Human2LocoMan: Learning Versatile Quadrupedal Manipulation with Human Pretraining

Yaru Niu[1*], Yunzhe Zhang[1*],
Mingyang Yu[1], Changyi Lin[1], Chenhao Li[1], Yikai Wang[1], Yuxiang Yang[2], Wenhao Yu[2],
Tingnan Zhang[2], Zhenzhen Li[3], Jonathan Francis[1,3], Bingqing Chen[3], Jie Tan[2], and Ding Zhao[1]
[1]Carnegie Mellon University [2]Google DeepMind [3]Bosch Center for AI
https://human2bots.github.io

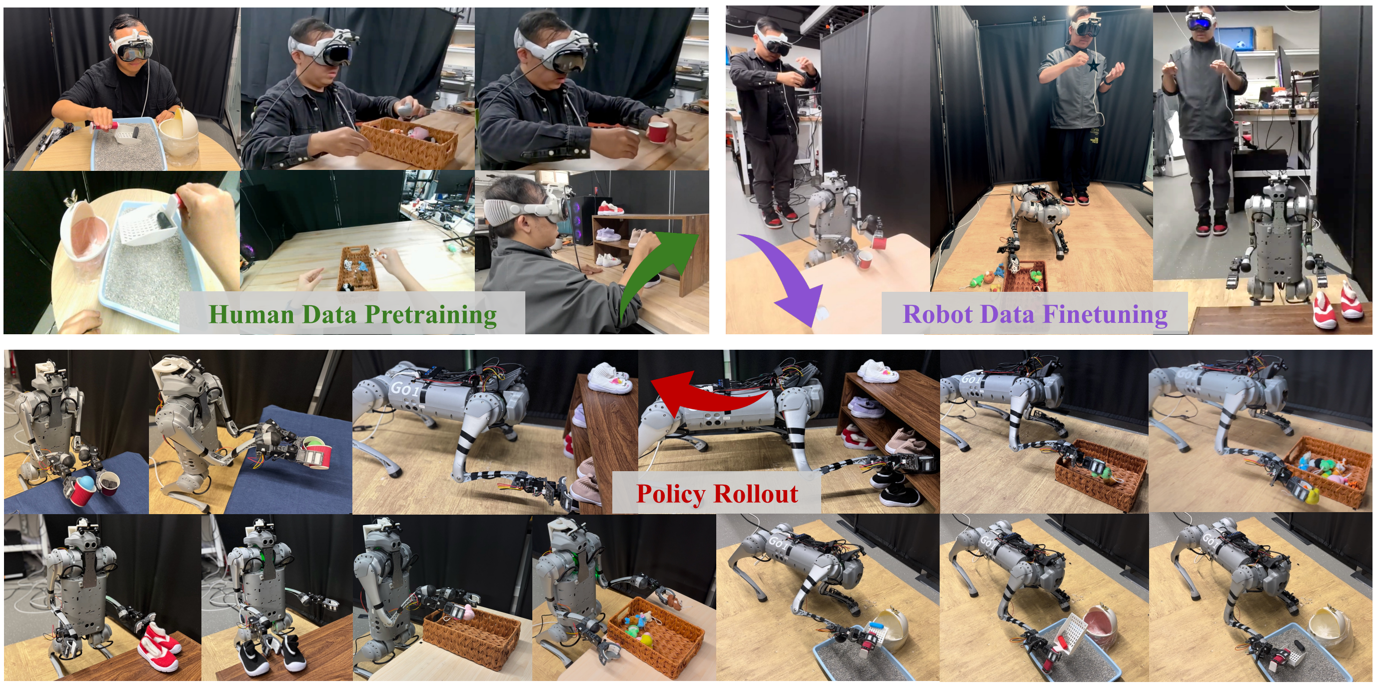

Fig. 1: **Human2LocoMan** provides a unified framework for collecting human demonstrations and teleoperated robot whole-body motions, enabling flexible and scalable data collection. Human data is used for cross-embodiment model pretraining, while robot data is leveraged for policy finetuning. Human2LocoMan achieves positive transfer from human to quadrupedal embodiments, facilitating versatile quadrupedal manipulation.

***Abstract*—Quadrupedal robots have demonstrated impressive locomotion capabilities in complex environments, but equipping them with autonomous versatile manipulation skills in a scalable way remains a significant challenge. In this work, we introduce a system that integrates data collection and imitation learning from both humans and LocoMan, a quadrupedal robot with multiple manipulation modes. Specifically, we introduce a teleoperation and data collection pipeline, supported by dedicated hardware, which unifies and modularizes the observation and action spaces of the human and the robot. To effectively leverage the collected data, we propose an efficient learning architecture that supports co-training and pretraining with multimodal data across different embodiments. Additionally, we construct the first manipulation dataset for the LocoMan robot, covering various household tasks in both unimanual and bimanual modes, supplemented by a corresponding human dataset. Experimental results demonstrate that our data collection and training framework significantly improves the efficiency and effectiveness of imitation learning, enabling more versatile quadrupedal manipulation capabilities. Our hardware, data, and code are open-sourced at: https://human2bots.github.io.**

*Authors contributed equally to this work.

I. Introduction

While quadrupedal robots have demonstrated impressive locomotion capabilities in complex environments [1, 2, 3, 4, 5, 6, 7], and recent advances have extended their abilities to manipulation tasks [8, 9, 10, 11, 12, 13, 14], enabling autonomous and versatile quadrupedal manipulation at scale remains a major challenge. In this work, we take inspiration from the open-source LocoMan platform [14], a quadrupedal robot equipped with two leg-mounted loco-manipulators, which offers a versatile foundation for learning manipulation skills across multiple operating modes. Imitation learning has long been a fundamental approach for teaching robots complex skills through demonstrations [15], with the acquisition of high-quality data being critical for achieving efficient and effective learning. Prior works have explored various strategies for collecting in-domain robot data, primarily focusing on robot arms [16, 17, 18, 19], humanoid robots [20, 21, 22], and quadrupeds equipped with top-mounted arms [10, 11, 23]. However, collecting egocentric manipulation data on a quadrupedal platform like LocoMan remains underexplored. To scale up data collection for imitation learning, recent works propose leveraging simulation data [24, 25, 26] or human data [17, 27, 28, 29, 30, 31]. Human data, in particular, have been used to provide high-level task guidance [17, 28], improve visual encoders [29], simulate in-domain robot data [27, 30], or serve as additional training data by treating humans as an alternative embodiment with similar kinematic structures [31]. However, transferring skills from humans to quadrupedal robots remains challenging due to the substantial embodiment gap, which complicates both data collection and policy transfer. To address these challenges, we propose `Human2LocoMan`, a unified framework that bridges the human-to-quadruped gap. `Human2LocoMan` introduces a novel teleoperation and data collection system that aligns human and robot data, coupled with a modular transformer-based architecture for robust cross-embodiment learning. Together, these components enable scalable learning of versatile manipulation skills on quadrupedal robots.

Specifically, to enable scalable data collection, our system leverages an extended reality (XR) headset to capture human motions while streaming a first-person or first-robot (during teleoperation) view to the operator. For human data collection, the operator simply wears the XR headset and performs tasks naturally. During teleoperation, we align the human and quadruped into a unified coordinate frame to bridge the embodiment gap. In addition to mapping human hand motions to the robot's grippers, we map human head motions to the robot's torso, expanding the robot's workspace and enhancing active sensing capabilities. Target poses are then passed to a whole-body controller to generate coordinated robot motions.

In contrast to works that use egocentric human data to pretrain vision encoders [29] or learn high-level intent [17], we treat the human as another embodiment and use human data for cross-embodiment learning. Despite mapping human and robot data to a unified frame, there exist obvious gaps ranging from differences in dynamics to extra wrist cameras on the robot. Thus, we design a modular transformer architecture, ***M**odularized **Cross**-embodiment **T**ransformer* (MXT), which shares the transformer trunk, but has embodiment-specific tokenizers / detokenizers. To enable positive transfer, the MXT policy is first pretrained on human data and subsequently finetuned with a small amount of robot data. We evaluate our approach on six household tasks, across both unimanual and bimanual manipulation modes. Our results demonstrate strong task performance by MXT compared to competitive baselines, effective positive transfer from human demonstrations to robot policies, and increased robustness to both in-distribution (ID) and out-of-distribution (OOD) scenarios.

In summary, our paper provides the following contributions:

- We propose `Human2LocoMan`, a framework that enables flexible and scalable data collection of human demonstrations and teleoperated robot trajectories for learning versatile quadrupedal manipulation skills.
- We design MXT, a modular transformer architecture that facilitates effective cross-embodiment learning despite large embodiment gaps between humans and quadrupedal robots.
- We introduce the first XR-based teleoperation system and manipulation dataset for the open-source LocoMan [14] hardware platform.
- We demonstrate positive human-to-robot transfer, high success rates, and strong robustness across six challenging household tasks, in both unimanual and bimanual manipulation modes.

II. Related Work

Embodiments for Diverse Loco-Manipulation Skills: Learning manipulation skills on quadrupedal robots has shown promise and popularity in recent years, due to the versatility and mobility of the platforms. Many manipulator configurations and capabilities have been proposed for quadrupeds, including non-prehensile manipulation using the quadruped's legs or body (e.g., dribbling a soccer ball, pressing buttons, closing appliance doors, etc.) [32, 33, 34, 35, 36, 37, 38, 39], using a back-mounted arm for tabletop tasks [8, 40], or using leg-mounted manipulators for spatially-constrained (e.g., reaching toys underneath furniture) or bi-manual manipulation tasks [14]. In this work, we take inspiration from the open-source LocoMan hardware platform [14], with two leg-mounted manipulators, which enable the training of policies across challenging tasks and multiple operating modes.

Learning Versatile Quadrupedal Manipulation: Reinforcement learning (RL) has been used for training individual non-prehensile manipulation skills [32, 33, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46] and for training whole-body controllers to track end-effector poses for uni-manual grasping [8, 9, 10, 47, 48, 49, 50]; here, policies are trained in simulation then transferred to the real robot platform, often with high cost in training complexity and training time. To mitigate some of these issues, imitation learning (IL) allows robots to directly learn from expert demonstrations [15, 51, 52, 53] and thus provides an alternative approach to efficiently acquiring more general manipulation skills [26, 54, 55, 56, 57]. However, collecting robot data for quadrupedal platforms remains challenging, due to their high degrees of freedom and the need for stable whole-body controllers. Prior works have trained non-prehensile quadrupedal manipulation policies by learning from demonstrations collected in simulation [12], or grasping policies for a top-mounted arm using data collected from real-world demonstrations [10, 11, 13]. Our work introduces a scalable way of achieving more versatile manipulation skills on quadrupedal platforms encompassing both unimanual and bimanual manipulation tasks, using a small amount of robot data combined with human demonstrations collected via our novel teleoperation and data collection system.

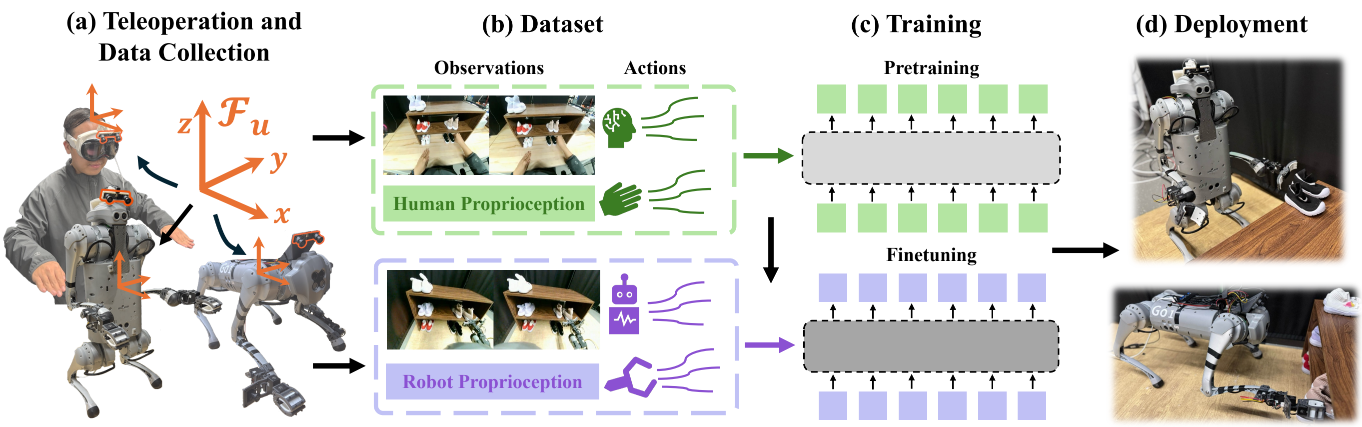

Fig. 2: **Human2LocoMan framework.** (a) The data collection system leverages an XR headset to collect egocentric human data and teleoperated robot data. Human and robot data are mapped to a unified coordinate frame. (b) The dataset consists of aligned vision, proprioception, and actions from the human and the robot. (c) During training, the network is first pretrained on easy-to-collect human data, and then finetuned on a small amount of robot data. (d) We evaluate the autonomous `Human2LocoMan` policies on six household tasks in unimanual and bimanual modes.

Data Collection for Imitation Learning: Various methods have been utilized to collect data for imitation learning. Joysticks and spacemouses [16, 58, 59] are commonly used to directly teleoperate the robot for data collection. Cameras are employed to capture human motions and map them to the robot [17, 20, 60, 61, 62]. VR controllers provide a more intuitive way for the human to teleoperate the robot with visual or haptic feedback for dexterous manipulation tasks on robot arms, quadrupeds, and humanoid robots [13, 21, 22, 31, 63, 64, 65]. While most works above teleoperate the robot in task space, other works employ ex-skeleton or leader-follower systems to collect robot demonstrations by mapping the joint positions of the leader system to the robot [18, 19, 23, 31, 66]. To ease the burdens of teleoperating real robots and to scale up data collection, recent works have achieved success by collecting human demonstrations in the wild with AR-assisted [30] or hand-held grippers [11, 67], though these are limited to a specific robot or end-effector type. Other works enable more ergonomic data collection with body-worn cameras [27, 68] or VR glasses [31]. We introduce a unified framework to collect cross-embodiment data including both robot and human demonstrations, where the teleoperation system considers the whole-body motions of the embodiments to extend its workspace and actively sense the environment. The different manipulation modes of both the robot and human are regarded as different embodiments and the collected data can be used for model pretraining.

Cross-Embodiment Learning: Drawing from the success of foundation models in computer vision and natural language, there have been many endeavors to replicate the success in robotics by training generalist policies on large-scale data from different embodiments [69, 70, 71, 72, 73, 74]. However, this remains an open challenge due to the heterogeneity of robot embodiments, and gaps in kinematics, vision, and proprioception.

Different architectures have been proposed to handle the heterogeneity. CrossFormer [73] formulates policy learning as a sequence-to-sequence problem, so that any number of camera views or proprioceptive sensors can be handled as sequences of tokens, and adds special readout tokens as part of the input sequence. In comparison, HPT [74] features a modularized structure and maps the variable observations to a fixed number of number tokens. In our work, we propose Modularized Cross-embodiment Transformer (MXT) that also employs a modularized design, but further enhances the modularity by identifying fine-granular alignment of data modalities between embodiments.

Notably, EgoMimic [31] treats humans as another embodiment and demonstrates positive transfer by co-training on both human and robot data. To enable such transfer, EgoMimic minimizes the kinematic gap by selecting a human-like robot embodiment, reduces the proprioception gap by normalizing and aligning action distributions, and addresses the appearance gap through visual masking. In comparison, `Human2LocoMan` offers greater flexibility and scalability, achieving positive transfer from humans to multiple quadrupedal embodiments without requiring explicit domain alignment.

III. Methodology

In this section, we describe the design and implementation of our system `Human2LocoMan`, which integrates teleoperation, data collection, and neural architecture for cross-embodied learning.

A. *`Human2LocoMan` System Overview*

We utilize the Apple Vision Pro headset and the Open-Television system [21] to capture human motions and stream first-person or first-robot video to the human operator. A lightweight stereo camera with a 120-degree horizontal field of view is mounted on both the VR headset and the LocoMan robot to provide egocentric views, while additional cameras, such as RGB wrist cameras, can be optionally attached to the robot. Through the `Human2LocoMan` teleoperation system (Section III-B), the human operator can control the LocoMan robot to perform versatile manipulation tasks in both unimanual and bimanual modes. In the unimanual mode, we also map human head motions to the robot's torso movements to expand the teleoperation workspace and enhance active sensing. The `Human2LocoMan` system enables the collection of both human and robot data, transforming them into a shared space. Masks are applied to distinguish across different embodiments and manipulation modes. The collected human data are used to pretrain an action model called the ***M**odularized **Cross**-embodiment **T**ransformer* (MXT). The in-domain robotic data collected via teleoperation are used to finetune the pretrained model to learn a manipulation policy that predicts the 6D poses of LocoMan's end effectors and torso, as well as gripper actions.

B. `Human2LocoMan` Teleoperation and Data Collection

A unified frame for both human and LocoMan. To map human motions to LocoMan's various operation modes via VR-based teleoperation—and to enhance the transferability of motion data across different embodiments—we establish a unified reference frame, $\mathcal{F}_u$, to align motions across embodiments. As shown in Figure 2(a), this unified frame is attached to the rigid body where the main camera is mounted. At the embodiment's reset pose, the x-axis points forward, aligned with the workspace and parallel to the ground; the y-axis points leftward; and the z-axis points upward, perpendicular to the ground.

Motion mapping. We map the human wrist motions to LocoMan's end-effector motions, map the human head motions to LocoMan's torso motions, and hand poses to LocoMan's gripper actions. The 6D poses of the human hand, head, and wrist poses in SE(3) in the VR-defined world frame are streamed from the VR set to the `Human2LocoMan` teleoperation server. The human head pose is represented as $(\boldsymbol{x}_{\text{vr}}^{\text{head}}, \boldsymbol{R}_{\text{vr}}^{\text{head}})$, and the wrist poses are $(\boldsymbol{x}_{\text{vr}}^{\text{r-wrist}}, \boldsymbol{R}_{\text{vr}}^{\text{r-wrist}})$ and $(\boldsymbol{x}_{\text{vr}}^{\text{l-wrist}}, \boldsymbol{R}_{\text{vr}}^{\text{l-wrist}})$, where $\boldsymbol{x}_{\text{vr}}^{\cdot}$ denotes the translation and $\boldsymbol{R}_{\text{vr}}^{\cdot}$ denotes the rotation in the VR-defined world frame. Then, the 6D poses can be transformed into the unified frame $\mathcal{F}_u$ $(\boldsymbol{x}_{\text{uni}}^{\cdot}, \boldsymbol{R}_{\text{uni}}^{\cdot}) = (\boldsymbol{R}_{\text{uni}}^{\text{vr}}\boldsymbol{x}_{\text{vr}}^{\cdot}, \boldsymbol{R}_{\text{uni}}^{\text{vr}}\boldsymbol{R}_{\text{vr}}^{\cdot})$, where $\boldsymbol{R}_{\text{uni}}^{\text{vr}}$ is the rotation matrix of the VR-defined frame relative to the unified frame $\mathcal{F}_u$.

To initialize the teleoperation for each manipulation mode, the robot is transferred to a reset pose randomly initialized within a small range, termed as $\boldsymbol{p}_0 = (\boldsymbol{x}_{\text{uni, 0}}^{\text{torso}}, \boldsymbol{R}_{\text{uni, 0}}^{\text{torso}}, \boldsymbol{x}_{\text{uni, 0}}^{\text{r-eef}}, \boldsymbol{R}_{\text{uni, 0}}^{\text{r-eef}}, \boldsymbol{x}_{\text{uni, 0}}^{\text{l-eef}}, \boldsymbol{R}_{\text{uni, 0}}^{\text{l-eef}}, \boldsymbol{\theta}_0^{\text{gripper}})$, including the 6D poses of the torso and both end effectors, and the gripper angles. The human operator starts to teleoperate the robot after a initializing posture. The target pose for the robot at time step t, $\boldsymbol{p}_t^{\text{t}} = (\boldsymbol{x}_{\text{uni},t}^{\text{torso,t}}, \boldsymbol{R}_{\text{uni},t}^{\text{torso,t}}, \boldsymbol{x}_{\text{uni},t}^{\text{r-eef,t}}, \boldsymbol{R}_{\text{uni},t}^{\text{r-eef,t}}, \boldsymbol{x}_{\text{uni},t}^{\text{l-eef,t}}, \boldsymbol{R}_{\text{uni},t}^{\text{l-eef,t}}, \boldsymbol{\theta}_t^{\text{gripper,t}})$, can be expressed as follows.

$$
\begin{aligned}
\boldsymbol{x}_{\text{uni},t}^{\text{torso,t}} &= \boldsymbol{x}_{\text{uni, 0}}^{\text{torso}} + \alpha^{\text{torso}}(\boldsymbol{x}_{\text{uni},t}^{\text{head}} - \boldsymbol{x}_{\text{uni, 0}}^{\text{head}}) \\
\boldsymbol{R}_{\text{uni},t}^{\text{torso,t}} &= \boldsymbol{R}_{\text{uni, 0}}^{\text{torso}}((\boldsymbol{R}_{\text{uni, 0}}^{\text{head}})^\top \boldsymbol{R}_{\text{uni},t}^{\text{head}}) \\
\boldsymbol{x}_{\text{uni},t}^{\text{r-eef,t}} &= \boldsymbol{x}_{\text{uni, 0}}^{\text{r-eef}} + \alpha^{\text{r-eef}}(\boldsymbol{x}_{\text{uni},t}^{\text{r-wrist}} - \boldsymbol{x}_{\text{uni, 0}}^{\text{r-wrist}}) \\
\boldsymbol{R}_{\text{uni},t}^{\text{r-eef,t}} &= \boldsymbol{R}_{\text{uni, 0}}^{\text{r-eef}}((\boldsymbol{R}_{\text{uni, 0}}^{\text{r-wrist}})^\top \boldsymbol{R}_{\text{uni},t}^{\text{r-wrist}}) \\
\boldsymbol{x}_{\text{uni},t}^{\text{l-eef,t}} &= \boldsymbol{x}_{\text{uni, 0}}^{\text{l-eef}} + \alpha^{\text{l-eef}}(\boldsymbol{x}_{\text{uni},t}^{\text{l-wrist}} - \boldsymbol{x}_{\text{uni, 0}}^{\text{l-wrist}}) \\
\boldsymbol{R}_{\text{uni},t}^{\text{l-eef,t}} &= \boldsymbol{R}_{\text{uni, 0}}^{\text{l-eef}}((\boldsymbol{R}_{\text{uni, 0}}^{\text{l-wrist}})^\top \boldsymbol{R}_{\text{uni},t}^{\text{l-wrist}}) \\
\boldsymbol{\theta}_t^{\text{gripper,t}} &= \frac{\boldsymbol{\theta}_{\text{max}}^{\text{gripper}} - \boldsymbol{\theta}_{\text{min}}^{\text{gripper}}}{d_{\text{max}}^{\text{tip}}} \circ \boldsymbol{d}_t^{\text{tip}} + \boldsymbol{\theta}_{\text{min}}^{\text{gripper}}
\end{aligned} \tag{1}
$$

Here, α^{torso}, $\alpha^{\text{r-eef}}$, and $\alpha^{\text{l-eef}}$, are the scaling factors to map human's motions to robot's torso, right end effector, and left end effector, respectively. $\boldsymbol{x}_{\text{max}}^{\text{gripper}}$ and $\boldsymbol{x}_{\text{min}}^{\text{gripper}}$ are the maximum and minimum gripper angles, respectively. $\boldsymbol{d}_t^{\text{tip}}$ represents the distances between the reference finger tips of both human hands at time step t, and $d_{\text{max}}^{\text{tip}}$ is the maximum finger tip distance for the human operator.

Whole-body controller. The robot target pose at time t, p_t^{t}, is calculated from the teleoperation server, and sent to the whole-body controller of the LocoMan robot, which is adapted from the one introduced in [14], a unified whole-body controller designed to track the desired poses of the torso, end effectors, and feet across multiple operation modes. We employ null-space projection for kinematic tracking and quadratic programming for dynamic optimization to compute the desired joint positions, velocities, and torques.

To handle the large embodiment gap between the human and the LocoMan robots, and to facilitate smooth teleoperation of a dynamic quadrupedal platform with whole-body motions, we consider the handling and recovery from robot's joint limits, singularity, and self-collision, and fast motions. We compute the manipulability index as:

$$
I_{\text{mani}} = \sqrt{\det(\mathbf{J}\mathbf{J}^\top)} \tag{2}
$$

to assess the proximity of the target pose to singularity, where $\mathbf{J}$ represents the Jacobian of the robot's manipulator at the target pose. If I_{mani} falls below a predefined threshold τ_{mani}, the target pose is considered near singularity. To detect self-collisions, we utilize the Pinocchio library [75] to compute collision pairs among the robot's body parts. If any of the following conditions are met—joint limit violation, singularity, or self-collision—the whole-body controller tracks p_{t-1}^{t} instead of p_t^{t}. To mitigate rapid movements, we apply linear interpolation between $x_{\text{uni},t}^{\text{torso,t}}$ and $x_{\text{uni},t-1}^{\text{torso,t}}$, $x_{\text{uni},t}^{\text{r-eef,t}}$ and $x_{\text{uni},t-1}^{\text{r-eef,t}}$, $x_{\text{uni},t}^{\text{l-eef,t}}$ and $x_{\text{uni},t-1}^{\text{l-eef,t}}$, as well as $\theta_t^{\text{gripper,t}}$ and $\theta_{t-1}^{\text{gripper,t}}$. Additionally, quaternion interpolation is applied between $\boldsymbol{R}_{\text{uni},t}^{\text{torso,t}}$ and $\boldsymbol{R}_{\text{uni},t-1}^{\text{torso,t}}$, $\boldsymbol{R}_{\text{uni},t}^{\text{r-eef,t}}$ and $\boldsymbol{R}_{\text{uni},t-1}^{\text{r-eef,t}}$, and $\boldsymbol{R}_{\text{uni},t}^{\text{l-eef,t}}$ and $\boldsymbol{R}_{\text{uni},t-1}^{\text{l-eef,t}}$ to smooth large action variations.

Data Collection. We record the robot data $\{\mathcal{D}_t^{\text{R}}\}_{t=1}^T$ during teleoperation, where $\mathcal{D}_t^{\text{R}} = \{\boldsymbol{o}_t^{\text{R}}, \boldsymbol{a}_t^{\text{R}}\}$ is the robot data at time step t including the robot observations o_t^{R} and robot actions a_t^{R}, and T is the episode length. We define the $\boldsymbol{I}_{\text{main},t}^{\text{R}}$ and $\boldsymbol{I}_{\text{wrist},t}^{\text{R}}$ are images obtained from the robot's main stereo camera and

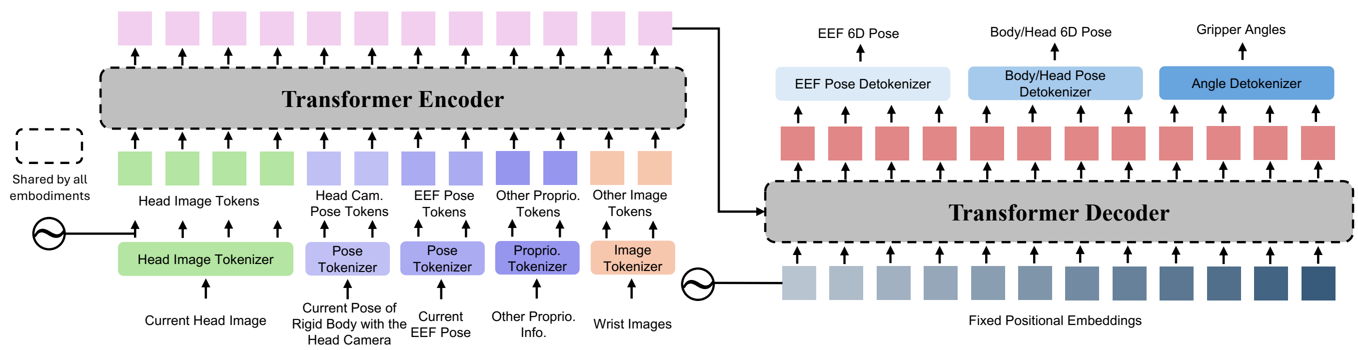

Fig. 3: **Modularized Cross-embodiment Transformer (MXT) architecture.** The inputs are organized as a list of modalities and encoded each by a separate tokenizer into a fixed number of tokens. The transformer trunk handles decision making by consuming the concatenated encoded tokens and producing a fixed number of raw output tokens. Each of the detokenizers at the end decodes a fixed subset of the output tokens into a modality of the final actions.

the wrist camera, respectively. Then, we can formulate o_t^{R} and a_t^{R} in the dataset as follows.

$$\begin{aligned}
\boldsymbol{o}_t^{\mathrm{R}}[\text{main image}] &:= I_{\text{main},t}, \\
\boldsymbol{o}_t^{\mathrm{R}}[\text{wrist image}] &:= I_{\text{wrist},t}, \\
\boldsymbol{o}_t^{\mathrm{R}}[\text{body pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{torso}}, \boldsymbol{R}_{\text{uni},t}^{\text{torso}}], \\
\boldsymbol{o}_t^{\mathrm{R}}[\text{EEF pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{r-eef}}, \boldsymbol{R}_{\text{uni},t}^{\text{r-eef}}, \boldsymbol{x}_{\text{uni},t}^{\text{l-eef}}, \boldsymbol{R}_{\text{uni},t}^{\text{l-eef}}], \\
\boldsymbol{o}_t^{\mathrm{R}}[\text{EEF to body pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{r-eef}} - \boldsymbol{x}_{\text{uni},t}^{\text{torso}}, (\boldsymbol{R}_{\text{uni},t}^{\text{torso}})^{\top} \boldsymbol{R}_{\text{uni},t}^{\text{r-eef}} \\
&\qquad \boldsymbol{x}_{\text{uni},t}^{\text{l-eef}} - \boldsymbol{x}_{\text{uni},t}^{\text{torso}}, (\boldsymbol{R}_{\text{uni},t}^{\text{torso}})^{\top} \boldsymbol{R}_{\text{uni},t}^{\text{l-eef}}], \\
\boldsymbol{o}_t^{\mathrm{R}}[\text{gripper angles}] &:= \boldsymbol{\theta}_t^{\text{gripper}}, \\
\boldsymbol{a}_t^{\mathrm{R}}[\text{body pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{torso, t}}, \boldsymbol{R}_{\text{uni},t}^{\text{torso, t}}], \\
\boldsymbol{a}_t^{\mathrm{R}}[\text{EEF pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{r-eef, t}}, \boldsymbol{R}_{\text{uni},t}^{\text{r-eef, t}}, \boldsymbol{x}_{\text{uni},t}^{\text{l-eef, t}}, \boldsymbol{R}_{\text{uni},t}^{\text{l-eef, t}}], \\
\boldsymbol{a}_t^{\mathrm{R}}[\text{gripper angles}] &:= \boldsymbol{\theta}_t^{\text{gripper, t}}
\end{aligned} \tag{3}$$

We record the human data $\{\mathcal{D}_t^{\mathrm{H}}\}_{t=1}^{T}$ in real time during human's manipulation. Similarly, the human data at time step t $\mathcal{D}_t^{\mathrm{H}} = \{\boldsymbol{o}_t^{\mathrm{H}}, \boldsymbol{a}_t^{\mathrm{H}}\}$ can be defined by human observations $\boldsymbol{o}_t^{\mathrm{H}}$ and human actions $\boldsymbol{a}_t^{\mathrm{H}}$ as follows.

$$\begin{aligned}
\boldsymbol{o}_t^{\mathrm{H}}[\text{main image}] &:= I_{\text{main},t}^{\mathrm{H}}, \\
\boldsymbol{o}_t^{\mathrm{H}}[\text{body pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{head}}, \boldsymbol{R}_{\text{uni},t}^{\text{head}}], \\
\boldsymbol{o}_t^{\mathrm{H}}[\text{EEF pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{r-wrist}}, \boldsymbol{R}_{\text{uni},t}^{\text{r-wrist}}, \boldsymbol{x}_{\text{uni},t}^{\text{l-wrist}}, \boldsymbol{R}_{\text{uni},t}^{\text{l-wrist}}], \\
\boldsymbol{o}_t^{\mathrm{H}}[\text{EEF to body pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{r-wrist}} - \boldsymbol{x}_{\text{uni},t}^{\text{head}}, (\boldsymbol{R}_{\text{uni},t}^{\text{head}})^{\top} \boldsymbol{R}_{\text{uni},t}^{\text{r-wrist}} \\
&\qquad \boldsymbol{x}_{\text{uni},t}^{\text{l-wrist}} - \boldsymbol{x}_{\text{uni},t}^{\text{head}}, (\boldsymbol{R}_{\text{uni},t}^{\text{head}})^{\top} \boldsymbol{R}_{\text{uni},t}^{\text{l-wrist}}], \\
\boldsymbol{o}_t^{\mathrm{H}}[\text{grasping states}] &:= \boldsymbol{\theta}_t^{\text{gripper}}, \\
\boldsymbol{a}_t^{\mathrm{H}}[\text{body pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{head, t}}, \boldsymbol{R}_{\text{uni},t}^{\text{head, t}}], \\
\boldsymbol{a}_t^{\mathrm{H}}[\text{EEF pose}] &:= [\boldsymbol{x}_{\text{uni},t}^{\text{r-wrist, t}}, \boldsymbol{R}_{\text{uni},t}^{\text{r-wrist, t}}, \\
&\qquad \boldsymbol{x}_{\text{uni},t}^{\text{l-wrist, t}}, \boldsymbol{R}_{\text{uni},t}^{\text{l-wrist, t}}], \\
\boldsymbol{a}_t^{\mathrm{H}}[\text{grasping actions}] &:= \boldsymbol{\theta}_t^{\text{gripper, t}}
\end{aligned} \tag{4}$$

In this way, we ensure that the human and robot data are unified in terms of both format and spatial interpretation, and can be used to train our proposed Modularized Cross-Embodiment Transformer introduced in Section III-C.

C. Modularized Cross-embodiment Transformer

To train a policy on LocoMan that benefits from heterogeneous human data, we opt for task-space control in this work, where the actions predicted by the policy are represented as key pose parameters of the physical embodiment, such as the end effector 6D pose and the body 6D pose. While previous works on learning robot skills [20, 22, 31] often choose joint-space action representations for the policy, the fundamental embodiment gap between the human and quadrupedal robots like LocoMan means that the joint spaces for the human and the robot are largely distinct, which will likely hinder the transfer of action prediction capabilities between the embodiments. Moreover, using the task space makes it easy to integrate our data collection pipeline with the unified pose frame into the learning framework.

Given our unified multi-embodiment data collection pipeline, we aim to train a cross-embodiment policy where the overall structure and the majority of parameters are transferrable. To this end, we propose a modularized design called ***M**odularized **Cross**-embodiment **T**ransformer* (MXT). MXT consists mainly of three groups of modules: tokenizers, transformer trunk, and detokenizers. The tokenizers act as encoders and map embodiment-specific observations to tokens in the latent space, and the detokenizers translate the output tokens from the trunk to actions in the action space of each embodiment. The tokenizers and detokenizers are specific to one embodiment and are reinitialized for each new embodiment, while the trunk is shared across all embodiments and reused for transferring the policy among embodiments. Figure 3 illustrates the architecture of our network.

Tokenizers. The tokenizers T transform raw observations into tokens for the transformer trunk. Drawing from the design in previous works [74], we use a cross attention layer to format observational features into a fixed number of tokens. For image inputs, the features are obtained from a pretrained ResNet encoder that can be finetuned during training; for proprioceptive or state-like inputs, the features are computed by passing the raw input through a trainable MLP network.

Detokenizers. The detokenizers D serve as action decoder heads and map output tokens from the trunk to actions in each embodiment's action space. We adopt the action chunking technique [18]. At each inference step, the detokenizers predict an action sequence of h steps and temporal ensemble is applied to the outputs, following [18]. Within each detokenizer, we use a cross attention layer to transform the latent action tokens output by the trunk to a sequence of actions with length h and appropriate action dimensions.

Trunk. The trunk is an encoder-decoder transformer, where the input sequence length and the output sequence length are both fixed, as the number of tokens for each input or output modality is fixed by design. By sharing the trunk weights across the human and robot embodiments, the trunk is trained to capture the common decision making patterns across different embodiments.

Modality Decomposition in Tokenizers / Detokenizers. Due to the aligned data format and the unified observation and action spaces across embodiments, we are able to separately transform each semantically distinct component of the observational input and the action output, which we refer to as *modality*, and specify the compositional structure at the interface of the transformer trunk and the tokenizers / detokenizers. This design provides another layer of modularization to training and is core to the effectiveness of our method.

Concretely, for tokenization in the embodiment e, we encode the input observation $\boldsymbol{o}_t$ with multiple tokenizers $\{T_{e,m_i}\}$ at the finer granularity of modalities denoted by $\boldsymbol{o}_t[m_i]$. For instance, instead of aggregating all image inputs before passing through the vision tokenizer, we use separate tokenizers for each camera view. All the encoded modalities are concatenated to compose the input tokens to the transformer trunk.

Similarly, for detokenization, we specify the subset of the transformer output tokens corresponding to each action modality, e.g. body pose, end effector pose, and gripper angles, and decode the selected tokens to yield each modality with separate detokenizers $\{D_{e,m_i}\}$. For convenience, we use the set of observation and action modalities as defined by the data collection formats in (3) and (4).

By explicitly decomposing the input and output modalities and encoding them separately, we are leveraging the innate structure of observations and actions and imposing such a structure on the token sequences processed by the transformer. Consequently, the knowledge of how to process different modalities learned during training can be shared across embodiments, fostering efficient transfer of the policy.

Although we employ a consistent data format and aligned input/output representations across embodiments, some modalities are not present or available for all embodiments. For example, the human operator is not equipped with a wrist camera, while the LocoMan robot has a wrist camera in some tasks to improve manipulation accuracy. In this case, we use masks defined during data collection to signify redundant dimensions in the observations as well as in the action labels. We refer the reader to Appendix Section VI-A for more implementation details.

In general, the highly modularized design of our learning framework offers great flexibility in handling all types of manipulation tasks across different embodiments, and effectively enhances the learning performance by capturing the common patterns in manipulation problems.

Algorithm 1 Pretraining MXT on human data and finetuning on LocoMan data

Input: Human dataset $\mathcal{D}_{\text{human}}$, LocoMan dataset $\mathcal{D}_{\text{LocoMan}}$
Output: Policy π for versatile LocoMan manipulation
Initialize the MXT policy network π_θ with parameters θ.
Set pretraining learning rate η_{pretrain}
for step $= 1, 2, ...$ **do** ▷ Pretraining Stage
    Sample a batch B from $\mathcal{D}_{\text{human}}$
    Compute $\mathcal{L}_{\text{human}}(B) = \sum_i \mathcal{L}_{\text{human},m_i}(B)$ with Eq.6
    Optimize the policy weights θ with backpropagation
Reinitialize the tokenizers and detokenizers of π. Preserve the trunk weights θ_{trunk} learned from pretraining.
Set finetuning learning rate η_{finetune}
for step $= 1, 2, ...$ **do** ▷ Finetuning Stage
    Sample a batch B from $\mathcal{D}_{\text{LocoMan}}$
    Compute $\mathcal{L}_{\text{LocoMan}}(B) = \sum_i \mathcal{L}_{\text{LocoMan},m_i}(B)$ with Eq.6
    Optimize the policy weights θ with backpropagation
return π

D. Training Paradigm

We leverage the human data to pretrain the network for versatile manipulation policies. Specifically, for a given task, we first pretrain our network with the human dataset, and then finetune it with the LocoMan dataset (Algorithm 1). Only the transformer trunk weights are loaded from the pretrained checkpoint for finetuning. For certain tasks that are similar in nature but with different manipulation modes, we also collectively pretrain the model on the human datasets from these tasks, and then finetune on each task with the corresponding LocoMan dataset.

Learning Objective. We use the behavioral cloning objective for both pretraining and finetuning. In general, given a dataset $\mathcal{D}_e$ on an embodiment e and aligned action modalities $m_1, ..., m_k$, the total loss to optimize when training on e is:

$$\mathcal{L}_e(\theta) = \sum_{i=1}^{k} \mathcal{L}_{e,m_i}(\theta), \tag{5}$$

where $\mathcal{L}_{e,m_i}$ is the ℓ_1 loss of the action modality m_i with respect to the dataset of embodiment e. In practice, we optimize the following batched loss for each training batch $B_e = \{(\boldsymbol{o}_j, A_j)\}_{j=1}^{n}$ as a proxy of $\mathcal{L}_{e,m_i}(\theta)$:

$$\mathcal{L}_{e,m_i}(B_e) = \frac{1}{n} \sum_{j=1}^{n} \left[\frac{1}{h} \sum_{l=1}^{h} \ell_1 \left(\boldsymbol{a}_{j,l}\left[m_i\right], \hat{\boldsymbol{a}}_{j,l}\left[m_i\right] \right) \right], \tag{6}$$

where $\boldsymbol{a}_{j,l}\left[m_i\right] = (A_j)_l\left[m_i\right]$ is the l-th step action of modality m_i in the action label sequence sample $A_j = \{\boldsymbol{a}_{j,l}\}_{l=1}^{h}$; $\hat{\boldsymbol{a}}_{j,l}\left[m_i\right] = \left[\pi_\theta(\boldsymbol{o}_j)\right]_l \left[m_i\right]$ is the predicted action of modality m_i at step l, and h is the chunk size or the action horizon.

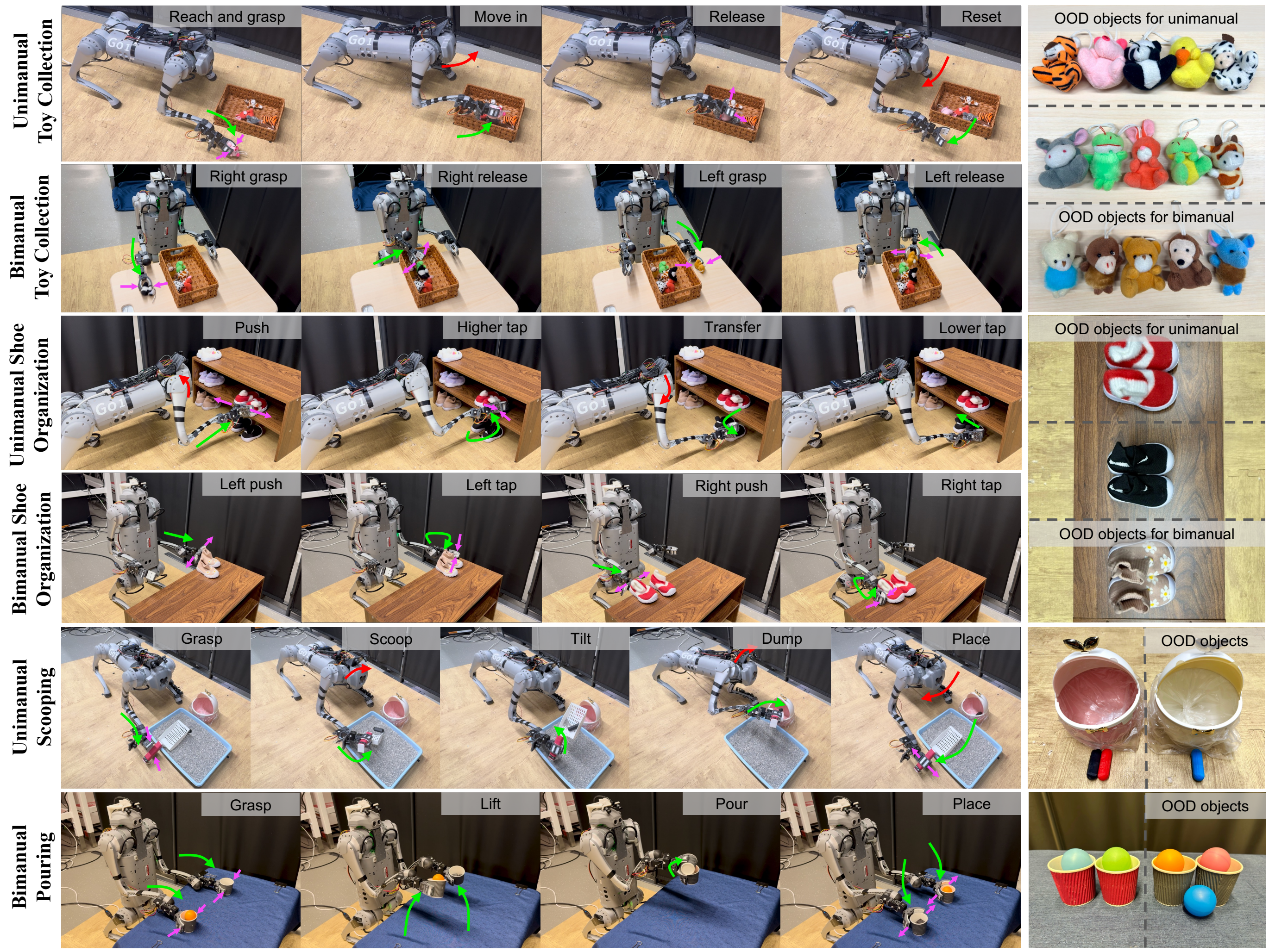

Fig. 4: Rollouts of the MXT policy and the objects used across manipulation tasks in our experiments. Green arrows indicate end-effector motions, red arrows denote torso movements, and pink arrows represent gripper actions. Both **unimanual** and **bimanual toy collection** tasks assess the robot's ability to grasp objects of varying shapes, colors, and positions. The unimanual variant emphasizes coordination between the torso and end-effector, while the bimanual variant highlights synchronized control of two loco-manipulators. **Unimanual** and **bimanual shoe rack organization** tasks evaluate non-prehensile manipulation skills such as pushing and tapping. The unimanual variant additionally requires torso articulation to reach shoes placed at different heights. **Scooping** is a complex task involving tool use, deformable object manipulation, and wide-range torso motion. **Pouring** is a long-horizon task that demands precise coordination of both loco-manipulators.

IV. Experiments

In this section, we aim to answer the following research questions: (1) Does the `Human2LocoMan` system enable versatile quadrupedal manipulation capabilities? (2) How does MXT compare to state-of-the-art imitation learning architectures? (3) How does human data collected by `Human2LocoMan` contribute to imitation learning performance? (4) Do the design choices in MXT facilitate positive transfer from Human to LocoMan?

A. Experimental Setup

1) Tasks: We evaluate MXT on six household tasks of varying difficulty, across unimanual and bimanual manipulation modes of the LocoMan robot, with data collected by the `Human2LocoMan` system:

- *Unimanual Toy Collection (TC-Uni).* In this task, the robot must pick up a toy randomly positioned within a rectangular area and place it into a designated basket on the ground. Completing this task requires the robot to coordinate its whole-body motions to efficiently and accurately reach various locations on the ground and above the basket. As shown in Figure 4, we use 10 objects for robot finetuning and all objects for human pretraining and real-robot evaluation. The substeps of this task include: grasp the toy, and release the toy.
- *Bimanual Toy Collection (TC-Bi).* Similar to *Unimanual*

TABLE I: `Human2LocoMan` embodiments (R=Right, L=Left).

Embodiments	Head Images	Wrist Image	Body Priop.	R-EEF Priop.	L-EEF Priop.	Body Pose	R-EEF Pose	L-EEF Pose	R-Grasp Action	L-Grasp Action
Human-Unimanual (R)	✓	×	✓	✓	×	✓	✓	×	✓	×
Human-Unimanual (L)	✓	×	✓	×	✓	✓	×	✓	×	✓
Human-Bimanual	✓	×	✓	✓	✓	✓	✓	✓	✓	✓
LocoMan-Unimanual (R)	✓	✓	✓	✓	✓	✓	✓	×	✓	×
LocoMan-Unimanual (L)	✓	✓	✓	✓	✓	✓	×	✓	×	✓
LocoMan-Bimanual	✓	×	✓	✓	✓	×	✓	✓	✓	✓

Toy Collection, this task requires the robot to pick up a toy randomly placed within two rectangular areas on either side of a basket. We use 10 objects for robot finetuning, while all objects are included in human pretraining and real-robot evaluation. The substeps of this task include: grasp the toy, and release the toy.

- *Unimanual Shoe Rack Organization (SO-Uni).* This longer-horizon task involves organizing two shoes placed on different levels of a shoe rack. The robot must coordinate whole-body motions to reach various rack levels and utilize both prehensile and non-prehensile manipulation skills. As shown in Figure 4, this task involves three pairs of shoes, with one pair being out-of-distribution (OOD). The substeps of this task include: push the shoe on the higher rack, tap the shoe on the higher rack, transfer the gripper to the lower level, and tap the shoe on the lower rack.
- *Bimanual Shoe Rack Organization (SO-Bi).* One pair of shoes is randomly placed at the edge of the third level of the shoe rack. The robot must push one shoe inward and align it with the other. The substeps of this task include: push the shoe, and tap the shoe.
- *Unimanual Scooping (Scoop-Uni).* The robot performs unimanual manipulation using a litter shovel to scoop a 3D-printed cat litter from varying locations and poses within a litter box, and then dump it into a trash bin. This long-horizon task involves both tool use and deformable object manipulation. The task is decomposed into the following substeps: grasp the shovel, scoop the litter, tilt the shovel, dump the litter, and place the shovel back.
- *Bimanual Pouring (Pour-Bi).* The robot performs bimanual manipulation to pour a Ping Pong ball from one cup to another. This longer-horizon task requires the robot to accurately reach both cups, which are randomly placed within a rectangular area on a table, lift one cup, pour the ball into the other, and then place both cups back on the table. This task evaluates the coordination and precision of the robot's bimanual manipulation. The substeps of this task include: pick up both cups, pour the ball, and place both cups.

2) `Human2LocoMan` Embodiments: As shown in Table I, the unimanual and bimanual modes of `Human2LocoMan` represent distinct embodiments, each differing in morphology, observations, and action spaces. In practice, we install and utilize wrist cameras on the LocoMan robot for the three unimanual manipulation tasks.

3) Data collection: For each task, we collect various numbers of human and robot trajectories with the `Human2LocoMan` system. The details of the collected data are demonstrated in Table II. About 10% data of each task is used for validation.

TABLE II: Records of data collection for different tasks.

Task	# human traj.	human time (min)	# robot traj.	robot time (min)
TC-Uni	300	25	150	15
TC-Bi	315	22	70	7
SO-Uni	240	34	90	23
SO-Bi	200	20	92	12
Scoop-Uni	340	96	66	22
Pour-Bi	210	35	64	22

4) Training details.: For Toy Collection and Shoe Rack Organzation, we pretrain a model that utilizes the human data of both the unimanual and bimanual versions of the task, then we finetune the model on each unimanual or bimanual task with the corresponding robot data. For each task, we choose a set of training hyperparameters (e.g. batch size, chunk size) that are kept the same for all methods. (See Appendix Section VI-C.) We also list the model hyperparameters we use for our method and the baselines in the Appendix Section VI-A and VI-B.

5) Baselines: We compare `Human2LocoMan` to the following SOTA imitation learning baselines:

- *Humanoid Imitation Transformer (HIT):* HIT [20] is an imitation learning framework designed for humanoid skill learning that also extends to any robot embodiment. It builds upon ACT [18] and employs a decoder-only architecture that simultaneously predict the future action sequence and future image features. It discourages the vision-based policy to ignore the visual input and overfit on proprioceptive states by introducing a L2 image feature loss to the original behavioral cloning policy. HIT itself is not capable of handling data from different domains and embodiments, and we position HIT as a reference implementation that efficiently learns from in-domain robot demonstrations.
- *Heterogeneous Pretrained Transformer (HPT):* HPT [74] is a framework for learning from vast amounts of data collected from humans, teleoperation, simulation, and real-life robots. HPT also has a modularized design and consists of the stems, the trunk, and the head, where the stems and heads are similar to our tokenizers and detokenizers. The trunk is designed to capture the complex mapping between the input and output in a unified latent space through large-scale pretraining. The implementation of HPT differs from our framework in several key aspects. Firstly, we leverage the unified observation and

TABLE III: Result Summary. We report success rate (SR) ↑ in % and task score (TS) ↑ for each task.

			Toy Collection								Shoe Rack Organization								Scooping				Pouring			
			Unimanual				Bimanual				Unimanual				Bimanual				Unimanual				Bimanual			
			ID		OOD		ID		OOD		ID		OOD		ID		OOD		ID		OOD		ID		OOD	
Method	Pretrained	Data	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS	SR	TS
HIT	-	smaller	54.2	42	41.6	15	45.8	37	41.6	16	87.5	112	75.0	50	66.7	52	25.0	14	58.3	96	16.7	30	58.3	62	16.7	17
HIT	-	larger	79.2	57	58.3	23	58.3	47	58.3	21	79.2	107	**83.3**	**52**	**83.3**	**63**	33.3	15	66.7	106	33.3	34	70.8	72	8.33	7
MXT	N	smaller	70.8	56	33.3	20	66.7	54	41.7	15	87.5	109	16.7	10	66.7	52	33.3	14	62.5	105	16.7	30	75.0	75	33.3	24
MXT	N	larger	87.5	**67**	83.3	31	70.8	53	41.7	16	83.3	107	50.0	37	75.0	60	58.3	23	62.5	98	41.7	38	79.2	76	33.3	22
MXT	Y	smaller	91.7	66	83.3	30	83.3	62	83.3	31	83.3	103	75.0	47	79.2	61	58.3	24	**87.5**	**129**	25.0	35	83.3	83	58.3	33
MXT	Y	larger	**95.8**	**67**	**91.7**	**34**	**91.7**	**67**	**100**	**36**	**95.8**	**116**	**83.3**	**52**	**83.3**	**63**	**75.0**	**29**	**87.5**	**129**	**66.7**	**52**	**91.7**	**88**	**83.3**	**42**

* Number of trajectories: TC-Uni smaller=20, larger=40; TC-Bi smaller=30, larger=60; SO-Uni smaller=40, larger=80; SO-Bi smaller=40, larger=80; Scoop-Uni smaller=30, larger=60; Pour-Bi smaller=30, larger=60.

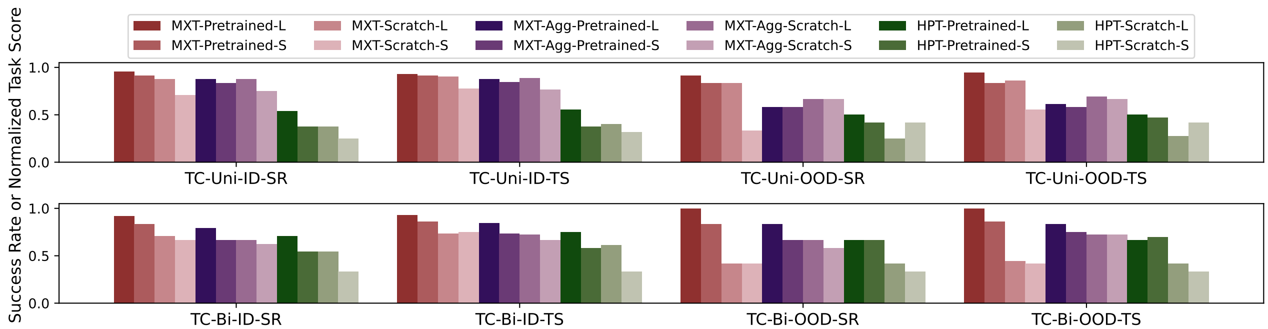

Fig. 5: Ablation study on unimanual and bimanual toy collection. We compare MXT, its ablation MXT-Agg, and baseline HPT on SR and TS. Here, "L" denotes the larger training set (40 trajectories for TC-Uni, 60 trajectories for TC-Bi), while "S" denotes the smaller training set (20 trajectories for TC-Uni, 30 trajectories for TC-Bi).

action frames to align data from different embodiments on the modality level, while HPT can only construct tokenizers for all image or proprioceptive data, and one detokenizer for all action dimensions. The ResNet image encoder in HPT is also frozen to achieve efficient learning with large models, while we opt to finetune the ResNet encoder along with the whole network end-to-end to better account for the visual gap between embodiments.

More implementation details of these baselines can be found in Appendix Section VI-B. For the HPT baseline, we train with several different settings: training with only LocoMan data, pretraining with our human data and finetuning on LocoMan data, and directly finetuning the released HPT checkpoints with LocoMan data. For the HIT baseline, we only train on LocoMan data, as it is unable to incorporate human data.

6) Evaluation Metrics: We present the evaluation results using three metrics: i) success rate (SR), ii) task score (TS), and iii) validation loss. To calculate the success rate and task score, we perform a fixed number of real world rollouts using the evaluated method for one task. The policy is rolled out for 24 times with in-distribution (ID) objects and 12 times with out-of-distribution (OOD) objects.

For each task, we define a set of critical substeps necessary to fully complete the task. When calculating the task score, successfully completing each intermediate substep earns one point, and reaching the final goal—i.e., completing the entire task—earns an additional point. The final task score is the sum of points across all rollouts for that task. The success rate of a method on a given task, under either the ID or OOD setting, is computed as the ratio of successful rollouts (i.e., rollouts where all substeps are completed) to the total number of rollouts performed.

In addition, we report the best validation loss as another metric for training performance. For all the included methods, we align how the loss is computed so that these losses can be meaningfully compared. Note that the validation loss is not a faithful indicator of the policy performance, but it does reflect how well the model is optimized, especially when there is a significant difference in the validation loss of different policies in the same setting. We mainly use this metric to analyze the training process of different architectures (MXT, HIT and HPT) and to provide a separate dimension to our evaluation.

B. Results and Analysis

(1) Does the `Human2LocoMan` system enable versatile quadrupedal manipulation capabilities?

Data collection. As shown in Table II, `Human2LocoMan` teleoperation enables the collection of a substantial amount of robot data (over 50 trajectories) within 30 minutes across all tasks. Using the `Human2LocoMan` human data collection system, over 200 trajectories can be gathered within the same time frame. Even for the most challenging task, a human can collect over 300 trajectories within one and a half hours. Notably, the robot's manipulation speed is comparable to, and in many tasks approaches, that of a human. These results highlight the data collection efficiency of our system.

Task versatility. As depicted in Figure 4, `Human2LocoMan`'s policy can perform tasks across a wide range of scenarios, including unimanual and bimanual manipulation, prehensile and non-prehensile manipulation, deformable object manipu-

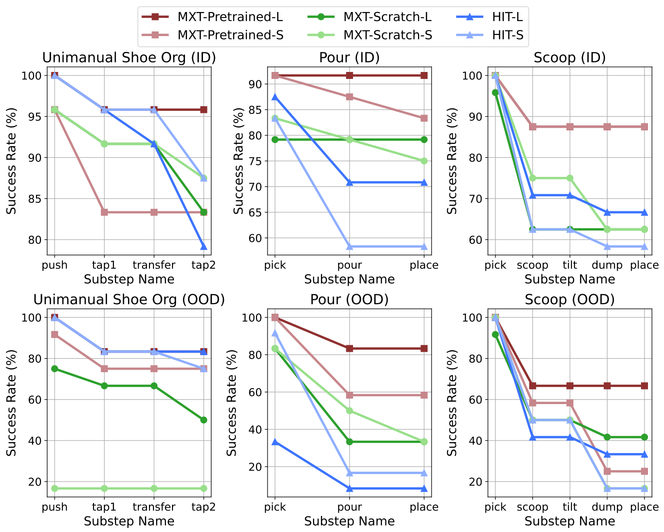

Fig. 6: Substep success rate. The success rate for some substep is calcuated as the percentage of trials where the robot successfully completed the substep. For each task, we calculate this with 24 ID rollouts and 12 OOD rollouts. **MXT-Pretrained**: MXT pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on the LocoMan data. **MXT-Scratch**: MXT trained only on the LocoMan data. "L" denotes the larger training set (80 trajectories for SO-Uni, 60 trajectories for Pour and Scoop), while "S" denotes the smaller training set (40 trajectories for SO-Uni, 30 trajectories for Pour and Scoop).

lation, and tool use, while also generalizing to OOD objects and conditions.

Task performance. We summarize the success rates and task scores of our method and HIT across all tasks in Table III. Human2LocoMan's MXT achieves strong performance on all tasks using a relatively small dataset. The baseline method also attains decent performance on most tasks. These results highlight the high quality of our collected data and demonstrate the effectiveness of Human2LocoMan's data collection and training pipeline.

(2) How does MXT compare to state-of-the-art imitation learning architectures?

Compared to HIT. As shown in Table III, in most evaluated tasks, spanning both unimanual and bimanual modes and across both ID and OOD inference scenarios, MXT without pretraining achieves comparable or superior performance relative to HIT. Moreover, pretrained MXT consistently outperforms the HIT baseline in terms of both success rate and task score. From Figure7, we find that MXT demonstrates lower validation loss compared to HIT on most tasks, indicating superior training convergence. The performance improvement is particularly evident in tasks with larger datasets, suggesting that MXT scales more effectively with increasing data availability. Notably, HIT achieves a significantly lower validation loss compared to the MXT variants, while attaining comparable performance in SR and TS metrics under both ID and OOD settings relative to the best MXT model. As shown in the substep success analysis in Figure 6, the primary failures of the lower-performing MXT models occur during the first two substeps, push" and tap1." One potential reason for this is that the unimanual shoe organization task exhibits relatively less variation in object locations and types compared to other tasks, which may favor HIT despite its lack of modular designs and pretraining.

Compared to HPT. From Figure 5, we observe that MXT consistently outperforms HPT in both SR and TS metrics across all combinations of pretraining and data sizes on the toy collection tasks. Validation loss results, shown in Figure 8, reveal a similar trend in the unimanual toy collection task across a broader range of dataset sizes. Notably, we observe severe overfitting in HPT experiments when training on our datasets, a phenomenon not observed in MXT. This further suggests that the modular design of the MXT architecture facilitates better generalization.

(3) How does human data collected by Human2LocoMan contribute to imitation learning performance?

Efficiency, robustness, and generalizability. As shown in Table III, pretraining on human data has a substantial positive impact on LocoMan manipulation performance. The policy maintains strong performance even when robot data is limited, highlighting both its efficiency and robustness. We hypothesize that MXT benefits from learning useful complementarities—i.e., positive transfer effects—between human demonstrations and LocoMan robot data. Specifically, comparing MXT-Pretrained to MXT-Scratch in Table III, we observe that pretraining improves performance on TC-Uni, TC-Bi, and Scooping tasks under ID settings, where objects exhibit diverse **locations**. MXT-Pretrained tends to produce smoother and more robust motions, enabling more accurate localization of target objects. For instance, as shown in Figure 6, MXT-Pretrained achieves substantially better scooping performance—which requires precise localization—compared to all other methods. Moreover, Table III reveals large performance gaps on OOD objects in tasks such as TC-Bi, SO-Uni, and Pouring, where OOD objects differ significantly from ID objects in **shape**, **texture**, and **color**. These results suggest that MXT, by leveraging human demonstrations during the pretraining stage, is able to generalize effectively to novel scenarios unseen during robot training.

Long-horizon performance. For a more detailed analysis on long-horizon tasks that require multiple manipulation steps, we present in Figure 6 how the success rate decays with each substep in tasks including SO-Uni, Pour-Bi and Scoop-Uni. MXT-Pretrained is shown to maintain a decent success rate as the long-horizon task progresses, while MXT-Scratch and HIT tend to fail more after the first substep, especially in Pouring and Scooping tasks. We note that the second substep in these tasks commonly involves moving and localizing an object with precision, and pretraining with human data appears to help with completing such challenging substeps. This suggests that human data incorporated during pretraining can promote manipulation accuracy, which is key to completing a sequential long-horizon task.

(4) Do the design choices in MXT facilitate positive transfer from Human to LocoMan?

Our framework presents positive cross-embodiment transfer despite substantial embodiment gaps. From Figure 8, we see the gap in validation loss between HPT-Pretrained and HPT-Scratch is not as much as for MXT. The HPT-Small and HPT-Base models also do not generalize as well as MXT-Pretrained. This highlights the ability of MXT to consume human data which has a large embodiment gap from the LocoMan data.

For more concrete comparisons, we present SR and TS results based on 36 trials, comprising 24 OOD and 12 ID trials, as shown in Figure 5. HPT performs consistently worse than MXT, both when finetuned and trained from scratch. We attribute part of this performance gap to HPT using frozen image encoders by default. We also provide additional ablations of MXT where we *aggregate* the input modalities, i.e. tokenize them with a single tokenizer, and decode actions with a single detokenizer; this baseline (marked with "Agg" in Figure 5(b)) incorporates the key HPT designs including cross attention tokenization of visual and proprioceptive inputs and trunk weight sharing, while finetuning the vision encoders and remaining architecturally comparable to MXT. MXT consistently benefits from pretraining and outperforms this baseline when both are finetuned, highlighting the advantage of modularized tokenization for leveraging human data.

Notably, MXT-Agg sometimes transfers suboptimally with respect to HPT, as evidenced by little to no improvement when finetuning the pretrained model compared with training from scratch. This is likely due to increased representation power in the tokenizer, which permits more overfitting in the transformer trunk and could negatively impact the trunk transferability. However, with the incorporation of our modular design, MXT is trained with additional regularization and exhibits improved transferability. The modular design effectively aids in the trade-off between more network representation power and better transferability in our framework, and allows attaining both qualities.

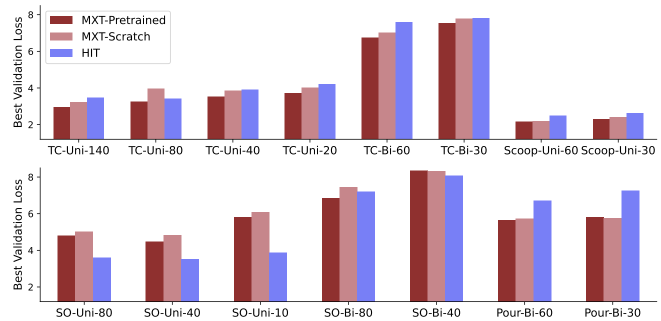

Fig. 7: Best validation loss of our method and HIT on all our tasks. **MXT-Pretrained**: MXT pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on the LocoMan data. **MXT-Scratch**: MXT trained only on the LocoMan data. The number suffix denotes the number of demonstrations in the LocoMan training set.

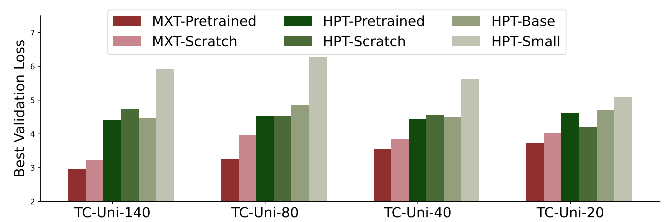

Fig. 8: Best validation loss of our method and HPT on the unimanual Toy Collection task. **MXT-Pretrained**: MXT pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on the LocoMan data. **MXT-Scratch**: MXT trained only on the LocoMan data. **HPT-Pretrained**: HPT trunk pretrained on our human data, then finetuned on the LocoMan data. **HPT-Scratch**: HPT network trained only on the LocoMan data. **HPT-Base**: Finetune with our LocoMan data with HPT trunk initialized with released `HPT-Base` weights. **HPT-Small**: Finetune with our LocoMan data with HPT trunk initialized with released `HPT-Small` weights.

V. Limitations

While our system introduces a novel approach to cross-embodiment manipulation and efficient data collection for quadrupedal robots, it has several limitations. First, the teleoperation system still requires some practice for human operators to achieve precise manipulations and may feel unintuitive in certain aspects, such as controlling torso movements via head motions. Second, although we envision our system enabling large-scale cross-embodiment learning, in this work we have not yet scaled it to other robotic platforms or incorporated additional robotic datasets. As future work, we plan to validate its scalability and robustness across different robot types, including robotic arms and humanoids.

VI. Conclusion

In this paper, we introduce `Human2LocoMan`, a unified framework for flexible data collection and cross-embodiment learning to enable versatile quadrupedal manipulation skills on the open-source LocoMan platform. Our teleoperation and human data collection systems allow efficient acquisition of large-scale, high-quality datasets by bridging the action spaces between human and robot embodiments. Built upon this foundation, we propose Modularized Cross-embodiment Transformer, a modular policy architecture that supports positive transfer from human demonstrations to robot policies. Through extensive experiments on six challenging household tasks, we demonstrate that `Human2LocoMan` enables strong performance, efficient training, and robust generalization to out-of-distribution scenarios, outperforming strong imitation learning baselines. Our results highlight the effectiveness of cross-embodiment learning and modularized policy design in advancing scalable, versatile quadrupedal manipulation.

References

[1] Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. Dtc: Deep tracking control. *Science Robotics*, 9(86):eadh5401, 2024.

[2] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[3] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5 (47):eabc5986, 2020.

[4] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *CVPR 2023*, 2023.

[5] Yuxiang Yang, Guanya Shi, Changyi Lin, Xiangyun Meng, Rosario Scalise, Mateo Guaman Castro, Wenhao Yu, Tingnan Zhang, Ding Zhao, Jie Tan, et al. Agile continuous jumping in discontinuous terrains. *arXiv preprint arXiv:2409.10923*, 2024.

[6] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[7] Björn Lindqvist, Samuel Karlsson, Anton Koval, Ilias Tevetzidis, Jakub Haluška, Christoforos Kanellakis, Ali-akbar Agha-mohammadi, and George Nikolakopoulos. Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue. *Robotics and Autonomous Systems*, 2022.

[8] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[9] Qi Wu, Zipeng Fu, Xuxin Cheng, Xiaolong Wang, and Chelsea Finn. Helpful doggybot: Open-world object fetching using legged robots and vision-language models. *arXiv preprint arXiv:2410.00231*, 2024.

[10] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.

[11] Mingyo Seo, H Andy Park, Shenli Yuan, Yuke Zhu, and Luis Sentis. Legato: Cross-embodiment imitation using a grasping tool. *arXiv preprint arXiv:2411.03682*, 2024.

[12] Xialin He, Chengjing Yuan, Wenxuan Zhou, Ruihan Yang, David Held, and Xiaolong Wang. Visual manipulation with legs. *arXiv preprint arXiv:2410.11345*, 2024.

[13] Ri-Zhao Qiu, Yuchen Song, Xuanbin Peng, Sai Aneesh Suryadevara, Ge Yang, Minghuan Liu, Mazeyu Ji, Chengzhe Jia, Ruihan Yang, Xueyan Zou, et al. Wildlma: Long horizon loco-manipulation in the wild. *arXiv preprint arXiv:2411.15131*, 2024.

[14] Changyi Lin, Xingyu Liu, Yuxiang Yang, Yaru Niu, Wenhao Yu, Tingnan Zhang, Jie Tan, Byron Boots, and Ding Zhao. Locoman: Advancing versatile quadrupedal dexterity with lightweight loco-manipulators. *arXiv preprint arXiv:2403.18197*, 2024.

[15] Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

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

[17] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

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

[19] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.

[20] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[21] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[22] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[23] Shiqi Yang, Minghuan Liu, Yuzhe Qin, Runyu Ding, Jialong Li, Xuxin Cheng, Ruihan Yang, Sha Yi, and Xiaolong Wang. Ace: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation. *arXiv preprint arXiv:2408.11805*, 2024.

[24] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4243–4250. IEEE, 2018.

[25] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

[26] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2:

Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[27] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[28] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911. IEEE, 2024.

[29] Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training. *arXiv preprint arXiv:2407.18911*, 2024.

[30] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C Karen Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. *arXiv preprint arXiv:2410.08464*, 2024.

[31] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. *arXiv preprint arXiv:2410.24221*, 2024.

[32] Yandong Ji, Gabriel B Margolis, and Pulkit Agrawal. Dribblebot: Dynamic legged manipulation in the wild. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5155–5162. IEEE, 2023.

[33] Fan Shi, Timon Homberger, Joonho Lee, Takahiro Miki, Moju Zhao, Farbod Farshidian, Kei Okada, Masayuki Inaba, and Marco Hutter. Circus anymal: A quadruped learning dexterous manipulation with its limbs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2316–2323. IEEE, 2021.

[34] Xuxin Cheng, Ashish Kumar, and Deepak Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5106–5112. IEEE, 2023.

[35] Mike Zhang, Yuntao Ma, Takahiro Miki, and Marco Hutter. Learning to open and traverse doors with a legged manipulator. *arXiv preprint arXiv:2409.04882*, 2024.

[36] Zhengmao He, Kun Lei, Yanjie Ze, Koushil Sreenath, Zhongyu Li, and Huazhe Xu. Learning visual quadrupedal loco-manipulation from demonstrations. *arXiv preprint arXiv:2403.20328*, 2024.

[37] Tianle Huang, Nitish Sontakke, K Niranjan Kumar, Irfan Essa, Stefanos Nikolaidis, Dennis W Hong, and Sehoon Ha. Bayrntune: Adaptive bayesian domain randomization via strategic fine-tuning. *arXiv preprint arXiv:2310.10606*, 2023.

[38] Seunghun Jeon, Moonkyu Jung, Suyoung Choi, Beomjoon Kim, and Jemin Hwangbo. Learning whole-body manipulation for quadrupedal robot. *IEEE Robotics and Automation Letters*, 9(1):699–706, 2023.

[39] Jonas Stolle, Philip Arm, Mayank Mittal, and Marco Hutter. Perceptive pedipulation with local obstacle avoidance. *arXiv preprint arXiv:2409.07195*, 2024.

[40] Ri-Zhao Qiu, Yafei Hu, Ge Yang, Yuchen Song, Yang Fu, Jianglong Ye, Jiteng Mu, Ruihan Yang, Nikolay Atanasov, Sebastian Scherer, et al. Learning generalizable feature fields for mobile manipulation. *arXiv preprint arXiv:2403.07563*, 2024.

[41] K Niranjan Kumar, Irfan Essa, and Sehoon Ha. Cascaded compositional residual learning for complex interactive behaviors. *IEEE Robotics and Automation Letters*, 8(8): 4601–4608, 2023.

[42] Yuming Feng, Chuye Hong, Yaru Niu, Shiqi Liu, Yuxiang Yang, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Learning multi-agent loco-manipulation for long-horizon quadrupedal pushing. *arXiv preprint arXiv:2411.07104*, 2024.

[43] Ziyan Xiong, Bo Chen, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Yang Gao. Mqe: Unleashing the power of interaction with multi-agent quadruped environment. *arXiv preprint arXiv:2403.16015*, 2024.

[44] Tianxu An, Joonho Lee, Marko Bjelonic, Flavio De Vincenti, and Marco Hutter. Solving multi-entity robotic problems using permutation invariant neural networks. *arXiv preprint arXiv:2402.18345*, 2024.

[45] Ofir Nachum, Michael Ahn, Hugo Ponte, Shixiang Gu, and Vikash Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. *arXiv preprint arXiv:1908.05224*, 2019.

[46] Yandong Ji, Bike Zhang, and Koushil Sreenath. Reinforcement learning for collaborative quadrupedal manipulation of a payload over challenging terrain. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 899–904. IEEE, 2021.

[47] Guoping Pan, Qingwei Ben, Zhecheng Yuan, Guangqi Jiang, Yandong Ji, Jiangmiao Pang, Houde Liu, and Huazhe Xu. Roboduet: A framework affording mobile-manipulation and cross-embodiment. *arXiv preprint arXiv:2403.17367*, 2024.

[48] Minghuan Liu, Zixuan Chen, Xuxin Cheng, Yandong Ji, Ruihan Yang, and Xiaolong Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2402.16796*, 2024.

[49] Jiazhao Zhang, Nandiraju Gireesh, Jilong Wang, Xiaomeng Fang, Chaoyi Xu, Weiguang Chen, Liu Dai, and He Wang. Gamma: Graspability-aware mobile manipulation policy learning based on online grasping pose fusion. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1399–1405. IEEE, 2024.

[50] Philip Arm, Mayank Mittal, Hendrik Kolvenbach, and Marco Hutter. Pedipulate: Enabling manipulation skills using a quadruped robot's leg. In *41st IEEE Conference on Robotics and Automation (ICRA 2024)*, 2024.

[51] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57

(5):469–483, 2009.

[52] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50 (2):1–35, 2017.

[53] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3(1):297–330, 2020.

[54] Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[55] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-thewild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[56] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[57] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[58] Xingyu Lin, John So, Sashwat Mahalingam, Fangchen Liu, and Pieter Abbeel. Spawnnet: Learning generalizable visuomotor skills from pre-trained network. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4781–4787. IEEE, 2024.

[59] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, page 02783649241273901, 2022.

[60] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[61] Zhijun Zhang, Yaru Niu, Ziyi Yan, and Shuyang Lin. Real-time whole-body imitation by humanoid robots and task-oriented teleoperation using an analytical mapping method and quantitative evaluation. *Applied Sciences*, 8 (10):2005, 2018.

[62] Louise Penna Poubel, Sophie Sakka, Denis Ćehajić, and Denis Creusot. Support changes during online human motion imitation by a humanoid robot using task specification. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1782–1787. IEEE, 2014.

[63] Bipasha Sen, Michelle Wang, Nandini Thakur, Aditya Agarwal, and Pulkit Agrawal. Learning to look around: Enhancing teleoperation and learning with a human-like actuated neck. *arXiv preprint arXiv:2411.00704*, 2024.

[64] Chenhao Lu, Xuxin Cheng, Jialong Li, Shiqi Yang, Mazeyu Ji, Chengjing Yuan, Ge Yang, Sha Yi, and Xiaolong Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024.

[65] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

[66] Hongjie Fang, Hao-Shu Fang, Yiming Wang, Jieji Ren, Jingjing Chen, Ruo Zhang, Weiming Wang, and Cewu Lu. Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15031–15038. IEEE, 2024.

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

[68] Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+ x: Retrieval and execution from everyday human videos. *arXiv preprint arXiv:2407.12957*, 2024.

[69] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[70] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[71] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[72] Lawrence Yunliang Chen, Kush Hari, Karthik Dharmarajan, Chenfeng Xu, Quan Vuong, and Ken Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting. *arXiv preprint arXiv:2402.19249*, 2024.

[73] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint arXiv:2408.11812*, 2024.

[74] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *arXiv preprint arXiv:2409.20537*, 2024.

[75] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 614–619. IEEE, 2019.

APPENDIX

A. Implementation and Training details of MXT

Training Details. We list the training optimizer and the transformer trunk hyperparamters in Table IV. These hyperparameters are kept the same for all our experiments.

TABLE IV: MXT trunk and training hyperparameters

Hyperparameters	Value
optimizer	AdamW
learning rate	5e-5 (finetuning/from scratch) 1e-4 (pretraining)
scheduler	constant
weight decay	1e-4
trunk encoder layers	4
trunk decoder layers	4
hidden dim	128
transformer feedforward dim	256
#attention heads	16

Cross Attention in Tokenizers and Detokenizers. In the tokenizers of MXT, we use a simple cross attention mechanism to transform the input feature of indefinite length into a fixed number of tokens. For the attention layer in all tokenizers, the hidden dim is 128, the number of attention heads is 4, each with a head dimension of 32, and the dropout rate is 0.1. Other hyperparameters of each tokenizer are shown in Table V.

Similarly, we also use cross attention to decode the action modalities in detokenizers from a fixed number of output transformer tokens. For the attention layer, the number of attention heads is 4, each with a head dimension of 16, and the dropout rate is 0.1. Other hyperparameters of each detokenizer are shown in Table VI

TABLE V: MXT tokenizer hyperparameters

Modality	Input dimensions	#tokens	MLP widths
main image	(3, 480 1280)	16	N/A
wrist image	(3, 480, 640)	8	
body pose	(6,)	4	[128, 128]
EEF pose	(12,)	4	
EEF to body pose	(12,)	4	
gripper angles	(2,)	4	

TABLE VI: MXT detokenizer hyperparameters

Modalities	Output dimensions	#tokens
body pose	(6,)	6
EEF pose	(12,)	6
gripper angle	(2,)	6

Masks for aligning embodiment modalities. We mentioned that masks are needed to exclude redundant dimensions or modalities that are not present in some embodiment, and here we give a more detailed description of our implemented masks.

a) Masks on images. We recognize that some image view are not available for all embodiments and tasks. In our current framework, we assume there are at most two camera views (or image modalities): the main camera and the wrist camera. However, this can be easily extended within our framework to cater to any number of camera views. When one of these camera views is not present, we directly mark this in the transformer mask of the trunk and fill in dummy tokens in the corresponding positions, so that the positions associated with this image modality will not be attended on.

b) Masks on proprioceptive states. In some cases, the proprioceptive states may have some or all dimensions that should not be considered for the task. For example, in single-arm tasks, the poses of the left end effector, or the last half of the end effector pose modality, will not be considered, and in bimanual tasks where the LocoMan body is upright, the body pose is fixed and therefore redundant in the observations. When part of a proprioception modality are redundant dimensions, we apply zero padding on these dimension and perform encoding through the tokenizer as usual. Different from how we treated masked image modalities, this has no effect on the transformer mask of the trunk. When an entire proprioception modality should be disregarded, however, we handle this modality in a similar to the image modalities and apply the transformer mask accordingly.

Data Normalization. For both human and LocoMan data, we apply data normalization on observations and action labels. For non-image data, we estimate the per-dimension mean the standard deviation from the dataset, and normalize the data with the usual approach:

$$\bar{x}_t = \frac{x_t - \text{mean}}{\text{std}}.$$

For image data, the mean and standard deviation are set as the ImageNet statistics for the RGB channels: mean = [0.485, 0.456, 0.406], and std = [0.229, 0.224, 0.225]. The images are normalized in the same way with these parameters.

Dropout in Pretraining. We discover that increasing the dropout in transformer trunk can improve the finetuning performance for MXT. In practice, we find that setting the pretraining dropout rate to 0.5 for scooping and 0.4 for all other tasks yield reasonably good performance. When training with LocoMan data, including training from scratch and finetuning, the transformer trunk dropout rate is reverted to 0.1.

TABLE VII: HIT hyperparameters

Hyperparameters	Value
optimizer	AdamW
learning rate	2e-5
scheduler	constant
weight decay	1e-4
encoder layers	4
decoder layers	4
hidden dim	128
#attention heads	8
feature loss weight	0.001
image backbone	ResNet18

B. Implementation details of baselines

HIT. Our implementation of Humanoid Imitation Transformer [20] is based on the released codebase, with only minor

TABLE VIII: HPT hyperparameters

Hyperparameters	Value
optimizer	AdamW
learning rate	5e-5 (finetuning/from scratch) 1e-4(pretraining)
scheduler	constant
weight decay	1e-4
trunk	
#transformer blocks	16
hidden dim	128
feedforward dim	256
#attention heads	8
action head	
#attention heads	8
head dim	64
dropout	0.1
output dim	20
image stem	
encoder	ResNet18
MLP widths	[128]
#tokens	16
state stem	
MLP widths	[128]
#tokens	16

modifications to accommodate our data format. The hyperparameters used for training are summarized in Table VII.

HPT. We follow the original implementation of HPT [74], with the main exception that we changed the data normalization method so as to align with the approach of other frameworks and to have a fair comparison of the validation loss. The hyperparameters we used when training HPT are summarized in Table VIII.

C. Global task-specific training parameters

We choose a set of training parameters for each specific task, and we keep these settings aligned across all methods as listed in Table IX.

TABLE IX: Global training parameters for each task

Task	Mode	Batch Size	Training Steps	Chunk Size
Toy Collection	Unimanual	16	60000	60
	Bimanual	16	60000	60
Shoe Organization	Unimanual	24	80000	180
	Bimanual	24	100000	120
Scooping	Unimanual	24	100000	120
Pouring	Bimanual	24	80000	180