Fast-UMI: A Scalable and Hardware-Independent Universal Manipulation Interface (Short Version)

Ziniu Wu[1†], Tianyu Wang[1†], Zhaxizhuoma[1†], Chuyue Guan[1*], Zhongjie Jia[1*],
Dong Wang[1], Nieqing Cao[2], Yan Ding[1‡], Bin Zhao[1,3], Xuelong Li[1,4]

[1]Shanghai AI Lab, [2]Xi'an Jiaotong-Liverpool University,
[3]Northwestern Polytechnical University, [4]Institute of AI, China Telecom Corp Ltd

† * Equal Contribution, ‡ Project Leader
Project Website: https://fastumi.com/

Abstract—Collecting real-world manipulation trajectory data involving robotic arms is essential for developing general-purpose action policies in robotic manipulation, yet such data remains scarce. Existing methods face limitations such as high costs, labor intensity, hardware dependencies, and complex setup requirements involving SLAM algorithms. In this work, we introduce *Fast-UMI*, an interface-mediated manipulation system comprising two key components: a *handheld device* operated by humans for data collection and a *robot-mounted device* used during policy inference. Our approach employs a decoupled design compatible with a wide range of grippers while maintaining consistent observation perspectives, allowing models trained on handheld-collected data to be directly applied to real robots. By directly obtaining the end-effector pose using existing commercial hardware products, we eliminate the need for complex SLAM deployment and calibration, streamlining data processing. Fast-UMI provides supporting software tools for efficient robot learning data collection and conversion, facilitating rapid, plug-and-play functionality. This system offers an efficient and user-friendly tool for robotic learning data acquisition.

I. INTRODUCTION

Collecting data of robotic arms interacting with objects in real-world environments is essential for advancing general-purpose action policies in robotic manipulation [1, 6, 15]. However, the scarcity of such interaction data has significantly hindered progress in this field. Existing data collection systems can be categorized into three types: direct human teleoperation [16], immersive technology-based teleoperation [7, 12], vision-based data collection [2, 9], and interface-mediated manipulation [3, 11, 13].

Direct human teleoperation involves operators controlling robots remotely or on-site to acquire comprehensive data, including visual inputs, motor states, and action commands. Although this method provides high-quality data, it is costly and labor-intensive. Even with devices like the SpaceMouse[1], a six-degree-of-freedom controller, collecting data for fine-grained operations remains challenging due to difficulties in precisely aligning with small target objects. *Vision-based data collection* uses cameras, such as wearable devices, to capture interaction data without direct robot control. While this approach gathers certain visual information, it lacks the ability to represent the complex interactions between robotic arms and their environments [8]. *Interface-mediated manipulation* systems, exemplified by Universal Manipulation Interface (UMI) [6], employ handheld grippers and specialized interfaces to collect data from human demonstrations, specifically capturing the end-effector poses of robotic arms. Algorithms like Diffusion Policy [5] then infer robotic actions from the collected data, reducing costs and simplifying the data collection process.

The UMI system addresses challenges in human demonstration data collection and supports action policy learning across various scenarios, but it still has two limitations: *strong coupling with specific robotic hardware and complexities arising from the use of open-source SLAM[2] in the system.* First, the system's strict hardware requirements—such as the necessity of using the Weiss WSG-50 gripper[3]—impose limitations. Users must procure these specific components to directly implement UMI, increasing costs and limiting adoption among those with different robotic configurations. Adapting UMI to other hardware requires redesigning grippers, recalibrating cameras, performing SLAM calibration, and modifying code parameters, which are labor-intensive tasks hindering plug-and-play functionality. Furthermore, these modifications often lack generalizability, complicating application across different laboratories and equipment. Second, while leveraging SLAM technology enables the estimation of the end-effector's pose, using open-source solutions like ORB-SLAM3 introduces additional challenges. SLAM performance highly depends on parameter settings of the handheld device, and deployment and debugging are complex and time-consuming. Users must invest considerable effort in data visualization and alignment during configuration. The system also requires global coordinate calibration involving multiple conversion steps, reducing user-friendliness. Additionally, the collected data's usability
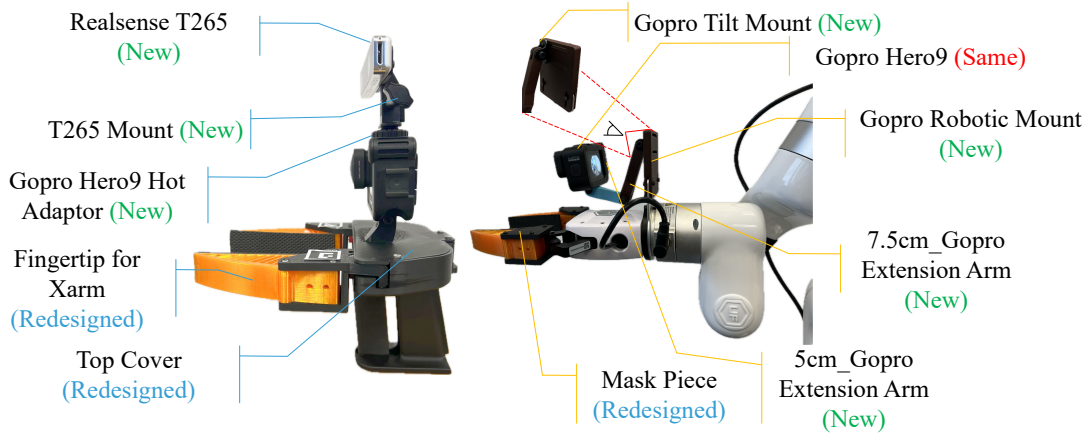
---

[2]ORB-SLAM3 [4] is used here.

Fig. 1. Physical prototypes of our *Fast-UMI* system. **Left**: *The handheld device* integrates a GoPro camera for visual monitoring, a RealSense T265 for capturing the end-effector's six-degree-of-freedom pose, and a yellow gripper equipped with fingertip markers to measure gripper aperture. **Right**: *The robot-mounted device* replicates the handheld configuration to ensure consistent observation perspectives between human demonstrations and robotic executions. We employ a color-coding scheme to differentiate the hardware architectures of our proposed Fast-UMI and the original UMI system. Green indicates new components not present in UMI; Blue represents components redesigned based on UMI's counterparts; Red denotes components shared between Fast-UMI and UMI. Fig. 2 shows the various components of the Fast-UMI device.

for training depends on the SLAM algorithm's performance; failures to obtain accurate end-effector coordinates may necessitate discarding data, thereby reducing collection efficiency.

To enable laboratory and industrial users to easily employ efficient devices for data collection easily, we have undertaken a redesign with several *objectives*:

- *Decoupling from robotic hardware to enhance adaptability*: Removing strict hardware dependencies allows the new design to integrate with a wide range of robotic arms and grippers, facilitating broader adoption across different platforms.
- *Facilitating rapid user deployment through plug-and-play functionality*: The reengineered system is developed for quick installation and minimal configuration, enabling users to deploy the interface swiftly without extensive setup procedures.
- *Providing supporting software tools for efficient data collection and conversion*: We offer software solutions that streamline data acquisition and processing, ensuring seamless integration with existing imitation learning algorithms, such as ACT [14] and and Diffusion Policy.
- *Laying the groundwork for enhanced scalability to support multimodal datasets*: The redesigned interface is prepared to accommodate various data types and sensors, such as tactile sensors, allowing for the potential collection of multimodal datasets to support more complex robotic learning tasks in future iterations.

To achieve these objectives, we adopt a decoupled design philosophy. We attach finger extensions identical to those on the handheld device to the robot's gripper, aligning the robotic system with the UMI handheld apparatus. By equipping existing robot grippers with these attachments, we ensure consistent observation perspectives, allowing models trained on handheld-collected data to be directly applied to real robots. While retaining the GoPro camera as in the

UMI system, our mechanical design ensures precise alignment the camera's viewpoint with the fingertips across different hardware configurations. We also refine the handheld device's mechanical structure to improve operational stability. Unlike UMI, which relies on a SLAM algorithm, we directly use the RealSense T265 camera[4] to obtain the robot's end-effector pose, eliminating the need for complex SLAM deployment and calibration, thereby simplifying data processing. Our method requires no repetitive extrinsic calibration, simplifying both software and hardware integration. To ensure that Fast-UMI almost meets UMI system performance while reducing costs and simplifying deployment, we rigorously test its observation consistency and data collection process. Consequently, we develop an integrated solution that combines the handheld device with robot-mounted equipment, providing an efficient and user-friendly tool for robotic learning data collection.

II. PROTOTYPE DESIGN

In this section, we detail the design of our Fast-UMI system, focusing on two key components: the *handheld device* operated by humans for data collection and the *robot-mounted device* used during policy inference. Our design aims to ensure visual alignment between these devices while decoupling from specific robotic hardware to enhance adaptability.

Design Challenges. Building upon the objectives outlined in the Introduction, our prototype design addresses several critical challenges. A significant challenge is *decoupling the system from specific robotic hardware* to enhance adaptability. Designing components that integrate seamlessly with a wide variety of robotic arms and grippers—each differing in size, shape, and mechanical interface—requires innovative mechanical solutions. Achieving *visual consistency* between the handheld and the robot-mounted devices present another

---

[4]https://dev.intelrealsense.com/docs/depth-and-tracking-cameras-alignment

(a) Gopro Hero9 Hot Adaptor **(b) T265 Mount** **(c) Fingertip for XArm** **(d) Gopro Extension Arm**

(e) Gopro Tilt Mount **(f) Top Cover** **(g) Mask Piece** **(h) Gopro Robotic Mount**
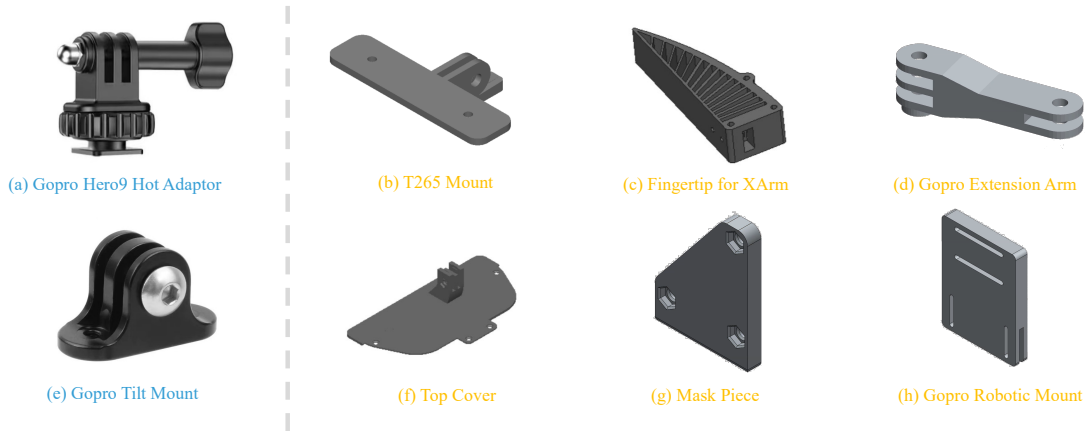
Fig. 2. Various components of the Fast-UMI prototype device. We employ a color-coding scheme to categorize hardware components based on procurement method: blue represents components to be purchased, while yellow denotes components requiring 3D printing.

challenge. Variations in gripper dimensions necessitate adjustable mechanical designs to maintain consistent camera perspectives, crucial for effective policy transfer in robotic learning algorithms. *Fast deployment* to facilitate rapid user setup is also a key concern. Creating a plug-and-play solution demands careful system architecture consideration, minimizing the need for extensive calibration, mechanical adjustments, or software configuration. Ensuring that users could install and configure the system with minimal effort is essential for broad adoption. Finally, *preparing for future scalability* to support multimodal datasets introduced challenges in modularity and flexibility. We need to design the system to accommodate additional sensors and data types in future iterations without significant redesign, requiring a forward-thinking approach to both hardware and software components.

Decoupled Design Philosophy. To address these challenges, we adopt a *decoupled design philosophy*. We attach identical fingertip extensions from the handheld device to the robot's gripper (see Fig. 1). This design maintains consistency between the robotic system and the handheld apparatus, allowing models trained on data collected via the handheld device to be directly applied to real robots. We develop insertable fingertip extensions compatible with five mainstream gripper models, including XArm gripper [5] and robotiq 2f-85[6]. This methodology can be adapted for other gripper types as well.

Handheld Device Design. The handheld device (see the left subfigure in Fig. 1) is used for manual data collection to train action policies. It consists of:

- *GoPro Camera with fisheye extension module*: Captures fisheye images for monitoring and data collection.
- *RealSense T265 Camera*: Obtains the six-degree-of-freedom pose of the end effector.
- *Handheld Gripper*: Equipped with two markers at its fingertips to record the gripper's width.

We pay special attention to aligning the camera's viewpoint

[5]https://uk.robotshop.com/products/xarm-gripper
[6]https://robotiq.com/products/adaptive-grippers

with the gripper's fingertips to ensure visual consistency with the robot-mounted device.

Robot-Mounted Device Design. The robot-mounted device (see the right subfigure in Fig. 1) is engineered to accommodate various robotic arm configurations. It primarily includes:

- *GoPro-Robot Mount (Brown Extension Plate)*: Serves as the mounting point for the GoPro camera.
- *Adjustable Extension Arm (Blue Extension Arm)*: Allows for lateral and vertical adjustments to align the camera's viewpoint.

By adjusting the extension arm, we can achieve visual consistency with the handheld device across different platforms. The insertable fingertip extensions ensure that, despite variations in gripper sizes and shapes, the visual perspective remains consistent.

Visual Alignment and Consistency. To ensure visual consistency between the handheld and robot-mounted devices, we established a visual alignment guideline: *the bottom of the GoPro's fisheye lens image aligns with the bottom of the gripper's fingertips*. This guideline enhances visual consistency and ensures proper camera positioning on both devices. Even with identical fingertip extensions, variations in gripper sizes can affect visual alignment. Our adjustable mechanical design is able to compensates for these displacements, allowing the extension arm to be adjusted as needed to maintain consistent observation perspectives. Figure 3 shows the views captured by the GoPro cameras on the handheld device and the robot-mounted device, respectively.

Camera Selection and Mounting. The choice of the GoPro camera was deliberate. Its fisheye lens captures wide-angle images that can potentially replace the combination of first-person and third-person planar cameras traditionally used in algorithms like ACT and DP. Our preliminary observations suggest that fisheye images from a single camera can provide sufficient spatio-temporal information, simplifying the hardware setup by eliminating the need for multiple cameras. This simplification is particularly beneficial for mobile robotic arms

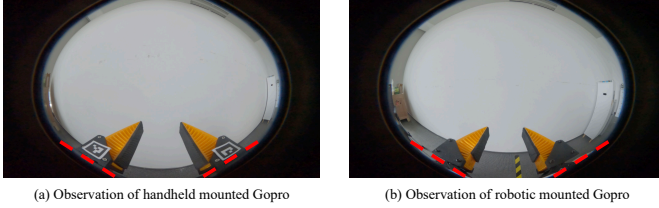| (a) Observation of handheld mounted Gopro | (b) Observation of robotic mounted Gopro |

Fig. 3. The views captured by the GoPro cameras on the handheld device and the robot-mounted device, respectively, with the red dashed line indicating the ends of the fingertips.

in real-world applications, where installing multiple cameras may be impractical as there may be occlusion when using kinesthetic teaching methods. We mounted the RealSense T265 camera using specially designed limiters to ensure it remains perpendicular to the GoPro camera. This design choice simplifies the installation process and guarantees precise alignment between the two cameras, facilitating accurate pose estimation without the need for complex SLAM algorithms.

Design Optimizations and Improvements. Unlike the original UMI system, we omits mirrors on the sides of the gripper. Experiments with UMI indicates that mirrors provide limited improvements on systems performance. Omitting them preserves valuable space on top of the gripper for integrating additional sensors, such as tactile sensors, thus enhancing the potential for future system expansion. To improve the stability and durability of the robot-mounted device, we have made several optimizations:

- *Reinforced the GoPro-Robot Mount*: Enhanced the structural integrity to reduce vibrations.
- *Used Carbon Fiber Materials*: Increased strength while reducing weight.
- *Standardized Male-Female Interface Design*: Allowed sequential connection of extension arms to adjust length without significant vibration (up to three extensions tested).

These enhancements ensure reliable performance during data collection and improve the user experience by simplifying hardware adjustments.

System Adjustability and Adaptability. Our configuration allows all users to share a standardized handheld device, while the robot-mounted device can be adjusted to fit various robotic arms and gripper models. This arrangement ensures consistency in data collection through the uniform handheld device, while the adjustable robot-mounted device enhances system versatility. The extension arm's length can be modified using the standardized interface, and its modular design facilitates easy adjustments. We believe that our design methodology can be applied to other types of grippers beyond the five mainstream models we have already adapted. This adaptability furthers our goal of decoupling the system from specific robotic hardware, making Fast-UMI accessible to a broader range of users.

III. DATA COLLECTION

This section details the procedure for data collection using our Fast-UMI prototype device. While comprehensive code and implementation specifics are available in our project website, we provide a concise description of the data collection workflow to facilitate rapid adoption.

Device Preparation. Data collection primarily involves capturing fisheye images from the GoPro and acquiring six-degree-of-freedom pose data from the RealSense T265. Unlike the original UMI system, which relies on complex SLAM-based pose estimation, we leverage the T265's built-in tracking capabilities to directly obtain end-effector pose data, simplifying the data processing pipeline. All data is transmitted via wired connections to ensure stability and real-time performance.

- *GoPro Camera*: A GoPro Hero 9 camera configured in ultra-wide mode captures fisheye images at a resolution of 1280×720 and 60 FPS, providing an extensive field of view for comprehensive scene coverage. Real-time image transmission is facilitated via an Elgato HD60 X capture card. For higher resolutions, more advanced capture cards may be employed; we plan to evaluate higher-resolution configurations in future work.
- *RealSense T265*: This device captures six-degree-of-freedom pose data of the handheld gripper, which we convert to the Tool Center Point (TCP) pose to represent the trajectory of human demonstrations. Compared to UMI, our design eliminates the need for a complex post-processing SLAM pipeline to reconstruct TCP trajectories, significantly simplifying data processing.

Data Synchronization and ROS Nodes. To coordinate data collection from multiple sensors, we utilize Robot Operating System (ROS) [10] as middleware. ROS provides a flexible framework for developing robotic applications, enabling communication between various nodes—independent processes executing specific tasks—and ensuring precise synchronization of data from multiple sources. n our data collection setup, we employ the following ROS nodes:

- *GoPro Node*: Captures fisheye images from the GoPro camera and publishes the image data stream for downstream processing. These images offer a wide field of view, crucial for capturing comprehensive environmental visual information.
- *T265 Node*: Interfaces with the RealSense T265 tracking camera to obtain the pose and orientation of the end-effector's pose and orientation. Accurate tracking of the end-effector is essential for imitation learning tasks, and this node publishes pose data in real time for monitoring and recording the movements.
- *Gripper Width Calculation Node*: Calculates the gripper aperture using fiducial markers on the handheld device.

Precise synchronization of these data streams is critical to ensure temporal alignment of sensor readings. Any inconsistencies could lead to errors in interpreting the robot's actions

(a) Spatial Trajectory in X, Y, and Z axis

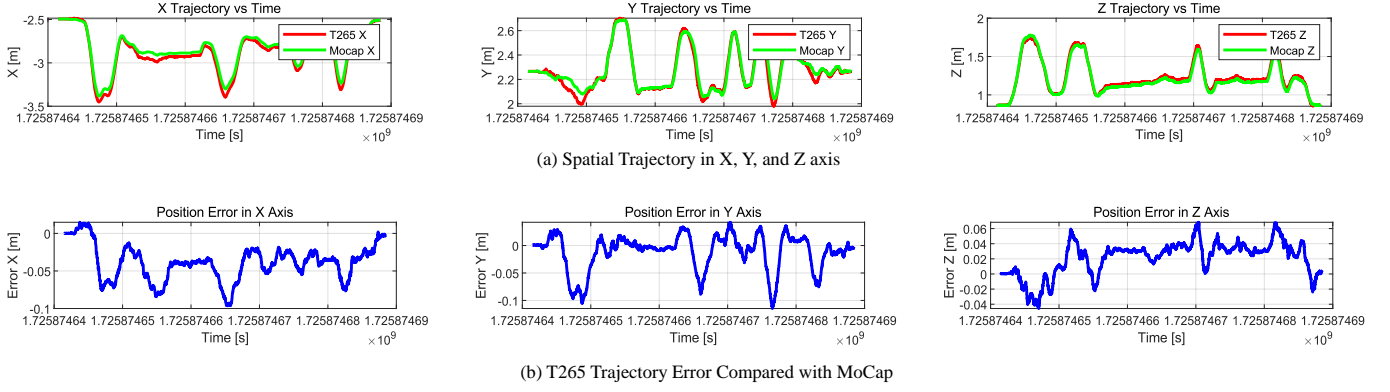(b) T265 Trajectory Error Compared with MoCap

Fig. 4. Evaluation of the *RealSense T265* trajectory accuracy compared to motion capture (MoCap) ground truth data. (a) Spatial trajectories along three axes: T265 measurements (red lines) and MoCap ground truth (green lines). (b) Positional errors of the T265 sensor relative to MoCap along the three axes.

Fig. 5. The task involves robotic manipulation to grasp a cup and place it into a sink. The first three images depict a human operator utilizing the Fast-UMI interface-mediated manipulation device to collect demonstration data. The subsequent two images show the robot executing an inferred action policy, trained on the collected data using the ACT algorithm.

during imitation learning, adversely affecting learning performance. To achieve temporal synchronization, we implement a dedicated *data collection node*. This node aggregates real-time data from the GoPro, T265, and gripper width calculations, recording them with unified timestamps. By storing these synchronized data points, we construct a comprehensive and accurate dataset representing the robot's actions and the surrounding environment, which is instrumental for training robotic learning models to replicate human demonstrations with high fidelity.

Data Collection Steps. The data collection procedure involves the following steps:

- *Step 1: Initialize Sensor Nodes*: Launch the GoPro node, T265 node, and gripper width calculation node to verify that data from all sensors are being published correctly.
- *Step 2: Execute Data Collection Using Handheld Device*: With all sensor nodes operational, a human operator performs the desired tasks using the handheld device. The data collection node records synchronized data from all sensors in real time as the operator executes the actions.
- *Step 3: Perform Data Conversion*: Upon completing data collection, run the data conversion node to transform the raw dataset into a format compatible with specific imitation learning models, such as ACT or Diffusion Policy.

This streamlined process simplifies data collection, enabling users to deploy our Fast-UMI system without complex configurations. Detailed code and implementation specifics are available in our project website.

IV. EVALUATION AND DEMONSTRATION

Quantitative Analysis of T265 Pose Estimation Accuracy. As shown in Fig. 4, subfigure (a) illustrates the spatial trajectory of the T265 sensor in red, while the MoCap system's trajectory, serving as the ground truth, is displayed in green. Subfigure (b) presents the positional errors of the T265 sensor compared to the MoCap data across the X, Y, and Z axes. For the *X-axis*, the mean positional error is -0.0384 *m* with a variance of 0.00056, indicating a slight negative bias in the T265 measurements. On the *Y-axis*, the mean error is -0.0116 *m*, with a higher variance of 0.00109, reflecting smaller positional error but greater variability compared to the X-axis. The *Z-axis* demonstrates a positive mean error of 0.0212 *m* and the smallest variance at 0.00051, suggesting a minor upward bias with relatively low variability. Overall, the T265 trajectory demonstrates an average positional error of **0.0237** *m*. These findings indicate that while the T265 sensor provides reasonably accurate pose estimation suitable for many robotic manipulation tasks, inherent biases and variances exist that should be accounted for in precision-critical applications.

Demonstration. We validate the Fast-UMI system in real-world environments by implementing an action policy inferred through the ACT algorithm trained on the collected dataset. Additional demonstrations illustrating the system's performance are available on our website.

V. Conclusion and Future Work

We have presented *Fast-UMI*, an interface-mediated manipulation system designed to simplify and enhance data collection for robotic manipulation tasks. By employing a decoupled design compatible with various grippers and maintaining consistent observation perspectives, Fast-UMI allows models trained on handheld demonstration data to be directly applied to various robots. This approach eliminates the need for complex SLAM deployment and calibration, streamlining the data processing pipeline. Fast-UMI provides user-friendly software tools for efficient data collection and conversion, facilitating rapid, plug-and-play functionality. By addressing hardware dependencies and setup complexities inherent in previous systems, Fast-UMI offers an accessible and effective solution for acquiring high-quality manipulation trajectory data, thereby advancing the development of general-purpose action policies in robotic manipulation.

While the current prototype constitutes version 1.0 of our Fast-UMI system, future work will focus on releasing enhanced iterations that offer improved performance and user experience. These advanced versions will integrate additional sensing modalities, such as tactile and force sensors, to facilitate multimodal data acquisition. By incorporating a broader array of sensors, we aim to augment the system's capabilities, enabling more sophisticated user-robot interaction modeling and supporting more complex robotic manipulation tasks. This progression will enhance the scalability and adaptability of Fast-UMI, further solidifying its utility as a comprehensive tool for robotic learning research.

References

[1] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.

[2] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild, 2022. URL https://arxiv.org/abs/2207.09450.

[3] Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, Oleg Sushkov, David Barker, Jonathan Scholz, Misha Denil, Nando de Freitas, and Ziyu Wang. Scaling data-driven robotics with reward sketching and batch reinforcement learning, 2020. URL https://arxiv.org/abs/1909.12200.

[4] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

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

[7] Wen Fan, Xiaoqing Guo, Enyang Feng, Jialin Lin, Yuanyi Wang, Jiaming Liang, Martin Garrad, Jonathan Rossiter, Zhengyou Zhang, Nathan Lepora, Lei Wei, and Dandan Zhang. Digital twin-driven mixed reality framework for immersive teleoperation with haptic rendering. *IEEE Robotics and Automation Letters*, 8(12): 8494–8501, 2023. doi: 10.1109/LRA.2023.3325784.

[8] 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. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[9] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, 2016. URL https://arxiv.org/abs/1603.02199.

[10] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[11] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play, 2023. URL https://arxiv.org/abs/2302.12422.

[12] Dandan Zhang, Ziniu Wu, Jin Zheng, Yifan Li, Zheng Dong, and Jialin Lin. Hubotverse: Toward internet of human and intelligent robotic things with a digital twin-based mixed reality framework. *IEEE Robotics Automation Magazine*, pages 2–12, 2024. doi: 10.1109/MRA.2024.3417090.

[13] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, 2018. URL https://arxiv.org/abs/1710.04615.

[14] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[15] Zhaxizhuoma, Pengan Chen, Ziniu Wu, Jiawei Sun, Dong Wang, Peng Zhou, Nieqing Cao, Yan Ding, Bin Zhao, and Xuelong Li. Alignbot: Aligning vlm-powered customized task planning with user reminders through fine-tuning for household robots, 2024. URL https://arxiv.org/abs/2409.11905.

[16] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Conference on Robot Learning*, pages 1199–1210. PMLR, 2023.