RaC: Robot Learning for Long-Horizon Tasks by Scaling Recovery and Correction

Zheyuan Hu[1], **Robyn Wu, Naveen Enock, Jasmine Li, Riya Kadakia, Zackory Erickson**[*], **Aviral Kumar**[*]

Carnegie Mellon University

Abstract: Modern paradigms for robot imitation train expressive policy architectures on large amounts of human demonstration data. Yet performance on contact-rich, deformable-object, and long-horizon tasks plateau far below perfect execution, even with thousands of expert demonstrations. This is due to the inefficiency of existing "expert" data collection procedures based on human teleoperation. To address this issue, we introduce *RaC*, a new phase of training on human-in-the-loop rollouts after imitation learning pre-training. In *RaC*, we fine-tune a robotic policy on human intervention trajectories that illustrate recovery and correction behaviors. Specifically, during a policy rollout, human operators intervene when failure appears imminent, first rewinding the robot back to a familiar, in-distribution state and then providing a corrective segment that completes the current sub-task. Training on this data composition expands the robotic skill repertoire to include retry and adaptation behaviors, which we show are crucial for boosting both efficiency and robustness on long-horizon tasks. Across three real-world bimanual control tasks: shirt hanging, airtight container lid sealing, takeout box packing, and a simulated assembly task, *RaC* outperforms the prior state-of-the-art using $10\times$ less data collection time and samples. We also show that *RaC* enables test-time scaling: the performance of *RaC* policy scales linearly in the number of recovery maneuvers it exhibits. Videos of the learned policy are available at https://rac-scaling-robot.github.io/.

1 Introduction

Imitation learning on human teleoperation data powers a large chunk of modern robotic learning. In fact, a number of recent academic and industrial bets have been on massively scaling up imitation learning as a form of pre-training for robots [1, 2, 3, 4, 5, 6, 7, 8]. However, results increasingly suggest that this paradigm is approaching a performance ceiling well below perfect task completion. For example, even with over 5000 human demonstrations, state-of-the-art task-specific models can only place a single t-shirt on a hanger with bimanual manipulators at roughly 75% success. While one might hope that more data or alternative learning frameworks could close this gap, in practice these methods still struggle to overcome compounding errors and stochasticity in long-horizon tasks.

This limitation of imitation is fundamental: while mimicking expert actions can imbue the policy with "basic" useful skills, doing so is inherently suboptimal when the robot faces task variations or new initial states, the environment is stochastic or noisy, or the task is inherently long-horizon, where failing at one stage inhibits success in the rest (i.e., when "compounding errors" can be catastrophic) [9]. Thus, policies trained via imitation often fail to generalize to real-world stochasticity and dynamism, and exhibit diminishing returns with additional data, leading to a performance plateau. Crucially, this failure stems not from the algorithm or the model but from the data distribution itself: demonstrations are biased toward clean, successful trajectories, but do not imbue the policy with behaviors needed to tackle compounding errors stemming from stochasticity in long-horizon tasks.

In this work, we propose an alternative paradigm for training robot policies that directly addresses the limitations of success-only imitation learning. We introduce a new phase of learning that is run subsequent to basic imitation learning on clean teleoperation data ("pre-training"), which we

[1]Corresponding author(s): zheyuanh@andrew.cmu.edu. [*]Co-advising.

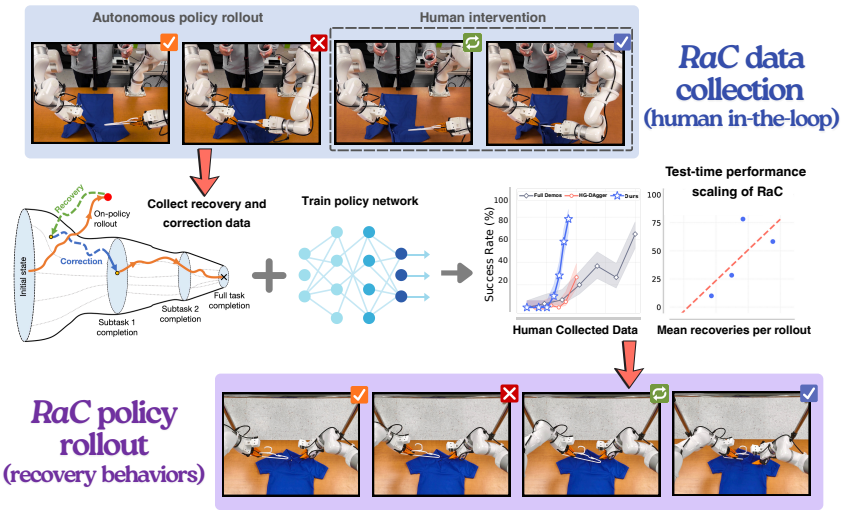

Figure 1: ***Illustrating* RaC.** Our approach enables imitation learning policies to robustly execute long-horizon tasks by explicitly learning skills such as recovery and correction to handle mistakes and failures. Doing so substantially improves data efficiency and results in effective performance scaling at test time.

refer to as **RaC**. The central idea of *RaC* is to train on trajectories that interleave successful task executions with segments that demonstrate recovery, retries, and adaptation, i.e., behaviors that are essential for robustness in complex or novel situations. While standard human teleoperation data may already contain some incidental recovery behavior,[1] *RaC* explicitly encourages and amplifies such behaviors. Conceptually, this phase is analogous to "mid-training" for large language model (LLM) reasoning [11], which aims to illustrate how to best combine basic knowledge (i.e., skills needed to solve individual sub-tasks in our case) with algorithmic behavior (e.g. backtracking, trial-and-error, self-verification, etc.) to solve complex reasoning problems by producing much longer responses.

Concretely, we introduce a lightweight **human-in-the-loop** data collection protocol: human teleoperators intervene to take control from the running policy as soon as it begins to deviate from the correct course. As shown in Figure 3, these interventions naturally fall into two categories: *a) error correction segments*, where human experts guide the robot to solving tasks (similar in spirit to DAgger-style supervision), and *b) recovery segments*, where the human rewinds or repositions the robot to a previously successful state. To scale up recovery and correction for imitation learning, *RaC* standardizes interventions with two rules. *Rule 1 (recover then correct)* structures every human takeover into a reset back to in-distribution states followed by a corrective segment that completes the current sub-task. *Rule 2 (termination after intervention)* ends the episode immediately once the intervention segment finishes, which avoids collecting data on later sub-tasks under distribution shifts from a mixture of learned policy and human expert. Crucially, *RaC* keeps the imitation objective unchanged; performance gains come purely from improved data composition. Applied to three challenging, long-horizon real-world bimanual control tasks, shirt hanging, airtight-lid sealing, and clamshell takeout-box packing, *RaC* outperforms batched full-demonstration and HG-DAgger style human-in-the-loop collection, both in performance and in data efficiency. In particular, *RaC* achieves higher success rates and steeper scaling trends than batched full demonstration and HG-DAgger-style human-in-the-loop data collection, demonstrating superior data efficiency up to 1 order of magnitude.

2 Related Work

Scaling data in robotic learning. Recent work shows that scaling real-robot data across tasks, embodiments, and environments enables generalization. Large robotic datasets [10, 12, 13, 3], paired with highly expressive neural network architectures [5, 6, 4, 14, 1, 15, 2], have produced *generalist* policies that achieve strong performance on many atomic skills (e.g., grasping an object, folding cloth). In parallel, a complimentary line of work [8, 16] demonstrates that a similar data-driven recipe can also produce *specialist* policies that perform very well on substantially more complex

[1]For instance, in the DROID [10] dataset, we find that only 3.68% of the episodes contain recovery behavior.

dexterous bimanual tasks. However, these approaches require collecting *thousands* high-quality expert demonstrations per skill and performance plateau once a certain scale of data is reached [8, 2].

Scaling laws for robot imitation learning. Inspired by work in LLMs [17, 18], several works aim to build scaling laws for robotic imitation [19, 20, 21]. Aimed at evaluating generalization across variations in the task, most of these works analyze the performance of short-horizon tasks as a function of the environmental diversity present in the training data. However, in all such studies, the demonstrations themselves are collected via human "expert" teleoperation and exhibit little variation within the sorts of skills shown in the data. In contrast, instead of studying the environmental diversity, we focus on the data collection strategy within a *trajectory*: specifically, the kinds of maneuvers, recovery behaviors, and variations within. As we show in our experiments, carefully designing atrajectory-level data collection strategy can improve efficiency by more than $10\times$.

Human-in-the-loop imitation learning. Our approach collects intervention data by emphasizing recovery and correction behaviors, which connects it to the broad literature on human-in-the-loop imitation learning. Classical approaches are rooted in DAgger [22], which alternates between **(1)** running on-policy rollouts from the learner, **(2)** querying the expert on visited states, and **(3)** retraining on the aggregated dataset. This framework assumes access to a high-quality expert policy. To adapt DAgger to human operators, HG-DAgger [23] enables teleoperators to provide interventions when policy visits undesirable states, while more recent systems such as RoboCopilot [24] extend these ideas to bimanual mobile manipulation by developing improved interfaces for teleoperation and intervention. Other works [25, 26] explore objectives that combine on-policy rollouts, intervention data, and full human demonstrations. Although our learning objective bears similarities to HG-DAgger[23], we depart from its formulation in a crucial way: prior works largely treat human intervention as an optimal expert solution to be imitated, but we show that collecting recovery segments which by themselves are not task-optimal and may even "undo" progress on a subtask, yields substantially better scaling. This challenges the conventional wisdom that only "expert" interventions are useful, and highlights the importance of trajectory-level data design. We discuss extended related work on shared autonomy and corrections in imitation in Appendix A.

3 Background, Notation, and Problem Setup

Robot setup. Our robot system (Figure 2) consists of two 7-DoF xArm-7 manipulators with scaled-down version of soft grippers [16, 27] to facilitate contact-rich and dexterous tasks. To obtain reactive control, a central server synchronizes and publishes RGB image streams from a top-view camera and two wrist cameras, robot state, and action commands at 60Hz.

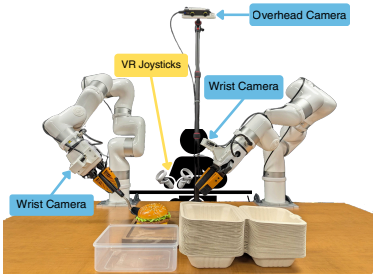

Figure 2: ***Our bimanual manipulation robot system.*** An illustration of our bimanual robot setup showing camera placements and workspace setup.

From a purely learning standpoint, our work is situated in the setting of iterative imitation learning with evolving robotic datasets. Each trajectory τ in this dataset consists of an action a_t for every observation s_t. We describe the precise observation set to our policy in Section 4.3. We develop an approach to collect data for imitation learning that results in better scaling by incorporating human interventions on a previous generation of the learned policy. Formally, our goal is to develop an *iterative human data collection strategy* that improves scaling of task performance as a function of data-collection budget. In other words, we aim to improve the *scaling behavior*, i.e., the slope of task success rate vs. data size. To study data compositions, our data consists of three types: **(i)** full, successful expert demonstrations ΔD^{full}; **(ii)** *recovery* segments, that begin in failure or out-of-distribution regions and return to in-distribution regions; and **(iii)** *correction* segments that complete the current sub-task.

Data collection protocol. Our data collection begins with collecting one round of expert demonstration using an initial budget size R_0, in terms of hours or the number of frames/timesteps. We then first train an initial policy π_0 using this "Round 0" full-demonstration data and evaluate its performance. Prior methods have then scaled the data in one of two ways. In the *batched data collection* protocol, practitioners allocate an additional budget of $K \times R_0$ frames, yielding a single batch of expert data

of size $(K+1) \times R_0$. In the *iterative human intervention* protocol [23, 24, 25, 26], experts instead perform K alternating rounds of intervention and training: in each round k, they provide interventions on rollouts of π_{k-1}, aggregate intervention segments with existing data (in different ways), and train π_k.We study the nature of interventions that improve data scaling of imitation learning the most.

4 *RaC*: Scaling *R*ecovery *a*nd *C*orrection for Imitation Learning

Our goal is to design an iterative data collection strategy for scaling imitation learning. Unlike prior approaches that collect corrective segments [23, 24, 25, 26], our approach guides human interventions to include a substantial proportion of "recovery" behavior alongside "corrective" segments. While recovery segments are *suboptimal* for completing any sub-task within the long-horizon task, they bring the policy back into an in-distribution state preemptively, giving it a chance to re-attempt sub-tasks (Figure 3). In contrast, corrective segments illustrate how to complete the task. ***Our main insight*** is that the ability to retry multiple times gives the policy a generic recipe to attenuate compounding errors that often bottlenecks imitation learning, by trading off acting longer for lower error. We formalize this notion and develop a data collection protocol naturally rich in these behaviors.

4.1 Understanding the Role of Recovery and Correction Segments in Imitation

Consider a robot policy π that executes a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_t)$, where s_t denotes the state at which a human expert intervenes. A sequence of human actions $(a^{\text{h}}_{t+1}, a^{\text{h}}_{t+2}, \ldots, a^{\text{h}}_{t+k})$ starting from s_t constitutes a **recovery segment** if the resulting state s^{h}_{t+k} that the robot reaches after the intervention lies within the distribution of states visited in the prefix of human demonstrations $\mathcal{D}^{\text{full}}[0:t]$. Conversely, this sequence of actions constitutes a **corrective segment** if the resulting state s^{h}_{t+k} lies within the distribution of states visited after timestep t in demonstrations $\mathcal{D}^{\text{full}}[t+1:H]$. We illustrate this concept in Figure 3.

How can recovery segments improve performance? Intuitively, recovery segments return the policy to familiar previous states, giving it another chance to attempt the task, whereas corrective segments show how to push the trajectory forward. This raises a question: Can a policy actually learn to "reset" itself by imitating recovery segments, and why would this improve performance? Our key intuition is that in tasks where the set of valid initial states is broad (e.g., for the task of hanging a shirt, any configuration where a shirt lies on a table and a hanger is held in one of the robot's gripper is an initial state) but the set of valid goal states is narrow (e.g., only when the shirt is correctly placed on the hanger resting on the rack), resetting to a previously encountered state is generally far easier than executing a sub-task correctly (e.g., inserting the collar of the shirt onto the hanger). Because there are multiple familiar past states to reset to, recovery requires less precision and can be more sample-efficient to learn than solving the task.

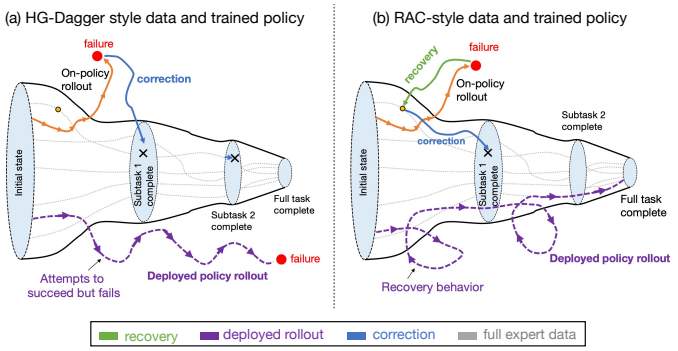

Figure 3: ***Schematic of* RaC.** Data collected via human interventions prescribed by *RaC* and a sample policy rollout when training on only correction data ("HG-DAgger") vs recovery and correction data (*RaC*).

This means that training via imitation learning on a mixture of recovery and corrective behavior should equip a policy with two complementary ways to improve performance: **(1)** by mimicking corrective segments (and full demonstration) to make progress in the first shot, and **(2)** by resetting to a previous familiar state and retrying to self-correct. This ability to recover can be acquired with relatively little data. Once the policy can reliably recover from an anticipated failure, repeated retries would then naturally amplify the overall probability of producing at least one attempt that correctly executes the sub-task. In fact, the probability of never succeeding on a sub-task decays exponentially with the number of retries. This means that total suboptimality in imitation learning performance should decrease. This mechanism is akin to sequential test-time scaling [28] in large language models

(LLMs): just as long chain-of-thought (CoT) models [29] improve performance and generalization by spending more tokens on backtracking and recovery before re-attempting a question, we expect *RaC* to achieve similar gains by performing backtracking and retrying directly in action space.

How can recovery segments improve data scaling relative to HG-DAgger style methods? Recovery segments improve data efficiency because returning to familiar in-distribution states requires less data than mastering corrective skills in many cases. From in-distribution states, the policy already has strong supervision from existing data and the newly added corrective segments amplify this supervision. In contrast, methods like HG-DAgger demonstrate an entirely new behavior from an unfamiliar out-of-distribution state and require the policy to master it. As a result, performance as a function of data scale is expected to be lower for HG-DAgger since it does not necessarily amplify coverage over either in-distribution states or new unfamiliar states within limited intervention budgets.

4.2 Scaling Recovery and Correction Segments in Human Teleoperation

Next, we turn to the question of how to collect imitation data that contains a substantial proportion of both recovery and correction segments. In principle, one could simply *instruct* human teleoperators to artificially stage possible failure states, and demonstrate recovery and corrective behaviors. However, such behaviors produced by humans from contrived or "fake" states may not reflect the out-of-distribution errors that a learned policy would actually encounter. Since policy mistakes are tightly coupled with the policy itself, a purely *offline* approach is unlikely to be effective (akin to LLMs [30]). A more effective alternative is to collect this data through *human-in-the-loop* interventions. Analyzing human intervention data in Section 5.3, we find that it is difficult to achieve a good balance between recovery and correction data with no standardization of human data collection protocol. To instantiate our approach concretely, we prescribe two simple but crucial rules for guiding human intervention:

Rule 1: Pair each recovery segment with a correction segment. Each intervention is structured to contain two phases. First, the human operator performs *recovery* behavior by executing a sequence of actions that bring the robot system back into a familiar in-distribution region of states. Then, the operator provides *corrective* behavior, attempting to push the current sub-task forward (see Figure 3 for an illustration). This simple structure ensures that every intervention teaches the policy both how to reset itself and how to make progress, rather than overemphasizing one or the other.

Rule 2: Terminate after intervention. After an intervention concludes, we terminate the entire episode. In long-horizon tasks, later sub-tasks depend on the correct execution of earlier ones. Allowing the rollout to continue after human intervention would contaminate later sub-tasks with a distribution of states induced by a combination of the learned policy and the human teleoperator. While not problematic itself, learning on this distribution of states might not necessarily improve the policy under its *own* induced distribution of states when it attempts the later sub-tasks, which can be fairly different from the joint human and policy distribution in a particular intervention rollout.

Summary: Balanced composition of recovery and correction

For the widely-used DROID dataset [10], an analysis on its 1% sub-sample reveals only 3.68% of episodes contain ≥ 1 recovery and 16.58% contain ≥ 1 correction. Similarly, in Section 5.3, our HG-DAgger data skews heavily toward corrections with scarce recovery. *RaC* collection protocol standardizes interventions: pair a recovery with a correction, then terminate, to produce a balanced mixture of skills, improving robustness and data efficiency.

Guiding teleoperators for intervention data collection. To facilitate operators in demonstrating trajectories that adhere to the recovery then correction rule, we build a lightweight software tool using an image segmentation model SAM2 [31] to render a robot end effector visitation frequency heatmap by tracking grippers across all RGB frames recorded by the overhead camera in the initial full demonstrations ("Round 0" data). As shown in Figure 4, during data collection, we overlay this heatmap onto the overheadcamera's display window to provide visual aids,

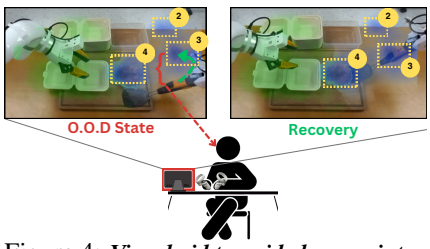

Figure 4: ***Visual aid to guide human intervention.*** In sub-task 3 of `box-packing`, when the policy fails to scoop the burger with spatula, the expert recovers into the bounding box of sub-task 3 before retrying again.

showing in-distribution regions where the robot grippers should recover back to upon intervention. Our approach is one way that can guide recovery demonstrations towards in-distribution regions.

4.3 Policy Architecture and Training via Imitation Learning

We now run imitation learning from a dataset containing multi-modal, long-horizon behaviors of various types: **1)** full demonstrations, **2)** the policy's own full successes from online rollouts, and **3)** human intervention segments with recoveries and corrections. Fitting various sources of data demands a high-capacity policy architecture, with sufficiently expressive output heads [32, 1, 15]. Therefore, we utilize a flow-matching [33] policy to fit an action chunk [34], $A_t = [a_t, a_{t+1}, ..., a_{t+H-1}]$ conditioned on observation $o_t = [I_t^1, I_t^2, I_t^3, q_t]$, where I_t^i is the i-th RGB camera image and q_t is a vector of robot states containing end effectors velocities and relative distance from each other at timestep t. For all tasks, we use $H = 60$, equivalent to predicting one second of actions into the future. Our policy is a 300 million parameter, multimodal diffusion transformer (MM-DiT) architecture [35]. We use separate ResNet-50 [36] vision encoders for all three camera views (one overhead and two wrist cameras) in our real-world experiments and utilize ResNet-18 encoders in simulation. We optimize a conditional flow matching loss [33] for training. Additional details regarding policy training, architecture, and inference are provided in Appendix B.

5 Experimental Evaluation of *RaC*

Our goal is to evaluate *RaC* on bimanual, long-horizon manipulation tasks. Concretely, we aim to answer the following questions: **(1)** Does *RaC* improve data scaling compared to standard human full demonstration data collection, including existing state-of-the-art results?, **(2)** How does *RaC* compare to human-in-the-loop imitation learning methods such as HG-DAgger [23]?, **(3)** Is enhancing the proportion of recovery behaviors critical for effective performance?, and **(4)** How do policies learned by *RaC* differ from traditional imitation learning policies? We answer these questions through experiments in three real-world long horizon tasks. We also use a combination of real and simulated experiments to provide ablations to establish the role of recovery behaviors in training more effective policies for long-horizon tasks, with extra ablations on design choices of *RaC* in Appendix E.

5.1 Evaluation Domains and Task Setups

We study four manipulation tasks; three of them are situated in the real world and one is in simulation (see Figure 5). Our real-world tasks are inspired from some of the most difficult challenges explored in prior work [8, 37]. These tasks are: **1)** `shirt-hanging`: the robot lifts a hanger, passes it between grippers, inserts it through both shirt collars, and rehangs the shirt. **2)** `lid-sealing`: the robot grasps a lid, places it on the container, snaps two tabs, rotates the bowl, and snaps the remaining tabs. **3)** `box-packing`: the robot takes one box, scoops and places a burger inside with a spatula, adjusts placement, closes the lid, and secures the tab. In simulation, we consider a long-horizon assembly task. **4)** `bimanual-assembly`: the robot inserts a white block into a pink socket, then joins with a blue socket, and finally places the assembly on the platform.

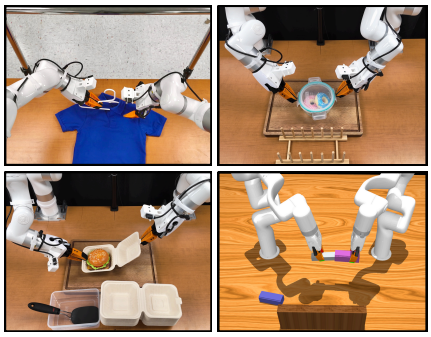

Figure 5: ***Our robot tasks.*** We study 3 real-world long-horizon tasks, `shirt-hang`, `lid-sealing`, `box-packing`, and a simulated `bimanual-assembly` task.

Comparisons and evaluation protocol. We compare the scaling characteristics, performance, and learned behaviors of *RaC* against two approaches for imitation learning: **(1)** scaling up batched full expert data collection, and **(2)** performing human-in-the-loop interventions as per HG-DAgger [23]. For each task, we allocate a total budget of $K \times N$ demonstrations for the batched setting, where N is a base number of demonstrations chosen in advance. To match this budget, we run K rounds of human-in-the-loop data collection, each with equivalent per-round budget, and train the policy in each round using the corresponding intervention data. We conduct evaluations with 60 trials for the real-world tasks and 100 trials in the simulation task with various initial configurations (videos on website). When rolling the trained policy out during evaluation, we record the performance for

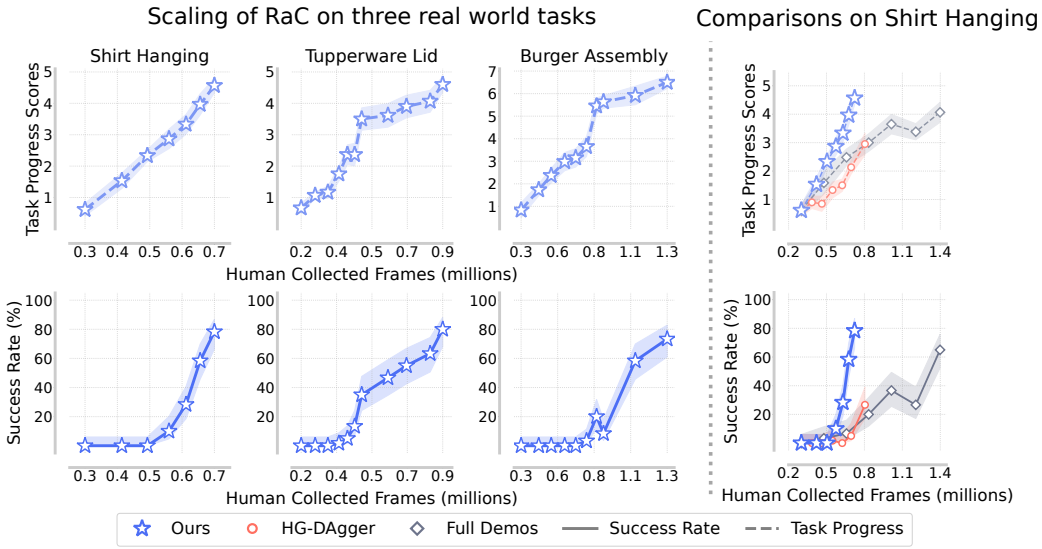

Figure 6: ***Performance scaling for* RaC** as a function of human-collected frames on real-world tasks. Note that within $K = 6$ rounds for shirt-hanging, $K = 10$ rounds for airtight-lid-sealing, and $K = 9$ rounds for takeout-box-packing, we observe the best-known results for tasks of a similar difficulty from prior work. The top row shows average progress over various sub-tasks, the bottom row shows full long-horizon task success rate. On the right, we compare *RaC* to various other baseline approaches based on HG-DAgger and cloning full demonstration data, and observe a substantial improvement in data efficiency.

each sub-task upto an irrecoverable failure, then we terminate the episode. We measure sub-task performance per a binary success or failure indicator function without assigning partial credits.

5.2 Main Results: One Order of Magnitude Improvement in Data Efficiency

Despite the challenges associated with coherent long-horizon execution, deformable object handling, and contact-rich manipulation, our policies reach high success rates and task progress scores with only modest data requirements. Strikingly, just 5 hours of training data suffice to surpass 75% full task success rate on average. To highlight data efficiency gains, consider the *shirt-hanging* task: prior works [8, 38] report needing thousands of expert demonstrations or more than one hundred hours of teleoperation data to achieve a comparable success to *RaC*. *RaC* achieves better results with an ***order of magnitude*** less data, illustrating its efficacy in scaling imitation learning (Figure 6).

Comparisons on real-world tasks. Since scaling up batched data collection on all real-world tasks was infeasible due to the prohibitive expert data costs, we instead scaled the batched data collection baseline on one representative task, `shirt-hanging`. Observe in Figure 6, *RaC* not only achieves substantially higher absolute performance and task progress, but also delivers at least a **2**$\times$ improvement in data efficiency compared to the batched data collection approach. *RaC* also consistently outperforms HG-DAgger. This result does not arise from a subpar baseline: our HG-DAgger implementation exhibits performance trends consistent with prior work, such that it outperforms batched data collection under the same amount of human collected data. Finally, we note that *RaC* exhibits a markedly steeper scaling curve ("higher slope") than either baseline in Figure 6.

5.3 Examining the Properties of *RaC* Policies

Result 1: Robustness of intermediate* RaC *policies. Having established the efficacy of *RaC*, we next analyze the properties of the learned policies in a more systematic manner. We visualize in Figure 8, the distribution of sub-tasks completed by intermediate policy checkpoints produced during successive rounds of human intervention (for *RaC*) and as we scale data (for batched data collection). We observe that the fraction of on-policy rollouts making little progress rapidly decreases with more rounds when using *RaC*. In other words, *RaC* systematically reduces/eliminates the long tail of rollouts that fail or stall early. In contrast, training on increasing amounts of batched full demonstration data does not exhibit the same kind of progress on all sub-tasks, especially in simulation (Figure 8, right). Because our evaluations begin from a broad set of initial configurations, this experiment in a sense highlights the robustness of *RaC*. To summarize, by explicitly scaling recovery, *RaC* drives progress even in the difficult "tail" cases, a persistent failure mode that is common lore with imitation learning.

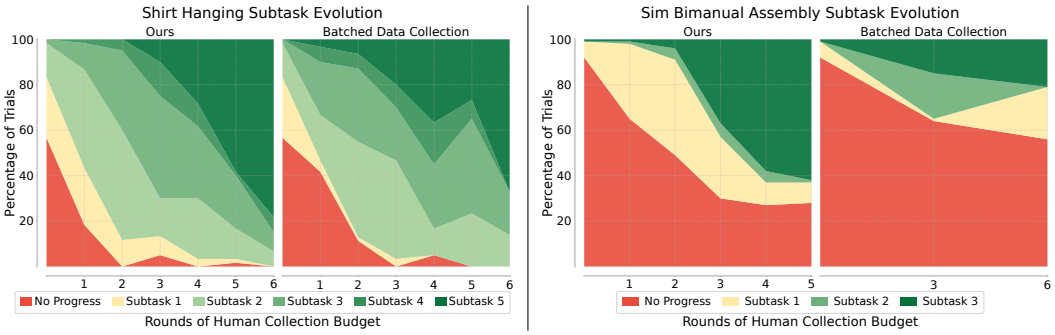

Figure 8: ***Performance profiles for* RaC *and batched data collection.*** For both real-world `shirt-hanging` (left two plots) and simulation `bimanual-assembly` (right two plots) tasks, *RaC* rapidly reduces the fraction of rollouts that make little progress and steadily shift probability mass toward later sub-task completions and full task success. This trend however is not consistent or strong enough for batched data collection.

Result 2: o1-style test-time scaling for robotic policies. Next, we study whether performance scales with more recovery behavior at deployment. To do so, we analyze the subset of evaluation rollouts that successfully solve all sub-tasks across different rounds, and annotate each rollout with the number of recovery attempts it contains. In Figure 7, we show the average number of recovery segments observed against the task success rates. The correlation coefficients r indicate a linear relationship between the task success and recovery frequency. In other words, as the policy learns to demonstrate more recovery behaviors, its overall performance improves. To readers familiar with LLMs, this pattern resembles favorable test-time scaling curves [39]: just as reasoning LLMs perform better when they produce longer CoTs that illustrate backtracking and error correction, robot policies that scale the number of recovery segments directly in the space of action sequences are likely to succeed more.

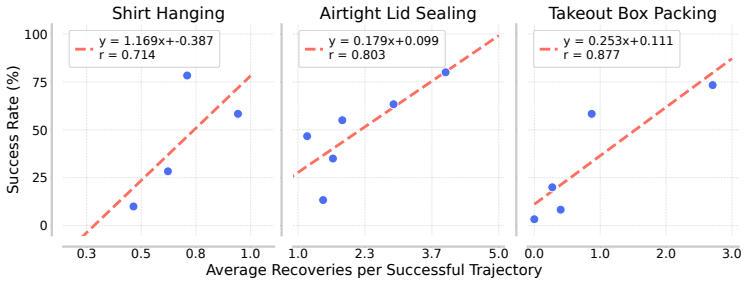

Figure 7: ***Test-time scaling with number of recovery segments.*** We observe a strong linear scaling relationship between the number of recovery segments upon policy deployment and success rate of policies produced by later rounds of *RaC*. This is a form of test-time scaling analogous to that in LLMs [39].

Result 3: Rollouts from* RaC *policies are generally longer and more successful. In simulation `bimanual-assembly` task, we analyzed the wall-clock duration of successful evaluation rollouts across methods (Figure 9). Successful rollouts from *RaC* are skewed towards longer lengths, reflecting recovery behaviors that keep the task on track. For *RaC*, longer length is also correlated with better average performance and more successful rollouts. Successful HG-DAgger rollouts attain the second highest median length, since the robot is trained to still utilize corrective segments to succeed from out-of-distribution states. Policies trained on full demonstration data can likely only succeed when they stay within distribution, resulting in shortest median successful rollout length.

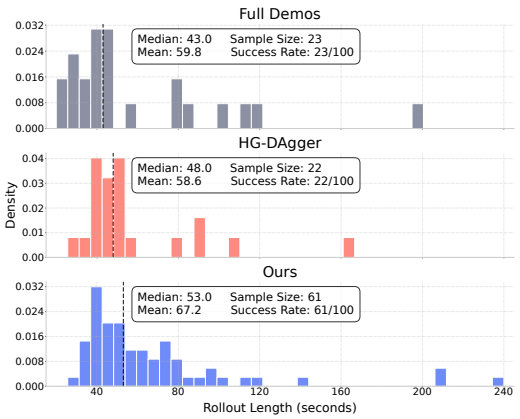

Figure 9: ***Distribution of lengths of successful rollouts*** for various methods. Note that *RaC* policies produce the longest rollouts on average, likely due to the presence of recovery behavior but also succeed more.

Ablation Studies. Please check the Appendix E and G for further ablation experiments, comparisons, and analysis on the data composition of the *RaC* datasets.

Discussion and Conclusion. Please check Appendix I for a discussion of future work.

Acknowledgements

We thank Yuxiao Qu, Bhavya Agrawalla, Lehong Wu, Max Sobol Mark, Anikait Singh, Yufei Wang, Divyam Goel, and Yiran Tao for feedback on an earlier version of this paper. We thank Jason Jingzhou Liu for help with RMPFLow infrastructure. We thank the members of CMU AIRe and RCHI labs for support and feedback. AK thanks Abhishek Gupta, Dhruv Shah, Amrith Setlur, and Max Simchowitz for informative discussions and feedback. This work was supported in part by an Apple seed grant, the Office of Naval Research under N00014-24-12206, and National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health under award number 1R01EB036842-01. We thank the Babel compute cluster at CMU, the TRC program of Google Cloud, and the National Centre for Supercomputing Applications for providing computational resources that supported this work.

References

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

[2] T. L. Team, J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, N. Kuppuswamy, K.-H. Lee, K. Liu, D. McConachie, I. McMahon, H. Nishimura, C. Phillips-Grafflin, C. Richter, P. Shah, K. Srinivasan, B. Wulfe, C. Xu, M. Zhang, A. Alspach, M. Angeles, K. Arora, V. C. Guizilini, A. Castro, D. Chen, T.-S. Chu, S. Creasey, S. Curtis, R. Denitto, E. Dixon, E. Dusel, M. Ferreira, A. Goncalves, G. Gould, D. Guoy, S. Gupta, X. Han, K. Hatch, B. Hathaway, A. Henry, H. Hochsztein, P. Horgan, S. Iwase, D. Jackson, S. Karamcheti, S. Keh, J. Masterjohn, J. Mercat, P. Miller, P. Mitiguy, T. Nguyen, J. Nimmer, Y. Noguchi, R. Ong, A. Onol, O. Pfannenstiehl, R. Poyner, L. P. M. Rocha, G. Richardson, C. Rodriguez, D. Seale, M. Sherman, M. Smith-Jones, D. Tago, P. Tokmakov, M. Tran, B. V. Hoorick, I. Vasiljevic, S. Zakharov, M. Zolotas, R. Ambrus, K. Fetzer-Borelli, B. Burchfiel, H. Kress-Gazit, S. Feng, S. Ford, and R. Tedrake. A careful examination of large behavior models for multitask dexterous manipulation, 2025. URL https://arxiv.org/abs/2507.05331.

[3] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, X. He, X. Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

[4] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[7] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, S. Bohez, K. Bousmalis, A. Brohan, T. Buschmann, A. Byravan, S. Cabi, K. Caluwaerts, F. Casarini, O. Chang, J. E. Chen, X. Chen, H.-T. L. Chiang, K. Choromanski, D. D'Ambrosio, S. Dasari, T. Davchev, C. Devin, N. D. Palo, T. Ding, A. Dostmohamed, D. Driess, Y. Du, D. Dwibedi, M. Elabd, C. Fantacci, C. Fong, E. Frey, C. Fu, M. Giustina, K. Gopalakrishnan, L. Graesser, L. Hasenclever, N. Heess, B. Hernaez, A. Herzog, R. A. Hofer, J. Humplik, A. Iscen, M. G. Jacob, D. Jain, R. Julian, D. Kalashnikov, M. E. Karagozler, S. Karp, C. Kew, J. Kirkland, S. Kirmani, Y. Kuang, T. Lampe, A. Laurens, I. Leal, A. X. Lee, T.-W. E. Lee, J. Liang, Y. Lin, S. Maddineni, A. Majumdar, A. H. Michaely, R. Moreno,

M. Neunert, F. Nori, C. Parada, E. Parisotto, P. Pastor, A. Pooley, K. Rao, K. Reymann, D. Sadigh, S. Saliceti, P. Sanketi, P. Sermanet, D. Shah, M. Sharma, K. Shea, C. Shu, V. Sindhwani, S. Singh, R. Soricut, J. T. Springenberg, R. Sterneck, R. Surdulescu, J. Tan, J. Tompson, V. Vanhoucke, J. Varley, G. Vesom, G. Vezzani, O. Vinyals, A. Wahid, S. Welker, P. Wohlhart, F. Xia, T. Xiao, A. Xie, J. Xie, P. Xu, S. Xu, Y. Xu, Z. Xu, Y. Yang, R. Yao, S. Yaroshenko, W. Yu, W. Yuan, J. Zhang, T. Zhang, A. Zhou, and Y. Zhou. Gemini robotics: Bringing ai into the physical world, 2025. URL https://arxiv.org/abs/2503.20020.

[8] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. URL https://arxiv.org/abs/2410.13126.

[9] D. Ghosh, J. Rahme, A. Kumar, A. Zhang, R. P. Adams, and S. Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in neural information processing systems*, 34:25502–25515, 2021.

[10] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, V. Guizilini, D. A. Herrera, M. Heo, K. Hsu, J. Hu, M. Z. Irshad, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[11] Z. Wang, F. Zhou, X. Li, and P. Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling, 2025. URL https://arxiv.org/abs/2506.20512.

[12] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[13] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K.

Srirama, M. Sharma, M. J. Kim, M. Z. Irshad, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, V. Guizilini, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[14] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[15] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[16] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[17] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint*, 2020.

[18] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[19] L. Zha, A. Badithela, M. Zhang, J. Lidard, J. Bao, E. Zhou, D. Snyder, A. Z. Ren, D. Shah, and A. Majumdar. Guiding data collection via factored scaling curves, 2025. URL https://arxiv.org/abs/2505.07728.

[20] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh. Efficient data collection for robotic manipulation via compositional generalization, 2024. URL https://arxiv.org/abs/2403.05110.

[21] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[22] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

[23] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. 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] P. Wu, Y. Shentu, Q. Liao, D. Jin, M. Guo, K. Sreenath, X. Lin, and P. Abbeel. Robocopilot: Human-in-the-loop interactive imitation learning for robot manipulation, 2025. URL https://arxiv.org/abs/2503.07771.

[25] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

[26] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, page 02783649241273901, 2022.

[27] Zhaxizhuoma, K. Liu, C. Guan, Z. Jia, Z. Wu, X. Liu, T. Wang, S. Liang, P. Chen, P. Zhang, H. Song, D. Qu, D. Wang, Z. Wang, N. Cao, Y. Ding, B. Zhao, and X. Li. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset, 2025. URL https://arxiv.org/abs/2409.19499.

[28] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *ICLR*, 2025.

[29] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[30] A. Setlur, N. Rajaraman, S. Levine, and A. Kumar. Scaling test-time compute without verification or rl is suboptimal. *arXiv preprint arXiv:2502.12118*, 2025.

[31] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

[32] 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*, 2024.

[33] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

[35] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] W. Li, Z. Han, L. Xu, X. Chen, H. Bounds, C. Zhang, and Y. Xu. Taming vr teleoperation and learning from demonstration for multi-task bimanual table service manipulation, 2025. URL https://arxiv.org/abs/2508.14542.

[38] C. Cheang, S. Chen, Z. Cui, Y. Hu, L. Huang, T. Kong, H. Li, Y. Li, Y. Liu, X. Ma, H. Niu, W. Ou, W. Peng, Z. Ren, H. Shi, J. Tian, H. Wu, X. Xiao, Y. Xiao, J. Xu, and Y. Yang. Gr-3 technical report, 2025. URL https://arxiv.org/abs/2507.15493.

[39] OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

[40] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969, 2024. doi:10.1109/ICRA57147.2024.10610040.

[41] J. Luo, C. Xu, J. Wu, and S. Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning, 2025. URL https://arxiv.org/abs/2410.21845.

[42] W. Wang, J. Song, C. Liu, J. Ma, S. Feng, J. Wang, Y. Jiang, K. Chen, S. Zhan, Y. Wang, T. Meng, M. Shi, X. He, G. Ren, Y. Yang, and M. Yao. Genie centurion: Accelerating scalable real-world robot training with human rewind-and-refine guidance, 2025. URL https://arxiv.org/abs/2505.18793.

[43] Z. Sun and S. Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution, 2025. URL https://arxiv.org/abs/2508.05941.

[44] L. Ke, Y. Zhang, A. Deshpande, S. Srinivasa, and A. Gupta. Ccil: Continuity-based data augmentation for corrective imitation learning. *arXiv preprint arXiv:2310.12972*, 2023.

[45] X. Xu, Y. Hou, Z. Liu, and S. Song. Compliant residual dagger: Improving real-world contact-rich manipulation with human corrections, 2025. URL https://arxiv.org/abs/2506.16685.

[46] D. Brandfonbrener, S. Tu, A. Singh, S. Welker, C. Boodoo, N. Matni, and J. Varley. Visual backtracking teleoperation: A data collection protocol for offline image-based reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11336–11342, 2023. doi:10.1109/ICRA48891.2023.10161096.

[47] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff. Rmpflow: A computational graph for automatic motion policy generation, 2019. URL https://arxiv.org/abs/1811.07049.

[48] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Appendices

A Extended Related Work

Shared autonomy. Effectively collecting intervention data requires responsive and intuitive teleoperation interfaces. Prior human-in-the-loop systems have typically relied on 6-DoF SpaceMouse devices [26, 40, 41] or smartphone-based controllers with on-screen buttons and IMU sensing [25]. While functional, these devices come with steep learning curves and are difficult to use for dexterous skills, particularly those requiring wrist rotation. As a result, they are mostly limited to single-arm settings or relatively simple manipulation tasks where end-effector poses remain constrained. More recent work [24] has explored combining VR joysticks with exoskeleton hardware to provide force feedback and richer intervention options, but this demands specialized equipment and additional cost. In contrast, we adopt widely available off-the-shelf VR joysticks as our teleoperation and intervention interface. With a lightweight software modification that we described in Section 4.2, our design enables users to take over control and provide interventions instantly, without the need to align the VR joystick poses with the robot end effector poses.

Corrections in imitation learning. Several works also study employing corrections for imitation learning policies. [42] proposes a "rewind-and-refine" data collection system that detects failures, returns the robot to a previous pose, and then the teleoperator collects corrective trajectories. [43] trains a base diffusion policy on expert data and a learned latent dynamics model that performs test-time steering, encouraging the policy to stay on the expert demonstration manifold. [44] introduces *CCIL*, which learns a locally Lipschitz dynamics model from expert demonstrations and synthesizes corrective labels near the demo manifold to mitigate compounding errors. [45] combines a compliant intervention interface to provide corrections and learns a residual policy to improve the performance of the contact-rich tasks. Instead of engineering the return to in-distribution states through an engineered rewind mechanism or modifications to the base imitation learning policy, *RaC* treats recovery as yet another skill to learn from human demonstrations and scales it explicitly alongside full demonstration and correction skills. Hence, without modifying existing imitation learning objectives or adding additional complexity to the robot system, *RaC* improves the robustness and performance of the policy by directly scaling human demonstration data. Brandfonbrener et al. [46] proposes a similar data collection protocol to *RaC*, in which operators deliberately collect sequences of visually similar failures, recoveries, and successes by backtracking to earlier visual states. However, Brandfonbrener et al. [46] studies the benefit of such data collection strategy through the lens of offline reinforcement learning, enabling efficient learning of accurate value functions from small datasets. *RaC* instead focuses on scaling properties of such recovery skills in dataset composition and their impact on imitation learning policy.

B Policy Architecture and Training Details

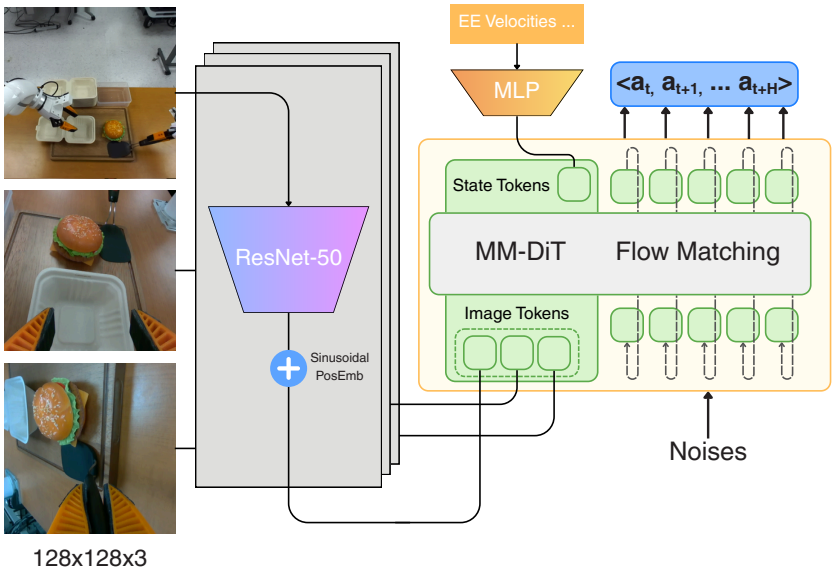

Figure 10: ***Policy architecture.*** We train all imitation learning policies using a multi-modal diffusion transformer architecture via a conditional flow-matching objective.

Concretely, we train the policy using a conditional flow-matching objective:

$$\mathcal{L}_{\text{Flow}}(\theta) = \mathbb{E}_{\substack{o_t,\, A_t \sim \mathcal{D}, \\ x^0 \sim \mathcal{N}(0, I_d), \\ \tau \sim \text{Unif}([0,1])}} \left[\left\| v_\theta(\tau, o_t, x^\tau) - \left(A_t - x^0\right) \right\|_2^2 \right], \tag{B.1}$$

where x^τ denotes an interpolant computed at time τ of the flow, $v_\theta(\tau, o_t, x^\tau) : [0,1] \times S \times \mathbb{R}^d \to \mathbb{R}^d$ is velocity at x^τ, and d is the total dimensionality of action chunks we use. Importantly, when sampling training data from $\mathcal{D}$, we do not include any transitions from the robot's own rollouts, unless the trajectory reaches full task completion without any human intervention. This design choice is consistent with HG-DAgger [23, 24], but different from other methods such as IWR and follow-ups [25, 26], that filter segments based on human knowledge. reweight the loss depending on different categories of trajectories or filtering out certain segments based on human knowledge.

We train all *RaC* imitation learning policies with the same model architecture and training configurations detailed below. With the multi-modal DiT (mm-dit) architecture [35], we use two separate modalities, i.e. two sets of transformer weights to model action generation conditioned on robot observations. The first set of transformer weights processes robot observations, including image tokens from the three camera views after ResNet encoders and a robot proprioceptive state token after a MLP encoder. The second set of transformer weights processes noised action tokens. mm-DiT joins the sequences of the two modalities for the attention operation, such that both representations can work in their own spaces while taking the other one into account. This design is similar to the action expert in [1].

ResNet encoders used in this work finetune on weights pre-trained on ImageNet.

All model trainings are conducted on 4-cards of RTX 6000 Ada GPU servers or 8-cards of L40S GPU servers.

During inference, we generate actions by taking 10 Euler integration steps using the learned vector field from $t = 0$ to $t = 1$, starting with random noise $A_t^0 \sim \mathcal{N}(0, I_d)$. Following [32, 16, 1], we run policy inference once every 0.5 seconds, i.e., we execute the first half of each action chunk and then replan. A complete pseudocode of the procedure is shown in Algorithm 1.

Config	Value
Optimizer	AdamW (default)
Learning Rate	1×10^{-4} (const.)
Global Batch Size	512
Training Length	200 epochs
State Dimension	40
Action Dimension	14
Action Horizon	60

Model Training Configs

Detail	Value
`MM-DiT` modalities [35]	2
Flow Matching Steps	10
MM-DiT Hidden Size	768
MM-DiT Depth	12
MM-DiT Heads	12
Vision Encoder	ResNet-50 / ResNet-18
Total Parameters	367.865M

Flow Matching Model Details

Table 1: **Training configurations and model details.** Left: training hyperparameters. Right: model architecture specifics.

C *RaC* Pseudocode

Algorithm 1 *RaC* Data Collection and Training Protocol

Given per-round human data collection budget $\mathcal{B}$ in numbers of frames; total human intervention data collection round K.
Initialize flow-matching policy $\pi_\theta^{k=0}$; dataset $\mathcal{D}_{0:K} \leftarrow \varnothing$
Collect $\mathcal{B}$ frames of expert demonstrations $\rightarrow \Delta\mathcal{D}_0$; $\mathcal{D}_{0:K} \leftarrow \Delta\mathcal{D}_0$; $\pi_\theta^{k=0} \leftarrow \textsc{Train}(\mathcal{D}_{0:K})$ via Flow Matching B.1;

Human Intervention Data Collection Rounds

for $k = 1$ to K **do**
  initialize human policy π_H, intervention function I
  $\Delta\mathcal{D}_k \leftarrow \varnothing$; $b \leftarrow 0$ ▷ b counts budget used this round
  while $b < \mathcal{B}_k$ **do**
    $s_0 \leftarrow$ `env.reset()`; `traj` $\leftarrow [\,]$; `intervened` $\leftarrow$ **false**; $t \leftarrow 0$
    while not `env.done()` **do**
      if $I(s_t) = 0$ **then** $a_t \sim \pi_\theta^{k-1}(\cdot \mid s_t)$; `is_human` $\leftarrow 0$
      else $a_t \sim \pi_H(\cdot \mid s_t)$; `is_human` $\leftarrow 1$; `intervened` $\leftarrow$ **true** ▷ Rule 1: Pair each recovery a correction
      $s_{t+1} \leftarrow$ `env.step`(a_t); `traj.push`$(s_t, a_t,$ `is_human`$)$; t+=1
      if `is_human` $= 0$ **and** INTERVENTIONDONE() **then break** ▷ Rule 2: Terminate after intervention concludes
    if `intervened`=**false then** ▷ If an entire trajectory has no human intervention $\Rightarrow$
      $\Delta\mathcal{D}_k \cup=$ `traj` ▷ add full trajectory into dataset, with no human budget counted
    else
      $\Delta\mathcal{D}_k \cup= \{(s,a) \in$ `traj` $:$ `is_human` $= 1\}$ ▷ add only human intervention transitions into dataset
      $b = b +$ `|traj|` ▷ charge full episode length to budget
  $\mathcal{D}_{0:K} \cup= \Delta\mathcal{D}_k$; $\pi_\theta^k \leftarrow \textsc{Train}(\mathcal{D}_{0:K})$ ▷ Aggregate datasets, then train policy via flow-matching B.1

D Example Rollouts on Various Tasks

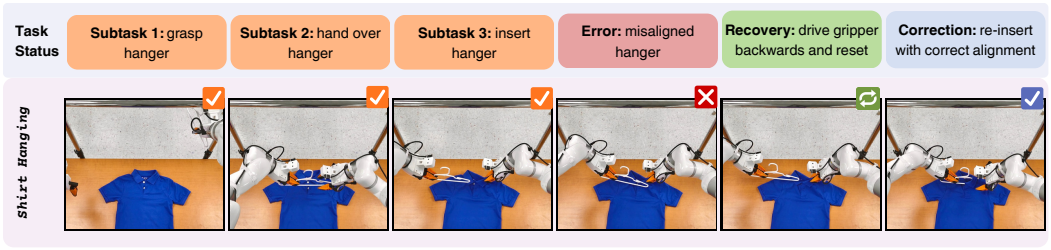

Figure 11: **RaC** ***rollout on the shirt-hanging task.*** In this task, recovery corresponds to driving the gripper and hanger backwards and correction corresponds to reinserting the hanger again.

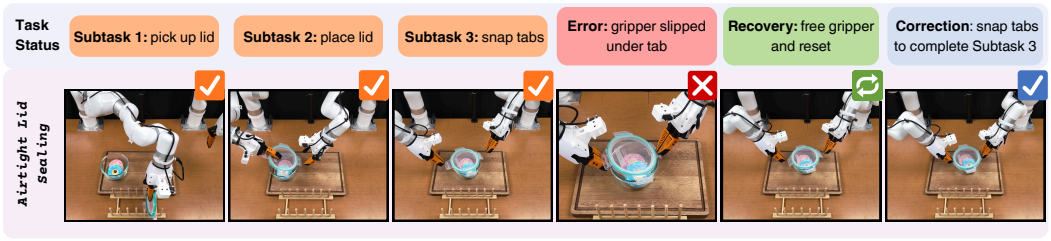

Figure 12: **RaC** ***rollout on the airtight-container-lid-sealing task.*** In this task, recovery corresponds to driving the gripper and hanger backwards and correction corresponds to reinserting the hanger again.

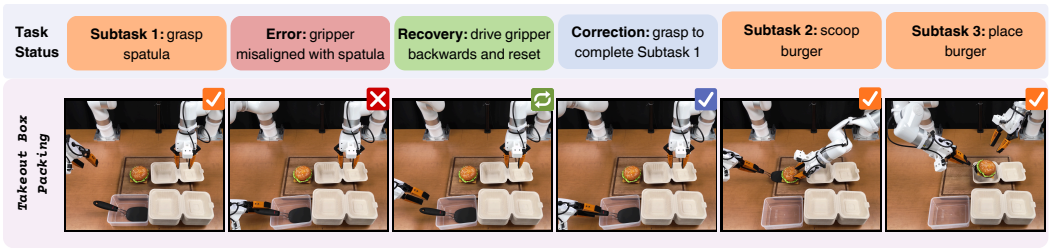

Figure 13: **RaC** ***rollout on the takeout-box-packing task.*** In this task, recovery corresponds to driving the gripper backwards and correction corresponds to regrasping the spatula again.

E Ablation Studies for the *RaC* Data Collection Protocol

Finally, we present ablation studies to better understand the properties of the human intervention data collected by *RaC* across training rounds. In Figure 7, we visualize the composition of intervention data over 4 rounds in the simulation task. We compare data collected using the full *RaC* approach ("Ours") and *RaC* without enforcing 'recover-then-correct' ("Ours w/o Rule 1", i.e. HG-DAgger with only Rule 2). Recall that these Rules were prescribed in Section 4.2.

We classify each intervention frame as either a recovery segment or a corrective segment. Observe in Figure 14 (left), that while *RaC* maintains a roughly balanced ratio of recovery to corrective frames (close to 1:1), conventional intervention data exhibits a highly skewed distribution dominated by corrective frames, with recovery frame's proportion decreasing sharply in later rounds. The total number of intervention frames naturally decreases as policies improve and require fewer interventions.

Next, we study the effect of Rule 2 in *RaC*: truncating an episode after human intervention concludes. In the simulation task (Figure 14), we observe that terminating early after an intervention alone ("Ours w/o Rule 1") yields more effective performance scaling than continuing policy rollouts after human intervention ("Ours w/o Rule 1&2", i.e., HG-DAgger). We hypothesize that this effect arises because allowing the rollout to continue after human intervention completes contaminates later parts of the trajectory with states influenced by both the human and the policy, producing data that are out-of-distribution for a learned policy. By terminating right after the intervention, we ensure that the collected data cleanly reflect recovery–correction behavior, while subsequent sub-tasks are reached only with the policy's own distribution in future rounds, leading to more efficient data scaling.

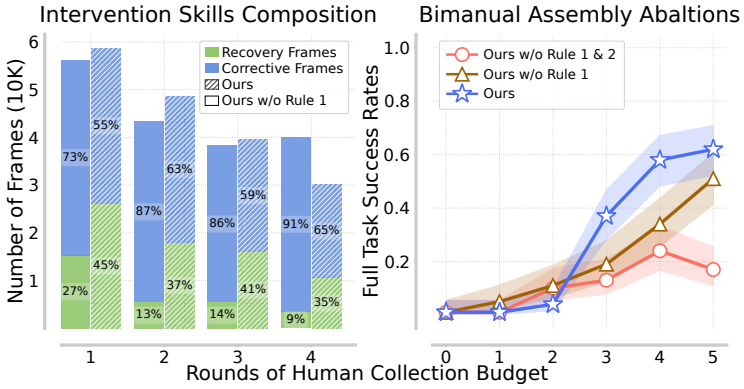

Figure 14: ***Ablation studies on the bimanual-assembly simulation task.*** **Left:** Assessing the composition of human intervention data collected in each round. Note that data collected via *RaC* maintains a high proportion of recovery segments along with corrective frames. On the other hand, the intervention data collected by HG-DAgger skews heavily towards corrective frames. **Right:** Utilizing "Rule 2" and terminating the intervention episode early yields better data scaling of performance than continuing the policy rollout after the recover-then-correct intervention is complete. This showcases the importance of both rules prescribed by *RaC*.

F Bimanual Manipulation Tasks Evaluation Protocols

- `shirt-hanging`. Our shirt hanging task setup follows ShirtEasy from ALOHA Unleashed[8] with identical child-size polo shirts, child-size hanger and hanging rack. During policy evaluation, we perform 60 trials per policy. For each of the five shirts, we randomize the initial shirt pose in three orientations (center, left, right) and four hanger placement locations on rack uniformly, resulting in 12 trials per shirt, and 60 trials in total. We also provide an entire uncut evaluation video recording on website https://rac-scaling-robot.github.io/ for reference.
- `airtight-container-lid-sealing`. In this task, we perform 60 trials evaluation for each policy tested. At the beginning of each trial, we randomly assign initial configurations uniformly to 5 different lid placement locations on the drying rack and 12 different container locations on the cutting board, resulting in a total of 60 trials.
- `clamshell-takeout-box-packing`. For this task, we perform 60 trials evaluation for each policy tested. At the beginning of each trial, we randomly assign the locations of the burger uniformly on the right half of the cutting board. For the placement of the takeout box pile and the spatula, we place them roughly in front of the cutting board with a small range of variations each trial.

For computing the confidence interval when reporting results and producing the scaling curves, we compute the 95% confidence interval for the task progress scores, where the max scores equal to the maximum number of sub-tasks within each task. For the full task success rates, where each trial receives a binary score for whether the robot completed the entire task successfully, we compute the 95% Wilson score interval, i.e. a formula for binomial proportion confidence interval.

G Comparison to Prior Works on the Shirt-Hanging Task

Name	Policy Architecture	Model Size	Training Data Size	SR
ALOHA Unleashed [8]	Diffusion Transformer policy	217M	$\sim$89 hours (5345 shirt-hanging expert demos)	75.0%
Seed GR-3 [38]	Vision-Language-Action model	4B	116 hours of shirt-hanging expert demos and vision-language data	$\sim$63.6%
Ours (*RaC*)	Flow-matching Transformer policy	368M	**5 hours** (*RaC* data: expert, recovery, and correction)	**78.3%**

Table 2: ***Comparison to similar shirt-hanging tasks in prior work.*** Under similar task setups and difficulty, the full task success rate ("SR" of *RaC* policy is higher than other methods using an order of magnitude less data. See Appendix G for details.

ALOHA Unleashed. In ALOHA Unleashed [8], the shirt-hanging task is performed with bimanual ALOHA robot [34] at two difficulty levels: ShirtEasy and ShirtMessy. ShirtEasy uses 5345 full trajectories and ShirtMessy uses 3313 full trajectories, with a fleet of robots and expert teleoperators. In our work, the shirt-hanging task is designed to be as close to ShirtEasy as possible. They report a full task success rate of 75% on the ShirtEasy task with Diffusion Policy trained on both the ShirtEasy and ShirtMessy data. To standardize the comparison of the size of the data between different works, we approximate the length of the ShirtEasy dataset in **hours** from ALOHA Unleashed by using an average of 1 minutes per trajectory. Thus, we estimate a total of $5345 * 60/3600 \approx 89$ hours for the ShirtEasy dataset.

Seed GR-3. In Seed GR-3 [38], the shirt-hanging task is performed on a custom-designed bimanual mobile manipulation platform. The task differs from ours and [8] in the final step, where the robot "needs to rotate its mobile base from the table to the drying rack to hang the clothes", while other sub-tasks remain largely consistent. Importantly, Seed GR-3 reports their performance in **average task progress**, where a full success corresponds to 1.0 or 100% and successful completion of each sub-task contributes a fractional score towards the overall task progress. This is different from the **success rate** metric (Table 2), where only full success trials are given score of 1.0 and other trials do not receive any partial credit. To standardize the evaluation metrics, since ALOHA Unleashed[8] does not report task progress scores, we estimate the full task success rate for GR-3[38] using the Sankey diagram displayed in Figure 10 of their paper, by dividing the vertical heights of the bar representing the last sub-task by the vertical height of the figure location representing the start. This results in a ratio of $7/11 \approx 0.636$.

H Teleoperation & Human Intervention Interface

To enable effective interventions for *RaC*, we design a lightweight shared-autonomy using Oculus Quest VR controllers. Our design uses a "clutch" mechanism that unifies policy execution and human takeover: when the side button is pressed, controller motions are mapped directly to the end effector enabling the human to take over control and intervene, and when the side button is released, the robot follows the learned policy. To reduce operator effort, we adopt a local-frame registration scheme with relative pose deltas. Let v denote the fixed VR headset coordinate frame, and let c_t denote the hand-controller frame at time t. At clutch engagement ($t = 0$), we define the controller's pose relative to the headset frame, $T^v_{c_0}$, as the local base frame. Subsequent poses are then expressed in this local frame as $T^{c_0}_c(t) = (T^v_{c_0})^{-1} T^v_{c_t}$, with incremental translational $\Delta p_k = p_k - p_{k-1}$ and rotational $\Delta R_k = R_{k-1}^\top R_k$ offsets used to parameterize end-effector commands. This design eliminates the need for global posture alignment, allowing operators to instantly take over and intervene with minimal friction. A picture is shown in Figure 15.

Our system employs RMPFlow [47] as the inverse-kinematic motion generator, enabling real-time collision avoidance and smooth arm motions.

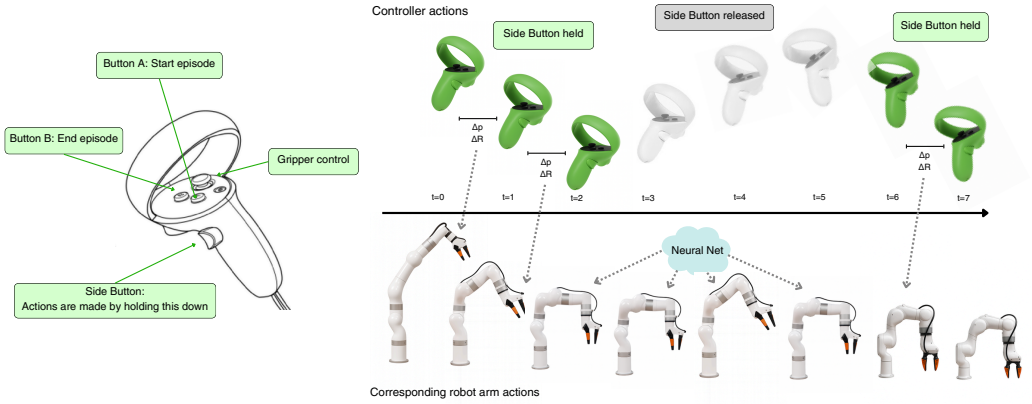

Figure 15: ***VR handset interface for shared autonomy in* RaC.** We design and implement a "clutch" design that enables smooth handover from the robot policy to the human teleoperator.

I Discussion, Conclusion, and Future Work

We presented an approach, *RaC*, for scaling imitation learning in the real world. Our core idea is to scale not just the quantity of data, but the *type* of data, explicitly pairing recovery and correction behaviors collected through human interventions. By doing so, we enabled policies to mitigate compounding errors, retry from failures, and achieve substantially higher data efficiency than standard teleoperation or correction-only approaches. Our experiments demonstrated that this paradigm yields robust policies on long-horizon, contact-rich tasks with orders of magnitude less data than prior work and much better data efficiency than our comparisons. We also illustrated a form of "test-time scaling" by showing that more recovery segments and longer action times correlate with higher performance.

We believe that there are quite a few avenues for future work. First, analogous to how autonomous RL began performing substantially better on top of properly mid-trained initializations for LLMs [11], we believe that policies trained via *RaC* bear the potential to serve as good initializations for online RL fine-tuning on a real robot. Unlike typical imitation pre-trained policies that attempt to perform "optimal" behavior (and typically lose track upon failing to accomplish the task), we hypothesize that policies from *RaC* would naturally provide more structured exploration and coverage during online RL due to the presence of recovery behavior. Recovery provides natural "stitching" points [48] which might also be amenable to value-based training. Another interesting direction for future work is to apply *RaC* on top of generalist vision-language-action (VLA) models [1, 14, 7]. Finally, while prior results do show some examples of recovery behaviors in VLA models, it is unclear if such behaviors systematically emerge in most settings or not, and studying this aspect rigorously (for example, by plotting test-time scaling curves analogous to Figure 7) is also useful for the community.