Appendix

A. Reward Function

Reward items are listed in TABLE III. The task reward helps the robot adapt to the specific task scenarios, and the details of the design (4,6,7) is shown in II.B above. Where we set the Goal Velocity's reward coefficient $k_{positive}$ to 1 and the penalty coefficient $k_{negative}$ to -3. This set of parameters was optimized based on our comprehensive simulation experiments. Additionally, we designed the regularization reward referring to [46], adjusted it according to the type of robots and task scenarios.

TABLE III: Reward Function

Term	Equation	Weight
Task reward		
Goal Velocity	$r_{vel}(4)$	1.5
Yaw Angular Velocity	$exp(-4\lvert\omega_{yaw}^{cmd} - \omega yaw\rvert)$	0.5
Hip Position	$r_{Pos}(6)$	-0.5(ω_1=ω_2=0.5)
Collision	$r_{collision}(7)$	-10.0
Regularization reward		
Z Velocity	v_z^2	-0.5
X&Y Velocity	$\omega_x^2 + \omega_y^2$	-0.01
Dof Acceleration	$\sum_{i=1}^{12} \ddot{q}_i^2$	$-2.5 * 10^{-7}$
Action Rate	$\sqrt{\sum_{i=1}^{12}(a_t - a_{t-1})^2}$	-0.1
Delta Torques	$\sum_{i=1}^{12}(\tau_t - \tau_{t-1})^2$	$-1.0 * 10^{-7}$
Torques	$\sum_{i=1}^{12}(\tau_t)^2$	$-1.0 * 10^{-5}$
Dof Error	$\sum_{i=1}^{12}(q - q_{default})^2$	-0.04
Feet Stumble	$\lvert F_{feet}^{hor}\rvert > 4 * \lvert F_{feet}^{ver}\rvert$	-1
Dof Position Limits	$\sum_{i=1}^{12}(q_i^{out}{}_{q_i > q_{\max} \parallel q_i < q_{\min}})$	-10.0

B. Additional Training Details

Network Architecture Our learning framework's overall network architecture includes not only the components shown in Figure 2 but also some additional modules not depicted in the figure. The teacher policy consists of six Multilayer Perceptron (MLP) parts: collision domain encoder, collision estimator, privileged information encoder, privileged information estimator, velocity estimator, and teacher policy network. Where $\boldsymbol{h}_t$ denotes collision domain information and $\boldsymbol{g}_t$ denotes privilege information. The privileged information encoder supervises the privileged information estimator and optimizes learning using the ROA method. Privileged Estimator's network type is also a CNN with a similar structure to the Collision Estimator. The student policy includes five parts: the collision estimator, velocity estimator, and privileged information estimator from the teacher policy, along with the hybrid imagination model comprised of a Gated Recurrent Unit (GRU) network and an MLP-based student policy network. Table I provides more details on each layer.

Training course Course learning is crucial for robots to effectively travel obstacles in complex environments. Without this capability, robots would struggle to learn effectively. We have implemented the velocity-travel course training method[43], [46], where the robot's linear velocity is randomly sampled within the range of [0, 1]. If the robot's travel distance in one iteration exceeds half of the preset heading speed integral, the terrain difficulty is increased; otherwise, it is decreased. The levels of terrain difficulty are detailed in TABLE I.

TABLE IV: Network architectures

Module	**Inputs**	**Hidden Layers**	**Outputs**
Teacher policy			
Collision Domain Enc	$\boldsymbol{h}_t$	[256, 128, 64]	$\boldsymbol{p}_t$
Collision Estimator	$\boldsymbol{o}_{t-10}, \ldots, \boldsymbol{o}_{t-1}, \boldsymbol{o}_t$	/	$\hat{\boldsymbol{c}}_t$
Privileged Encoder	$\boldsymbol{g}_t$	[64, 32]	$\boldsymbol{e}_t$
Privileged Estimator	$\boldsymbol{o}_{t-5}, \ldots, \boldsymbol{o}_{t-1}, \boldsymbol{o}_t$	/	$\hat{\boldsymbol{e}}_t$
Velocity Estimator	$\boldsymbol{o}_t$	[128, 64]	$\hat{\boldsymbol{v}}_t$
Teacher Network	$\boldsymbol{o}_t, \boldsymbol{p}_t, \hat{\boldsymbol{c}}_t, \boldsymbol{e}_t, \hat{\boldsymbol{v}}_t$	[512, 256, 128]	$\boldsymbol{a}_t$
Student policy			
Hybrid Imagination Model	$\boldsymbol{o}_{t-10}, \hat{\boldsymbol{c}}_{t-10} \ldots, \hat{\boldsymbol{c}}_t, \boldsymbol{o}_t$	/	$\hat{\boldsymbol{p}}_t$
Student Network	$\boldsymbol{o}_t, \hat{\boldsymbol{p}}_t, \hat{\boldsymbol{c}}_t, \hat{\boldsymbol{e}}_t, \hat{\boldsymbol{v}}_t$	[512, 256, 128]	$\hat{\boldsymbol{a}}_t$

Hyperparameter	Value
Discount Factor	0.99
GAE Parameter	0.95
Timesteps per Rollout	21
Epochs per Rollout	5
Minibatches per Epoch	4
Entropy Bonus (α_2)	0.01
Value Loss Coefficient (α_1)	1.0
Clip Range	0.2
Reward Normalization	yes
Learning Rate	2e$-$4
# Environments	4096
Optimizer	Adam

TABLE V: PPO hyperparameters.

Policy training and Imitation training We use PPO with hyperparameters listed in TABLE V to train the teacher policy. We regard the process of teacher policy supervising student policy learning as imitation learning process, a_{t} and $\hat{a}_{\mathrm{t}}$ are the action vectors from the teacher and student respectively in an actor network. The gradient of the loss $\mathcal{L}$ can be defined as the sum of the squared norms of the difference between a_{t} and $\hat{a}_{\mathrm{t}}$ over all actions:

$$\mathcal{L}_{Imitation} = \left\|\pi_{\text{teacher}}(\cdot|s) - \pi_{\text{student}}(\cdot|s)\right\|_2^2$$

Here, $\|\cdot\|_2$ denotes the L^2 norm. Where $\pi_{\text{teacher}}(\cdot|s)$ is the action from the Teacher policy and $\pi_{\text{student}}(\cdot|s)$ is the action from the Student policy .