Supplementary Material
Helpful DoggyBot: Open-World Object Fetching using Legged Robots and Vision-Language Models

Qi Wu[1] Zipeng Fu[1] Xuxin Cheng[2] Xiaolong Wang[2] Chelsea Finn[1]
[1]Stanford University [2]UC San Diego
https://helpful-doggybot.github.io/

I. Details of Whole-Body Controller

A. Details of Simulation Environment

To ensure robust performance across diverse scenarios for the expert policy, we use Isaac Gym Preview 4 to train 6144 robots in 400 terrains. We introduce a curriculum learning approach which generates 10 different levels of stair heights in simulation. The criteria of updating the curriculum in training is the proportion of the terrain each episode the robot finishes. We randomly generate these environments for each training episode, varying parameters like stair height, number of stairs, and terrain friction as shown in Table I. We then distilled a policy with 384 robots in simulation with real-time depth image rendering. For Oracle policy, we trained 20k iterations in 10 hours on a single GeForce RTX 4090 GPU. For distilled policy, we trained 5k iterations in 6 hours.

Parameters	Values
num of envs	6144
num of vision envs	384
num of terrains	400
num of difficulty levels	10
stair height	[0, 0.65]
stair per env	[0, 6]
stair width	[0.8, 3]
stair length	[1.5, 2]
goal y range	[-0.1,0.1]
terrain noise	[0.02, 0.06]
friction	[0.2, 2]
curriculum up threshold	0.8*total length
curriculum down threshold	0.5*total length

TABLE I: **Environment and terrain setup**

B. Details of Domain Randomization

To further promote generalization and ensure robust performance in real world application, we employ domain randomization. We uniformly sample values in Table II to change the robots' dynamics and perturbations, enabling it to bridge the sim2real gap.

Parameters	Values
push interval (s)	8
max push vel xy (m/s)	0.5
max push vel z (m/s)	0.5
added mass range (kg)	[0., 3.]
added com range (m)	[-0.2, 0.2]
motor strength range	[0.8, 1.2]
action delay(s)	[0 0.02]
vision delay(s)	0.1
vision position rand(m)	0.005
vision angle rand(degree)	[24,34]

TABLE II: **Domain randomization**

C. Details of Reward Function

To enhance learning efficiency and policy performance, we employed reward shaping techniques. Auxiliary rewards were introduced to promote balance maintenance, energy minimization, and smooth transitions between locomotion modes such as walking, climbing, and tilting. The specific reward terms are listed in Table III

D. Details of Deployment

Depth images were captured using a Realsense D435 camera connected to the Nvidia Jetson Orin via a USB 3.0 interface. We applied hole-filling filters, spatial filters, and temporal filters, followed by resizing and normalization—mirroring the process used in simulation. The depth encoder network operates at 10 Hz with a fixed delay and communicates with the main process via UDP. The main process executes the distilled policy at 50 Hz, while proprioceptive data is obtained at 500 Hz through Cyclone DDS. Computed joint angles and PD parameters are transmitted to the Unitree low-level controller via ROS 2 messages, where motor torques are calculated using the internal PD controller.

II. Details of Zero-shot Deployment Using VLMs

As the robot approaches the target object, a transition to a precise grasping policy is triggered, allowing for more accurate command following. This transition is governed by the overhead camera and occurs when the robot is within 1 meter of the target object and oriented within 30 degrees of it. After successful grasping, the policy switches again once the robot is aligned within 30 degrees of the termination point.

reward	expression	scale
tracking goal vel	$\min\left(\frac{\vec{v}\cdot\hat{\vec{t}}}{v_{cmd}+10^{-5}}, \frac{v_{cmd}}{v_{cmd}+10^{-5}}\right), \quad \text{where } \hat{\vec{t}} = \frac{\vec{t}}{\|\vec{t}\|+10^{-5}}$	1.5
tracking yaw vel	$\exp\left(-\lvert\omega_z - \omega_{cmd}\rvert\right)$	1.
tracking pitch	$\exp\left(-3\lvert p_{cmd} - p\rvert\right)$	1.5
lin vel z walking	v_z^2	-9.0
ang vel xy	$\sum \omega_{xy}^2$	-0.05
dof acc	$\sum\left(\frac{\dot{q}_{t+1}-\dot{q}_t}{\Delta t}\right)^2$	-2.5e-7
collision	$\sum \mathbf{1}\left(\lvert\lvert f_{\text{contact}}\rvert\rvert > 0.1\right)$	-5.
action rate	$\lvert\lvert \mathbf{a}_{t+1} - \mathbf{a}_t \rvert\rvert$	-0.1
delta torques	$\sum(\tau_{t+1} - \tau_t)^2$	-1.0e-7
torques	$\sum \tau^2$	-0.00001
hip pos	$\sum(q_{\text{hip}} - q_{\text{hip, default}})^2$	-1
dof error	$\sum(q - q_{\text{default}})^2$	-0.2
feet stumble	$\mathbf{1}\left(\lvert\lvert f_{\text{contact, xy}}\rvert\rvert > 4 \cdot \lvert f_{\text{contact, z}}\rvert\right)$	-5
feet edge	$\sum \mathbf{1}(\text{feet at edge})$	-1
feet drag	$\sum\left(\mathbf{1}(\text{contact}) \cdot \lvert\lvert v_{\text{xy}}^{\text{feet}}\rvert\rvert\right)$	-0.1
energy	$\lvert\lvert \tau \cdot \dot{q} \rvert\rvert$	-1e-3

TABLE III: **Reward terms**