

7 Appendix

7.1 Training Details

7.1.1 Observation Space

The observation vector $\mathbf{s}_t \in \mathbb{R}^{187}$ consists of five main components: joint position history, proprioceptive feedback, privileged information, previous actions, and command signals. Table 2 summarizes each observation component. The student policy is trained without any privileged information

Observation Group	Observation Name	Dimension	Group Dim
History \mathbf{h}_t	Joint position history	72	72
Proprioception \mathbf{o}_t	Projected gravity	3	45
	Base linear velocity	3	
	Base angular velocity	3	
	Joint position	18	
	Joint velocity	18	
Privileged info \mathbf{p}_t	Feet contact state	4	32
	Static friction	4	
	Feet air time	4	
	Base external wrench	6	
	Base external push velocity	6	
	Base mass disturbance	1	
	End-effector external wrench	6	
	End-effector mass disturbance	1	
Previous action \mathbf{a}_{t-1}	Previous actions	18	18
Commands \mathbf{c}_t	Desired Base Velocity	3	20
	Desired End-Effector twist command	6	
	Desired End-Effector final goal pose	7	
	Desired Feet swing heights	4	

Table 2: Observation space for the teacher policy.

Observation Group	Observation Name	Noise range
History	Joint position history	$[-0.01, 0.01]$
Proprioception	Projected gravity	$[-0.01, 0.01]$
	Base linear velocity	$[-0.01, 0.01]$
	Base angular velocity	$[-0.1, 0.1]$
	Joint position	$[-0.01, 0.01]$
	Joint velocity	$[-0.2, 0.2]$
Previous actions	Previous actions	-
Commands	Desired Base Velocity	-
	Desired EE twist	-
	Desired Feet swing heights	-
	Desired Feet swing heights	-

Table 3: Observation space for the student policy and the range of noise applied during teacher-student distillation.

such that it can be deployed on hardware. During teacher-student distillation, uniform noise is added to the observations that are passed to the student policy. Table 3 summarizes the noise added to each observation group.

7.1.2 Rewards

The rewards are categorized into three groups: locomotion, manipulation, and contact schedule. For most quantities, we employ a Gaussian tracking reward function $\Phi(\mathbf{v}, \sigma^2) = \exp(-\mathbf{v}^T \mathbf{v} / \sigma^2)$. Table 4 summarizes the reward functions, while Table 5 describes the symbols used in these functions. We observe that multi-critic learning requires significantly less reward tuning compared to single-critic approaches. Unlike previous works [7, 27], our method avoids weighted averaging during advantage estimation, which further reduces the number of hyperparameters that require tuning.

Group	Reward Name	Reward Function	Weight
Loco	Base linear velocity	$\Phi(\hat{\mathbf{v}}_{b_{x,y}} - \mathbf{v}_{b_{x,y}}, 0.1)$	2.0
	Base angular velocity	$\Phi(\hat{\boldsymbol{\omega}}_{b_z} - \boldsymbol{\omega}_{b_z}, 0.05)$	2.0
	Torso height	$\Phi(\hat{h}_{b_z} - h_{b_z}, 0.1)$	0.5
	Base roll pitch angles	$\Phi(\boldsymbol{\theta}_{b_{xy}}, 0.1)$	0.1
	Torso linear velocity	$\Phi(\mathbf{v}_{b_z}, 0.2)$	0.5
	Torso roll pitch velocities	$\Phi(\boldsymbol{\omega}_{b_{xy}}, 0.2)$	2.5
	Is alive	$!z_{terminated}$	0.05
	Is terminated	$z_{terminated}$	-400.0
	Undesired robot contacts	$n_{contacts, robot}$	-1.0
	Robot action rate	$\Phi(\mathbf{a}_{t, robot} - \mathbf{a}_{t-1, robot}, 0.1)$	0.001
	Robot joint torque	$\Phi(\boldsymbol{\tau}_{t, robot}, 40.0)$	0.00001
	Robot joint velocity	$\Phi(\dot{\mathbf{q}}_{t, robot}, 4.0)$	0.0001
Mani	End-Effector position	$\Phi(r_{EE_t} - (r_{EE_{t-1}} + \hat{\mathbf{v}}_{EE} \cdot \Delta t), 0.005)$	5.0
	End-Effector orientation	$\Phi(\mathbf{R}_{EE_t} \boxminus (\mathbf{R}_{EE_{t-1}} \boxplus \hat{\boldsymbol{\omega}}_{EE} \cdot \Delta t), 0.01)$	4.0
	Undesired arm contacts	$n_{contacts, arm}$	-1.0
	Arm action rate	$\Phi(\mathbf{a}_{t, robot} - \mathbf{a}_{t-1, robot}, 0.5)$	0.1
	Arm joint torque	$\Phi(\boldsymbol{\tau}_{t, robot}, 40.0)$	0.00001
	Arm joint velocity	$\Phi(\dot{\mathbf{q}}_{t, robot}, 4.0)$	0.0001
CS	Feet Contact	$\Sigma_f (1 - C_f) \cdot \Phi(F_f, 1.0) \cdot \Phi(\hat{h}_z - h_z, 0.05) + \Sigma_f C_f \cdot n_{F_{f_z} > 1.0} \cdot \Phi(v_{f_{xy}}, 0.01)$	1.0
	Feet air time variance	$\text{Var}(t_{air_{t-2..t}}) + \text{Var}(t_{contact_{t-2..t}})$	1.0
	Feet air time	$\Sigma_{i=1}^4 n_{contacts, feet} \cdot t_{air_i}$	0.25

Table 4: Reward functions for training the teacher policy categorized into three groups, Loco (locomotion), Mani(manipulation and CS (contact schedule)).

7.1.3 Training Setup

We train the policy by simulating 4096 parallel agents using IsaacLab [26], a GPU-accelerated simulation framework. The training process incorporates a curriculum approach where agents begin on flat terrain and progressively advance to more challenging, moderately rough terrains as their performance improves. To facilitate effective zero-shot sim-to-real transfer, we implement extensive domain randomization across multiple physical parameters, as comprehensively documented in Table 6. We observe that randomization of foot friction and the application of external disturbances to the torso contribute to the development of stable walking behaviors and enhance the robustness of the policy during rapid arm movements. The external torso push velocity components are implemented by augmenting the measured torso velocity with random values for variable durations during simulation. This perturbation strategy teaches the policy to maintain and recover balance when faced with unexpected environmental disturbances.

The teacher policy architecture comprises an actor MLP network with 3 hidden layers ([512, 256, 128] units) and ReLU activations. The multi-critic implementation consists of 3 critic MLP networks with identical dimensions. The policy is trained using PPO [31], with the simulation

Symbol	Description
$\hat{v}_{b_{x,y}}$	Desired base linear velocity in x and y direction
$\hat{\omega}_{b_z}$	Desired base angular velocity in z direction
\hat{h}_{b_z}	Desired torso height
$\theta_{b_{xy}}$	Base roll and pitch angles
v_{b_z}	Torso linear velocity in z direction
$\omega_{b_{xy}}$	Torso roll and pitch velocities
$z_{terminated}$	Termination signal
$n_{contacts,robot}$	Number of undesired robot contacts
$n_{contacts,arm}$	Number of undesired arm contacts
$a_{t,robot}$	Robot action at time t
$\tau_{t,robot}$	Robot joint torques at time t
$\dot{q}_{t,robot}$	Robot joint velocities at time t
r_{EEt}	End-effector position at time t
R_{EEt}	End-effector orientation at time t
\hat{v}_{EE}	Desired end-effector velocity
$\hat{\omega}_{EE}$	End-effector angular velocity
C_f	Feet contact probability
F_f	Feet contact force
h_z	Feet height
$v_{f_{xy}}$	Feet velocity in x and y direction
t_{air_i}	Feet air time
$t_{contact_i}$	Feet contact time
$n_{F_{fz}>1.0}$	Number of feet force signal in z direction

Table 5: Description of symbols used in the reward function.

Disturbance	Range
Feet static friction	$[0.5, 1.2]$ N
Feet dynamic friction	$[0.3, 1.2]$ N
Torso mass	$[-10, 10]$ kg
End-effector mass	$[0, 1.8]$ kg
External force on torso	$[-50, 50]$ N
External torque on torso	$[-20, 20]$ Nm
External force on end-effector	$[-3, 3]$ N
Torso push linear velocity	$[-0.2, 0.2]$ m/s
Torso push angular velocity	$[-0.2, 0.2]$ rad/s

Table 6: Disturbances applied to the robot during training.

419 running at 400 Hz while the control policy operates at 50 Hz. We list the hyperparameters used for
420 training in Table 7.

421 The teacher policy has access to privileged information that is inaccessible in the real world. To
422 implement the policy on hardware, a student policy with the same network structure as the teacher is
423 distilled without access to privileged information using supervised learning [16]. During distillation,
424 the student learns to implicitly estimate the privileged information by minimizing the mean squared
425 error between the teacher and student actions. The hyperparameters used for training the student
426 policy are listed in Table 8.

Hyperparameter	Value
Learning rate	$3.0e - 4$
Entropy coefficient	0.002
Value loss coefficient	1.0
Clip parameter	0.2
Number of learning epochs	8
Number of mini-batches	4
Discount factor γ	0.99
GAE λ	0.95
Number of steps per environment	24
Number of environments	4096
Number of training iterations	50000

Table 7: Hyperparameters used for training the teacher policy.

Hyperparameter	Value
Learning rate	$1.0e - 3$
Number of learning epochs	8
Gradient length	15
Loss type	Mean Squared Error

Table 8: Hyperparameters used for training the student policy.

7.2 Ablation Studies

7.2.1 Effect of Curriculum Learning on End-Effector Twist Commands

Similar to previous works [7, 11] in pose-based end-effector control, we formulate the end-effector twist commands in the control frame, which is defined as the robot-centric gravity-aligned frame that follows only the yaw of the robot’s torso. However, as shown by the Control Frame signal in Figure 7, representing the twist command in the control frame makes it challenging for the policy to learn to precisely follow the end-effector trajectory. We observe that as the robot learns to stabi-

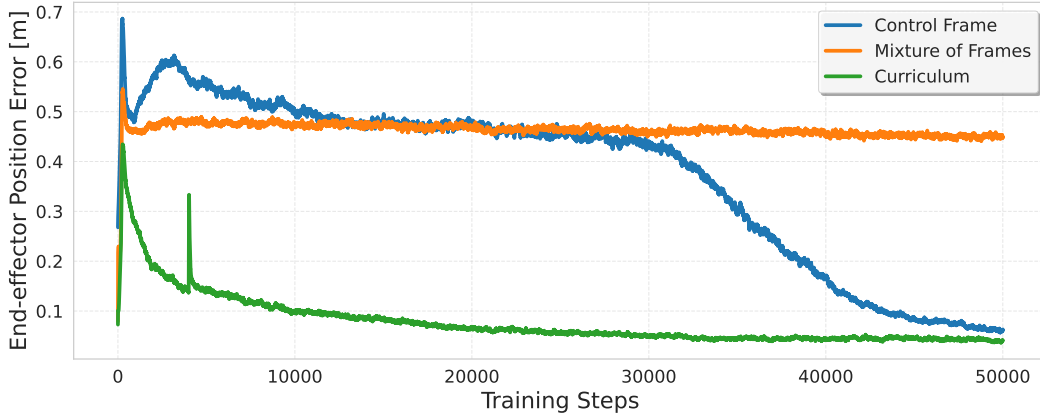


Figure 7: **End-effector position tracking for different command frame definitions:** The curriculum approach demonstrates superior performance compared to using either fixed frame representation or a mixture of frames.

lize its torso and walk, the control frame moves significantly, making the state-reward pairs sparse in representation and leading to difficulties in policy learning. As a solution, we attempted two approaches:

- **Mixture of frames:** In this approach, we provide 50% of the commands to the policy in the base frame, while representing the rest in the control frame.
- **Curriculum:** In this approach, we represent the commands in the base frame until a set number of iterations, after which the commands are provided with respect to the control frame.

In Figure 7, it can be seen that the mixture of frames does not lead to better end-effector position tracking. This could be due to the conflicting representations caused by the two frames in which

the commands are represented. In contrast, we observe that the curriculum approach leads to faster policy learning. In this case, we command the end-effector twist command with respect to the base frame until 3000 iterations, following which the frame switches to the control frame. The switching iteration is based on the observation that robots learn to walk with a stable torso within 3000 iterations. Although we see a jump in the tracking error soon after the frame switch, the tracking error continues to reduce for the rest of the training.

7.2.2 Single-Critic vs Multi-Critic Learning

In this section, we compare the performance of single-critic versus multi-critic learning approaches. We train policies with identical parameters, modifying only the critic architecture, while maintaining consistent reward functions and weights. As shown in Fig. 8, we observe an interesting trade-off: the single-critic policy achieves marginally lower end-effector tracking errors, but completely fails to learn locomotion behaviors. This superior end-effector performance can be attributed to the

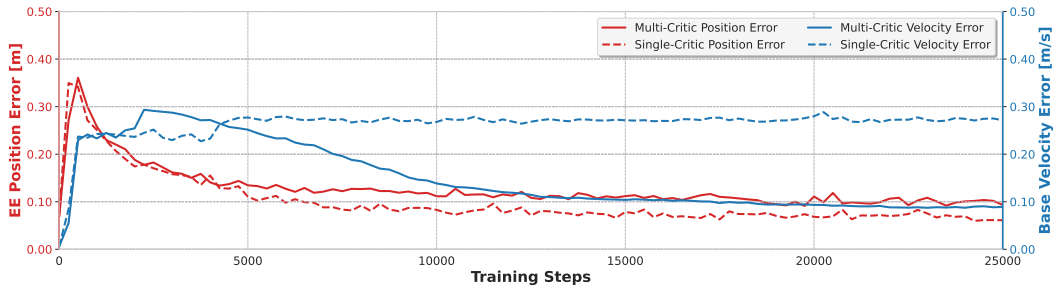


Figure 8: **Comparison of single-critic vs multi-critic frameworks:** Multi-critic policy learns to track both the base and end-effector commands whereas the single-critic policy only learns to track the end-effector commands.

policy that adopts a stationary posture at all times that essentially ignores base-velocity commands, allowing it to focus on the arm control task. In contrast, the multi-critic approach successfully learns to simultaneously satisfy both locomotion and manipulation objectives, demonstrating the ability to coordinate whole-body movement while maintaining precise end-effector motion.