TriPilot-FF: Coordinated Whole-Body Teleoperation with Force Feedback

Zihao Li Yanan Zhou Ranpeng Qiu Hangyu Wu Guoqiang Ren Weiming Zhi

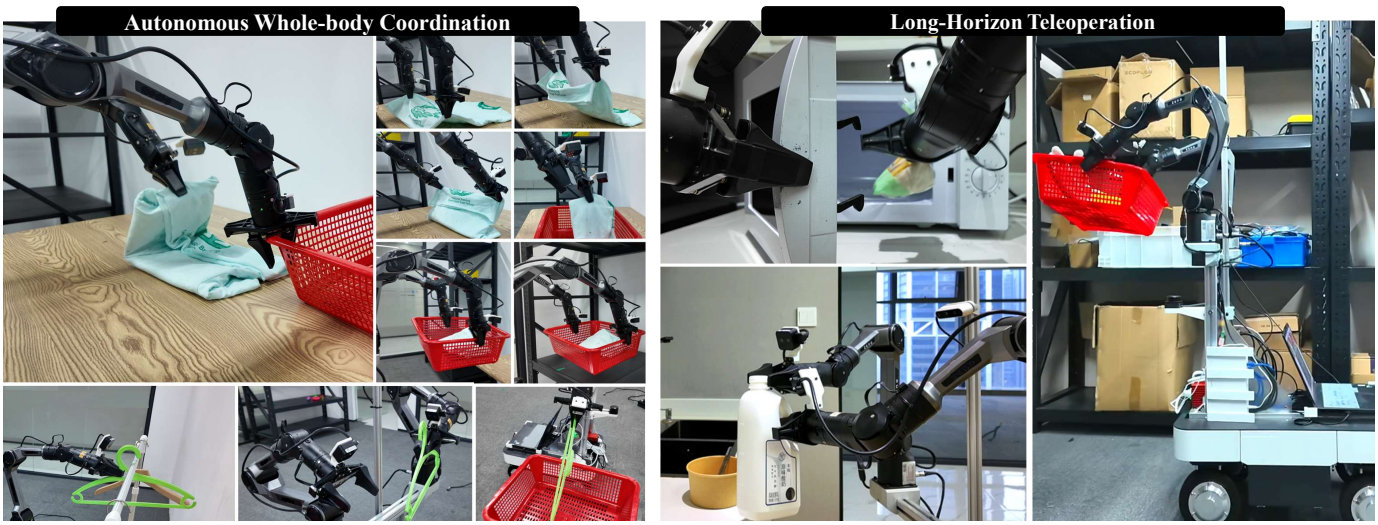


Fig. 1: TriPilot-FF is a whole-body teleoperation system that guides human operators via force feedback on both upper and lower limbs. We demonstrate the ability to operate over long-time horizons and in tasks requiring precise collision avoidance and bimanual coordination. Data from TriPilot-FF is used to train a policy that can robustly complete tasks which demand careful whole-body coordination.

***Abstract*—Mobile manipulators broaden the operational envelope for robot manipulation. However, the whole-body teleoperation of such robots remains a problem: operators must coordinate a wheeled base and two arms while reasoning about obstacles and contact. Existing interfaces are predominantly hand-centric (e.g., VR controllers and joysticks), leaving foot-operated channels underexplored for continuous base control. We present *TriPilot-FF*, an open-source whole-body teleoperation system for a custom bimanual mobile manipulator that introduces a foot-operated pedal with lidar-driven pedal haptics, coupled with upper-body bimanual leader-follower teleoperation. Using only a low-cost base-mounted lidar, TriPilot-FF renders a resistive pedal cue from proximity-to-obstacle signals in the commanded direction, shaping operator commands toward collision-averse behaviour without an explicit collision-avoidance controller. The system also supports arm-side force reflection for contact awareness and provides real-time force and visual guidance of bimanual manipulability to prompt mobile base repositioning, thereby improving reach. We demonstrate the capability of TriPilot-FF to effectively "co-pilot" the human operator over long time-horizons and tasks requiring precise mobile base movement and coordination. Finally, we incorporate teleoperation feedback signals into an Action Chunking with Transformers (ACT) policy and demonstrate improved performance when the additional information is available. We release the pedal device design, full software stack, and conduct extensive real-world evaluations on a bimanual wheeled platform. The project page of TriPilot-FF is http://bit.ly/46H3ZJT.**

The authors are with Zeno AI, Zhejiang University, Zhejiang University of Technology, and the University of Sydney.

I. Introduction

Mobile manipulators are increasingly deployed in settings where value is created through whole-body interactions. Compared with fixed-base arms, these platforms offer substantially larger operational envelopes and the ability to recover from failed interactions by moving the body rather than re-grasping. However, this capability comes with a core systems challenge: whole-body control couples navigation, collision avoidance, and manipulation. Operators must continuously coordinate a mobile base and two arms while reasoning about obstacles, contact forces, reachability, and task intent. In typical interfaces, most control bandwidth is allocated to the hands, such as joysticks, while the feet are often unused. This design choice is at odds with human motor capabilities: feet can provide a robust, low-latency control channel that naturally complements hand-centric manipulation, especially for base motion. As a result, operators often lack an informative modality for continuous base control that reduces collisions and coordination burden.

Additionally, autonomous robot policies are increasingly trained from large-scale real-world datasets [1]. Teleoperation is therefore not merely a fallback for autonomy, but is becoming a primary data acquisition and supervision mechanism for scalable robot learning. Interfaces that reduce operator workload, prevent common failures such as base collisions, and expose informative feedback channels can translate directly

into cleaner demonstrations and training signals for policy learning. As a result, we are motivated by a design philosophy that treats whole-body teleoperation as an interactive setup, not simply a mapping from user inputs to robot actions. The operator is continuously balancing base safety, contact stability at the end-effectors, and kinematic feasibility of the bimanual configuration. Providing this information through on-board cameras alone is often insufficient; it forces the operator to mentally simulate constraints that the system can encounter in real time. A complementary approach is to render task-relevant constraints over visual inputs and, through haptic guidance, shape user commands without removing control authority or requiring an explicit autonomy stack.

We present *TriPilot-FF*, an open-source whole-body teleoperation system for a custom bimanual mobile manipulator that operationalises this approach. Here, we reclaim the feet as a continuous base-control modality, augmented with lightweight, real-time guidance signals that reduce cognitive load while preserving intuitive operation. TriPilot-FF introduces a foot-operated pedal device that provides *lidar-driven force feedback*. Using only a low-cost base-mounted lidar, the system estimates proximity to obstacles in the commanded direction and renders an opposing resistive cue at the pedal. This physically shapes base commands toward collision-averse behaviour, allowing the operator to remain in control while receiving immediate, embodied feedback about imminent constraints. TriPilot-FF couples this with upper-body bimanual leader-follower teleoperation for intuitive arm control with arm-side force reflection to improve contact awareness during manipulation. To address reachability and coordination failures common in mobile bimanual tasks, the system additionally provides real-time force and visual guidance of bimanual manipulability, prompting operators with force cues when base repositioning is required to restore reach. These feedback signals act as a "co-pilot" for operators.

Finally, we connect interface design to robot learning by incorporating feedback signals into an Action Chunking with Transformers (ACT) policy [2]. Teleoperation produces more than state-action pairs: it also yields structured resistive feedback that reflects task constraints. We demonstrate that including torque into the state information of the policy improves downstream performance, supporting the thesis that better teleoperation interfaces can be leveraged for better learning. Figure 1 illustrates a few of the autonomous and teleoperated tasks demonstrated in this work.

Concretely, our technical contributions include:

- TriPilot-FF: a whole-body teleoperation system coupling foot-operated base control with bimanual leader-follower arm teleoperation for mobile manipulation.
- Lidar-driven pedal feedback: direction-conditioned resistive cues from obstacle proximity, biasing commands toward collision-averse behaviour.
- Multi-modal whole-body guidance: arm-side force reflection for contact awareness and real-time bimanual manipulability guidance on the pedal to mitigate reachability-driven failures.
- Autonomous robot executions by an augmented ACT formulation that incorporates collected force information.

II. Related Works

Mobile and Whole-body Manipulation: Mobile manipulation has traditionally been approached via motion planning and control pipelines [3], [4], [5], [6], coordinating an arm mounted on a wheeled base through hand-crafted hierarchical systems. More recently, robot learning, most notably MobileALOHA [7], has enabled hardware and data-collection setups for training whole-body policies. Earlier efforts also studied mobile manipulation data collection in simulation [8], including task-specific settings such as door opening [9]. Beyond wheeled bases, prior work explores other mobile platforms, including quadrupeds with manipulators [10], [11]. A related thread studies humanoid teleoperation [12], [13], [14], including interfaces that provide visual feedback to the operator [15]. Our system targets bimanual mobile manipulators and is most closely related to MobileALOHA [7] and recent mobile manipulation systems [16], [17], with conceptual links to novel humanoid teleoperation hardware [18]. Unlike MobileALOHA, our operator need not be co-located with the robot, and receives both visual and force feedback across upper- and lower-limb control channels.

Imitation Learning: Beyond the teleoperation interface, our work connects to imitation learning of visuomotor policies [19], which seeks to map the robot's sensory inputs directly to robot actions. Recent policy classes include diffusion models [20], flow-based models [21], [22], and transformer-based policies [2], [23], [24]. Complementary to model design, several low-cost systems have been proposed for demonstration collection in imitation learning [25], [26], [27]. TriPilot is similarly low-cost, and enables the collection of high-quality data: we use it to train ACT [2] policies, condition them on additional force/torque signals logged by TriPilot-FF, and show that the resulting dataset supports data-efficient training.

III. Preliminaries

Platform and task setting: We consider whole-body teleoperation of a bimanual mobile manipulator comprising an omnidirectional base and two n-DoF arms (left/right). The operator must simultaneously (i) drive the base to manage obstacle clearance and reachability and (ii) manipulate with both arms under contact, grasp stability, and kinematic limits. TriPilot-FF targets *interactive* teleoperation: the human retains primary control authority, while the system continuously renders task-relevant constraints through haptic and visual cues to reduce cognitive burden.

State, commands, and notation: Let the base command be a planar body twist,

$$u_b = [v_x,\, v_y,\, \omega]^\top \in \mathbb{R}^3 \tag{1}$$

defined in the base frame. For each arm, let joint position, velocity, and torque be respectively $(q,\, \dot{q},\, \tau)$. We denote the end-effector pose as $T_{ee} \in SE(3)$ and the geometric Jacobian

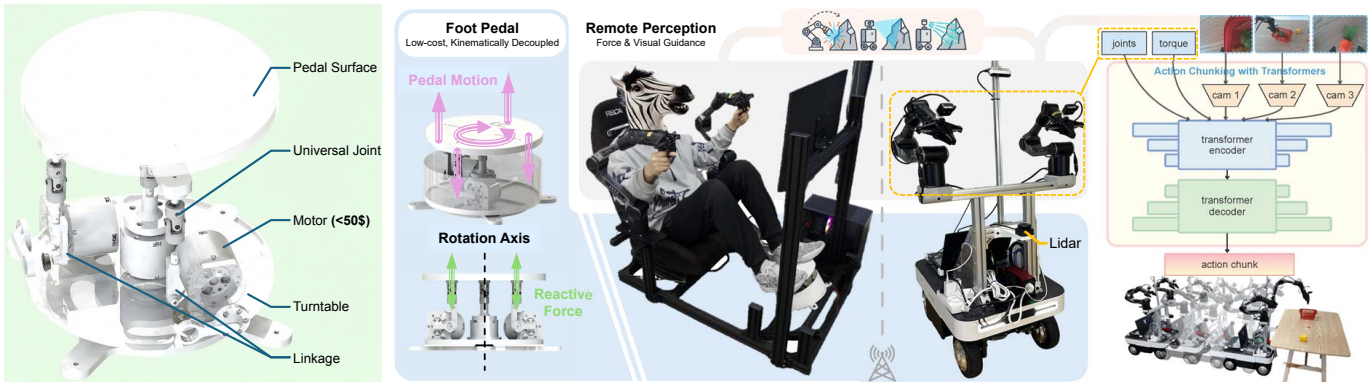


Fig. 2: TriPilot-FF combines bimanual leader-follower arms with a pedal that commands the mobile base. The low-cost pedal has 3 DoF, and the design is shown on the left. An overview of the system is shown on the right. TriPilot-FF provides force feedback on both the operator-side leader arms and pedal based on contact, nearby obstacles, and to adjust the base for greater reachability.

as $J(q) \in \mathbb{R}^{6\times n}$. We use superscripts $(\cdot)^{\text{lead}}$ and $(\cdot)^{\text{fol}}$ for leader and follower arms in the teleoperation pair.

Sensing and frames: The bimanual manipulator controlled by TriPilot-FF includes wrist-mounted cameras on each gripper, along with a head camera to perceive the environment and provide situational awareness. The perception stack requires consistent extrinsics across camera and robot frames. We denote a rigid transform by ${}^{A}\mathbf{T}_{B} \in SE(3)$, which maps coordinates expressed in frame B to frame A.

IV. The TriPilot-FF Teleoperation System

The key systems principle is to allocate whole-body control across complementary human modalities. Specifically, continuous base motion is commanded through a kinematically decoupled foot pedal, while bimanual manipulation is performed via an isomorphic leader-follower arm teleoperation setup. The operator retains full control authority, while the system renders guidance task-relevant constraints: (i) Lidar-driven resistive pedal haptics that discourage collision-prone base commands; (ii) arm-side force reflection that improves contact awareness during manipulation; (iii) a manipulability module provides force cues for when and how to reposition the base when kinematic feasibility is limited.

We begin by outlining the system design and hardware setup in Section IV-A, then we outline the software design in Section IV-B and robot policy learning with data collected with the system in Section IV-C. An overview of TriPilot-FF along and pedal design details shown in Fig. 2.

A. System Hardware

The teleoperation station provides three synchronised modalities: (i) visual input from two realsense [28] wrist cameras, one head camera, and depth sensing from a low-cost 360-degree lidar; (ii) arm and gripper input via bimanual leader-follower arms, i.e., isomorphic mapping from the leader pair to the remote follower pair; (iii) foot input via a 3-DoF pedal that commands the mobile base. The remote robot executes a unified whole-body command consisting of base twist and dual-arm joint targets; concurrently, the robot streams proprioceptive and force-related signals back to the operator for force reflection and guidance. In this work, the AgileX Piper arms are used as leader-follower pairs.

Pedal design: To leverage human lower-limb capability for continuous base control, we introduce a low-cost, kinematically decoupled foot pedal: fine manipulation is handled by the upper limbs while locomotion and heading regulation are managed by the lower limbs. The pedal measures (and can render resistive forces along) three approximately independent control dimensions: x translation, y translation, and yaw. Concretely, the device integrates three motors for sensing and actuation along (x, y, yaw), while the remaining mechanical components are 3D printed for low cost and replication. A turntable-linkage mechanism couples the moving pedal surface to a stationary base, so that the operator's foot motion is transmitted to the corresponding motor and motor-generated resistance is reflected back along the same direction, enabling bidirectional interaction.

Perception calibration: To provide feedback of the reachable region of the robot at the current mobile base position in a consistent base-centred frame, the perception stack must express camera observations and kinematic quantities in a shared robot-centered coordinate system. We assume that the lidar can be placed accurately in the forward-facing direction. In particular, we require accurate rigid transforms for (i) each gripper-to-wrist-camera mount and (ii) the base-to-head camera, so that point clouds, end-effector poses, and manipulability indicators can be overlaid *consistently* in a shared coordinate system with minimal misalignment.

We therefore estimate the unknown extrinsics from multi-view checkerboard observations using a pose-graph style formulation over $SE(3)$. For a set of calibration configurations indexed by i, let ${}^{B}\mathbf{T}_{G,i}$ denote the base-to-gripper transform computed from forward kinematics, ${}^{C_h}\mathbf{T}_{\text{tag},i}$ and ${}^{C_w}\mathbf{T}_{\text{tag},i}$ denote the transform from head-camera and wrist-camera marker observations respectively. We solve for constant transforms ${}^{G}\mathbf{T}_{C_w}$ and ${}^{B}\mathbf{T}_{C_h}$ that best satisfy, for all i,

$$ {}^{B}\mathbf{T}_{G,i}\,{}^{G}\mathbf{T}_{C_w}\,{}^{C_w}\mathbf{T}_{\text{tag},i} \approx {}^{B}\mathbf{T}_{C_h}\,{}^{C_h}\mathbf{T}_{\text{tag},i}. \tag{2}$$

We optimise the aggregated $SE(3)$ residual across all poses using a Lie-algebra parameterisation and Levenberg–Marquardt, which yields consistent extrinsics for computing

manipulability guidance in the correct frame.

B. System Software

At each control cycle, the operator provides (i) leader-arm joint states for bimanual manipulation and (ii) pedal inputs for the base. The robot executes a combined command $a_t = [u_{b,t}; q_t^L; q_t^R]$, where $u_{b,t} = [v_x, v_y, \omega]^\top$ is the base twist and q_t^L, q_t^R are follower-arm joint targets. This unified representation is also logged as the action for policy learning.

Feedback on bimanual leader arms: Both leader and follower arms run a joint-space impedance controller. For each joint $i \in \{1, \dots, n\}$, with measured state $(q_i, \dot{q}_i)$ and desired state $(q_{i,\text{des}}, \dot{q}_{i,\text{des}})$, the commanded torque is

$$\tau_i = K_{p,i}\big(q_{i,\text{des}} - q_i\big) + K_{d,i}\big(\dot{q}_{i,\text{des}} - \dot{q}_i\big) + \tau_{i,\text{ff}}, \quad (3)$$

where $K_{p,i}, K_{d,i}$ are gains and $\tau_{i,\text{ff}}$ is an additive feedforward torque (e.g., model-based compensation or reflected interaction), specified separately for the leader and follower. The asymmetric gain choice (high follower, low leader) preserves accurate tracking while keeping the leader easy to move and the reflected forces relatively undistorted.

Leader-follower mapping: The follower is configured as a high-gain position tracker, with targets copied from the leader, $q_{\text{des}}^{\text{fol}}(t) = q^{\text{lead}}(t)$, and follower feedforward is disabled, i.e. $\tau_{\text{ff}}^{\text{fol}} = \mathbf{0}$ to maintain compliance under unexpected contact. In practice, the control runs at 50 Hz, and joint states are published at 100 Hz.

Gravity-compensated force reflection: To reflect contact while limiting operator fatigue, the leader runs in a gravity-compensated compliant mode. We estimate follower torques τ_i^{fol} from motor currents and subtract model gravity $\hat{g}_i(q^{\text{fol}})$ to isolate interaction torque. We then set the leader feedforward term in (3) to

$$\tau_{i,\text{ff}}^{\text{lead}} = -\big(\tau_i^{\text{fol}} - \hat{g}_i(q^{\text{fol}})\big)\, s_i \; + \; \hat{g}_i(q^{\text{lead}}), \quad (4)$$

which combines leader gravity compensation with scaled reflection of the follower's interaction torque, where $s_i \in [0, 1]$ sets reflection strength.

Collision-avoidance pedal assistance: Base driving failures in whole-body teleoperation are often dominated by near-collisions and small clearance errors, especially when attention is on bimanual contact. TriPilot-FF therefore renders a resistance on the pedal, so that unsafe base motions become physically harder to command.

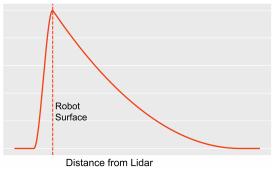


Fig. 3: The potential ϕ produces an increasingly larger repulsive force as obstacles approach the robot.

We leverage the lidar information to produce resistance forces on the pedal. Let the operator's instantaneous intended *translational* base direction in the horizontal plane be $\hat{d} \in \mathbb{S}^1$, computed from the translational command (v_x, v_y), or equivalently from the pedal displacement, i.e., $\hat{d} \propto [v_x, v_y]^\top$ when $\|(v_x, v_y)\| > 0$. Let $r(\hat{d}) \in \mathbb{R}_{\geq 0}$ denote the lidar-estimated free-space distance to the closest obstacle along the ray from the base origin in direction $\hat{d}$. We define a smooth repulsive potential

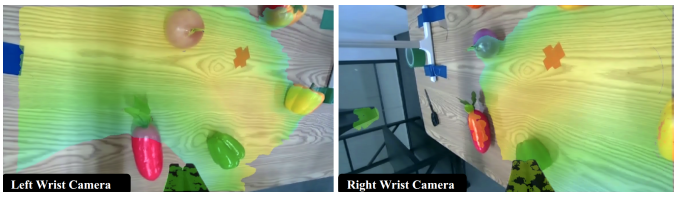


Fig. 4: The manipulable regions highlighted over two wrist camera inputs. This informs the user whether an object is reachable without moving the base.

$\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with a robot surface radius $r_0 > 0$ where potentials near this surface rapidly increase, so that a smaller obstacle distance r produces a larger repulsive effect (Fig. 3). The pedal resistance magnitude is

$$f_{\text{Rep}}(\hat{d}) = k_\phi \left| \frac{\partial \phi}{\partial r}\big(r(\hat{d})\big) \right|, \quad (5)$$

where $k_\phi > 0$ scales the force output. This resistance is applied at the pedal end-effector in the direction opposite to $\hat{d}$, by projecting it onto the available pedal actuation axis. The mapping is *direction-conditioned*: we sum the resistance vector in task space and command the components that reduce the clearance along the composite direction, thereby encouraging collision-averse driving without an explicit collision-avoidance controller while keeping the operator in the loop.

Reachability pedal assistance: In bimanual mobile manipulation, targets frequently lie near or outside the instantaneous workspace; operators must reposition the base while maintaining stable bimanual control. This coordination is a major source of distraction. TriPilot-FF therefore computes a real-time cue that points toward improved dexterity when the arms approach stretched or low-manipulability configurations.

Manipulability metric and field: We use the definition of manipulability in [29], given by

$$w(q) = \sqrt{\det(J(q)J(q)^\top)} \quad (6)$$

as a dexterity proxy, where $q \in \mathbb{R}^n$ are joint angles and $J(q) \in \mathbb{R}^{6\times n}$ is the geometric Jacobian. While $w(q)$ is defined in joint space, our interface requires a *Cartesian* representation: we wish to visualise and guide the operator using a manipulability value associated with an end-effector position $x \in \mathbb{R}^3$, rather than with a particular joint configuration.

Let $f(q) \in \mathbb{R}^3$ denote the forward kinematic mapping from the joints to the end-effector position. We define the *manipulability field* as the best achievable manipulability among all joint configurations that realise a given position:

$$m(x) = \max_{q \in \mathbb{R}^n} \; w(q) \quad \text{s.t.} \quad f(q) = x. \quad (7)$$

Computing $m(x)$ online is impractical, and does not readily yield real-time gradients in x. We therefore learn a differentiable approximation with a neural network, denoted as $\hat{m}(x)$, where $\hat{m}(x) \approx m(x)$ regresses the Cartesian manipulability field. We train $\mathcal{N}_\theta$ from large-scale forward-kinematics sampling. After training, we can then use automatic differentiation to compute $\nabla_x \hat{m}(x)$ efficiently to provide feedback.

The system provides force feedback to help guide operators about *when* and *how* to reposition the base.

Here, we take a weighted combination of the end-effector's

Fig. 5: Left: The bimanual manipulator repositions the bar vertically with both arms, before slipping through the narrow door, and then places the bar back horizontally. Right: The robot carries a box, which obstructs the view, and navigates through avoiding crate obstacles.

current orientation and a gradient direction that ascends manipulability, into a single direction v_{guide}:

$$v_{\text{guide}} = \alpha \frac{x}{\|x\|} + (1-\alpha)\frac{\nabla_x \hat{m}(x)}{\|\nabla_x \hat{m}(x)\|}, \tag{8}$$

where $\alpha \in (0,1)$ is a weighting factor. Projecting onto the horizontal plane gives a pedal feedback direction

$$f_{\text{guide}}(v_{\text{guide}}) = k_{\text{guide}} \frac{[v_{\text{guide}}]_{xy}}{\|[v_{\text{guide}}]_{xy}\|}. \tag{9}$$

Here, $k_{\text{guide}} > 0$ scales the pedal feedback. To preserve operator authority and avoid unnecessary base motion, assistance is activated only when (i) the arm is stretched out and (ii) predicted manipulability is below a threshold.

Manipulability as Visual Guidance: We can also provide manipulability details to the operator visually, rendered in Fig. 4. Here, we expose manipulability to the operator as a scalar dexterity indicator per arm, indicating which regions in Cartesian space are reachable. For each arm, this is achieved by obtaining a point cloud via the camera parameters, then evaluating the manipulability field Eq. (7). This gives us a per-pixel manipulability value, at the current base position. We can filter by a threshold, and manipulability can be optionally visualised by overlaying over the video streams. This enables the operator to efficiently identify when objects can be comfortably reached.

C. Policy Learning

Torque-augmented observation space: The standard Action Chunking with Transformers (ACT) policy conditions on visual observations $\mathbf{I}_t$ and joint positions $\mathbf{q}_t$. We augment the observation space with measured joint torques $\boldsymbol{\tau}_t \in \mathbb{R}^n$, which provide a compact proprioceptive signal of contact, interaction forces, and unmodeled dynamics during teleoperation:

$$\hat{\mathbf{a}}_{t:t+H} = \pi_\theta(\mathbf{q}_t, \boldsymbol{\tau}_t, \mathbf{I}_t), \tag{10}$$

where $\hat{\mathbf{a}}_{t:t+H}$ denotes the predicted length-H action chunk. To enable whole-body control of the mobile manipulator, we extend the action space to include base velocity commands. Specifically, we use

$$\mathbf{a}_t = \left[\mathbf{v}_t^{\text{base}}, \mathbf{q}_t^{\text{arms}}\right]^\top \in \mathbb{R}^{(2n+3)}, \tag{11}$$

where $\mathbf{v}_t^{\text{base}} = (v_x, v_y, \omega)$ are planar base velocities and $\mathbf{q}_t^{\text{arms}} \in \mathbb{R}^{2n}$ stacks the dual-arm joint positions.

We inject torque information into both the encoder and decoder of the ACT via learned linear projections. In the encoder, we compute a torque embedding $\mathbf{e}_\tau = \mathbf{W}_\tau \boldsymbol{\tau}_t$ and concatenate it with the proprioceptive embedding and the action-sequence embeddings, allowing the latent variable to capture correlations between force and demonstrated actions:

$$\mathbf{X}_{\text{enc}} = [\mathbf{e}_{\text{cls}}, \mathbf{e}_q, \mathbf{e}_\tau, \mathbf{E}_a] + \mathbf{P}_{\text{enc}}. \tag{12}$$

In the decoder, the torque embedding provides an additional conditioning signal alongside the latent sample, proprioception, and visual features:

$$\mathbf{X}_{\text{dec}} = [\mathbf{e}_z, \mathbf{e}'_q, \mathbf{e}'_\tau, \mathbf{F}_{\text{img}}]. \tag{13}$$

We use learned embeddings to disambiguate tokens from different modalities. This symmetric integration ensures that force information influences both the latent representation during training and the action generation during inference.

Training and Inference: We train the model with the standard ACT objective, combining an ℓ_1 reconstruction loss over action chunks with KL regularisation:

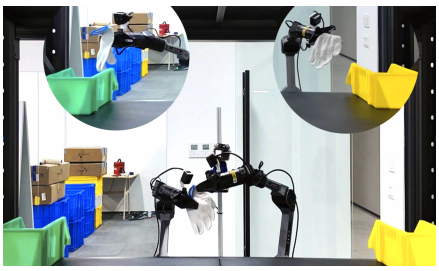
$$\mathcal{L} = \mathbb{E}[\|\hat{\mathbf{a}}_{1:H} - \mathbf{a}_{1:H}\|_1] + \beta\, D_{\text{KL}}(q(\mathbf{z} \mid \cdot) \,\|\, p(\mathbf{z}))\,. \tag{14}$$

At inference time, we sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ from the prior and execute actions using temporal aggregation across overlapping predicted chunks with exponential weighting to improve smoothness and reduce jitter. Real-time joint-torque measurements from the robot are fed directly into the policy, enabling closed-loop, force-reactive behaviour.

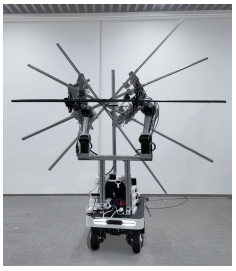
V. Empirical Evaluations

We rigorously evaluate TriPilot-FF across a variety of tasks, both in real-world and simulation environments. Here, we focus our investigation on the following questions:

- **Q1** *Do arm-side force reflection and lidar-driven resistive pedal cues jointly improve coordination in tight or cluttered space?*
- **Q2** *Does differentiable manipulability guidance reduce coordination burden and time-to-completion in reach-limited retrieval?*
- **Q3** *Does arm-side force reflection improve efficiency and contact regulation in constrained contact tasks that require simultaneous base motion and manipulation?*
- **Q4** *Can TriPilot-FF sustain reliable performance and usability over long-horizon, real-world task sequences representative of daily-life and production workflows?*



(a) In the *GuidedReach* task, the base needs to make minor adjustments so the left gripper can reach the glove object, transfer to the right gripper, and place it at another corner of the shelf.



(b) The robot rigidly grips the rod and can swing it in a coordinated manner, owing to the feedback.

Fig. 6: TriPilot-FF enables fine-grained bimanual actions.

Q5 *Do teleoperation haptic signals provide useful supervision for imitation learning, improving downstream policy performance when injected into ACT?*

We teleoperate our robot over a variety of tasks and report metrics over 20 runs.

A. Coordination in Tight Space

TriPilot-FF teleoperation gives rise to coordinated behaviours that enabling mobile manipulators to smoothly traverse tight spaces. Here, we test both the capabilities of the guided arms and mobile base, via the following tasks:

- **BlindCarry** (occluded bimanual transport): The robot carries a bulky object that substantially occludes the forward camera view. The operator must maintain stable bimanual support while navigating through a cluttered environment. This task stresses collision avoidance under poor visibility and bimanual load regulation.
- **NarrowTransport** (constrained passage): The robot transports an elongated bar object through a narrow door passage. Success requires coordinated base motion and bimanual posture adjustments to prevent contacts with the passage boundaries and maintain grasp alignment.

Illustrations of the tasks are provided in Fig. 5. The *BlindCarry* task requires feedback mapped onto the pedal to guide the robot, while the *NarrowTransport* task requires feedback of the arms for the operator to quickly adjust the bar, while simultaneously leveraging the pedal feedback to accurately pass the passage. We report the following metrics in Table I,

- **Success rate** S (%): Percentage of trials successfully completing the task within a fixed time-frame. Here, we consider failure as being in a state that is unable to continue the task, with minor collisions counted.
- **Average collision count** N_{coll}: number of contact events between the base or carried object and the environment. (Multiple contacts within a short window are counted as a single event to avoid over-counting chatter.)
- **Completion time** T (s): Time used to complete the task.

Table I shows that whole-body haptic feedback improves robustness in occluded bimanual transportation. In *BlindCarry*, enabling lidar-driven resistive pedal cues yields a large gain in task reliability, from 55% to 100%, while eliminating average collision count and reducing completion time. This

TABLE I: Coordination in tight spaces. Here, S is success rate, N_{coll} is collision count, and T is completion time.

Task		S (%) ↑	N_{coll} ↓	T (s) ↓
BlindCarry	TriPilot-FF	**100.0**	**0.0**	**131.34**
	w/o Pedal Feedback	55.0	1.0	150.08
NarrowTransport	TriPilot-FF	**100.0**	**0.0**	**16.32**
	w/o Pedal Feedback	75.0	0.3	48.36
	w/o Any Feedback	60.0	0.4	49.16

pattern is consistent with the perceptual bottleneck of the task: When the object carried occludes the forward view, the operator must infer the risk of collision primarily from base-centric sensing. The resistive pedal feedback effectively converts lidar proximity into a "risk gradient" that discourages aggressive commands near obstacles, thereby reducing corrective behaviour and collisions. Here, we highlight that the force reflection provided on the bimanual arms, enables grasps jointly with two grippers which move rigidly relative to each other. The robot can then firmly hold a rigid object with both grippers and swing with around with no slippage, as demonstrated in Fig. 6b.

B. Manipulability guidance for reach-limited retrieval

We seek to evaluate the capability of the *reachability assistance* which guides the operator to adjust the base position to increase reachability when the arms are over-extended. Here, we consider the task of **GuidedReach** (reach-limited retrieval): The robot retrieves a glove object placed near the left corner of a shelf that is just beyond the reachable region from the initial base pose, and is required to place it into a container in the right corner after a handover of the object from one arm to the other. This requires the mobile base to make slight adjustments to refine its position, thereby reaching each of the corners, and isolates the impact of manipulability guidance on base repositioning decisions. We record the time proportion required to complete this task, along with the proportion of time in *low manipulability regions*, defined as regions in joint space that have manipulability values (given in Eq. (6)), below the median. This is denoted as r_{low}.

Table IIa shows that manipulability guidance improves both efficiency and kinematic feasibility in *GuidedReach*. With guidance enabled, the mean completion time drops and the fraction of low-manipulability timesteps decreases. Importantly, these reductions are coupled: the time saving is not simply due to executing faster in the same configurations, but reflects spending less time in kinematically fragile postures, such as those with low dexterity or near singular regions. Guidance accelerates the decision of *when* and *how* to reposition the base, shortening the low-dexterity portion of the trajectory and reducing trial-and-error. This matches the intended role of manipulability: low scores act as a proxy for limited directional capability.

C. Whole-Body Movement for Surface Contact

Force reflection on the robot's upper limbs enables the operator to understand the robot's interaction in contact-rich settings. Here, we consider a task, **MobileSwipe**, with constrained contact in which a roller held by the robot must

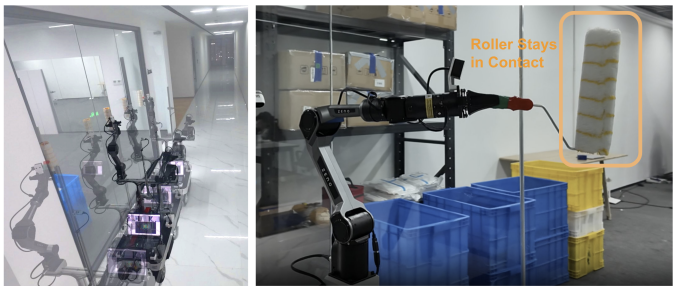


Fig. 7: In the *MobileSwipe* task, the robot needs to swipe the glass surface, moving from one side to the other and back, ensuring the roller remains in contact with the glass.

TABLE II: TriPilot-FF benefits from both arm-side feedback and manipulability guidance on the pedal.

(a) Results on *GuidedReach*.

	T (s) ↓	r_{low} (%) ↓
TriPilot-FF	**7.39**	**30.59**
w/o guidance	15.17	41.04

(b) Results on *MobileSwipe*.

	T (s) ↓	σ_{tor} ↓	P ↓
TriPilot-FF	**30.67**	**2.532**	**0.196**
w/o Force Refl.	32.06	3.239	0.313

maintain tight contact with a vertical glass surface while the base executes a lateral trajectory. This is illustrated in Fig. 7. The task demands coordinated whole-body control: base motion continuously changes the arm's reachable set and effective kinematics, so the arm must regulate contact while compensating for the tool motion induced by the moving base. In practice, removing arm-side force reflection reduces contact awareness and tends to increase unnecessary corrective actions. We capture this by taking the average, over all task episodes, of the following metrics: (1) the standard deviation of torque, σ_{tor}, along the duration of the roller swiping across the glass; (2) the total energy consumption P.

We tabulate the results of the *MobileSwipe* task in Table IIb. We observe that arm-side force reflection improves contact stability and energetic efficiency, rather than markedly reducing completion time. Without force reflection, the completion time is slightly worse, but it substantially worsens contact consistency, as illustrated by the standard deviation of the energy and raises the power consumed. This pattern is expected for constrained contact: operators can often still complete the task without force feedback, but typically do so by making higher-frequency corrective motions by visually inspecting the camera streams to maintain contact and compensate for uncertainty in friction. Force reflection provides a direct signal to the operator for impending contact-state changes (e.g., sticking, slipping, or losing contact), reducing oscillatory corrections and lowering energy expenditure while maintaining comparable task time. In practical terms, force-aware arm control supports sustained contact with less operator effort and reduced mechanical stress. This is particularly important for long-horizon contact tasks where cumulative fatigue and wear can dominate performance.

D. Long-Horizon Teleoperation

While short-horizon benchmarks isolate specific failure modes (collision avoidance, reachability, and contact regulation), real deployments require *task chaining*: the operator must sustain performance over extended episodes while switching between navigation, bimanual manipulation, and

Fig. 8: A long-horizon episode of our robot in a kitchen. Sub-tasks such as opening the fridge door, opening then placing items into a microwave, fetching and pouring milk, and wiping a bench.

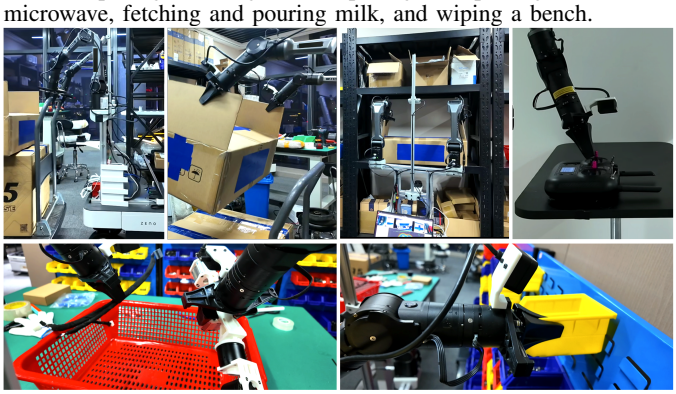


Fig. 9: A long-horizon episode of our robot in a tool room. Sub-tasks include trolley pushing, box handling, shelf logistics, bimanual packing, and operating a game controller.

contact-rich interactions. To evaluate whether TriPilot-FF supports this regime, we conduct long-horizon teleoperation in two representative scenes: a **kitchen** (Fig. 8) and a **tool room** (Fig. 9). In each scene, the operator completes *multiple tasks in one continuous run* without resetting the robot pose, interface state, or environment layout. We provide uncut videos of the robots operating in the supplementary materials.

E. Policy Learning with Torque-Augmented Whole-Body ACT

We evaluate the quality of data collected by TriPilot-FF, and the impact of torque information in policy learning, both in simulation and in real-world deployment. In all our following experiments, ACT models are trained for 20k iterations on NVIDIA RTX 4090 GPUs.

Validation in Simulation: In simulation, we evaluate two tasks, which we set up in MuJoCo [30]: **CubeTransfer** requires grasping a randomly placed cube with one arm and transferring it to the other arm; **CubePickPlace** requires picking a cube from a randomly selected start pose and placing it at a goal location while coordinating mobile-base motion and dual-arm manipulation. We deploy the same whole-body control and logging stack in simulation, collecting 100 trajectories for training, and evaluating 50 rollouts of ACT policies. To keep supervision consistent across tasks, we use a staged

TABLE III: Results in MuJoCo.

Task	Method	S (%) ↑	r_{avg} ↑
CubeTransfer	ACT	22.0	826
	ACT + Torque	**50.0**	**1222**
CubePickPlace	ACT	28.0	1685
	ACT + Torque	**36.0**	**1714**

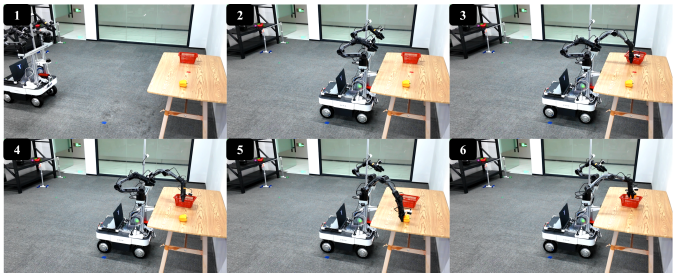


Fig. 10: The robot is placed at a random distance away from the table. The autonomous policy navigates the robot to the table, then it moves the basket to the center of the table, and places a toy pepper (in yellow) into the basket.

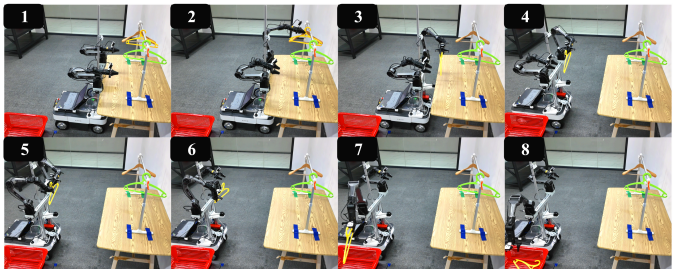


Fig. 11: Autonomous execution of a policy in an out-of-distribution scenario, where many hangers not seen in training are placed on the rack. The robot takes a hanger off, transfers hands, and simultaneously moves to put it into a basket.

progress reward, where the instantaneous reward at each stage is $\{0, 1, 2, 3, 4\}$. Each stage corresponds to a discrete milestone along the task trajectory. We report two evaluation metrics: (i) success rate S, defined as the fraction of episodes that reach the terminal stage within a fixed horizon, and (ii) average cumulative reward r over all runs.

Table III summarises the results. Across both tasks, augmenting ACT with joint-torque signals improves performance, suggesting that torque provides task-relevant information that is difficult to infer from kinematics alone, particularly under bimanual load sharing, transient contacts, and handover dynamics. In *CubeTransfer*, adding torque increases success from 22% to 50% and substantially raises reward, indicating more reliable completion and fewer mid-trajectory failures. In *CubePickPlace*, adding torque on top of base-velocity inputs yields a consistent success gain, from 28% to 36%, with a modest improvement in return, consistent with the task being more driven by navigation.

TABLE IV: We train ACT policies on real-world data collected with TriPilot-FF, and deploy.

Task	Method	S (%) ↑
BasketPack	ACT	12
	ACT + Torque	24
	ACT + Co-training	36
	ACT + Torque + Co-training	**60**
HangerHandOff	ACT	68
	ACT + Torque	**92**
OOD-HangerHandOff	ACT	56
	ACT + Torque	**72**

Real-World Validation: We evaluate policies learned on two constructed real-world bimanual mobile-manipulation tasks: **BasketPack** (Fig. 10), where a robot starts at a random distance away from a table, and needs to approach and stop at the table, then move a basket to the center, and move a toy pepper vegetable into the basket; **HangerHandOff** (Fig. 11), where the robot removes a hanger from a clothes rack, moves while simultaneously performing a self-handover between its hands, and stows the hanger in a bin. For *BasketPack*, we collect 20 trajectories where of whole-body robot motion. Then, similar to [7], we collect 100 trajectories of the robot stationary at the table with only upper-body motion, and add them to the dataset as *co-training* data. For *BasketPack*, as the motion of the upper limbs is highly coupled with the mobile base, we collect 100 trajectories of whole-body robot motion. We rollout 25 episodes per policy, and compute success rates, S, of completing the tasks. To further test the ability for our model to adapt to unexpected events, we evaluate an **out-of-distribution (OOD)** variant of *HangerHandOff*, where additional obstruction hangers, not observed in training, are placed on the rack.

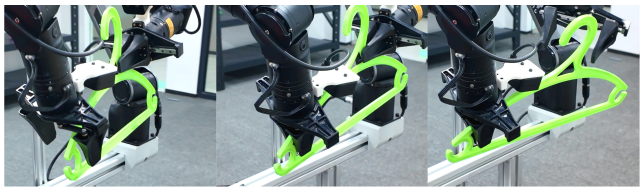


Fig. 12: The torque-augmented ACT policy produces smoother and more coordinated transfer of the hanger while the base is in motion.

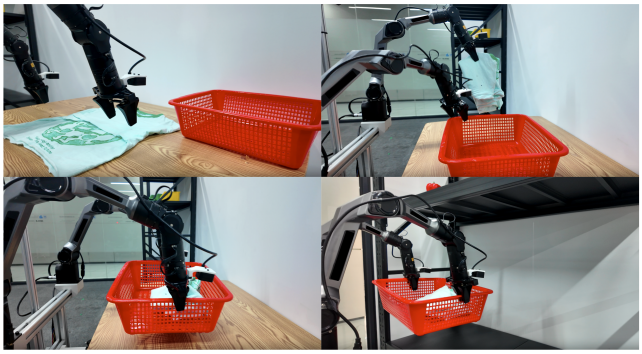


Fig. 13: Long-horizon autonomous execution of *LaundryTransport*. Robot folds a T-shirt, packs it into a basket, and carries it to a shelf.

The performance of the trained ACT policies with and without force information is presented in Table IV. We observe that, for both tasks, injecting the torque information into the state of the ACT policy provides significant performance increases. Additionally, an ACT policy for the *BasketPack* task, trained with 20 trajectories, has a relatively low performance due to the limited amount of data. However, after integrating the additional upper-body motion data to *co-train* the policy the performance improves significantly. Furthermore, we observe that the trained policy can handle the OOD variant with only a modest degradation in performance. Completing the *HangerHandOff* task requires careful coordination between the two manipulators. This behaviour is shown in Fig. 12.

Long-Horizon Autonomous Execution: We additionally stress training autonomous robot behaviour with TriPilot-FF, by training a torque-augmented ACT model on a long-horizon task, **LaundryTransport**. Here, the robot needs to fold a t-shirt, place it into a basket, and then transport the basket to a shelf. Figure 13 illustrates a successful execution of the task, with an additional deployment video in the supplementary.

VI. Conclusion and Future Work

We present TriPilot-FF, a whole-body teleoperation system for bimanual mobile manipulators that combines foot-operated base control with bimanual leader-follower arm teleoperation and force feedback. TriPilot-FF renders resistive pedal cues that improve collision-averse driving in tight or occluded settings, and provides guidance on how to position the mobile base for greater reachability, while arm-side force reflection improves contact regulation in whole-body manipulation. We further show that teleoperation feedback signals are useful for learning: injecting joint-torque observations into a whole-body ACT policy improves imitation performance. Avenues for future research include: (i) adapting cue gains online to operator style for greater personalisation; (ii) providing guidance based on predicted user intent.

References

[1] A. O'Neill and et al., "Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[2] T. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems (RSS)*, 2023.

[3] J. Tan, N. Xi, and Y. Wang, "Integrated task planning and control for mobile manipulators," *The International Journal of Robotics Research*, vol. 22, no. 5, 2003.

[4] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, "Whole-body mpc for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters*, 2019.

[5] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Ffrob: Leveraging symbolic planning for efficient task and motion planning," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 104–136, 2018.

[6] T. Sandakalum and M. H. Ang, "Motion planning for mobile manipulators—a systematic review," *Machines*, vol. 10, no. 2, 2022.

[7] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Conference on Robot Learning (CoRL)*, 2024.

[8] J. Wong, A. Tung, A. Kurenkov, A. Mandlekar, L. Fei-Fei, and S. Savarese, "Error-aware imitation learning from teleoperation data for mobile manipulation," in *Conference on Robot Learning*, 2021.

[9] H. Xiong, R. Mendonca, K. Shaw, and D. Pathak, "Adaptive mobile manipulation for articulated objects in the open world," *arXiv*, 2024.

[10] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Proceedings of The 6th Conference on Robot Learning*, 2023.

[11] W. Zhi, T. Zhang, and M. Johnson-Roberson, "Instructing robots by sketching: Learning from demonstration via probabilistic diagrammatic teaching," in *IEEE International Conference on Robotics and Automation*, 2024.

[12] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint*, 2024.

[13] J. Li, X. Cheng, T. Huang, S. Yang, R. Qiu, and X. Wang, "Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control," *Robotics: Science and Systems*, 2025.

[14] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *International Conference on Intelligent Robots and Systems*, 2024.

[15] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.

[16] S. B. Moyen, R. Krohn, S. Lueth, K. Pompetzki, J. Peters, V. Prasad, and G. Chalvatzaki, "The role of embodiment in intuitive whole-body teleoperation for mobile manipulation," *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*, 2025.

[17] D. Honerkamp, H. Mahesheka, J. O. von Hartz, T. Welschehold, and A. Valada, "Whole-body teleoperation for mobile manipulation at zero added cost," *IEEE Robotics and Automation Letters*, 2025.

[18] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, "Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit," *Robotics: Science and Systems*, 2025.

[19] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, 2016.

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

[21] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *International Conference on Learning Representations*, 2023.

[22] S. Jiang, X. Fang, N. Roy, T. Lozano-Pérez, L. P. Kaelbling, and S. Ancha, "Streaming flow policy: Simplifying diffusion / flow-matching policies by treating action trajectories as flow trajectories," *arXiv preprint*, 2025.

[23] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems*, 2021.

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

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

[26] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[27] J. J. Liu, Y. Li, K. Shaw, T. Tao, R. Salakhutdinov, and D. Pathak, "Factr: Force-attending curriculum training for contact-rich policy learning," *Robotics: Science and Systems*, 2025.

[28] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," *ArXiv*, 2017.

[29] T. Yoshikawa, "Manipulability and redundancy control of robotic mechanisms," in *IEEE International Conference on Robotics and Automation*, 1985.

[30] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

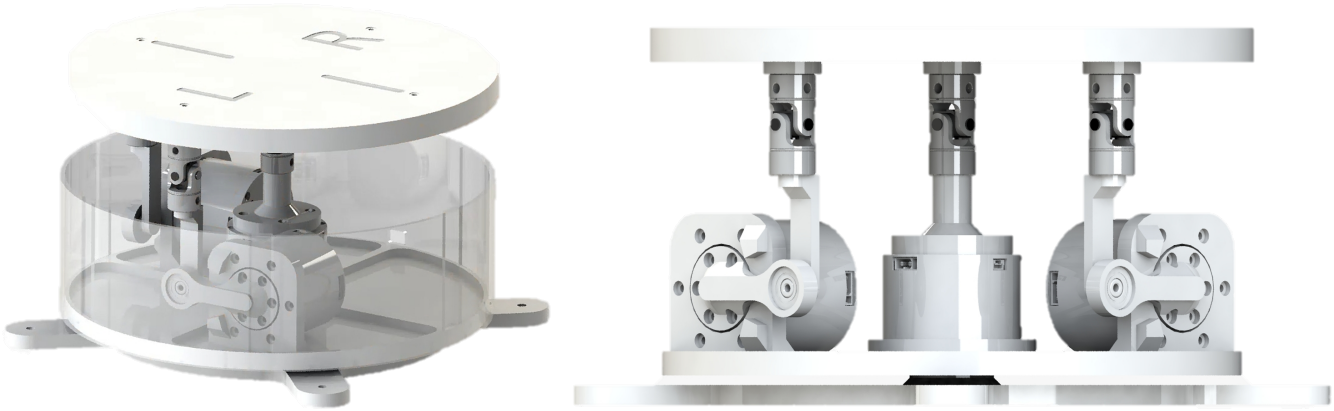


Fig. 14: Detailed structure of foot pedal, including three kinematically independent motors and corresponding linkages for transmission. Because the three axes are perpendicular to each other on the plane, the movements of the two motors on the side will not interfere with each other. All of the components, excluding motors and purchased U-joints, can be 3D printed.

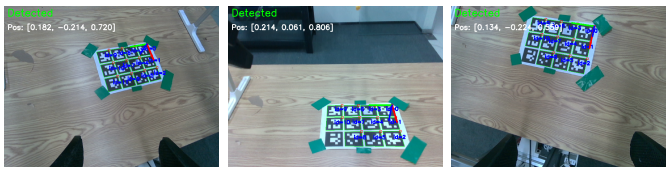


Fig. 15: One of the samples in the joint calibration process (left wrist camera, head camera, and right wrist camera). We sample 10 joint poses, construct several equations of transformation, and jointly solve for constant transforms.

APPENDIX

A. List of Materials

Materials and robots used in our system are listed in Table V.

B. Details of Pedal Design

TriPilot-FF assigns continuous base motion to the lower limbs via a compact, low-cost pedal that provides three approximately decoupled control dimensions: planar translation along the base-frame x/y axes and yaw rotation about the vertical axis. This design choice leverages the operator's ability to regulate locomotion and heading with high bandwidth, while keeping the hands dedicated to bimanual manipulation. The pedal is designed to be bidirectional: it measures the operator's commanded motion and can also render resistive cues along the same motion directions.

Mechanical structure and kinematic decoupling: The pedal consists of a foot contact surface mounted on a turntable-linkage mechanism. A universal-joint and linkage assembly couples the moving pedal surface to a stationary base, transmitting foot motion to three motorised sensing/actuation channels. The mechanism is designed such that each channel dominantly responds to one motion component (forward/backward for x, lateral for y, and rotation for yaw), with residual cross-coupling treated as a small perturbation. Most structural components are 3D printed for ease of replication, while the actuation is provided by low-cost motors (Table V). Detailed structure is shown in Fig. 14.

TABLE V: Material list for both teleoperation platform and bimanual mobile robot.

Category	Name	Supplier/Brand
Teleoperation Platform	Structural Components: Frames	Lab Assembled
	Structural Components of Pedal	3D Printed
	Motors of Pedal	dmBots DM-J4310 (< 50$ per motor)
	Leader Arms	AgileX Robotics PiPER
	U-joints ×3	Generic Supplier
Bimanual Mobile Robot	Structural Components: Frames	Lab Assembled & 3D Print
	Follower Arms	AgileX Robotics PiPER
	DC-AC Inverter	KLW
	Wirst Cameras	Intel RealSense D405
	Head Camera	Intel RealSense D435
	Front Lidar	SLAMTEC C1 Lidar
	Mobile Base	AgileX Robotics Ranger Mini V3

C. Problem formulation

TriPilot-FF overlays perception outputs (e.g., camera point clouds and marker-based estimates) with kinematic quantities (e.g., end-effector pose and manipulability) in a consistent robot-centered coordinate system. This requires estimating constant extrinsics between (i) each gripper and its wrist-mounted camera, and (ii) manipulators and the head camera. We formulate calibration as a pose-graph style estimation problem over $SE(3)$, using multiple robot configurations and multi-view observations of a fiducial target.

Let ${}^{B}\mathbf{T}_{G,i} \in SE(3)$ be the base-to-gripper transform in calibration configuration i (a typical frame is shown in Fig. 15), computed from forward kinematics. Let ${}^{C_h}\mathbf{T}_{\text{tag},i}$ and ${}^{C_w}\mathbf{T}_{\text{tag},i}$ denote the head-camera and wrist-camera target poses estimated from fiducial detection at configuration i. The unknown constant extrinsics are ${}^{G}\mathbf{T}_{C_w}$ (gripper to wrist camera) and ${}^{B}\mathbf{T}_{C_h}$ (base to head camera). For each configuration i, consistency requires that the target pose expressed in the base frame agrees across the two sensing chains:

$$ {}^{B}\mathbf{T}_{G,i}\,{}^{G}\mathbf{T}_{C_w}\,{}^{C_w}\mathbf{T}_{\text{tag},i} \approx {}^{B}\mathbf{T}_{C_h}\,{}^{C_h}\mathbf{T}_{\text{tag},i}. \tag{15}$$

We estimate the unknown transforms by minimising the aggregated $SE(3)$ residual across all calibration configurations. Concretely, define the per-sample error transform

$$ \mathbf{E}_i = \left({}^{B}\mathbf{T}_{G,i}\,{}^{G}\mathbf{T}_{C_w}\,{}^{C_w}\mathbf{T}_{\text{tag},i}\right)^{-1}\left({}^{B}\mathbf{T}_{C_h}\,{}^{C_h}\mathbf{T}_{\text{tag},i}\right), \tag{16}$$

and map it to the Lie algebra using the logarithm map $\log$:

TABLE VI: Hyperparameters of ACT used in our autonomous experiments.

Category	Hyperparameter	Value
ResNet-18 (Visual Backbone)	Pretrained Weights	ImageNet1K_V1
	Replace Final Stride with Dilation	False
	Input Image Size	$244 \times 244 \times 3$
	Number of Cameras	3
	Image Normalisation Mean	$[0.485, 0.456, 0.406]$
	Image Normalisation Std.	$[0.229, 0.224, 0.225]$
Transformer Encoder	Hidden Dimension	512
	Number of Attention Heads	8
	Feed-Forward Dimension	3200
	Number of Encoder Layers	4
	Feed-Forward Activation	ReLU
	Pre-Normalisation	False
	Dropout	0.1
Transformer Decoder	Hidden Dimension	512
	Number of Attention Heads	8
	Feed-Forward Dimension	3200
	Number of Decoder Layers	1
	Feed-Forward Activation	ReLU
	Dropout	0.1
VAE	Latent Dimension	32
	Number of VAE Encoder Layers	4
	KL Weight	10.0
Action Chunking	Chunk Size	100
	Number of Action Steps	100
	Number of Observation Steps	1
	Temporal Ensemble Coefficient	False

TABLE VII: Hyperparameters of the training configuration for ACT policies used.

Category	Hyperparameter	Value
Optimisation	Optimiser	AdamW
	Learning Rate	1e-5
	Backbone Learning Rate	1e-5
	Weight Decay	1e-4
	Batch Size	32

$SE(3) \to \mathfrak{se}(3)$. The objective is then

$$\min_{{}^{G}\mathbf{T}_{C_w},\, {}^{B}\mathbf{T}_{C_h}} \sum_{i=1}^{N} \left\| \log(\mathbf{E}_i)^{\vee} \right\|_2^2, \tag{17}$$

where $(\cdot)^{\vee}$ converts a matrix in $\mathfrak{se}(3)$ to its 6D twist vector. This formulation naturally supports joint optimisation of multiple constant extrinsics using data from diverse robot poses.

D. Optimization Details

a) Data collection and preprocessing: We collect $N = 10$ calibration configurations that span diverse end-effector positions and orientations. For each configuration i, we (i) record synchronised robot joint states to compute ${}^{B}\mathbf{T}_{G,i}$ via forward kinematics, and (ii) detect the fiducial target in both head and wrist cameras to estimate ${}^{C_h}\mathbf{T}_{\text{tag},i}$ and ${}^{C_w}\mathbf{T}_{\text{tag},i}$. To improve robustness, we discard target detections with low confidence or large re-projection error, and we ensure that the retained samples include nontrivial rotations (not merely pure translations).

b) Nonlinear least squares solver: We solve (17) using Levenberg–Marquardt (LM). The residual for each sample is

$$\boldsymbol{e}_i = \log(\mathbf{E}_i)^{\vee} \in \mathbb{R}^6.$$

We initialise both extrinsics from approximate CAD measurements to start within a reasonable basin of convergence.

c) Termination and validation: We terminate when the relative decrease in objective falls below a small threshold.

TABLE VIII: Hyperparameters of control algorithm.

Category	Hyperparameter	Value
Manipulators	Follower K_p	10.0
	Leader K_p	[0.05, 0.1, 0.1, 0.05, 0.1, 0.1]
	Follower K_d	[0.1, 0.1, 0.01, 0.1, 0.1, 0.1]
	Leader K_d	0.8
	Follower Torque Scale s_i	1.0
	Leader Torque Scale s_i	[0.5, 0.15, 0.6, 0.5, 0.15, 0.6]
Reachability Pedal Assistance	Manipulability Threshold	0.1
	Max Linear Velocity	$0.3(m/s)$
	Max Angular Velocity	$0.5(\text{rad}/s)$
	Stretch Radius	$0.3(m)$
	Weighting Factor α	0.5
Collision-avoidance Assistance	Robot Surface Radius	$0.4(m)$
	Far Repulsive Radius	$0.5(m)$
	Scale Factor k_ϕ	1.0

After convergence, we validate calibration by checking cross-view consistency: transforming the detected target pose into the base frame through both chains in (15) should yield a small $SE(3)$ discrepancy across held-out poses. Empirically, this calibration is sufficient for stable overlay of reachability/manipulability cues and for consistent interpretation of perception outputs in the robot-centered frame.

The hyperparameters used for training the ACT agent are summarised in Table VI. The control parameters are shown in Table VIII. The training configuration is shown in Table VII.