西北工業大學
NORTHWESTERN POLYTECHNICAL UNIVERSITY

Humanoid Control

Locomotion

Teleoperation

Motion Tracking

Zhengwei Yang

2026.07.01

Major Concern

Model-based: MPC, LQR, …

- Strong interpretability & safety
- Data efficiency
- Suitable for structured problems
- Modeling difficulty
- Struggle with real-world uncertainties
- High computation payload at inference

Learning-based: RL, IL, E2E, …

- Offer robustness to uncertainties
- Less modeling effort but complex reward design
- Strong perception integration & multi-modal
- Data hungry
- Poor interpretability
- High training cost of computation and time

Pros (+)
Cons (-)

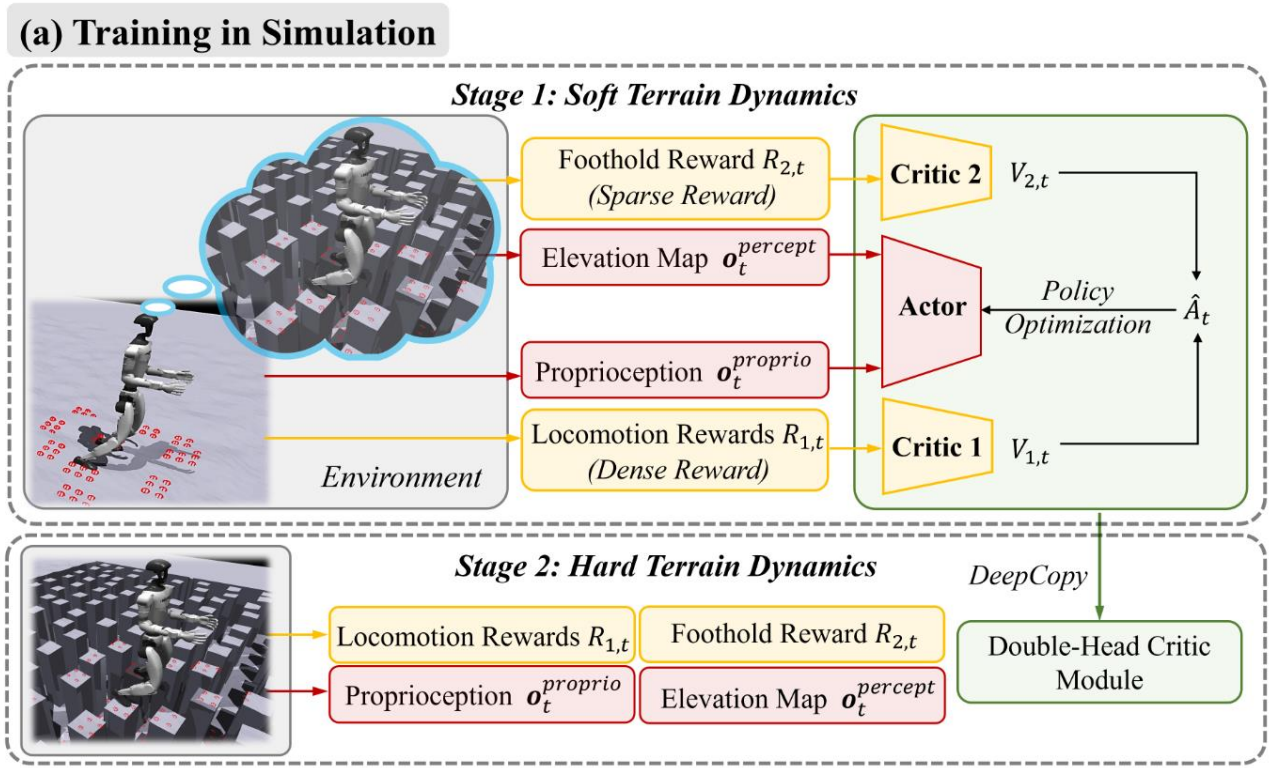
Respective Concern

- How to achieve agile and stable locomotion in reality? —— BeamDojo, Attn-Based Map & ASAP
- How to produce human-like and versatile motions? —— BeyondMimic, Homie & KungfuBot
- How to scale humanoid control? —— SONIC

Question —— Solution

How to achieve agile and stable locomotion in reality?

BeamDojo: Learning Agile Humanoid Locomotion on Sparse Footholds



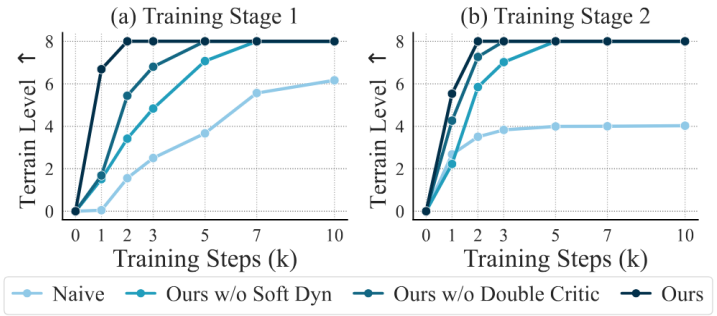
Double Critic

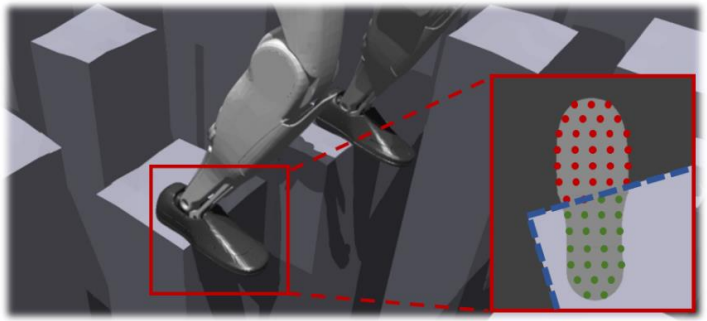
$$\hat{A}_t = w_1 \cdot \frac{\hat{A}_{1,t} - \mu_{\hat{A}_{1,t}}}{\sigma_{\hat{A}_{1,t}}} + w_2 \cdot \frac{\hat{A}_{2,t} - \mu_{\hat{A}_{2,t}}}{\sigma_{\hat{A}_{2,t}}}$$

Two-stage RL Framework

Encourage sufficient trial-and-error exploration from flat terrain to actual terrain

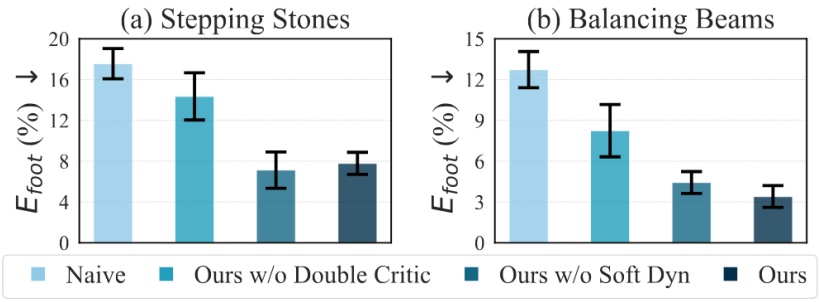
Ablation Study





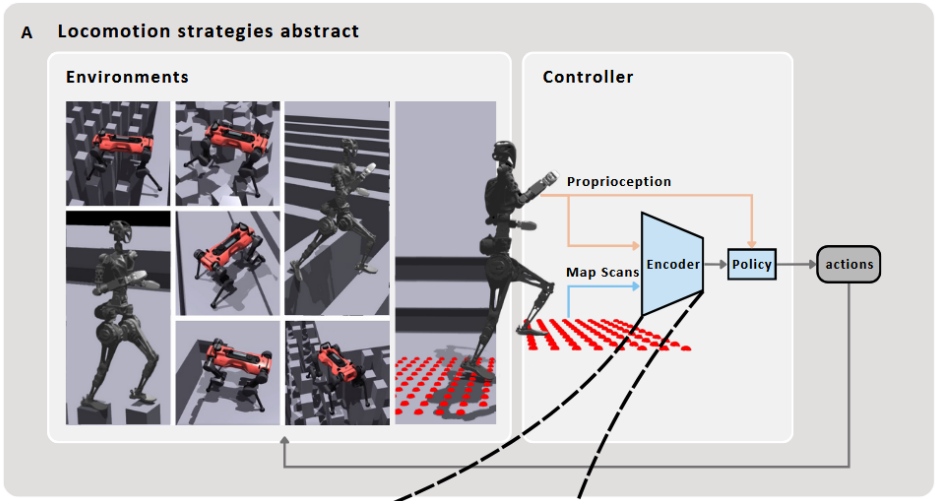
Sampling-based Foothold Reward

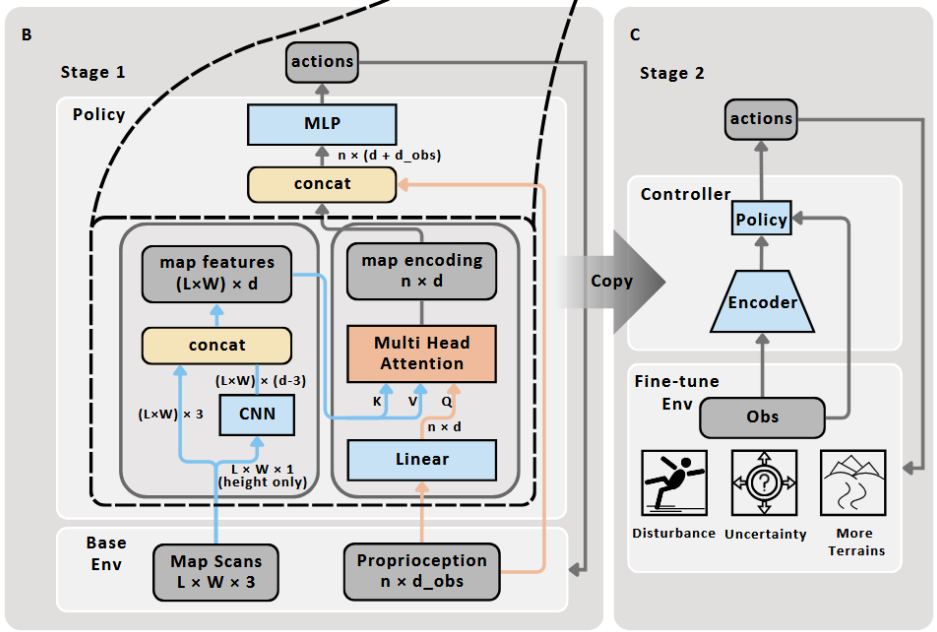
$$r_{\text{foothold}} = -\sum_{i=1}^{2} \mathbb{C}_i \sum_{j=1}^{n} \mathbb{1}\{d_{ij} < \epsilon\}$$



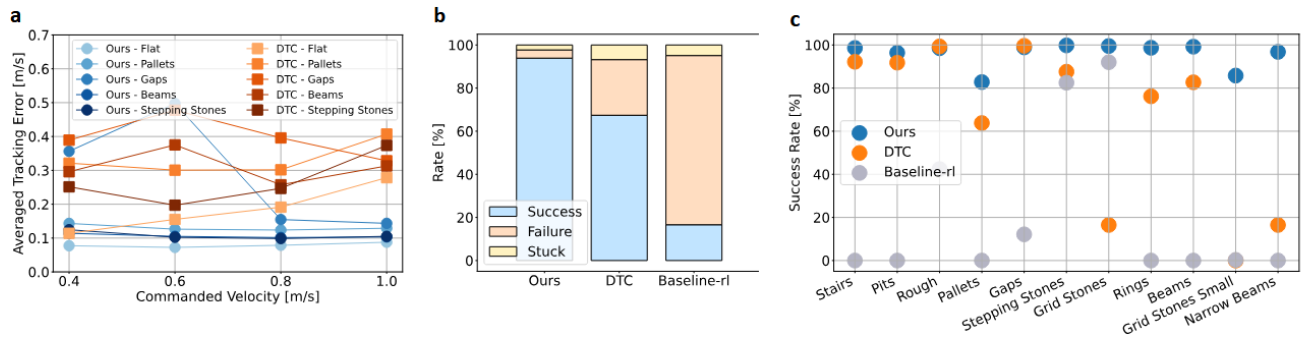
Science Robotics
AAAS

Attention-Based Map Encoding for Learning Generalized Legged Locomotion

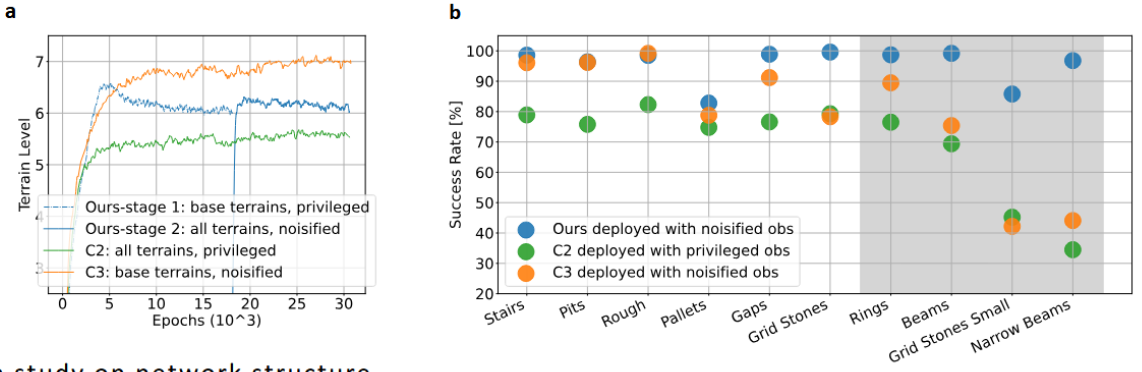




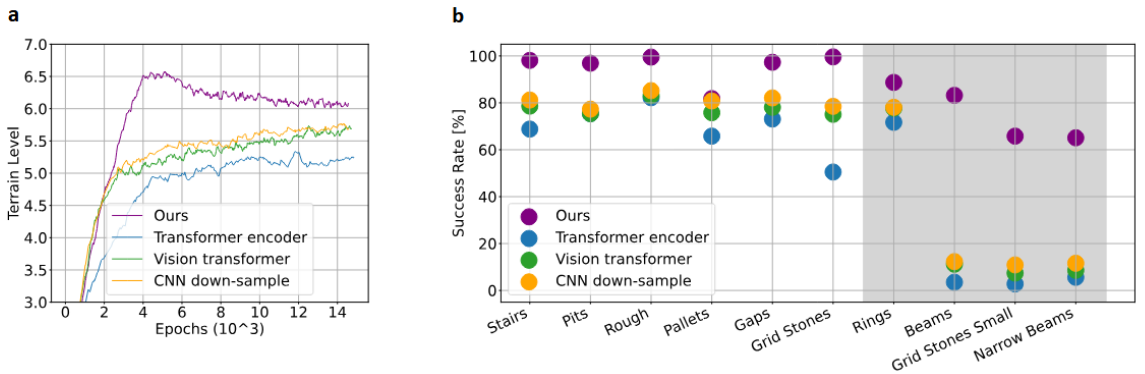
A. Benchmark with DTC and Baseline RL



B. Ablation study on two-stage training



C. Ablation study on network structure

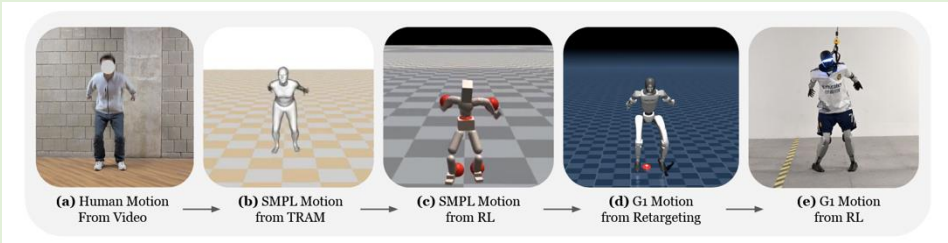


ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills

Pre-training

Acquiring basic abilities

Data Generation



$\mathcal{D}_{\text{SMPL}}$ $\mathcal{D}_{\text{SMPL}}^{\text{Cleaned}}$ $\mathcal{D}_{\text{Robot}}^{\text{Cleaned}}$

Phase-based Motion Tracking Policy Training

- Asymmetric Actor-Critic Training
- Termination Curriculum of Tracking Tolerance
- Reference State Initialization
- Reward Terms: Penalty + Regularization + Reward
- Domain Randomizations

Post-training

Aligning sim-to-real gap

Data Collection

$$\mathcal{D}^r = \{s_0^r, a_0^r, \dots, s_T^r, a_T^r\}$$

↓

Training Delta Action Model

$$\Delta a_t = \pi_\theta^\Delta(s_t, a_t)$$

$$s_{t+1} = f^{\text{sim}}(s_t, a_t^r + \Delta a_t)$$

↓

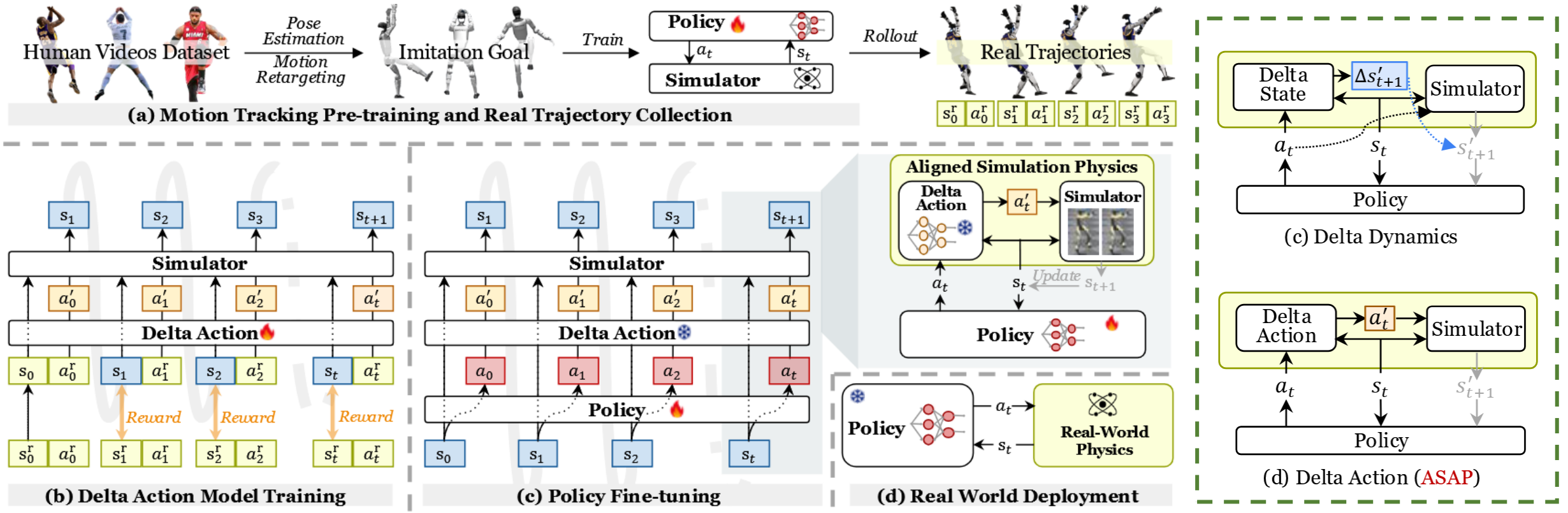
Fine-tuning Motion Tracking Policy under New Dynamics

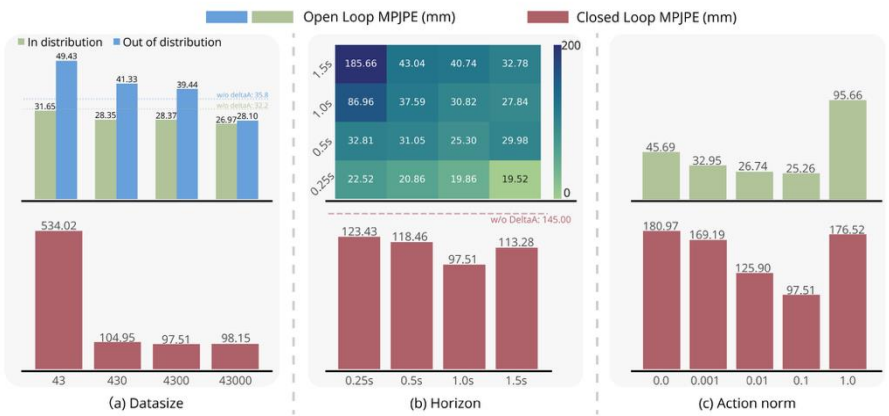
$$s_{t+1} = f^{\text{ASAP}}(s_t, a_t) = f^{\text{sim}}\left(s_t, a_t + \pi^\Delta(s_t, a_t)\right)$$

↓

Policy Deployment

Overview of ASAP





Analysis on key factors

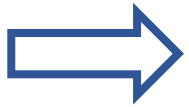
- Open-loop testing measures one-step prediction accuracy
- Closed-loop testing evaluates long-term stability and robustness under real-world execution

Advantages of action correction vs. state correction

- Physical consistency
- Less parameters
- Simulator serves as low pass filter

How to produce human-like and versatile motions?

- Heavy domain randomization
- Heavy reward regularization
- Complex observations



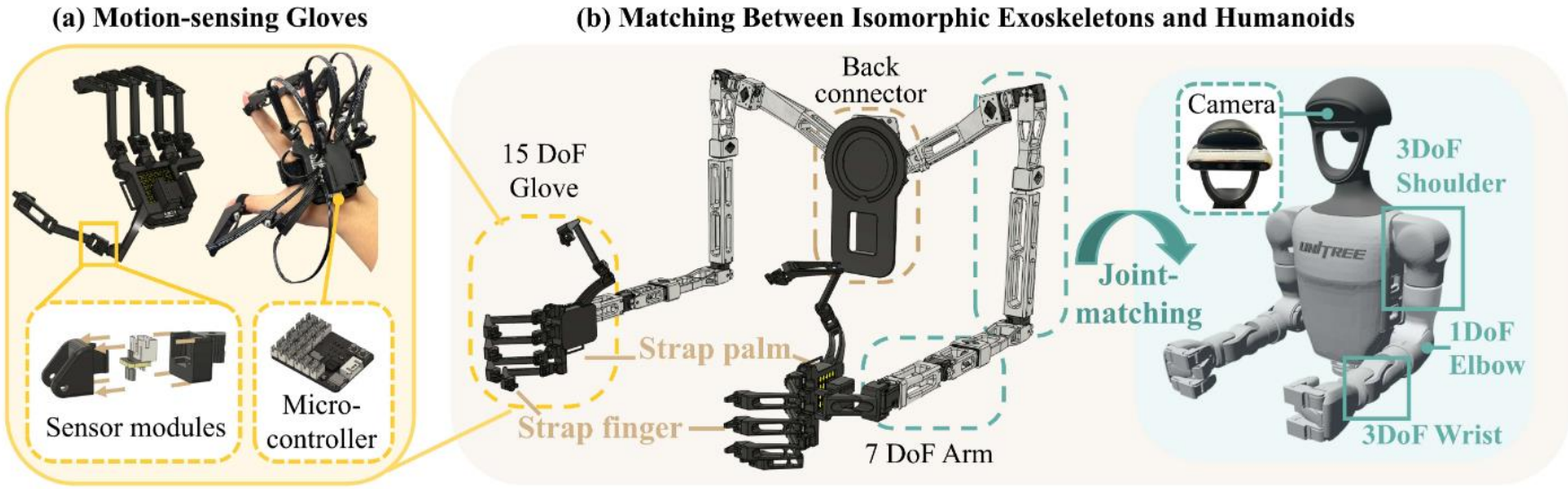
- same model parameters, reward function, and hyperparameters across a diverse set of skills and different physics engines

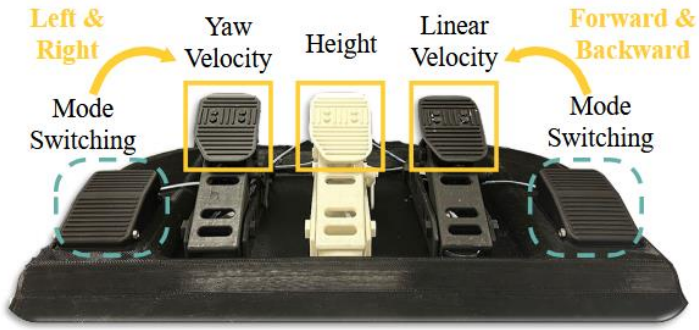
$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\varepsilon(\mathbf{z}|\mathbf{x})}[\|\mathbf{x} - \mathcal{D}(\mathbf{z})\|^2] + \beta D_{\text{KL}}(q_\varepsilon(\mathbf{z}|\mathbf{x}) \| \mathcal{N}(\mathbf{0}, \mathbf{I}))$$

$$\mathcal{L}_{\text{Diffusion}} = \mathbb{E}\left[\left\| z_\phi(\mathbf{z^k}, k) - \mathbf{z^0} \right\|^2\right]$$

$$\mathbf{z^{k-1}} = \alpha_k \left(\mathbf{z^k} - \gamma_k \left(\mathbf{z^k} - z_\phi(\mathbf{z^k}, k) \right) \right) + \sigma_k \mathcal{N}(0, \mathbf{I})$$

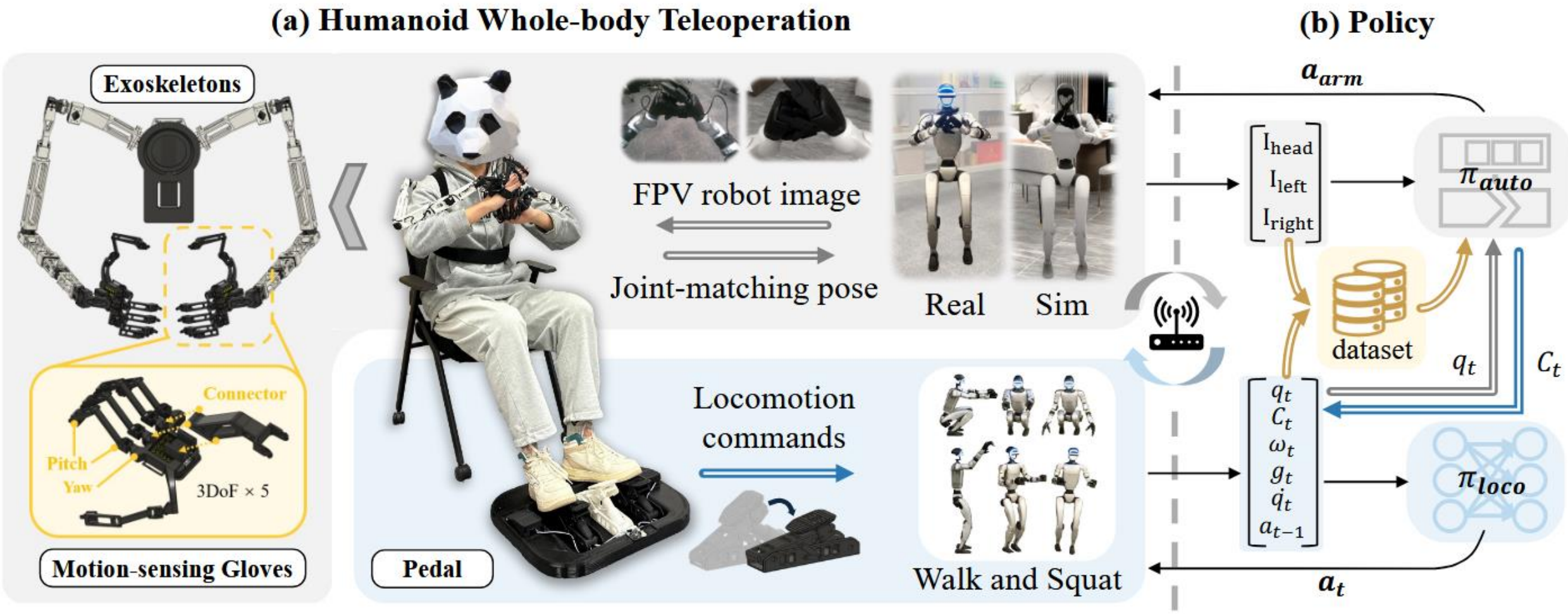
HOMIE: Humanoid Loco-Manipulation with Isomorphic Exoskeleton Cockpit





Hardware of Semi-autonomous Teleoperation System

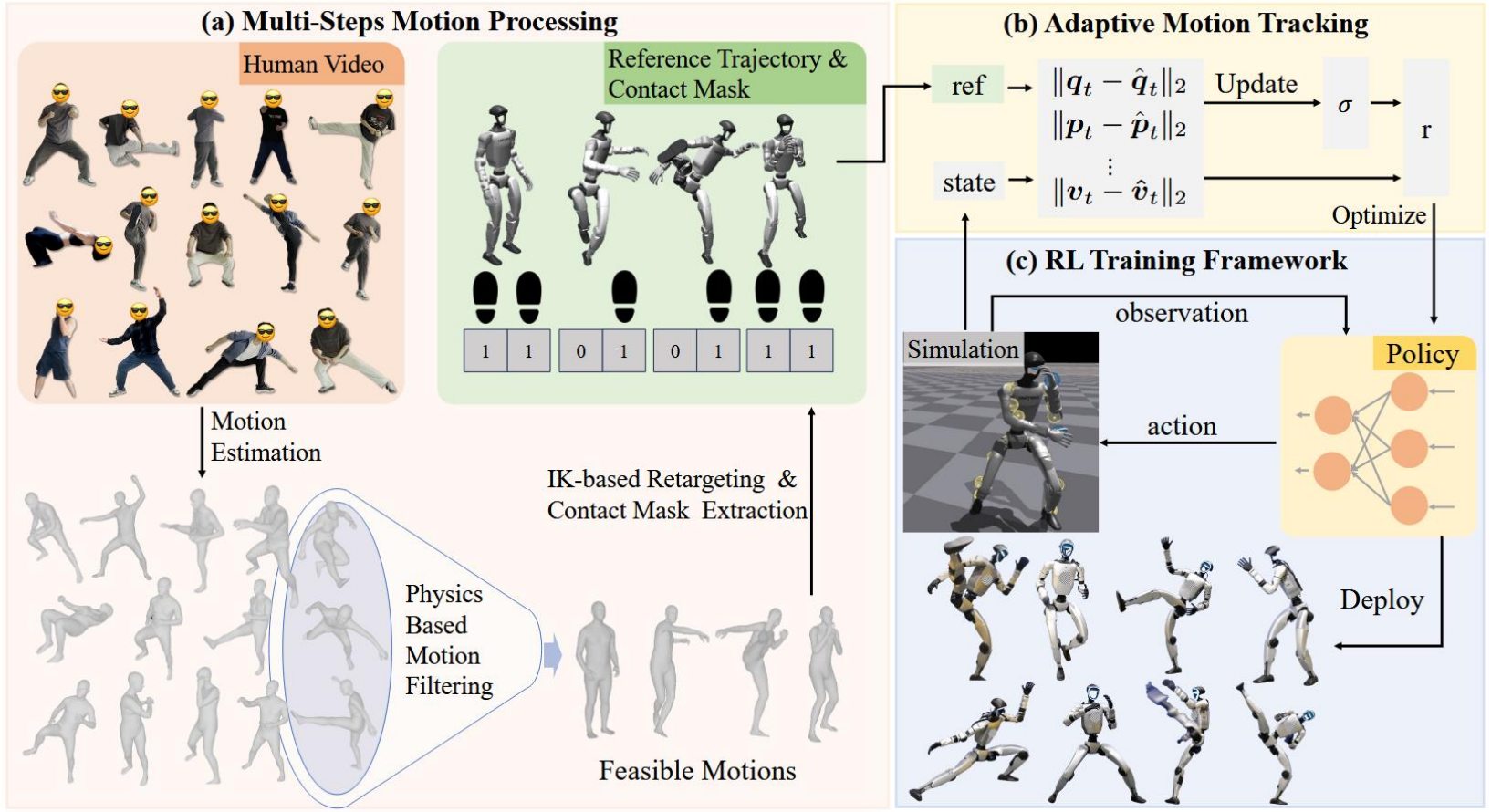
- Gloves utilizing Hall sensors: freely adapt to any model of dexterous hands
- Exoskeleton: delivers faster and more precise arm control by eliminating the reliance on inverse dynamics
- The pedal design: removes the need for continuous synchronized walking between the operator and the robot

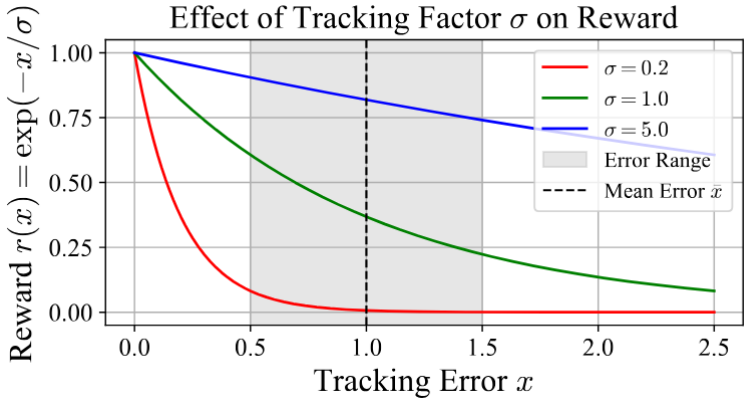


- Upper-body Pose Curriculum for training π_{loco}
- Height Tracking Reward: $r_{knee} = -\left\|(h_{r,t} - h_t) \times \left(\frac{q_{knee,t} - q_{knee,min}}{q_{knee,max} - q_{knee,min}} - \frac{1}{2}\right)\right\|$
- Symmetry Utilization

NEURAL INFORMATION PROCESSING SYSTEMS

KungfuBot: Physics-Based Humanoid Whole-Body Control for Learning Highly-Dynamic Skills





$$J^{\text{in}}(\boldsymbol{x}, \sigma) = \sum_{i=1}^{N} \exp(-x_i/\sigma)$$

$$\max_{\boldsymbol{x} \in \mathbb{R}_+^{\mathbb{N}}} J^{\text{in}}(\boldsymbol{x}, \sigma) + R(\boldsymbol{x})$$

$$\max_{\sigma \in \mathbb{R}_+} J^{\text{ex}}(\boldsymbol{x}^*),$$
$$\text{s.t.} \quad \boldsymbol{x}^* \in \arg\max_{\boldsymbol{x} \in \mathbb{R}_+^{\mathbb{N}}} J^{\text{in}}(\boldsymbol{x}, \sigma) + R(\boldsymbol{x}).$$

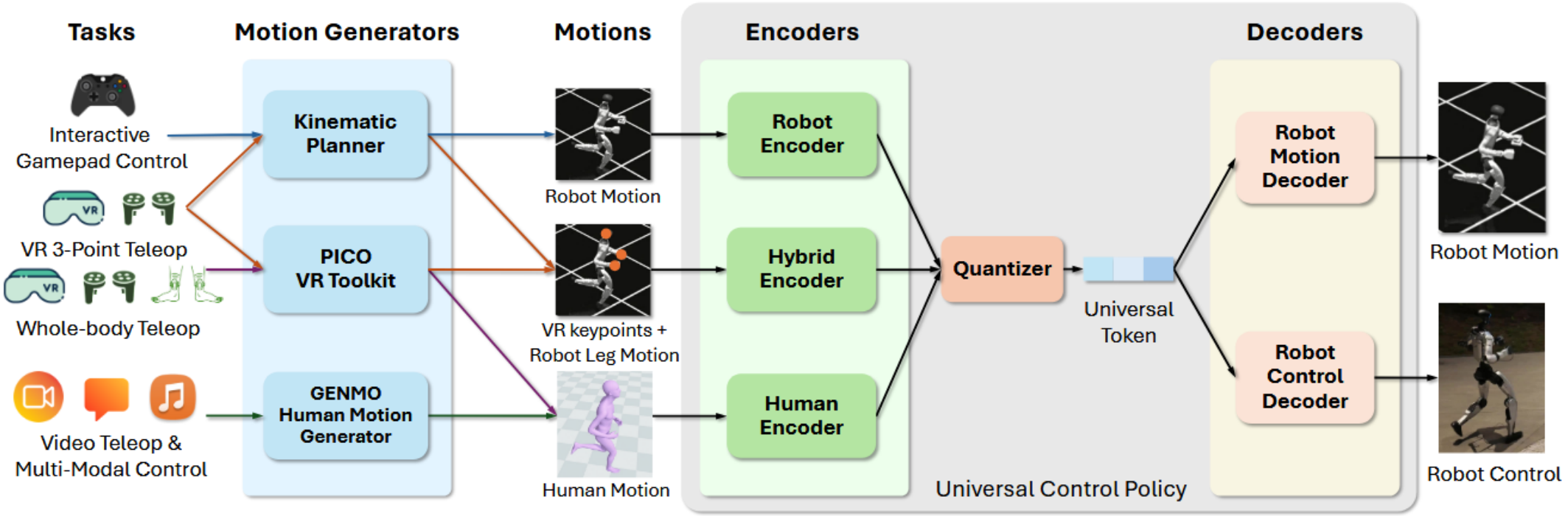
$$\Delta d_t = \left\| \overline{\boldsymbol{p}_t^{\text{CoM}}} - \overline{\boldsymbol{p}_t^{\text{CoP}}} \right\|_2 < \epsilon_{\text{stab}}$$

- Boundry stability
- Maximum instability gap < threshold

$$\sigma^* = \left(\sum_{i=1}^{N} x_i^* \right) / N.$$

How to scale humanoid control?

SONIC: Supersizing Motion Tracking for Natural Humanoid Whole-Body Control

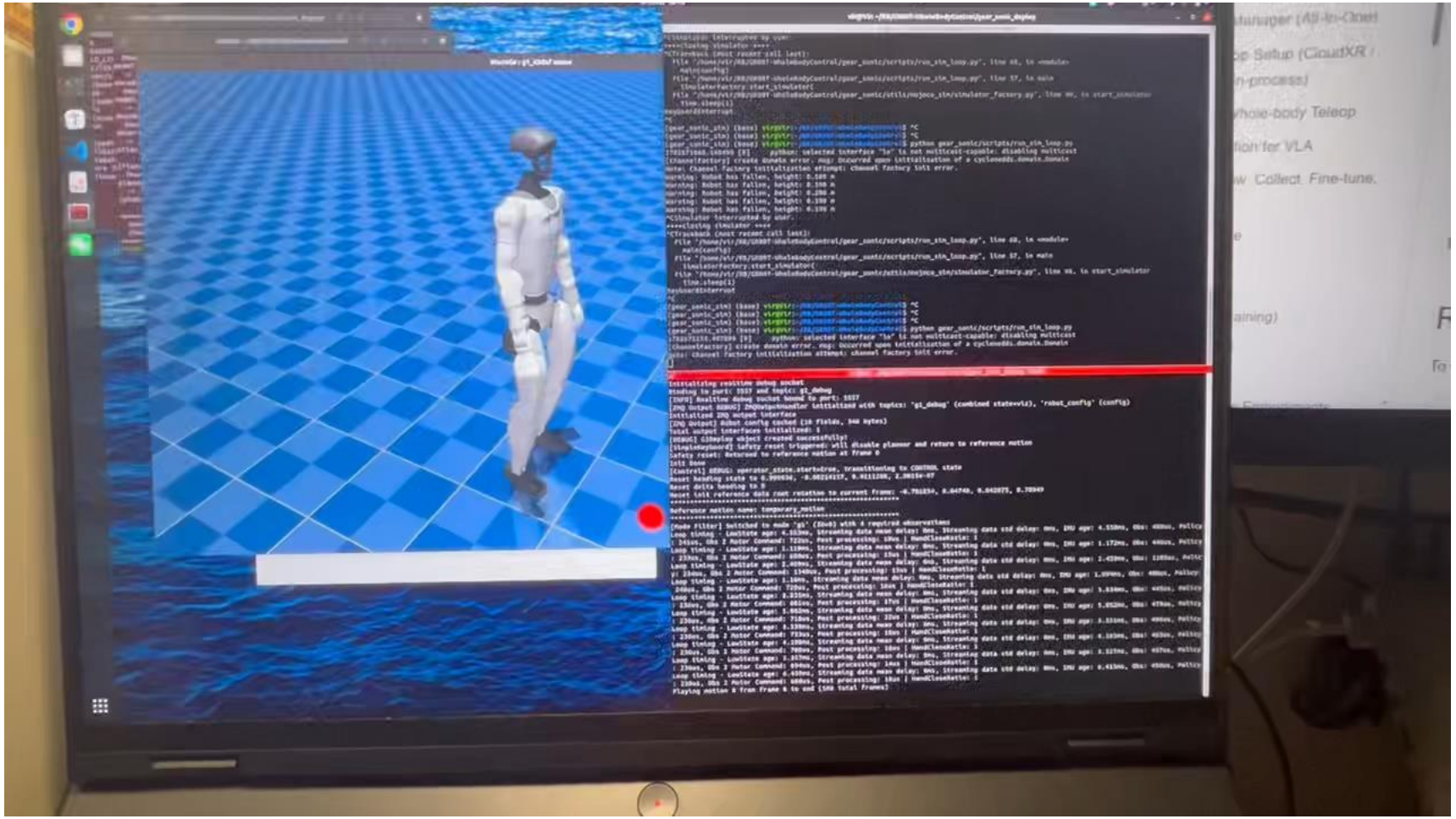


$$\mathrm{L} = \mathcal{L}_{\text{ppo}} + \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{token}} + \mathcal{L}_{\text{cycle}}$$

$$\mathcal{L}_{\text{recon}} = \|\mathcal{D}_r(z_r) - \boldsymbol{g_r}\|^2 + \|\mathcal{D}_r(\boldsymbol{z_h}) - \boldsymbol{g_r}\|^2 + \|\mathcal{D}_r(\boldsymbol{z_m}) - \boldsymbol{g_r}\|^2$$

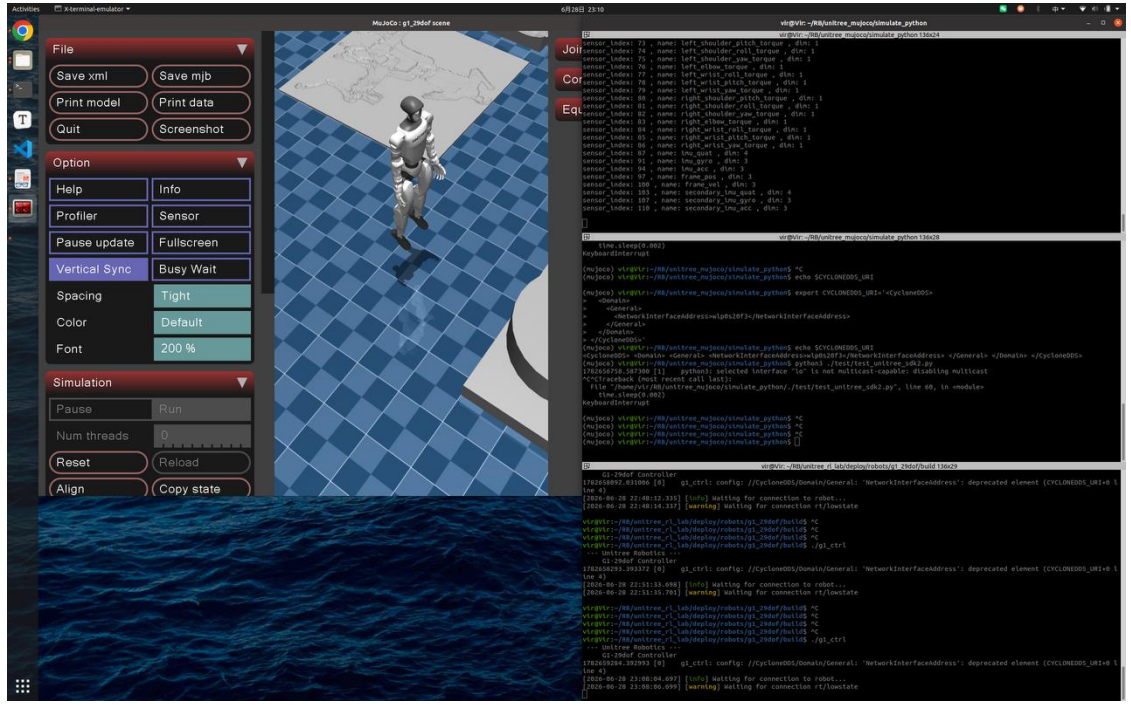
$$\mathcal{L}_{\text{token}} = \|z_r - z_h\|^2 + \|z_r - z_m\|^2 + \|z_m - z_h\|^2$$

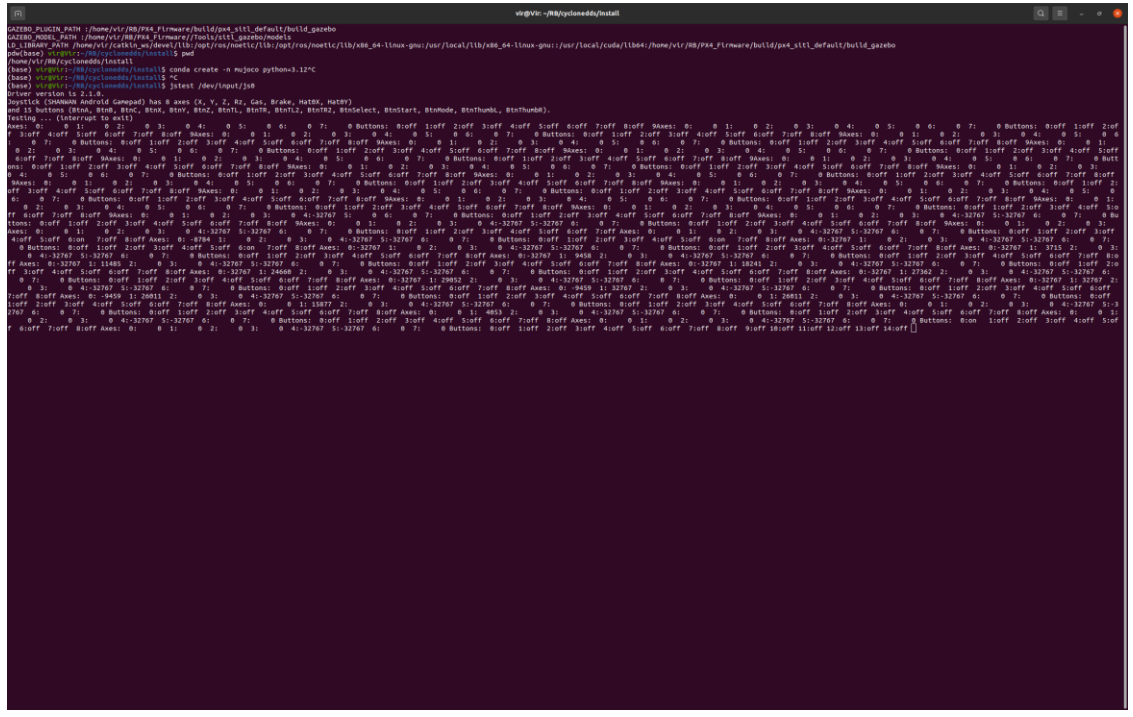
$$\mathcal{L}_{\text{cycle}} = \left\|\mathcal{E}_r\left(\mathcal{D}_r(z_h)\right) - z_r\right\|^2$$

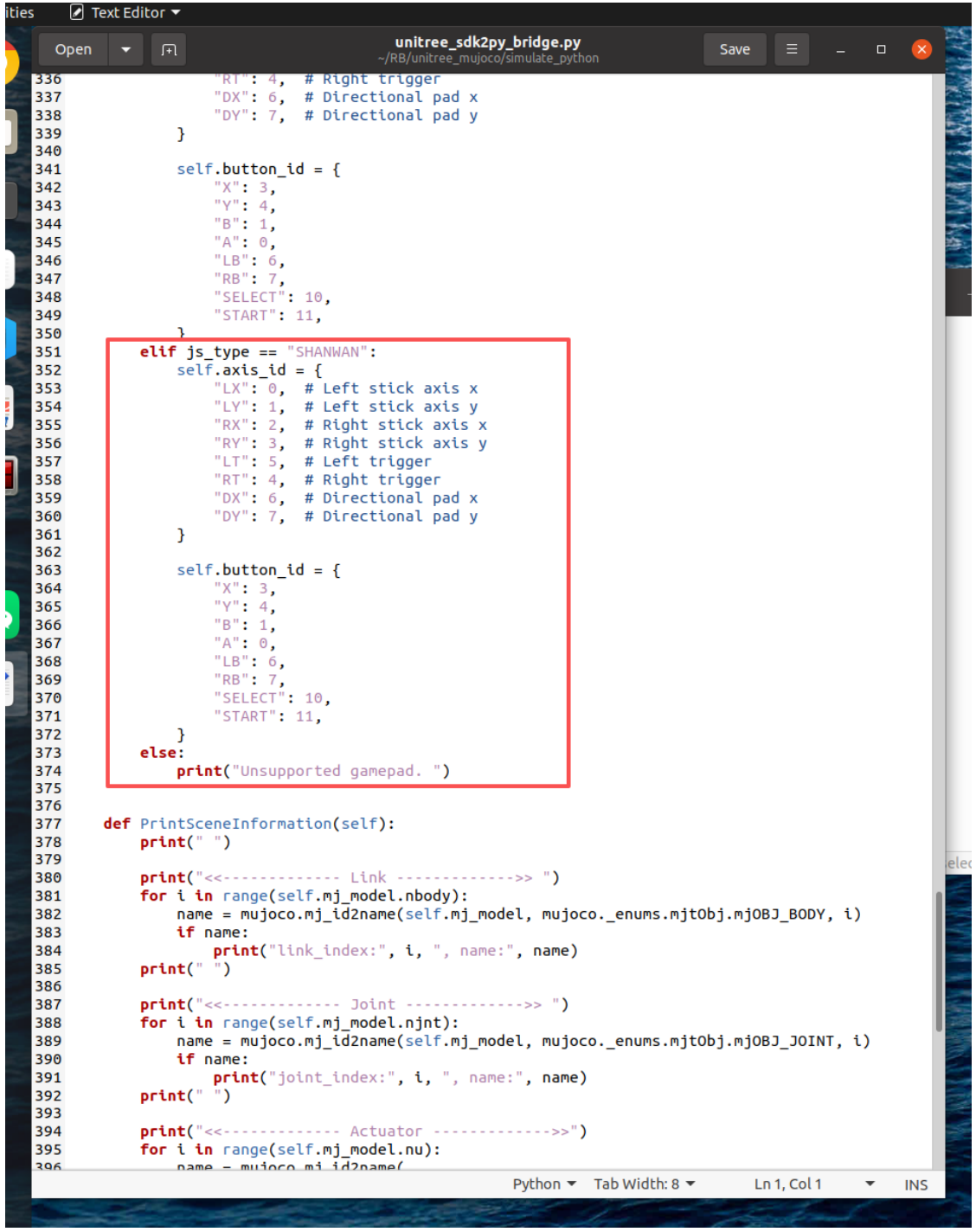


SONIC

Modify src and scripts before reproducing


File
Save xml
Save mjb
Print model
Print data
Quit
Screenshot
Option
Help
Info
Profiler
Sensor
Pause update
Fullscreen
Vertical Sync
Busy Wait
Spacing
Tight
Color
Default
Font
200 %
Simulation
Pause
Run
Num threads
0
Reset
Reload
Align
Copy state




Text Editor
Open
unitree_sdk2py_bridge.py
~/RB/unitree_mujoco/simulate_python
Save
"RT": 4, # Right trigger
"DX": 6, # Directional pad x
"DY": 7, # Directional pad y
}
self.button_id = {
"X": 3,
"Y": 4,
"B": 1,
"A": 0,
"LB": 6,
"RB": 7,
"SELECT": 10,
"START": 11,
}
elif js_type == "SHANWAN":
self.axis_id = {
"LX": 0, # Left stick axis x
"LY": 1, # Left stick axis y
"RX": 2, # Right stick axis x
"RY": 3, # Right stick axis y
"LT": 5, # Left trigger
"RT": 4, # Right trigger
"DX": 6, # Directional pad x
"DY": 7, # Directional pad y
}
self.button_id = {
"X": 3,
"Y": 4,
"B": 1,
"A": 0,
"LB": 6,
"RB": 7,
"SELECT": 10,
"START": 11,
}
else:
print("Unsupported gamepad. ")
def PrintSceneInformation(self):
print(" ")
print("<<------------- Link ------------->> ")
for i in range(self.mj_model.nbody):
name = mujoco.mj_id2name(self.mj_model, mujoco._enums.mjtObj.mjOBJ_BODY, i)
if name:
print("link_index:", i, ", name:", name)
print(" ")
print("<<------------- Joint ------------->> ")
for i in range(self.mj_model.njnt):
name = mujoco.mj_id2name(self.mj_model, mujoco._enums.mjtObj.mjOBJ_JOINT, i)
if name:
print("joint_index:", i, ", name:", name)
print(" ")
print("<<------------- Actuator ------------->>")
for i in range(self.mj_model.nu):
Python
Tab Width: 8
Ln 1, Col 1
INS

Thanks for listening!

Zhengwei Yang

2026.07.01