
Genie Centurion: Accelerating Scalable Real-World Robot Training with Human Rewind-and-Refine Guidance

Wenhao Wang* Jianheng Song* Chiming Liu*

Jiayao Ma Siyuan Feng Jingyuan Wang Yuxin Jiang Kylin Chen

Sikang Zhan Yi Wang Tong Meng Modi Shi Xindong He

Guanghui Ren Yang Yang Maoqing Yao

AgiBot
genie-centurion.github.io

Abstract

While Vision-Language-Action (VLA) models show strong generalizability in various tasks, real-world deployment of robotic policy still requires large-scale, high-quality human expert demonstrations. However, passive data collection via human teleoperation is costly, hard to scale, and often biased toward passive demonstrations with limited diversity. To address this, we propose Genie Centurion (GCENT), a scalable and general data collection paradigm based on human rewind-and-refine guidance. When the robot execution failures occur, GCENT enables the system revert to a previous state with a rewind mechanism, after which a tele-operator provides corrective demonstrations to refine the policy. This framework supports a one-human-to-many-robots supervision scheme with a Task Sentinel module, which autonomously predicts task success and solicits human intervention when necessary, enabling scalable supervision. Empirical results show that GCENT achieves up to 40% higher task success rates than state-of-the-art data collection methods, and reaches comparable performance using less than half the data. We also quantify the data yield-to-effort ratio under multi-robot scenarios, demonstrating GCENT’s potential for scalable and cost-efficient robot policy training in real-world environments.

1 Introduction

In the field of robotics, the primary goal is to enable robots to perform productive tasks in real-world environments. Recent advances leverage large models, especially Vision-Language-Action (VLA) models [4, 5, 40, 28, 34, 52, 43], which can interpret high-level instructions and generate corresponding actions based on visual observations. Training such models typically requires large amounts of human demonstration data for imitation learning. However, obtaining large-scale and high-quality demonstrations remains a significant bottleneck for deploying powerful robotic systems in real-world scenarios.

Currently, the predominant imitation learning data collection approach is human teleoperation, which, although effective, is expensive and time-consuming [51, 10, 60]. Due to the direct one-to-one

*Equal contribution.

correspondence between human operation time and collected data, one hour of robot data typically requires at least one hour of skilled human operation. The actual efficiency is usually much lower than 1:1, considering additional overhead from environment setup, task resets, and human errors, significantly reducing the net usable data collected per operator hour. This inefficiency presents a major barrier to scaling imitation learning datasets for robotic systems.

Alternative approaches include sim-to-real transfer and offline RL [26, 37, 39, 30]. However, sim-to-real methods often suffer from the reality gap, especially in contact-rich and high-precision manipulation tasks, where reliable transfer is not guaranteed. Offline RL approaches, on the other hand, are limited by distribution shift and typically fail to cover critical regions such as failure and recovery states. The Dataset Aggregation (DAgger) algorithm [46] improves data efficiency by querying expert corrections in the states visited by the policy autonomous execution.

To address these challenges, we propose GCENT, a DAgger-inspired data collection framework tailored for efficient real-world robotic policy learning. In this paradigm, the human operator acts primarily as a guardian, intervening only when the policy fails or is about to fail. GCENT introduces a rewind mechanism that allows the operator to reset the robot to a recent valid state, thereby enhancing the diversity and coverage of critical state space.

Additionally, GCENT incorporates a Task Sentinel module, a vision-language-based model designed to autonomously detect task completion states and request human intervention as necessary. This progressively reduces reliance on continuous human supervision. We hypothesize that structured, failure-triggered human interventions significantly improve data efficiency and accelerate the optimization of robot policies, as compared to conventional continuous teleoperation methods.

Our primary contributions include:

1. We introduce Genie Centurion (GCENT), a unified framework designed for efficient robot policy training through corrective interventions triggered by failures, complemented by a rewind mechanism to enhance state-space coverage.
2. We perform extensive real-world experiments comparing GCENT to standard teleoperation-based data collection methods, demonstrating significant improvements in task success rates and substantially reduced human operational effort.
3. We propose the Task Sentinel module, which solicits human interventions selectively by predicting task completion over the policy autonomous execution process. Our experimental results demonstrate that Task Sentinel enables scalable supervision, allowing a single operator to effectively oversee multiple robots simultaneously.

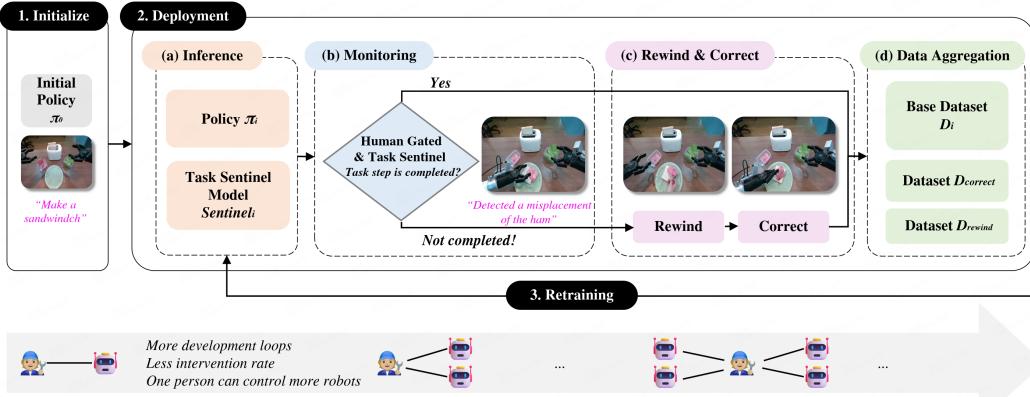


Figure 1: Overall GCENT System Pipeline. Illustrating the workflow, including initialization, deployment (inference, monitoring, rewind & correction, data aggregation), and retraining. As deployment iterations proceed, the single-robot success rate increases, and the intervention rate decreases, allowing a single operator to supervise more robots, thereby enhancing the efficiency of the data loop.

2 Related Work

2.1 Learning-Based Manipulation and the Role of Data

Learning-based approaches have demonstrated promising capabilities for robotic manipulation, particularly in multi-task, multi-modal, and open-ended instruction-following scenarios. Most existing methods follow adopt a two-stage training paradigm: large-scale pre-training followed by task-specific fine-tuning[29, 50]. In this setting, the performance heavily depends on the quality and diversity of the fine-tune data. Recent studies indicate that merely increasing data quantity is not sufficient for robust generalization; instead, factors such as coverage of failure cases, and diversity of environments and objects have a greater impact [31, 32]. To address this, some researches construct standard cross-embodiment dataset on various tasks [41, 53, 1], while others explore efficient collection strategies, such as compressing diverse spatial and linguistic exploration into minimal demonstrations [22]. However, most approaches still focus on repetitive collecting full trajectories and lack mechanisms to capture policy failed states systematically, limiting policy performance convergence efficiency.

2.2 Interactive Imitation Learning and Human-in-the-loop Supervision

To address the distribution shift in behavior cloning (BC) [59, 9], DAgger [46] introduces expert supervision during rollouts to iteratively aggregate the data distribution. SafeDAgger and LazyDAgger [58, 20] reduce human burden through safety prediction and switching cost modeling. HG-DAgger [27] uses human expert as a gating function and maintaining exclusive control authority. EnsembleDAgger [38] leverages an ensemble-based uncertainty estimation, while ThriftyDAgger [19] learns a budget-aware switching policy to triggers human interventions at high-risk states. While these works improve intervention strategies, they are mostly designed for simulation or driving contexts, with limited application to real-world manipulation. RoboCopilot [55, 56] emphasizing fluent control transfer, but still relying on constant human monitoring. Fleet-DAgger [21] and SIRIUS-FLEET [33] extend these strategies to multi-robot systems with supervision scheduling and prediction, but lack verifying on real-world complex tasks such as bimanual, long-horizon and high-precision tasks. In contrast to recent approaches using VQA-style success checking [35, 36, 13], we adopt a reward model-style value predictor head as Task Sentinel that independently determines whether each step has succeeded. This module is fine-tuned on a InternVL2.5-2B and can be deployed alongside the policy on real robots, ensuring both real-time inference and deployment stability.

2.3 Data Collection Interfaces and Shared Autonomy Systems

Learning-based robotic manipulation systems strongly depend on high-quality human demonstrations. Teleoperation remains the predominant data collection approach due to its intuitive control and high precision [11, 23]. Vision-based teleoperation systems [42, 8, 18] offer simplified setups but often induce operator fatigue due to absolute positional retargeting. Systems based on exoskeletons and motion capture [54, 14, 15, 3, 16] enhance teleoperation intuitiveness, but their reliance on extensive hardware setups reduces portability and ease of deployment. These devices have limited capability to switch between different modes. We use virtual reality as human-robot interfaces, enhanced ability to systematically capture structured demonstration data by multiple modes, especially failure cases and their corresponding corrective actions.

To alleviate human demonstration workload and enhance robot-operator collaboration, shared autonomy paradigms gain increasing attention. Early studies introduced adjustable authority to facilitate smooth human-robot collaboration[48, 47, 49, 44]. Later approaches improved shared autonomy through methods such as intent prediction and eye-hand behavior in single-robot scenarios [25, 12, 24, 2, 17]. Further research [45, 57] then explore the topic in multi-robot settings, to improved the efficiency of human supervision.

While prior approaches primarily focus on real-time shared control, our proposed method complements existing work by enabling corrections specifically at failure points, decoupled from the robot’s real-time execution. This strategy creates opportunities for scalable and targeted data collection.

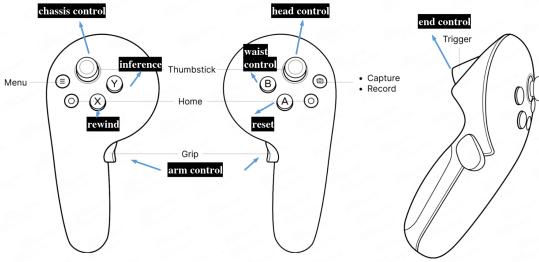


Figure 2: **Teleoperation Setup** Illustrating of button assignments for the dual 6-DoF controllers and their functions in GCENT.

3 GCENT Data Collection System

We designed and deployed an effective data collection system to support the GCENT learning paradigm. This section introduces the hardware configuration and the processing pipeline. Core functionalities like Task Sentinel, interactive rewind and correction mechanism are discussed in detail.

3.1 Hardware Setup and Teleoperation

The GCENT data collection system is built upon the AgiBot G01[1] robotic platform. The operator station utilizes a VR system, where two 6-DoF VR controllers are used to independently control the robot's dual-arm end-effectors and to execute operations such as intervention and rewind. This intuitive and precise hardware interface enables operators to effectively monitor and intervene during policy rollouts.

As shown in Figure 2, the system assigns different operations for each key-button. The Y button initiates the inference mode, starts default policy execution on the robot. The X button triggers the rewind mode, restoring the system to a previous time point. The side gripper initiates the takeover mode, enables manual control for human demonstration or correction. The A button reset the system, returns the robot to initial pose and ends the current data collection. Other buttons are assigned to control rest components of the robot's body.

3.2 Data Processing Pipeline

The GCENT data collection system operates within a continuously iterating data loop. It is designed to progressively refine the policy model through online interaction. The goal is to gradually reduce the intervention rate and increase the success rate. The core process is illustrated in Figure 1.

1. **Initialization:** A small set of seed data, D_0 , is collected through human teleoperation to train an initial policy π_0 .
2. **Deployment:** This stage includes four key steps:
 - (a) **Inference:** The robot performs tasks autonomously using the current policy π_i and the Task Sentinel model Sentinel_i .
 - (b) **Monitoring:** The system determines whether the current task step is complete based on both human supervision and signals from the Task Sentinel model (detailed in Section 3.3). If completed, it proceeds to the next step; otherwise, a rewind or intervention is requested.
 - (c) **Rewind and Correction:** This is a core interactive capability on GCENT, detailed in Section 3.4 (highlighted in pink in Figure 1). It enables state restoration and corrective demonstration.
 - (d) **Data Aggregation:** After completion or correction of the task, the effective trajectory data, particularly the successful corrective trajectories D_{correct} from step (2c), are aggregated in the dataset D_{i+1} .
3. **Retraining:** The updated dataset D_{i+1} is then used to fine-tune both policy model and Task Sentinel model, yielding new versions π_{i+1} and Sentinel_{i+1} . These updated models are then



Figure 3: **Labels in sandwich assembly task.** It shows the mode transitions among four core interaction modes: “Intervention”, “Rewind”, “Inference” and “Awaiting Intervention”. The first five plots were in mode “Inference”. In plot 6, GCENT detected an incorrect grasp when the robot attempted to pick up the ham, and thus transitioned to “Awaiting Intervention”. The plot 7 shows the “Intervention” mode when the operator pressed the side button and the policy was paused. Afterwards, the operator controlled the robot to grasp the ham correctly. Subsequently, plot 9 shows the operator put the robot back to the “Inference” mode for autonomous operation by pressing the Y button. In plot 12, the Task Sentinel detected a misplacement of the ham and requested intervention from the operator. The operator then long-pressed the X button to rewind the robot to an appropriate historical state. Then the operator controlled the robot to grasp the ham again, and transitioned the robot back to “Inference”. In the end the robot proceeded to pick lettuce and another slice of bread to complete the entire task.

deployed on the robot, and the deployment cycle (step 2) is further repeated. This iterative process continues until the robot can complete tasks and reliably monitor the task status autonomously.

Our data pipeline logs real-time data on the robot, including multi-view observations (o_t), joint states (joints_t), policy actions, instructions, and other mode labels. This data is first saved locally in HDF5 format, then validated, and uploaded to the cloud. Cloud services further process the data, including frame alignment, step labeling, storage management, and more, before model training begins.

The GCENT data collecting system automatically logs precise mode labels to the data stream. There are four modes: “Intervention”, “Rewind”, “Inference”, and “Awaiting Intervention”. The trajectory segments labeled “Intervention”, denoted as $\{(o_\tau, a_\tau^{\text{human}})\}$, are automatically identified as high quality supervised samples for training π_i . The “Awaiting Intervention” mode is autonomously determined by the Sentinel. This fine-grained labeling enables GCENT to efficiently capture real-world interaction data that are crucial to addressing model weaknesses.

3.3 Task Sentinel: A Multimodal Large Language Model-based Robot Step Detection Model

In addition to human-gated intervention, we designed an autonomous mechanism, the Task Sentinel, for robot to determine when intervention is necessary. This model, inspired by the reward model architecture in [6], takes the current image observation o_t and task instruction l_{task} as input at time t . As shown in Equation 1, the model outputs a boolean value z_t indicating the completion status of the current step:

$$z_t := \text{Sentinel}(o_t, l_{\text{task}}) \in \{0, 1\} \quad (1)$$

When the step is completed, denoted as $z_t = 1$, the robot automatically proceeds to next step. If it is not completed within a predefined time T_{\max} (i.e., $z_t = 0$ and $\Delta t > T_{\max}$), GCENT transitions to the “Awaiting Intervention” mode until a human operator intervenes, as described by the condition:

$$\text{if } (z_t = 0 \text{ and } \Delta t > T_{\max}), \text{ then request human intervention.} \quad (2)$$

Unlike methods similar to DAgger [46], we chose not to explicitly detect exact failure moments. This decision stems from the significant challenge of accurately identifying failure instances, particularly with limited data. The types and frequencies of errors can vary at different stages of model training. In contrast, the definition of successful task (or sub-task) completion remains clear and stable, rendering it more suitable for training robust models under the GCENT paradigm.

The Task Sentinel model constantly monitors the task execution. Tasks are decomposed into atomic action steps (e.g., grasp, place, push, press), and human annotators identify the start and end frames

for each step. We designate the frames within the final second of a successfully executed step as completed, while all other frames as not completed. It converts the training of Task Sentinel model into a binary classification task.

During deployment, we combine both Task Sentinel and human supervision (Human-Gated mode) to enhance data collection efficiency in GCENT. In Human-Gated mode, an operator continuously monitors the robot’s actions, and intervenes immediately if any error occurs. As depicted in Figure 1, during the early stages of GCENT iteration, data collectors primarily use the Human-Gated method on a single robot due to low model success rate. As iterations progress and model performance improves, the Task Sentinel mechanism enables a single operator to monitor multiple robots simultaneously, requesting intervention only when necessary. Task sentinel is a key factor for scaling the GCENT approach to a one-operator-multiple-robots system, significantly improving operational efficiency and system safety.

3.4 Rewind and Correction Mechanism

As depicted in Figure 3, when the Task Sentinel requests intervention, or if the operator decides so, the operator can press the X button to trigger a rewind operation. The system maintains a real-time state buffer of the past 3 seconds on the robot itself. When rewind mode is initiated, the system restores the robot to the selected historical state s_{t-k} . After state rewind, the operator can apply physical perturbations, and provide a correction demonstration. The rewind and correction mechanism is a core interactive feature of GCENT, enabling efficient recovery and precise corrections.

4 Experiments

This section demonstrates that GCENT serves as an efficient and cost-effective data collection strategy, capable of achieving significant performance improvements across diverse real-world tasks with limited data. Specifically, we addresses the following questions:

- **Q1:** Can GCENT achieve superior performance compared to alternative data collection strategies?
- **Q2:** Can GCENT improve data efficiency and reduce human supervision costs, enabling one-to-many robot supervision?
- **Q3:** How do different rewind strategies impact policy performance?

4.1 Experimental Setup

We fine-tune the policy π from the GO-1 embodied foundation model [1], pre-train on thousands of hours of robot manipulation data. All data collection and evaluation are performed on the AgiBot G01 platform. Task Sentinel model is built upon the InternVL 2.5 2B backbone [7] with an additional MLP-based binary classifier head.

The policy π is trained using only the human intervention segments from GCENT data, while Task Sentinel is trained on the full GCENT dataset. We fine-tune the model for 100 epochs using 1 A800 node in each iteration, which takes approximately 16 hours. To ensure practical relevance and task diversity, we designed four tasks based on real-world applications.

- **Sandwich Assembly:** Eight sequential bi-manual pick-and-place actions to stack ingredients into a sandwich.
- **Connector Insertion:** Grasp and insert a component into a tight terminal, requiring fine contact-rich control.
- **Microwave-Heating:** Completing a microwave heating task requires five different atomic operations including pull, pick, place, push, and press.
- **Typing:** Type out text using a small keyboard, including back-space handling for instruction-following evaluation.

We adopt a batched DAgger-style iteration [55]. Each task starts with 20 trajectories via passive data collection, followed by 4 GCENT iterations, one of which includes 20 demonstrations, until

the average score surpasses 0.9. We compare three strategies under equal data volume: (1) Passive Data Collection (PDC), (2) Adversarial Data Collection (ADC), and (3) GCENT. Tasks completed 10 trials with average scores as performance metrics. The 1.0 score indicates complete success, while partial completions received proportional values. This approach offers more detailed measurement than binary success rates.

4.2 Q1: Can GCENT Lead to Better Performance?

Table 1: Comparison of average scores across data collection methodologies for four tasks.

Task	Traj.	Passive	Adversarial	GCENT (Ours)
Sandwich Assembly	20 (warmup)	0.28 ± 0.06	0.23 ± 0.04	-
	40	-	-	0.69 ± 0.07
	60	0.30 ± 0.03	0.59 ± 0.14	0.79 ± 0.08
	80	-	-	0.76 ± 0.05
	100	0.45 ± 0.02	0.53 ± 0.15	0.81 ± 0.04
	120	-	-	0.91 ± 0.01
Connector Insertion	20 (warmup)	0.00 ± 0.00	0.64 ± 0.07	-
	40	0.10 ± 0.09	0.75 ± 0.08	0.64 ± 0.09
	60	0.40 ± 0.15	0.82 ± 0.09	0.90 ± 0.05
Microwave-Heating	20 (warmup)	0.36 ± 0.07	0.48 ± 0.05	-
	40	-	-	0.74 ± 0.06
	60	0.48 ± 0.06	0.72 ± 0.09	0.89 ± 0.03
	80	0.55 ± 0.05	0.76 ± 0.11	0.97 ± 0.01
Typing	20 (warmup)	0.11 ± 0.05	0.10 ± 0.07	-
	40	-	-	0.26 ± 0.08
	60	0.18 ± 0.10	0.05 ± 0.03	0.78 ± 0.09
	80	-	-	0.85 ± 0.06
	100	0.10 ± 0.05	0.03 ± 0.02	0.95 ± 0.03
Average	20 (warmup)	0.19	0.36	-
	60	0.26	0.48	0.84
	Final Round	0.38	0.53	0.93

Table 1 shows that GCENT achieves the highest performance across all tasks, with an average final score of 0.93, 55% improvement over passive methods and 40% over ADC under identical data budgets. All tasks achieve an average score exceeding 0.9 within 3-5 GCENT rounds. In the most challenging task, sandwich assembly, GCENT reaches 0.91 in the fifth round, while microwave and connector tasks converge earlier. The GCENT approach demonstrated superior efficiency compared to both passive and adversarial collection methods, requiring significantly fewer data samples while simultaneously achieving a higher performance ceiling. In the typing task, when other methods plateaued, GCENT still maintained a 0.95 score through its superior sampling strategy.

Passive Data Collection (PDC) assume all samples are equally important. However, in robotic manipulation, task success often depends on critical moments, like precise alignment or following instructions, that make up only a small part of the data but have a huge impact. These crucial samples are poorly learned by conventional methods.

Adversarial Data Collection (ADC) addresses this limitation by introducing adversarial dynamics during the collection process, thereby improving learning efficiency on critical samples. However, manually crafted failures provide limited coverage and inadequately capture the diverse error patterns encountered during policy deployment. Furthermore, as policies evolve, the distribution of critical failure states shifts dynamically, reducing the effectiveness of static adversarial scenarios over time. In contrast, GCENT employs autonomous policy execution to identify critical samples dynamically, adjusting their representation in the dataset through rewind and invention mechanisms. This approach yields a higher-quality dataset that prioritizes learning from the most informative failure states, resulting in substantial performance improvements across diverse tasks.

Empirical results show that GCENT achieves the most significant performance improvements during the initial iteration, while subsequent rounds tend to plateau or even exhibit slight regressions. This trend may be attributed to limitations in the evaluation protocol, which struggles to capture subtle improvements across iterations. This behavior reveals an important difference between artificial

failure cases and real errors that arise during model execution: naturally occurring errors provide more valuable learning signals. With continued training, GCENT models are able to overcome these intermediate plateaus and ultimately converge to consistently high performance levels.

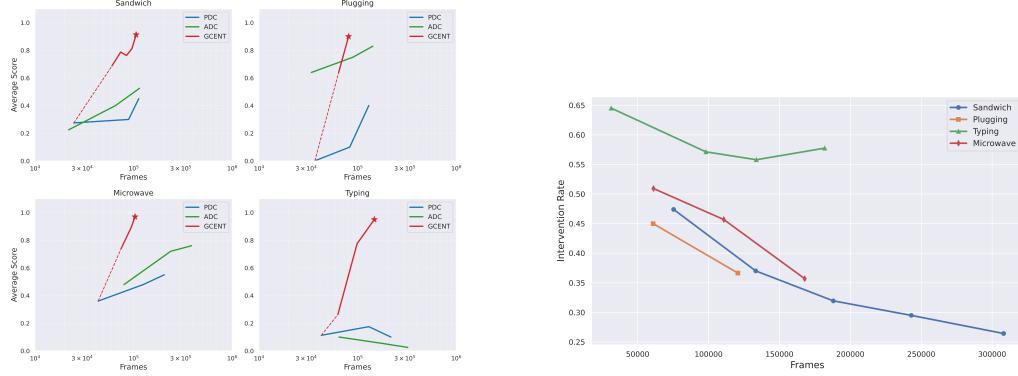


Figure 4: Comparison of data efficiency across methods. GCENT achieves 0.9+ task score with significantly fewer frames. At the same frame count, GCENT improves model performance by an average of 40%; at the same performance level, GCENT requires only 44.5% of the frames compared to passive data collection on average.

Figure 5: Intervention rates decreased consistently across all tasks as iteration rounds progressed. The typing task, however, exhibited a slight rebound in the final round specifically, as rewind mechanism proportions were intentionally increased only at this terminal stage.

4.3 Q2: Can GCENT Improve Data Efficiency?

Figure 4 demonstrates that GCENT achieves comparable or superior performance using only 44.5% of the frames required by alternative methods. Notably, in the instruction-following task (typing), GCENT attains performance scores exceeding 0.9 with merely 30k frames. Despite utilizing only human intervention segments for training, GCENT surpasses other approaches due to the enhanced quality of data.

As shown in Figure 5, the human intervention rate, defined as the proportion of frames requiring intervention, decreases significantly over successive GCENT iterations. This reflects continuous policy improvement and a reduced need for human oversight, enabling more efficient parallel supervision of multiple robots.

To assess the scalability of the GCENT framework in single-operator-multi-robot configurations, we conducted a dual-robot experiment with Task Sentinel assistance. As presented in Table 2, we evaluated policy models at 40%, 60%, and 80% success rates on the sandwich assembly task, measuring both intervention rates and collection efficiency.

Collected frames represent the combined data collected from both robots. Paused frames measure the waiting time from Task Sentinel’s intervention request to operator response. Collection efficiency is calculated by converting the effective frames (collected frames minus paused frames) into a human efficiency ratio, with a maximum of 2.0 for single-operator-dual-robot operation.

Experimental results demonstrate that despite a low model success rate of 40%, the system still achieves a collection efficiency of 1.86, which increases progressively with improved policy performance. This finding establishes that GCENT enhances both data efficiency and quality while substantially reducing human labor costs. These results underscore GCENT’s potential for scalable multi-robot supervision and establish a foundation for future extensions to fully concurrent single-operator-N-robot deployments through systematic optimization.

4.4 Q3: How do rewind strategies make impact?

We observed that the rewind mechanism produces different effects at various stages of GCENT. During initial stages, direct intervention without rewinding enables the model to develop failure recovery capabilities, thereby enhancing overall task performance. In subsequent stages, after

Table 2: Comparison of data collection efficiency across models with varying success rates under a one-human-two-robot setup in the sandwich task.

Metric	40% Success Rate	60% Success Rate	80% Success Rate
Intervention Rate (%)	47	39	27
Collected Frames	52,197	61,639	48,523
Paused Frames	3,746	2,931	1,831
Collection Efficiency	1.86	1.90	1.92

achieving higher success rates, activating the rewind mechanism substantially improves first-attempt success rates by eliminating redundant error sequences and refining trajectory quality.

To validate our findings, we conducted controlled experiments on the sandwich assembly task using models with initial success rates of 20% and 80%. We compared two intervention strategies: (1) *Direct Intervention* and (2) *Rewind*. As presented in Table 3, the results strongly support our observations: during early stages, direct intervention strategy enables models to learn from corrective demonstrations and enhance robustness through failure recovery mechanisms. In later stages, the rewind strategy facilitates optimal trajectory learning, reduces error attempts, and improves both efficiency and final performance.

Table 3: Comparison of rewind strategies across training stages in the sandwich assembly task. Direct intervention demonstrates superior performance during early stages, while rewind enhances performance in later stages.

Rewind Strategy	Start at 20%	Start at 80%
Direct Intervention	0.69 ± 0.07	0.84 ± 0.03
Rewind	0.50 ± 0.06	0.91 ± 0.01

5 Conclusion

Training high success rate and deployable real-world robot policies, particularly Vision-Language-Action (VLA) models, faces a major bottleneck in data collection. Passive Data Collection is costly, inefficient, and difficult to scale. To address this, we introduce GCENT, a scalable training paradigm for real-world robot policy deployment. GCENT introduces Human Rewind-and-Refine Guidance, of which human operators intervene only upon failure, and a rewind mechanism restores the robot to a valid prior state, to collect corrective demonstrations focused on failure recovery.

Empirical results show that GCENT improves both data efficiency and final task success rates by over 40% compared to state-of-the-art collection methods. We further propose a Task Sentinel mechanism that allows the model to autonomously detect potential failures and proactively request human intervention, thereby significantly reducing the need for laborious human monitoring. As the policy improves, the frequency of intervention declines, ultimately enabling efficient 1-to-N supervision, where a single human can oversee multiple robots. This is crucial for a scalable data collection system.

In summary, GCENT provides a practical, efficient, and scalable framework for training high-performance, deployable robot policies. Its scalability based on Task Sentinel has the potential to significantly lower the cost of robot learning and accelerate the real-world deployment of intelligent robotic systems.

Limitations. While our current framework is largely automated, manual involvement is still required for step annotation and verification. Our multi-robot control system presents operational challenges, requiring complex initialization procedures before each teleoperation session. Future work will develop more user-friendly systems for larger-scale deployments. Additionally, we have not thoroughly investigated the failed trajectories collected by Genie Centurion; future studies will utilize post-training algorithms to better leverage these negative examples.

References

- [1] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengan Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. URL <https://arxiv.org/abs/2503.06669>.
- [2] Reuben M. Aronson, Thiago Santini, Thomas C. Kübler, Enkelejda Kasneci, Siddhartha Srinivas, and Henny Admoni. Eye-hand behavior in human-robot shared manipulation. In *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 4–13, 2018.
- [3] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025. URL <https://arxiv.org/abs/2502.13013>.
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.
- [5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Chormanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricu, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.
- [6] Zheng Cai, Maosong Cao, Haojong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhihai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024. URL <https://arxiv.org/abs/2403.17297>.

- [7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.
- [8] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Tele-operation with immersive active visual feedback. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.
- [9] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL <https://arxiv.org/abs/2303.04137>.
- [10] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [11] Shivin Dass, Wensi Ai, Yuqian Jiang, Samik Singh, Jiaheng Hu, Ruohan Zhang, Peter Stone, Ben Abbatematteo, and Roberto Martín-Martín. Telemoma: A modular and versatile teleoperation system for mobile manipulation. *arXiv preprint arXiv:2403.07869*, 2024. URL <https://arxiv.org/abs/2403.07869>.
- [12] Anca Dragan and Siddhartha Srinivasa. A policy blending formalism for shared control. *International Journal of Robotics Research*, 32(7):790 – 805, June 2013.
- [13] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors, 2023. URL <https://arxiv.org/abs/2303.07280>.
- [14] Hongjie Fang, Hao-Shu Fang, Yiming Wang, Jieji Ren, Jingjing Chen, Ruo Zhang, Weiming Wang, and Cewu Lu. Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild. *arXiv preprint arXiv:2309.14975*, 2023. URL <https://arxiv.org/abs/2309.14975>.
- [15] Hongjie Fang, Chenxi Wang, Yiming Wang, Jingjing Chen, Shangning Xia, Jun Lv, Zihao He, Xiyan Yi, Yunhan Guo, Xinyu Zhan, Lixin Yang, Weiming Wang, Cewu Lu, and Hao-Shu Fang. Airexo-2: Scaling up generalizable robotic imitation learning with low-cost exoskeletons. *arXiv preprint arXiv:2503.03081*, 2025. URL <https://arxiv.org/abs/2503.03081>.
- [16] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024. URL <https://arxiv.org/abs/2401.02117>.
- [17] Deepak Gopinath, Siddarth Jain, and Brenna D. Argall. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robotics and Automation Letters*, 2(1):247–254, 2017. doi: 10.1109/LRA.2016.2593928.
- [18] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision based teleoperation of dexterous robotic hand-arm system. *arXiv preprint arXiv:1910.03135*, 2019. URL <https://arxiv.org/abs/1910.03135>.
- [19] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. In *Conference on Robot Learning (CoRL)*, 2021. URL <https://arxiv.org/abs/2109.08273>.
- [20] Ryan Hoque, Ashwin Balakrishna, Carl Puterman, Michael Luo, Daniel S. Brown, Daniel Seita, Brijen Thananjeyan, Ellen Novoseller, and Ken Goldberg. Lazydagger: Reducing context switching in interactive imitation learning. In *Proceedings of the IEEE Conference on Automation Science and Engineering (CASE)*, 2021.

- [21] Ryan Hoque, Lawrence Yunliang Chen, Satvik Sharma, Karthik Dharmarajan, Brijen Thananjeyan, Pieter Abbeel, and Ken Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In *Conference on Robot Learning (CoRL)*, 2022. URL <https://arxiv.org/abs/2206.14349>.
- [22] Siyuan Huang, Yue Liao, Siyuan Feng, Shu Jiang, Si Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Adversarial data collection: Human-collaborative perturbations for efficient and robust robotic imitation learning. *arXiv preprint arXiv:2503.11646*, 2025. URL <https://arxiv.org/abs/2503.11646>.
- [23] Aadithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *Proceedings of Machine Learning Research*, 270:2372–2395, 2024. ISSN 2640-3498. Publisher Copyright: © 2024 Proceedings of Machine Learning Research.; 8th Conference on Robot Learning, CoRL 2024 ; Conference date: 06-11-2024 Through 09-11-2024.
- [24] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming, 2017. URL <https://arxiv.org/abs/1706.00155>.
- [25] Hong Jun Jeon, Dylan P. Losey, and Dorsa Sadigh. Shared autonomy with learned latent actions, 2020. URL <https://arxiv.org/abs/2005.03210>.
- [26] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
- [27] Michael Kelly, Chelsea Sidrane, Katherine Rose Driggs-Campbell, and Mykel J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. *CoRR*, abs/1810.02890, 2018. URL <http://arxiv.org/abs/1810.02890>.
- [28] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [29] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL <https://arxiv.org/abs/2502.19645>.
- [30] Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiro Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials, 2023. URL <https://arxiv.org/abs/2210.05178>.
- [31] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024. URL <https://arxiv.org/abs/2410.18647>.
- [32] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Data-efficient fine-tuning for llm-based recommendation. *arXiv preprint arXiv:2401.17197*, 2024. URL <https://arxiv.org/abs/2401.17197>.
- [33] Huihan Liu, Yu Zhang, Vaarij Betala, Evan Zhang, James Liu, Crystal Ding, and Yuke Zhu. Multi-task interactive robot fleet learning with visual world models. In *8th Conference on Robot Learning (CoRL)*, 2024. URL <https://arxiv.org/abs/2410.22689>.
- [34] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.
- [35] Fiona Luo. Vision-language models for robot success detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21):23750–23752, Mar. 2024. doi: 10.1609/aaai.v38i21.30552. URL <https://ojs.aaai.org/index.php/AAAI/article/view/30552>.

- [36] Yecheng Jason Ma, Joey Hejna, Ayzaan Wahid, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, Jonathan Tompson, Osbert Bastani, Dinesh Jayaraman, Wenhao Yu, Tingnan Zhang, Dorsa Sadigh, and Fei Xia. Vision language models are in-context value learners, 2024.
- [37] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.
- [38] Kunal Menda, Katherine Rose Driggs-Campbell, and Mykel J. Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. *CoRR*, abs/1807.08364, 2018. URL <http://arxiv.org/abs/1807.08364>.
- [39] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. *arXiv preprint arXiv:2504.13059*, 2025.
- [40] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yun-liang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.
- [41] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guiest, Soroush Nasiriany, Stefan Schaaf, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay

- Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.
- [42] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
 - [43] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025. URL <https://arxiv.org/abs/2501.15830>.
 - [44] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Shared autonomy via deep reinforcement learning, 2018. URL <https://arxiv.org/abs/1802.01744>.
 - [45] Ariel Rosenfeld, Noa Agmon, Oleg Maksimov, and Sarit Kraus. Intelligent agent supporting human–multi-robot team collaboration. *Artificial Intelligence*, 252:211–231, 2017. ISSN 0004-3702. doi: <https://doi.org/10.1016/j.artint.2017.08.005>. URL <https://www.sciencedirect.com/science/article/pii/S0004370217301029>.
 - [46] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. *CoRR*, abs/1011.0686, 2010. URL <http://arxiv.org/abs/1011.0686>.
 - [47] P. Scerri, D. V. Pynadath, and M. Tambe. Towards adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17:171–228, September 2002. ISSN 1076-9757. doi: 10.1613/jair.1037. URL <http://dx.doi.org/10.1613/jair.1037>.
 - [48] B. Sellner, F.W. Heger, L.M. Hiatt, R. Simmons, and S. Singh. Coordinated multiagent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE*, 94(7):1425–1444, 2006. doi: 10.1109/JPROC.2006.876966.
 - [49] Thomas B. Sheridan. Adaptive automation, level of automation, allocation authority, supervisory control, and adaptive control: Distinctions and modes of adaptation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(4):662–667, 2011. doi: 10.1109/TSMCA.2010.2093888.
 - [50] Arjun Singh, Nikhil Pandey, Anup Shirgaonkar, Pavan Manoj, and Vijay Aski. A study of optimizations for fine-tuning large language models, 2024. URL <https://arxiv.org/abs/2406.02290>.
 - [51] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024. URL <https://arxiv.org/abs/2403.07788>.
 - [52] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ran Cheng, Chaomin Shen, Yixin Peng, Feifei Feng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024.
 - [53] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, Shichao Fan, Xinhua Wang, Fei Liao, Zhen Zhao, Guangyu Li, Zhao Jin, Lecheng Wang, Jilei Mao, Ning Liu, Pei Ren, Qiang Zhang, Yaoxu Lyu, Mengzhen Liu, Jingyang He, Yulin Luo, Zeyu Gao, Chenxuan Li, Chenyang Gu, Yankai Fu, Di Wu, Xingyu Wang, Sixiang Chen, Zhenyu Wang, Pengju An, Siyuan Qian, Shanghang Zhang, and Jian Tang. Robomind: Benchmark on multi-embodiment intelligence

- normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. URL <https://arxiv.org/abs/2412.13877>.
- [54] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. *arXiv preprint arXiv:2309.13037*, 2023. URL <https://arxiv.org/abs/2309.13037>.
 - [55] Philipp Wu, Yide Shentu, Qiayuan Liao, Ding Jin, Menglong Guo, Koushil Sreenath, Xingyu Lin, and Pieter Abbeel. Robocopilot: Human-in-the-loop interactive imitation learning for robot manipulation, 2025. URL <https://arxiv.org/abs/2503.07771>.
 - [56] Zhiyuan Xu, Yinuo Zhao, Kun Wu, Ning Liu, Junjie Ji, Zhengping Che, Chi Harold Liu, and Jian Tang. Hacts: a human-as-copilot teleoperation system for robot learning, 2025. URL <https://arxiv.org/abs/2503.24070>.
 - [57] Sebastián A. Zanlongo, Peter Dirksmeier, Philip Long, Taskin Padir, and Leonardo Bobadilla. Scheduling and path-planning for operator oversight of multiple robots. *Robotics*, 10(2):57, 2021. URL <https://doi.org/10.3390/robotics10020057>.
 - [58] Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving, 2016. URL <https://arxiv.org/abs/1605.06450>.
 - [59] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL <https://arxiv.org/abs/2304.13705>.
 - [60] Zhaxizhuoma, Kehui Liu, Chuyue Guan, Zhongjie Jia, Ziniu Wu, Xin Liu, Tianyu Wang, Shuai Liang, Pengan Chen, Pingrui Zhang, Haoming Song, Delin Qu, Dong Wang, Zhigang Wang, Nieqing Cao, Yan Ding, Bin Zhao, and Xuelong Li. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset. *arXiv preprint arXiv:2409.19499*, 2024. URL <https://arxiv.org/abs/2409.19499>.

A Hardware Setup

We used the Genie-1 general-purpose humanoid robot for all experiments. The robot has 20 degrees of freedom, including 7-DoF arms capable of handling up to 5 kg per arm, and a waist joint for pitch and vertical motion. The arms offer high-precision control with ± 0.1 mm repeatability, enabling reliable execution of long-horizon, bimanual tasks.

Each end-effector is equipped with one RGB-D camera or one fisheye camera, and a six-axis force sensor. The head includes an additional RGB-D and fisheye camera array, providing full-scene perception.

B Training Details

B.1 Hyperparameter

Table 4 summarizes the key hyperparameters used in our experiments.

Hyperparameter	Value
Learning rate	2e-5
Batch size	16×8
Input image size	$3 \times 224 \times 224$
Weight decay	0.01
Action chunk	30
Infer denoise timesteps	5
Train denoise timesteps	1000

Table 4: Key hyperparameters used in our training process.

B.2 Data Augmentation

During data preprocessing, we enhanced the input images through three augmentation strategies:

1. **Color Jitter:** We stochastically adjusted brightness (± 0.3), contrast (± 0.4), saturation (± 0.5), and hue (± 0.03) with 50% probability to enhance the model’s invariance to illumination and color variations.
2. **Noise and Blur:** We applied random Gaussian, Laplacian, or Poisson noise (intensity: 0-5% of pixel values) with 50% probability. Additionally, we implemented average blur (kernel size: 2-7 pixels) with 50% probability to simulate sensor noise and motion artifacts encountered in real-world deployments.
3. **Image Dropout:** We implemented a controlled dropout mechanism wherein specific visual inputs (head, left, and right camera images) were replaced with uniform color fields based on ImageNet means at 10% probability, simulating sensor failures and encouraging robust feature extraction under incomplete observational conditions.

These augmentation techniques enhanced the model’s generalizability by increasing data diversity, simulating environmental perturbations, and enforcing adaptation to partial information scenarios which is a critical factors for robust performance in real-world applications.

B.3 Fine-tune Details

To optimize computational efficiency and accelerate convergence, we implemented three methodological refinements to the training pipeline. First, we computed inter-frame joint angle differentials and established a minimal motion threshold ($\pi/180/30$ radians). Frames exhibiting sub-threshold angular displacement were classified as static and subsequently excluded from the training data.

Second, we transformed raw joint angle measurements into end-effector pose representations. Specifically, we derived the differential pose between consecutive frames based on forward kinematics.



Figure 6: Sequence of four real-world manipulation tasks demonstrated in our experiments: (a) sandwich assembly, (b) connector insertion, (c) microwave-heating, and (d) typing.

This relative pose-based representation provided more precise action parameterization for robotic manipulation tasks.

Finally, we implemented dimension-wise min-max normalization on all action components, constraining values to the $[-1, 1]$ interval. This standardization mitigated the adverse effects of heterogeneous scaling across action dimensions, promoting gradient stability during backpropagation and facilitating more efficient optimization dynamics throughout the training process.

C Task Visualization

Visual examples of the four tasks are shown in Figure 6.

Sandwich Assembly: The robot must use two slices of bread, a piece of bacon, and a leaf of lettuce to assemble a sandwich. The entire process involves eight sequential pick-and-place operations that require bimanual coordination. The ingredients must be stacked neatly in the correct order. This task primarily evaluates the robot’s ability to perform long-horizon planning and dual-arm manipulation.

Connector Insertion: The robot is tasked with picking up a hardware component and precisely inserting it into a small connector at the correct angle. This task tests the model’s proficiency in fine-grained manipulation under rich contact conditions.

Microwaving Heating: The robot needs to open a microwave, place the food item inside, close the door, and press the heating button. The task involves executing a precise sequence of five actions—pulling, grasping, placing, pushing, and pressing—all using both arms. It assesses the robot’s bimanual manipulation capabilities in structured multi-step scenarios.

Typing Task: The robot must type a given user input on a compact keyboard, sequentially pressing keys corresponding to the characters A, G, I, B, O, T, and the spacebar. If it makes a mistake, it must press the delete key to correct the error. This task focuses on evaluating the model’s ability to follow instructions accurately and perform precise, symbolic actions.