HOVER: Versatile Neural Whole-Body Controller for Humanoid Robots

Tairan He[*1,2]   Wenli Xiao[*1,2]   Toru Lin[1,3]   Zhengyi Luo[1,2]   Zhenjia Xu[1]   Zhenyu Jiang[1,4]
Jan Kautz[1]   Changliu Liu[2]   Guanya Shi[2]   Xiaolong Wang[1,5]   Linxi "Jim" Fan[†1]   Yuke Zhu[†1,4]

Abstract— Humanoid whole-body control requires adapting to diverse tasks such as navigation, loco-manipulation, and tabletop manipulation, each demanding a different mode of control. For example, navigation relies on root velocity or position tracking, while tabletop manipulation prioritizes upper-body joint angle tracking. Existing approaches typically train individual policies tailored to a specific command space, limiting their transferability across modes. We present the key insight that full-body kinematic motion imitation can serve as a common abstraction for all these tasks and provide general-purpose motor skills for learning multiple modes of whole-body control. Building on this, we propose HOVER (Humanoid Versatile Controller), a multi-mode policy distillation framework that consolidates diverse control modes into a unified policy. HOVER enables seamless transitions between control modes while preserving the distinct advantages of each, offering a robust and scalable solution for humanoid control across a wide range of modes. By eliminating the need for policy retraining for each control mode, our approach improves efficiency and flexibility for future humanoid applications.

I. INTRODUCTION

Humanoid is a versatile form factor that supports a wide variety of robotic tasks and applications, including bimanual manipulation [1–3], bipedal locomotion [4–7], and agile whole-body control [8–14]. While showing impressive results, each of these efforts uses a different formulation for whole-body control based on the need for their specific task and scenario. Some use root velocity tracking [5, 6] to support locomotion, some choose joint angle tracking [12, 13] to enable expressive movements, and others use kinematic tracking of selected body keypoints [9, 10] to support teleoperation. Although these approaches are similar in terms of the end goal of motion tracking, they require task-specific controller interface and rewards design. This not only makes the development process repetitive and time-consuming, but also limits the versatility of the resultant whole-body controller. For instance, a robot performing bipedal locomotion on uneven terrain using root velocity tracking [5, 6] would struggle to seamlessly switch to a task requiring precise bimanual manipulation, where joint angle or end-effector tracking [2, 12, 13] might be necessary. These task-specific dependencies limit versatility, as each controller is restricted to a single mode of control. In addition to motion tracking, many pretrained manipulation policies [15, 16] require operating in different configuration spaces, such as joint angles and end-effector positions. This variability highlights

*Equal Contributions, †GEAR Team Leads
[1]NVIDIA, [2]CMU, [3]UC Berkeley, [4]UT Austin, [5]UC San Diego
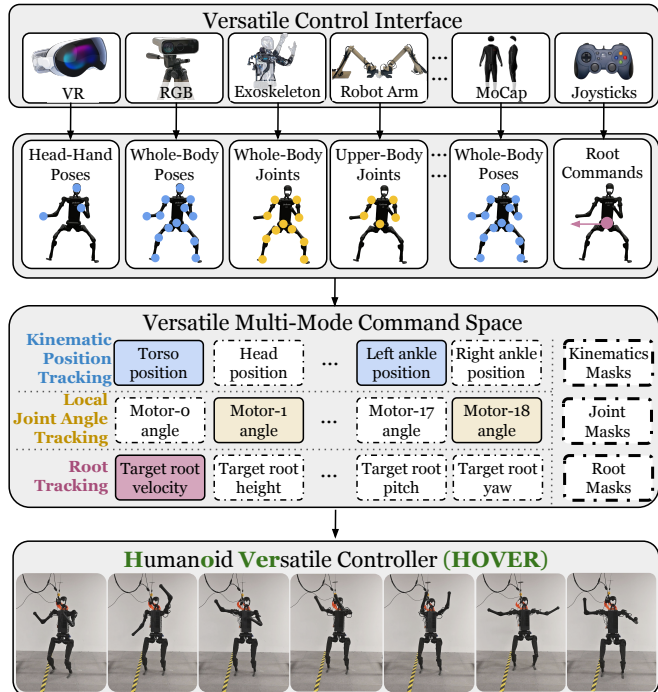Paper website: https://hover-versatile-humanoid.github.io

Fig. 1: HOVER enables versatile humanoid control with a unified multi-mode command space. The versatile multi-mode command space supports **kinematic position tracking (blue)**, **local joint angle tracking (yellow)**, and **root tracking (purple)**. Highlighted boxes indicate active commands being tracked, while the masks (dashed boxes on the right) allow selective activation of different command spaces to accommodate various tasks.

the need for a unified low-level humanoid controller capable of adapting to diverse control mode configurations.

Since all these modes are applied to a shared hardware platform, a natural question arises: *Can we create a unified controller that supports all control modes, combining the strengths of each?* This is a non-trivial challenge, as each mode operates within a distinct command space, making direct integration impractical. However, despite differences in control interfaces, the underlying motion objectives often align: stable, human-like motion for humanoid control.

To this end, we present HOVER, a unified neural controller for humanoid whole-body control that supports diverse control modes as shown in Figure 1, including over 15 useful modes for real-world applications to a 19-DOF humanoid robot. This versatile command space covers most modes used in prior works [9, 10, 12, 13]. To ensure a robust foundation of motor skills that generalize well across tasks, we train an oracle motion imitator to mimic large-scale human motion data from MoCap [17], covering a wide variety
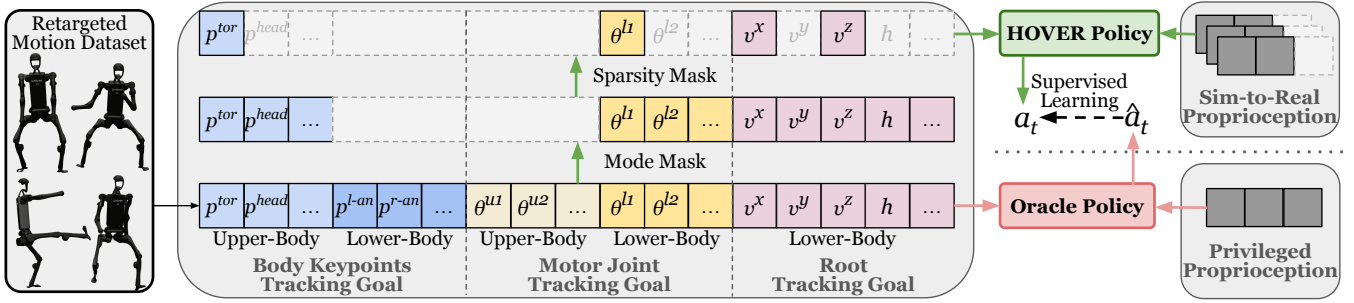
Fig. 2: **Overview of HOVER distillation process.** The **HOVER policy** is distilled from the **Oracle policy** through proprioception and command masking. The task commands for the student are determined via *mode-specific* and *sparsity-based* masks, applied to both upper and lower body motions independently. These masks generate diverse task command modes, refining the student's inputs. The distillation employs DAgger to align the student's actions with those of the oracle, optimizing through supervised learning on the oracle's actions.

of movements and control objectives. This design choice leverages the inherent adaptability and natural efficiency of human movements, providing the policy with rich motor priors that can be reused across multiple control modes. By grounding the training process in human-like motion, the policy gains a deeper understanding of balance, coordination, and motion control, which are crucial for effective whole-body humanoid behavior. Through a policy distillation process, we transfer these motor skills from the oracle policy into a single "generalist policy" capable of handling multiple control modes. The resulting multi-mode policy not only supports diverse control modes but also *outperforms* policies trained individually for each mode as shown in Figure 3. We hypothesize that this is due to the policy leveraging shared physical knowledge across modes, such as maintaining balance, human-like motion, and precise limb control. These shared skills enhance generalization, leading to better performance across all modes. In contrast, single-mode policies often overfit to specific reward structures and training environments, limiting their adaptability. Our multi-mode generalist policy also enables seamless transitions between modes, making it both robust and versatile.

To summarize, our contributions are threefold: 1) we present HOVER, a unified neural controller for humanoid whole-body control supporting multiple control modes; 2) we show that, through policy distillation, HOVER effectively shares motor skills across modes and outperforms individually trained policies; and 3) experiments in both simulation and on a real humanoid robot demonstrate that HOVER achieves seamless transitions between modes and delivers superior multi-mode control compared to other baselines.

II. METHOD

A. Goal-Conditioned RL for Humanoid Control

We formulate our problem as a goal-conditioned reinforcement learning (RL) task, where the policy π is trained to track real-time human motion. The state $\boldsymbol{s}_t$ comprises both the agent's proprioception $\boldsymbol{s}_t^{\mathrm{p}}$ and the target goal state $\boldsymbol{s}_t^{\mathrm{g}}$. The goal state $\boldsymbol{s}_t^{\mathrm{g}}$ provides a unified representation of the target motion goal, which we will describe in detail in Section II-B. Using the agent's proprioception $\boldsymbol{s}_t^{\mathrm{p}}$ and the goal state $\boldsymbol{s}_t^{\mathrm{g}}$, we define the reward $r_t = \mathcal{R}\left(\boldsymbol{s}_t^{\mathrm{p}}, \boldsymbol{s}_t^{\mathrm{g}}\right)$ for policy optimization. The action $\boldsymbol{a}_t \in \mathbb{R}^{19}$ represents the target joint
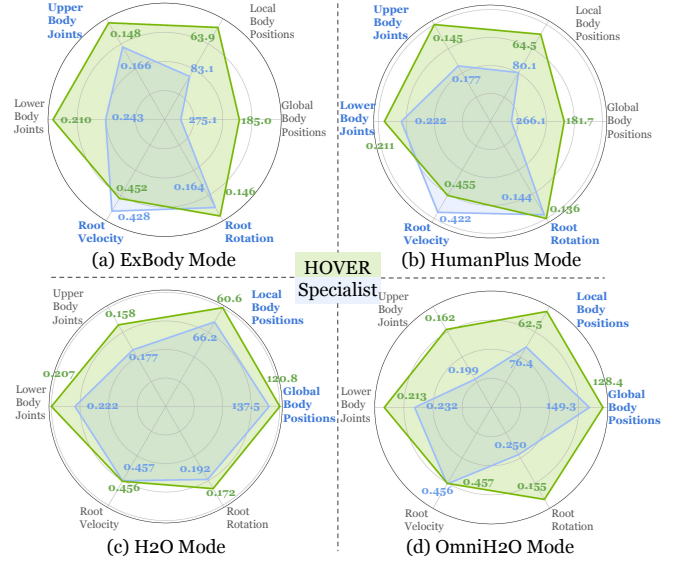

Fig. 3: Comparison between **prior work specialists (blue)** and our **generalist policy (green)** under corresponding modes. The metrics used are: upper/lower joint error (rad), global/local body position error (mm), root velocity error (m/s), and root rotation error (rad). These metrics evaluate how accurately each policy tracks reference motions and joint configurations across different control modes.

positions, which are fed into the PD controller to actuate the robot's degrees of freedom. We employ the proximal policy optimization (PPO) algorithm [18] to maximize the cumulative discounted reward $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$. This setup is framed as a command-tracking task, where the humanoid learns to follow the target commands at each timestep.

B. Command Space Design for Humanoid Control

In legged locomotion, root velocity [19] or position tracking [20] is a commonly employed command space. However, focusing solely on root tracking imposes limitations on the full capabilities of humanoid robots, especially for whole-body loco-manipulation tasks. We observe that while prior works [9, 10, 12, 13] have introduced control modes with varying advantages and disadvantages, each is typically tailored to specific subsets of tasks, thus lacking the flexibility required for general-purpose humanoid control. In contrast, our goal is to design a comprehensive control framework that accommodates a wide range of scenarios and is adaptable to

TABLE I: Command space for priors works on whole-body humanoid control. HOVER covers all the command designs designed by prior works with a unified command space, and support multi-mode control by tracking arbitrary subsets of the command elements.

Controller	Multi-Mode Control	Upper-Body Command		Lower-Body Command		
		Kinematic Position	Joint Angle	Kinematic Position	Joint Angle	Root
ExBody [12]	✗	✗	✓	✗	✗	✓
H2O [10]	✗	✓	✗	✓	✗	✗
OmniH2O [9]	✗	✓	✗	✗	✗	✗
HumanPlus [13]	✗	✗	✓	✗	✓	✓
HOVER (ours)	✓	✓	✓	✓	✓	✓

various humanoid tasks. To achieve this, the command space must be constructed to satisfy the following key criteria:

- **Generality**: The command space should encompass most existing configurations, allowing a general-purpose controller to replace task-specific controllers without sacrificing performance or versatility. And the space should be sufficiently expressive to interface with real-world control devices, including joysticks, keyboards, motion capture systems, exoskeletons, and virtual reality (VR) headsets as shown in Figure 1.
- **Atomicity**: The command space should be composed of independent dimensions, enabling arbitrary combinations of control options to support various modes.

Based on these criteria, we define a unified command space for humanoid whole-body control. This space consists of two primary control regions—upper-body and lower-body control—and incorporates three distinct control modes:

- **Kinematic Position Tracking**: target 3D positions of key rigid body points on the robot.
- **Local Joint Angle Tracking**: target joint angles for each robot motor.
- **Root Tracking**: target root velocity, height, and orientation, specified by roll, pitch, and yaw angles.

In our framework, as shown in Figure 1, a one-hot masking vector is introduced to specify which components of the command space are activated for tracking. Recent work on learning-based humanoid whole-body control [9, 10, 12, 13], as shown in Table I can be viewed as subsets of our unified command space, each representing specific configurations.

C. Motion Retargeting

Recent works has shown the advantage of learning robust whole-body control for humanoid robots from large motion datasets [9, 10, 12, 13]. The retargeting procedure from human motion dataset [17] to humanoid motion dataset has three steps: **Step-1**: We first compute the keypoints positions of the humanoid using forward kinematics, mapping its joint configurations to workspace coordinates. **Step-2**: Next, we fit the SMPL model to match the humanoid's kinematics by optimizing the SMPL parameters to align with the computed keypoints from forward kinematics. **Step-3**: Finally, the AMASS dataset is retargeted by matching corresponding keypoints between the fitted SMPL model and the humanoid with gradient descent. We follow the same motion retargeting and "sim-to-data" procedure with [10] to convert the large-scale human motion dataset [17] to dataset $\hat{Q}$ that only contains feasible motions for humanoids.

TABLE II: Reward designs for the oracle policy.

Term	Weight	Term	Weight
		Penalty	
Torque limits	-2	DoF position limits	$-1.25e^2$
Termination	$-2.5e^2$	DoF velocity limits	$-5e^1$
		Regularization	
DoF acceleration	$-1.1e^{-5}$	DoF velocity	$-4e^{-3}$
Lower Action rate	-3	Upper Action rate	$-6.25e^{-1}$
Torque	$-1e^{-4}$	Feet orientation	$-6.25e^1$
Feet air time	$1e^3$	Feet contact force	$-7.5e^{-1}$
Stumble	$-1.25e^3$	Slippage	$-7.5e^1$
In the air	$-2e^2$	Max feet height per step	$-3e^3$
		Task Reward	
DoF position	$3.2e^1$	DoF velocity	$1.6e^1$
Body position	$8e^1$	Body rotation	$2e^1$
Body velocity	8	Body angular velocity	8
Root velocity	$1e^2$	Root rotation	$2e^1$

D. Oracle Policy Training from Large-Scale Human Motions

State Space Design. We train an oracle motion imitator $\pi^{\text{oracle}}(a_t | s_t^{\text{p-oracle}}, s_t^{\text{g-oracle}})$. The proprioception is defined as $s_t^{\text{p-oracle}} \triangleq [\boldsymbol{p}_t, \boldsymbol{\theta}_t, \dot{p}_t, \boldsymbol{\omega}_t, \boldsymbol{a}_{t-1}]$, which contains the humanoid rigid-body position $\boldsymbol{p}_t$, orientation $\boldsymbol{\theta}_t$, linear velocity $\dot{p}_t$, angular velocity $\boldsymbol{\omega}_t$, and the previous action $\boldsymbol{a}_{t-1}$. The goal state is defined as $s_t^{\text{g-oracle}} \triangleq [\hat{\boldsymbol{\theta}}_{t+1} \ominus \boldsymbol{\theta}_t, \hat{\boldsymbol{p}}_{t+1} - \boldsymbol{p}_t, \hat{\boldsymbol{v}}_{t+1} - \boldsymbol{v}_t, \hat{\boldsymbol{\omega}}_{t+1} - \boldsymbol{\omega}_t, \hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{p}}_t]$, which contains the reference pose $(\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{p}}_t)$ and one-frame difference between the reference and current state for all rigid bodies of the humanoid. We use the same policy network structure with [9], a three-layer MLP with layer dimensions of $[512, 256, 128]$.

Reward Design and Domain Randomizations. We formulate the reward r_t as the sum of three components: 1) penalty, 2) regularization, and 3) task rewards, detailed in Table II. We follow the same domain randomization in [9] to randomize the physical parameters of the simulated environment and humanoids for successful sim-to-real transfer.

E. Multi-Mode Versatile Controller via Distillation

Proprioception. For the student policy $\pi^{\text{student}}(s_t^{\text{p-student}}, s_t^{\text{g-student}})$ distilled from the oracle teacher π^{oracle}, the proprioception is defined as $s_t^{\text{p-student}} \triangleq [q, \dot{q}, \omega^{\text{base}}, g]_{t-25:t} \cup [a_{t-25:t-1}]$, where q is the joint position, $\dot{q}$ is the joint velocity, ω^{base} is the base angular velocity, g is the gravity vector, and a is the action history. Following [9], we stack these terms over the last 25 steps to represent the student's proprioceptive input.

Command Mask. As illustrated in Figure 2, the task command input for the student policy is defined using mode-based and sparsity-based masking. Specifically, the student's task command input, $s_t^{\text{g-student}}$, is represented as $s_t^{\text{g-student}} \triangleq M_{\text{sparsity}} \odot [M_{\text{mode}} \odot s_t^{\text{g-upper}}, M_{\text{mode}} \odot s_t^{\text{g-lower}}]$. The mode mask, M_{mode}, selects a specific task command mode for

TABLE III: Simulation motion imitation evaluation of HOVER and baselines on dataset $\hat{Q}$. Metrics that are tracked by different modes are highlighted in corresponding colors. Results that are statistically significant are highlighted in bold across 5 random seeds.

Method	Survive ↑	Kinematic Position				Joint Angle		Root				
		$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓	$E_{\text{upper-j}}$ ↓	$E_{\text{lower-j}}$ ↓	$E_{\text{root-vel}}$ ↓	$E_{\text{root-h}}$ ↓	$E_{\text{root-r}}$ ↓	$E_{\text{root-p}}$ ↓	$E_{\text{root-y}}$ ↓
Oracle policy	99.3%±0.203	119±0.442	59.4±0.234	2.63±0.008	5.43±0.024	0.153±0.001	0.206±0.001	0.456±0.002	0.066±0.001	0.065±0.001	0.083±0.001	0.282±0.002
ExBody Mode - Upper: joint angle tracking, Lower: root tracking												
ExBody (Specialist)	99.1%±0.212	275±1.650	83.1±0.499	**2.63±0.007**	6.31±0.034	0.166±0.002	0.243±0.003	**0.428±0.007**	0.074±0.001	0.070±0.001	0.147±0.002	0.276±0.003
HOVER (Ours)	99.1%±0.220	**185±1.110**	**63.9±0.384**	3.01±0.009	**6.06±0.033**	**0.148±0.002**	**0.210±0.004**	0.452±0.006	**0.063±0.001**	0.068±0.001	**0.091±0.001**	0.279±0.002
HumanPlus Mode - Upper: joint angle tracking, Lower: joint angle tracking, root tracking												
HumanPlus (Specialist)	98.4%±0.259	266±1.597	80.1±0.481	**2.53±0.007**	6.16±0.033	0.177±0.002	0.222±0.002	**0.422±0.006**	**0.061±0.001**	0.080±0.001	0.124±0.001	0.228±0.002
HOVER (Ours)	98.9%±0.285	**182±1.093**	**64.5±0.387**	2.85±0.008	**5.91±0.032**	**0.145±0.001**	**0.211±0.002**	0.455±0.007	0.067±0.001	**0.069±0.001**	**0.104±0.001**	0.237±0.002
H2O Mode - Upper: kinematic position tracking (left/right hand, left/right shoulder, left/right elbow), Lower: kinematic position tracking (left/right ankle)												
H2O (Specialist)	99.2%±0.233	137±0.827	66.3±0.398	**2.63±0.008**	5.75±0.028	0.177±0.003	0.221±0.002	0.457±0.004	0.078±0.001	0.067±0.001	0.095±0.001	0.415±0.003
HOVER (Ours)	98.9%±0.276	**121±0.726**	**60.6±0.361**	2.73±0.008	**5.49±0.028**	**0.158±0.002**	**0.207±0.002**	0.456±0.005	**0.065±0.001**	0.067±0.001	**0.086±0.001**	**0.365±0.003**
OmniH2O Mode - Upper: kinematic position tracking (head, left/right hand), Lower: N/A												
OmniH2O (Specialist)	99.0%±0.301	149±0.897	76.4±0.459	2.69±0.007	6.18±0.037	0.199±0.002	0.232±0.003	0.456±0.002	0.071±0.001	0.070±0.001	0.125±0.002	**0.306±0.002**
HOVER (Ours)	99.0%±0.297	**128±0.768**	**62.5±0.368**	2.69±0.008	**5.65±0.032**	**0.162±0.002**	**0.213±0.002**	0.457±0.004	**0.065±0.001**	0.068±0.001	**0.089±0.001**	0.310±0.002

the upper and lower body independently. For instance, the upper body may track kinematic positions, while the lower body focuses on joint angle and root tracking, as shown in Figure 2. After the mode-specific masking, the sparsity mask, M_{sparsity}, is applied. For example, in some scenarios, the upper body may track only the kinematic positions of the hands, while the lower body tracks only the joint angles of the torso. Every bit of the mode and sparsity binary mask is from a Bernoulli distribution $\mathcal{B}(0.5)$. Mode and sparsity masks are randomized at the episode beginning and remain fixed until the episode ends

Policy Distillation. We perform policy distillation using the DAgger framework [21]. For each episode, we roll out the student policy $\pi^{\text{student}}(\mathbf{a}_t | s_t^{\text{p-student}}, s_t^{\text{g-student}})$ in simulation to obtain trajectories of $(s_t^{\text{p-student}}, s_t^{\text{g-student}})$. At each timestep, we also compute the corresponding oracle states $(s_t^{\text{p-oracle}}, s_t^{\text{g-oracle}})$. Using these oracle states, we query the oracle teacher policy $\pi^{\text{oracle}}(\hat{\mathbf{a}}_t | s_t^{\text{p-oracle}}, s_t^{\text{g-oracle}})$ to obtain the reference action $\hat{\mathbf{a}}_t$. The student policy π^{student} is then updated by minimizing the loss function: $\mathcal{L} = \|\hat{\mathbf{a}}_t - \mathbf{a}_t\|_2^2$, where $\hat{\mathbf{a}}_t$ is the reference action from the oracle, and $\mathbf{a}_t$ is the action taken by the student policy.

III. EXPERIMENT

In this section, we present extensive experimental results in both IsaacGym [22] and the real-world Unitree H1 [23] robot to address the following questions:

- **Q1**: Can HOVER as a generalist policy outperform policies trained for a specific command configuration?
- **Q2**: Can HOVER outperform other methods of training a multi-mode humanoid controller?
- **Q3**: Can HOVER transfer to real-world hardware and execute versatile multi-mode control?

Experiment Setup. To answer these questions, we evaluate HOVER on motion tracking in both simulation (Section III-A and Section III-B) and real-world settings (Section III-C). In simulation, we evaluate using the retargeted AMASS dataset

$\hat{Q}$. In the real world, we test 20 standing motion sequences focusing on quantitative tracking and locomotion tasks for qualitative multi-mode control. Our real robot employs a 19-DOF Unitree H1 platform [23] with a total mass of around 51.5kg and a height of around 1.8m.

Baselines. To address **Q1** and **Q3**, we compare HOVER with several specialists. As shown in Table I, ExBody [12] focuses on tracking upper body joint angles and root velocity, HumanPlus [13] tracks whole-body joints and root velocity, H2O [10] tracks the kinematic positions of eight keypoints (shoulders, elbows, hands, ankles), and OmniH2O [9] tracks the kinematic positions of the head and both hands. We also compare other useful tracking modes (e.g., left-hand mode, right-hand mode, two-hand mode, head-only mode). For each control mode, we provide only the relevant observation input to the controller and train the specialist baseline with RL. For instance, in left-hand-only mode, only reference motion of the left hand is provided. To address **Q2**, we compare with another multi-mode RL policy, which follows the same masking process on the goal commands, but trains the baseline with RL objective from scratch. During the multi-mode RL baseline training, mode and sparsity are randomized at the beginning of each episode and remain fixed until the episode ends, which is same as the randomized masking process during distillation.

Metrics. We report survival rate, where the episode terminates if the humanoid hits the ground, not by feet. We calculate tracking error in terms of kinematic pose, joint angles, and root twist and rotations. The mean values of the metrics are computed across all motion sequences from dataset $\hat{Q}$. We evaluate policy's ability to imitate the reference motion by compare the tracking error of the global body position $E_{g-\text{mpjpe}}$ (mm), the root-relative mean per-joint (MPJPE) E_{mpjpe} (mm), joint tracking error E_j (rad), root velocity $E_{\text{root-vel}}$ (m/s), and root orientation tracking error $E_{\text{root-rpy}}$ (rad). To show physical realism, we report average joint acceleration E_{acc} (mm/frame^2), and velocity

TABLE IV: Comparison between HOVER and specialists. We only report tracking metrics that are tracked by this mode.

Method	$E_{\text{g-mpjpe-mode}}\downarrow$	$E_{\text{mpjpe-mode}}\downarrow$	$E_{\text{acc-mode}}\downarrow$	$E_{\text{vel-mode}}\downarrow$
Left Hand Mode - Upper: kinematic position tracking (left hand), Lower: N/A				
Specialist	$189\pm_{1.526}$	$\mathbf{147}\pm_{\mathbf{1.324}}$	$5.82\pm_{0.029}$	$11.2\pm_{0.089}$
HOVER (Ours)	$\mathbf{138}\pm_{\mathbf{1.025}}$	$151\pm_{0.934}$	$\mathbf{5.45}\pm_{\mathbf{0.031}}$	$\mathbf{10.3}\pm_{\mathbf{0.104}}$
Right Hand Mode - Upper: kinematic position tracking (right hand), Lower: N/A				
Specialist	$220\pm_{1.345}$	$216\pm_{1.451}$	$6.77\pm_{0.051}$	$12.5\pm_{0.152}$
HOVER (Ours)	$\mathbf{128}\pm_{\mathbf{0.774}}$	$\mathbf{141}\pm_{\mathbf{0.821}}$	$\mathbf{5.83}\pm_{\mathbf{0.049}}$	$\mathbf{10.8}\pm_{\mathbf{0.129}}$
2 Hands Mode - Upper: kinematic position tracking (left-right hands), Lower: N/A				
Specialist	$137\pm_{0.998}$	$145\pm_{1.010}$	$5.72\pm_{0.037}$	$11.2\pm_{0.004}$
HOVER (Ours)	$\mathbf{120}\pm_{\mathbf{0.901}}$	$\mathbf{119}\pm_{\mathbf{0.827}}$	$\mathbf{5.60}\pm_{\mathbf{0.045}}$	$\mathbf{10.1}\pm_{\mathbf{0.134}}$
Head Mode - Upper: kinematic position tracking (robot head), Lower: N/A				
Specialist	$186\pm_{1.149}$	$104\pm_{0.814}$	$\mathbf{2.22}\pm_{\mathbf{0.008}}$	$6.63\pm_{0.065}$
HOVER (Ours)	$\mathbf{133}\pm_{\mathbf{0.849}}$	$\mathbf{80.0}\pm_{\mathbf{0.711}}$	$2.31\pm_{0.011}$	$\mathbf{6.40}\pm_{\mathbf{0.029}}$

E_{vel} (mm/frame) error. To better show the correspondence between control modes and metrics, we highlight the metrics that are actively tracked by each mode with corresponding colors in Table III, Table IV and Table V. For instance, in Table III, upper joint and root metrics are colored with corresponding mode for ExBody mode.

A. Comparison with Specialists

Comparison with Specialists of Prior Work's Control Mode. To address **Q1** (*Can HOVER as a generalist policy outperform policies trained for a specific command configuration?*), we compare the performance of the same HOVER policy across different control modes against corresponding specialist policies. For example, the performance of HOVER under ExBody mode is evaluated with a fixed mask to match ExBody mode across the entire dataset $\hat{Q}$. As shown in Table III and Figure 3, HOVER consistently demonstrates superior generalization. In every command mode, HOVER outperforms prior work specialist controllers in at least 7 out of the 12 metrics, as highlighted by the bold values in Table III. This consistent advantage across various control modes underscores the versatility of HOVER. Furthermore, this means that even when focusing on a single control mode without considering multi-mode versatility, distilling from an oracle policy still surpasses RL-trained specialists.

Comparison with Other Specialists of Other Useful Control Mode. In addition to the aforementioned baselines, we also evaluate four additional modes: left-hand mode, right-hand mode, two-hand mode, and head mode. We train four RL specialists to track these modes individually. The results in Table IV show that HOVER consistently outperforms specialists in terms of tracking metrics that are trained for specific command configurations.

B. Comparison with Other Generalist Training Methods.

To address **Q2** (*does HOVER outperform other methods of training an multi-mode humanoid controller?*), we compare HOVER with a multi-mode RL baseline that follows the same masking process on the commands but trains with RL objective from scratch. In Figure 4, we assess tracking error across four metrics: root orientation, upper joint angle,
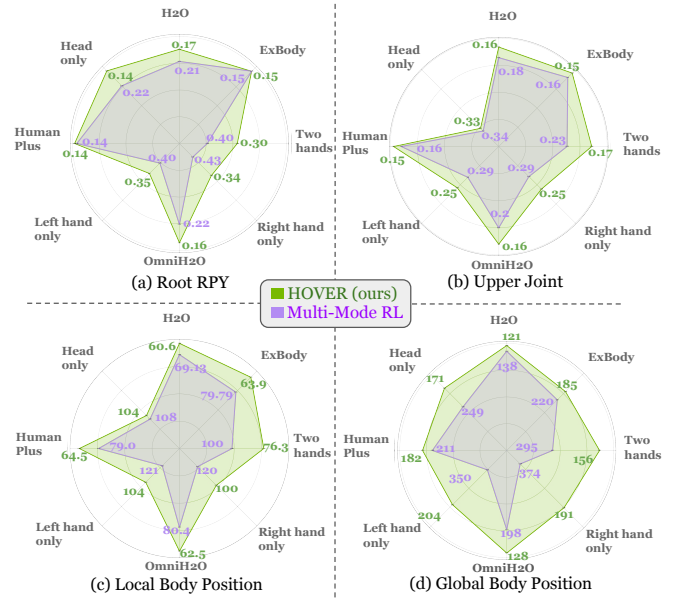

Fig. 4: We assess the tracking accuracy of two multi-mode control policies—**HOVER (green)** and **Multi-Mode RL (purple)**—across eight distinct humanoid control modes. The comparison is visualized across four key performance metrics in the radar charts.
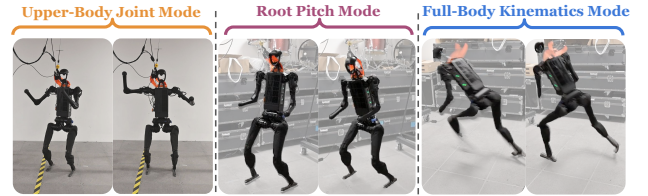

Fig. 5: Real-World Evaluations on different control modes.

and local and global body positions, measured in eight different modes. We scale the tracking error via $\frac{E^{\max}-E^{(\cdot)}}{E^{\max}-E^{\min}}$ for visualization, where larger radar webs indicate better tracking performance. The results show that HOVER achieves consistently lower tracking error across 32/32 metrics and modes. This performance boost underscores the importance of distilling from an oracle policy that tracks full-body kinematics for learning a generalist whole-body controller.

C. Real-World Evaluation

To address **Q3** (*does HOVER transfer to real-world hardware and execute versatile multi-mode control?*), we conduct quantitative tracking experiments and locomotion tests for qualitative multi-mode control.

Standing Motion Evaluations. We evaluate HOVER's performance in the real world by tracking 20 different standing motions in $\hat{Q}$. Among these, two motions are visually illustrated in Figure 5 (left). The quantitative metrics presented in Table V demonstrate HOVER's outperforms specialist policies in 11 out of 12 metrics. Moreover, we demonstrate successful tracking of root pitch motion, as shown in the middle of Figure 5, and full-body kinematic tracking, as shown on the right of Figure 5, where the robot is capable of tracking highly dynamic running motions.

Multi-Mode Evaluations. We also evaluate HOVER's generalization to locomotion in Figure 6, where we abruptly

TABLE V: Real-world tracking evaluation on 20 standing motions between prior works specialists and our method. Results that are statistically significant are highlighted in bold across 5 tests.

Method	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	$E_{\text{upper-j}}$ ↓	$E_{\text{root-rpy}}$ ↓
ExBody Mode				
ExBody (Specialist)	51.3 $\pm_{0.279}$	39.3 $\pm_{0.214}$	0.131 $\pm_{0.001}$	0.036 $\pm_{0.001}$
HOVER (Ours)	**48.9** $\pm_{0.470}$	**36.8** $\pm_{0.201}$	**0.126** $\pm_{0.001}$	**0.032** $\pm_{0.001}$
HumanPlus Mode				
Specialist	51.0 $\pm_{0.275}$	36.7 $\pm_{0.202}$	0.128 $\pm_{0.001}$	**0.035** $\pm_{0.001}$
HOVER (Ours)	**47.4** $\pm_{0.359}$	**35.3** $\pm_{0.194}$	**0.121** $\pm_{0.001}$	0.038 $\pm_{0.001}$
OmniH2O Mode				
Specialist	51.2 $\pm_{0.497}$	42.1 $\pm_{0.233}$	0.153 $\pm_{0.002}$	0.040 $\pm_{0.001}$
HOVER (Ours)	**47.5** $\pm_{0.261}$	**41.0** $\pm_{0.227}$	**0.145** $\pm_{0.001}$	**0.037** $\pm_{0.001}$

switch command modes during operation to simulate real-life scenarios. HOVER successfully transitions from ExBody mode to H2O mode during forward walking in Figure 6(a), and from HumanPlus mode to OmniH2O mode while performing turning and backward walking, in Figure 6(b). Additionally, we conduct a real-test teleoperation demo with Vision Pro, randomly masking out the positions of the head and hands. For example, in the middle of Figure 6(c), the humanoid tracks only the human's head position, ignoring the waving hands in *head mode*. The results demonstrate that HOVER can smoothly track motions across different modes, showcasing its robustness for real-world scenarios (e.g., when there are occlusions in the reference motions).

IV. RELATED WORK

Humanoid Whole-Body Controller. Performing whole-body control on humanoid hardware is a long-standing challenge in robotics due to the complex structure of humanoid robots. Before the rise in popularity of learning-based controllers, classical humanoid controllers [24–33] often use a hierarchical model-based optimization to solve for the low-level torque or position commands sent to hardware motors, where actuator-level dynamics on single joints are abstracted to multi-joint [26] or whole-body [27] controllers. Learning-based controllers follow the same design pattern in spirit, where high-level inputs are translated into low-level motor commands via neural networks. The design of controller abstraction and task specification varies by user needs and applications [34–36]. Recent works on learning-based humanoid whole-body control [9, 10, 12, 13, 37] generally have three design patterns for humanoid whole-body controller: kinematic motion tracking [9, 10, 37], local joint angle tracking [1, 13], and root velocity tracking [5, 6]. Kinematic motion tracking means tracking the full-body kinematic motion for each rigid body on the humanoid's body, and is heavily inspired by the motion imitation literature in the graphics community [38–41]. Local joint angle tracking tracks the local joint angles of the humanoid, which can be considered a specialized case of kinematic motion imitation where global position information is discarded. Root velocity tracking only serves for locomotion ability, and is used for navigation and terrain traversal of humanoids
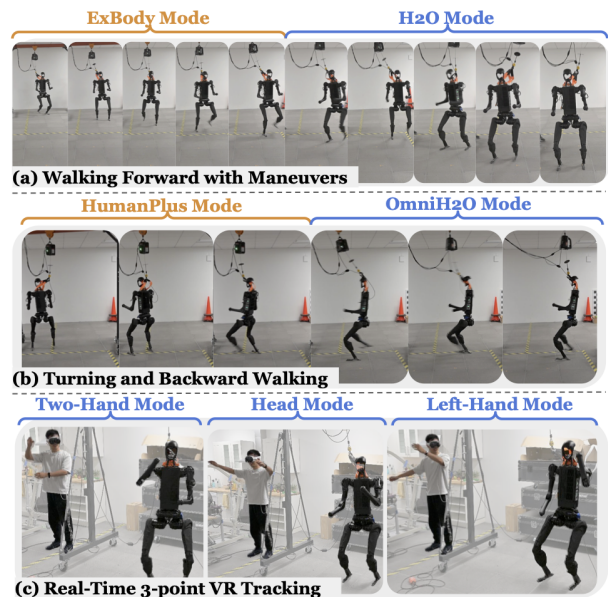

Fig. 6: HOVER shows robustness under control mode switches during locomotion and real-time teleoperation tests.

[5, 6]. One can also combine different control modes for upper and lower body: for instance, the upper body can be controlled with local joint angle tracking and the lower body with velocity tracking [12]. Even within the same kinematic tracking pattern, the sparsity design varies depending on the selected keypoints [9, 10]. So far, each of these control modes is independently developed and are not compatible with each other. In this work, we aim to unify all of these control modes.

Unified Neural Whole-Body Controller for Humanoid. MHC [42] learns multi-mode humanoid controller using RL from retargeted motion, but does not support arbitrary subset of chosen modes and is limited to local joint angles and root tracking. In computer graphics, MaskedMimic [43] enables multi-mode control with flexible kinematic tracking constraint by distillation. Other graphics works leverage reusable motion latent space for downstream flexible control modes [44–46]. However, additional policies need to be trained. Also, these efforts have been limited to using simplified humanoid morphologies that are not applicable in the real world. In this work, we aim to learn a unified control policy that can be directly used to control real humanoids using different control modes.

V. CONCLUSIONS

In this work, we introduced HOVER, a unified neural controller for humanoid whole-body control that supports diverse control modes. Through the use of a kinematic motion imitator and policy distillation, HOVER consolidates motor skills across multiple control modes into a unified policy that outperforms specialized controllers. Our evaluations collectively illustrate HOVER's ability to handle diverse real-world control modes, offering and superior performance compared to specialist policies. Future work will explore further developing automated mode-switching module for real-world applications.

REFERENCES

[1] 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.

[2] T. Lin *et al.*, "Learning visuotactile skills with two multifingered hands," *arXiv preprint arXiv:2404.16823*, 2024.

[3] J. Li *et al.*, "Okami: Teaching humanoid robots manipulation skills through single video imitation," in *8th Annual Conference on Robot Learning*, 2024.

[4] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, eadi9579, 2024.

[5] 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.

[6] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," *arXiv preprint arXiv:2406.10759*, 2024.

[7] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath, "Berkeley humanoid: A research platform for learning-based control," *arXiv preprint arXiv:2407.21781*, 2024.

[8] Boston Dynamics, *Atlas — partners in parkour*, https://www.youtube.com/watch?v=tF4DML7FIWk, 2021.

[9] T. He *et al.*, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024.

[10] T. He *et al.*, "Learning human-to-humanoid real-time whole-body teleoperation," *arXiv preprint arXiv:2403.04436*, 2024.

[11] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," *arXiv preprint arXiv:2409.15610*, 2024.

[12] 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.

[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] C. Zhang, W. Xiao, T. He, and G. Shi, "Wococo: Learning whole-body humanoid control with sequential contacts," *arXiv preprint arXiv:2406.06005*, 2024.

[15] A. Padalkar *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[16] M. J. Kim *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[17] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5442–5451.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[19] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024.

[20] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," *arXiv preprint arXiv:2401.17583*, 2024.

[21] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[22] V. Makoviychuk *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[23] Unitree, *Unitree's first universal humanoid robot*, 2023.

[24] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.

[25] D. Gouaillier *et al.*, "Mechatronic design of nao humanoid," in *2009 IEEE international conference on robotics and automation*, IEEE, 2009, pp. 769–774.

[26] N. A. Radford *et al.*, "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[27] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: System overview and integration," in *IEEE/RSJ international conference on intelligent robots and systems*, IEEE, vol. 3, 2002, pp. 2478–2483.

[28] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 2641–2648.

[29] Y. Ishiguro *et al.*, "Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit tablis," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6419–6426, 2020.

[30] S. Dafarra *et al.*, "Icub3 avatar system: Enabling remote fully immersive embodiment of humanoid robots," *Science Robotics*, vol. 9, no. 86, eadh3834, 2024.

[31] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2021, pp. 1–8.

[32] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba, "Humanoid motion generation system on hrp2-jsk for daily life environment," in *IEEE International Conference Mechatronics and Automation, 2005*, IEEE, vol. 4, 2005, pp. 1772–1777.

[33] S. Kuindersma *et al.*, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, pp. 429–455, 2016.

[34] K. Darvish *et al.*, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, 2023.

[35] K. Hauser *et al.*, "Analysis and perspectives on the ana avatar xprize competition," *International Journal of Social Robotics*, pp. 1–32, 2024.

[36] M. Wonsick and T. Padır, "Human-humanoid robot interaction through virtual reality interfaces," in *2021 IEEE Aerospace Conference (50100)*, IEEE, 2021, pp. 1–7.

[37] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer, "Vmp: Versatile motion priors for robustly tracking motion on physical characters," 2024.

[38] Z. Luo, J. Cao, A. W. Winkler, K. Kitani, and W. Xu, "Perpetual humanoid control for real-time simulated avatars," in *International Conference on Computer Vision (ICCV)*, 2023.

[39] J. Won, D. Gopinath, and J. Hodgins, "A scalable approach to control diverse behaviors for physically simulated characters," *ACM Trans. Graph.*, vol. 39, 4 2020.

[40] N. Chentanez, M. Müller, M. Macklin, V. Makoviychuk, and S. Jeschke, "Physics-based motion capture imitation with deep reinforcement learning," *Proceedings - MIG 2018: ACM SIGGRAPH Conference on Motion, Interaction, and Games*, 2018.

[41] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, 143:1–143:14, Jul. 2018.

[42] P. Dugar, A. Shrestha, F. Yu, B. van Marum, and A. Fern, "Learning multi-modal whole-body control for real-world humanoid robots," *arXiv preprint arXiv:2408.07295*, 2024.

[43] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, "Maskedmimic: Unified physics-based character control through masked motion," in *ACM Transactions On Graphics (TOG)*, ACM New York, NY, USA, 2024.

[44] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters," *arXiv preprint arXiv:2205.01906*, 2022.

[45] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng, "Calm: Conditional adversarial latent models for directable virtual characters," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, Los Angeles, CA, USA: Association for Computing Machinery, 2023.

[46] Z. Luo *et al.*, "Universal humanoid motion representations for physics-based control," *arXiv preprint arXiv:2310.04582*, 2023.