HoMeR: Learning In-the-Wild Mobile Manipulation via Hybrid Imitation and Whole-Body Control

Priya Sundaresan, **Rhea Malhotra**, **Phillip Miao**, **Jingyun Yang**, **Jimmy Wu**, **Hengyuan Hu**, **Rika Antonova**, **Francis Engelmann**, **Dorsa Sadigh**, **Jeannette Bohg**

Stanford University

Abstract: We introduce HoMeR, an imitation learning framework for mobile manipulation that combines whole-body control with hybrid action modes that handle both long-range and fine-grained motion, enabling effective performance on realistic in-the-wild tasks. At its core is a fast, kinematics-based whole-body controller that maps desired end-effector poses to coordinated motion across the mobile base and arm. Within this reduced end-effector action space, HoMeR learns to switch between absolute pose predictions for long-range movement and relative pose predictions for fine-grained manipulation, offloading low-level coordination to the controller and focusing learning on task-level decisions. We deploy HoMeR on a holonomic mobile manipulator with a 7-DoF arm in a real home. We compare HoMeR to baselines without hybrid actions or whole-body control across 3 simulated and 3 real household tasks such as opening cabinets, sweeping trash, and rearranging pillows. Across tasks, HoMeR achieves an overall success rate of $\mathbf{79.17\%}$ using just 20 demonstrations per task, outperforming the next best baseline by $\mathbf{29.17\%}$ on average. HoMeR is also compatible with vision-language models and can leverage their internet-scale priors to better generalize to novel object appearances, layouts, and cluttered scenes. In summary, HoMeR moves beyond tabletop settings and demonstrates a scalable path toward sample-efficient, generalizable manipulation in everyday indoor spaces. Supplementary materials are available at: http://homer-manip.github.io.

Keywords: Imitation Learning, Mobile Manipulation, Whole-Body Control

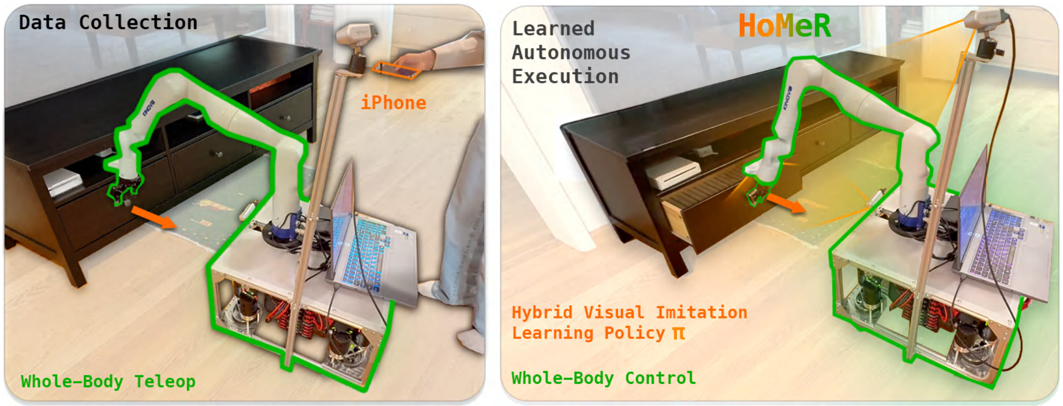


Fig. 1. HoMeR. Left: A demonstrator uses whole-body iPhone teleoperation to collect data with a mobile manipulator in a real home. Right: From these collected demonstrations, HoMeR learns a hybrid imitation learning policy that switches between absolute actions for reaching, and relative actions for fine manipulation. A whole-body controller maps these end-effector commands to arm and base joint commands for execution.

1 Introduction

To unlock the full potential of robots, we must move beyond controlled lab spaces and into the diverse, unstructured environments of everyday life. Unlike stationary tabletop robots, which lack the mobility to perform the wide range of tasks found in homes, offices, and warehouses, mobile manipulators are capable of navigating and interacting in these human-centric spaces. Tasks like watering a row of plants, wiping a spill on a long table, or moving to open a cabinet (Fig. 1) require not only precision, but also manipulation and mobility working hand-in-hand.

Modern mobile embodiments such as wheeled base-arm platforms [1–8], humanoids [9–17], and quadrupeds [18–22] enable robots to operate beyond fixed workspaces. However, these embodiments introduce significant control complexity. In particular, they require careful coordination between the base and arm in wheeled embodiments, or the limbs and torso in legged systems. One common approach to managing this complexity is to use whole-body controllers (WBCs), which map high-level end-effector commands to coordinated whole-body motion using analytical or learning-based approaches. Prior work on WBCs has focused primarily on quadrupeds [23–26], where achieving balance and stability requires learning WBCs from scratch with extensive reward shaping and sim-to-real transfer techniques, often tailored to the dynamics of legged robots. Furthermore, real-world tasks are naturally multi-phase, combining long-range movements (e.g., reaching or repositioning) with fine-grained local manipulation of objects.

On the policy learning side, recent imitation learning (IL) methods [27–29] have shown the benefits of hybrid action spaces, although their application has been restricted to tabletop domains. These policies typically learn to alternate between predicting (1) *keypose* actions (6 DoF absolute end-effector poses, equivalently referred to as *waypoint* [27–29] or *keyframe* [30, 31] actions) for long-range movement and (2) *dense* actions (*i.e.*, delta end-effector actions) for fine-grained interactions. While these approaches achieve strong performance and generalization from limited demonstrations in tabletop settings, they have not been introduced in the mobile manipulation space with greater embodiment complexity, larger workspaces, and longer task horizons.

Our key insight is that scalable mobile manipulation requires not only an effective strategy for managing control complexity, but also a means of generalizing to novel scenarios. As mobile robots move through diverse human environments, they are exposed to far greater variability in objects, spatial configurations, and task conditions than static arms. To address both challenges, we propose **HoMeR**: Hybrid whole-body policies for Mobile Robots. HoMeR (Fig. 1) combines a fast, kinematics-based whole-body controller (which maps end-effector actions to coordinated base-arm motion) with a hybrid IL policy that switches between keypose predictions for long-range movement and dense delta actions for fine-grained manipulation. This structured approach addresses high-dimensional control and multi-phase execution. Additionally, we show that HoMeR is modular enough to incorporate task-relevant keypoints derived from vision-language models (VLMs), providing a path towards improved generalization in unfamiliar environments.

We deploy **HoMeR** in a real home environment and evaluate it on a suite of challenging mobile manipulation tasks that reflect everyday household demands. Overall, our contributions are:

1. A **sample-efficient imitation learning framework for mobile manipulation** that leverages WBC and hybrid action representations to outperform strong non-hybrid and non-WBC baselines using only 20 demonstrations per task.
2. A **modular policy architecture that can be conditioned on VLM keypoints**, enabling generalization to novel object geometries, appearances, and cluttered environments.
3. A **practical whole-body controller that supports intuitive teleoperation**, facilitating efficient demonstration collection in real household settings.

2 Related Work

Mobile Manipulation and Control. *Legged platforms* typically focus on navigation across diverse terrains (*e.g.*, with Boston Dynamics Spot [21] and ANYmal [32] quadrupeds). A few recent works equip quadrupeds with lightweight arms and develop whole-body controllers, utilizing either model-based motion planning (*e.g.*, RoLoMa [33]) or learning-based manipulation policies (*e.g.*, DeepWBC [24], Visual WBC [25], UMI-on-Legs [23]). Our work draws inspiration from these works, but our framework is agnostic to the exact implementation of the WBC (*e.g.*, inverse kinematics (IK) or learning-based). In our work, we use a task-agnostic IK-based WBC that is reusable across many scenarios. Furthermore, in contrast to these prior works, we learn policies with hybrid action modes to encourage both spatial generalization and precision. More recently, *humanoid* research has advanced rapidly, with most efforts centered on whole-body control for expressive behaviors such as dancing, walking, or jumping [10, 11]. Although some humanoid systems perform manipulation, they often rely on using high-dimensional joint-space actions (*e.g.*, 50+ DoF for HumanPlus [12]) for imitation learning. In contrast, our work adopts a whole-body control strategy with hybrid action modes to enable more tractable learning.

Wheeled platforms consist of a wheeled mobile base with onboard arms, and include examples like TidyBot [1], TidyBot++ [2], mobile Franka Pandas [4–6], the Fetch robot [7], the HSR [8], Mobile Aloha [34], and the Everyday Robot [35]. Although these embodiments are physically capable of performing many mobile manipulation tasks, they typically assume decoupled control of the base and arm. Having to switch between different movement modes adds complexity to teleoperation, and often requires the use of ad hoc and task-specific strategies.

A variety of other works study *navigation* with legged or wheeled embodiments [36–38]. Our work is different and complementary in that we focus on the "last mile" of manipulation, where mobility is necessary for task completion, but not at the scale or complexity of full-scene navigation.

Visual Imitation Learning. Visual imitation learning (IL) refers to learning from demonstrations using visual observations [39–41], with recent methods exploring various alternatives for action granularity and input/output space structure.

Dense policies, such as Diffusion Policy [42], ACT [43], or Visual-Language-Action (VLA) models (Gemini [44], π_0 [45], RT-X [46], OpenVLA [47], Octo [48]) predict low-level actions (*e.g.*, 6-DoF deltas or joint velocities) at every timestep. While effective for reactive manipulation, dense policies struggle with long-horizon tasks and spatial generalization, since even simple movements like reaching can involve hundreds of consecutive actions.

Keypose-based policies, such as PerAct [30], RVT/RVT-2 [49, 50], and KITE [51], predict 6-DoF end-effector poses that are executed via low-level controllers or motion planners. While sample-efficient, these keypose actions can be too sparse to handle precise or reactive control.

Recently proposed *hybrid policies* combine keypose and dense actions for both long-range and precise local manipulation. Hydra [29] and AWE [28] use keypose and/or dense actions but rely solely on images, limiting spatial generalization. SPHINX [27], most similar to our approach, uses images and point clouds with learned attention to task-relevant keypoints to switch between modes. Critically, all these methods are limited to static, tabletop-mounted manipulators. We extend SPHINX to the mobile manipulation setting by incorporating whole-body control, enabling mobility while retaining an end-effector–centric action space. We further support optional conditioning on object keypoints from VLMs, allowing generalization to unseen objects in clutter.

3 HoMeR: Hybrid whOle-body policies for MobilE Robots

In the following sections, we describe HoMeR (Fig. 2), our imitation learning framework for mobile manipulation in the wild. Section 3.1 formalizes the problem setup. Section 3.2 presents a kinematics-based *whole-body controller (WBC)* that enables control in a simplified end-effector action space. Finally, Section 3.3 describes our *hybrid imitation learning (IL) agent*, which maps point clouds and RGB image inputs to hybrid actions for both long-range and fine-grained manipulation.

3.1 Problem Formulation

We consider a mobile manipulator composed of a holonomic mobile base and an N-DoF robotic arm, with joint configuration $\mathbf{q}_t = (\mathbf{q}_t^{\textbf{base}}, \mathbf{q}_t^{\textbf{arm}}) \in \mathbb{R}^{3+N}$, where $\mathbf{q}_t^{\textbf{base}} = (x, y, \theta) \in SE(2)$ represents the base pose, and $\mathbf{q}_t^{\textbf{arm}} \in \mathbb{R}^N$ represents the arm joints.

At each timestep t, an observation $o_t = \big(\mathbf{q}_t, g_t, \{\mathbf{I}_t^k, \mathbf{D}_t^k\}_{k=1}^K\big)$ includes the joint configuration $\mathbf{q}_t$, gripper state $g_t \in \mathbb{R}$, and RGB-D images from K cameras (with at least one wrist-mounted and one third-person view). To get 3D point clouds, we assume known camera intrinsics and extrinsics.

Rather than learning actions directly in joint space, our goal is to train an imitation learning (IL) policy $\pi(o_t) = (\mathbf{x}_{t+1}^{\text{ee}}, g_{t+1})$ that predicts a 6-DoF end-effector target pose $\mathbf{x}_{t+1}^{\text{ee}} \in SE(3)$, which can

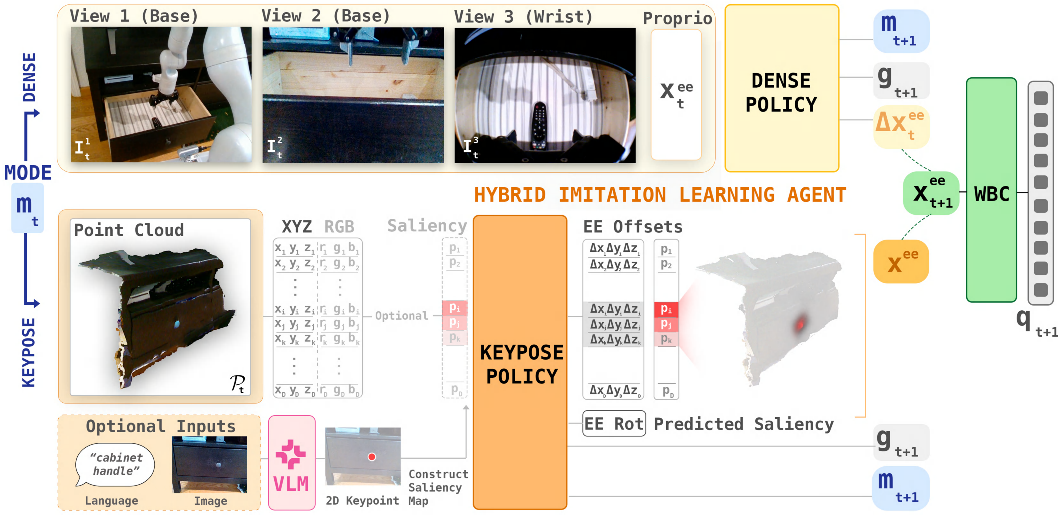


Fig. 2. HoMeR Policy Architecture: HoMeR consists of a *dense policy* that uses RGB images to predict relative actions for fine-grained manipulation, and a *keypose policy* that uses point clouds to predict absolute end-effector poses for long-range motion. Each policy also predicts the next control mode, enabling learned transitions. Optionally, the keypose policy can be conditioned on externally provided salient points derived from a VLM to support dynamic goal specification (HoMeR-Cond). Finally, a whole-body controller (WBC) converts predicted end-effector actions into joint commands for the mobile base and arm.

then be executed by a whole-body controller (WBC). With this formulation, the IL policy reasons in task space while delegating low-level control and embodiment-specific constraints to the WBC.

3.2 Whole-Body Controller

We implement a kinematics-based WBC that maps high-level end-effector poses into joint position commands for the full embodiment, delegating joint-space coordination, constraints, and redundancy resolution to the controller rather than the IL agent. Though HoMeR is agnostic to the exact WBC implementation, ours is based on MuJoCo [52] and the mink IK library [53].

Formally, the WBC is a mapping $\mathcal{W}: SE(3) \to \mathbb{R}^{3+N}$ from a desired end-effector pose $\mathbf{x}^{\text{ee}} \in SE(3)$ to a joint position command $\mathbf{q} \in \mathbb{R}^{3+N}$ for the mobile base and arm. We implement an IK solver based on iterative IK that finds $\mathbf{q}$ minimizing the pose error. To compute a velocity that moves the end-effector toward $\mathbf{x}^{\text{ee}}$, we define a pose error as a body-frame twist [54]:

$$\mathbf{e}^{\text{ee}} = \log\left((\mathbf{x}_t^{\text{ee}})^{-1}\mathbf{x}^{\text{ee}}\right),$$

where $\mathbf{x}_t^{\text{ee}} \in SE(3)$ is the current end-effector pose from forward kinematics. The geometric Jacobian $\mathbf{J}_{\text{ee}}(\mathbf{q}_t) \in \mathbb{R}^{6\times(3+N)}$ maps joint velocities to the induced end-effector twist. At each iteration, the IK solver finds $\dot{\mathbf{q}}$ that minimizes the discrepancy between the Jacobian-induced twist and $\mathbf{e}^{\text{ee}}$, moving the end-effector toward the desired pose. Specifically, the IK solver optimizes the following:

$$\min_{\dot{\mathbf{q}}} \quad \|\mathbf{J}_{\text{ee}}(\mathbf{q}_t)\dot{\mathbf{q}} - \mathbf{e}^{\mathbf{ee}}\|^2_{W_{\text{ee}}} + \|\mathbf{q}_t + \dot{\mathbf{q}}\cdot\Delta t - \mathbf{q}_{\text{retract}}\|^2_{W_{\text{posture}}} + \|\dot{\mathbf{q}}^{\mathbf{base}}\|^2_{W_{\text{damping}}}$$

$$\begin{aligned} \text{s.t.} \quad & \dot{\mathbf{q}}_{\min} \le \dot{\mathbf{q}} \le \dot{\mathbf{q}}_{\max} && \text{(a) Joint velocity limits} && \text{(1a)} \\ & \mathbf{q}_{\min} \le \mathbf{q}_t + \dot{\mathbf{q}}\cdot\Delta t \le \mathbf{q}_{\max} && \text{(b) Joint position limits} && \text{(1b)} \\ & -\mathbf{n}_i^\top \mathbf{J}_i(\mathbf{q}_t)\dot{\mathbf{q}} \le \underbrace{\frac{\gamma(d_i - d_{\min})}{\Delta t}}_{\text{collision margin}} + \epsilon, \quad \forall i \in \mathcal{C} && \text{(c) Collision avoidance} && \text{(1c)} \end{aligned}$$

Objective: The first term encourages the solved joint motion to move towards the target pose $\mathbf{x}^{\mathbf{ee}}$. The second term encourages the joint configuration $\mathbf{q}_t + \dot{\mathbf{q}}\cdot\Delta t$ to stay close to a neutral resting posture $\mathbf{q}_{\text{retract}} \in \mathbb{R}^{3+N}$, shown in Fig. 3. The third term damps the motion of the base. Weights W_{ee}, W_{posture}, and W_{damping} specify the influence of each term.

Constraints: The optimization is subject to constraints that ensure safe and feasible execution. We impose joint velocity Eq. (1a) and position limits Eq. (1b), $\dot{\mathbf{q}}_{\min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{\max}$ and $\mathbf{q}_{\min} \leq \mathbf{q}_t + \dot{\mathbf{q}} \cdot \Delta t \leq \mathbf{q}_{\max}$, to satisfy hardware bounds. In constraint Eq. (1c), we enforce velocity-based collision avoidance between selected pairs of robot components, each modeled as geometric primitives (*geoms*) in the MuJoCo simulator [52]. For each pair $i \in \mathcal{C}$, we identify the closest points between geoms and compute the signed distance d_i, the contact normal $\mathbf{n}_i$, and the Jacobian $\mathbf{J}_i$ of the contact point with respect to joint motion. The constraint aims to slow the robot's motion as the clearance d_i between geoms approaches a minimum threshold, effectively acting as a velocity damper. In our setup, $\mathcal{C}$ includes the (arm, base) and (arm, camera mount) pairs, where the camera mounts are represented as cylinders.

The IK solver iteratively integrates the optimized joint velocities using a fixed timestep Δt to obtain the final joint position command: $\mathbf{q}_{t+1} = \mathbf{q}_t + \dot{\mathbf{q}} \cdot \Delta t$. The joint position commands are subsequently executed on hardware using low-level controllers. All WBC hyperparameters are given in Appendix A. These values are held constant and reused across all tasks without any per-task retuning.

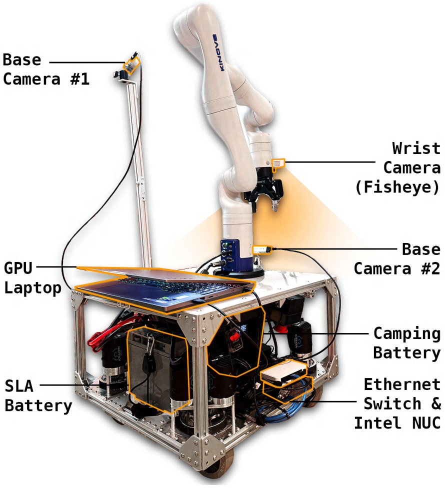


Fig. 3. Hardware: We use the TidyBot++ holonomic mobile manipulator [2] with two base cameras and a wrist-mounted fisheye camera. An onboard NUC handles real-time control, and an onboard GPU laptop runs policy inference.

3.3 Hybrid Imitation Learning Agent

Built on the WBC's end-effector control space, our hybrid imitation learning (IL) agent consists of two sub-policies: a **keypose sub-policy** for long-range motion and a **dense sub-policy** for fine-grained manipulation. Each sub-policy predicts both the next end-effector action and next control mode $m_{t+1} \in \{\texttt{keypose}, \texttt{dense}, \texttt{terminate}\}$, indicating the sub-policy choice for the next action.

Teleoperation. We collect data for HoMeR using the iPhone-based interface from [2], which streams the phone's real-time 6-DoF pose via the WebXR API and maps it to the robot's end-effector. Gripper commands are issued through swipe gestures. The WBC from Section 3.2 solves for joint-space actions at each timestep. We record observations and actions at 10 Hz during teleoperation.

Keypose Sub-policy. The keypose sub-policy π^{keypose} handles long-range movements such as reaching, where predicting an absolute end-effector pose provides greater stability than step-wise deltas. It takes as input a third-person point cloud $\mathcal{P}_t \subset \mathbb{R}^3$, constructed by deprojecting RGB-D images using known intrinsics and extrinsics, and outputs a 6-DoF end-effector pose $\mathbf{x}^{\mathbf{ee}}$, gripper state g_{t+1}, and next control mode m_{t+1}. Following SPHINX [27], we avoid directly regressing the target end-effector pose. Instead, the policy predicts per-point saliency probabilities over the input cloud and per-point 3D offsets to the ground-truth end-effector position. The point with the highest saliency defines the *salient point*—a task-relevant 3D location such as a keypoint on a cabinet handle (Fig. 2). During training, we supervise offset predictions only at points with high predicted or ground-truth saliency, encouraging the model to focus on task-relevant regions (Fig. 2, shaded offsets). End-effector orientation, gripper state, and control mode are predicted using additional learnable tokens. At test time, we apply the predicted offset at the highest-saliency point to obtain the positional component, and combine it with the predicted orientation and gripper state to form the full end-effector action $\mathbf{x}^{\text{ee}}$. We then execute interpolated poses from the current pose $\mathbf{x}_t^{\text{ee}}$ to reach the keypose $\mathbf{x}^{\text{ee}}$. Training the keypose policy requires action labels, mode labels, and salient point annotations. For a given dataset of 20 demos, we post-hoc annotate salient points and modes using a lightweight interface, which takes ∼15 minutes (see Appendix B.1.1). We train the policy using the Transformer-based architecture from SPHINX [27].

Conditioned Keypose Sub-policy. We additionally extend the keypose sub-policy to a salient point-conditioned variant, HoMeR-Cond, which optionally accepts an externally provided 3D keypoint. This enables us to tap into the internet-scale visual and semantic knowledge encoded in vision-language models (VLMs) by prompting them to localize unseen objects in cluttered scenes, and conditioning HoMeR-Cond on the resulting deprojected 3D keypoints (Fig. 2). Taking the original point cloud, we first construct a distance-weighted saliency map, where each point's value is inversely proportional to its distance from the provided keypoint, and the map is normalized to represent probabilities of saliency. We concatenate this saliency map as an additional channel at the input. During training, we apply a masked supervision strategy: in 50% of samples, the conditioned saliency map is masked out, and the model learns to predict both saliency and actions as in the unconditioned setting; in the remaining 50%, we pass the unmodified conditioned saliency map and supervise only the action predictions, with offsets penalized only for points with high ground-truth saliency in the conditioned map. This formulation allows the policy to leverage external guidance when available. We further apply visual augmentations during training to promote generalization: adding randomly generated clusters of points to the input point cloud to mimic distractors, and omitting the RGB channel entirely to reduce overfitting to object appearance.

Dense Sub-policy. The dense sub-policy π^{dense} is intended for fine-grained manipulation near salient points, such as inserting, aligning, or grasping objects. The input consists of both third-person and wrist-mounted RGB images $\{\mathbf{I}_t^k \in \mathbb{R}^{H\times W\times 3}\}_{k=1}^K$, along with the current end-effector state $\mathbf{x}_t^{\text{ee}} \in SE(3)$ computed via forward kinematics from $\mathbf{q}_t$. The dense policy predicts a 6D delta action $\Delta\mathbf{x}_t^{\text{ee}} \in \mathbb{R}^6$ relative to the current end-effector pose (Fig. 2), as well as the next control mode m_{t+1}. We obtain the target pose as $\mathbf{x}_{t+1}^{\text{ee}} = \mathbf{x}_t^{\text{ee}} + \Delta\mathbf{x}_t^{\text{ee}}$. We instantiate π^{dense} using Diffusion Policy [42], which in practice predicts a horizon of 16 future actions and executes 8 before re-planning, rather than predicting a single delta action at each timestep.

Execution. The agent automatically switches between sub-policies based on the current mode m_t:

$$(\mathbf{x}_{t+1}^{\mathbf{ee}}, m_{t+1}) = \begin{cases} \pi^{\text{keypose}}(\mathcal{P}_t) & \text{if } m_t = \text{keypose} \\ \pi^{\text{dense}}(\{\mathbf{I}_t^k\}) & \text{if } m_t = \text{dense} \end{cases}$$

We assume that m_1 corresponds to keypose mode, as nearly all manipulation tasks involve first reaching before performing fine-grained manipulation. For each timestep thereafter, the predicted action $\mathbf{x}_{t+1}^{\mathbf{ee}}$ is passed to the WBC to solve for and execute $\mathbf{q}_{t+1} = \mathcal{W}(\mathbf{x}_{t+1}^{\mathbf{ee}})$ (Section 3.2). HoMeR uses the predicted mode m_{t+1} to select the next sub-policy, enabling dynamic alternation between reaching and manipulation based on learned transitions.

4 Experiments

We deploy HoMeR on the TidyBot++ robot [2], consisting of a 7-DoF Kinova arm and holonomic base (Fig. 3). With this platform, we evaluate a diverse set of challenging manipulation tasks in both simulation and real-world to investigate three core questions, focusing on the benefits of HoMeR's imitation learning (IL) agent, whole-body controller (WBC), and generalization capabilities:

(Q1) *Do hybrid actions help with multi-step tasks combining reaching and fine manipulation?*
(Q2) *Does the WBC action space improve performance compared to decoupled base-arm actions?*
(Q3) *Can* HoMeR *generalize to novel object instances and spatial configurations?*

4.1 Q1 & Q2: Are hybrid actions and whole-body control beneficial?

Baselines. We compare HoMeR to baselines varying along two axes: hybrid vs. dense-only action spaces, and whole-body vs. decoupled base-arm control. Hybrid variants are trained on data annotated post-hoc with control modes and salient points. WBC baselines are trained from whole-body teleoperation demonstrations (Fig. 1), while base+arm (B+A) baselines use decoupled teleoperation, in which the base has to be directly teleoperated separately from the arm [2].

Diffusion Policy (B+A): A dense policy that predicts 10-DoF relative poses: 3-DoF base pose, 6-DoF end-effector pose, and 1-DoF gripper command. This is comparable to the dense, base-arm policies used in [2, 34, 55] which notably were trained with 50-200 demos.

	Task	Description	R-R	R-O	P	LH
Sim	*Cube*	Pick up cube placed randomly across large workspace		✔	✔	
	Dishwasher	Open randomly placed dishwasher door		✔	✔	
	Cabinet Opening	Open randomly placed side-hinged cabinet door		✔	✔	
Real	*Pillow*	Move pillow placed randomly on carpet to target couch position	✔	✔	✔	✓
	TV Remote	Grasp and open cabinet, retrieve remote, place on stand	✔	✓	✔	✔
	Sweep Trash	Grasp brush & sweep at least 3/4 trash clumps into bin.	✔	✓	✔	✔

Tab. 1. Mobile Manipulation Tasks. We evaluate our approach on 3 simulated and 3 real-world tasks, covering randomized robot poses (R-R), randomized object poses (R-O), need for precision (P), and long-horizon reasoning (LH). Darker checkmarks indicate greater emphasis on the corresponding aspect.

HoMeR (B+A): A hybrid policy identical to HoMeR, but predicting either a 3-DoF base keypose, a 6-DoF arm keypose, or a 10-DoF relative pose (as above).

Diffusion Policy (WBC): A dense policy that predicts 7-DoF relative poses: 6-DoF end-effector pose and 1-DoF gripper command, executed through the WBC.

HoMeR (Ours): A hybrid IL agent that predicts either a 6-DoF end-effector keypose or a 6-DoF relative end-effector pose, plus a 1-DoF gripper command, executed through the WBC.

We expect hybrid action modes and whole-body control (WBC) to each provide advantages in tasks involving wide workspaces, precise phases, and long horizons. HoMeR (B+A) and DP (WBC) each capture one of these components, and may perform competitively by partially addressing these challenges. In contrast, DP (B+A), which lacks both hybrid and whole-body action abstractions, must learn base-arm coordination and long-horizon planning from scratch without structural priors, making the learning problem significantly harder.

Our benchmark tasks are described in Table 1. We train and evaluate all methods using 20 demonstrations on 3 simulated and 3 real-world tasks. In Fig. 4, we show illustrations of each task *(top)* and benchmark results *(bottom)*. Across tasks, DP (B+A) struggles the most with reaching and aligning to targets, particularly in tasks like *Cube*, *Dishwasher*, *Cabinet Opening*, and *Pillow*, with significant randomization in either initial robot poses (R-R) or object placements (R-O). The dense end-effector deltas output by the policy often veer off course, leading to failures in reaching pre-manipulation configurations. DP (WBC) exhibits similar limitations without exploiting keyposes, but performs slightly better. We posit that the simplified end-effector action space enabled by the

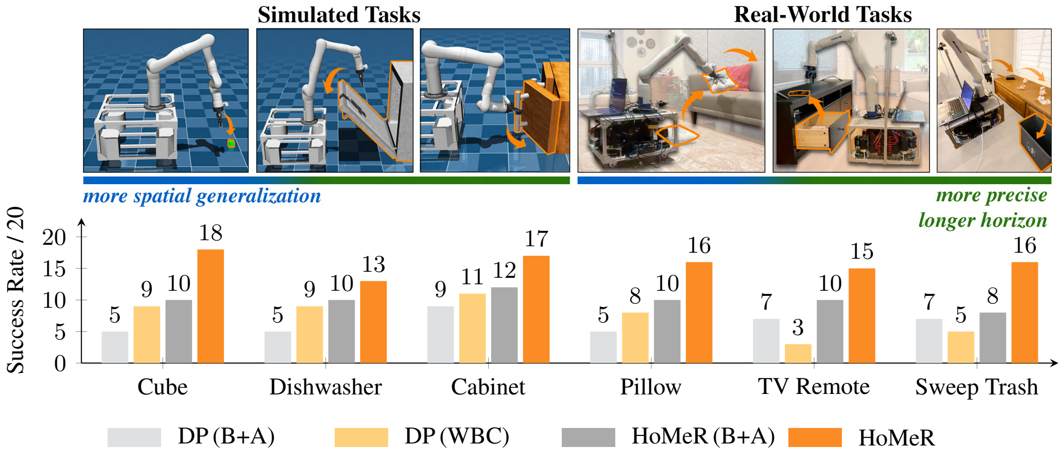


Fig. 4. Benchmarking Results. We evaluate HoMeR on six simulated and real-world tasks (top) that require spatial generalization, precision, and long-horizon reasoning. *TV Remote* and *Sweep Trash* are particularly challenging due to their multi-step nature. HoMeR consistently outperforms baselines that use only dense actions or decoupled base-arm control, highlighting the benefits of hybrid action modes and whole-body coordination. The performance of all methods is best understood through videos available here.

WBC can be beneficial in the low-data regime. HoMeR (B+A) is the strongest baseline, using base and arm keyposes to move to favorable poses before manipulation. This highlights the value of our hybrid IL architecture. However, it struggles when smooth base-arm coordination is required (*Cabinet*, *Dishwasher*, *Sweep Trash*), or when base misalignment affects arm reachability.

HoMeR achieves the highest success rates across tasks (Fig. 4). In particular, HoMeR is able to perform challenging maneuvers like manipulating appliances larger than the robot itself (*Cabinet*, *Dishwasher*), perform smooth long-horizon motions (*Sweep Trash*), execute precise actions (*TV Remote*), and generalize with randomization in both object poses and initial robot poses.

4.2 Q3: Generalization to Novel Scenarios

To assess generalization, we evaluate HoMeR-Cond, a variant of HoMeR that (1) conditions the keypose policy on external salient points from a VLM, and (2) trains on point clouds without color and with randomly generated distractor points to improve visual robustness (Section 3.3). We use MolMo 7B-D [56], a VLM capable of detecting pixel-level keypoints from language prompts (*e.g.*, Fig. 2 detect *"cabinet handle"*), and evaluate HoMeR-Cond on challenging *Cube* variants in simulation: (1) randomizing cube sizes, (2) adding distractors, and (3) retrieving different-colored cubes. Appendix B.1.2 details the language prompts used with MolMo and shows qualitative keypoint predictions. Both HoMeR and HoMeR-Cond-NoAugs perform well in simple settings, but struggle with distractors and novel object appearances. In contrast, HoMeR-Cond maintains high performance, highlighting the combined importance of salient point conditioning and augmentations for handling clutter and unseen objects.

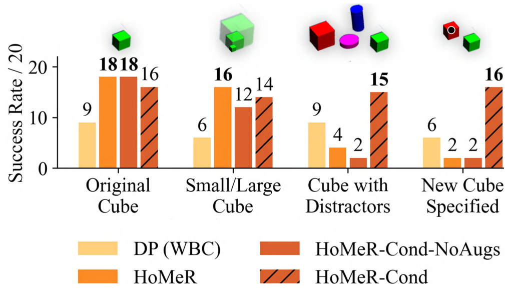
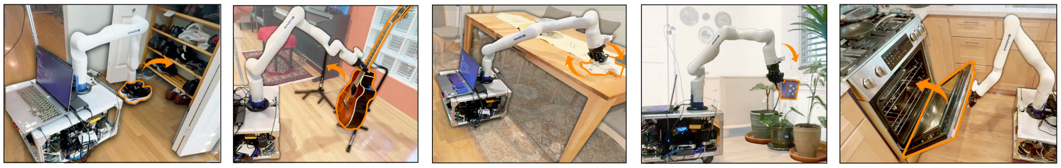


Fig. 5. Generalization Results. HoMeR-Cond achieves strong generalization to unseen scenarios by combining salient point conditioning with point cloud augmentations (videos here). Without augmentations (HoMeR-Cond-NoAugs) or conditioning (HoMeR), performance drops with distractors or novel appearances.

4.3 Qualitative Results: Whole-Body Teleoperation in the Wild

We also qualitatively assess our WBC through teleoperated demonstrations in a real home. The WBC enables smooth, reliable teleoperation of diverse tasks, including opening and closing cabinets, doors, blinds, and ovens; coordinated motions such as wiping tables and watering plants; and precise maneuvers like putting away shoes or moving a guitar between stands (Fig. 6). The WBC optionally avoids collisions between the arm, base, and camera mounts. These results, best viewed on our website, highlight our WBC's potential for scalable in-the-wild teleoperation.



Put Away Shoes **Transfer Guitar** **Wipe Table** **Water Plants** **Close Oven**

Fig. 6. Teleoperated Tasks: We demonstrate a range of teleoperated tasks enabled by our WBC interface, including coordinated whole-body motions and precise behaviors in real household environments.

5 Conclusion

We present HoMeR, a hybrid imitation learning framework for mobile manipulators that combines spatially grounded policy learning with a whole-body controller for executing end-effector actions. By switching between keypose and dense control modes, and operating within a lower-dimensional action space, HoMeR enables generalizable and precise manipulation. Through real-world evaluations in a real home, we demonstrate that HoMeR can perform diverse, everyday tasks with high success rates after training with just 20 demonstrations per task. Our results highlight the benefits of hybrid control and whole-body execution for sample-efficient and generalizable mobile manipulation. We believe HoMeR provides a foundation for scalable deployment of assistive robots in real-world, human-centered environments.

6 Limitations and Future Work

While this work demonstrates the benefits of whole-body control (WBC) and hybrid action representations for mobile manipulation, several limitations remain, as described below.

Collision avoidance. First, although our WBC accounts for self-collisions and posture regularization, it does not consider collisions with the external environment. Recent works such as [57] have proposed point-cloud-based collision avoidance for tabletop manipulators, and incorporating similar constraints into our whole-body controller could enable safe operation in tightly constrained or cluttered environments. We leave this integration as an exciting direction for future work.

Active perception. We use fixed base-mounted and wrist-mounted camera viewpoints that were manually chosen to cover a wide range of tasks. However, the question of which viewpoints are most useful remains underexplored. With a mobile platform, we are particularly excited about incorporating active perception to select or adapt viewpoints dynamically during task execution.

Navigation. We note that this work focuses on manipulation and does not address navigation. In practice, navigation is highly complementary and could be interleaved with our manipulation policies to enable truly long-horizon mobile manipulation.

Multi-task. Lastly, the policies explored in this work are all single-task and demonstrate promising performance in the limited data regime. In the future, we are excited about scaling HOMER to the multi-task setting, and scaling up teleoperation using our interface. A benefit of our action space is that it simplifies to end-effector actions, which opens up the possibility of co-training on other (both mobile and non-mobile) multi-task datasets in the future.

Acknowledgments

We thank TODO

References

[1] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 2023.

[2] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning. *arXiv preprint arXiv:2412.10447*, 2024.

[3] H. P. Brøndmo. Introducing the everyday robot project. https://blog.x.company/introducing-the-everyday-robot-project-27860f3461a4, 2019. Accessed: 2025-04-06.

[4] J. Haviland, N. Sünderhauf, and P. Corke. A holistic approach to reactive mobile manipulation. *IEEE Robotics and Automation Letters*, 2022.

[5] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[6] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.

[7] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. Fetch and freight: Standard platforms for service robot applications. In *Workshop on autonomous mobile service robots*, 2016.

[8] T. Yamamoto, T. Nishino, H. Kajima, M. Ohta, and K. Ikeda. Human Support Robot (HSR). In *ACM SIGGRAPH 2018 emerging technologies*, 2018.

[9] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[10] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive Whole-body Control for Humanoid Robots. *arXiv preprint arXiv:2402.16796*, 2024.

[11] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, L. Paulsen, G. Yang, S. Yi, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[12] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[13] 1X Technologies. 1x world model challenge for humanoid robots, 2025. URL https://github.com/1x-technologies/1xgpt.

[14] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-TeleVision: Predictive Motion Priors for Humanoid Whole-Body Control. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2025.

[15] M. Ji, X. Peng, F. Liu, X. Cheng, R. Yang, G. Yang, and X. Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[16] Unitree Robotics. Unitree h1: Full-size universal humanoid robot, 2025. URL https://www.unitree.com/h1.

[17] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel. HumanoidNench: Simulated Humanoid Benchmark for Whole-body Locomotion and Manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

[18] D. Shah, B. Osinski, B. Ichter, and S. Levine. LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. *arXiv preprint arXiv:2207.04429*, 2022.

[19] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. *Science Robotics*, 2020.

[20] A. Kumar, Z. Xu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. *arXiv preprint arXiv:2107.04034*, 2021.

[21] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, 2023.

[22] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal. Learning to Jump from Pixels. *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2022.

[23] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. In *arXiv preprint arXiv:2407.10353*, 2024.

[24] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning (CoRL)*, 2022.

[25] M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. *The 8th Conference on Robot Learning*, 2024.

[26] J. Brüdigam, A. A. Abbas, M. Sorokin, K. Fang, B. Hung, M. Guru, S. Sosnowski, J. Wang, S. Hirche, and S. Le Cleac'h. Jacta: A versatile planner for learning dexterous and whole-body manipulation. *arXiv preprint arXiv:2408.01258*, 2024.

[27] P. Sundaresan, H. Hu, Q. Vuong, J. Bohg, and D. Sadigh. What's the Move? Hybrid Imitation Learning via Salient Points. *Proc. Int. Conf. on Learning Representations*, 2024.

[28] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn. Waypoint-Based Imitation Learning for Robotic Manipulation. *Conference on Robot Learning (CoRL)*, 2023.

[29] S. Belkhale, Y. Cui, and D. Sadigh. Hydra: Hybrid Robot Actions for Imitation Learning. In *Conference on Robot Learning (CoRL)*, 2023.

[30] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-Actor: A Multi-task Transformer for Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[31] S. James and A. J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

[32] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch, et al. ANYMal - Toward Legged Robots for Harsh Environments. *Advanced Robotics*, 2017.

[33] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar. Roloma: Robust loco-manipulation for quadruped robots with arms. *Autonomous Robots*, 47(8):1463–1481, 2023.

[34] 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.

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

[36] N. Hirose, C. Glossop, A. Sridhar, D. Shah, O. Mees, and S. Levine. LeLan: Learning a Language-conditioned Navigation Policy from In-the-wild Videos. *Conference on Robot Learning (CoRL)*, 2024.

[37] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher. VLFM: Vision-language Frontier Maps for Zero-shot Semantic Navigation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2024.

[38] S. Jauhri, S. Lueth, and G. Chalvatzaki. Active-perceptive Motion Generation for Mobile Manipulation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2024.

[39] S. Schaal. Learning from demonstration. *Advances in neural information processing systems*, 1996.

[40] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[41] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research*, 2023.

[43] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-grained Bimanual Manipulation with Low-cost Hardware. In *Proc. Robotics: Science and Systems (RSS)*, 2023.

[44] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini Robotics: Bringing AI into the Physical World. *arXiv preprint arXiv:2503.20020*, 2025.

[45] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. pi_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[46] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. https://arxiv.org/abs/2310.08864, 2023.

[47] 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.

[48] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[49] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. RVT: Robotic View Transformer for 3D Object Manipulation. In *Conference on Robot Learning (CoRL)*, 2023.

[50] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. RVT-2: Learning Precise Manipulation from Few Demonstrations. *Proc. Robotics: Science and Systems (RSS)*, 2024.

[51] P. Sundaresan, S. Belkhale, D. Sadigh, and J. Bohg. KITE: Keypoint-Conditioned Policies for Semantic Manipulation. In *Conference on Robot Learning*, 2023.

[52] E. Todorov. MuJoCo: A Physics Engine for Model-Based Control. http://www.mujoco.org, 2012. Accessed: 2025-04-15.

[53] K. Zakka. Mink: Python inverse kinematics based on MuJoCo, July 2024. URL https://github.com/kevinzakka/mink.

[54] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[55] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024.

[56] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[57] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak. Neural mp: A generalist neural motion planner. *arXiv preprint arXiv:2409.05864*, 2024.

Appendix: Learning In-the-Wild Mobile Manipulation via Hybrid Imitation and Whole-Body Control

A Whole-Body Controller Implementation Details

We implement the whole-body controller (WBC) described in Section 3.2 using MuJoCo [52] and the mink inverse kinematics library [53].

Model and Tasks. We load the MuJoCo model of the robot, with two camera mounts attached to the base, from an MJCF file. The WBC includes the following tasks:

- **End-effector pose task:** A 6-DoF frame task is defined at the end-effector, with cost weights $W_{\text{ee}} = 1.0$ for both position and orientation tracking.
- **Posture task:** A quadratic penalty encourages the robot to remain near a neutral joint configuration. We set $W_{\text{posture}} = 2 \times 10^{-3}$ for all non-base joints. The target configuration corresponds to a tucked arm posture used across all tasks.
- **Base damping task:** A damping cost of $W_{\text{damping}} = 1.5$ is applied to base velocities to prevent excessive motion.

Constraints. We enforce the following limits during IK:

- **Velocity limits:** Base velocities are capped at $(0.5, 0.5, \pi/2)$ m/s and rad/s. Arm joints are limited to approximately 80°/s for the first four joints and 140°/s for the wrist joints.
- **Joint position limits:** All joint limits of the robot are enforced.
- **(Optional) Collision limits:** We define geometric collision pairs between the arm, gripper, base, and camera mounts, with a 2 cm safety margin and 10 cm detection range. This constraint was not needed in our benchmarking experiments (Fig. 4) as our placement of cameras was not at high collision risk with the whole-body motions, but remains available as a flexible add-on and is demonstrated during teleoperation (Fig. 6).

Solver parameters. We solve the IK problem using mink's QP solver with a Levenberg-Marquardt damping factor of 1.0. The solver runs for up to 20 iterations with a convergence threshold of 10^{-4} for both position and orientation errors. Joint velocities are integrated using Euler integration.

Usage. At runtime, the solver takes as input a desired end-effector pose and the current joint configuration, and returns a joint position command by solving the constrained IK problem and integrating the resulting joint velocities. All weights and thresholds are fixed and reused across all tasks without any per-task tuning.

B Hybrid IL Implementation Details

B.1 Keypose Policy

We implement the keypose policy using a Transformer that operates on point clouds to predict a 6-DoF end-effector pose. The policy first classifies per-point saliency and then regresses a per-point offset to the target end-effector position. Rotation (as quaternions), gripper state, and control mode are predicted using additional learnable tokens. The network architecture uses 6 Transformer layers with 512-dimensional embeddings and 8 attention heads. No positional encodings are used, as the point cloud input is unordered.

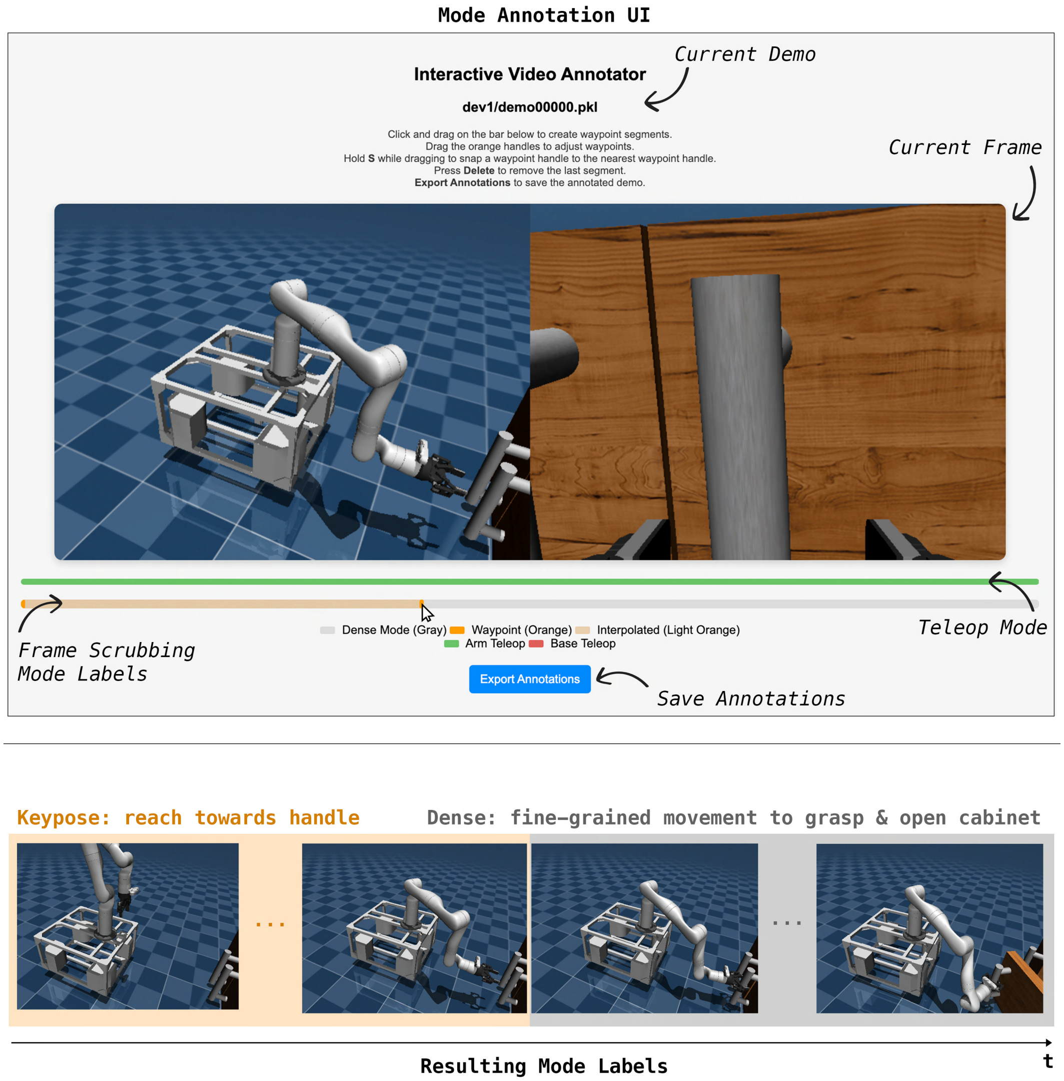
Following [27], the full training objective is a simple unweighted sum of the following: (1) salient point classification loss, (2) offset regression loss on high-saliency points, (3) MSE on normalized quaternions, (4) binary cross-entropy on gripper state, and (5) cross-entropy loss on control mode.

We apply temporal augmentation by including intermediate steps from the controller's motion trajectory toward each annotated keypose. For each waypoint segment, we train not only on the initial observation but also on a prefix of the interpolated segment. We use $\alpha = 0.2$, meaning we sample the first 20% of timesteps in the segment. This increases the data sixfold in most cases and improves performance across tasks.

Additionally, we apply spatial augmentations by randomly translating the entire point cloud and corresponding action label within a 5 cm cube. No vision-based pre-processing or segmentation is used beyond cropping to workspace bounds. We train for 2000 epochs using Adam with a base learning rate of $1e^{-4}$ and cosine decay, gradient clipping (max norm 1), dropout of 0.1, batch size 64, and exponential moving average (EMA) with decay annealed up to 0.9999. All evaluations use the final checkpoint.

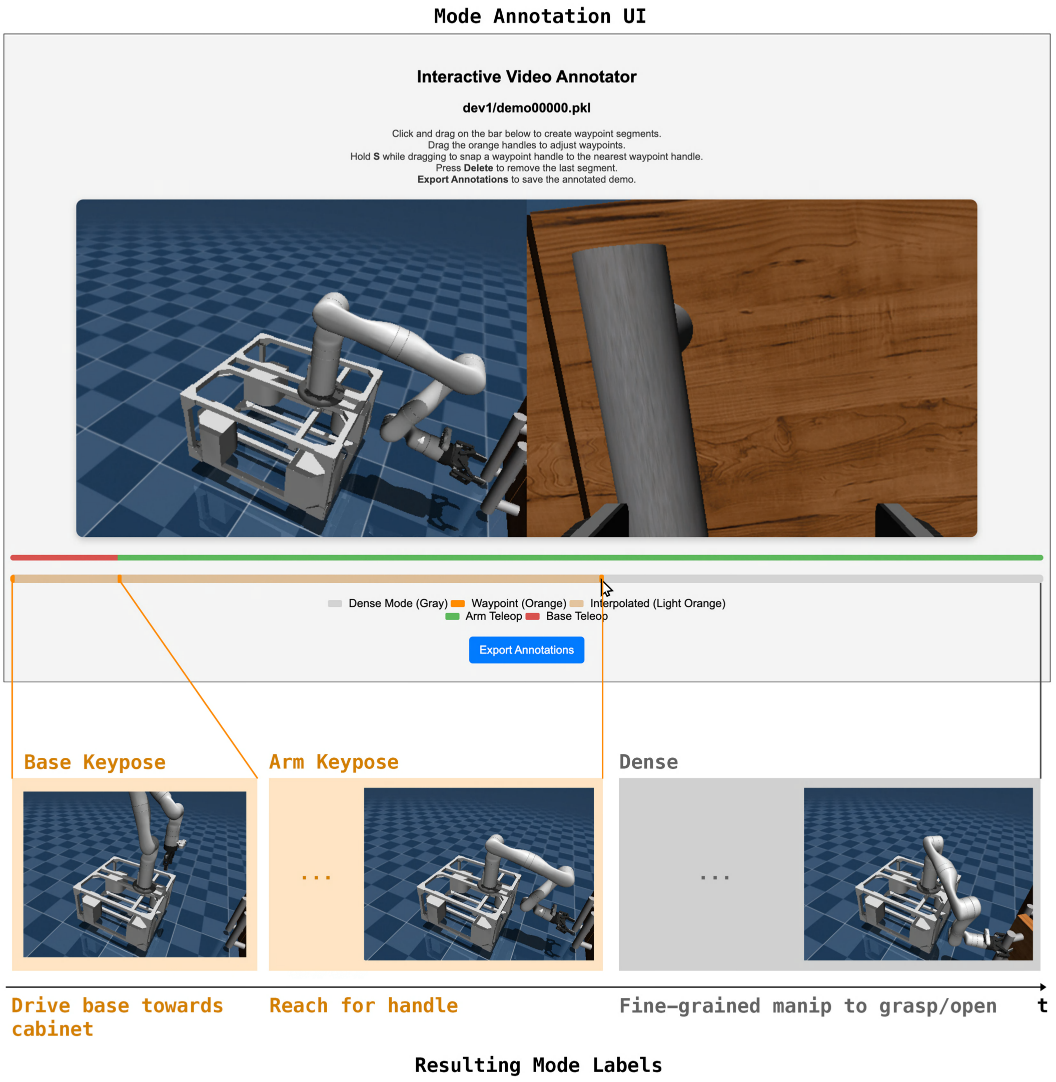
B.1.1 Data Annotation

Training the keypose policy requires labels for modes and salient points. We provide these annotations on teleoperated demonstrations using a lightweight custom interface. As shown in Fig. 7 and Fig. 8, annotators first segment each demonstration into keypose and dense control modes by clicking and dragging on a timeline. For frames labeled as keypose, annotators then specify a salient point by clicking on a task-relevant location in the 3D point cloud interface (Fig. 9). Each demonstration typically contains 1–3 such annotations, and full annotation of a 20-demo dataset takes around 15 minutes. These labels supervise both the saliency classification and offset regression components of the keypose policy.



Click and drag on the bar below to create waypoint segments.
Drag the orange handles to adjust waypoints.
Hold **S** while dragging to snap a waypoint handle to the nearest waypoint handle.
Press **Delete** to remove the last segment.
Export Annotations to save the annotated demo.

Fig. 7. HOMER Mode Annotation Example: To train HOMER, we annotate control modes using a custom UI that supports frame-by-frame scrubbing and Shift + Click/Drag segmentation. For the *Cabinet* task shown above, we label the reaching motion as keypose (orange) and the grasping and opening phase as dense (gray). A demonstration of this annotation process is available here.



Interactive Video Annotator

dev1/demo00000.pkl

Click and drag on the bar below to create waypoint segments.
Drag the orange handles to adjust waypoints.
Hold **S** while dragging to snap a waypoint handle to the nearest waypoint handle.
Press **Delete** to remove the last segment.
Export Annotations to save the annotated demo.

Fig. 8. HOMER (B+A) Mode Annotation Example: Above we visualize mode annotation on a *Cabinet* demo collected with separate base+arm teleoperation, with which to train HOMER (B+A). The demonstration consists of first driving the base towards the cabinet (keypose), reaching the arm towards the handle (keypose), and finally grasping and opening the cabinet (dense). We visualize the base (red) or arm (green) control mode for more intuitive labeling.

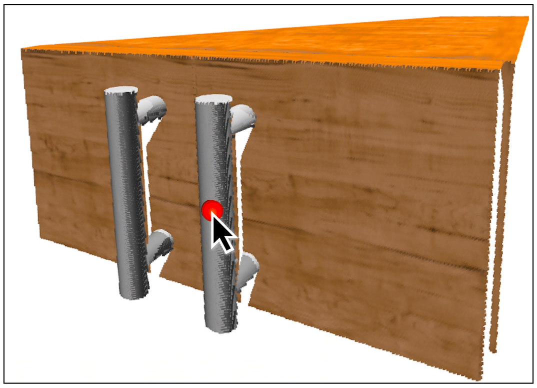


Fig. 9. Salient Point Annotation Interface. For frames labeled as keypose using the Mode Annotation Interface Fig. 7, demonstrators can specify a task-relevant salient point (i.e. the handle in the *Cabinet* task) by simply clicking a desired point within the reconstructed point cloud (video here). Together, these lightweight interfaces provide all the additional supervision necessary to train the keypose policy.

B.1.2 Salient-Point Conditioned Keypose Policy

To improve robustness and generalization, we extend the keypose policy to accept externally specified salient points rather than learning to predict them from scratch. These points are encoded as a soft saliency map over the input point cloud and allow the keypose model to attend to a pre-specified point.

We train this variant with a masked supervision strategy. 50% of the time, we include the saliency map, and the policy learns to predict actions relative to given salient points when available. In the other 50%, we mask out the saliency map, and the model learns to predict the map in addition to the action, in order to encourage the model to learn useful features of the input point cloud. We also apply data augmentations to the input point clouds, including removal of color channels and injection of distractor points, to improve visual robustness in cluttered scenes.

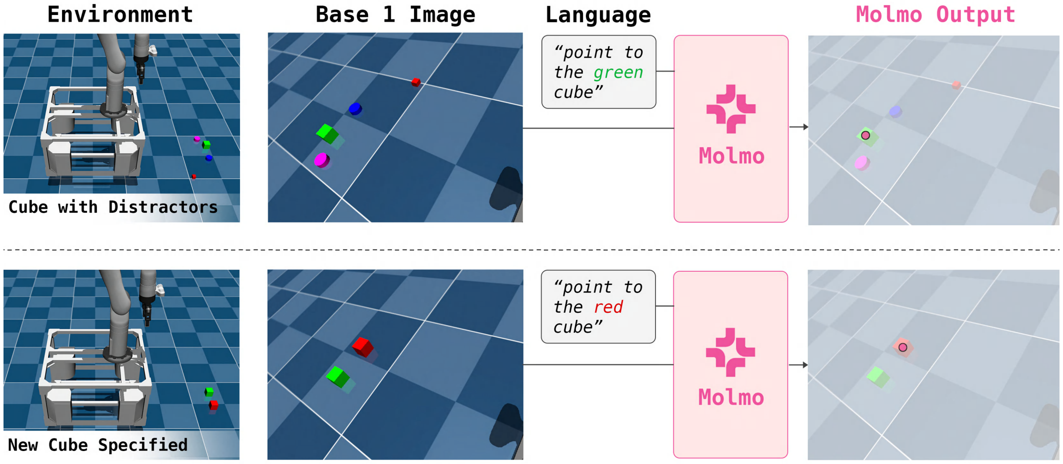


Fig. 10. Extracting Salient Points from MolMo. In the *Cube* generalization tasks (Section 4.2), we use MolMo 7B-D [56] to detect task-relevant keypoints (●) from third-person images given language prompts like "point to the green cube." The predicted pixel is backprojected into the 3D point cloud and used as the salient point input to the keypose policy. The top row shows correct selection among distractors; the bottom row shows generalization to a novel red cube.

In our experiments (Section 4.2), we consider variants of the *Cube* task, where the goal is to pick up a cube subject to different environment variations. Salient points are extracted using MolMo 7B-D [56], a vision-language model that returns pixel-level keypoints given a language prompt. We use the prompt "point to the <green/red> cube," and backproject the returned pixel into 3D to obtain the salient point (Fig. 10).

While we use a fixed prompt for this simple, single-object task, our architecture is agnostic to the exact number of salient points and which objects or object parts they refer to. Future work can explore more complex settings that involve multiple objects, dynamic keypoint selection, and more general VLM prompting strategies that evolve with the task phase.

B.2 Dense Policy

We implement the dense policy of HoMeR using a diffusion model that predicts fine-grained delta end-effector motions. Following [42], we use a ResNet-18 encoder to process RGB images and append proprioceptive features before passing them to a 1D convolutional UNet denoiser. The model is trained using DDPM to predict noise added to delta action sequences.

At test-time, the policy predicts a future horizon of 16 actions and executes the first 8 before replanning. Observations include third-person and wrist-mounted RGB images. We train the model using the Adam optimizer with cosine learning rate decay and weight decay regularization. The policy used for evaluation is the final saved checkpoint.