Teleoperation-Based Data Collection and Automatic Annotation for Embodied Robotics

Qiuqi Xu[1], Haibao Liu[2] and Fei Zhang[3]

***Abstract*— Generalized embodied agents train on large-scale datasets spanning diverse robotic embodiments, yet existing teleoperation solutions are tailored to specific devices, challenging to scale, and suboptimal for synchronized multi-robot data collection. To address these limitations, we propose a teleoperation platform that supports heterogeneous robotic systems through a unified input normalization layer and MQTT-based coordination framework. Our platform concurrently supports four teleoperating and six teleoperated devices with time-synchronized operation through a unified input normalization layer and coordination driven by message queuing telemetry transport (MQTT). With $\pm$5ms temporal alignment and 32.80 ± 3.74 end-to-end latency, multiple operators can simultaneously transition between different robots and control interfaces without reconfiguration, as the automated data processing and annotation pipeline converts raw sensor and video streams into structured datasets suitable for vision-language-action (VLA) models. To validate the platform, the dataset generated through our platform trained a long-horizon bimanual laboratory task. We hope that our results could facilitate teleoperation-based training for biological and chemical laboratory techniques, and the development of generalized intelligence across diverse robotic embodiments.**

I. INTRODUCTION

Training robust embodied agents that can perceive, reason, and act in the real world requires access to long-horizon, multimodal data that capture sensory experience and interaction dynamics. Existing egocentric datasets provide passive observations that could inform robotic perception and actions, despite the ambiguous boundary of what constitutes actionable data [1]. The limited alignment between egocentric sensory streams and real-time robotic actions renders observational datasets alone suboptimal for training embodied agents that tightly couple perception and interaction.

In parallel, imitation learning on teleoperation datasets can train long-horizon policies: low-cost teleoperation [2] and mixed reality (MR) interfaces [3] support mobile and dexterous demonstrations. However, existing teleoperation systems are often tied to specific devices and prioritize fast data collection over end-to-end pipelines that produce temporally synchronized, semantically structured multimodal datasets at scale. Automated semantic annotation helps build such datasets for vision-language-action (VLA) learning.

*This work was supported in part by the the United Fund of the National Synchrotron Radiation Laboratory of USTC KY2100000173, and Young Innovative Fund WK2100000040 of USTC.

[1]Qiuqi Xu is with Washington University in St. Louis

[2]Haibao Liu is with University of Science and Technology of China

[3]Fei Zhang is with Faculty of Department of Automation, & Experiment Teaching Center of Information and Computer, University of Science and Technology of China. zfei@ustc.edu.cn

Addressing these challenges requires an embodied platform that satisfies several key design imperatives. It should produce high-fidelity, temporally aligned, multimodal data with automated post-processing to generate semantically structured datasets. It must support a variety of teleoperating devices, teleoperated devices, and human operators, with intuitive interfaces and robust system-level monitoring to minimize training overhead and operational errors.

Guided by these principles, we present a scalable and extensible platform for synchronized teleoperation and embodied data processing. Unlike existing teleoperation systems that are often tightly coupled to specific robots or operator interfaces, our system introduces a unified intermediate representation and modular communication architecture that decouple teleoperating devices from robot-specific controllers. This design enables rapid integration of new input and output devices without redesigning the overall control pipeline, facilitating scalable deployment across heterogeneous robotic embodiments. Beyond real-time teleoperation, to efficiently construct large-scale embodied datasets, the platform supports temporally synchronized logging of multimodal streams: video, sensor states, control commands, and system events. Furthermore, the modular post-processing pipeline provides customizable filtering, temporal segmentation, and semantic annotation workflows for downstream vision-language-action model training and embodied AI benchmarking. Our contributions are as follows:

1. We propose a teleoperation platform that decouples operator interfaces from robot-specific controllers through a unified intermediate representation, for scalable integration across multiple teleoperating and teleoperated devices.

2. We introduce a multimodal temporal synchronization that records video streams, robot states, control commands, and sensor data for embodied data collection and analysis.

3. We develop a post-processing pipeline with configurable filtering, segmentation, and annotation, to organize teleoperation recordings and facilitate embodied data management.

II. RELATED WORKS

To support heterogeneous teleoperating and teleoperated devices [3] [4], modular designs decouple human interfaces from robot-specific controllers: open-teach [5] uses Meta Quest 3 headset to teleoperate three models of robotic arms, one model of dexeterous hands and an arm-hand combination, whereas assisted and shared-autonomy approaches [6] use MR to teleoperate one model of robotic arm and simulated robot fleets, increasing demonstration throughput in complex manipulation tasks by combining human input

Fig. 1. High-level architecture: parallel pipelines for real-time teleoperation and data annotation. The web-based teleoperation pipeline in the left panel interacts with the data annotation pipeline in the right panel through a unified backend

with partial automation. Building on this line of work, our platform supports 4 teleoperating and 6 teleoperated devices with the flexibility to incorporate new devices, leveraging teleoperation to collect demonstration data for learning-based robotics.

While teleoperation alone can generate large volumes of demonstration data, insufficient coverage and temporal segmentation leave imitation learning policies vulnerable to compounding errors [7]. To improve policy robustness, expert interventions strengthen demonstrations with recovery behaviors and off-nominal trajectories [8] [9], while reinforcement learning refines robotic manipulation policies [10]. More recently, VLA models, fine-tuned with residual policies [11], train policies for long-horizon, language-conditioned manipulations. Nevertheless, many models need large amounts of temporally coherent and semantically grounded data [12]. Our platform approaches this need with temporal alignment across visual, proprioceptive, and control signals, buffered by message queuing telemetry transport (MQTT).

Egocentric perception addresses these limitations by capturing intent, object affordance, and task structure, which support embodied robot learning [13] [1] [14]. Teleoperation systems with MR devices [15] or humanoids [16] have started to include head-mounted cameras to record demonstrations. Indeed, distributed platforms [17] leverage augmented reality and cloud-based interfaces to scale the collection of data from egocentric interaction. While these approaches facilitate high-volume recordings and additional automated annotation workflows, extending these principles beyond controlled settings introduces additional challenges of heterogeneous lighting, occlusions, long-horizon activities, and concurrent actions. These factors motivate us to pair our teleoperation pipeline with automated filtering, temporal action segmentation, and scene-level semantic labeling to render large-scale egocentric data suitable for downstream VLA learning.

III. OUR SYSTEM

Code and demonstration videos are available at https://github.com/autumnxu/embodied

A. Overview of Pipelines

Our system consists of two pipelines: low-latency teleoperation and high-throughput data annotation (Figure 1). The real-time control pipeline interfaces with 4 teleoperating devices to teleoperate 2 humanoid robots, 2 robotic arms, and dexterous hands, temporally aligning video streams, sensor signals, and control commands. In parallel, the data annotation pipeline processes first-person egocentric video streams to detect hands, segment action, and semantically label scenes with agentic multimodal models, supporting downstream VLA models.

Both pipelines are coordinated through a Java Springboot backend that abstracts away hardware dependencies and facilitates multi-operator and multi-robot deployment. Operators use a web interface (implemented in TypeScript, served via NGINX) that authenticates devices, manages sessions, initiates and stop teleoperation sessions, monitors videos and data processing status. If a configuration of teleoperation fails, the system can either halt all teleoperation tasks or only pause the affected device, maintaining continuous operation

TABLE I

DEVICES USED IN THE TELEOPERATION PIPELINE.

Device	Role	DoF	Modality	Protocol
NOKOV Mars Series Motion Capture Cameras	Teleoperating	NA	skeletal joints, IMU (6-axis)	VRPN
Manus Metagloves Pro Hand Motion Capture Gloves	Teleoperating	25	finger joints, gestures, haptic feedback	socket
Meta Quest 2 Headset	Teleoperating	6	controller tracking	UDP
Pico 4 Ultra MR Headset	Teleoperating	6	skeletal joints, controller tracking	protobuf
Dobot Nova 5 Robotic Arm	Teleoperated	6	joints	ROS, TCP
Universal Robots UR3e Robotic Arm	Teleoperated	6	joints	ROS
ARX LIFT II mobile manipulator	Teleoperated	6	joints	ROS
Inspire RH56DFX Dexterous Hand	Teleoperated	6	finger joints, end-effector pose	ROS, Modbus
Unitree G1 Humanoid	Teleoperated	23	joint angles, end-effector pose	ROS, UDP
DH-PGI-140-80 Industrial Electric Parallel Gripper	Teleoperated	1	joint angles, end-effector pose	ROS, UDP
Intel Realsense Depth Camera d435i	Monitoring	NA	RGB stream, depth map, IMU (6-axis)	streaming media

for the others. The backend manages user authentication and system state, while a dedicated module in python automates data processing with perception and annotation models.

In addition to the aforementioned web-based interaction, we designed a MR controller interface optimized to collect similar movements of teleoperation in rapid succession. Pressing A ends an active session, B resets the robot for safe repositioning, and A returns it to the neutral configuration. Once reset, B starts the next recording. During data collection, X restarts the current episode, while Y stops collection, discards the current episode, and saves all previous recordings.

B. Cross-device Input Normalization

Through a unified interface, our teleoperation platform supports 4 input teleoperating devices: Nokov motion capture system for skeletal joints, Manus haptic gloves for hand pose and classifying gestures, and MR devices (Meta Quest, Pico) for body and hand orientations. These inputs control robotic arms (dobot, ur5, ARX), or gripper (DH-PGI-140-80), dexterous hands (Inspire), and/or humanoid robots (Unitree G1) (Table 1), each with different kinematics, joint limits, and end-effector modalities. This mix of teleoperating and teleoperated devices yields discordant signal types, data rates, and coordinate systems, to which the system introduces a shared intermediate representation consisting of hand poses, body skeletons, and confidence status. Early normalization reduces conversion overhead so that downstream action mapping and visualization operate independently of specific hardware, simplifying support for new devices.

To account for height and individual preferences, each new operator undergoes a calibration procedure. During this process, the operator holds their hands straight and parallel to the ground for 30 seconds, while the system enumerates orthogonal rotations about the X, Y, and Z axes to select the transformation that best aligns the neutral position of the operator with that of the teleoperated devices.

Human motion inputs are subsequently transformed into a shared action representation that decouples operator interfaces from robot-specific control commands, so one teleoperation interface can control multiple platforms (Figure 1). For safety considerations, the platform limits the maximum initial velocities of all devices. Upon stable control, the system gradually relaxes the maximum speeds to prevent sudden movements or collisions. By abstracting robot-specific control logic, the platform facilitates comparative studies across robots and simplifies integration of future devices.

C. Teleoperation Runtime Logic

Operator-specific calibration is loaded prior to teleoperation. Once teleoperation starts, the pipeline synchronously acquires normalized inputs from heterogeneous devices and filters invalid data: NaN or out-of-range values, frames with abnormally large inter-frame intervals, and signals with low confidence (Figure 2).

To control robotic arms, human joint poses are transformed from the world frame of the teleoperating device into the local reference frame of the teleoperated manipulator, anchored at the estimated shoulder and wrist positions. The transformed poses are then scaled according to calibration-defined human-to-manipulator ratios to accommodate morphological differences. Scaled poses are mapped to end-effector positions and orientations. Inverse kinematics generate joint commands, validated against angle and velocity constraints before execution. Meanwhile, dexterous hands are controlled via gloves or MR devices, retargeted according to ergonomic and anthropometric priors, and mapped to valid joint angles respecting hardware limits.

A clustered MQTT broker delivers teleoperation commands, sensor, and system status messages between teleoperating and teleoperated stations. Monitoring the broker's message throughput and individual machine's CPU utilization revealed delays, prompting a queue-based refactoring of the messaging interface to remove blocking operations. Control commands for arms and hands are sent in ROS messages at 100 Hz and 30 Hz, respectively, as the hardware feedback monitors actual joint states and motor status.

The teleoperation pipeline reports warnings and errors to a dedicated MQTT topic, which the web frontend monitors to issue corresponding alerts to users. The system maintains the most recent valid command during transient data loss, immediately halts execution upon hardware faults or joint limit violations, and triggers a software emergency stop to disable all actuators. To mitigate potential hardware or network failures, our platform caches video and sensor streams before committing them to network-attached storage:

sensor data are stored in MongoDB, video in local storage and Redis.

At the end of a teleoperation session, the system evaluates the size of collected data and video recordings to decide whether to immediately upload the data to the backend and network-attached storage or schedule a deferred upload, avoiding saturation of CPU, memory, and network resources that subsequent teleoperation sessions need.

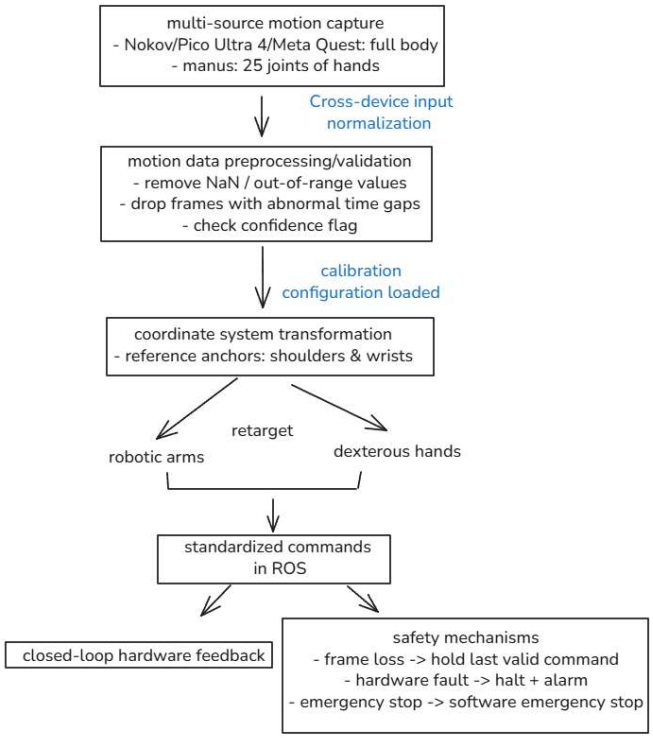


Fig. 2. Flowchart of teleoperation in its runtime.

D. Temporal alignment

Our system keeps temporal alignment among visual, proprioceptive, and control signals. When a teleoperation session starts, the backend computes a global start timestamp by adding a 5 s offset. The MQTT clustered broker then broadcasts this start timestamp to all teleoperating and teleoperated devices, so that data acquisition, control, and logging begin simultaneously. All incoming signals (operator inputs, sensor streams, robot states, and video frames) are timestamped upon arrival accordingly. Camera streams at 30 fps as the temporal anchor, with non-visual data buffered and aligned to the nearest frame. This frame-centric alignment pairs each visual observation with the corresponding robot state and operator command for downstream models.

Latency benchmarking revealed that the naive MQTT broker caused uneven message arrival intervals, ranging from 10 to 834 ms, resulting in laggy and bursty movements on the teleoperated devices. To address this, we tuned a clustered MQTT broker with bounded queues of 64 messages and lightweight buffering with a 500 ms processing interval, which stabilized message delivery and eliminated late or bursty arrivals. We release open-source C++ and python implementations that define how teleoperating and teleoperated devices interface with the MQTT cluster. Robot actuation commands are issued from the most recent synchronized state, while intermediate signals are logged asynchronously to avoid interference. After a 5 s warmup, the end-to-end input-to-actuation delay averages 32.80 ± 3.74 ms, successive actuating command at an interval of 30.54 ± 4.21 at 30 fps, and visual and proprioceptive data aligned within ± 5ms relative to camera frames. CPU workloads are distributed across processes to preserve high-fidelity synchronized recordings under multi-device operation.

E. Post collection data processing and semantic annotation

Raw videos retrieved from network-attached storage are filtered to remove empty or unstable frames using motion heuristics, keeping segments containing at least one hand. The remaining clips, $61.2\% \pm 8.3\%$ of the original length, are concatenated chronologically with the original encoding to avoid quality loss. Videos without hands are noted as empty to preserve dataset consistency. Compared to Willow and MediaPipe, we found YOLOv8x.pose [18] superior in speed, accuracy, and robustness when detecting hands wearing gloves, under occlusions or unusual poses. Those filtered videos are temporarily stored in a cache.

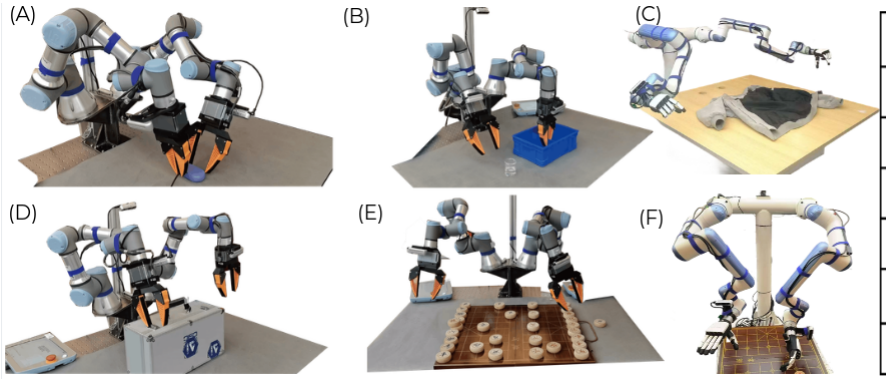
Those videos in cache are then processed with OpenTAD [19], to identify an action's start and end times at 0.01s resolution. For each video, a structured JSON file lists each detected action with its action label, confidence score, and temporal boundary. Given that long-horizon recordings may contain multiple overlapping operations, this step selects egocentric segments that are more informative for training and tuning robotic control policies. Nevertheless, videos with no detected actions are explicitly marked for dataset integrity.

Once action segments are identified, our platform clips each segment into an independent MP4 file with a globally unique UUID. The clip is then base64 encoded and submitted to multimodal agentic language model, qwen3_vl-plus [20] or Silicon Cloud, to generate structured annotations. To capture actions and environmental context, annotations consist of object inventories (names, quantities, and key attributes), scene classifications (such as factory floor and assembly station), and natural action descriptions capturing agents, objects, and interaction dynamics.

Inspired by Ego4D [1]and LeRobot [21] datasets, all data are stored in a file system structured by scene type (such as lab, factory and service). Each video clip has its annotation file. Synchronized sensor streams (IMU, depth, frame metadata action segments) are stored in parallel directories. A global metadata json indexes all processed samples, which links raw recordings to processed clips and records frame rate, duration, device ID, and recording date. A dedicated java module supports conversion to LeRobot format.

IV. Experiments

We evaluate our platform along three axes: its performance as an embodied data collection and annotation system, its



task	average completion time
a	103.42 ± 37.71s
b	41.78 ± 12.16s
c	182.35 ± 27.34s
d	98.47 ± 48.89s
e	29.36 ± 4.39s
f	219.18 ± 31.82s

Fig. 3. Demonstrations of teleoperating UR3e with Pico 4 Ultra (A, B, D, E) and of teleoperating Dobot and Inspire hands with motion capture system (C, F). The teleoperation tasks comprise (A) inserting charging cords into wireless earbuds, (B) pouring liquid into a laboratory flask, (C) folding clothes, (D) opening locked toolboxes, (E) playing Chinese chess by grasping and placing pieces, and (F) playing chess.

ability to support teleoperation across varied teleoperating and teleoperated devices, and the utility of resulting datasets for training VLA models. All experiments were conducted using the same system described in the previous section.

To assess system-level performance, we ran our system under concurrent video streaming, sensor logging, and action segmentation. Teleoperation remained stable across 4 devices and 6 video streams with no command drops for up to 3 hours or until the MR device went out of charge, while the data annotation pipeline processed 12 days of continuous recording, totaling 1697.54 hours of raw footage. The hand detection stage filtered this to 1005.01 hours, which action segmentation then cut into 258.25 hours of annotated segments for which agentic models labeled scenes.

Following the robustness validation, we assessed our platform on collecting long-horizon manipulation data, using Nokov motion capture system with Manus haptic gloves to teleoperate Dobot robotic arms and Inspire dexterous hands. Operators picked up and placed small lab objects, reoriented items with both hands, folded clothes, and put wrenches into toolboxes, totaling 520 GB of teleoperation data. While expressive control was achievable, tasks requiring sustained precision led to noticeable operator fatigue, reducing task throughput to 3.76 ± 0.67 minute/task. Teleoperating dexterous hands, though high-fidelity, cannot amount to the amount of data required to train VLA models.

Motivated by these findings, we tested a lighter teleoperation configuration using PICO Ultra 4 MR controllers to teleoperate dual UR3e robotic arms with two-fingered gripper. The operator found this configuration easier to control for single-and bimanual tasks. Using this configuration, we collected synchronized teleoperation and egocentric observation data in manipulation tasks (Figure 3), outperforming the motion capture system in speed.

To evaluate the utility of the collected dataset, we trained two VLA models, π_0 and $\pi_{0.5}$ [22]. The target bimanual laboratory task requires one robotic arm to position a blue sink between the arms while the other robotic arm pours liquid from a transparent beaker into the sink (Figure 4). This task requires precise coordination and timing, typical of biological and chemical laboratory workflows.

The policy trained on the $\pi_{0.5}$ model succeeded only once in 64 attempts, while the policy trained on the π_0 model achieved a success rate of 10% over 30 total trials, with an average completion time of 31.80 ± 0.46 s. Experiments without our platform data show similar performance trends, but our platform improves data efficiency.

V. Discussion

We present a novel platform for teleoperation-based data collection and automatic annotation for embodied robotics. The platform supports 4 teleoperating and 6 teleoperated devices, maintains precise temporal alignment across multimodal streams, and converts long-horizon egocentric recordings into structured, semantically labeled datasets. Teleoperation experiments using a motion capture system and MR controllers demonstrate reliable real-time control and dataset generation that can train VLA models.

Nonetheless, the lack of real-time monitoring around operator fatigue can compromise demonstration quality and constrain the duration of high-precision teleoperation sessions (ie. dexterous, coordinated hand tasks using Inspire hands).

Our platform also faces limitations for pretraining general-purpose VLA models, which benefit from consistent scenes, object placements, and actions. Although our system collects and annotates diverse data at scale, it does not enforce strict sample consistency or capture fine-grained spatial relationships, likely contributing to the low task success rates observed during policy evaluation. The slower execution time of trained policies compared to humans suggests inadequate coverage of recovery behaviors. Further, teleoperation may encourage interface-specific control strategies rather than transferable laboratory manipulation skills. The effort required to master the system can slow skill acquisition.

Future work will focus on data quality and system robustness. Real-time monitoring of operator fatigue could help with more consistent long-duration demonstrations. Automated scene normalization, spatial calibration and a variety of recovery behaviors may produce resulting datasets that enhance VLA model performance.

Fig. 4. Three perspectives of pouring the water in the beaker into the tank. Sequence of robot actions from left to right: initial pose, grasp and move the sink with the left claw, grasp the beaker to pour water into the sink with the right claw, and return to the neural position

REFERENCES

[1] K. Grauman *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 47, 2025.

[2] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Proceedings of the 8th Conference on Robot Learning (CoRL)*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, Nov. 2024, pp. 4066–4083.

[3] Z. Zhao, L. Yu, K. Jing, and N. Yang, "Xrobotoolkit: A cross-platform framework for robot teleoperation," in *2026 IEEE/SICE International Symposium on System Integration (SII)*, 2026. [Online]. Available: https://arxiv.org/abs/2508.00097

[4] D. Ghosh *et al.*, "Octo: An Open-Source Generalist Robot Policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[5] A. Iyer *et al.*, "Open teach: A versatile teleoperation system for robotic manipulation," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 2372–2395.

[6] S. Dass, K. Pertsch, H. Zhang, Y. Lee, J. J. Lim, and S. Nikolaidis, "PATO: Policy Assisted TeleOperation for Scalable Robot Data Collection," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[7] S. Ross, G. J. Gordon, and D. Bagnell, "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. Proceedings of Machine Learning Research, vol. 15. Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 627–635.

[8] E. Jang *et al.*, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 991–1002.

[9] 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)*, 2019, pp. 8077–8083.

[10] A. Mandlekar *et al.*, "IRIS: Implicit Reinforcement without Interaction at Scale for Learning Control from Offline Robot Manipulation Data," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[11] Y. Chen, S. Tian, S. Liu, Y. Zhou, H. Li, and D. Zhao, "ConRFT: A Reinforced Fine-tuning Method for VLA Models via Consistency Policy," in *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[12] Y. J. Ma *et al.*, "Vision language models are in-context value learners," in *International Conference on Learning Representations*, Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, Eds., vol. 2025, 2025, pp. 33 984–34 009.

[13] X. Wang *et al.*, "Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 20 270–20 281.

[14] D. Damen *et al.*, "The EPIC-KITCHENS dataset: Collection, challenges and baselines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4125–4141, Nov. 2021.

[15] A. H. Dugan, R. Sadykov, D. Roozbahani, M. Alizadeh, and H. Handroos, "Immersive telepresence via a humanoid robotic head using a VR-headset with real-time stereoscopic vision and binaural audio," *J. Intell. Robot. Syst.*, vol. 111, no. 2, May 2025.

[16] Y. Ze *et al.*, "Twist2: Scalable, portable, and holistic humanoid data collection system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

[17] Y. Park, J. S. Bhatia, L. Ankile, and P. Agrawal, "Dart: Dexterous augmented reality teleoperation platform for large-scale robot data collection in simulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 13 883–13 889.

[18] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "YOLO-Pose: Enhancing yolo for multi-person pose estimation using object keypoint similarity loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Jun. 2022, pp. 2636–2645.

[19] S. Liu *et al.*, "OpenTAD: A unified framework and comprehensive study of temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, Jun. 2025, pp. 2650–2660.

[20] S. Bai *et al.*, "Qwen3-VL technical report," Nov. 2025. [Online]. Available: https://arxiv.org/abs/2511.21631

[21] R. Cadene *et al.*, "LeRobot: An Open-Source Library for End-to-End Robot Learning," in *Proc. International Conference on Learning Representations (ICLR)*, 2026. [Online]. Available: https://arxiv.org/abs/2602.22818

[22] K. Black *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 17–40.