Embodiment-Aware Generalist Specialist Distillation for Unified Humanoid Whole-Body Control

Quanquan Peng[1*], Yunfeng Lin[1*], Yufei Xue[1], Jiangmiao Pang[2], Weinan Zhang[1 2†]
[1]Shanghai Jiao Tong University, [2]Shanghai AI Lab

Fig. 1: In this work, we propose a distillation framework that yields a *single* whole-body controller that runs on heterogeneous humanoids. Through cyclic forking of embodiment-specific specialists and distillation back into the generalist, the resulting policy executes rich commands (walking, squatting, and leaning) in simulation and the real world, outperforming baselines in tracking accuracy and robustness.

***Abstract*—Humanoid Whole-Body Controllers trained with reinforcement learning (RL) have recently achieved remarkable performance, yet many target a single robot embodiment. Variations in dynamics, degrees of freedom (DoFs), and kinematic topology still hinder a single policy from commanding diverse humanoids. Moreover, obtaining a generalist policy that not only transfers across embodiments but also supports richer behaviors—beyond simple walking to squatting, leaning—remains especially challenging. In this work, we tackle these obstacles by introducing EAGLE, an iterative generalist-specialist distillation framework that produces a single unified policy that controls multiple heterogeneous humanoids without per-robot reward tuning. During each cycle, embodiment-specific specialists are forked from the current generalist, refined on their respective robots, and new skills are distilled back into the generalist by training on the pooled embodiment set. Repeating this loop until performance convergence produces a robust Whole-Body Controller validated on robots such as Unitree H1, G1, and Fourier N1. We conducted experiments on five different robots in simulation and four in real-world settings. Through quantitative evaluations, EAGLE achieves high tracking accuracy and robustness compared to other methods, marking a step toward scalable, fleet-level humanoid control.**

I. Introduction

Training one policy that can transfer across different robot platforms is emerging as a key challenge in robotics research. In manipulation, for example, large-scale visuomotor models trained on cross-domain datasets have been shown to generalize to different manipulators in various tasks [1, 2], while similar results also appear in navigation [3, 4]. These successes suggest that cross-embodiment learning is feasible and promising.

For humanoid whole-body control (WBC), however, the situation is different. Large-scale, data-hungry paradigms that work in manipulation, where human operators can teleoperate robot arms [5–7] to collect hundreds of demonstrations for training, do not easily translate to legged locomotion, since a robot without an existing controller simply cannot be controlled via teleoperation. Therefore, this imitation-learning pipeline stalls at the data-collection stage. As an alternative, reinforcement learning (RL) algorithms [8, 9] have propelled WBC to impressive heights, allowing robots to run, hop, dance, and manipulate [10–13]. Yet almost every milestone is tied to one specific embodiment. Discrepancies in hardware attributes across dynamics, kinematics, and morphology block a single policy from transferring directly; each new robot typically forces the entire training and reward-tuning pipeline to start over, slowing down deployment. Recent studies have explored bridging this gap with diffusion-based motion priors [14] or by massively randomizing URDF parameters during training [15–17]. However, these methods are

* denotes equal contribution † denotes corresponding author

still limited to low-dimensional velocity commands, leaving other behaviors such as torso pitching beyond reach, and importantly, they have not yet been validated on diverse real humanoid hardware.

Naturally, a question was raised: *Can one policy control multiple humanoids while still supporting diverse whole-body commands?* In this work, we propose **EAGLE**, an embodiment-aware generalist-specialist distillation framework for unified humanoid whole-body control. EAGLE couples an iterative generalist-specialist distillation loop with a unified, high-dimensional command interface. The loop begins with a generalist trained in a simulator that pools multiple different humanoid models (Unitree H1, Unitree G1, Booster T1, Fourier N1, and PNDbotics Adam); from this policy, we spawn embodiment-specific specialists, fine-tune them on their respective robots, and then distill their new skills back into the generalist in a DAgger-based [18] way, repeating the cycle until the generalist policy converges. Built on the HugWBC [11], our command vector encodes base linear and angular velocities, body pitch, and base height, enabling behaviors far richer than pure walking—such as leaning and squatting—without any per robot-specific reward tuning.

In summary, our key contributions are as follows:

- We introduce an embodiment-aware generalist-specialist distillation loop that unifies whole-body control across heterogeneous humanoids without per-robot reward tuning.
- We deploy a unified high-dimensional command interface, which allows one policy to perform squatting, leaning, and base velocity tracking, capabilities that previous approaches cannot support.
- We conduct extensive experiments on five humanoid robots in simulation and four in the real world, showing that EAGLE achieves higher command-tracking accuracy and superior performance compared with baseline methods.

II. Related Works

Cross-embodiment learning. Training a model that is able to transfer across different robot platforms can be promising [19–21]. Many works have attempted to address this issue by training on diverse datasets [1, 2, 22, 23], finding a unified action representation [24], or domain randomization and adaptation [25]. Meanwhile, humanoid robots have more DoFs and distinct morphological structures, making the task of finding a general controller more challenging. Some RL-based works tried to adopt diffusion models along with a residual term [14] or by training Transformers with a large pool of URDF randomized robots [15–17]. Though substantial progress has been made, most methods still support only low-dimensional velocity commands, and other diverse behaviors remain underexplored. Moreover, these approaches remain limited to simulation studies and do not demonstrate transfer to multiple physical humanoid robots, which is the central focus of our work.

Learning-based humanoid control. Recent learning-based methods combine massively parallel simulation and reinforcement learning to achieve robust locomotion, recovery, and terrain traversal [26–28]. Building on these advances, recent work on humanoids has spanned real-world whole-body control [12, 13, 29], teleoperation interfaces [5, 30], and improved robustness under disturbances [31]. Meanwhile, HugWBC [11] introduces a unified, high-dimensional command space that enables a single policy to easily switch between walking, standing, and hopping, providing a versatile backbone for whole-body control. We adopt this framework and focus on closing the cross-embodiment gap.

Policy distillation. Many works study transferring knowledge from teacher to deployable student models. Classical knowledge distillation compresses a network into a smaller policy via soft targets [32–34]. For multi-task RL, Distral distills task policies into a shared policy to aid transfer [35], and Kickstarting warm-starts learners from a fixed teacher with a decaying KL [36]. DAgger aggregates on-policy states with teacher relabels to reduce covariate shift [18]. In robotics, teacher-student pipelines have proved effective for legged locomotion [26, 37]. Building on these ideas, we adopt iterative generalist-specialist distillation, producing a generalist policy that scales across heterogeneous humanoids while preserving rich whole-body commands.

III. Preliminaries

We consider N humanoid robots with heterogeneous morphologies. For robot $i \in \{1,\dots,N\}$, we model the control problem as a Markov Decision Process (MDP): $\mathcal{M}_i = \langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{P}_i, \mathcal{R}_i, H \rangle$, where $x_t^i \in \mathcal{S}_i, a_t^i \in \mathcal{A}_i$ is the robot state and action at time step t, $\mathcal{P}_i$ is the state-action transition dynamics, $\mathcal{R}_i$ is the reward function, and H is the episode horizon. Formally, our goal is to get a generalist policy π_g such that, given command vector c_t, it can generate $a_t \sim \pi_g(o_t)$ to maximize the objective function:

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\tau_i \sim \pi_g}\Big[\sum_{t=0}^{H} \gamma^t \mathcal{R}_i\big(s_t^i, c_t, a_t^i\big)\Big], \tag{1}$$

where $\gamma \in (0,1)$ is the reward discount factor, o_t is the (s_t, c_t) pair, and $\tau_i = \{(s^i, s^{i\prime}, a^i, r^i)_{0:H}\}$ denotes a rollout on embodiment i.

IV. Methodology

In this section, we introduce EAGLE, our cross-embodiment training framework. Sec. IV-A describes how we build a unified, high-dimensional command and observation space for humanoid control for diverse robot behaviors. Because the robots differ in their DoF counts and kinematic layouts, Sec. IV-B explains how we align heterogeneous observation-action vectors , enabling the neural network to exploit morphological information while handling all embodiments. Sec. IV-C then outlines the reward-function design used for RL training. Finally, Sec. IV-D details the iterative generalist–specialist distillation loop that transfers embodiment-specific skills back into a shared policy.

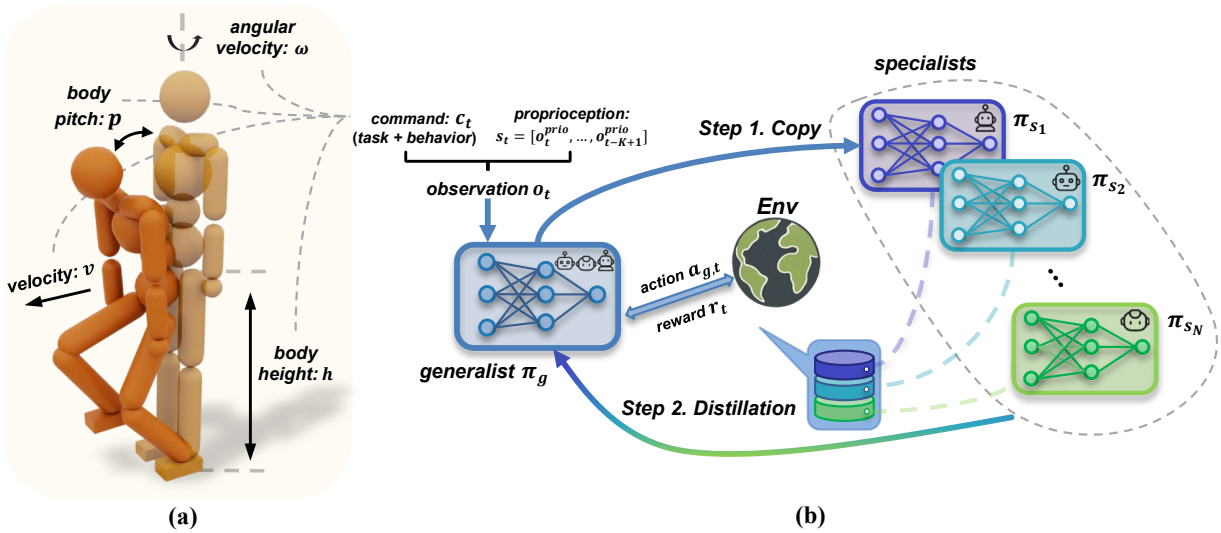


Fig. 2: **Method Overview. (a)** Unified command interface. The command vector $\mathbf{c}_t$ comprises *task* commands $\mathbf{v}_t$ (linear velocities v_x, v_y, angular velocity ω) and *behavior* commands $\mathbf{b}_t$ (base height h, body pitch p); together with a window of robot proprioception s_t, they form the observation o_t. **(b)** Generalist-specialist distillation framework. Each round *copies* π_g to N specialists $\{\pi_{s_i}\}$ for per-robot fine-tuning, then *distills* back by running π_g, relabeling actions with the corresponding specialist, and updating with the loss in Eq. 9. Repeating this loop yields a single controller that scales across embodiments while retaining rich whole-body commands.

A. *Unified Command and Observation Space.*

1) Command Design

To make the robot support richer behaviors, not limited to walking, we should design a unified high-dimensional command space. Inspired by the generalized command interface of HugWBC [11], we design the command vector (shown in Fig. 2(a))

$$\mathbf{c}_t = \left[v_x,\ v_y,\ \omega,\ h,\ p\right]^\top \in \mathbb{R}^5, \tag{2}$$

where v_x and v_y are desired base-frame linear velocities (m/s), ω is the angular velocity (rad/s), h is the base height offset (m), and p is the body-pitch angle (rad).

The first three components $\mathbf{v}_t = [v_x, v_y, \omega]^\top$, specify *task command*: they define where the robot should go. The last two $\mathbf{b}_t = [h, p]^\top$, modulate *behavior command*, let the robot do behaviors like squatting, leaning, or standing. Separating $\mathbf{v}_t$ and $\mathbf{b}_t$ allows a single policy to realize a rich repertoire of movements beyond simply walking.

2) Embodiment-aware Observation

As for the policy observation, the actor receives a chunk of proprioception o^{prio} (joint positions q, joint velocities $\dot{q}$, base angular velocity ω_{base}, and projected gravity $\mathbf{g}^{\text{proj}}$ w.r.t. the robot's base frame) of $K = 5$ frames: $s_t = [o_t^{prio}, \cdots, o_{t-K+1}^{prio}]$. To help the robot learn foot lifting that follows a gait pattern, we add a gait clock function $\sin(2\pi\phi)$ in observation:

$$\begin{aligned} \phi_{1,t+1} &= \phi_{1,t} + f\mathrm{d}t, \\ \phi_{2,t+1} &= \phi_{1,t+1} + \varphi, \end{aligned} \tag{3}$$

where $f \in \mathbb{R}$ is the command gait frequency (Hz) and $\varphi \in [0, 1)$ is the phase offset between the two legs. Moreover, we adopt the asymmetric actor-critic paradigm [38], where the critic receives additional privileged information: the base linear velocity, height error, foot clearance and contact forces, link collision state, terrain friction coefficient, and height scan samples.

To let the network reason about the differences in robot morphology, we additionally provide *embodiment-aware observation* $o_{ea} \in \mathbb{R}^{m\times 10}$ as privileged information for the critic and task the actor to estimate it. Here m is the number of predicted rigid bodies, and in this work, we pick the torso and both feet ($m = 3$) to preserve locomotion-critical signals instead of upper-body links. For each rigid body, o_{ea} contains information of the corresponding mass (in $\mathbb{R}$), CoM position (in $\mathbb{R}^3$), and inertia matrix (in $\mathbb{R}^6$). Exp. V-C shows that this helps the model to distinguish across robots and their locomotion patterns.

B. *Embodiment Alignment.*

As mentioned above, different humanoids expose distinct observation and action spaces, which prevents a single network from sharing weights across embodiments. We resolve this by embedding all robots into *unified* observation-action spaces via zero padding and fixed index mappings. Specifically, we choose a unified action length $D_a = 32$, covering lower limbs (6 DoFs per leg with hip RPY, knee P, ankle RP), the waist (RPY), the head (RPY), and upper limbs (7 DoFs per arm with shoulder RPY, elbow P, wrist RPY), where RPY denotes roll/pitch/yaw. For robot i, its native action $a_t^i \in \mathbb{R}^{n_i}$ is embedded as

$$\tilde{a}_t^i = P_m \begin{bmatrix} a_t^i \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{D_a}, \tag{4}$$

Algorithm 1 Overall Process

Require: Number of different robots: N, Rollout buffer: $\mathcal{D}$;

Train generalist policy π_g on N robots; ▷ see Sec. IV-B for details
while π_g doesn't converge **do**
  For $i \in \{1,\dots,N\}$, $\pi_{s_i} \leftarrow \pi_g$; ▷ initialize model by copying weight
  Fine-tune each π_{s_i} on i-th robot;
  for $k = 1$ **to** T epochs **do**
    $\mathcal{D}_{\pi_g} \leftarrow \{(s, s', a_g, r)\}$; ▷ collect parallel env trajectories
    Relabel $\mathcal{D}_{\pi_g}$ with specialist action a_s from $\pi_{s_{1:N}}$;
    Update π_g by Eq. 9;
  end for
end while
return π_g;

where $\mathbf{0} \in \mathbb{R}^{D_a - n_i}$ is a zero vector and $P_m \in \{0,1\}^{D_a \times D_a}$ is a robot-specific permutation matrix that maps each joint to a fixed global index (e.g., `Right-Hip-Pitch` joint $\to 0$, `Left-Hip-Pitch` joint $\to 1$, ...). At execution time, the environment wrapper applies the inverse mapping to recover native actions:

$$a_t^i = S_i \tilde{a}_t^i, \quad \text{where } S_i \in \{0,1\}^{n_i \times D_a},\ S_i P_i \begin{bmatrix} I_{n_i} & \mathbf{0} \end{bmatrix}^\top = I_{n_i}. \quad (5)$$

We preserve robot-specific parameters, such as the swing height of both feet $\ell_{\text{target},j} (j \in \{0,1\})$, as well as joint stiffness and damping parameters, as these will significantly impact the training phase.

Moreover, to train the generalist policy π_g, we will also spawn different humanoids together in parallel within the same training environment with equal probability. As for how to align them into the reward function $\mathcal{R}_i(\cdot)$, please see Sec. IV-C for details.

C. Reward Function Design.

We adopt PPO [8] algorithm with asymmetric actor-critic paradigm (see Sec. IV-A) as our Reinforcement Learning algorithm. The per-step reward is decomposed into a task term r_t^{task}, a behavior term r_t^{beh}, and a regularization term r_t^{reg}:

$$r_t = \omega_1 r_t^{\text{task}} + \omega_2 r_t^{\text{beh}} + \omega_3 r_t^{\text{reg}}. \quad (6)$$

Here, r_t^{task} encourages tracking of the task command $\mathbf{v}_t$, r_t^{beh} encourages tracking of the behavior command $\mathbf{b}_t$, and r_t^{reg} penalizes failures and undesirable actions (e.g., falls, excessive torques, abrupt changes). For most of the reward, we inherit them from HugWBC [11]. See Tab. I for details.

Since we train multiple robots jointly, reward terms must be aligned across embodiments. In practice, we keep the weights shared across robots and adjust only term-specific targets or scales to account for morphology. For example, in the base-height tracking term $\|h^{\text{target}} - h\|^2$, the target h^{target} is set per robot (e.g., its nominal base height).

TABLE I: **Training reward terms** (for simplicity, note that $\rho_\kappa(x) = \exp(-x/\kappa)$). For the definition of $C(\phi)$ function, please refer to HugWBC [11].

Term	Definition	Coefficient
Task Reward: r_t^{task}		
Lin. vel. tracking	$\rho_{0.25}\big(\|\mathbf{v}_{xy}^{\text{target}} - \mathbf{v}_{xy}\|^2\big)$	2
Ang. vel. tracking	$\rho_{0.25}\big(\|\omega_z^{\text{target}} - \omega_z\|^2\big)$	2.5
Behavior Reward: r_t^{beh}		
Base height	$\|h^{\text{target}} - h\|^2$	-60
Body pitch	$\|p^{\text{target}} - p\|^2$	-1
Foot-swing position	$\sum_j (1 - C(\phi_j)) \left\|\ell_{\text{target},j} - \ell_{\text{foot},j}\right\|^2$	-30
Contact vel.	$\sum_j C(\phi_j) \left[1 - \rho_5(\|\mathbf{v}_{xy,j}^{\text{foot}}\|^2)\right]$	-1
Contact force	$\sum_j (1 - C(\phi_j)) \left[1 - \rho_{50}(\|\mathbf{f}_{xy,j}^{\text{foot}}\|^2)\right]$	-1
Regularization Reward: r_t^{reg}		
Roll-pitch Ang. vel.	$\|\omega_{xy}\|^2$	-0.1
Vertical speed	$\|v_z\|^2$	-2
Foot slip	$1 - \sum_j \rho_1(\|\mathbf{v}_{xy}^{\text{foot},j}\|^2)$	-0.1
Action rate	$\|a_t - a_{t-1}\|^2$	-2×10^{-3}
Action smoothness	$\|a_{t-2} - 2a_{t-1} + a_t\|^2$	-2×10^{-3}
Joint torque	$\|\tau\|^2$	-1×10^{-5}
Joint acceleration	$\|\ddot{\mathbf{q}}\|^2$	-5×10^{-8}
Upper-joint deviation	$\|\mathbf{q}_{\text{upper}}^{\text{default}} - \mathbf{q}_{\text{upper}}\|^2$	-5
Hip-joint deviation	$\|\mathbf{q}_{\text{hip}}^{\text{default}} - \mathbf{q}_{\text{hip}}\|^2$	-0.4
Base Orientation	$\|\mathbf{g}_{xy}^{\text{proj}}\|^2$	-5

D. Distillation Loop Overview.

So far, the alignment scheme in Sec. IV-B enables a PPO policy π_g to control different robots, but its performance soon saturates at a mediocre level. To further improve it, we introduce an iterative *generalist-specialist distillation loop* (shown in Fig. 2(b)), that alternates between *specialize* phase and *generalize* phase:

1) **specialize.** Copy the current generalist to N robot-specific specialists $\{\pi_{s_i}\}$ and fine-tune each on its own robot only.
2) **generalize.** Collect trajectories of different robots by running the π_g. Then relabel actions proposed by the corresponding specialists, and distill π_g with it.

As for the distillation mentioned above, prior approaches such as standard DAgger only align the output action distributions:

$$\mathcal{L}_a = \mathbb{E}_{\tau\sim\pi_g} \mathbb{E}_{o_t\sim\tau} \big(\pi_g(o_t) - \pi_s(o_t)\big)^2. \quad (7)$$

However, since the generalist and specialist share the same model architectures, our framework introduces an additional representation-level alignment loss:

$$\mathcal{L}_e = \mathbb{E}_{\tau\sim\pi_g} \mathbb{E}_{o_t\sim\tau} \big(e_{\pi_g}(s_t) - e_{\pi_s}(s_t)\big)^2, \quad (8)$$

where $e(\cdot)$ denotes the *hidden feature* of the actor network by taking proprioception buffer s_t, so $\mathcal{L}_e$ aligns the two

policies in representation space rather than actions alone. We show in Sec. V-C that this loss term improves cross-embodiment training. Moreover, we want to maintain the critic updated while also preserving the generalist model's exploration ability, instead of simply mimicking specialists' behavior. Therefore, we keep the PPO loss for RL exploration, and the total distillation loss becomes:

$$\mathcal{L} = \mathcal{L}_{PPO} + \alpha \cdot \mathcal{L}_a + \beta \cdot \mathcal{L}_e, \tag{9}$$

where $\alpha = 0.02$ and $\beta = 1$ are constant coefficients. Algo. 1 summarizes the full procedure.

This two-way exchange lets the generalist improve steadily while each specialist begins the next round from a stronger baseline, boosting overall control quality and eliminating per-robot reward tuning or network redesign.

V. Experiments

In our experiment, we evaluate our method on five different robots in simulation and four in the real world. We intend to answer the following questions:

- **Q1**: How much does EAGLE improve command-tracking accuracy compared to other methods?
- **Q2**: How does each component of the generalist-specialist cycle contribute to final performance?
- **Q3**: What embodiment-aware representation does the EAGLE policy learn?
- **Q4**: How well does the EAGLE policy transfer to real robots in a zero-shot way?

Experiment Setup. In this work, we test on five humanoid models: Unitree H1, Unitree G1, Booster T1, Fourier N1, and PNDbotics Adam, whose DoFs are 19, 29, 23, 23, 25, respectively. The simulation training is done in Isaac Gym [39] with 4096 parallel environments, each robot type being sampled with equal probability.

Comparison Method. For simplicity, we abbreviate the baseline methods as follows:

- PPO: RL baseline policy trained on all robots together, *i.e.*, the π_g at line 1 of Algo. 1.
- PPO *w/o* EO: the same policy but trained without embodiment-aware observation (Sec. IV-A) and the corresponding reconstruction loss.
- COMPASS [4]: We reimplement COMPASS, a three-stage cross-embodiment method that first imitates demonstrations to learn a world-model-conditioned base policy and then trains residual RL specialists and uses KL loss to distill them into a general policy.
- Kickstarting [36]: a distillation method in which π_g is updated by $\mathcal{L} = \mathcal{L}_{\text{PPO}} + \beta_t \,\text{KL}(\pi_s \| \pi_g)$, where π_s is the specialist policy and the coefficient $\beta_t = \alpha exp(-\lambda t)$ decays exponentially.
- EAGLE (ours): our framework after a *single* round of distillation, the same as Kickstarting.
- EAGLE *w/* ID (ours): the full iterative distillation variant until the generalist's performance converges.

We also use the same model architecture and reward functions for fair comparison.

A. Command Tracking Accuracy Analysis.

To quantitatively understand how well our method compares to baseline methods, we refer to the command tracking error.

Metric definition. For each command dimension $d \in \{v_x, v_y, \omega, h, p\}$ we define the *episode tracking error*:

$$E_d = \mathbb{E}_{\tau\sim\pi}\left[\left|\hat{c}_t^d - c_t^d\right|\right], \tag{10}$$

where c_t^d is the commanded target drawn from the predefined range (Tab. III), and $\hat{c}_t^d$ is the value read from the robot's proprioceptive state. We report five errors $E_{v_x}, E_{v_y}, E_\omega, E_h, E_p$ separately; lower values indicate more accurate command tracking.

Results. Tab. II summarizes the episode-averaged single command tracking errors. We can see that compared to the PPO method, which was trained on all different robots together, both EAGLE and EAGLE *w/* ID surpass it on most metrics. As for the ablation on distillation method (Tab. II(b)), compared with the KL-based Kickstarting baseline, our single-round EAGLE already wins on 80% robot–metric pairs and is never catastrophic, whereas Kickstarting can become unstable on certain embodiment like G1, Adam and T1 (e.g. E_{v_x}=0.761 on T1 is five times higher than PPO). The result confirms that the DAgger-style loss in Eq. 9 is more robust across very different embodiments.

We also compare with the cross-embodiment baseline **COMPASS** [4]. Although it achieves good results on some robots like Fourier N1, it fails to generalize across all embodiments. In contrast, our method maintains low E_{v_x} on all robots while keeping other metrics competitive or superior, demonstrating stable, single-policy control across heterogeneous humanoids.

B. Ablation of the Distillation Loop.

Furthermore, Tab. II contrasts EAGLE after a *single* distillation round with its iterative variant. For most task commands, EAGLE *w/* ID achieves lower task-command errors compared to EAGLE, which wasn't trained after the iterative loop. Although a few behavior-command errors increase slightly, the overall gains outweigh these minor regressions. Fig. 3 plots the evolution of task-command errors across rounds and shows that, after each distillation cycle, both the specialists (highlighted in yellow) and the generalists improve in most cases. This suggests that repeating the loop until convergence is more effective than a single distillation pass.

We additionally train *single-robot* PPO policies that are exposed to only one robot type. As Tab. IV shows, on robots such as H1, our generalist achieves performance comparable to these specialized baselines, while the corresponding EAGLE-specialist even surpasses them. This means cross-embodiment training has the potential to match, and sometimes exceed, policies tailored to a single embodiment.

C. Analysis of Learned Representations.

As we can see in Tab. II(a), without the embodiment-aware observation, PPO *w/o* EO's overall performance drops,

TABLE II: **Command-tracking errors** (lower is better). E_{v_x} and E_{v_y} are linear-velocity error in m/s, E_ω is angular-velocity error in rad/s, E_h is base height error in m, and E_p is body-pitch error in rad. Each tracking error is averaged over 2048 environments and 1000 steps. Our method maintains low errors across all robots compared to other cross-embodiment baselines.

Robot Type	**Unitree H1**					**Unitree G1**					**PNDbotics Adam**				
Tracking Error (↓)	E_{v_x}	E_{v_y}	E_ω	E_h	E_p	E_{v_x}	E_{v_y}	E_ω	E_h	E_p	E_{v_x}	E_{v_y}	E_ω	E_h	E_p
(a) Ablation: embodiment-aware observation															
PPO	0.108	0.129	0.114	**0.075**	0.182	0.092	0.127	0.121	0.051	**0.164**	0.328	0.107	0.235	0.051	0.204
PPO *w/o* EO	0.290	0.161	0.152	0.078	0.189	0.104	0.113	0.095	0.052	0.165	0.375	0.158	0.310	0.063	0.218
(b) Ablation: baseline & distillation method															
COMPASS [4]	0.725	0.390	0.476	0.085	0.273	0.171	0.119	0.124	**0.035**	0.169	0.763	0.371	0.339	0.045	0.233
Kickstarting [36]	0.323	0.141	0.122	**0.075**	0.182	0.117	0.118	0.075	0.050	0.167	0.399	0.086	0.141	**0.041**	**0.192**
EAGLE (ours)	0.061	0.117	**0.054**	0.076	**0.169**	**0.066**	0.105	0.077	0.049	**0.164**	0.182	**0.084**	**0.115**	0.050	0.196
EAGLE *w/* ID (ours)	**0.051**	**0.110**	**0.054**	0.082	0.186	0.067	**0.095**	**0.071**	0.047	0.176	**0.156**	0.101	0.117	0.056	0.197

Robot Type	**Fourier N1**					**Booster T1**				
Tracking Error (↓)	E_{v_x}	E_{v_y}	E_ω	E_h	E_p	E_{v_x}	E_{v_y}	E_ω	E_h	E_p
(a) Ablation: embodiment-aware observation										
PPO	0.124	0.141	0.107	0.073	0.165	0.147	0.203	0.105	0.047	0.156
PPO *w/o* EO	0.126	0.144	0.076	0.079	0.162	0.140	0.194	0.104	0.046	0.151
(b) Ablation: baseline & distillation method										
COMPASS [4]	0.060	**0.117**	0.137	**0.056**	0.160	0.831	0.307	0.179	**0.044**	0.151
Kickstarting [36]	0.147	0.142	0.130	0.077	0.168	0.761	0.190	0.100	0.047	**0.149**
EAGLE (ours)	0.068	0.126	0.058	0.074	**0.157**	0.109	**0.187**	0.045	0.045	0.152
EAGLE *w/* ID (ours)	**0.056**	0.135	**0.056**	0.074	0.168	**0.090**	**0.187**	**0.041**	**0.044**	0.159

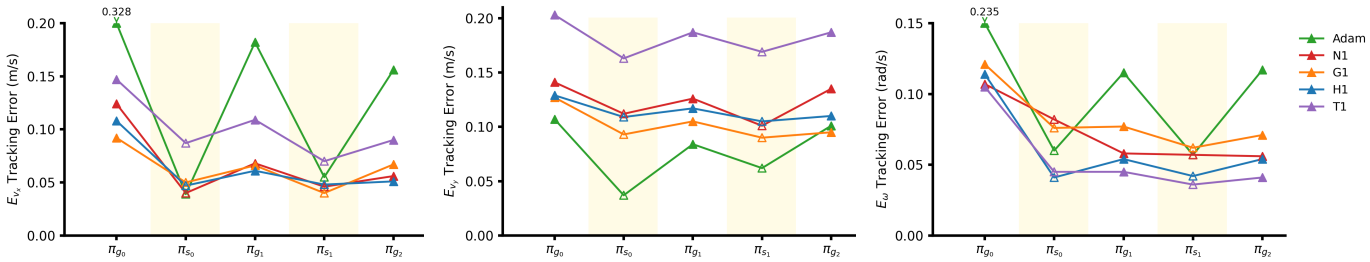


Fig. 3: **Tracking-error curves for E_{v_x}, E_{v_y}, and E_ω** (left to right). The △, ▲ represents the specialist and generalist policy respectively (as described at lines 2-9 of Algo. 1). We can see the trend that, after each distillation cycle, both symbols move downward, showing that iterative distillation steadily improves tracking accuracy for specialists and generalists.

TABLE III: **Command ranges.** "Initial" and "Finishing" denote the RL curriculum learning stage (all in SI units).

Group	**Term**	**Initial Range**	**Finishing Range**
Task Commands	v_x	$[-0.3, 0.6]$	$[-0.6, 1.2]$
	v_y	$[-0.5, 0.5]$	$[-0.4, 0.4]$
	ω	$[-0.5, 0.5]$	$[-1.0, 1.0]$
Behavior Commands	h	$[-0.3, 0]$	$[-0.3, 0]$
	p	$[-0.3, 0.5]$	$[-0.3, 0.5]$

TABLE IV: **Unitree H1 command-tracking errors** (lower is better). After cross-embodiment training, EAGLE-generalist matches, and its robot-specific policy EAGLE-specialist surpasses a policy trained exclusively on H1.

Tracking Error (↓)	E_{v_x}	E_{v_y}	E_ω	E_h	E_p
single-robot PPO	0.067	**0.101**	0.052	**0.072**	0.181
EAGLE-generalist	0.061	0.117	0.054	0.076	**0.169**
EAGLE-specialist	**0.048**	0.105	**0.042**	**0.072**	0.174

and for some robots like H1, its E_{v_x} tracking error increases significantly, showing that the policy fails to integrate different robot-specific knowledge together.

In Fig. 4, we also use t-SNE to visualize the latent embedding $e_\pi(\cdot)$ (in Eq. 8) by commanding the robot to walk forward. The *left* panel depicts the PPO trained *without* embodiment-aware observation: several robot types collapse into an overlapping cluster (dashed ellipse), indicating that the encoder cannot distinguish their dynamics. In contrast, the *right* panel shows our policy, whose latent space forms well-separated clusters for all five embodiments, confirming that the embodiment cue helps the network learn morphology-specific representations.

D. Zero-shot Sim2real Performance.

To demonstrate the policy's ability to deploy on real-world hardware, we tested it on four real humanoid robots: Unitree H1, Unitree G1, Booster T1, and Fourier N1 in a zero-shot

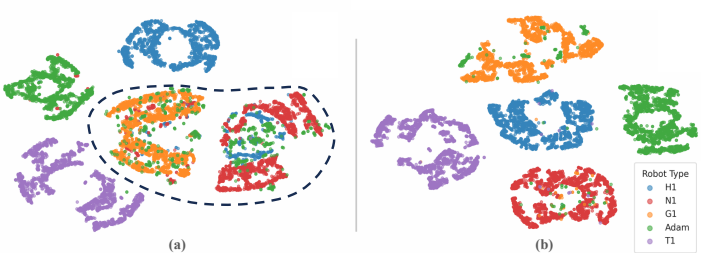


Fig. 4: **t-SNE visualization of policy latent-space representation.** By commanding different robots to walk forward for certain timesteps, we can see *w/o* embodiment-aware observation *(left)*, the policy outputs latent vectors $e_\pi(\cdot)$ collapse into overlapping clusters. In contrast, our method *(right)* is better separated, showing successful morphology-aware representation.

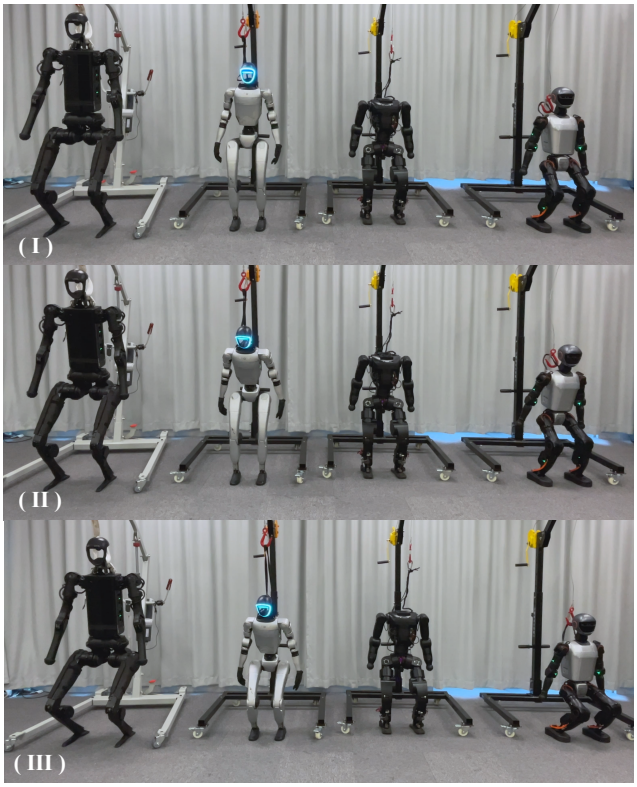
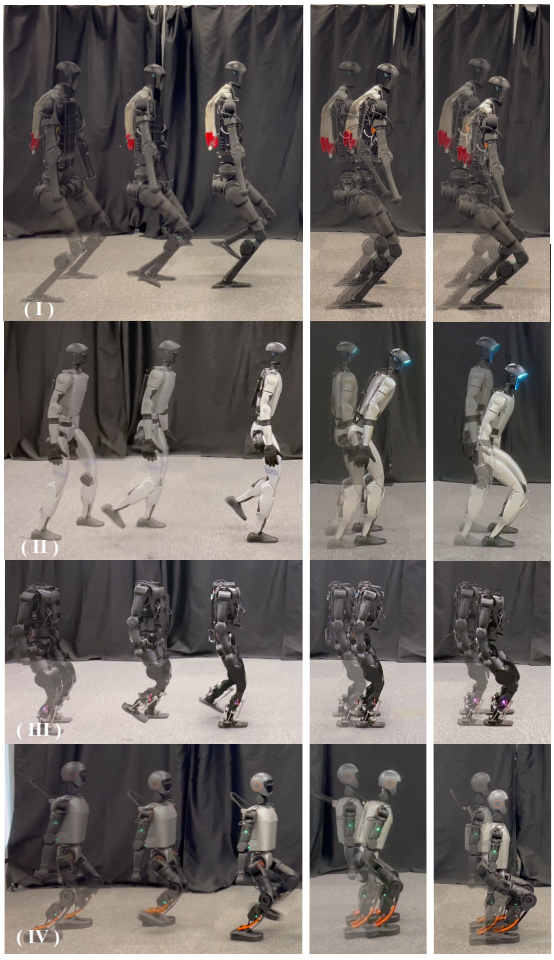


Fig. 5: **Real-world synchronous movement.** We also developed a shared low-level control framework to show our policy can let different robots execute synchronous movement, from leaning (II) to squatting (III).

manner. We use UniCon [40] as a unified control interface for operating different robot platforms.

In Fig. 6 and Fig. 5, the policy, trained purely in simulation, executes diverse behaviors including walking, leaning, and squatting. Despite variations in morphology and hardware dynamics, the controller consistently produces stable motions across embodiments. These results confirm that our distillation framework supports robust sim2real transfer and enables a single policy to generalize effectively to heterogeneous humanoid platforms.



(a) walking (b) leaning (c) squatting

Fig. 6: **Real-world experiment.** We evaluate our policy across four embodiments: (I) Unitree H1, (II) Unitree G1, (III) Fourier N1, and (IV) Booster T1. Each robot successfully executes diverse commands, including (a) walking, (b) leaning, and (c) squatting, demonstrating robust zero-shot transfer from simulation to the real world.

VI. Discussion

In this work, we do not exhaustively evaluate on *unseen* humanoid embodiments at test time. A promising direction is to combine our distillation framework with explicit URDF randomization or morphology randomization during training, as explored by previous works like [15–17], so the generalist learns to cover a wider space of structures.

Another limitation is that our embodiment-aware observation is still coarse; future work could enrich it with finer-grained morphology descriptors such as limb lengths, joint topology, and kinematic tree structure, which may further improve cross-embodiment generalization.

VII. Conclusion

In this paper, we present EAGLE, an embodiment-aware generalist-specialist distillation framework for unified humanoid whole-body control. By combining a high-dimensional

command interface with an iterative distillation loop, EAGLE enables a single policy to control five structurally diverse humanoids without per-robot reward tuning. Through extensive simulation and real-world experiments, we show that our approach not only achieves superior command-tracking accuracy but also supports rich behaviors like squatting and leaning. These results highlight the potential of embodiment-aware distillation as a scalable solution for cross-embodiment learning in complex locomotion tasks.

VIII. Acknowledgment

The SJTU team is partially supported by the National Natural Science Foundation of China (62322603).

References

[1] 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.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[3] K.-H. Zeng, Z. Zhang, K. Ehsani, R. Hendrix, J. Salvador, A. Herrasti, R. Girshick, A. Kembhavi, and L. Weihs, "Poliformer: Scaling on-policy rl with transformers results in masterful navigators," *arXiv preprint arXiv:2406.20083*, 2024.

[4] W. Liu, H. Zhao, C. Li, J. Biswas, S. Pouya, and Y. Chang, "Compass: Cross-embodiment mobility policy via residual rl and skill synthesis," *arXiv preprint arXiv:2502.16372*, 2025.

[5] 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.

[6] S. Luo, Q. Peng, J. Lv, K. Hong, K. R. Driggs-Campbell, C. Lu, and Y.-L. Li, "Human-agent joint learning for efficient robot manipulation skill acquisition," *arXiv preprint arXiv:2407.00299*, 2024.

[7] 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.

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

[9] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[10] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024.

[11] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang, "A unified and general humanoid whole-body controller for fine-grained locomotion," *arXiv preprint arXiv:2502.03206*, 2025.

[12] 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.

[13] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang *et al.*, "Hover: Versatile neural whole-body controller for humanoid robots," *arXiv preprint arXiv:2410.21229*, 2024.

[14] S. Yang, Z. Fu, Z. Cao, G. Junde, P. Wensing, W. Zhang, and H. Chen, "Multi-loco: Unifying multi-embodiment legged locomotion via reinforcement learning augmented diffusion," *arXiv preprint arXiv:2506.11470*, 2025.

[15] N. Bohlinger, G. Czechmanowski, M. Krupka, P. Kicki, K. Walas, J. Peters, and D. Tateo, "One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion," *arXiv preprint arXiv:2409.06366*, 2024.

[16] B. Ai, L. Dai, N. Bohlinger, D. Li, T. Mu, Z. Wu, K. Fay, H. I. Christensen, J. Peters, and H. Su, "Towards embodiment scaling laws in robot locomotion," *arXiv preprint arXiv:2505.05753*, 2025.

[17] M. Liu, D. Pathak, and A. Agarwal, "Locoformer: Generalist locomotion via long-context adaptation," *arXiv preprint arXiv:2509.23745*, 2025.

[18] 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.

[19] W. Yu, J. Tan, C. K. Liu, and G. Turk, "Preparing for the unknown: Learning a universal policy with online system identification," *arXiv preprint arXiv:1702.02453*, 2017.

[20] C. Ying, H. Zhongkai, X. Zhou, X. Xu, H. Su, X. Zhang, and J. Zhu, "Peac: Unsupervised pre-training for cross-embodiment reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 54 632–54 669, 2024.

[21] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine, "Pushing the limits of cross-embodiment learning for manipulation and navigation," *arXiv preprint arXiv:2402.19432*, 2024.

[22] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, "Xskill: Cross embodiment skill discovery," 2023.

[23] 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.

[24] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2020.

[25] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

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

[27] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023.

[28] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

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

[30] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, "Twist: Teleoperated whole-body imitation system," *arXiv preprint arXiv:2505.02833*, 2025.

[31] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang, "Learning humanoid standing-up control across diverse postures," *arXiv preprint arXiv:2502.08378*, 2025.

[32] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[33] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International journal of computer vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[34] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3967–3976.

[35] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, "Distral: Robust multitask reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[36] S. Schmitt, J. J. Hudson, A. Zidek, S. Osindero, C. Doersch, W. M. Czarnecki, J. Z. Leibo, H. Kuttler, A. Zisserman, K. Simonyan *et al.*, "Kickstarting deep reinforcement learning," *arXiv preprint arXiv:1803.03835*, 2018.

[37] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[38] 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.

[39] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv*

preprint arXiv:2108.10470, 2021.

[40] Y. Lin, L. Xu, Y. Yu, J. Pang, and W. Zhang, "Unicon: A unified system for efficient robot learning transfers," *arXiv preprint arXiv:2601.14617*, 2026.