Learning Multi-Modal Whole-Body Control for Real-World Humanoid Robots

Pranay Dugar, Aayam Shrestha, Fangzhou Yu, Bart van Marum, Alan Fern
{dugarp, shrestaa, yufangzh, vanmarub, afern}@oregonstate.edu *

Abstract: We introduce the Masked Humanoid Controller (MHC) for whole-body tracking of target trajectories over arbitrary subsets of humanoid state variables. This enables the realization of whole-body motions from diverse sources such as video, motion capture, and VR, while ensuring balance and robustness against disturbances. The MHC is trained in simulation using a carefully designed curriculum that imitates partially masked motions from a library of behaviors spanning pre-trained policy rollouts, optimized reference trajectories, retargeted video clips, and human motion capture data. We showcase simulation experiments validating the MHC's ability to execute a wide variety of behavior from partially-specified target motions. Moreover, we also highlight sim-to-real transfer as demonstrated by real-world trials on the Digit humanoid robot. To our knowledge, this is the first instance of a learned controller that can realize whole-body control of a real-world humanoid for such diverse multi-modal targets.

Keywords: Humanoid Robot Learning, Motion Tracking, Multi-Modal Control

1 Introduction

Humanoid robots hold immense potential as highly capable and adaptive platforms for tackling complex real-world tasks, thanks to their dexterous multi-purpose body structure that mirrors our own. However, the development of versatile and robust whole-body controllers for bipedal humanoids remains a critical challenge in robotics. Traditional approaches involve meticulous manual engineering of separate controllers for different skills such as standing [1], walking [2], and manipulation [3], resulting in specialized controllers with limited versatility and adaptability.

A truly versatile whole body humanoid controller should exhibit several key properties. First, it should comprehend and execute target motions specified through *multiple input modalities*, such as video demonstrations, motion capture data, or high-level locomotion and end-effector targets. Second, the controller should be be *robust* to dynamic command updates, noisy or inexact inputs, inaccurate simulation parameters, and natural external disturbances. Finally, the controller should be *versatile*, allowing straightforward extension of its repertoire as new motion examples become available, with no or minimal retraining.

To address these challenges, we train the Masked Humanoid Controller (MHC), a whole-body controller that is able to accommodate target motions specified as future trajectories over full or partial robot poses. This allows the MHC to support the above multi-modality property, for example, following walking trajectories specified solely by desired velocities or imitating arm-only motions extracted from video clips. We train the MHC via reinforcement learning using a carefully designed curriculum and an expansive library of behaviors spanning optimized reference trajectories, re-targeted video clips, and human motion capture data. The tailored curriculum over motion commands and disturbances gradually introduces capabilities with the aim of achieving the above desired robustness and versatility properties.

*All authors are with the Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University. Corvallis, Oregon, USA. Project webpage: masked-humanoid.github.io/mhc/

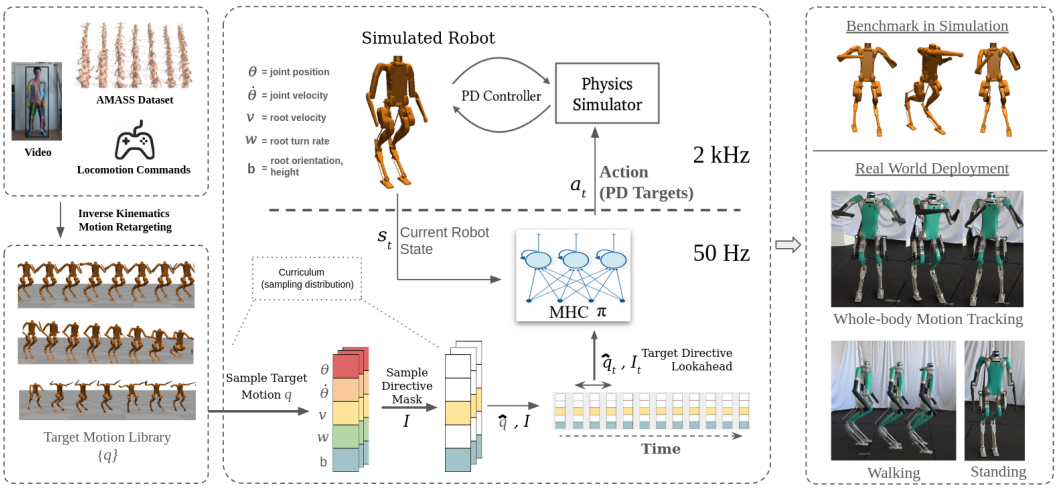

Figure 1: The Masked Humanoid Controller (MHC) is learned from a dataset of retargeted human motions paired with torso locomotion commands. During training and testing masking can be applied to target motion trajectories to yield masked motion directives that are given to the MHC. The MHC then produces PD setpoints for the whole body in order to track the current motion directive. Training includes domain randomization and force perturbations to facilitate robustness and transfer from simulation to the real robot.

We validate the MHC through simulated benchmarking and showcase qualitative results for sim-to-real transfer using the Digit V3 robot platform. The simulation experiments demonstrate the importance of curriculum and architecture choices and demonstrate generalization to new motions. Our real world experiments demonstrate the robustness of the MHC over a variety of target behaviors such as walking, boxing, and box pickup, while also highlighting current shortcomings in sim-to-real transfer. To our knowledge, this is the first learning-based framework for whole-body real-world humanoid control that can tractably accommodate such a wide spectrum of multi-modal motion directives within a single unified controller.

2 Related Work

Robust Locomotion Control. Recent advancements in robotic locomotion showcase adaptive control strategies that enable robots to navigate complex terrains and perform dynamic maneuvers. Significant progress has been made in the locomotion of quadrupeds, demonstrating enhanced stability [4] [5] and agility [6] in various environments. Fu et al. [7] train a unified whole body manipulation and locomotion controller that attaches an arm on top of a quadraped to perform mobile manipulation tasks. Notable achievements for bipedal locomotion with the Cassie platform (no upper torso or arms) include blind locomotion across multiple gaits and behaviors [8, 9], locomotion under varying loads [10], and visually-guided locomotion over irregular terrain using deep learning to process visual input and adapt locomotion strategies accordingly [11]. More recently a learned controller for blind locomotion on the full-body humanoid Digit was demonstrated across varying terrain and disturbances [12]. These advancements underscore the importance of robust data and simulation technologies in developing and refining control strategies. However, managing transitions between different modes of locomotion, such as standing and walking, remains a significant challenge. Van Marum et al. [13] addressed this complexity with the development of the SaW controller for Digit, which integrates stability and adaptive walking strategies into a single operational framework that maintains balance against natural disturbances. However, standing and walking alone do not constitute whole-body control. Building on the foundation laid by the SaW controller, our approach enhances it by incorporating comprehensive whole-body control, allowing for more versatile and adaptive humanoid robot behaviors.

Motion Tracking for Simulated Humanoids. Motion capture data has been used extensively in generating motions for simulated characters. Most closely related to our work is the application

of reinforcement learning with tracking objective over fully specified target motions [14, 15, 16, 17, 18, 19, 20]. However, tracking rewards alone do not always yield smooth transitions between skills and failure recovery. Recent works augment training with adversarial losses to encourage natural motions during transitions [20, 21] or define an explicit fail state recovery policy [16]. While some works allow for more intuitive under-specified control, they focus on controllers for predefined types of sparsity. Adapting to new sparsity specifications, such as VR, requires retraining [22, 23]. Moreover, discrepancies between simulated and physical environments, such as differences in environmental physics and unrealistic joint definitions in simulated characters, can lead to behaviors that are not physically plausible or efficient when applied to actual robots. Our work aims to both incorporate a wide variety of sparsity types in a single controller and address the simulation to reality gap.

Whole-Body Motion Tracking for Humanoids. Efforts to closely replicate human motion and interaction within robotic frameworks have been significantly enhanced by both simulation techniques and biological inspirations. Early works by Dariush et al. [] and recent advancements by Li et al. [24] have advanced the field of humanoid motion tracking for locomotion, emphasizing the utilization of optimized trajectories and multi-stage training setups to facilitate transfer from simulation to the real world. The latest research by Cheng et al. [25] employs human motion capture data to achieve partial whole body control, applying imitation constraints on the upper body and relaxing those constraints for lower body. However, all these approaches focus on specific subsets of human motion, such as only upper-body movements or only locomotion control, and may not fully capture the intricate whole-body coordination required for natural and versatile humanoid motion. Our work addresses this limitation by leveraging the MHC framework to enable comprehensive whole-body motion tracking and coordination in humanoids, facilitating more natural and adaptable motion generation.

3 Problem Formulation

Given a dataset of humanoid motions from various behaviors, our objective is to learn a controller which can match target motion directives that are representative of the data distribution. We are particularly interested in supporting partially-specified motion directives e.g. only specifying the upper-body joint trajectories, or just the torso velocity, or both. It is expected that the controller "fills in the blanks" for joints that are not specified by a directive, e.g. details of the lower-body joints. Such a controller can support directives derived from various input modalities. For example, fully-specified directives that cover all humanoid joints can be derived from MoCap data, while joystick commands regarding the root velocity and arm movements correspond to partial directives.

More formally, a *motion* is a sequence of poses $q_{1:H}$ for a humanoid with J joints over H time steps. Each pose is represented by a tuple $q_i = (\theta_i, \dot{\theta}_i, v_i, w_i, b_i)$, where $\theta_i \in \mathbb{R}^J$ denotes the joint angles, $\dot{\theta}_i \in \mathbb{R}^J$ denotes the joint angular velocities, $v_i \in \mathbb{R}^2$ denotes the root planar linear velocities, $w_i \in \mathbb{R}$ denotes the turn rate for the humanoid, and $b_i \in \mathbb{R}^3$ denotes the Euler base orientation in the x-axis, y-axis, and the height of the base from the ground plane. Motion directives are used to specify constraints on a desired motion to be generated. Specifically, a *motion directive* d is defined as a masked motion sequence represented by $d = (\hat{q}_{1:H}, I_{1:H})$, where $\hat{q}_{1:H}$ is a masked motion and $I_{1:H}$ is a sequence of binary masks such that I_i indicates which dimensions of pose q_i are selected as motion constraints. We follow the convention of setting the masked dimensions of $\hat{q}$ to zero. Note that while the definition of a directive allows for arbitrary sets of state variables to be masked, in practice, we focus training and evaluation masking patterns relevant to the multiple modalities the controller needs to support as described in Section 4.

The Masked Humanoid Controller (MHC) is a controller π that at each time step t takes an input containing the current *humanoid state* $s_t = (\theta_t, \dot{\theta}_t, \omega_t)$ and a target motion directive d, where $\theta_t \in \mathbb{R}^J$ and $\dot{\theta}_t \in \mathbb{R}^J$ are the joint positions and velocities, and ω_t is the orientation in quaternion form. The output of the MHC is an action $a_t = \pi(s_t, d)$, which in our work corresponds to PD setpoints for all robot motors. The objective of the MHC is to select actions such that the future humanoid

motion agrees as much as possible with the directive, while maintaining stability and robustness to disturbances. In particular, a major goal of this work is to demonstrate real-world performance of the MHC on the Digit V3 humanoid robot.

4 Masked Humanoid Controller

Due to the lack of low-level supervisory information, we train the MHC via reinforcement learning (RL) in simulation and then transfer the resulting MHC to the real world. Below we describe our choices for the MHC network architecture. Next we describe the training approach, which includes the curriculum of training episodes and domain randomization choices to support sim-to-real transfer. Finally, we describe the reward function used to guide training.

Network Architecture. The MHC model is a Long Short-Term Memory (LSTM) recurrent neural network which takes as input the current humanoid state s_t and the next step of the current directive $d_i = (\hat{q}_i, I_i)$. The MHC first processes the directive with a single-layer feed-forward encoder to produce a 160 dimensional directive embedding. This is concatenated with s_t and fed into an LSTM block configured with two 64-unit recurrent layers. A final linear decoding layer outputs an offset for each actuated joint. The MHC action a_t is computed by adding this offset to the actuated joint values in $\hat{q}_i$, which yields the next PD setpoints. Note that for the masked joints in d_i, the corresponding values in $\hat{q}_i$ are zero and the offset corresponds to the actual PD setpoint. In our work, we use the Digit V3 humanoid, which has 20 actuated joints. The MHC is run at 50Hz to compute PD setpoints which are sent to a PD controller running at 2kHz.

Training Approach. The MHC is trained via the PPO RL algorithm [26] in simulation using the MuJoCo physics engine with a Digit V3 humanoid model. Below we describe the way we generate a motion dataset for creating directives, how we generated training episodes, the curriculum stages used for effective training, and domain randomization to facilitate sim-to-real transfer.

Motion Data Generation. We create a diverse set of reference motions for training the MHC using a combined dataset consisting of human motion capture (MoCap) datasets (AMASS [27], Reallusion [28]), and video demonstrations. To help bridge the human to Digit embodiment gap, we employ an inverse kinematics (IK) based retargeting procedure to map the dataset trajectories to Digit's kinematic model. For each frame in the dataset, we solve an IK problem for the generalized position vector of digit $q = (\theta, b)$ formulated as a nonlinear program (NLP). We use the IK module in Drake [cite] to set up the costs and constraints associated with the NLP, and SNOPT [cite] as the underlying solver. Critical kinematic feasibility requirements such as locking the stance foot to the ground and respecting the closed kinematic chain topology of Digit's legs are expressed as constraints in task space. Upper body motion targets that serve more of a stylistic purpose such torso pose, and hand positions are expressed using costs to aid in convergence. Any frames that the IK optimization did not generate a feasible solution for are simply dropped. Linear interpolation is applied to the sequence of states q_i and their timestamps t_i to produce the final time-parameterized trajectory $q_{1:H}$ used for training.

Episode Generation. Each episode is initialized with the robot in a standing position. Next a random command window length w is generated and a random directive d is drawn from a distribution defined by the curriculum stage. The MHC then cycles through d until reaching w steps. This sampling of a w and d continues until reaching the maximum episode length e, which depends on the curriculum stage, or the robot falls. By switching between multiple random directives during an episode, the MHC can learn to smoothly transition between different types of motion. In addition, to encourage robustness, during the execution of each episode we apply perturbations on the torso according to a distribution that depends on the curriculum stage.

Curriculum Stages. We employ a three-stage curriculum that progressively learns locomotion, robustness to perturbations, and whole-body motion tracking. This approach enables the model to gradually acquire complex skills while ensuring stable learning.

(Stage 1: Locomotion) The objective of this stage is to focus learning on developing basic balance and locomotion control, which is a core capability required for more complex behaviors. For this purpose, the episodes involve only randomized *locomotion directives*, which are partial directives that mask all variables except for those specifying root motion (v_i, w_i, b_i). Instantaneous torso perturbations (80-800N) are also introduced, each affecting a single policy step with a 1% probability. Training lasts for 300 policy steps (6 seconds) with a command window $w \in [40, 100]$.

(Stage 2: Stability) Building upon Stage 1, we focus on enhancing the controller's stability during locomotion. In this stage continuous torso perturbations (20-150N) are applied over windows of 20-50 policy steps. In order to facilitate recovery, we increase episode length to 800 policy steps (16 seconds) and the command window length to $w \in [100, 400]$.

(Stage 3: Whole-Body Directives) In this stage, we train the MHC to follow both fully and partially specified directives. Fully-specified directives provide the complete motion reference without any masking, requiring the MHC to perform whole-body imitation. Partially specified directives may take three forms, which correspond to key input modalities: 1) providing only locomotion commands without any upper or lower-body motion specification, corresponding to a joystick input modality, 2) specifying only the upper-body motion while standing in place, upper body imitation from VR or video modalities, or 3) specifying upper-body target motions with locomotion commands, corresponding to a richer VR controller with a locomotion interface. The MHC learns to generate coherent whole-body motions while satisfying the constraints specified by the unmasked components of the directive. Throughout this stage, we maintain the episode length at 800 policy steps and the command window length w within the range [100, 400].

Domain Randomization. To facilitate transfer from simulation to the real robot, we employ dynamics randomization throughout all stages of the curriculum. This involves randomizing various physical parameters of the simulated environment, such as joint damping, link masses, center of mass positions, encoder noise, ground friction, and terrain variations. By exposing the MHC to a wide range of dynamics during training, we aim to learn a policy that can generalize to the dynamics of the real robot without requiring precise system identification or extensive real-world fine-tuning. The randomization ranges are carefully chosen to strike a balance between creating sufficiently diverse training scenarios and maintaining realistic behavior of the simulated robot. Details of the specific randomization ranges used in this work can be found in the the supplementary material.

Reward Design. Our reward function must encourage stability and robustness during locomotion and standing as well as tracking of unmasked joint in the current directive. For locomotion and standing stability we use the reward function used to train a recent state-of-the-art humanoid standing and walking controller [13]. This reward function include three components: 1) *Task Reward* r_{task} to encourage command following for locomotion directives, 2) *Style Reward* r_{style} to encourage natural body postures for standing and walking, 3) *Regularization Reward* r_{reg} to encourage smooth joint motion and low torques. We refer the reader to the original paper for details of these components. To encourage accurate tracking, we include an additional *Tracking Reward* r_{track}. For a given time step let θ be the current joint values and $\hat{\theta}_i$ be the target joint for that step along with the corresponding directive mask I_i. The corresponding tracking reward of step t is

$$r_{track} = \exp\left(-1.5 \cdot ||(\theta - \hat{\theta}_i) \cdot I_i||\right),$$

which decreases with the L2 distance between the target and actual unmasked joints.

The overall reward function depends on whether the directive masks the lower-body joints or not. When the lower-body is masked, e.g. for pure locomotion commands or locomotion combined with upper body tracking, the reward function is $r = r_{task} + r_{style} + r_{reg} + r_{track}$. However, when the lower-body joints are not masked in the directive, the reward function removes the style reward and becomes $r = r_{task} + r_{reg} + r_{track}$. This is done because the motion preferences of the style reward may conflict with the specific lower-body target directives that should be tracked.

5 Experimental Results

In this section, we present our evaluation protocol and simulation experiments to assess the MHC's performance in executing diverse behaviors from full and partially-specified target motions. We compare the MHC with baseline approaches and analyze the impact of training dataset diversity as well as learning curriculum. Additionally, we discuss qualitative observations and the successful sim-to-real transfer of the MHC through real-world trials on the Digit V3 humanoid robot.

Motion Directive Dataset. We used IK retargetting to generate 75 kinematic motion trajectories selected from our motion sources as described in Section 4. We aimed to select a diverse set of motions, while avoiding aggressive motions that involve highly dynamic behavior such as jumping, kicking, and large sudden swinging of limbs. In order to test models on motions outside of their training sets, we divided the data into three sets, each containing a mix of motions from the original motion sources.

- *setA* [20 motions]: Amass Boxing (5), Amass misc (9), Reallusion (2), optimized (2), video (2)
- *setB* [20 motions]: Amass Boxing (6), Amass misc (6), Reallusion (5), optimized (3)
- *setC* [35 motions]: Amass Boxing (19), Amass misc (12), Reallusion (1), optimized (3)

While each set contains some basic motions related to hand waving and simple upper body movements, the general trend is an increasing level of difficulty as judged by the authors. We note that *setC* has a number of the most difficult motions, involving highly dynamic motion, such as tennis smashes and difficult boxing moves. The feasibility of these more difficult motions is unclear for our realistic Digit model.

Experiment Setup and Metrics. Each experiment involves evaluating an MHC controller on one of the retargeted motion datasets. For each motion we create 3 motion directives corresponding to a whole body directive, upper body command with lower body standing directive, upper body command with lower body locomotion directive. For each motion directive we evaluate the MHC over 200 episodes, each having a maximum time of 50 seconds (2500 steps). Each episode starts from a randomized point in the directive and then repeatedly cycles through the directive three times before resetting to another randomly selected starting point. Based on these episode runs we evauate the following metrics, which depend on the type of directive.

- *Failure Rate* (Fail %): Percentage of episodes that result in failure (i.e. falling) before 50 seconds.
- *Mean per joint positional error* ($\mathbb{E}_{\text{MPJPE}}$): Mean positional error for end effectors (hands, elbows, knees, feet) is calculated as the L2 norm of the current Euler position relative to the torso, compared to values from the directive. For partial directives, the error is computed only for unmasked joints. This is done for both partial and fully-specified directives before episode termination.
- *Root Drift* (Root_{Δ}): The mean of drift in root position during an episode from its commanded position when following partially or fully-specified directives. Computed as an L2 norm of current root position against expected root position in previous directive step.

Comparison to Baselines. There are no existing approach that handle the masked directive inputs of the MHC. Thus, we have developed two baselines to compare against: 1) *Locomotion+Offset*. This controller is trained on the first two curriculum stages and only sees the locomotion commands at its input. At inference time, it receives locomotion commands from the directives and for other

Model	Fail %	$\mathbb{E}_{\text{MPJPE}}$	Root_{Δ}
Locomotion+Offsets	49.213	0.156	0.168
LocomotionTrain	4.800	0.227	0.203
MHC	0.450	0.088	0.134

Table 1: Comparison of baseline models on various metrics, including failure rate, mean per-joint position error ($\mathbb{E}_{\text{MPJPE}}$), and root mean squared error ($\text{Root}\Delta$). The MHC model achieves the lowest failure rate and error metrics compared to the Locomotion+Offsets and LocomotionTrain models.

Metric	**Fail %**			$\mathbb{E}_{\text{MPJPE}}$			**Root**$_{\Delta}$		
Dataset	setA	setB	setC	setA	setB	setC	setA	setB	setC
MHC$_{\text{A}}$	0.450	0.588	7.250	0.088	0.436	0.589	0.134	0.461	0.245
MHC$_{\text{AB}}$	0.963	2.225	1.843	0.092	0.435	0.598	0.133	0.383	0.211
MHC$_{\text{ABC}}$	1.488	0.550	3.957	0.098	0.431	0.605	0.155	0.428	0.260

Table 2: Comparison of Models across Different Datasets. Note that higher failure rates tend to be biased toward smaller (better) values of the other metrics, since those metrics are only averaged over executions before failure.

unmasked joints directly uses the IK offsets in the target motions as the PD setpoints, 2) *LocomotionTrain*. This is the same as Locomotion+Offset during training and inference, but includes the IK offsets as part of the controller input. The motivation for LocomotionTrain is that by seeing the IK offsets that will be used during motion generation, the controller can adjust to account for the upcoming motion and be more robust.

We train the MHC and two baselines on the SetA data. The evaluation results are provided in Table 1. We see that the MHC outperforms the baselines across all metrics. Most significantly it achieves a failure rate of less than 1% which is significantly lower than the baselines. Interestingly, we see that LocomotionTrain achieves a significantly lower rate than Locomotion+Offsets, showing the utility having the controller learn to prepare for upcoming target motions.

Overall Performance for Varying Training Sets. We investigate the impact of training set diversity on the MHC's ability to generalize to unseen motion directives. We trained three MHC models: MHC$_{\text{A}}$ trained on *setA*, MHC$_{\text{AB}}$ trained on *setA* and *setB*, and MHC$_{\text{ABC}}$ trained on all three sets. Each model was trained until the learning curve saturated over a uniform distribution of trajectories from their respective training sets. We then evaluated these models on each dataset and report the individual performances in Table 2.

We first observe that MHC$_{\text{A}}$ shows strong performance on its training motions in *setA* and also generalizes well to the new motions in *setB*. However, it struggles to generalize to the much more difficult motions in *setC*, showing a much higher failure rate. This is not surprising given that the most difficult motions in *setC* are far from anything seen in the training set. MHC$_{\text{AB}}$ shows much stronger generalization to *setC*, indicating that the wider training set may have resulted in more diverse capabilities than MHC$_{\text{A}}$. However, surprisingly, it performs worse on *setB*, thought still maintaining a relatively low failure rate. Our current hypothesis is that the training run for MHC$_{\text{A}}$ may have been stopped prematurely at a point before certain motion types were fully mastered. MHC$_{\text{ABC}}$ shows respectable performance across the datasets, however, underperforms MHC$_{\text{AB}}$ on both *setA* and *setC*. This may be due to the challenge of learning with the most difficult motions, which may not be fully feasible. Training on such data is similar to training on noisy labels, since departures from the ground truth targets are necessary.

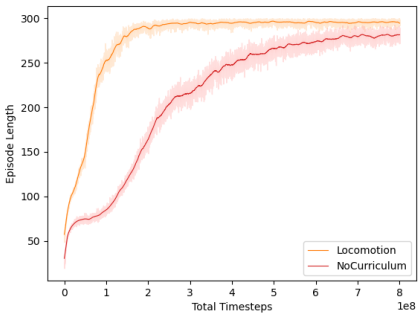

Figure 2: Comparison of curriculum stage 1 vs. training from scratch

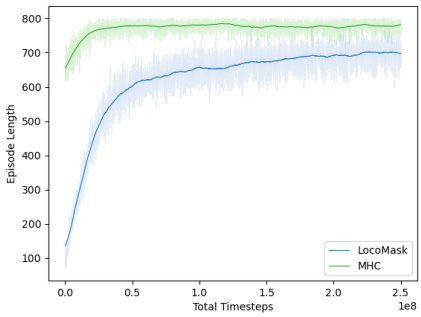

Figure 3: Comparison of MHC vs. training whole-body directives without stability stage

Figure 4: A) Locomotion Directives to follow joystick commands; B) Fully-specified directive for a boxing jab displaying whole body torso motion and feet coordination; C) Masked partial directives for upper body motion tracking and lower body locomotion commands to achieve the task of moving to a location, picking up the box, moving while holding the box and placing it at a location.

Overall, these results show that our training approach can generalize to motions outside of the training set. However, more investigation is needed to understand the significance of training and testing with more difficult motions. In particular, the exploration of much longer training runs and continually monitoring of individual motion performance is likely to provide useful insights.

Curriculum Results. To evaluate the effectiveness of our curriculum design, we conducted an ablation study comparing the MHC trained with our complete curriculum against two variants: (1) *NoCurriculum*, a controller trained without any curriculum, and (2) *LocoMask*, a controller trained without the stabilization stage. All controllers were trained on *setA*. The results, shown in Figure 2 and Figure 3, demonstrate the importance of our structured curriculum. The *NoCurriculum* controller took significantly longer to learn and did not reach the same performance level as the MHC in the initial locomotion stage (Figure 2). The *LocoMask* controller exhibited greater instability, resulting in shorter average episode lengths (Figure 3). These findings highlight that our staged curriculum, which progressively learns locomotion, stability, and whole-body motion tracking, is very important for achieving our observed robustness and stability.

Sim-to-Real Results. We conduct a real-world demonstration of the MHC using the Digit V3 humanoid robot. Our supplementary video illustrates a number of successful examples and some failures. Some examples of successes include tracking of fully specified boxing directives, partial directives such as robust walking using joystick commands, box pick and place with locomotion + upper body partial directives as well as non-traditional zombie walking gaits. Failure modes included dancing motions that involved balancing on one feet for a substantial amount of time and fully extended arm poses for box pick up motions. Here holding the box outwards, away from the torso caused imbalance leading to a continuous drift from standing posture in order to stabilize. Wheile there is significant room to improvement these sim-to-real results demonstrate that the MHC framework is a promising path toward highly robust and versatile whole-body humanoid control.

6 Limitations

One of the major current limitations is a significant sim-to-real gap. This is particularly prevalent for motions where the feet are placed widely or where one foot remains in the air for an extended period. Our hypothesis is that this sim-to-real limitation requires research that is more generally focused on sim-to-real transfer, possibly via the incorporation of real-world data. The MHC has been a useful vehicle for highlighting these gaps. Another limitation of our current training approach is that we only consider a limited set of masking patterns during training. Supporting fully general masking may lead to additional robustness and open up new possibilities for authoring behaviors. Finally, the MHC is currently does not account for interaction with external objects, such as the box, during a box pickup. Explicitly incorporating objects to be manipulated into the MHC input and training is an important direction for supporting practical applications.

References

[1] D. Crowley, J. Dao, H. Duan, K. Green, J. Hurst, and A. Fern. Optimizing bipedal locomotion for the 100m dash with comparison to human running. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12205–12211. IEEE, 2023.

[2] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid. Robust feedback motion policy design using reinforcement learning on a 3d digit bipedal robot. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5136–5143. IEEE, 2021.

[3] J. Dao, H. Duan, and A. Fern. Sim-to-real learning for humanoid box loco-manipulation. *arXiv preprint arXiv:2310.03191*, 2023.

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[5] X. Da, Z. Xie, D. Hoeller, B. Boots, A. Anandkumar, Y. Zhu, B. Babich, and A. Garg. Learning a contact-adaptive controller for robust, efficient legged locomotion. In *Conference on Robot Learning*, pages 883–894. PMLR, 2021.

[6] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter. Dynamic locomotion on slippery ground. *IEEE Robotics and Automation Letters*, 4(4):4170–4176, 2019.

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

[8] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 7309–7315. IEEE Press, 2021. doi:10.1109/ICRA48506.2021.9561814. URL https://doi.org/10.1109/ICRA48506.2021.9561814.

[9] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[10] J. Dao, K. Green, H. Duan, A. Fern, and J. Hurst. Sim-to-real learning for bipedal locomotion under unsensed dynamic loads. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10449–10455. IEEE, 2022.

[11] H. Duan, B. Pandit, M. S. Gadde, B. van Marum, J. Dao, C. Kim, and A. Fern. Learning vision-based bipedal locomotion for challenging terrain. *arXiv preprint arXiv:2309.14594*, 2023.

[12] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Learning humanoid locomotion with transformers. *arXiv preprint arXiv:2303.03381*, 2023.

[13] B. van Marum, A. Shrestha, H. Duan, P. Dugar, J. Dao, and A. Fern. Revisiting reward design and evaluation for robust humanoid standing and walking. *arXiv preprint arXiv:2404.19173*, 2024.

[14] J. Won, D. E. Gopinath, and J. K. Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics (TOG)*, 41:1 – 12, 2022. URL https://api.semanticscholar.org/CorpusID:250956798.

[15] J. Won, D. E. Gopinath, and J. K. Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)*, 39:33:1 – 33:12, 2020. URL https://api.semanticscholar.org/CorpusID:219569865.

[16] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu. Perpetual humanoid control for real-time simulated avatars. URL http://arxiv.org/abs/2305.06456.

[17] J. Merel, L. Hasenclever, A. Galashov, A. Ahuja, V. Pham, G. Wayne, Y. W. Teh, and N. M. O. Heess. Neural probabilistic motor primitives for humanoid control. *ArXiv*, abs/1811.11711, 2018. URL https://api.semanticscholar.org/CorpusID:53831933.

[18] Z. Dou, X. Chen, Q. Fan, T. Komura, and W. Wang. C·ase: Learning conditional adversarial skill embeddings for physics-based characters. *ArXiv*, abs/2309.11351, 2023. URL https://api.semanticscholar.org/CorpusID:262064161.

[19] N. Wagener, A. Kolobov, F. V. Frujeri, R. Loynd, C.-A. Cheng, and M. Hausknecht. MoCapAct: A multi-task dataset for simulated humanoid control. URL http://arxiv.org/abs/2208.07363.

[20] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[21] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng. Calm: Conditional adversarial latent models for directable virtual characters. *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. URL https://api.semanticscholar.org/CorpusID:258461220.

[22] J. C. Cerón, M. S. H. Sunny, B. Brahmi, L. M. Mendez, R. Fareh, H. U. Ahmed, and M. H. Rahman. A novel multi-modal teleoperation of a humanoid assistive robot with real-time motion mimic. *Micromachines*, 14, 2023. URL https://api.semanticscholar.org/CorpusID:256964199.

[23] Y. Du, R. Kips, A. Pumarola, S. Starke, A. K. Thabet, and A. Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 481–490, 2023. URL https://api.semanticscholar.org/CorpusID:258187221.

[24] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*, 2024.

[25] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[27] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5441–5450, 2019.

[28] Reallusion. 3d animation and 2d cartoons made simple. URL http://www.reallusion.com.

A Dynamics Randomization

All models in our work were trained with a fixed dynamics randomization setup in order to transfer from simulation to real without learning the exact dynamics of the system. The ranges for all elements are given in Table 3.

Element	**Range(%)**
Joint Damping	[-50, 250]
Body Mass	[-25, 25]
CoM Position	[-5, 5]
Spring Stiffness	[-10, 10]
Friction	[-20, 20]
	Range(radians)
Encoder Noise	[-0.05, 0.05]
Ground Slope	[-0.05, 0.05]

Table 3: Dynamics Randomization Ranges.

B Implementation Details

MHC has been trained using actor critic based PPO algorithm. Table 4 specifies the hyperparameters used for each training cycle independent of the curriculum.

Hyperparameter	**Value**
Learning Rate (Actor & Critic)	3e-4
Buffer Size	50000
Number of Epochs	5
Batch Size (of episodes)	8
PPO Log-Clip Range	0.2
Maximum Gradient Norm	0.05
Discount Factor (Gamma)	0.95
GAE Lambda	0.95

Table 4: PPO Hyperparameters

C Reward structure

Table 5 gives a detailed view of our complete reward function. Note that style rewards are only ignored when following a fully-specified directive denoted by $I_i == 1$.

Following notation has been used:

- i indicates current time step
- $I_i == 1$ indicated if all joints are being mimicked and its a whole body motion directive;
- v_{xy} = base linear velocity;
- s = current robot parameter;
- $\boldsymbol{b}$ = base acceleration;
- c_{feet} = default foot position xyz and orientation rpy;
- $qd(\cdot)$ = quaternion distance function;

- n_c = number of feet in contact with ground;
- $\mathbb{1}_{td}$ = boolean variable indicating a foot touchdown in the current timestep;

	Reward Term	Definition		Weight	Active if $I_i == 1$
		if $v_i == 0$ [Standing]	**else** [Walking]		
Task	x, y velocity	$\exp\left(-5 \cdot (v_{xy})\right)$	$\exp\left(-5 \cdot (v_{xy} - v_i)^2\right)$	0.15	✓
	Tracking Reward	$\exp\left(-1.5 \cdot \|\|(\theta - \hat{\theta}_i) \cdot I_i\|\|\right)$		0.2	✓
Task, Style	Yaw orient	$\exp\left(-300 \cdot qd(\omega_z, w_i)\right)$		0.1	✓
	Roll, pitch orient	$\exp\left(-30 \cdot qd\big(\omega_{xy}, (b_i[xy])\big)\right)$		0.2	✓
	Base height	$\exp\left(-20 \cdot \left\|s_z - (b_i[z])\right\|\right)$		0.05	✓
Style	Feet orientation	$\exp\left(-\sum \left\|s_{\text{feet, rpy}} - c_{\text{feet, rpy}}\right\|\right)$		0.05	×
	Feet position	$\exp\left(-3 \cdot \left\|S_{\text{feet, xyz}} - c_{\text{feet, xyz}}\right\|\right)$	1	0.05	×
	Feet airtime	1	$\sum_{f \in (l,r)} (t_{air,f} - 0.4) * \mathbb{1}_{td,f}$	1.0	×
	Feet contact	1	$\begin{cases} 1 & \text{if } n_{c,t^*} = 1 \text{ for any } t^* \in [t-0.2, t] \\ 0 & \text{else} \end{cases}$	0.1	×
	Arm	$\exp\left(-3 \cdot \left\|\left|(\theta[\hat{J}] \text{ where } \hat{J} \in \text{ arms})\right|\right|\right)$		0.03	×
Reg	Base acceleration	$\exp\left(-0.01 \cdot \sum \left\|b_{xyz}\right\|\right)$		0.1	✓
	Action difference	$\exp\left(-0.02 \cdot \sum \left\|a_t - a_{t-1}\right\|\right)$		0.02	✓
	Torque	$\exp\left(-0.02 \cdot \frac{1}{N} \sum \left\|t_{\text{motor}}\right\| / t_{\text{max}}\right)$		0.02	✓

Table 5: Reward Terms