
基于深度强化学习的四足机器人控制器的迁移

汇报人：杨鹏志 (Pengzhi Yang) 导师：周城老师

目录

• Reinforcement Learning in Control Tasks	3
• Sim-to-real Gap	4
• Domain Randomization	5
• Domain Adaptation	
Introduction	6
Illustration of our approach	9
Loss Function	10
• Results	
Gazebo	11
Real-world	15
• Reference	16

Advantages:

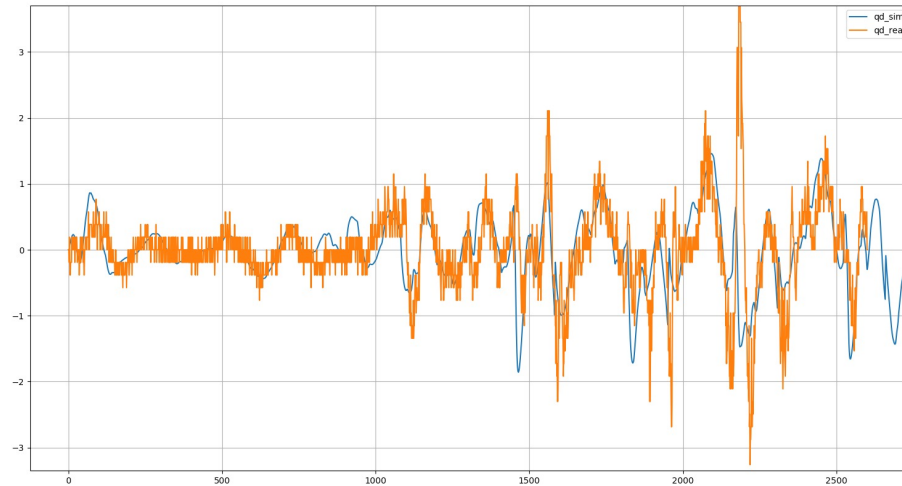
1. Model free: do not rely on a model of conditional control system
2. Generality: can handle nonlinear and stochastic dynamics

Disadvantages:

1. Data inefficiency
2. Sim-to-real Gap

Buşoniu, Lucian, et al. "Reinforcement learning for control: Performance, stability, and deep approximators." *Annual Reviews in Control* 46 (2018): 8-28.

1. Noise from the sensors in real-world environments (data processing)
2. Discrepancies between the dynamics of the simulation and the real world (domain randomization and adaptation)



Peng, Xue Bin, et al. "Learning agile robotic locomotion skills by imitating animals." *arXiv preprint arXiv:2004.00784* (2020).

Randomly initialize the values of environment's dynamic parameters to train a robust control policy for the robot.

And according to the experiments, we chose

- (1) Lateral friction, body mass (f , m : stay the same in one episode)
- (2) Torque damping (τ : changes rapidly from step to step)

to randomize during training.

In the future, randomize more dynamic parameters for a more robust results.

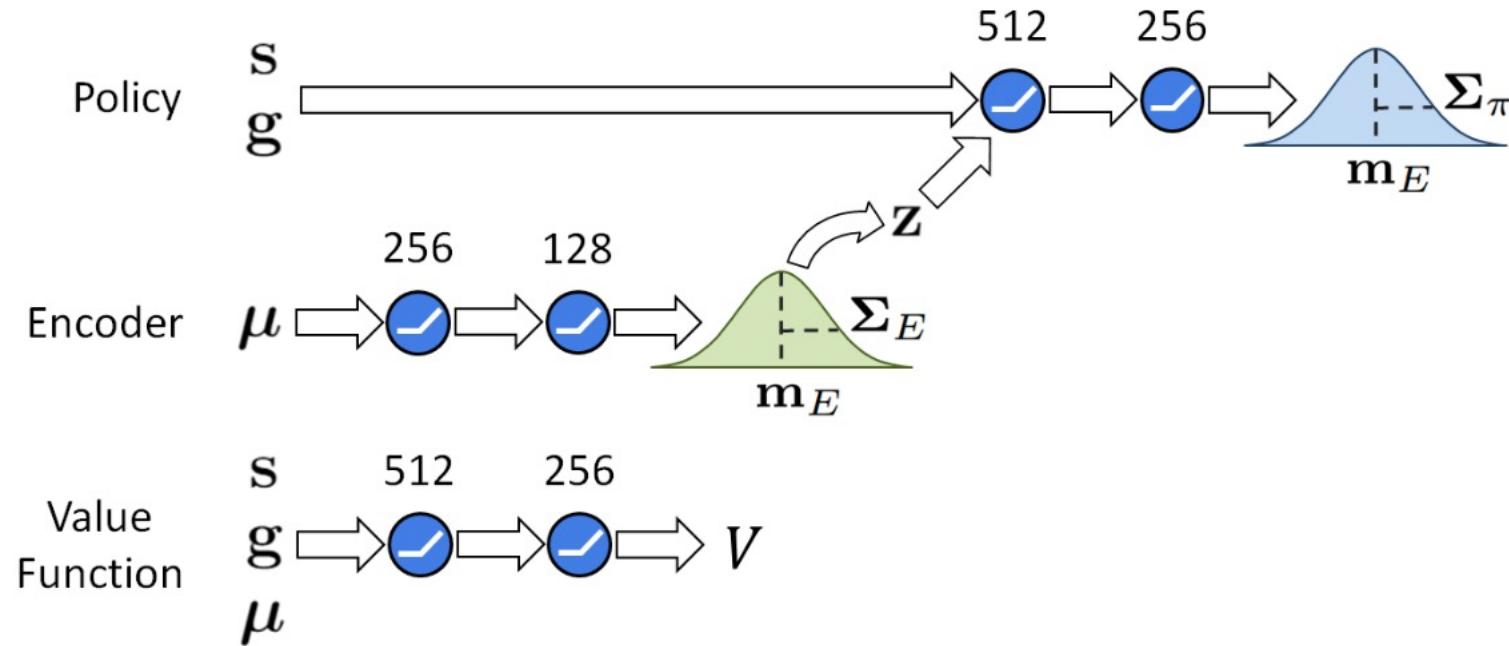
Peng, Xue Bin, et al. "Sim-to-real transfer of robotic control with dynamics randomization." *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018.

Domain randomization trades optimality for robustness leading to an over conservative policy.



Feed the dynamic parameters into the network during training to train a policy which could handle different dynamic environments better.

Luo, Jingru, and Kris Hauser. "Robust trajectory optimization under frictional contact with iterative learning." Autonomous Robots 41.6 (2017): 1447-1461.



Domain randomization trades optimality for robustness leading to an over conservative policy.



Feed the dynamic parameters into the network during training to train a policy which could handle different dynamic environments better.

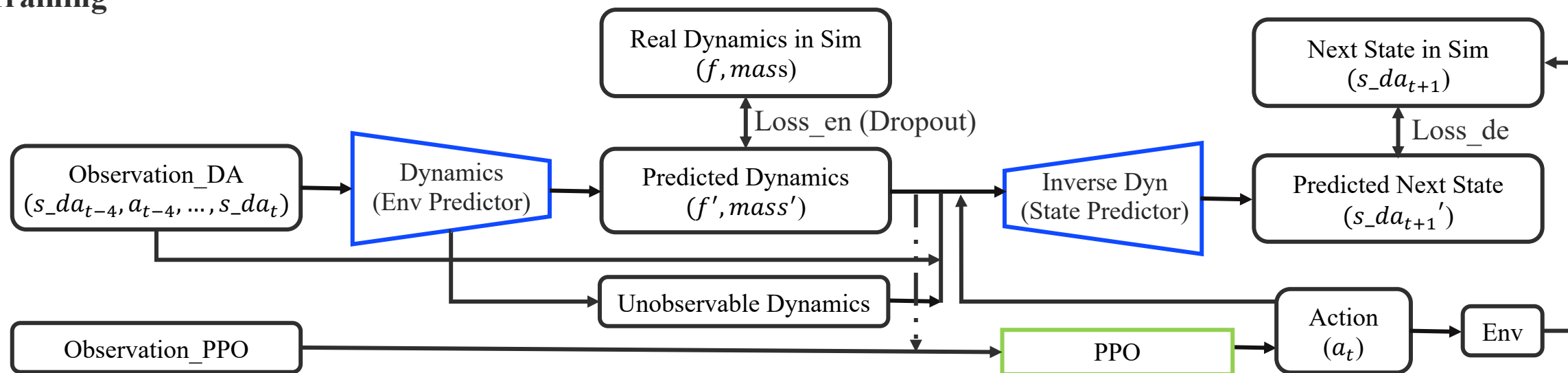


We use a series of state, action data: $\tau = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_2, \dots, \mathbf{s}_{T-1}, \mathbf{a}_{T-1}, \mathbf{s}_T\}$ to predict environment's dynamic parameters. Then feed the predicted dynamic parameters together with PPO observation into PPO network to give a adaptable policy.

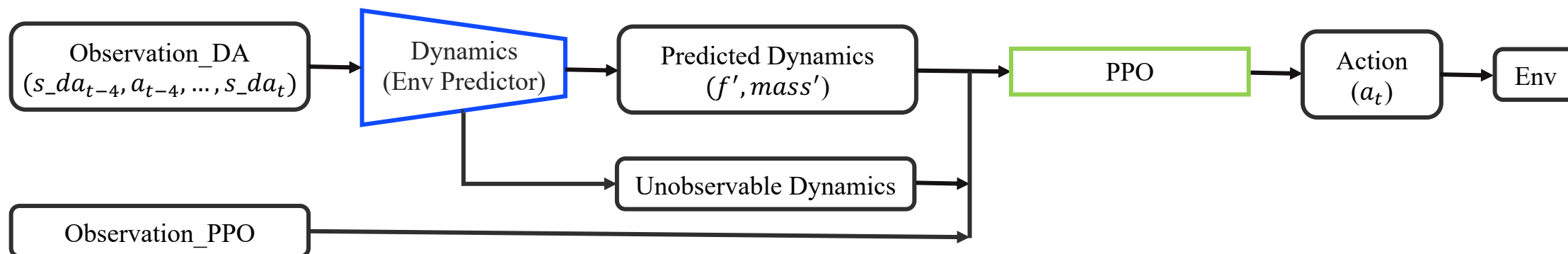
Luo, Jingru, and Kris Hauser. "Robust trajectory optimization under frictional contact with iterative learning." *Autonomous Robots* 41.6 (2017): 1447-1461.

Xie, Annie, James Harrison, and Chelsea Finn. "Deep reinforcement learning amidst lifelong non-stationarity." *arXiv preprint arXiv:2006.10701* (2020).

Training



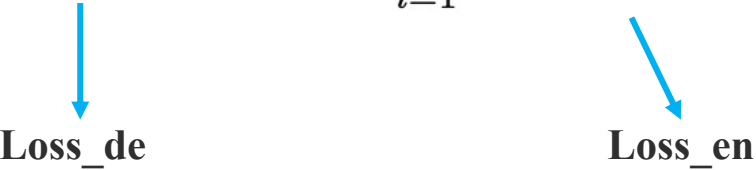
Collecting data or testing



τ^i : i'th episode's state, action trajectory; F_ϕ : encoder network (Env Predictor) with parameters ϕ ;

H_ψ : decoder network (State Predictor) with parameters ψ ; \mathbf{h} : Predicted Dynamics; \mathbf{w} : Unobservable Dynamics;

1. Encoder and decoder loss:

$$\min_{\phi_w, \phi_h, \psi} \lambda_1 \sum_{i=0}^N \sum_{t=0}^{T-1} \|H_\psi(\mathbf{s}_t^i, \mathbf{a}_t^i, F_\phi(\tau^i)) - \mathbf{s}_{t+1}^i\| + \lambda_2 \sum_{i=1}^N \|F_{\phi_h}^{\mathbf{h}}(\tau^i) - \mathbf{h}^i\|,$$


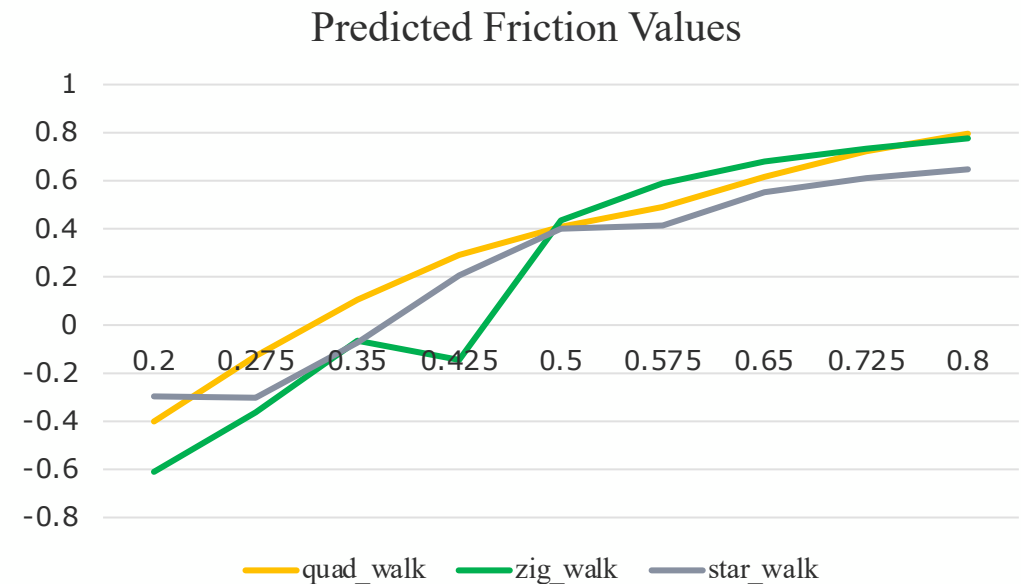
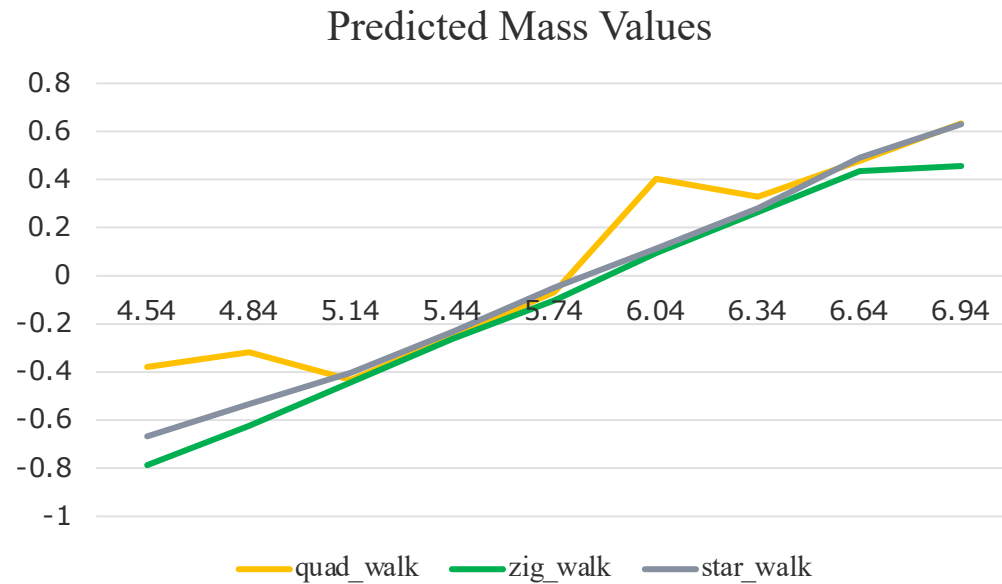
Loss_de Loss_en

2. Retraining with real-world data:

$$\mathcal{L}_{\text{Cluster}} = \sum_{i \neq j} \|F_\phi(\tilde{\tau}^i) - F_\phi(\tilde{\tau}^j)\| \quad \text{for } 1 \leq i, j \leq \tilde{N},$$

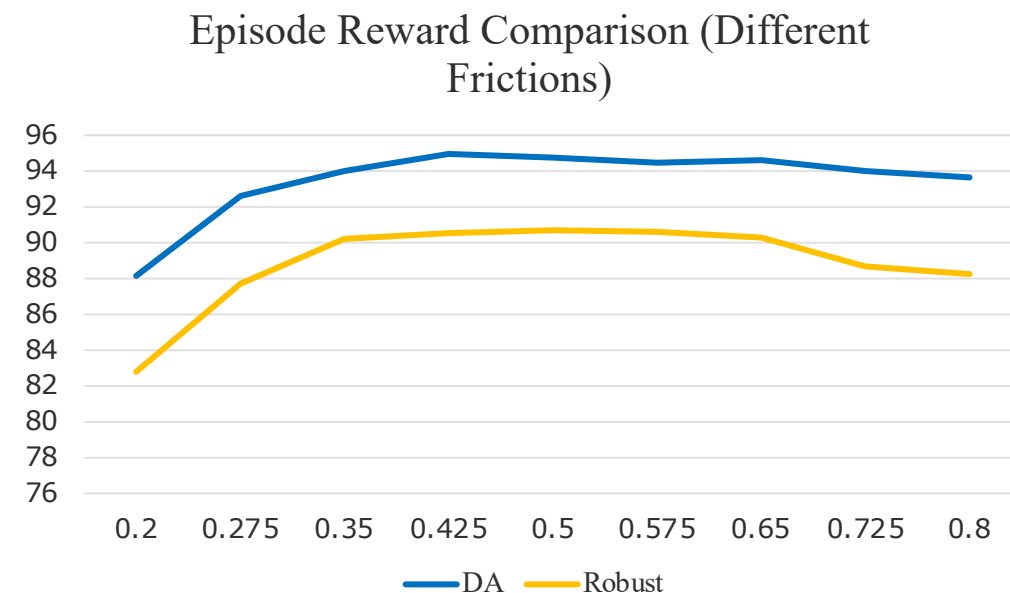
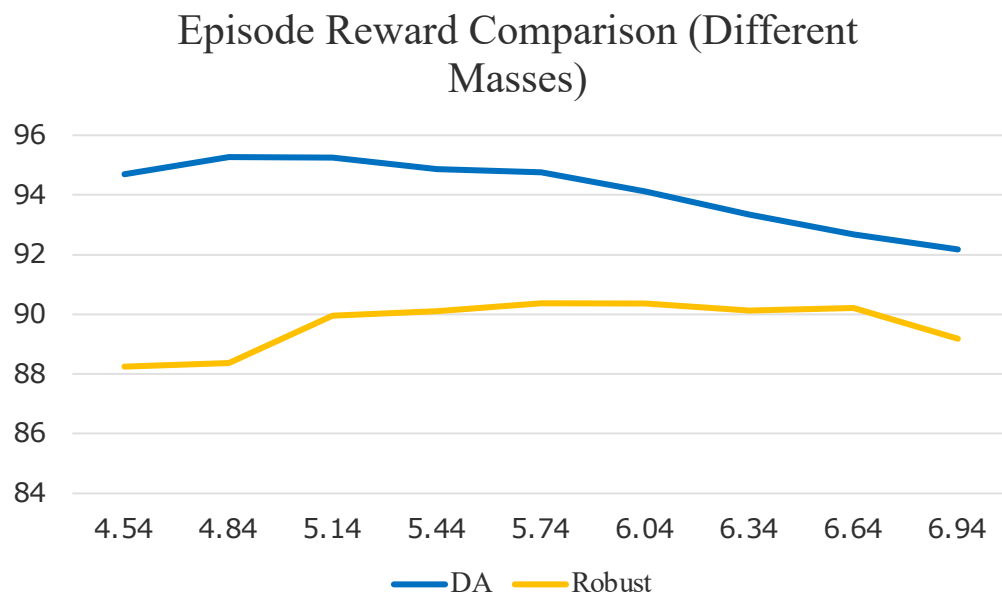
During training, the friction is randomized in $[0.5, 1.4]$ Ns/m; Body mass is randomized in $[0.8, 1.2] * \text{initial_body_mass}$ kg; Torque is damped in $[0, 2.]$ N.

Testing results of predicting masses and frictions in Gazebo are shown below:



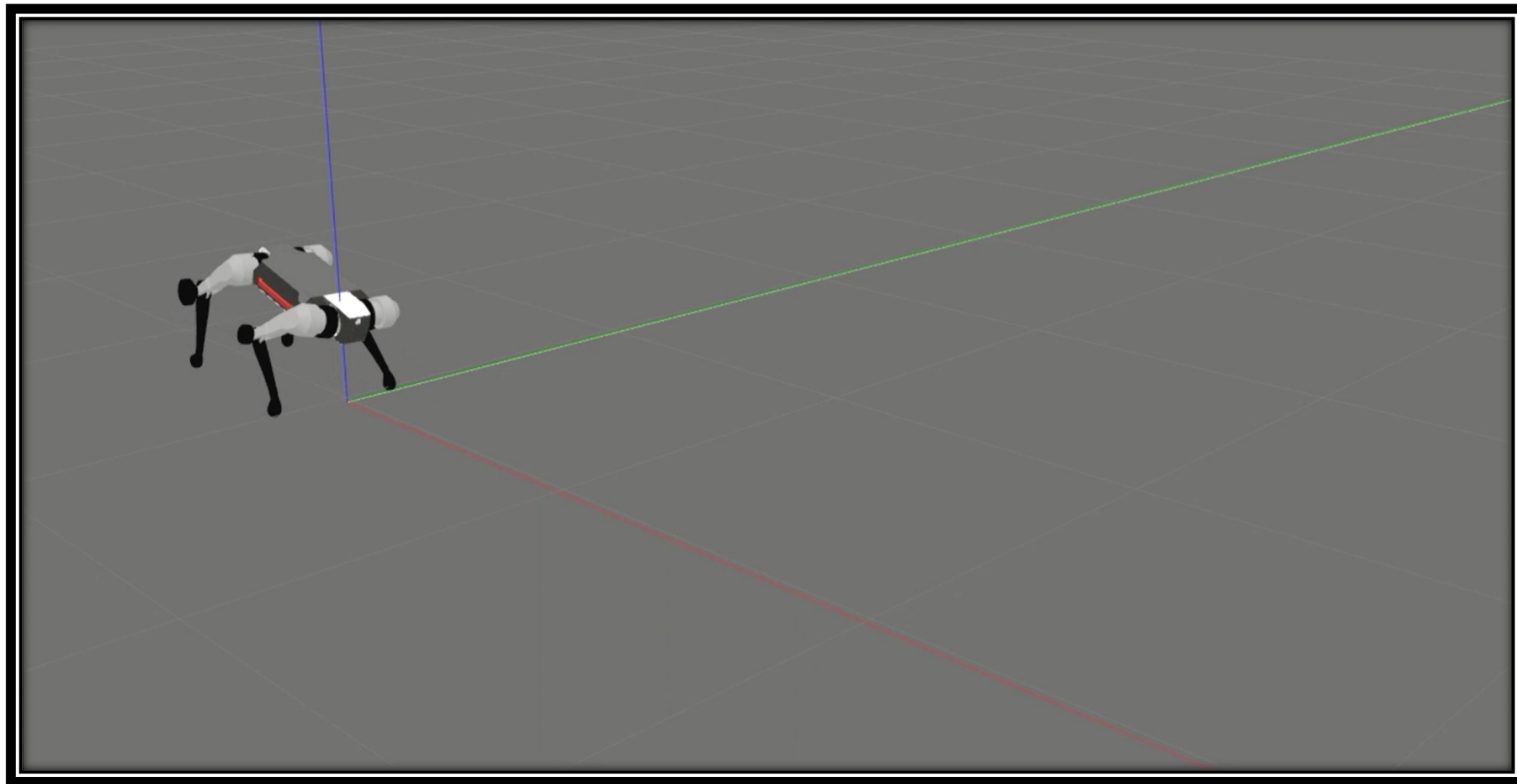
During training, the friction is randomized in $[0.5, 1.4]$ Ns/m; Body mass is randomized in $[0.8, 1.2] * \text{initial_body_mass}$ kg; Torque is damped in $[0, 2.]$ N.

Comparison of the episode reward between proposed Domain Adaptation Approach (DA) and Domain Randomization (Robust) are shown below:



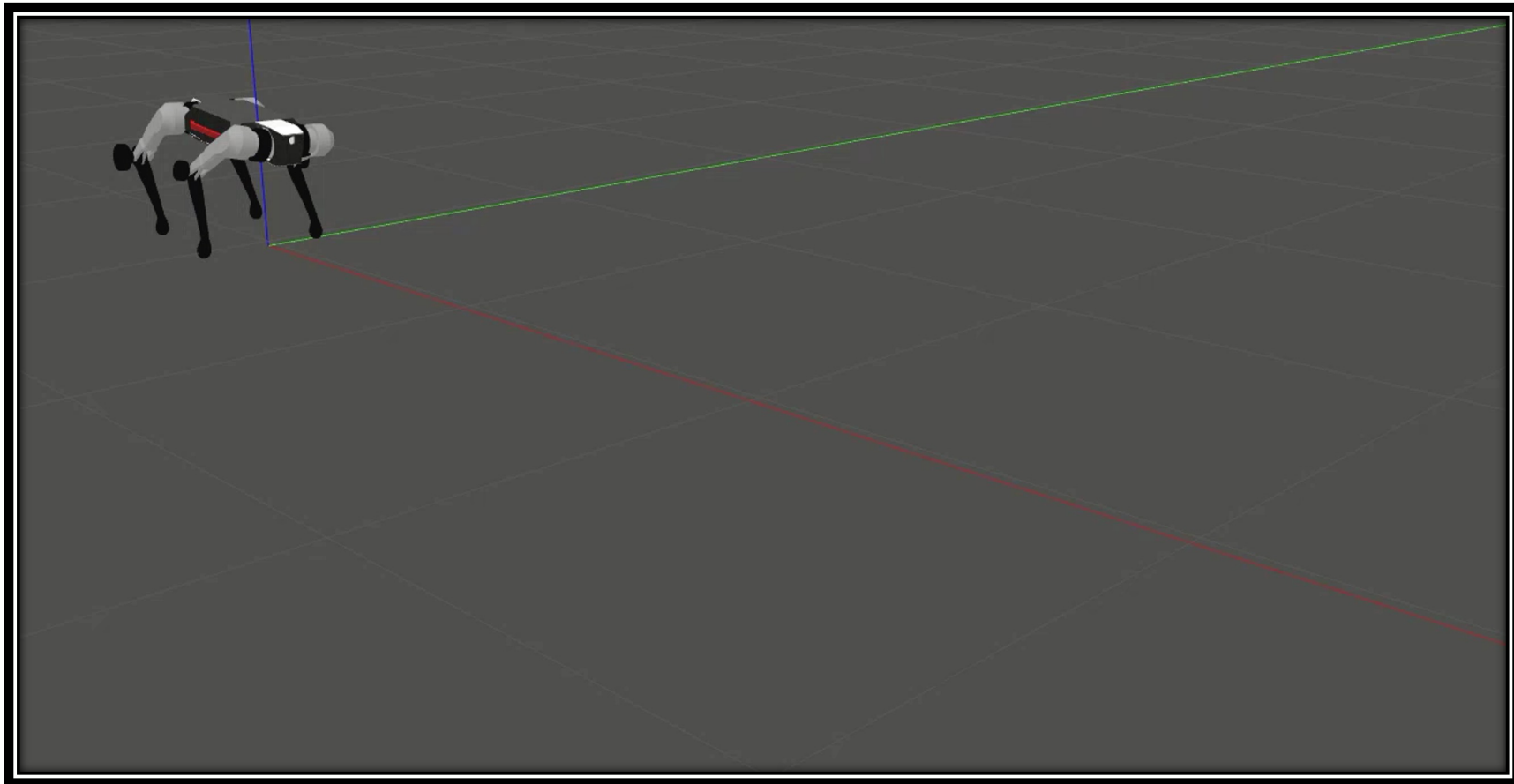
During training, the friction is randomized in $[0.5, 1.4]$ Ns/m; Body mass is randomized in $[0.8, 1.2] * \text{initial_body_mass}$ kg; Torque is damped in $[0, 2.]$ N.

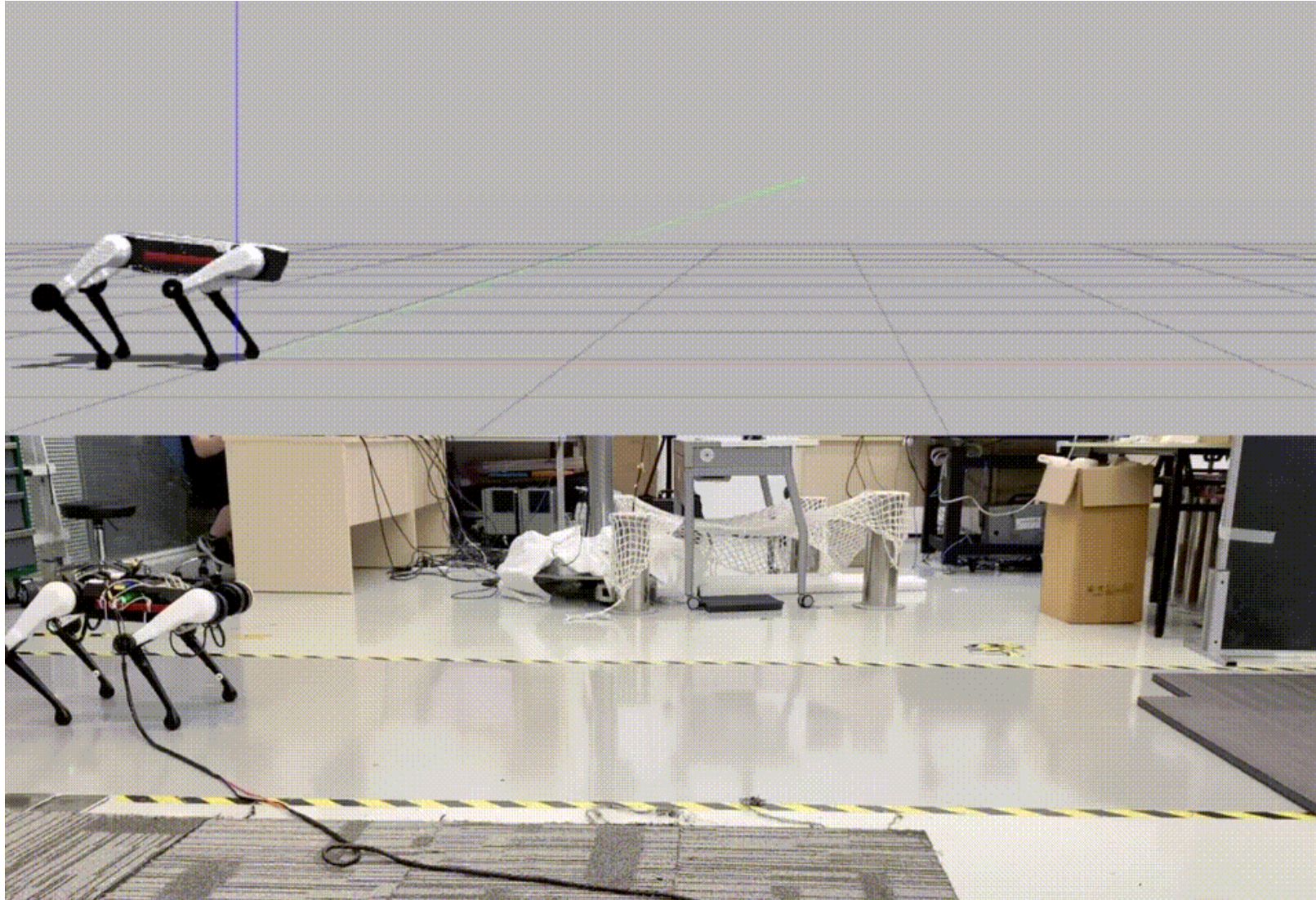
One test with a friction of 0.2 in Gazebo:



During training, the friction is randomized in $[0.5, 1.4]$ Ns/m; Body mass is randomized in $[0.8, 1.2] * \text{initial_body_mass}$ kg; Torque is damped in $[0, 2.]$ N.

One test with a friction of 0.8 in Gazebo:





- [1] Buşoniu, Lucian, et al. "Reinforcement learning for control: Performance, stability, and deep approximators." *Annual Reviews in Control* 46 (2018): 8-28.
- [2] Peng, Xue Bin, et al. "Learning agile robotic locomotion skills by imitating animals." *arXiv preprint arXiv:2004.00784* (2020).
- [3] Peng, Xue Bin, et al. "Sim-to-real transfer of robotic control with dynamics randomization." *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018.
- [4] Tan, Jie, et al. "Sim-to-real: Learning agile locomotion for quadruped robots." *arXiv preprint arXiv:1804.10332* (2018).
- [5] Xie, Annie, James Harrison, and Chelsea Finn. "Deep reinforcement learning amidst lifelong non-stationarity." *arXiv preprint arXiv:2006.10701* (2020).
- [6] Lee, Joonho, et al. "Learning quadrupedal locomotion over challenging terrain." *Science robotics* 5.47 (2020).
- [7] Kumar, Ashish, et al. "Rma: Rapid motor adaptation for legged robots." *arXiv preprint arXiv:2107.04034* (2021).
- [8] Luo, Jingru, and Kris Hauser. "Robust trajectory optimization under frictional contact with iterative learning." *Autonomous Robots* 41.6 (2017): 1447-1461.

感谢聆听

<https://git.woa.com/mikechzhou/motionkit-envs/tree/dev-model-v1-domain-adaptation>

pengzhiyang@tencent.com/tyypz2590477658@gmail.com