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 Artificial Intelligence
[*]Equal contributions

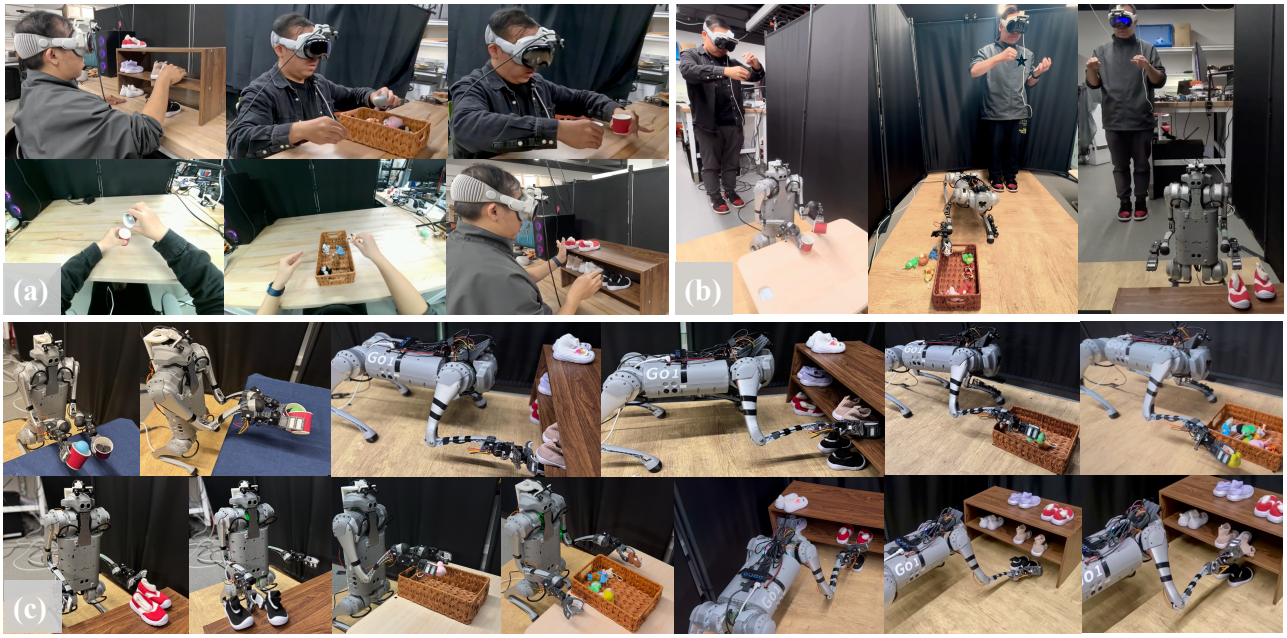

Fig. 1: **Human2LocoMan** is a framework for enabling flexible data-collection of (a) human demonstrations and (b) teleoperated robot trajectories, and for performing cross-embodiment training to synthesize (c) versatile quadrupedal manipulation skills.

***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 single-gripper 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 will be open-sourced at: https://human2bots.github.io.**

I. Introduction

While quadrupedal robots have demonstrated impressive locomotion capabilities in complex environments [1]–[7], it remains challenging to endow them with autonomous, versatile manipulation skills in a scalable way. In this work, we take inspiration from the open-source LocoMan hardware platform [8], which is a quadrupedal robot equipped with two leg-mounted manipulators. We focus on LocoMan as it is a versatile platform and allows for learning manipulation skills in multiple operating modes. Imitation learning has been a long-standing approach to teach robots complex skills through human demonstrations [9]. A key challenge in efficiently transferring skills from humans to quadrupedal robots lies in their embodiment gap, which leads to difficulties in both data collection and transfer learning. To bridge this gap, we develop Human2LocoMan, leveraging a novel teleoperation and data collection system to align human and robot data, and a modular Transformer architecture for cross-embodiment learning.

To collect data at scale, our data-collection system leverages an extended reality (XR) headset to capture human motions, and streams first-person or first-robot (during teleoperation) view to the human operator. To collect human data, the human operator simply wears the XR headset while performing any task. During teleoperation, we map the human and the quadruped to a unified frame to bridge the embodiment gap. In addition to mapping human hand motions to the grippers, we map human head motions to the robot's torso to expand the workspace and enhance

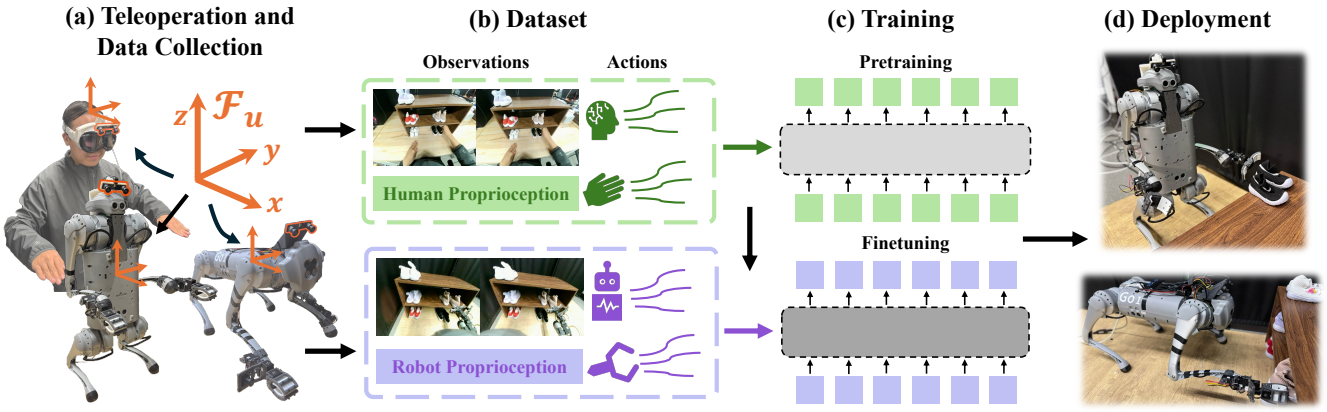

Fig. 2: **Human2LocoMan framework.** (a) The teleoperation system leverages a XR headset to collect human data and robot data via teleoperation. Human and robot data are mapped to a unified coordinate frame. (b) The dataset consists of aligned vision and proprioception from human and robot. (c) During training, the network is first pretrained on easy-to-collect human data, and then fine-tuned on a small amount of robot teleoperation data. (d) We evaluate the autonomous Human2LocoMan policies on three household tasks in single-gripper and bimanual mode.

active sensing. The target poses are passed to a whole-body controller to coordinate the robot motions.

In contrast to works that use egocentric human data to pretrain vision encoders [10] or learn high-level intent [11], we treat human as another embodiment and use human data for cross-embodiment learning. Despite mapping human and robot data to a unified frame, there exists obvious gaps ranging from 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 achieve positive transfer, the transformer-based policy is first pre-trained on human data, and then fine-tuned on a small amount of robot data.

We evaluate the trained polices on three household tasks under both unimanual and bimanual mode. We observe strong performance of MXT in comparison to strong baselines, positive transfer from human to robot, and that Human2LocoMan is more robust to out-of-distribution (OOD) scenarios.

In summary, our paper provides the following contributions:

- We propose Human2LocoMan, a framework that enables flexible data-collection of human demonstrations and teleoperated robot trajectories of versatile quadrupedal manipulation skills.
- We design a modular transformer architecture, MXT, to facilitate effective cross-embodiment learning even with a large embodiment gap.
- We introduce the first VR-based teleoperation system and manipulation dataset for the open-source LocoMan [8] hardware platform.
- We evaluate our policies on challenging household tasks across both unimanual and bimanual manipulation modes, demonstrating positive transfer from human to robot with high task success rates, strong performance scores, and robustness to OOD scenarios across five household manipulation tasks.

II. 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 OpenTelevision system [12] 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 ego-centric views, while additional cameras, such as RGB wrist cameras, can be optionally attached to the robot. Through the Human2LocoMan teleoperation system (Section II-B), the human operator can control the LocoMan robot to perform versatile manipulation tasks in both unimanual and bimanual modes. In the uni-manual 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 fine-tune 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

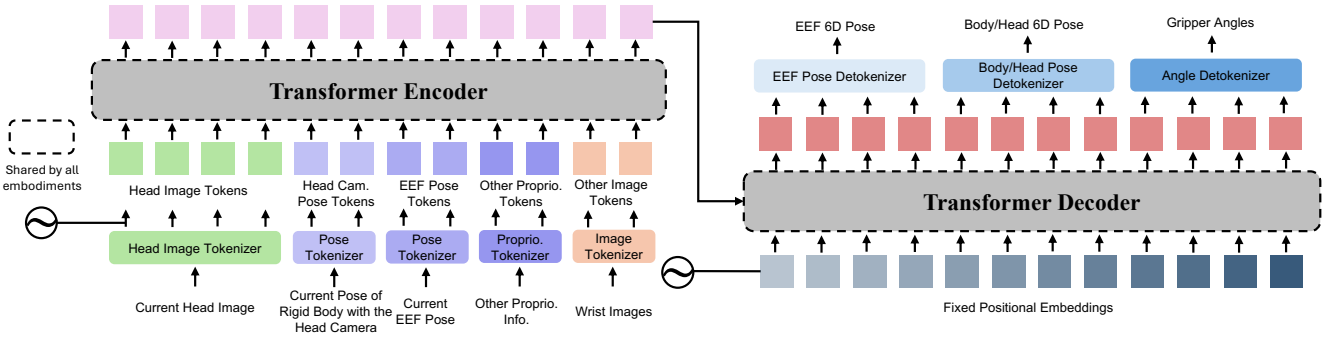

Fig. 3: **Modularized Cross-embodiment Transformer (MXT) architecture.** The inputs are organized as a list of modalities and encoded each by a separate tokenizer into a fixed number of tokens. The transformer trunk handles decision making by consuming the concatenated encoded tokens and producing a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

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 $\boldsymbol{R}_{\text{vr}}^{\cdot}$ defines 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 [8]. Please refer to Appendix Section IV-A for more details.

Data Collection. The details of `Human2LocoMan` data collection can be found in Appendix Section IV-B. We ensure that human and robot data are unified in both format and spatial interpretation, and can be used to train our proposed Modularized Cross-Embodiment Transformer introduced in Section II-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. 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 ***Modularized Cross-embodiment Transformer*** (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. The design details and training paradigm of MXT are elaborated in Appendix Section IV-C and IV-D, respectively.

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

			Toy Collection								**Shoe Rack Organization**								**Pouring**			
			Unimanual				Bimanual				Unimanual				Bimanual				Bimanual			
			In-distribution		OOD		In-distribution		OOD		In-distribution		OOD		In-distribution		OOD		In-distribution		OOD	
Method	Pre-trained	Data	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	110	50.0	34	66.7	52	25.0	14	66.7	66	8.33	8
HIT	-	larger	79.2	57	58.3	23	58.3	47	58.3	21	70.8	92	41.7	40	**83.3**	**63**	33.3	15	87.5	84	0	4
MXT	N	smaller	70.8	56	33.3	20	66.7	54	41.7	15	87.5	107	33.3	27	66.7	52	33.3	14	87.5	85	75	38
MXT	N	larger	87.5	**67**	83.3	31	70.8	53	41.7	16	83.3	107	25.0	15	75.0	60	58.3	23	79.2	79	66.7	33
MXT	Y	smaller	91.7	66	83.3	30	83.3	62	83.3	31	87.5	109	58.3	35	79.2	61	58.3	24	**91.7**	**88**	83.3	42
MXT	Y	larger	**95.8**	**67**	**91.7**	**34**	**91.7**	**67**	**100**	**36**	**100**	**120**	**83.3**	**56**	**83.3**	**63**	**75.0**	**29**	87.5	87	**91.7**	**45**

III. Experiments

In this section, we aim to answer the following research questions: (1) How does MXT compare to state-of-the-art imitation learning architectures? (2) How does human data collected by `Human2LocoMan` help with the performance of imitation learning with regards to its efficiency, robustness, and generalizability? (3) Do the design choices in MXT facilitate positive transfer from Human to LocoMan?

A. Experimental Setup

We evaluate MXT on five diverse household manipulation tasks—unimanual and bimanual toy collection, unimanual and bimanual shoe rack organization, and bimanual pouring—under both in-distribution and out-of-distribution (OOD) settings (Figure 5), using the LocoMan robot and data collected via the `Human2LocoMan` system. Success rates and task scores are used as evaluation metrics, with HIT [13] and HPT [14] serving as baselines. For detailed information on the experimental setups, including data statistics, model hyperparameters, masking strategies for embodiment alignment, and training configurations for MXT and the baselines, please refer to Appendix Section IV-G.

B. Results and Analysis

(1) How does MXT compare to state-of-the-art imitation learning architectures? We summarize the success rate (SR) and task score (TS) of our method and HIT across all tasks in Table I. In most evaluated tasks, spanning both unimanual and bimanual modes and across both in-distribution and out-of-distribution 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.

(2) How does human data collected by `Human2LocoMan` help with the performance of imitation learning with regards to its efficiency, robustness, and generalizability? As depicted in Table I, the pretraining on human data had a largely positive effect on `Human2LocoMan`'s downstream task performance—when the model is further finetuned on robot data. The policy can maintain strong performance when the robot data is scarcer, underscoring its efficiency and robustness. Our intuition is that the MXT is able to learn helpful complementarities (positive transfer artifacts) between the human and LocoMan data. The superior performance on OOD objects in tasks including TC-Bi, SO-Uni, and Pouring indicates that MXT is capable of adapting to scenarios unseen in the robot training data by learning from the human data on those scenarios in the pretraining stage.

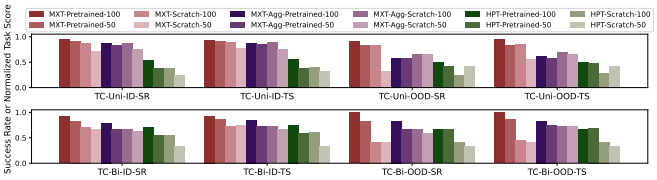

Fig. 4: Ablation study on unimanual and bimanual toy collection. We compare MXT, its ablation MXT-Agg, and baseline HPT on SR and TS. Here, 100 denotes the larger training set (40 trajectories for TC-Uni, 60 for TC-Bi), while 50 denotes the smaller set (20 for TC-Uni, 30 for TC-Bi).

(3) Do the design choices in MXT facilitate positive transfer from Human to LocoMan? Our framework presents positive cross-embodiment transfer despite substantial embodiment gaps. For comparisons, we provide SR and TS results from 36 trials in Fig. 4(b). 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, tokenize them with a single tokenizer, and decode actions with a single detokenizer; this baseline (marked with "Agg" in Fig. 4(b)) incorporates the key HPT designs, 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, likely due to the lack of modular design and the trade-off between improved performance from adaptive tokenization (image encoder finetuning) and reduced trunk transferability.

IV. Conclusions

In this paper, we presented `Human2LocoMan`, a framework for flexible data-collection via human demonstrations, teleoperation, and cross-embodiment training for versatile quadrupedal manipulation skills on the open-source LocoMan platform. Our data-collection framework unifies the action space of both human and robot, enabling the system to bridge the gap between diverse agent morphologies; with this framework we provide the first quadrupedal manipulation dataset for the LocoMan platform, which we use for policy training across different embodiments (human, LocoMan) and different operating modes (unimanual, bimanual). We demonstrated the effectiveness of our framework on five challenging household tasks with significant performance improvements compared to strong baselines.

REFERENCES

[1] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, p. eadh5401, 2024.

[2] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[4] R. Yang, G. Yang, and X. Wang, "Neural volumetric memory for visual locomotion control," in *CVPR 2023*, 2023.

[5] Y. Yang, G. Shi, C. Lin, X. Meng, R. Scalise, M. G. Castro, W. Yu, T. Zhang, D. Zhao, J. Tan *et al.*, "Agile continuous jumping in discontinuous terrains," *arXiv preprint arXiv:2409.10923*, 2024.

[6] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[7] B. Lindqvist, S. Karlsson, A. Koval, I. Tevetzidis, J. Haluška, C. Kanellakis, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue," *Robotics and Autonomous Systems*, 2022.

[8] C. Lin, X. Liu, Y. Yang, Y. Niu, W. Yu, T. Zhang, J. Tan, B. Boots, and D. Zhao, "Locoman: Advancing versatile quadrupedal dexterity with lightweight loco-manipulators," *arXiv preprint arXiv:2403.18197*, 2024.

[9] S. Schaal, "Learning from demonstration," *Advances in neural information processing systems*, vol. 9, 1996.

[10] M. K. Srirama, S. Dasari, S. Bahl, and A. Gupta, "Hrp: Human affordances for robotic pre-training," *arXiv preprint arXiv:2407.18911*, 2024.

[11] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," *arXiv preprint arXiv:2302.12422*, 2023.

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

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

[14] L. Wang, X. Chen, J. Zhao, and K. He, "Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers," 2024. [Online]. Available: https://arxiv.org/abs/2409.20537

[15] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. 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)*. IEEE, 2019, pp. 614–619.

[16] L. Wang, X. Chen, J. Zhao, and K. He, "Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers," *arXiv preprint arXiv:2409.20537*, 2024.

[17] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[18] Y. Ji, G. B. Margolis, and P. Agrawal, "Dribblebot: Dynamic legged manipulation in the wild," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5155–5162.

[19] F. Shi, T. Homberger, J. Lee, T. Miki, M. Zhao, F. Farshidian, K. Okada, M. Inaba, and M. Hutter, "Circus anymal: A quadruped learning dexterous manipulation with its limbs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2316–2323.

[20] X. Cheng, A. Kumar, and D. Pathak, "Legs as manipulator: Pushing quadrupedal agility beyond locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5106–5112.

[21] M. Zhang, Y. Ma, T. Miki, and M. Hutter, "Learning to open and traverse doors with a legged manipulator," *arXiv preprint arXiv:2409.04882*, 2024.

[22] Z. He, K. Lei, Y. Ze, K. Sreenath, Z. Li, and H. Xu, "Learning visual quadrupedal loco-manipulation from demonstrations," *arXiv preprint arXiv:2403.20328*, 2024.

[23] T. Huang, N. Sontakke, K. N. Kumar, I. Essa, S. Nikolaidis, D. W. Hong, and S. Ha, "Bayrntune: Adaptive bayesian domain randomization via strategic fine-tuning," *arXiv preprint arXiv:2310.10606*, 2023.

[24] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning whole-body manipulation for quadrupedal robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, 2023.

[25] J. Stolle, P. Arm, M. Mittal, and M. Hutter, "Perceptive pedipulation with local obstacle avoidance," *arXiv preprint arXiv:2409.07195*, 2024.

[26] R.-Z. Qiu, Y. Hu, G. Yang, Y. Song, Y. Fu, J. Ye, J. Mu, R. Yang, N. Atanasov, S. Scherer *et al.*, "Learning generalizable feature fields for mobile manipulation," *arXiv preprint arXiv:2403.07563*, 2024.

[27] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149.

[28] K. N. Kumar, I. Essa, and S. Ha, "Cascaded compositional residual learning for complex interactive behaviors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4601–4608, 2023.

[29] Y. Feng, C. Hong, Y. Niu, S. Liu, Y. Yang, W. Yu, T. Zhang, J. Tan, and D. Zhao, "Learning multi-agent loco-manipulation for long-horizon quadrupedal pushing," *arXiv preprint arXiv:2411.07104*, 2024.

[30] Z. Xiong, B. Chen, S. Huang, W.-W. Tu, Z. He, and Y. Gao, "Mqe: Unleashing the power of interaction with multi-agent quadruped environment," *arXiv preprint arXiv:2403.16015*, 2024.

[31] T. An, J. Lee, M. Bjelonic, F. De Vincenti, and M. Hutter, "Solving multi-entity robotic problems using permutation invariant neural networks," *arXiv preprint arXiv:2402.18345*, 2024.

[32] O. Nachum, M. Ahn, H. Ponte, S. Gu, and V. Kumar, "Multi-agent manipulation via locomotion using hierarchical sim2real," *arXiv preprint arXiv:1908.05224*, 2019.

[33] Y. Ji, B. Zhang, and K. 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)*. IEEE, 2021, pp. 899–904.

[34] G. Pan, Q. Ben, Z. Yuan, G. Jiang, Y. Ji, J. Pang, H. Liu, and H. Xu, "Roboduet: A framework affording mobile-manipulation and cross-embodiment," *arXiv preprint arXiv:2403.17367*, 2024.

[35] Q. Wu, Z. Fu, X. Cheng, X. Wang, and C. Finn, "Helpful doggybot: Open-world object fetching using legged robots and vision-language models," *arXiv preprint arXiv:2410.00231*, 2024.

[36] M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Yang, and X. Wang, "Visual whole-body control for legged loco-manipulation," *arXiv preprint arXiv:2402.16796*, 2024.

[37] J. Zhang, N. Gireesh, J. Wang, X. Fang, C. Xu, W. Chen, L. Dai, and H. Wang, "Gamma: Graspability-aware mobile manipulation policy learning based on online grasping pose fusion," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1399–1405.

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

[39] P. Arm, M. Mittal, H. Kolvenbach, and M. Hutter, "Pedipulate: Enabling manipulation skills using a quadruped robot's leg," in *41st IEEE Conference on Robotics and Automation (ICRA 2024)*, 2024.

[40] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[41] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[42] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, no. 1, pp. 297–330, 2020.

[43] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis, "Strap: Robot sub-trajectory retrieval for augmented policy learning," in *The Thirteenth International Conference on Learning Representations*, 2025.

[44] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-thewild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

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

[46] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-

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

[47] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[48] X. He, C. Yuan, W. Zhou, R. Yang, D. Held, and X. Wang, "Visual manipulation with legs," *arXiv preprint arXiv:2410.11345*, 2024.

[49] M. Seo, H. A. Park, S. Yuan, Y. Zhu, and L. Sentis, "Legato: Cross-embodiment imitation using a grasping tool," *arXiv preprint arXiv:2411.03682*, 2024.

[50] R.-Z. Qiu, Y. Song, X. Peng, S. A. Suryadevara, G. Yang, M. Liu, M. Ji, C. Jia, R. Yang, X. Zou *et al.*, "Wildlma: Long horizon loco-manipulation in the wild," *arXiv preprint arXiv:2411.15131*, 2024.

[51] X. Lin, J. So, S. Mahalingam, F. Liu, and P. Abbeel, "Spawnnet: Learning generalizable visuomotor skills from pre-trained network," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4781–4787.

[52] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu, "Robot learning on the job: Human-in-the-loop autonomy and learning during deployment," *The International Journal of Robotics Research*, p. 02783649241273901, 2022.

[53] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *Conference on Robot Learning*. PMLR, 2023, pp. 1199–1210.

[54] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation," *arXiv preprint arXiv:2403.04436*, 2024.

[55] Z. Zhang, Y. Niu, Z. Yan, and S. Lin, "Real-time whole-body imitation by humanoid robots and task-oriented teleoperation using an analytical mapping method and quantitative evaluation," *Applied Sciences*, vol. 8, no. 10, p. 2005, 2018.

[56] L. P. Poubel, S. Sakka, D. Ćehajić, and D. 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)*. IEEE, 2014, pp. 1782–1787.

[57] B. Sen, M. Wang, N. Thakur, A. Agarwal, and P. Agrawal, "Learning to look around: Enhancing teleoperation and learning with a human-like actuated neck," *arXiv preprint arXiv:2411.00704*, 2024.

[58] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu, "Egomimic: Scaling imitation learning via egocentric video," *arXiv preprint arXiv:2410.24221*, 2024.

[59] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang, "Mobile-television: Predictive motion priors for humanoid whole-body control," *arXiv preprint arXiv:2412.07773*, 2024.

[60] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024.

[61] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang, "Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning," *arXiv preprint arXiv:2407.03162*, 2024.

[62] S. Yang, M. Liu, Y. Qin, R. Ding, J. Li, X. Cheng, R. Yang, S. Yi, and X. Wang, "Ace: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation," *arXiv preprint arXiv:2408.11805*, 2024.

[63] H. Fang, H.-S. Fang, Y. Wang, J. Ren, J. Chen, R. Zhang, W. Wang, and C. Lu, "Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 031–15 038.

[64] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. 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)*. IEEE, 2024, pp. 12 156–12 163.

[65] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu, "Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback," *arXiv preprint arXiv:2410.08464*, 2024.

[66] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.

[67] G. Papagiannis, N. Di Palo, P. Vitiello, and E. Johns, "R+ x: Retrieval and execution from everyday human videos," *arXiv preprint arXiv:2407.12957*, 2024.

[68] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," *arXiv preprint arXiv:2403.07788*, 2024.

[69] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

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

[71] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. 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)*. IEEE, 2024, pp. 6892–6903.

[72] L. Y. Chen, K. Hari, K. Dharmarajan, C. Xu, Q. Vuong, and K. Goldberg, "Mirage: Cross-embodiment zero-shot policy transfer with cross-painting," *arXiv preprint arXiv:2402.19249*, 2024.

[73] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," *arXiv preprint arXiv:2408.11812*, 2024.

APPENDIX

A. `Human2LocoMan` 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 [8], 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 facilitate smooth teleoperation of a dynamic quadrupedal platform with whole-body motions, we consider the handle 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 [15] 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.

B. `Human2LocoMan` 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 the wrist camera, respectively. Then, we can formulate o_t^{R} and a_t^{R} in the dataset as follows.

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

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

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

C. Design Details of MXT

Tokenizers. The tokenizers T transform raw observations into tokens for the transformer trunk. Drawing from the design in previous works [16], 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 pre-trained ResNet encoder that can be fine-tuned 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 [17]. At each inference step, the detokenizers predict an action sequence of h steps and temporal ensembling is applied to the outputs, following [17]. 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 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 tokens 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 Section IV-H 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. MXT 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 pre-training and fine-tuning. 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.

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

Require: Human dataset $\mathcal{D}_{\text{human}}$, LocoMan dataset $\mathcal{D}_{\text{LocoMan}}$
Ensure: Policy π for versatile LocoMan manipulation
  Initialize the MXT policy network π
  for $n = 1, 2, ...$ **do** ▷ Pretraining Stage
    Sample a batch from $\mathcal{D}_{\text{human}}$
    Optimize the MXT policy π with the BC objective 5 on the batch
  Reinitialize the tokenizers and detokenizers of π
  for $n = 1, 2, ...$ **do** ▷ Finetuning Stage
    Sample a batch from $\mathcal{D}_{\text{LocoMan}}$
    Optimize the MXT policy π with the BC objective 5 on the batch

E. `Human2LocoMan` Embodiments

Notably, the unimanual and bimanual modes represent distinct embodiments, each differing in morphology, observations, and action spaces (Table II).

TABLE II: Human2LocoMan embodiments (LM=LocoMan).

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-Uni (R)	✓	×	✓	✓	×	✓	✓	×	✓	×
Human-Uni (L)	✓	×	✓	×	✓	✓	×	✓	×	✓
Human-Bi	✓	×	✓	✓	✓	✓	✓	✓	✓	✓
LM-Uni (R)	✓	✓	✓	✓	✓	✓	✓	×	✓	×
LM-Uni (L)	✓	✓	✓	✓	✓	✓	×	✓	×	✓
LM-Bi	✓	×	✓	✓	✓	×	✓	✓	✓	✓

F. 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.) [18]–[25], using a back-mounted arm for tabletop tasks [26], [27], or using leg-mounted manipulators for spatially-constrained (e.g., reaching toys underneath furniture) or bi-manual manipulation tasks [8]. In this work, we take inspiration from the open-source LocoMan hardware platform [8], 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 [18], [19], [21]–[25], [28]–[33] and for training whole-body controllers to track end-effector poses for uni-manual grasping [27], [34]–[39]; 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 [9], [40]–[42] and thus provides an alternative approach for efficiently acquiring more general

manipulation skills [43]–[47]. 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 [48], or grasping policies for a top-mounted arm using data collected from real-world demonstrations [38], [49], [50]. Our work introduces a scalable way of achieving more versatile manipulation skills on quadrupedal platforms encompassing both single-gripper and bi-manual 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 [51]–[53] are commonly used to directly teleoperate the robot for data collection. Cameras are employed to capture human motions and map them to the robot [11], [13], [54]–[56]. 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 [12], [50], [57]–[61]. While most above works 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 [17], [58], [62]–[64]. 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-assist [65] or hand-held grippers [49], [66], although these are constrained to a specific robot or end-effector type. Other works enable more ergonomic data collection with body-worn cameras [67], [68] or VR glasses [58]. 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 pre-training.

Cross-Embodiment Learning: Drawing from the success of foundation models in computer vision and natural language, there are many endeavors to replicate the success in robotics by training generalist policies on large-scale data from different embodiments [16], [69]–[73]. However, this remains an open challenge due to the heterogeneity of robot embodiments, and gaps in kinematics, vision, and proprioception.

Different neural architecture were proposed to handle the heterogeneity. CrossFormer [73] formulated policy learning as a sequence-to-sequence problem, so that any number of camera views or proprioceptive sensors can be handled as sequence of tokens, and add special readout tokens as part of the input sequence. In comparison, HPT [16] features a modularized structure and maps the variable observations to a fixed number of number tokens. In our work, we propose

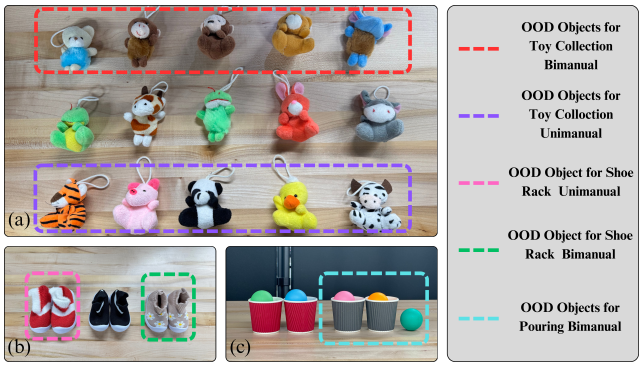

Fig. 5: Objects utilized in our experiments. The highlighted objects are out-of-distribution (OOD) objects for robot fine-tuning, while all the objects are utilized for human pretraining and real-robot evaluation. (a) Toy collection. (b) Shoe rack organization. (c) Pouring.

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 [58] proposed the idea that human be treated as another embodiment and demonstrated positive transfer by co-training on human and robot data. To achieve such positive transfer, EgoMimic minimizes human and robot kinematic gap by choosing a human-like robot embodiment, proprioception gap by normalizing and align action distributions, and appearance gap with visual masking. In comparison, `Human2LocoMan` is more flexible and scalable, transferring from human to quadruped without explicit domain alignment.

G. Experiment Setups

1) Tasks: We evaluate MXT on five 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 5, we use 10 objects for robot finetuning and all objects for human pretraining and real-robot evaluation.
- *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.
- *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

TABLE III: Records of data collection for different tasks.

Task	# human traj.	human time (min)	# robot traj.	robot time (min)
TC-Uni	240	20	160	16
TC-Bi	315	22	100	10
SO-Uni	240	34	90	23
SO-Bi	200	20	120	15
Pouring	210	40	70	25

coordinate whole-body motions to reach various rack levels and utilize both prehensile and non-prehensile manipulation skills. As shown in Figure 5, this task involves three pairs of shoes, with one pair being out-of-distribution (OOD).

- *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.
- *Pouring.* 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.

2) 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.

3) Training details.: We pretrain a model for TC that utilizes the human data of both unimanual TC and bimanual TC, then we finetune the model on each TC task with its robot data. Similarly, we pretrain a model for SO that utilizes the human data of both unimanual SO and bimanual SO, then we finetune the model on each SO task with its 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 Section IV-L.) We also listed the model hyperparameters we use for our method and the baselines in the Section IV-H and IV-I.

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

- *Humanoid Imitation Transformer (HIT):* HIT [13] is an imitation learning framework designed for humanoid skill learning that also extends to any robot embodiment. It builds upon ACT [17] 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 [16] 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. Note that HPT is established in a different context than ours: while HPT focuses on scaling up robot imitation learning, our work emphasizes transferring from one embodiment to another on a given task. Consequently, the implementation of HPT differs from our framework in several key aspects. Firstly, we leverage the unified observation and 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 Section IV-I. 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.

5) 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 a fixed number of times with in-distribution (ID) objects or out-of-distribution (OOD) objects, as shown in Table IV.

For each task, we specify a few critical subgoals in order to complete the task in full. When calculating the task score, reaching each intermediate subgoal counts as 1 point, and reaching the final goal, i.e., completing the task, rewards 2 points. The final task score is summed over all the rollouts on this task. The success rate for one method on one task with either the in-distribution or OOD setting is computed as the ratio of successful (i.e., where all subgoals are reached) rollouts in all performed rollouts.

In addition, we report the best validation loss as another metric for training performance. Because it is much easier to evaluate validation loss, we are able to conduct more ablation on how the performance scales with the data size.

H. Implementation and Training details of MXT

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

TABLE IV: Number of rollouts for each task on in-distribution (ID) objects and out-of-distribution (OOD) objects.

Task	# ID rollouts	# OOD rollouts
Unimanual Toy Collection	24	12
Bimanual Toy Collection	24	12
Unimanual Shoe Rack Organization	10	5
Bimanual Shoe Rack Organization	24	12
Pouring	8	4

TABLE V: 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 VI.

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 VII

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 general, we find that setting the pretraining dropout rate to 0.4 yields reasonably good performance on all tasks. When training with LocoMan data, including training from scratch and finetuning, the transformer trunk dropout rate is reverted to 0.1.

TABLE VI: 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 VII: MXT detokenizer hyperparameters

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

I. Implementation details of baselines

HIT. Our implementation of Humanoid Imitation Transformer [13] is based on the released codebase, with only minor modifications to accommodate our data format. The hyperparameters used for training are summarized in Table VIII.

HPT. We follow the original implementation of HPT [16], 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 IX.

TABLE VIII: 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

TABLE IX: 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

J. OOD Analysis

Figure 6 showcases different subtask success rates for subtasks of various manipulation skills, including picking, pouring, placing for pouring, and pushing, tapping, and transferring for SO-Uni. MXT maintains the performance consistently for most subtasks, indicating its robustness to unseen task settings.

K. Validation Loss Analysis

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. For the more complex Shoe Rack Organization task, MXT achieves validation loss on par with HIT despite the challenge of whole-body coordination. It shows that MXT achieves better or equivalent training efficiency compared to HIT across various household tasks.

We also provide validation loss results, in comparison with HPT in Figure 8, which shows strong performance of our method in the Toy Collection (TC) and Pouring (Pour) and equivalent performance in the challenging Shoe Organization (SO) task in greater number of demonstrations. It is worth noting that we observe severe overfitting in HPT experiments when training on our own datasets, which is not observed in MXT. This further suggests that the MXT architecture induces better generalization by leveraging a more modularized design. However, we do observe that the performance advantage is not consistent across all tasks, and the relative performance of MXT, in either the in-distribution or OOD case, is strongly correlated with the task type. This suggests that the data quality could play an important role in the effectiveness of our method, which we intend to investigate as part of our future work.

From Figure 8, we find that the MXT-Finetuned model shows a significant reduction in validation loss compared to MXT-FromScratch, highlighting the benefits of pretraining on human demonstrations. Similarly, HPT-Finetuned outperforms HPT-FromScratch, but MXT-Finetuned achieves lower validation loss than all HPT models. The HPT-Small and HPT-Base models do not generalize as well as MXT-Finetuned. By comparing HPT-From Scratch with HPT-Finetuned, we find that the performance is not improved by pretraining with our collected human data on HPT. These indicate the benefits of the modularized design of MXT to consume human data which has a large embodiment gap from the LocoMan data.

TABLE X: Global training parameters for each task

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

L. 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 X.

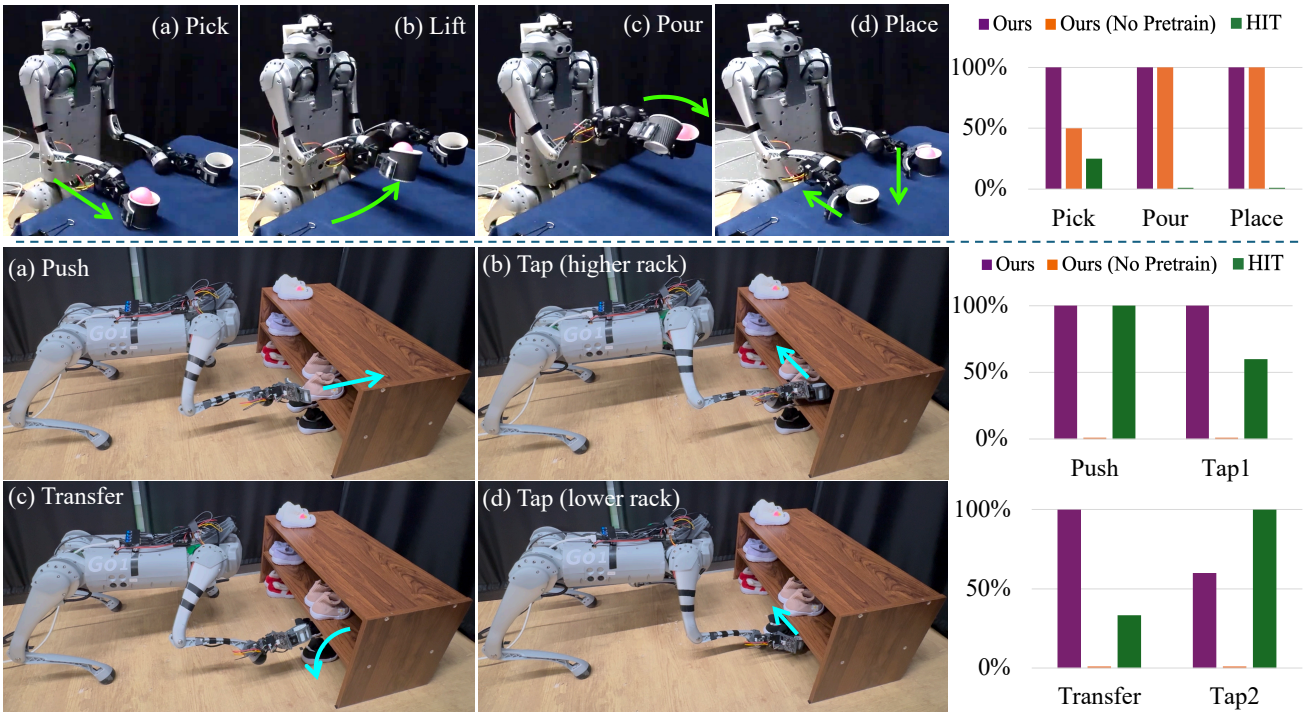

Fig. 6: This figure illustrates the performance of MXT on the pouring (upwards) and unimanual SO-Uni (downwards) tasks with OOD settings respectively. The right figures demonstrate the success rates of the subtasks for each task.

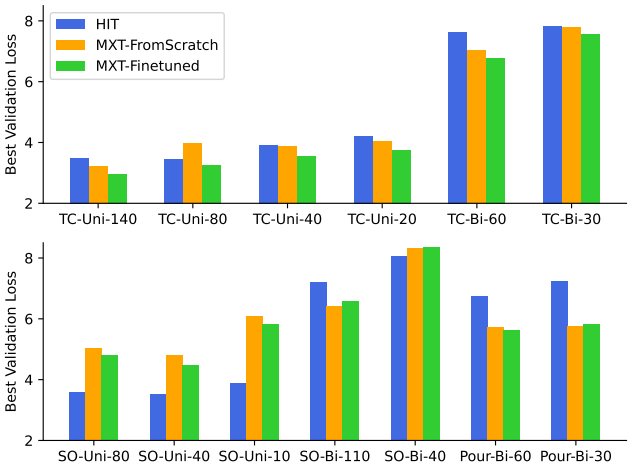

Fig. 7: Best validation loss of our method and HIT on all our tasks. **MXT-FromScratch**: Ours, trained only on LocoMan data. **MXT-Finetuned**: Ours, pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on LocoMan data. Task and mode identifiers: **TC** - Toy Collection, **SO** - Shoe rack Organization, **Pour** - Pouring, **Uni** - Unimanual mode, **Bi** - Bimanual mode. The number suffix denotes the number of demonstrations in the LocoMan training set.

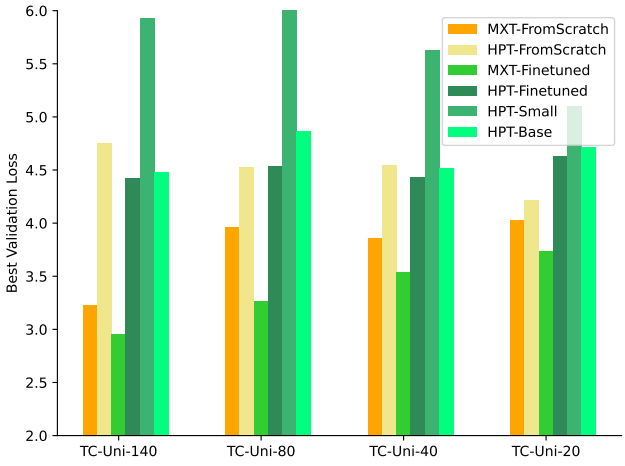

Fig. 8: Best validation loss of our method and HPT on the unimanual Toy Collection task. **MXT-FromScratch**: Ours, trained only on LocoMan data. **MXT-Finetuned**: Ours, pretrained on human dataset (including unimanual and bimanual if applicable), then finetuned on LocoMan data. **HPT-FromScratch**: HPT network trained only on LocoMan data. **HPT-Finetuned**: HPT trunk pretrained on our human data, then finetuned on LocoMan data. **HPT-Small**: Finetune with our LocoMan data with HPT trunk initialized with released `HPT-Small` weights. **HPT-Base**: Finetune with our LocoMan data with HPT trunk initialized with released `HPT-Base` weights. See Fig. 7 for task identifiers.