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

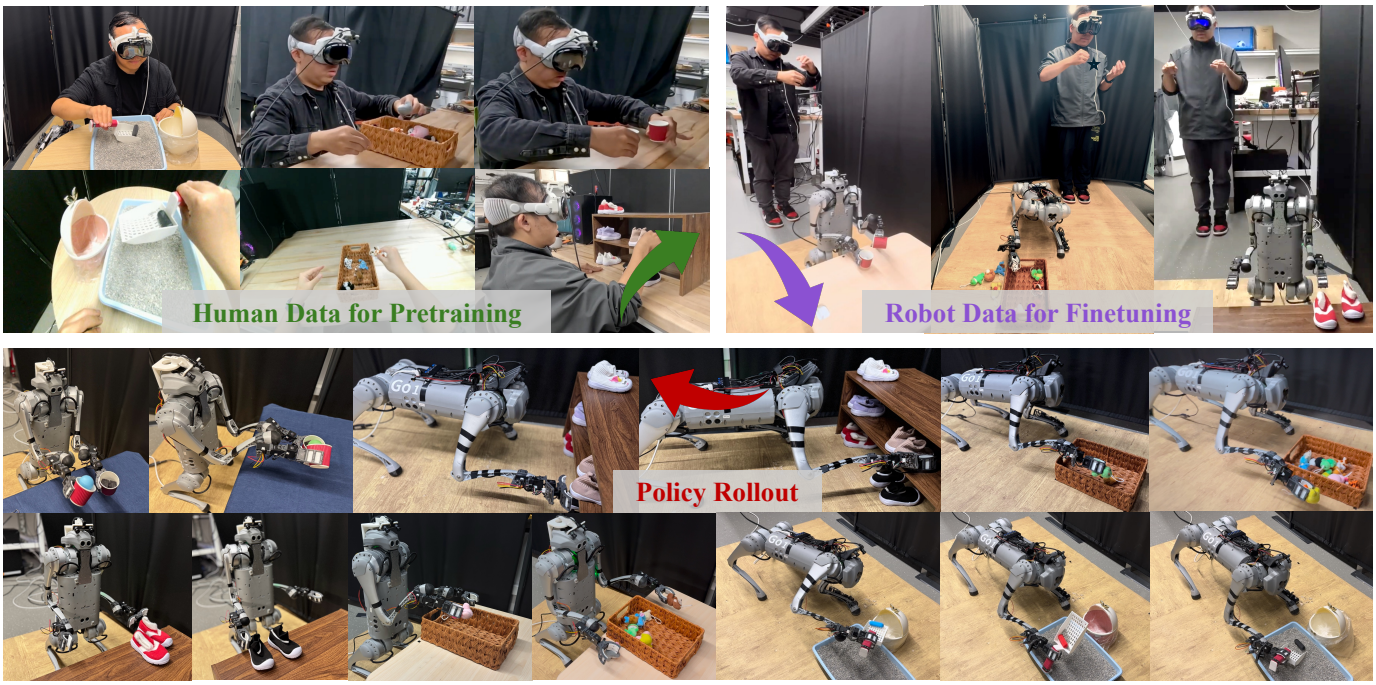

Fig. 1: **Human2LocoMan** provides a unified framework for collecting human demonstrations and teleoperated robot whole-body motions, along with cross-embodiment policy learning for quadrupedal manipulation. Human data is used for model pretraining, while robot data is leveraged for policy finetuning. Human2LocoMan achieves positive transfer from human to quadrupedal embodiments, facilitating versatile manipulation skills for unimanual and bimanual, non-prehensile and prehensile, precise tool-use, and long-horizon tasks.

***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 cross-embodiment imitation learning system for quadrupedal manipulation, leveraging data collected from both humans and LocoMan, a quadruped equipped with multiple manipulation modes. Specifically, we develop a teleoperation and data collection pipeline, 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 modularized architecture that supports co-training and pretraining on structured modality-aligned 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. We validate our system on six real-world manipulation tasks, where it achieves an average success rate improvement of 41.9% overall and 79.7% under out-of-distribution (OOD) settings compared to the baseline. Pretraining with human data contributes a 38.6% success rate improvement overall and 82.7% under OOD settings, enabling consistently better performance with only half the amount of robot data. Our code, hardware, and data 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. 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 egocentric data by treating humans as an alternative embodiment [31, 32]. However, the effectiveness of human data for manipulation tasks involving non-traditional embodiments such as quadrupeds has yet to be demonstrated. From another perspective, effective teleoperation or human demonstrations typically require a kinematically similar control system—either biological [21, 31, 32] or mechanical [18, 19, 27, 33, 34]—to the target robot, or a specialized end effector [10, 11, 35] at the end of the human operator. However, the substantial embodiment gap between humans and quadrupedal robots poses challenges to both data collection and policy transfer.

To address these challenges, and drawing inspiration from the LocoMan platform [14]—a quadrupedal robot equipped with two leg-mounted loco-manipulators that offers a versatile foundation for learning manipulation skills across multiple operating modes—we propose `Human2LocoMan`, a unified framework for quadrupedal manipulation learning using data collected through human teleoperation and demonstrations. Specifically, to enable scalable data collection, our system leverages an extended reality (XR) headset to capture human motions while streaming a first-person view (during human data collection) or a first-robot view (during teleoperation) to the operator. For human data collection, the operator simply wears the XR headset and performs tasks naturally. During teleoperation, in addition to mapping human hand motions to the robot's grippers, we also map head motions to the robot's torso, expanding the robot's workspace and enhancing its active sensing capabilities. The resulting target poses are passed to a whole-body controller to generate coordinated robot motions. To structure the data and bridge the embodiment gap, we align motions of the human and the quadruped within a shared unified coordinate frame.

Different from the works that use egocentric human data to pretrain vision encoders [29, 36], learn interaction plan prediction [28], or co-train models with data from robots that share similar kinematics with humans [31, 32], we treat the human as a distinct embodiment from the target robot and leverage human data for model pretraining. 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. To facilitate cross-embodiment learning and preserve modality-specific distributions unique to each embodiment, we design a modular Transformer architecture, ***M**odularized **C**ross-embodiment **T**ransformer* (MXT), which shares a common Transformer trunk across embodiments while maintaining embodiment-specific tokenizers and detokenizers for shared modalities. The MXT policy is first pretrained on human data and subsequently finetuned with a small amount of robot data. A single pretrained model can be adapted to different robot embodiments through finetuning. Notably, our approach is orthogonal to prior work that leverages egocentric human videos to pretrain visual encoders or co-trains models with target robot data. Our architecture is compatible with any pretrained visual encoder and supports co-training with multi-embodiment data during both the pretraining and finetuning stages. We evaluate our approach on six household tasks across both unimanual and bimanual manipulation modes, achieving a 41.9% average improvement and 79.7% under OOD settings over the baseline. We also find that pretraining with human data boosts success by 38.6% overall and 82.7% under OOD scenarios, demonstrating effective positive transfer from humans to quadrupedal embodiments despite the large embodiment gap. These results highlight the effectiveness of our system for learning versatile quadrupedal manipulation skills and underscore its potential for scalable, large-scale cross-embodiment learning.

In summary, our paper provides the following contributions:

- We propose `Human2LocoMan`, a framework that enables flexible and scalable collection of human demonstrations and teleoperated robot trajectories for learning versatile quadrupedal manipulation skills.
- We design MXT, a modularized 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.) [37, 38, 39, 40, 41, 42, 43, 44], using a back-mounted arm for tabletop tasks [8, 45], or using leg-mounted manipulators for spatially-constrained (e.g., reaching toys underneath furniture) or bimanual 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 [37, 38, 40, 41, 42, 43, 44, 46, 47, 48, 49, 50, 51] and for training whole-body controllers to track end-effector poses for uni-manual grasping [8, 9, 10, 52, 53, 54, 55]; 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, 56, 57, 58] and thus provides an alternative approach to efficiently acquiring more general manipulation skills [26, 59, 60, 61, 62]. 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.

Data Collection for Imitation Learning: Various methods have been utilized to collect data for imitation learning. Joysticks and spacemouses [16, 63, 64] 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, 65, 66, 67]. 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, 68, 69, 70]. 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, 71]. 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, 35], though these are limited to a specific robot or end-effector type. Other works enable more ergonomic data collection with body-worn cameras [27, 72] 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 robot embodiments [73, 74, 75, 76, 77, 78, 79], where the heterogeneity and gaps in kinematics, vision, and proprioception have to be handled. For example, CrossFormer processes variable observation inputs by tokenizing images and proprioceptive information, predicts variable action outputs using action readout tokens, and conditions on language instructions or goal images [77]. HPT provides a reusable trunk for cross-embodiment learning and encodes observations into a fixed number of tokens, explicitly balancing image and proprioception [78]. In our work, we propose Modularized Cross-embodiment Transformer (MXT), which adopts a modular design for both tokenization and detokenization, and further enhances modularity by identifying fine-grained alignments of data modalities across embodiments.

While these works primarily rely on robot data, human videos and demonstrations offer potentials for scaling up cross-embodiment learning. Recent robotic foundation models [80, 81, 82] leverage large-scale Internet data, including egocentric human videos, to pretrain higher-level models that enhance reasoning and semantic understanding. However, effectively utilizing such data for robot motor policy learning remains a significant challenge. Notably, EgoMimic [31] treats humans as another embodiment and demonstrates strong positive transfer by co-training on both human and robot data. To enable such transfer, EgoMimic narrows the kinematic gap by selecting a human-like robot, reduces the proprioceptive gap through action normalization and alignment, and mitigates the appearance gap via visual masking. Our concurrent works similarly leverage human data for co-training on target robots that share similar embodiments with humans or the human hand, and apply action normalization to bridge the embodiment gap [32, 34]. In comparison, `Human2LocoMan` does not require observation and action normalization to align human data with a specific embodiment. Instead, it structures data into distinct modalities during collection and explicitly accounts for distributional gaps on these modalities across embodiments during training. This design enables greater flexibility and scalability for cross-embodiment learning, allowing us to achieve positive transfer from humans to multiple quadrupedal embodiments without requiring explicit data processing for domain alignment.

III. Methodology

In this section, we present the design and implementation of our system, `Human2LocoMan`, which integrates teleoperation, data collection, and a Transformer-based 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

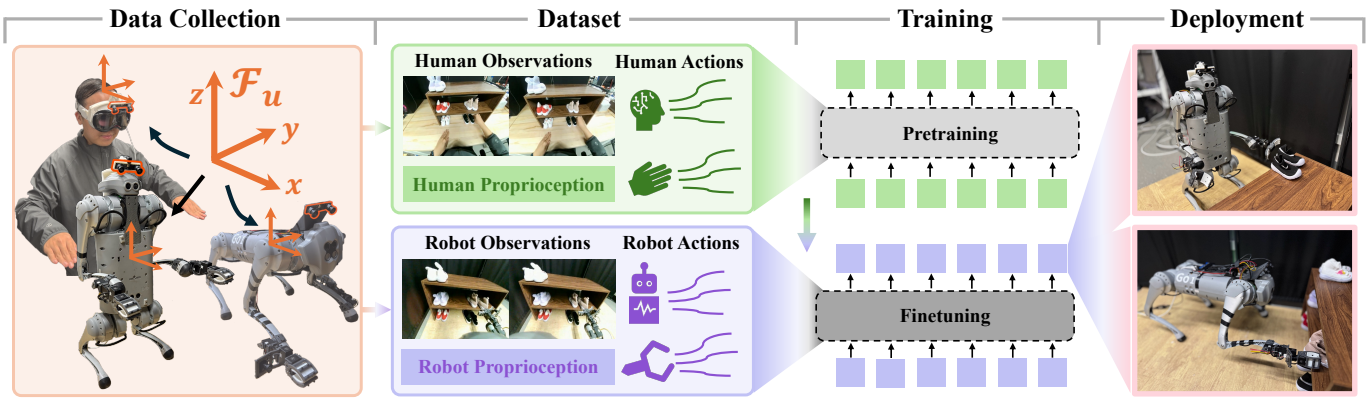

Fig. 2: **Human2LocoMan framework.** Our system uses an XR headset for data collection, capturing egocentric human data and teleoperated robot data, all mapped to a unified coordinate frame. The dataset consists of aligned vision, proprioception, and actions from the human and the robot. We adopt a two-stage training process: the modularized cross-embodiment model is first pretrained on easy-to-collect human data, and then finetuned on a small amount of robot data. The resulting Human2LocoMan policies can be deployed on real robots for versatile manipulation tasks in both unimanual and bimanual modes.

(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 ***Modularized Cross-embodiment Transformer*** (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 $\boldsymbol{x}_{\text{vr}}$ denotes the translation and $\boldsymbol{R}_{\text{vr}}$ denotes the rotation in the VR-defined world frame. Then, the 6D poses can be transformed into the unified frame $\mathcal{F}_u$ $(\boldsymbol{x}_{\text{uni}}, \boldsymbol{R}_{\text{uni}}) = (\boldsymbol{R}_{\text{uni}}^{\text{vr}}\boldsymbol{x}_{\text{vr}}, \boldsymbol{R}_{\text{uni}}^{\text{vr}}\boldsymbol{R}_{\text{vr}})$, 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 actions. The human operator starts to teleoperate the robot after an 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 [83] 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}_{t-1} instead of p^{t}_t. To mitigate rapid movements, we apply linear interpolation between $x^{\text{torso,t}}_{\text{uni},t}$ and $x^{\text{torso,t}}_{\text{uni},t-1}$, $x^{\text{r-eef,t}}_{\text{uni},t}$ and $x^{\text{r-eef,t}}_{\text{uni},t-1}$, $x^{\text{l-eef,t}}_{\text{uni},t}$ and $x^{\text{l-eef,t}}_{\text{uni},t-1}$, as well as $\theta^{\text{gripper,t}}_t$ and $\theta^{\text{gripper,t}}_{t-1}$. Additionally, quaternion interpolation is applied between $\boldsymbol{R}^{\text{torso,t}}_{\text{uni},t}$ and $\boldsymbol{R}^{\text{torso,t}}_{\text{uni},t-1}$, $\boldsymbol{R}^{\text{r-eef,t}}_{\text{uni},t}$ and $\boldsymbol{R}^{\text{r-eef,t}}_{\text{uni},t-1}$, and $\boldsymbol{R}^{\text{l-eef,t}}_{\text{uni},t}$ and $\boldsymbol{R}^{\text{l-eef,t}}_{\text{uni},t-1}$ to smooth large action variations.

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

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

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

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

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

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, while accounting for modality-specific distributions unique to each embodiment. To this end, we propose a modularized design called ***M**odularized **Cross**-embodiment **T**ransformer* (MXT). As illustrated in Figure 3, MXT consists mainly of three groups of modules: tokenizers, Transformer trunk, and detokenizers. The tokenizers act as encoders and map embodiment-specific observation modalities to tokens in the latent space, and the detokenizers translate the output tokens from the trunk to action modalities 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.

Tokenizers. The tokenizers T transform raw observations into tokens for the Transformer trunk. Similar to the design in [78], we use a cross-attention layer to format observational features into a fixed number of tokens for each modality. 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, mapping the output tokens from the trunk to the action modalities in each embodiment's action space. To reduce compounding errors and lower inference frequency, we adopt action chunking [18], where the detokenizers predict a sequence of h actions at each inference step. Within each detokenizer, a cross-attention layer is used to transform the latent action tokens—produced by the trunk at fixed positions—into a sequence of h actions matching the dimensions of the corresponding action modality.

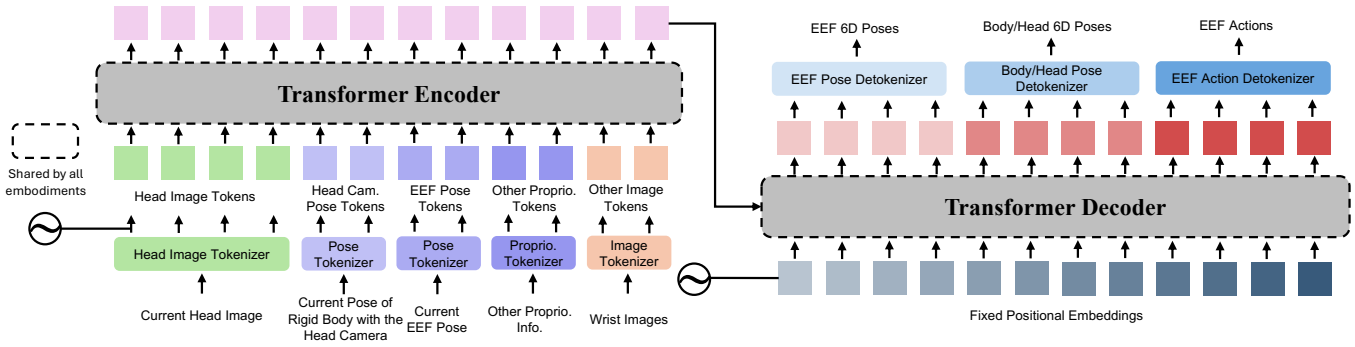

Fig. 3: **Modularized Cross-embodiment Transformer (MXT) architecture.** The inputs are organized as a list of modalities and encoded each by a separate tokenizer into a fixed number of tokens. The Transformer trunk handles decision making by consuming the concatenated encoded tokens and producing a 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.

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 observation or action modality is fixed by design. By sharing the trunk weights across human and robot embodiments, the model learns to process and generate tokens of different modalities and capture common decision-making patterns across 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 preserves modality-specific distributions unique to each embodiment and enables the model to explicitly account for distributional gaps across embodiments, which 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 the main camera and the wrist camera views . 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 actions, 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 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.

D. Training Paradigm

The details of our MXT training pipeline are outlined in Algorithm 1. For a given task, we first pretrain the model using the human dataset, followed by finetuning with the corresponding LocoMan dataset. During finetuning, only the Transformer trunk weights are initialized from the pretrained checkpoint. For tasks that share similar semantics but differ in manipulation modes (representing distinct embodiments in Table I), we jointly pretrain the model on human datasets across these tasks with different manipulation modes and then finetune it on each task using the corresponding LocoMan robot 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

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	✓	×	✓	✓	✓	×	✓	✓	✓	✓

optimize the following batched loss for each training batch $B_e = \{(\boldsymbol{o}_j, A_j)\}_{j=1}^n$ as a proxy of $\mathcal{L}_{e,m_i}(\theta)$:

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

where $\boldsymbol{a}_{j,l}\left[m_i\right] = \left(A_j\right)_l\left[m_i\right]$ 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.

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 D_human
    Compute L_human(B) = Σ_i L_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 D_LocoMan
    Compute L_LocoMan(B) = Σ_i L_LocoMan,m_i(B) with Eq.6
    Optimize the policy weights θ with backpropagation
return π
```

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 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

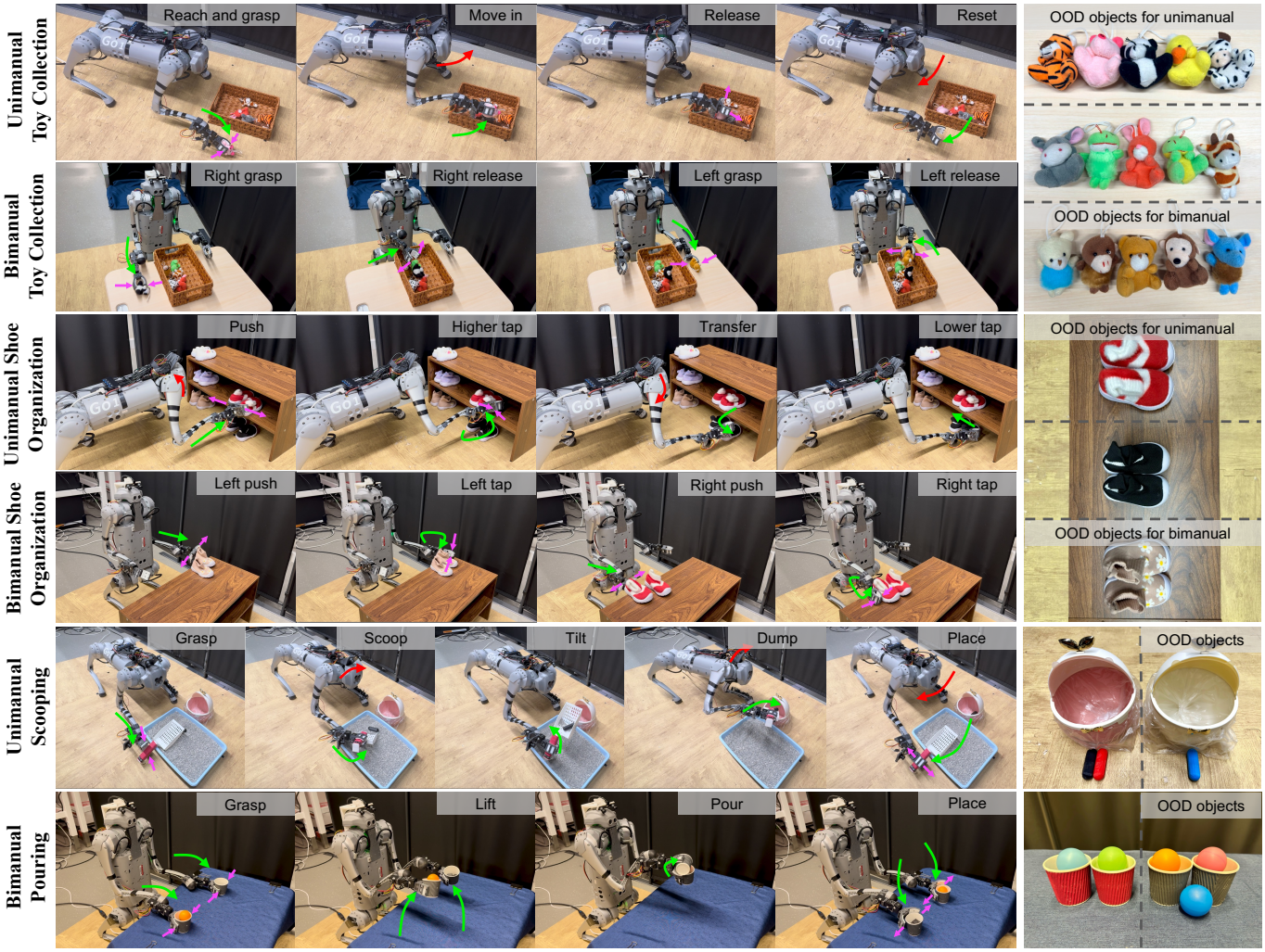

Fig. 4: Rollouts of the MXT policy and the objects used across manipulation tasks in our experiments. Green arrows indicate end-effector motions, red arrows denote torso movements, and pink arrows represent gripper actions. Both **unimanual** and **bimanual toy collection** tasks assess the robot's ability 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.

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 III. About 10% data of each task is used for validation.

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 C.) We also list the model hyperparameters we use for our method and the baselines in the Appendix Section A and B.

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

TABLE II: Result Summary. We report success rate (SR) ↑ in % and task score (TS) ↑ for each task. We highlight the best performance in **bold** and the second best in underline. ID results are based on 24 trials, and OOD results on 12 trials.

			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.

TABLE III: 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

- *Humanoid Imitation Transformer (HIT):* HIT [20] is an imitation learning framework originally developed for humanoid skill learning, with the capability to generalize to other robot embodiments. It builds on ACT [18] and employs a decoder-only architecture that simultaneously predicts future action sequences and future image features. To prevent the vision-based policy from ignoring visual inputs and overfitting to proprioceptive states, HIT introduces an L2 loss on image features in addition to the standard behavioral cloning objective. Empirically, HIT consistently outperforms ACT across our evaluated tasks. While it is not designed to handle data from diverse domains or embodiments, we position HIT as a strong reference implementation for efficient imitation learning from in-domain robot demonstrations.
- *Heterogeneous Pretrained Transformer (HPT):* HPT [78] is a framework that pretrains a policy on heterogeneous datasets comprising simulation data, real robot trajectories, and human videos. It consists of stems, a trunk, and a head, where the stems and head serve roles analogous to our tokenizers and detokenizers. The trunk is designed to learn the complex mapping between inputs and outputs within a unified latent space through large-scale pretraining. The implementation of HPT differs from our framework in several aspects. First, we align data at the modality level and design the modular architecture in MXT to preserve modality-specific information across embodiments. In contrast, HPT uses a single tokenizer for visual inputs and another for proprioceptive inputs, along with a single detokenizer for all action dimensions. Additionally, HPT freezes its ResNet image encoder, whereas we finetune the ResNet encoder together with the rest of the network in an end-to-end manner for better adaptation to a specific embodiment.

More implementation details of these baselines can be found in Appendix Section 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 III, `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

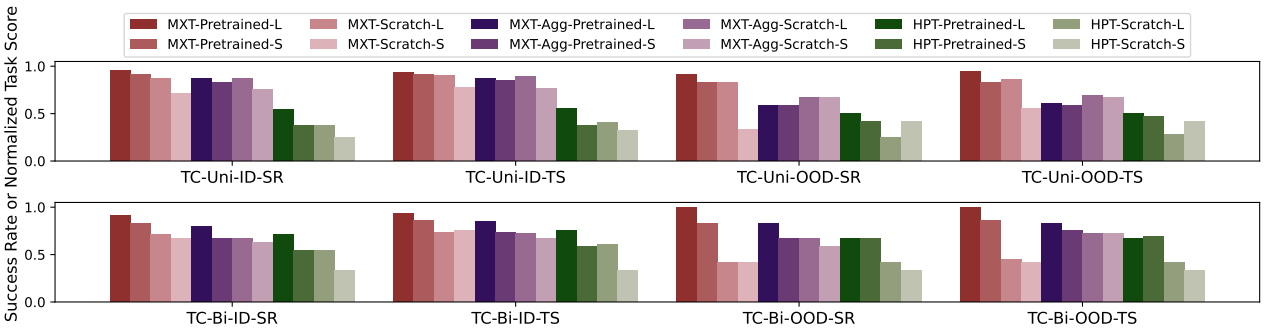

Fig. 5: Ablation study on unimanual and bimanual toy collection. We compare MXT, its ablation MXT-Agg, and baseline HPT on SR and TS. Here, "L" denotes the larger training set (40 trajectories for TC-Uni, 60 trajectories for TC-Bi), while "S" denotes the smaller training set (20 trajectories for TC-Uni,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

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 manipulation, 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 II. `Human2LocoMan`'s MXT presents strong performance using a relatively small robot dataset, achieving success rates above 79% across all tasks. 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 II, 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 SR and TS. Specifically, pretrained MXT achieves an average SR improvement of 41.9% overall and 79.7% under OOD settings, as well as an average TS improvement of 28.9% overall and 51.4% under OOD settings. From Figure 7, 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 in unimanual shoe organization, 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

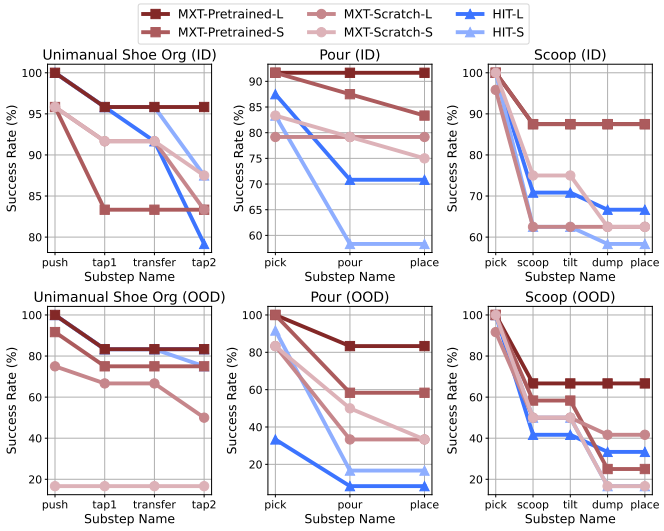

Fig. 6: Substep success rate. The success rate for some substep is calcuated as the percentage of trials where the robot successfully completed the substep. For each task, we calculate this with 24 ID rollouts and 12 OOD rollouts. **MXT-Pretrained**: MXT pretrained on human dataset (including unimanual 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).

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. As shown in Figure 5, we present SR and TS results based on 36 trials, comprising 24 OOD and 12 ID trials. HPT consistently underperforms compared to MXT, both when finetuned and when trained from scratch,

across all data sizes on the toy collection tasks. We attribute this performance gap to a combination of HPT's lack of modular architecture and its use of frozen image encoders. 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, HPT-Small and HPT-Base fail to achieve lower validation loss compared to HPT-Scratch, even pretrained on the large public dataset. Their policy rollout performance are evidently worse than HPT-Scratch and HPT-Pretrained so we did not scale up the real-robot evaluation. This indicates the challenge posed by the large embodiment gap and underscores the effectiveness of the human data collected by our system. Furthermore, 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. Summarizing from Table II, pretraining with human data improves the MXT policy SR by 38.6% overall and 82.7% under OOD settings, and boosts TS by 24.1% overall and 58.9% under OOD settings. This enables consistently stronger performance with only half the amount of robot data, demonstrating both the efficiency and robustness of our system. 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 II, 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 II 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.

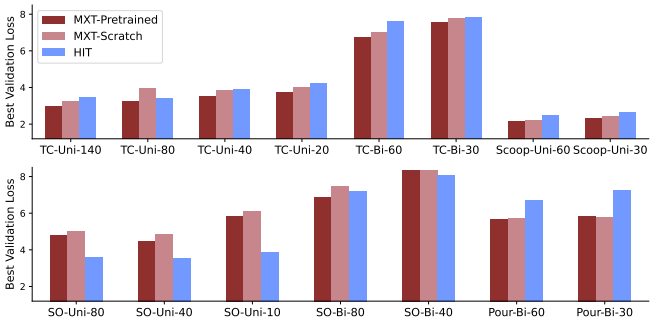

Fig. 7: Best validation loss of our method and HIT on all our tasks. **MXT-Pretrained**: MXT pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on the LocoMan data. **MXT-Scratch**: MXT trained only on the LocoMan data. The number suffix denotes the number of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

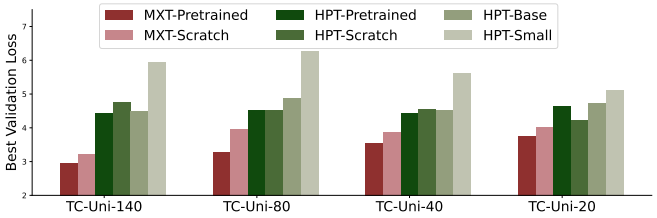

Fig. 8: Best validation loss of our method and HPT on the unimanual Toy Collection task. **MXT-Pretrained**: MXT pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on the LocoMan data. **MXT-Scratch**: MXT trained only on the LocoMan data. **HPT-Pretrained**: 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.

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

We have shown in Figure 5 that MXT outperforms HPT under both finetuning and training-from-scratch settings, benefiting from its modularized design and unfrozen image encoders in its architecture. Here, we observe the delta performance brought by human data across different methods. While both MXT and HPT benefit from pretraining on human data, MXT exhibits more effective Human-to-LocoMan transfer, achieving larger gains, especially in low-data and OOD settings. Additionally, as depicted in Figure 8, MXT-Pretrained consistently achieves lower validation loss than MXT-Scratch, whereas the gap between HPT-Pretrained and HPT-Scratch is less consistent and does not always show the same trend. These results highlight the ability of MXT to consume human data,

despite the large embodiment gap with LocoMan.

To further investigate the impact of modularity, we introduce an ablation variant of MXT, referred to as MXT-Agg, in which we *aggregate* the input modalities: a single visual tokenizer is used to encode all visual observations, a single proprioceptive tokenizer for all proprioceptive inputs, and a single detokenizer for all action dimensions—mirroring HPT's design. MXT-Agg incorporates HPT's key features, including cross-attention tokenization and trunk weight sharing, while still finetuning the vision encoders and remaining architecturally comparable to MXT. Across evaluations, MXT consistently benefits from pretraining and outperforms MXT-Agg when both are fine-tuned, highlighting the advantages of modularized tokenization for effectively leveraging human data. Notably, MXT-Agg exhibits weaker transfer performance with respect to MXT and 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.

V. Limitations

While our system introduces a novel and effective approach for cross-embodiment learning and data collection in quadrupedal manipulation, it has several limitations. First, although the teleoperation interface enables whole-body control and flexible data collection, it still requires a learning curve for operators. In particular, controlling the robot's torso via head motions may feel unintuitive and demand some practice to achieve high precision. Second, despite the modular design and cross-embodiment capability of our architecture, this work focuses solely on the LocoMan platform. We have not yet validated the generality of our system across a broader range of robot embodiments, such as table-top or mobile arms and humanoid robots. Lastly, while we have demonstrated effective human-to-robot transfer, we rely on a pretraining-finetuning pipeline for single robotic tasks, and future work could explore co-training with human data in multi-task settings. Extending Human2LocoMan to diverse robots, larger-scale datasets, and more learning paradigms remains promising future directions.

VI. Conclusion

We present `Human2LocoMan`, a unified framework for efficient data collection and cross-embodiment learning, enabling versatile quadrupedal manipulation skills on the open-source LocoMan platform. Our system integrates XR-based teleoperation with aligned human and robot data collection, supporting a shared observation-action representation across embodiments. To leverage this data effectively, we introduce the Modularized Cross-embodiment Transformer, a modular policy architecture designed for scalable pretraining and fine-tuning across different embodiments and manipulation modes. Extensive real-world experiments across six challenging household tasks demonstrate that `Human2LocoMan` achieves strong performance, robust generalization to out-of-distribution settings, and high data efficiency, attaining high success rates with only a small amount of robot data. Comparisons against strong baselines highlight the effectiveness of our modular design and the benefits of cross-embodiment pretraining using human data collected through our system.

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] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[33] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024.

[34] Tony Tao, Mohan Kumar Srirama, Jason Jingzhou Liu, Kenneth Shaw, and Deepak Pathak. Dexwild: Dexterous human interactions for in-the-wild robot policies. *arXiv preprint arXiv:2505.07813*, 2025.

[35] 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.

[36] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint*

arXiv:2203.12601, 2022.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

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

[79] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[80] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[81] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[82] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[83] 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 employ a simple cross-attention mechanism to convert input features of arbitrary length into a fixed set of tokens. Each attention layer in the tokenizers uses a hidden dimension of 128, with 4 attention heads, each having a head dimension of 32, and a dropout rate of 0.1. Other hyperparameters of each tokenizer are shown in Table V.

Similarly, we use cross-attention in the detokenizers to decode action modalities from a fixed number of output Transformer tokens. The attention layer is configured with 4 attention heads, each with a head dimension of 16, and a dropout rate of 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. As previously mentioned, masks are required to exclude redundant dimensions or modalities that are absent in certain embodiments. Below, we provide a more detailed description of our mask implementation.

a) Masks on images. We recognize that some image views may not be available across all embodiments. In the ecperiments of this paper, we assume at most two camera views (or image modalities): a main camera and a wrist camera. However, the framework can be easily extended to support any number of camera views. When a particular view is unavailable, we indicate this in the Transformer's trunk mask and insert dummy tokens at the corresponding positions. This ensures that tokens associated with the missing image modality are not attended to.

b) Masks on proprioceptive states. In some cases, certain dimensions of the proprioceptive state may be irrelevant for a specific embodiment. For example, in unimanual manipulation modes for both the human and LocoMan, the proprioceptive states of the inactive end-effector are not used. When only part of a proprioceptive modality contains redundant dimensions, we apply zero padding to those dimensions and encode them through the tokenizer as usual. Unlike masked image modalities, this does not affect the Transformer's trunk mask. However, when an entire proprioception modality should be disregarded, we handle it similarly to image modalities by applying the Transformer mask accordingly.

Data Normalization. For both human and LocoMan data, we apply normalization to observations and action labels. For non-image data, we compute the per-dimension mean and standard deviation from the dataset, and normalize using the standard formula:

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

For image data, we adopt the standard ImageNet statistics for the RGB channels to normalize each image in the dataset using the same formula, with mean $= [0.485, 0.456, 0.406]$ and std $= [0.229, 0.224, 0.225]$.

Dropout in Pretraining. We find that increasing the dropout rate in the Transformer trunk improves finetuning performance for MXT. In practice, setting the pretraining dropout rate to 0.5 for the scooping task and 0.4 for all other tasks yields consistently good results. When training with LocoMan data, whether training from scratch or during 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 [78], with the main exception of modifying the data normalization method to align with the approach used in other frameworks. This ensures a fair comparison of validation loss. The hyperparameters used for training HPT are summarized in Table VIII.

C. Global task-specific training parameters

We select a specific set of training parameters for each task and keep these settings consistent across all methods, as summarized 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