisons

We compare CR-DAgger with baselines across two dimensions: correction method and policy update method. We present the quantitative results in Fig. 6, and explain key findings in § 4.4.

Correction data collection methods. We compare our Compliant Intervention Interface with the two most commonly used correction data collection strategies:

- *Observe-then-Collect* includes two steps: first, the policy is deployed and human demonstrators observe the initial settings that could cause failures; then, demonstrators provide completely new demonstrations starting from similar initial settings. As explained in § 3.1, this type of offline correction potentially misses critical timing information, and the resulting demonstrations may deviate from the policy's original behavior distribution.
- *Take-over-Correction* (HG-DAgger) [23] is another common correction strategy where human demonstrators monitor policy execution and take complete control when failures are anticipated. We implement Take-over-Correction on our Compliant Intervention Interface by cleaning up command buffer to the compliance controller and switching stiffness to zero upon correction starts, so the robot policy does not affect the robot during correction. However, as explained in § 3.1, take-over correction introduces an abrupt transition around control authority switching, which may cause distributional discontinuities in the training data.
- *On-Policy Delta (Ours):* the details are described in § 3.1.

Policy update methods. We compare with two common policy update methods:

- *Retrain Policy:* Retrain the base policy using both the original training data and the correction data from scratch. As explained in § 3.2, this approach is reliable but may require access to the orignal data and large amount of new data to work well.
- *Finetune Policy:* Finetune the base policy using only the correction data (freezing visual encoders). As explained in § 3.2, this approach can be sensitive to data quality and distribution shifts.
- *Fintune Policy with KL Regularization:* A recent method [11] that stabilizes finetuning training by encouraging the predicted action to be close to the training data distribution.
- *Residual Policy:* an ablation of our method where force is removed from both input and outputs.
- *Compliant Residual Policy (Ours):* Residual policy update with additional force input and outputs, see details in § 3.2.

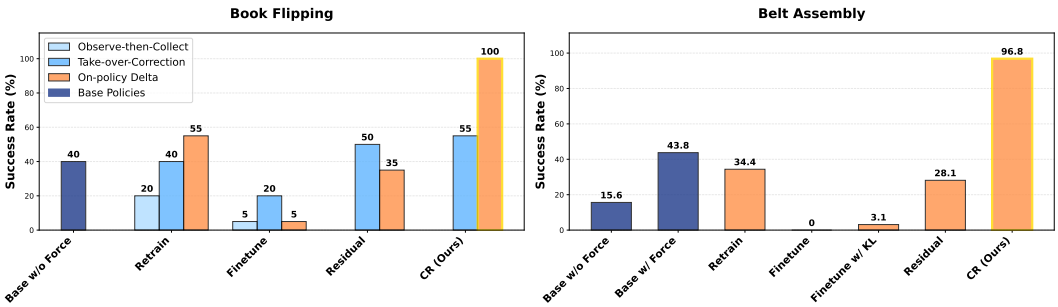

Figure 6: **Results.** We compare CR-DAgger across two dimensions: correction method and policy update method. The result shows that our [Compliant Residual (CR)] policy trained with [On-Policy Delta] data is able to improve upon base policies on both tasks and outperforms other variations.

4.4 Key Findings

Finding 1: Compliant Residual Policy is able to improve base policy by a large margin. As shown in Fig. 6, [Compliant Residual] policy trained with [On-Policy Delta] data improves the base policy success rate by 60% and 50% on the two tasks respectively. It effectively learns useful corrective strategies from the limited demonstrations. For example, in the book flipping task, the policy learns to pitch the fingers down more before finger insertion to increase the insertion success; in the belt assembly task, the policy learns to correct the height of the belt when misaligning to the pulley slot. Results are best viewed in our supplementary video.

Finding 2: Residual policy allows additional useful modality to be added during correction. [Compliant Residual] policy performs significantly better than other methods without force (45% higher success rate than the best position-only baseline on the book task and 53% higher on the belt task) as it can both take in force feedback that indicates critical task information and predict adequate contact forces to apply. For example, the last stage of the book flipping task requires the robot to firmly push the book against the shelf wall to let it stand on its own. [Compliant Residual] policy predicts large pushing forces at this stage to make the books stand stably with a 100% success rate, while [Residual]'s success rate drops from 70% to 35% (§ A.2). The second stage of the belt assembly task (threading the belt on the large pulley) requires delicate belt height adjustments under ambiguous visual information due to occlusions and the lack of depth. [Compliant Residual] policy learns to move along the pulley to find the slot when the finger touches the top of the pulley.

Finding 3: Smooth On-Policy Delta data enables stable residual policy. [Compliant Residual] policy has 45% higher success rate when trained on [On-Policy Delta] data instead of [Take-over-Correction] data on the book flipping task. Residual policy trained with [Take-over-Correction] data sometimes exhibits large noisy motions that trigger task failures, such as retracting the fingers too early in the book flipping task. On the contrary, residual policy trained with [On-policy Delta] data have much smoother action trajectories and better reflect human's correction intentions, providing suitable magnitudes of corrections.

Finding 4: Retraining base policy is stable but learns correction behavior slowly. Retraining from scratch with the initial and correction data together leads to policies with stable motions. However, its behavior is less affected by the small amount of correction data compared to the dominant portion of initial data, leading to insignificant improvements over the base policy (1.67% success rate drop on the book task averaged across all data collection methods, 18.8% success rate improve on the belt task, both are much less improvements than [Compliant Residual]).

Finding 5: Finetuning base policy is unstable. Policy finetuning with either correction data has the worst performance across all policy update methods and even underperforms the base policy (30% success rate drop on the book task averaged across all data collection methods, 15.6% drop on the belt task). The finetuned policy predicts unstable and noisy motions, quickly leading to out-of-distribution states, such as inserting too high in the book flipping task and drifting away from the board in the belt assembly task. This is likely due to the distribution mismatch between the base policy training data and correction data, causing training instabilities. Adding KL-regularization effectively reduced the noisy behavior, however, the overall success rate is still lower than other baselines.

4.5 Ablations

We study two important design decisions with ablation studies on the book flipping task.

Training frequency and batch size. One important parameter in DAgger is the batch size between policy updates. With a smaller batch size, the policy is updated more frequently, then new correction data can better reflect the most recent policy behavior. However, DAgger with small batch sizes is known to suffer from *catastrophic forgetting* [27, 12] since it finetunes neural networks on data with non-stationary distribution. Common solutions include retraining the residual policy at the end of DAgger using all available correction data collected from all the intermediate residual policies [41]. Another way is to rely on the base policy training data as a normalizer [17]. In this work, we empirically found that larger batch sizes can effectively stabilize residual training. With batch size = 50, the book flipping task reach 100% with one batch, while the belt assembly task performs better with two batches. We compare our method with a smaller batch size on the book flipping task, where we warm up the residual with 20 episodes of initial correction data, then update every ten more episodes for three times.

Finding: Large-batch DAgger is more suitable for training Compliant Residual Policy. The small-batch training becomes unstable and the demonstrator needs to provide large magnitudes of corrections as the number of iterations increases. During evaluation, the final policy always fails by inserting too high, while our single-batch policy achieves a 100% success rate with the same amount of data and training epochs.

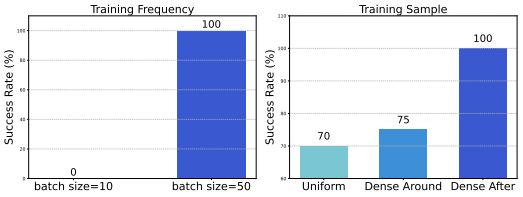

Figure 7: **Effect of Training Frequency and Sample.** Batch size=50 leads to more stable training and dense sampling after correction starts achieves better performance on the book flipping task.

Sampling strategy during training. The start of a human intervention contains critical information of the timing and direction of correction. Accurate delta action predictions right after correction starts are important for reactive corrective behaviors. We investigate three strategies for sampling from online correction data during training: 1. Uniform sample, where the whole episode is sampled uniformly. 2. Denser sample around the start of a human intervention, and 3. denser sample only after the human intervention starts. For 2 and 3, we uniformly increase the sample frequency four times for a fixed period before and/or after intervention starts.

Finding: Sampling denser right after intervention starts leads to more reactive and accurate corrections. As shown in Fig. 7 (right), the best performance comes from densely sampling after the beginning of interventions. Sampling denser around the start of a human intervention also adds more samples right before the intervention starts, which is where humans observe signs of failures. These are mostly negative data, and using them for training decreases the policy success rate.

5 Conclusion and Discussion

In this work, we evaluate practical design choices for DAgger in real-world robot learning, and provide a system, CR-DAgger, to effectively collect human correction data with a Compliant Intervention Interface and improve the base policy with a Compliant Residual Policy. We demonstrate the effectiveness of our designs by comparing them with a variety of alternatives on four contact-rich manipulation tasks.

Limitations and Future Work.

The base policy should have a reasonable success rate for the residual policy to learn effectively. From our experiments, we recommend starting to collect correction data for the residual policy when the base policy has at least 10% $\sim$ 20% success rate. A future direction is to derive theoretical guidelines for the trade-off between the base and residual improvements.

Throughout this work, we use a MLP as the action head of our Compliant Residual Policy and directly regress the actions. Although it works well in our tasks, it may experience difficulty for tasks that involve more distinctive action multi-modalities. More expressive policy formulations, such as Flow Matching [28, 6] might be useful for these tasks.

Our data collection system is based on kinesthetic teaching. Although it provides richer data with higher quality than teleoperation as explained in the paper, it may require more labor during training data collection since the demonstrator needs to grasp the handle on the robot.

6 Acknowledgment

We would like to thank Chendong Xin for his support on experiments and rebuttal. We would like to also thank Eric Cousineau, Huy Ha, and Benjamin Burchfiel for thoughtful discussions on the proposed method, thank Mandi Zhao, Maximillian Du, Calvin Luo, Mengda Xu, and all REALab members for their suggestions on the experiment setup and the manuscript. This work was supported in part by the NSF Award #2143601, #2037101, and #2132519, Sloan Fellowship, Toyota Research Institute, and Samsung. We would like to thank Google and TRI for the UR5 robot hardware. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

[2] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement–residual rl for precise assembly. *arXiv preprint arXiv:2407.16677*, 2024.

[3] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

[4] Maria Bauza, Antonia Bronars, Yifan Hou, Ian Taylor, Nikhil Chavan-Dafle, and Alberto Rodriguez. Simple, a visuotactile method learned in simulation to precisely pick, localize, regrasp, and place objects. *Science Robotics*, 9(91):eadi8808, 2024.

[5] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[7] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

[8] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[9] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[10] Hojung Choi, Jun En Low, Tae Myung Huh, Gabriela A Uribe, Seongheon Hong, Kenneth AW Hoffman, Julia Di, Tony G Chen, Andrew A Stanley, and Mark R Cutkosky. Coinft: A coin-sized, capacitive 6-axis force torque sensor for robotic applications. *arXiv preprint arXiv:2503.19225*, 2025.

[11] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

[12] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[13] Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. See to touch: Learning tactile dexterity through visual incentives. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13825–13832. IEEE, 2024.

[14] Irmak Guzey, Yinlong Dai, Georgy Savva, Raunaq Bhirangi, and Lerrel Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. *arXiv preprint arXiv:2410.23289*, 2024.

[15] Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and match: Supercharging imitation with regularized optimal transport. In *Conference on Robot Learning*, pages 32–43. PMLR, 2023.

[16] Siddhant Haldar, Jyothish Pari, Anant Rai, and Lerrel Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *arXiv preprint arXiv:2303.01497*, 2023.

[17] Zhanpeng He, Yifeng Cao, and Matei Ciocarlie. Uncertainty comes for free: Human-in-the-loop policies with diffusion models. *arXiv preprint arXiv:2503.01876*, 2025.

[18] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

[19] Ryan Hoque, Ashwin Balakrishna, Carl Putterman, Michael Luo, Daniel S Brown, Daniel Seita, Brijen Thananjeyan, Ellen Novoseller, and Ken Goldberg. Lazydagger: Reducing context switching in interactive imitation learning. In *2021 IEEE 17th international conference on automation science and engineering (case)*, pages 502–509. IEEE, 2021.

[20] Yifan Hou, Zhenzhong Jia, and Matthew T Mason. Manipulation with shared grasping. *arXiv preprint arXiv:2006.02996*, 2020.

[21] Yifan Hou, Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4829–4836, 2025. doi: 10.1109/ICRA55743.2025.11128452.

[22] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[23] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[24] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[25] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[26] Kenneth Kimble, Karl Van Wyk, Joe Falco, Elena Messina, Yu Sun, Mizuho Shibata, Wataru Uemura, and Yasuyoshi Yokokohji. Benchmarking protocols for evaluating small parts robotic assembly systems. *IEEE robotics and automation letters*, 5(2):883–889, 2020.

[27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[29] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, page 02783649241273901, 2022.

[30] Yun Liu, Xiaomeng Xu, Weihang Chen, Haocheng Yuan, He Wang, Jing Xu, Rui Chen, and Li Yi. Enhancing generalizable 6d pose tracking of an in-hand object with tactile sensing. *IEEE Robotics and Automation Letters*, 9(2):1106–1113, 2023.

[31] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024.

[32] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

[33] Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy agnostic rl: Offline rl and online rl fine-tuning of any class and backbone. *arXiv preprint arXiv:2412.06685*, 2024.

[34] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[36] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[37] Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from interventions: Human-robot interaction as both explicit and implicit feedback. In *16th robotics: science and systems, RSS 2020*. MIT Press Journals, 2020.

[38] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[39] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016.

[40] Luigi Villani and Joris De Schutter. Force control. In *Springer handbook of robotics*, pages 195–220. Springer, 2016.

[41] Philipp Wu, Yide Shentu, Qiayuan Liao, Ding Jin, Menglong Guo, Koushil Sreenath, Xingyu Lin, and Pieter Abbeel. Robocopilot: Human-in-the-loop interactive imitation learning for robot manipulation. *arXiv preprint arXiv:2503.07771*, 2025.

[42] Haoyu Xiong, Xiaomeng Xu, Jimmy Wu, Yifan Hou, Jeannette Bohg, and Shuran Song. Vision in action: Learning active perception from human demonstrations. *arXiv preprint arXiv:2506.15666*, 2025.

[43] Mengda Xu, Manuela Veloso, and Shuran Song. Aspire: Adaptive skill priors for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:38600–38613, 2022.

[44] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *Conference on robot learning*, pages 3536–3555. PMLR, 2023.

[45] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

[46] Xiaomeng Xu, Yanchao Yang, Kaichun Mo, Boxiao Pan, Li Yi, and Leonidas Guibas. Jacobinerf: Nerf shaping with mutual information gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16498–16507, 2023.

[47] Xiaomeng Xu, Huy Ha, and Shuran Song. Dynamics-guided diffusion model for robot manipulator design. *arXiv preprint arXiv:2402.15038*, 2024.

[48] Xiaomeng Xu, Dominik Bauer, and Shuran Song. Robopanoptes: The all-seeing robot with whole-body dexterity. *arXiv preprint arXiv:2501.05420*, 2025.

[49] Patrick Yin, Tyler Westenbroek, Simran Bagaria, Kevin Huang, Ching-an Cheng, Andrey Kobolov, and Abhishek Gupta. Rapidly adapting policies to the real world via simulation-guided fine-tuning. *arXiv preprint arXiv:2502.02705*, 2025.

[50] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[51] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. *arXiv preprint arXiv:2412.13630*, 2024.

A Technical Appendices and Supplementary Material

A.1 Control Implementation Details

A.1.1 Compliance Control

Compliance is a motor property that describes how motion responds to force. For example, a balloon has high compliance, meaning that they deform significantly under external forces. Industrial robots typically have very low compliance, meaning that their motion tracking has little deviation under external forces.

Compliance control refers to the technique that makes a rigid robot behave softly using feedback control. It is a standard technique widely adopted in the manufacturing industry. It lets robots interact with the environment safely without creating huge forces. With compliance, the robot retreats when it experiences external forces, larger force leads to bigger position deviations. When there is no external force, compliance control tracks the target position accurately just like position-based control. Compliance control allows specification of the exact desired compliance profile, typically described by parameters such as stiffness, damping and inertia.

Impedance control and admittance control are two methods to implement compliance control. The users are free to choose the controller that works with their robot. Robots with good position control accuracy (e.g., most industrial robots) can use admittance control. Robots with low gear ratio (e.g. "quasi-direct-drive" robots) can use impedance control. We use admittance control in this work and open-sourced our implementation at https://github.com/yifan-hou/force_control. A good reference for different compliance control schemes is the "Force Control" chapter of the Handbook of Robotics [40].

A.1.2 Control Architecture

Our software system consists of three independent loops:

1. **The base policy loop** that runs the diffusion policy. The base policy loop updates at about 1Hz, each time predicts 32 frames of actions corresponding to 3.2s of future robot positions. In this work we have not optimized the implementation for computation speed.
2. **The residual policy loop** that runs at approximately 50Hz, each time predicts five frames of delta actions corresponding to 0.1s of delta positions. The delta actions are added to the corresponding base policy actions based one time, before being sent to the low-level compliance controller for execution. The loop rate is limited by our current implementation and can be improved if needed.
3. **The admittance controller loop** that runs at exactly 500Hz. This loop implements 6D Cartesian compliance on the robot. It takes reference positions and forces from the residual policy. When there is zero reference force and no external force, the admittance controller lets the robot track the reference position precisely. When external force exists, the robot will deviate from the reference position like a spring centered on the base policy output position.

Apart from the above controller/policy loops, each hardware (e.g. camera, force-torque sensor) has a standalone driver loop maintaining 1. communication with the hardware, and 2. buffers for action and feedback for this hardware.

A.2 Stage-Wise Success Rate

We report the success rate of the book flipping task into three key stages.

A.3 Correction Data Decomposition

As mentioned in the "Training strategy" part of § 3.2, we used two strategies to ensure the residual policy behaves stably around low correction data regions. The first strategy is to include the no correction portion of online data for training and label them with zero residual actions. The second strategy is to collect a few trajectories (15 out of the 50 total correction episodes) in which the

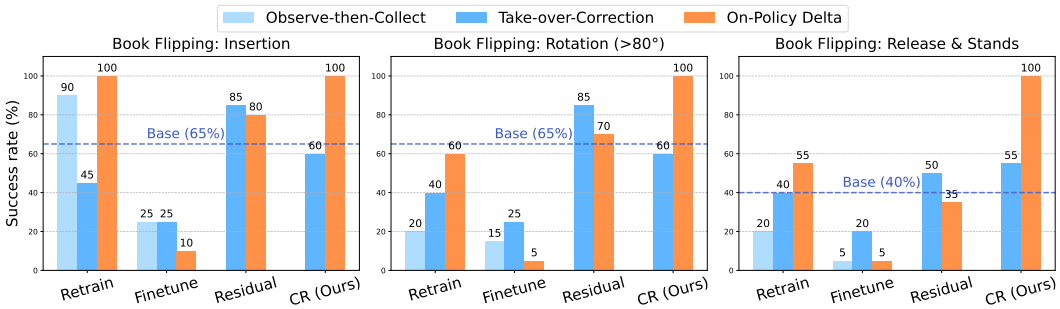

Figure 8: **Stage-wise Success rates.** Each row represents the results for one task, while each column shows the success rate up to the corresponding stage.

demonstrator marks the whole trajectory as correction, even when the correction is small or zero. In practice, we find that the first strategy works better when the base policy is more stable and has a higher success rate, while the second strategy works better otherwise. In our experiments, we use the first strategy for the book flipping task and use the second strategy for the belt assembly task.

A.4 Experiments Compute Resources

We use a desktop with a NVIDIA GeForce RTX 4090 GPU for training and deployment.

A.5 Hardware Design

Our kinesthetic correction hardware setup features a tool interface that allows task-specific tool swapping. For the book flipping task, we designed a customized fork-shaped tool that can easily insert under the books and flip them. For the belt assembly task, we used a standard WSG-50 gripper and fin-ray fingers [9]. An interesting future direction is to leverage generative models for automatic manipulator design [47]. Future work can also incorporate other types of force or tactile sensors, such as capacitive F/T sensors [10] and vision-based tactile sensors [50, 30, 4].

A.6 Broader Impact

CR-DAgger contributes to the field of robotics by improving pretrained real-world manipulation policies with a small amount of human correction data. The proposed Compliant Intervention Interface provides an intuitive and safe way for humans to directly interact with robots and correct the robot policy on the spot. We demonstrate significant policy improvements on two real-world contact-rich manipulation tasks, book flipping and belt assembly, which can lead to useful applications in industry and households.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a novel system for real-world DAgger, and we explain the method in detail (§ 3) and carry out extensive real-world experiments (§ 4).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in § 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We explain our algorithm in detail in § 3 and our experiment setups in § 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We submit code and data in supplemental material. Our code base is also open-sourced at https://github.com/yifan-hou/cr-dagger.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The full details are provided with the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report success rates over 20 to 32 real-world rollouts across various test cases for all evaluated methods (§ 4.4). For robotics experiments involving physical hardware in real-world settings, we chose to report aggregate success rates rather than error bars, as this is the standard practice in the field.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on computer resources in § A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that the paper conform with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact of the paper in § A.6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We collect data ourselves and only use open-source (MIT license) codebases and libraries, which are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation along with our code and data.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.