Supplementary Material: Deep Whole-Body Control: Learning a Unified Policy for Manipulation and Locomotion

Zipeng Fu[*†] **Xuxin Cheng**[*] **Deepak Pathak**
Carnegie Mellon University

1 Experiment Videos

We perform thorough real-world analysis of our framework and our custom-built legged manipulator. We urge the reader to look at the compiled result videos at https://maniploco.github.io . As we can see in the video, legs and arm function in coordination with each other where legs bend and stretch to increase the reach of the arm as well as to attain stability.

2 Regularized Online Adaptation Details

Algorithm 1 Regularized Online Adaptation

Randomly initialize privileged information encoder μ, adaptation module ϕ, unified policy π
Initialize with empty replay buffer D
for $itr = 1, 2, \ldots$ **do**
  for $i = 1, 2, \ldots, N_{env}$ **do**
    $s_0, e_0 \leftarrow$ envs[i].reset()
    for $t = 0, 1, \ldots, T$ **do**
      if itr mod $H == 0$ **then**
        $z_t^{\phi} \leftarrow \phi(s_{t-10:t-1}, a_{t-11:t-2})$
        $a_t \leftarrow \pi((s_t, a_{t-1}, z_t^{\phi}))$
      else
        $z_t^{\mu} \leftarrow \mu(e_t)$
        $a_t \leftarrow \pi((s_t, a_{t-1}, z_t^{\mu}))$
      end if
      $s_{t+1}, r_t \leftarrow$ envs[i].step(a_t)
      Store $((s_t, e_t), a_t, r_t, (s_{t+1}, e_{t+1}), z_t^{\phi}, z_t^{\mu})$ in D
    end for
  end for
  if itr mod $H == 0$ **then**
    Update θ_{ϕ} by optimizing $||\text{sg}[z_t^{\mu}] - z_t^{\phi}||_2$
  else
    Update $\theta_{\pi}, \theta_{\mu}$ by optimizing $-J(\theta_{\pi}, \theta_{\mu}) + \lambda||z_t^{\mu} - \text{sg}[z_t^{\phi}]||_2$, where $J(\theta_{\pi}, \theta_{\mu})$ is the advantage mixing RL objective in Section 2.1 of the main paper
  end if
  Empty D
  $\lambda \leftarrow$ Linear_Curriculum(itr)
end for

[*]equal contribution, [†]Zipeng Fu is now at Stanford University

We presented the details of Regularized Online Adaptation (Section 2.2 of the main paper) in Algorithm 1. We set H to be 20. The regularization coefficient λ follows a linear curriculum which starts at 0 and stops at 1: $\lambda = \min(\max(\frac{itr-5000}{5000}, 0), 1)$.

3 Simulation Details

We obtained URDF files for the quadraped and the robot arm from Unitree and Interbotix separately. We customized the URDF files to connect the two parts rigidly. Shown in Figure 1, we use Nvidia's IsaacGym [1] for parallel simulation. We use fractal noise to generate the terrain. The parameters for

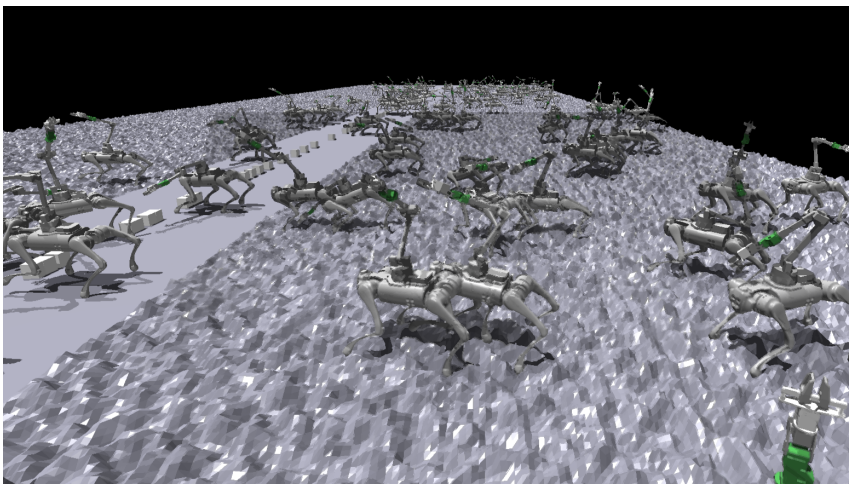

Figure 1: Customized simulation environment based on IsaacGym

the fractual noise are number of octaves = 2, fractal lacunarity = 2.0, fractal gain = 0.25, frequency = 10Hz, amplitude = 0.15m. We found that the generated rough terrain will enforce foot clearance and replace the complex rewards that are needed if flat terrain is used for simulation [2].

We sample an EE position command by first sampling a spherical coordinate (l, p, y) from Table 2 of the main paper. Then world coordinate of p^{end} is obtained as $T(\text{S2C}[(l, p, y)]) + (p_x^{\text{base}}, p_y^{\text{base}}, p_z^{\text{base}})$, where T is the linear transformation according to the base orientation, S2C[] is the operator to transform spherical coordinates to Cartesian coordinates, and p^{base} is the base position. To encourage smooth arm motion and whole-body coordination, we set p_z^{base} to be a constant (0.53) and row and pitch in T to be 0, so EE position commands are z, row, pitch-independent of the base.

We simulate each episode for a maximum of 1000 steps and terminate the episode earlier if the height of the robot drops below 0.28m, body roll angle exceeds 0.2 radians if EE position command if on the left of the body base ($p_{\text{y}}^{\text{cmd}} > 0$) , or is less than -0.2 radians if EE position command is on the right of the body base ($p_{\text{y}}^{\text{cmd}} < 0$), or the body pitch exceeds 0.2 radians if EE position command is above body base ($p_{\text{p}}^{\text{cmd}} > 0$), or is less than -0.2 radians if EE position command is below body base ($p_{\text{p}}^{\text{cmd}} < 0$). We do not early terminate if the arm self-collide and any body parts with the terrain, but the EE command positions are sampled in a way that

The control frequency of the policy is 50Hz, and the simulation frequency is 200Hz. We set the stiffness (K_p) for leg joints and arm joints to be 50 and 5 respectively and the damping (K_d) to be 1 and 0.5 respectively. The default target joint positions for leg joints are $[-0.1, 0.8, -1.5, 0.1, 0.8, -1.5, -0.1, 0.8, -1.5, 0.1, 0.8, -1.5]$ and for arm joints are zeros. The delta range of target joint positions for leg joints is 0.45 and for arm joints are $[2.1, 1.0, 1.0, 2.1, 1.7, 2.1]$.

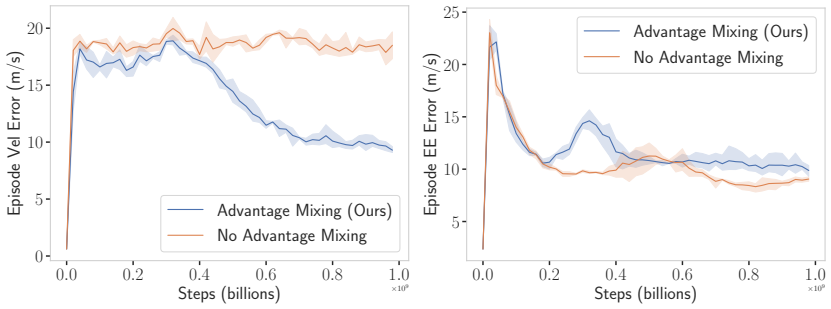

Figure 2: Advantage mixing helps the unified policy to learn to walk and grasp at the same time. Without Advantage Mixing, the unified policy fails to learn to walk where the Episode Vel Error (episodic sum of L1 error between velocity commands and current velocities) is constantly high. In this case, the unified policy stays at local minima of only following EE commands.

4 Training Details

The policy is a multi-layer perceptron which takes in the current state $s_t \in \mathbb{R}^{75}$, which is concatenated with the environment extrinsics $z_t \in \mathbb{R}^{20}$. The first hidden layer has 128 dimensions and after that the network splits into 2 heads, where each has 2 hidden layers of 128 dimensions. The outputs of two heads are concatenated, where the leg actions $a_t^{\text{leg}} \in \mathbb{R}^{12}$ and arm actions $a_t^{\text{arm}} \in \mathbb{R}^{6}$. We train for 10000 iterations / training batches, which are 2 billions of samples and 200k gradient updates. We list the hyperparameters of PPO [3] in Table 1 of the Supplementary.

5 Advantage Mixing Details

For a policy with diagonal Gaussian noise and a sampled transition batch D, the training objective with respect to policy's parameters θ_π is

$$
\begin{aligned}
J(\theta_\pi) &= \frac{1}{|\mathcal{D}|} \sum_{(s_t,a_t)\in\mathcal{D}} \log \pi\left(a_t \mid s_t\right) A(s_t, a_t) \\
&= \frac{1}{|\mathcal{D}|} \sum_{(s_t,a_t)\in\mathcal{D}} \log \Big(\pi(a_t^{\text{arm}} \mid s_t)\pi(a_t^{\text{leg}} \mid s_t)\Big) \left(A^{\text{manip}}(s_t, a_t) + A^{\text{loco}}(s_t, a_t)\right) \\
&\rightarrow \frac{1}{|\mathcal{D}|} \sum_{(s_t,a_t)\in\mathcal{D}} \log \pi(a_t^{\text{arm}} \mid s_t) \left(A^{\text{manip}} + \beta A^{\text{loco}}\right) + \log \pi(a_t^{\text{leg}} \mid s_t) \left(\beta A^{\text{manip}} + A^{\text{loco}}\right)
\end{aligned}
$$

In Figure 2 of the Supplementary, we plot the episodic velocity command following error (Episode Vel Error) and EE comand following error (Episode EE Error) against number of steps during training. Advantage mixing helps the unified policy to learn to walk and grasp at the same time. Without Advantage Mixing, the unified policy fails to learn to walk where the Episode Vel Error (episodic sum of L1 error between velocity commands and current velocities) is constantly high. In this case,

Table 1: Training Hyper-parameters

PPO clip range	0.2
Learning rate	2e-4
Reward discount factor	0.99
GAE λ	0.95
Number of environments	5000
Number of environment steps per training batch	40
Learning epochs per training batch	5
Number of mini-batches per training batch	4
Minimum policy std	0.2

the unified policy stays at local minima of only following EE commands by not exploring in leg action space, since the initial exploration phase in leg action space will destabilize the base which harms manipulation tasks.

6 Real-World Setup and Experiment Details

Table 2: Camera Parameters for Vision Tracking

Resolution	640×400
Frequency	10 Hz
Tag/Cam offset	(-0.02, -0.03, 0.12)

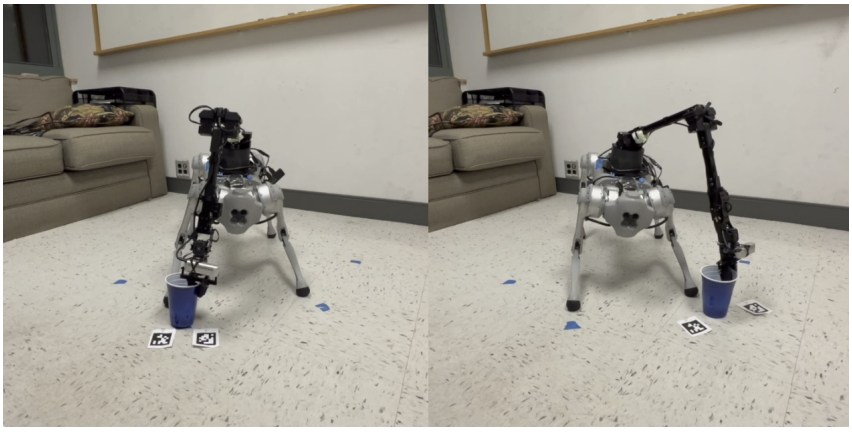

Figure 3: Vision-guided tracking by using the average pose of the two AprilTags as the target pose.

	Ground Points (p^{end})	Success Rate ↑	TTC ↓	IK Failure Rate ↓	Self-Collision Rate ↓
Easy tasks (tested on 3 points)					
Ours	$\begin{bmatrix}(0.62,-1.27,-1.11)\\(0.57,-1.16,0.55)\\(0.58,-1.14,1.78)\end{bmatrix}$	**0.8**	**5s**	-	**0**
MPC+IK		0.3	17s	0.4	0.3
Hard tasks (tested on 5 point)					
Ours	$\begin{bmatrix}(0.72,-0.51,0.34)\\(0.55,-0.75,-0.43)\\(0.56,-0.73,0.5)\\(0.45,-0.74,1.80)\\(0.45,-0.76,-1.8)\end{bmatrix}$	**0.8**	**5.6s**	-	**0**
MPC+IK		0.1	$22.0s$	0.2	0.5

Table 3: Comparison of our method v.s. MPC+IK on pick-up tasks. p^{end} is the goal position sampled from the points on the ground. TTC is the average time to tompletion. All data are averaged on 10 real-world trials.

The robot platform is comprised of a Unitree Go1 quadraped [4] with 12 actuatable DoFs, and a robot arm which is the 6-DoF Interbotix WidowX 250s [5] with a parallel gripper. We mount the arm on top of the quadruped. The RealSense D435 provides RGB visual information and is mounted close to the gripper of WidowX. Both power of Go1 and WidowX (60 Watts) are provided by Go1's battery.

In real-world experiments, we directly deploy the unified policy with the adaptation module with weights fixed onto the onboard computation of Go1, both modules operate at 50Hz. The inference of policy and adaption module are done on Raspberry Pi 4. The software stack of the WidowX 250s arm is setup on Nvidia TX2 by using the official codebase at `https://github.com/Interbotix/interbotix_ros_manipulators`. UDP is used as the communication protocal between Pi and TX2. EE gripper closing and opening are not a part of the policy.

In teleoperation experiments, the gripper action is directly controlled by a joystick controller. In vision-guided tracking experiments, we use a scripted policy to control the gripper: when the gripper

position is close to the desired position specified by the AprilTag [6] for 1 second, the gripper closes; otherwise, it keeps open.

We listed the camera parameters used in vision-guided tracking in Table 2 of the Supplementary. The "Tag/Cam offset" describes what the desired translation of the tag should be viewed in the camera frame when using the position controller to specify desired end-effector position in spherical coordinate. Shown in Figure 3 of the Supplementary, we also performed additional experiments on vision-guided tracking suggested by Reviewer bkQw by using two AprilTags and averaging their pose to get the target pose. Video results are at here. We listed the positions of ground points for visual-guided tracking tasks in Table 3. More results on hard tasks are at here.

References

[1] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and Gavriel State. Isaac gym: High performance GPU-Based physics simulation for robot learning. Aug. 2021.

[2] 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*, 2019.

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

[4] X. Wang. Unitree go1. https://www.unitree.com/products/go1/.

[5] WidowX 250 robot arm 6DOF - X-Series robotic arm. https://www.trossenrobotics.com/widowx-250-robot-arm-6dof.aspx.

[6] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.