Human-Robot Copilot for Data-Efficient Imitation Learning

Rui Yan* Zaitian Gongye* Lars Paulsen Xuxin Cheng Xiaolong Wang
UC San Diego

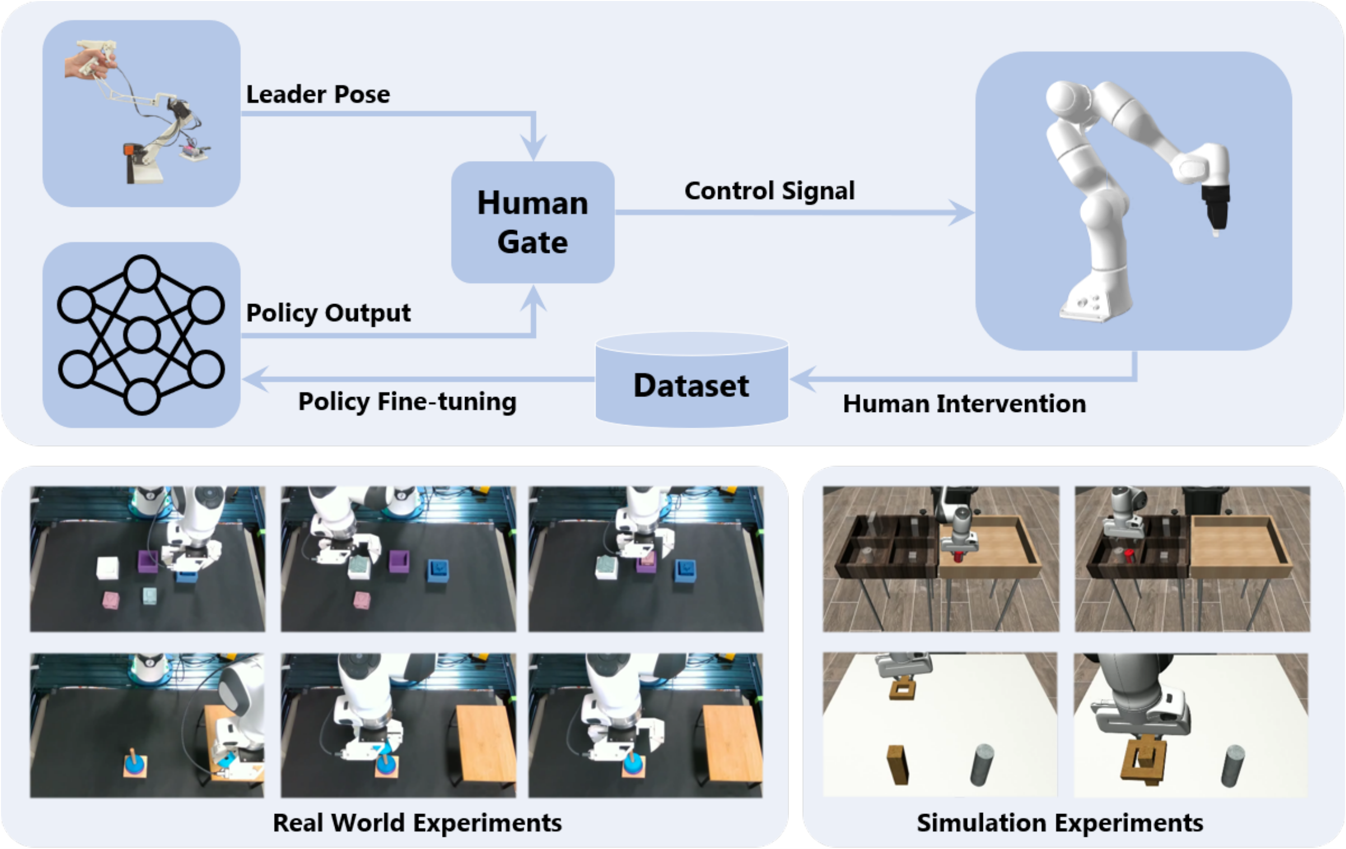


Fig. 1: Overview of the Human-Robot Copilot framework. The human teleoperator determines when to intervene in policy execution and collect augmentation data for policy fine-tuning. The bottom right illustrates simulation experiments in robosuite, and the bottom left shows real-world experiments: cube sorting in a highly randomized environment and tower of Hanoi insertion requiring high-precision actions.

***Abstract*— Collecting human demonstrations via teleoperation is a common approach for teaching robots task-specific skills. However, when only a limited number of demonstrations are available, policies are prone to entering out-of-distribution (OOD) states due to compounding errors or environmental stochasticity. Existing interactive imitation learning or human-in-the-loop methods try to address this issue by following the Human-Gated DAgger(HG-DAgger) paradigm, an approach that augments demonstrations through selective human intervention during policy execution. Nevertheless, these approaches struggle to balance dexterity and generality: they either provide fine-grained corrections but are limited to specific kinematic structures, or achieve generality at the cost of precise control. To overcome this limitation, we propose the Human-Robot Copilot framework that can leverage a scaling factor for dexterous teleoperation while maintaining compatibility with a wide range of industrial and research manipulators. Experimental results demonstrate that our framework achieves higher performance with the same number of demonstration trajectories. Moreover, since corrective interventions are required only intermittently, the overall data collection process is more efficient and less time-consuming.**

I. Introduction

Collecting human demonstrations via teleoperation has become a popular paradigm for enabling robots to acquire task-specific skills [1]–[4]. Although such demonstrations can effectively bootstrap imitation learning policies, their deployment often reveals critical limitations. Due to compounding errors during deployment and the stochasticity of environments, the learned policy often falls into out-of-distribution (OOD) states where the policy struggles to generalize suitable actions. While collecting additional data with careful randomization may increase the coverage of states in the demonstrations, the approach still suffers from low data efficiency in the absence of prior knowledge about the OOD states.

Several recent efforts have attempted to address this challenge through human-in-the-loop data augmentation, in which the robot runs automatically most of the time while human intervention is introduced when the policy fails in order to provide corrective demonstrations. These corrective behaviors are subsequently incorporated into the demonstration dataset, thereby directly guiding the robot on how to act in OOD states.

Sirius [5], for example, leverages a space mouse for human intervention and correction to enable mixed control between a learned policy and human teleoperation during deployment, while simultaneously collecting new data for online fine-tuning. However, the corrective capabilities of the space mouse are inherently constrained—it only allows uniform translational or rotational adjustments, making it unsuitable for more complex refinements. Robo-Copilot [6] instead introduces a dual-robot setup in which a ”follower” robot copys joint positions of a teleoperated ”leader” robot for human demonstrations while the leader mirrors the follower robot during policy execution. This design benefits from the shared workspace and kinematic equivalence of the two robots, enabling intuitive data collection during deployment. Yet, the requirement of identical or proportionally scaled kinematics across robots fundamentally limits its applicability and prevents broader generalization.

These approaches highlight the tension between dexterity and generality in human-in-the-loop robot learning. Although existing systems provide useful correction methods, they struggle to support fine-grained control while remaining broadly compatible with heterogeneous robot platforms at the same time.

To address these limitations, we propose Human-Robot Copilot, a cross embodiment framework designed to improve the entire pipeline of human demonstration collection and imitation learning. Our system ensures that the leader and follower robots share overlapping workspaces, enabling intuitive corrective teleoperation during deployment. At the same time, heterogeneity in arm design allows us to introduce scaling factors that facilitate fine-grained control, while providing full 6-DoF end-effector pose commands that are compatible with a wide range of industrial and research manipulators.

We demonstrate that fine-tuning with human data collected through this framework significantly improves policy performance on contact-rich, high-precision, and logically complex tasks. This highlights the potential of heterogeneous teleoperation systems to bridge the gap between efficient human data collection and robust robot learning in real-world deployment scenarios.

II. Related Work

A. From offline imitation learning to human-in-the-loop.

Traditional imitation learning paradigms [1], [4], [7]–[10] typically follow a three-stage pipeline: collecting a fixed set of human demonstrations, training a policy on this dataset, and subsequently deploying the learned policy. Once the demonstrations are provided, the human supervisor is excluded from the control loop, leaving the policy fully responsible for execution. In contrast, human-in-the-loop imitation learning [5], [6], [11]–[14] maintains continuous human involvement during deployment, enabling supervision, intervention, supplementary demonstrations, and evaluation. This paradigm not only addresses data sparsity by allowing humans to provide additional demonstrations on failure cases [13], [14] but also enhances safety through real-time teleoperation override [5]. Moreover, it improves data efficiency by focusing human input on the most challenging parts of the task where the current policy underperforms. As the policy improves, the human workload can gradually decrease. A key requirement for these systems is the ability to seamlessly switch between autonomous policy control and human teleoperation, which fundamentally depends on aligning the workspaces of the human and robot. However, existing teleoperation devices often fall short of this requirement. [5], [12], [15]

B. Teleoperation devices.

Current human-in-the-loop teleoperation interfaces largely fall into two categories. The first category, exemplified by Smartphone, SpaceMouse and VR controllers [5], [12], [16]–[19], provides cross embodiment control. Here, the human input is added as a displacement or velocity increment to the current robot states. While these devices are general-purpose and capable of both coarse and fine motions, the delta control is unintuitive [20] and ill-suited for tasks requiring both large-scale movements and fine-grained precision, since the controller's displacement is relative and lacks a fixed spatial reference to the robot's base, making it difficult for operators to perceive absolute position or scale. Moreover, demonstrations collected via these devices show higher variance and jerk, making them harder for the policy to learn [20].

The second category, relies on joint-space mirroring using paired, isomorphic robot arms [3], [6], [21]–[25]. These devices enable intuitive one-to-one control by directly mapping joint angles, but their limitations are twofold: (1) the teleoperation scale is fixed, making it difficult to perform fine-tuning in precision-demanding subtasks, and (2) their homomorphic design restricts transferability, as a leader arm can only control a structurally identical follower. Adapting to a new robot therefore requires redesigning and rebuilding the leader hardware, significantly reducing flexibility.

C. Towards cross-embodiment copilots.

To overcome these limitations, we propose a cross embodiment copilot framework that integrates hardware, control, and learning. By leveraging kinematic workspace alignment rather than strict joint homomorphism, humans can perform both large-scale and fine-grained control across diverse robot morphologies. Moreover, the framework allows humans to efficiently provide supplementary demonstrations and fine-tuning data during deployment, directly targeting the failure

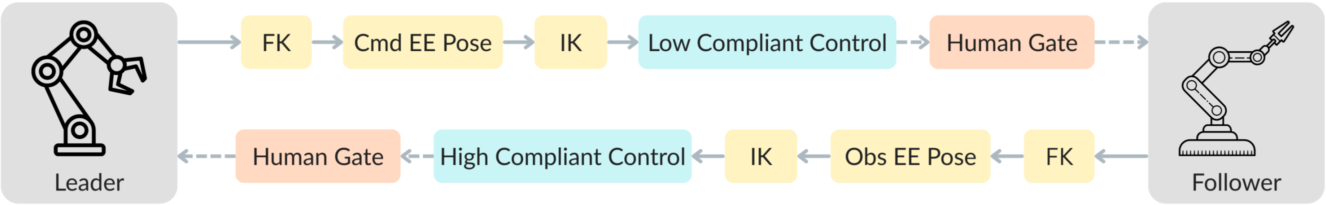


Fig. 2: Bidirectional control and observation communication. Forward and inverse kinematics (FK/IK) are continuously computed for both the leader and follower robots. Dashed lines denote control signals, of which only one is selected for synchronization between the two robots. The human teleoperator determines which control signal the two robots executes.

modes of the current policy. This design achieves the intuitiveness of joint-space mirroring with the generality lacking in existing approaches, while closing the loop between control and learning to enable data-efficient human-in-the-loop imitation learning.

III. Method

Our framework consists of two main components: an interactive heterogeneous teleoperation system and a human-in-the-loop imitation learning pipeline. We first introduce the architecture and usage of the teleoperation system, which enables interactive data collection across heterogeneous embodiments at different control scales. This system is used to collect both the initial demonstrations and the fine-tuning data during deployment. We then describe how the collected fine-tuning data are integrated into the imitation learning process to continually improve the policy.

A. *Teleoperation System for Human-in-the-loop*

To realize a cross embodiment teleoperation system, a common approach is to leverage the inverse-kinematic (IK) solver to translate the desired end-effector pose derived from the leader device into the joint positions of the follower arm. Although these IK-based teleoperation devices have been explored previously [18], [26], they are typically limited to unidirectional control, where a leader arm solely drives a follower arm. Such designs have primarily been applied in traditional offline imitation learning settings.

However, for human-in-the-loop teleoperation, bidirectional communication and control capabilities are needed for synchronizing the two devices and seamlessly switching between the two control modes. By augmenting the IK-based device [26] with bidirectional control, we extend its applicability to human-in-the-loop scenarios, where it enables efficient collection of fine-tuning data during policy deployment.

Our control logic operates in two modes. During the initial data collection, the leader arm reads the motor encoders and IMU signals. The joint readings are passed through forward kinematics to obtain the end-effector position, which is then combined with the IMU rotation to estimate the full end-effector pose. This pose is provided to the IK solver of the follower arm, and the resulting command joint positions are sent to the follower arm for execution. In parallel, the follower returns observed joint positions, which are converted via forward kinematics into Cartesian end-effector positions. During data collection, these states of the follower are only treated as observations.

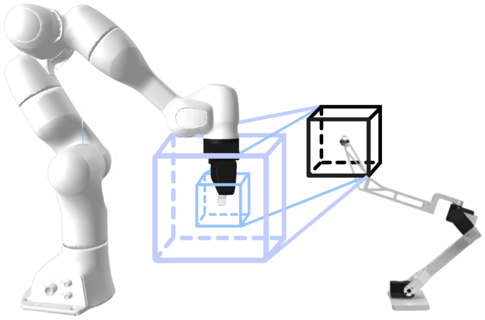


Fig. 3: Illustration of task workspaces under different scaling factors. The black cube represents the workspace of the leader arm, while the two blue cubes correspond to task workspaces under different scaling factors. A larger task workspace facilitates rapid large-scale movements, whereas a smaller task workspace supports precise and accurate actions for high-precision tasks.

During human-in-the-loop fine-tuning, teleoperation and policy execution run as two parallel channels, although only one channel's output is forwarded to the follower arm at any given time. The operator can switch between teleop and policy channels with a single command. Importantly, because of the bidirectional control logic, even when the follower listens to the policy channel, the leader arm continues to receive the follower's joint positions. These joint positions are converted via forward kinematics into Cartesian end-effector pose, and then solved again through IK to update the joint positions of the leader arm, ensuring both arms remain aligned in the same workspace. This makes switching control intuitive for the human operator. Meanwhile, both channels can be subscribed by the data collection program, which records synchronized command joint positions, observed joint positions, and images under a unified timestamp to construct data clips. Furthermore, when running the policy, switching into teleoperation mode allows us to compare the policy outputs against human demonstrations, thereby inspecting the abnormal output of the policy and identifying potential out-of-distribution (OOD) states.

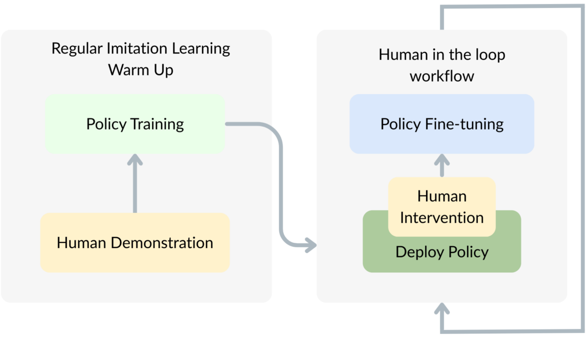


Fig. 4: Training and data augmentation workflow. The base policy is first initialized through regular imitation learning. It is then deployed to identify potential failure modes. During deployment, a human teleoperator intervenes when necessary, providing corrective actions. These corrective demonstrations are recorded and incorporated into the original dataset, which is subsequently used to fine-tune the policy.

To stabilize control, both leader and follower arms are equipped with gravity and friction compensation, allowing them to be driven with low-gain PD controllers. In teleoperation mode, the returned control commands use even lower PD gains to suppress oscillations and improve compliance. In policy mode, by contrast, friction compensation of the leader arm is disabled and its PD gains are increased, ensuring real-time synchronization between the two end-effectors. The bidirectional communication pipeline is illustrated in Fig.2

B. Control Scale Adjustment

By leveraging the heterogeneous characteristics of the system, the teleoperation can be flexibly adjusted, enabling coarse control for large-scale movements and fine-grained control for precision-demanding tasks. The illustration of the task workspaces under different scaling factors are in Fig.3.

Since the orientation of the end-effector does not require scaling, we consider only the positional components. Let follower end-effector position be $\mathbf{x}_f \in \mathbb{R}^3$, and the leader end-effector position be $\mathbf{x}_l \in \mathbb{R}^3$, the mapping between the two positions is defined as

$$\mathbf{x}_f = \alpha(\mathbf{x}_l - \mathbf{c}_l) + \mathbf{c}_t, \tag{1}$$

where $\mathbf{c}_l$ denotes the center of the leader robot's workspace, $\mathbf{c}_t$ denotes the center of the task workspace, and α is the scaling factor.

For large-scale tasks, such as object transfer, we employ a larger scaling factor ($\alpha = 2.0$) for the alignment between the two workspaces. Conversely, for precision-demanding tasks, such as object insertion, a smaller scaling factor($\alpha = 0.5$) is adopted to enable more accurate teleoperation.

C. Human-in-the-loop Imitation Learning

After obtaining the base policy, we proceed to human-in-the-loop data collection. The training and data augmentation workflow is illustrated in Fig.4. During deployment, the robot executes the learned policy, but the human operator can take over control via teleoperator at any time. When intervening, the operator can also selectively record demonstrations. Unlike full trajectories, we only log the segments after human takeover, which we call data clips. These clips may correspond to different parts of a task rather than entire demonstrations, but they provide targeted supervision precisely where the policy struggles.

Algorithm 1 Human-in-the-loop Imitation Learning with Clip-Based Batch Finetuning

```
Require: Base demos $\mathcal{D}_0$, initial policy $\pi_0$, human expert $\pi^*$, trigger $K$, iterations $N$
Base train: $\pi_1 \leftarrow \mathrm{BC}(\mathcal{D}_0)$
for $i = 1$ to $N$ do
    $\mathcal{C} \leftarrow \emptyset$                                    ▷ clip buffer
    Deploy $\pi_i$
    while task not done do
        observe $s$
        if human intervenes then
            start clip $C \leftarrow \emptyset$
            while human in control do
                execute $a^* \leftarrow \pi^*(s)$; append $(s, a^*)$ to $C$; step env; update $s$
            end while
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$
        else
            execute $a \leftarrow \pi_i(s)$; step env; update $s$
        end if
        if $|\mathcal{C}| \geq K$ then
            $\mathcal{D} \leftarrow \mathcal{D}_0 \cup \bigcup_{C \in \mathcal{C}} C$
            $\pi_{i+1} \leftarrow \mathrm{Finetune}(\pi_i, \mathcal{D})$; $\mathcal{C} \leftarrow \emptyset$; Redeploy $\pi_{i+1}$
        end if
    end while
end for
Output: final policy
```

Compared to traditional offline training pipelines such as Action Chunking Transformer (ACT) [3], human-in-the-loop imitation learning requires the policy to support fast iterations. Therefore, we utilize the ACT policy while reducing the network size to enable rapid finetuning after data augmentation. Moreover, we choose ResNet-18 as our vision backbone instead of other more powerful but larger choices, such as Dino-V2, used by other ACT-based policies [18].

For fine-tuning, we combine the collected data clips with the original demonstrations (to avoid catastrophic forgetting of previously learned skills). We summarize the fine-tuning procedure in Algorithm 1. With the smaller network, training the base policy from scratch takes about 40 minutes while finetuning takes less than 10 minites to converge, enabling us to rapidly redeploy the updated policy. This fast retraining cycle supports repeated rounds of human-in-the-loop data collection and refinement, making the framework practical for real-world iterative improvement.

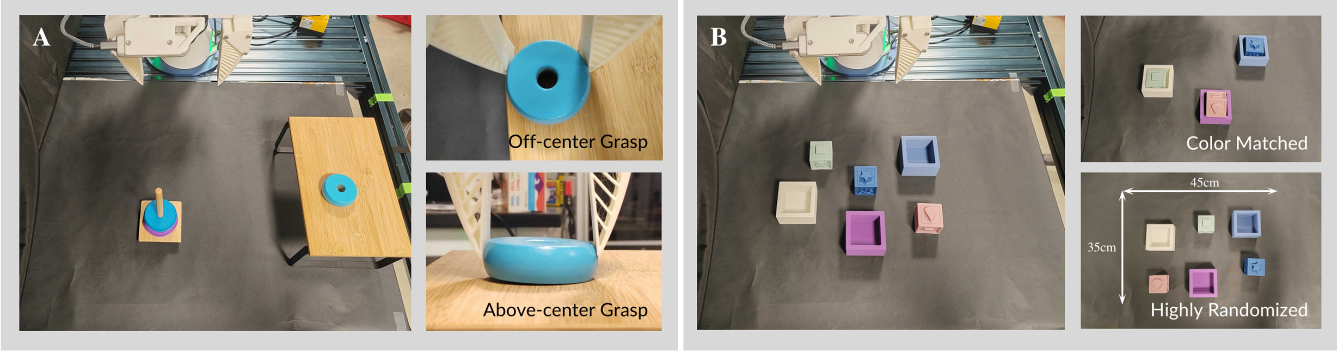


Fig. 5: The real-world experiments and their key challenges. Fig. A illustrates the tower of hanoi insertion task. In addition to the narrow tolerance required for insertion, grasping the disk itself is challenging. From the top view (top-right), the gripper must align precisely with the disk's center; otherwise, the disk slips out. From the side view (bottom-right), the gripper must also engage below the midpoint of the disk's curved edge. Fig. B presents the cube sorting task, where cubes of different colors must be placed into their corresponding containers. The six objects are randomly distributed within a 45 cm × 35 cm workspace, creating a highly randomized environment that significantly increases the difficulty of learning correct actions.

IV. Experiments

The experimental setups are illustrated in Fig. 1. We conducted a series of experiments to demonstrate the capabilities of the proposed framework to locate the states where the policy outputs abnormal actions and then collect targeted fine-grained demonstrations with different scaling factors. This therefore shows the data efficiency of the framework.

All policies are initialized with the same set of human demonstrations as a warmup. While more demonstrations are then added to the dataset of the base policy to train it again from scratch, we augment the dataset with our proposed Human-Robot Copilot and ensure that the total number of trajectories matches that of the base policy for a fair comparison.

A. Tasks

1) Simulation: We conducted two simulation experiments on the standard Robomimic benchmark [27] to validate the effectiveness of our proposed framework in a controlled reproducible environment. We choose PickPlace(Can) and NutAssembly(Square) tasks from the benchmark.

For the can pick-and-place task, precise grasping is required. If the gripper fails to clamp the can at its center, the cylindrical structure of the can leads to uneven force distribution, causing the object to gradually slip during transportation.

In the square nut assembly task, randomization of the nut's orientation and position increases the diversity of possible states, thereby requiring a larger amount of demonstrations to ensure sufficient coverage. Moreover, successful assembly demands highly precise placement, further compounding the difficulty of the task. In order to generate relatively easy demonstrations for the policy to learn, we adopted a two-stage procedure: first orienting the nut correctly, followed by grasping. In detail, if the nut was already in the right direction, a grasping action was directly executed. Otherwise, the nut was rotated incrementally—by up to 90 degrees per step—until it reached the correct orientation, after which the grasping action was performed. With this method, the policy does not need to learn to generate right grasp actions for all orientations of the nut.

2) Real World: For real world experiments, we designed a cube sorting task with a wide range of randomization to increase the likelihood of encountering out-of-distribution (OOD) states, as well as a tower of hanoi insertion task to evaluate the framework's capability in executing precise, contact-rich actions. The experimental setup and key challenges are illustrated in Fig.5

For the cube sorting task, we randomly placed the three cubes with different colors and three corresponding square containers in a 45cm × 35cm area. The robot is required to pick those cubes and then place them into the right container. Since the six objects have a wide range of randomized positions, there is a large amount of data needed to cover all the possible states.

For the tower of hanoi insertion task, the blue disk to be inserted is randomly placed on the table, while the tower itself is positioned along a line with a randomization range of 3 cm. Since the task is designed primarily for evaluating the ability of performing precise actions, we did not choose a wide range for randomization. The diameter of the tower's pole is 13.6 mm, whereas the diameter of the disk's central hole is 15.6 mm, resulting in a tolerance margin of only 2 mm for successful insertion. Under this condition, the 3 cm randomization is sufficient to ensure that the learned policy is genuinely reasoning about how to place the disk, rather than merely replaying the placement trajectories observed in the human demonstrations.

B. Experiment Results

The experiment results are reported in Table.I. To better illustrate the advantages of the proposed Human–Robot

Task	**Num of Traj**	**Base Policy**			**Proposed**		
		Stage 1	Stage 2	Total	Stage 1	Stage 2	Total
(a) Simulation							
Can Pick & Place	20 (warmup)	60	85	50	–	–	–
	40	90	75	**65**	85	75	60
Nut Assembly	20 (warmup)	30	85	25	–	–	–
	40	45	95	45	55	100	**55**
(b) Real World							
Cube Sorting	30 (warmup)	73	27	20	–	–	–
	40	83	43	33	70	90	**60**
Tower of Hanoi Insertion	30 (warmup)	70	80	50	–	–	–
	40	60	90	55	90	85	**75**

TABLE I: Comparison of baseline and our method across simulation and real-world tasks. The execution of each task is divided to two stages in evaluation. We reported the success rate (%) for each stage of each task to better analyze the failure conditions.

Task	Added Traj	Base Policy	Proposed
Can Pick-and-Place	20	501.7	189.8
Nut Assembly	20	713.7	313.3
Cube Sorting	10	493.7	284.5
Tower of Hanoi Insertion	10	279.6	174.6

TABLE II: Total time (seconds) required to collect augmentation data to the dataset for each task.

Copilot framework, we divide each task into two stages. The first stage ends when the manipulator successfully grasps the target object (i.e. can, nut, cube, disk), while the second stage ends upon completion of the full task. Notably, even the manipulator fails in first stage, we intervene to reposition it at the beginning of the second stage and still evaluate its success rate of the second stage.

The results indicate that the proposed human–robot copilot consistently outperforms the base policy across most tasks. Furthermore, Table.II demonstrates that the time required to collect corrective data is substantially lower than that needed to acquire full trajectories, highlighting the data efficiency of our framework.

In the following two sections, we analyze the factors contributing to the superior data efficiency of the proposed framework. We attribute this efficiency to two key aspects: the ability to accurately identify failure conditions, and the capability to perform fine-grained, concise corrective interventions.

C. Locating Failure Conditions

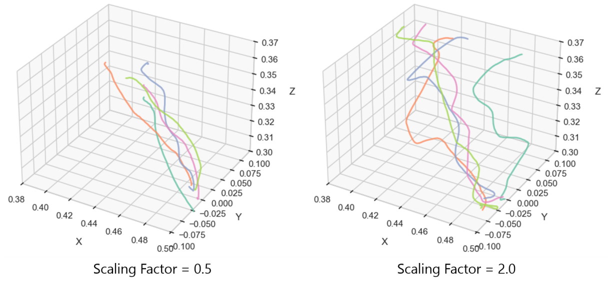
Through the simulation and real world experiments, we were able to identify the failure conditions of the base policy and collect targeted demonstrations. Notably, several of these failure cases were unexpected, indicating that augmenting the dataset with additional demonstrations without prior knowledge of such conditions may be insufficient to improve policy performance in these scenarios.

For the can pick-and-place task in simulation, the primary challenge lies in accurately clamping the can at its center. Failures observed during Stage 2 (placing the can in the target location) are primarily a consequence of imprecise grasps in Stage 1, which cause the can to slip from the gripper during transfer.

For the cube sorting task in real world experiments, as we illustrated in Fig.5, since there is a wide range of randomization of both the cubes and the containers, it is also obvious that the robot will struggle to grasp a cube or put it in the container that is located in a seldom visited place in human demonstrations. During deployment, the policy's difficulties effectively indicate the out-of-distribution (OOD) states, allowing the collected augmentation data to be targeted specifically at these challenging states rather than covering the entire state distribution.

While the failure modes of the two above tasks are quite easily to expect, the policy failed unexpectedly in the other two tasks. Both of the other two tasks require the policy to generate precise actions for assembly or insertion, however, from Table.I we can see that it is not difficult for the policy to successfully insert the nut or disk, but it is hard to pick them.

For the nut assembly task, although the rotation procedure simplifies the demonstrations and facilitates policy learning, it can also introduce unintended shifts of the nut, potentially leading to the OOD states for the policy. For the tower of hanoi insertion task, the near-cylindrical geometry of the disk requires the same precise actions as the can pick-and-place task in the simulation. Moreover, the curved side edges of the disk requires the gripper to grasp below its widest diameter,



Scaling Factor = 0.5

Scaling Factor = 2.0

Fig. 6: Human demonstrations collected under different scaling factors for tower of hanoi insertion. The intervention begins when the gripper approaches the tower(top left), ends when the insertion is completed(bottom right).

or the disk is prone to slipping during transfer.

D. Fine-grained Correction

When abnormal actions occur—such as attempting to grasp an object without proper alignment, failing to initiate the next action, or overshooting the intended target location—we intervene to provide corrective demonstrations that guide the policy.

For tasks requiring high precision, we observed that even human teleoperators face considerable challenges. For instance, in the Tower of Hanoi insertion task, the human teleoperator achieved only about a 40% success rate. While failed demonstrations can be discarded during the collection of training data for the base policy, failure cases encountered during base policy deployment cannot be reliably reproduced, hence the correction demonstration is expected to succeed in a single attempt. This highlights the need to refine the policy using more precise teleoperation, underscoring the importance of adopting a smaller scaling factor.

We illustrate the trajectories collected under different scaling factors in Fig.6. We intervene the follower when the gripper approaches the tower ather than when misalignment occurs to generate longer trajectories for better visualization. The results show that demonstrations collected with a larger scaling factor exhibit greater variance and often require multiple attempts to align the column with the disk.

During data augmentation procedure, with a smaller scaling factor (0.5 in our experiments), the corrective demonstrations achieved a higher success rate and fewer attempts for precise actions. This not only reduced the time cost of data collection but also resulted in more concise datasets, which in turn facilitated more effective policy learning.

V. Conclusion

In this paper, we proposed the Human-Robot Copilot framework that is compatible with a wide range of manipulators and leverages a scaling factor to enable dexterous teleoperation in high-precision tasks. Experimental results demonstrate that the human-in-the-loop data augmentation effectively identifies failure conditions and out-of-distribution (OOD) states, guiding the teleoperator to collect targeted demonstrations. Meanwhile, the scaling factor enhances the success rate of correction actions in precision-demanding scenarios, enabling concise and easily imitable actions. With these two advantages, the proposed framework achieves data-efficient learning, requiring the same number of demonstrations (and even less collection time) to outperform the baseline policy trained with traditional data collection methods.

Similar to how the quality of demonstrations affects policy performance in traditional data collection pipelines, the limitation of the proposed framework lies in the strategy for intervention and correction. Given the constraints of policy fine-tuning and the inability of ACT to handle multimodal demonstrations, correction actions should focus on refining and enhancing the existing policy (e.g., executing more precise grasps or placements) rather than introducing entirely new modes of execution (e.g., attempting to pick up the nut in all possible orientations). In the future, training an auxiliary policy with the correction data, or adopting a base policy capable of handling multimodality, may provide feasible solutions to these challenges.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 1, 2

[2] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto, "Behavior generation with latent actions," *arXiv preprint arXiv:2403.03181*, 2024. 1

[3] 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. 1, 2, 4

[4] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," *arXiv preprint arXiv:2410.13126*, 2024. 1, 2

[5] 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," in *Robotics: Science and Systems (RSS)*, 2023. 2

[6] 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," *arXiv preprint arXiv:2503.07771*, 2025. 2

[7] G. Yan, J. Zhu, Y. Deng, S. Yang, R.-Z. Qiu, X. Cheng, M. Memmel, R. Krishna, A. Goyal, X. Wang *et al.*, "Maniflow: A general robot manipulation policy via consistency flow training," *arXiv preprint arXiv:2509.01819*, 2025. 2

[8] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022. 2

[9] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183. 2

[10] S. Haldar, J. Pari, A. Rai, and L. Pinto, "Teach a robot to fish: Versatile imitation from one minute of demonstrations," *arXiv preprint arXiv:2303.01497*, 2023. 2

[11] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," *arXiv preprint arXiv:2109.08273*, 2021. 2

[12] 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. 2

[13] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback. 07 2020. doi: 10.15607/rss. 2020." 2

[14] 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)*. IEEE, 2019, pp. 8077–8083. 2

[15] J. Luo, C. Xu, J. Wu, and S. Levine, “Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning,” *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025. 2

[16] 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)*. IEEE, 2024, pp. 16 961–16 969. 2

[17] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, “Deep imitation learning for complex manipulation tasks from virtual reality teleoperation,” in *2018 IEEE international conference on robotics and automation (ICRA)*. Ieee, 2018, pp. 5628–5635. 2

[18] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, “Open-television: Teleoperation with immersive active visual feedback,” *arXiv preprint arXiv:2407.01512*, 2024. 2, 3, 4

[19] S. Dass, W. Ai, Y. Jiang, S. Singh, J. Hu, R. Zhang, P. Stone, B. Abbatematteo, and R. Martín-Martín, “Telemoma: A modular and versatile teleoperation system for mobile manipulation,” *arXiv preprint arXiv:2403.07869*, 2024. 2

[20] H. Li, Y. Cui, and D. Sadigh, “How to train your robots? the impact of demonstration modality on imitation learning,” *arXiv preprint arXiv:2503.07017*, 2025. 2

[21] Z. Fu, T. Z. Zhao, and C. Finn, “Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation,” *arXiv preprint arXiv:2401.02117*, 2024. 2

[22] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, “Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 156–12 163. 2

[23] A. Bazhenov, S. Satsevich, S. Egorov, F. Khabibullin, and D. Tsetserukou, “Echo: An open-source, low-cost teleoperation system with force feedback for dataset collection in robot learning,” *arXiv preprint arXiv:2504.07939*, 2025. 2

[24] J. J. Liu, Y. Li, K. Shaw, T. Tao, R. Salakhutdinov, and D. Pathak, “Factr: Force-attending curriculum training for contact-rich policy learning,” *arXiv preprint arXiv:2502.17432*, 2025. 2

[25] Y. Ze, Z. Chen, J. P. AraÃšjo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, “Twist: Teleoperated whole-body imitation system,” *arXiv preprint arXiv:2505.02833*, 2025. 2

[26] S. Yang, “Ace: A cross-platform visual-exoskeleton system for low-cost dexterous teleoperation,” Master’s thesis, University of California, San Diego, 2025. 3

[27] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, “What matters in learning from offline human demonstrations for robot manipulation,” in *arXiv preprint arXiv:2108.03298*, 2021. 5