VBC

Visual Whole-Body Control for Legged Loco-Manipulation

Course Project Presentation
AI60209: AI for Robot Autonomy
IIT Kharagpur

Oct 29, 2025

Presented By:

Team B

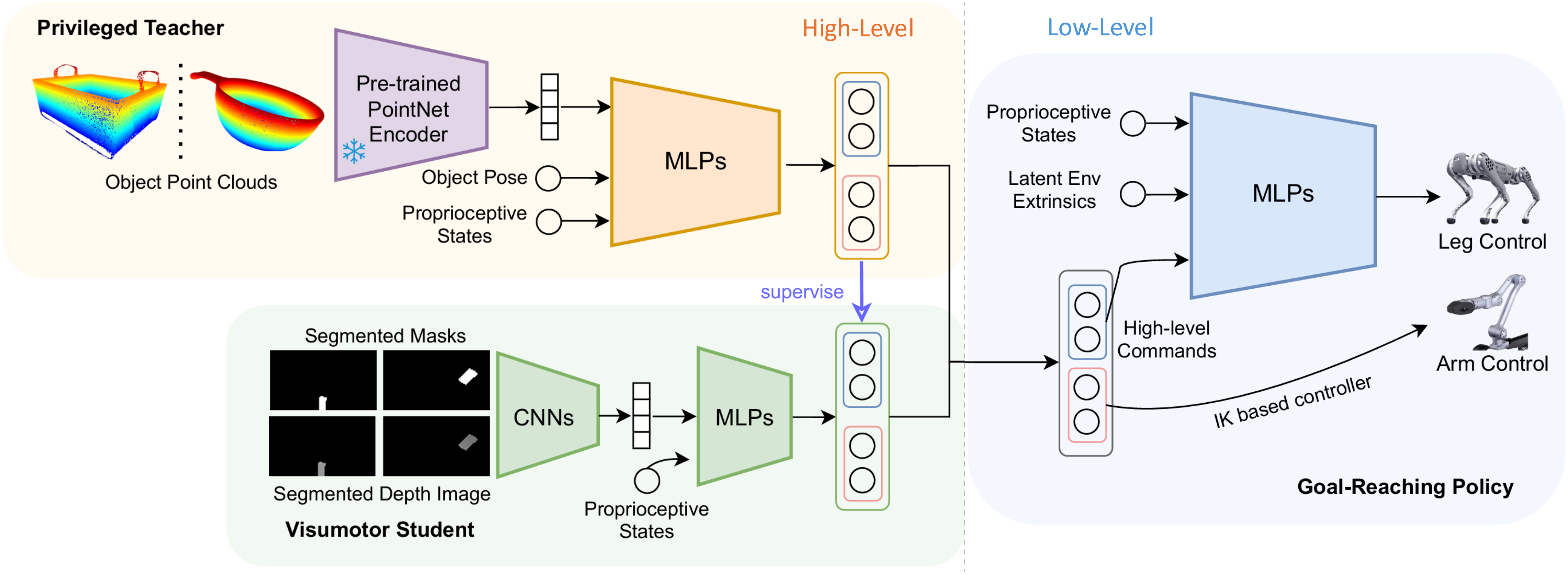
Ranjan Sarkar	24AI92R01
Ravindra Mina	25AI60R02
Anviksha Dharmale	21MT30005
Kilari Shiva Prasad	21ME31072
Debraj Das	21ME3AI31

Introduction: Quadruped Loco-Manipulation

* Taken from the original project owner's YouTube link.

Content

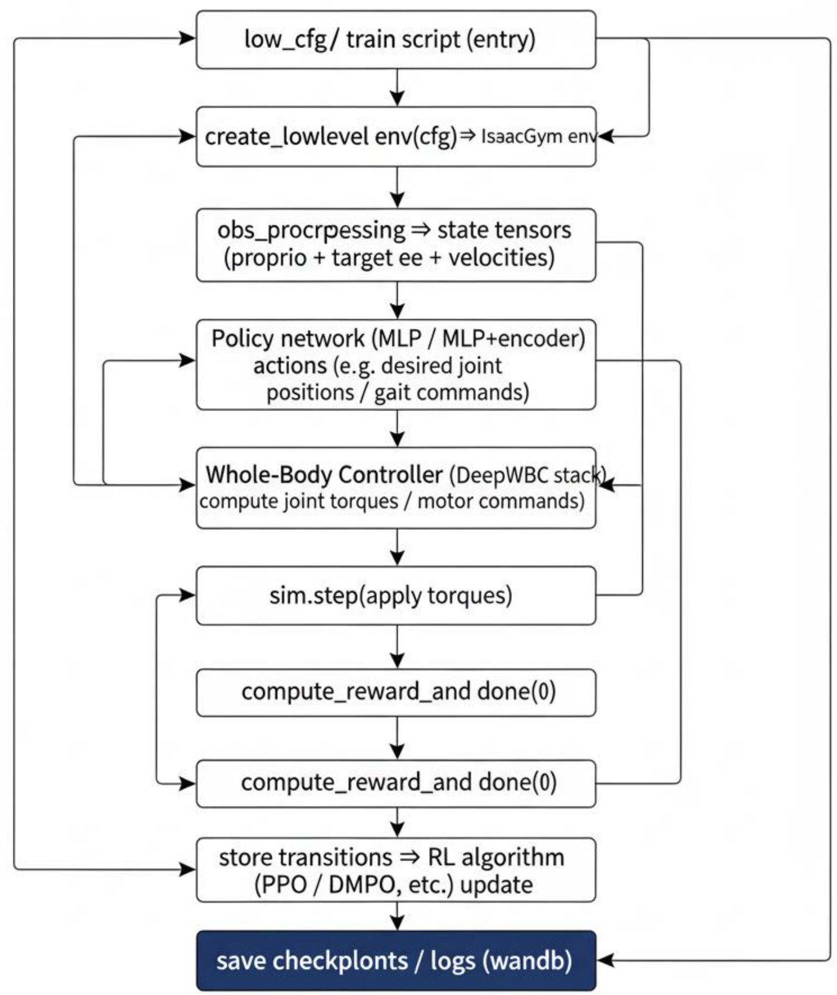
VBC Framework



* Taken from the original paper [1]

Low Level Policy (General controller for quadruped + arm)

Low-Lowel Policy Training Workflow


low_cfg / train script (entry)
create_lowlevel env(cfg) ⇒ IsaacGym env
obs_procrpessing ⇒ state tensors (proprio + target ee + velocities)
Policy network (MLP / MLP+encoder) actions (e.g. desired joint positions / gait commands)
Whole-Body Controller (DeepWBC stack) compute joint torques / motor commands)
sim.step(apply torques)
compute_reward_and done(0)
compute_reward_and done(0)
store transitions ⇒ RL algorithm (PPO / DMPO, etc.) update
save checkplonts / logs (wandb)

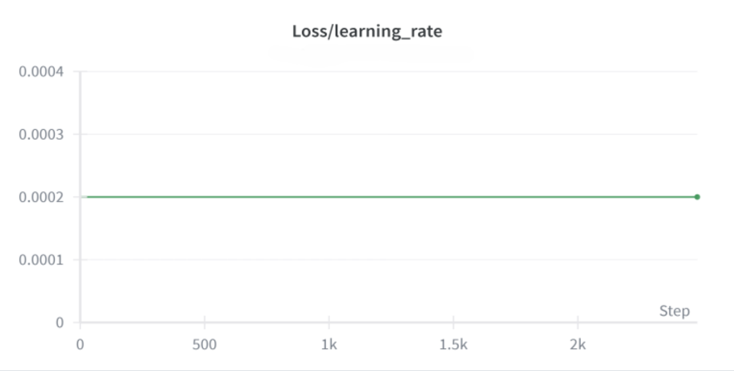
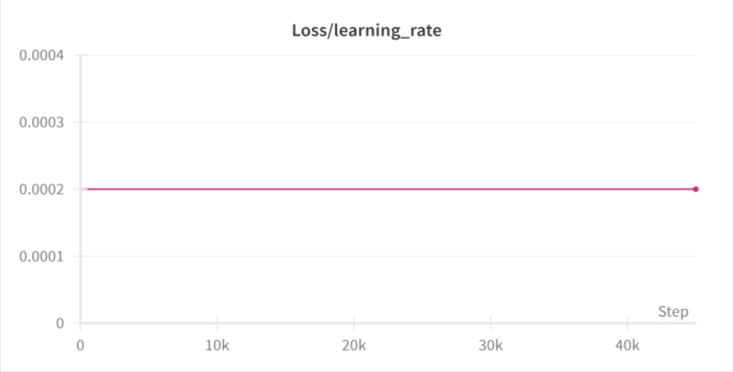
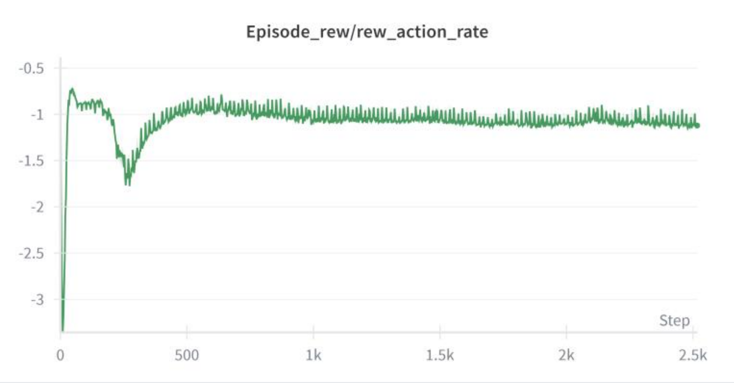
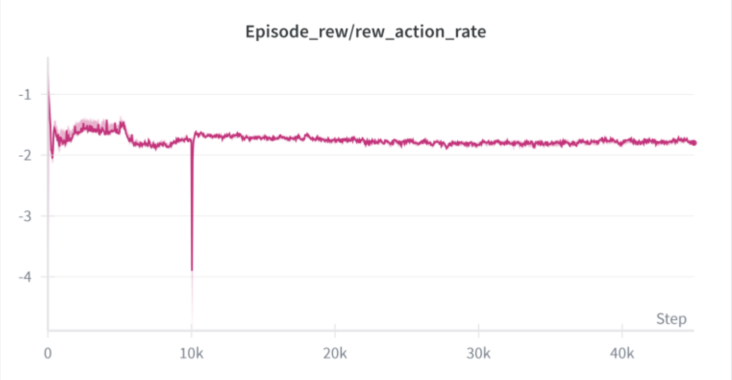
Aspect	Description
Purpose	Learns motor-level control (joint torques, locomotion, end-effector tracking) for quadruped with manipulator arm.
Entry Point	**low-level/train.py , low-level/play.py**
Functions	load_cfg_lowlevel(), create_lowlevel_env(), obs_preprocessing(), PolicyNetwork, WholeBodyController, PPOTrainer
Algorithm	**Proximal Policy Optimization (PPO)**
Input	**Proprioceptive data: joint angles, velocities, base pose, target end-effector position/velocity.**
Output	Low-level skill policy checkpoint (.pt) used by the high-level controller.
Key Parameters	**--timesteps**, --rl_device, --sim_device, --arm_kp, --arm_kd, **--controlFrequencyLow**, --**table_height**, **--wandb**
Controller Used	Deep Whole-Body Controller (DeepWBC) — computes joint torques from desired EE pose.
Observation Space	Local proprioception + task-space targets from high-level commands.
Reward function	Encourages stable walking, accurate end-effector tracking, low energy, and smooth motion.
Training Loop	**Collect rollouts → compute reward → PPO update → save checkpoint**
Usage	Serves as motor primitive module for high level tasks like grabbing, picking the objects

Demonstration (Low-Level)

Low-level-1 video • regenerated by our team.

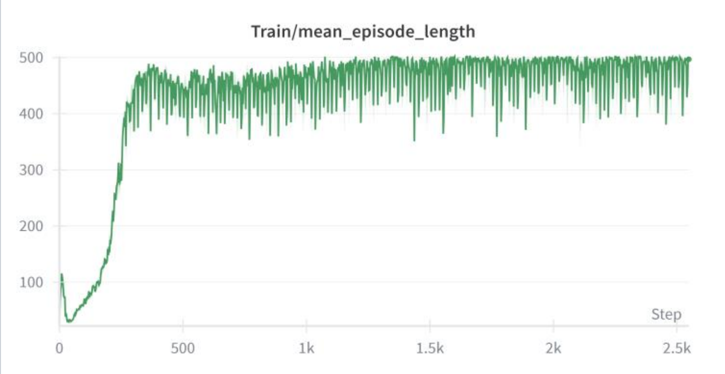
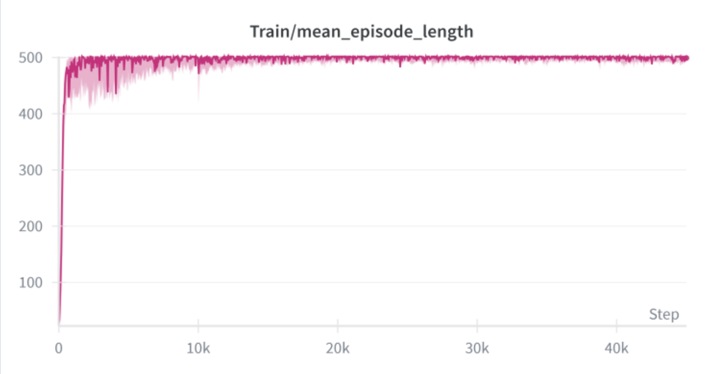
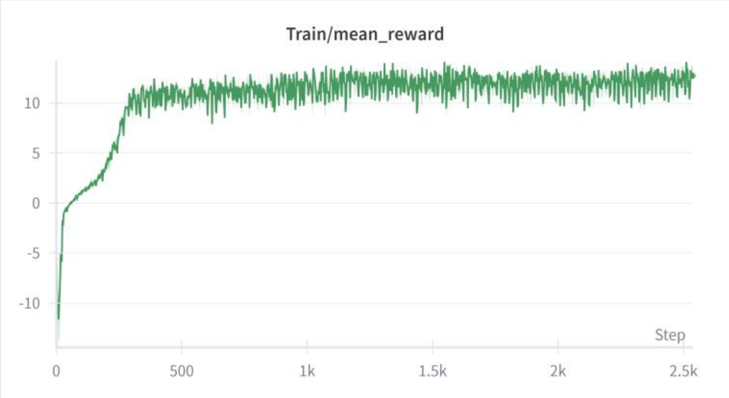
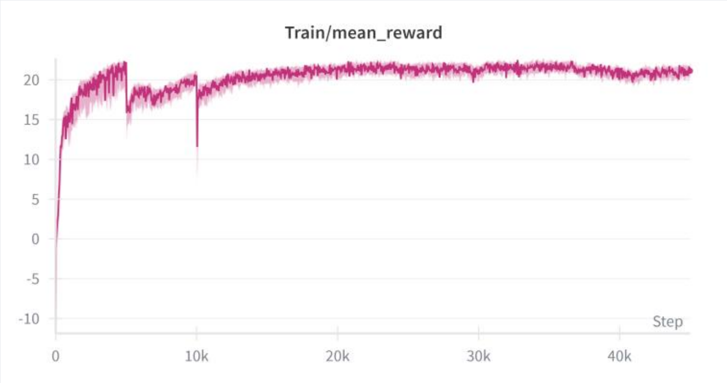
Evaluation Metrics : Low Level Policy

The model's **logs** were saved to WandB.

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Loss (Learning Rate)		
Episode Reward (Action Rate)		

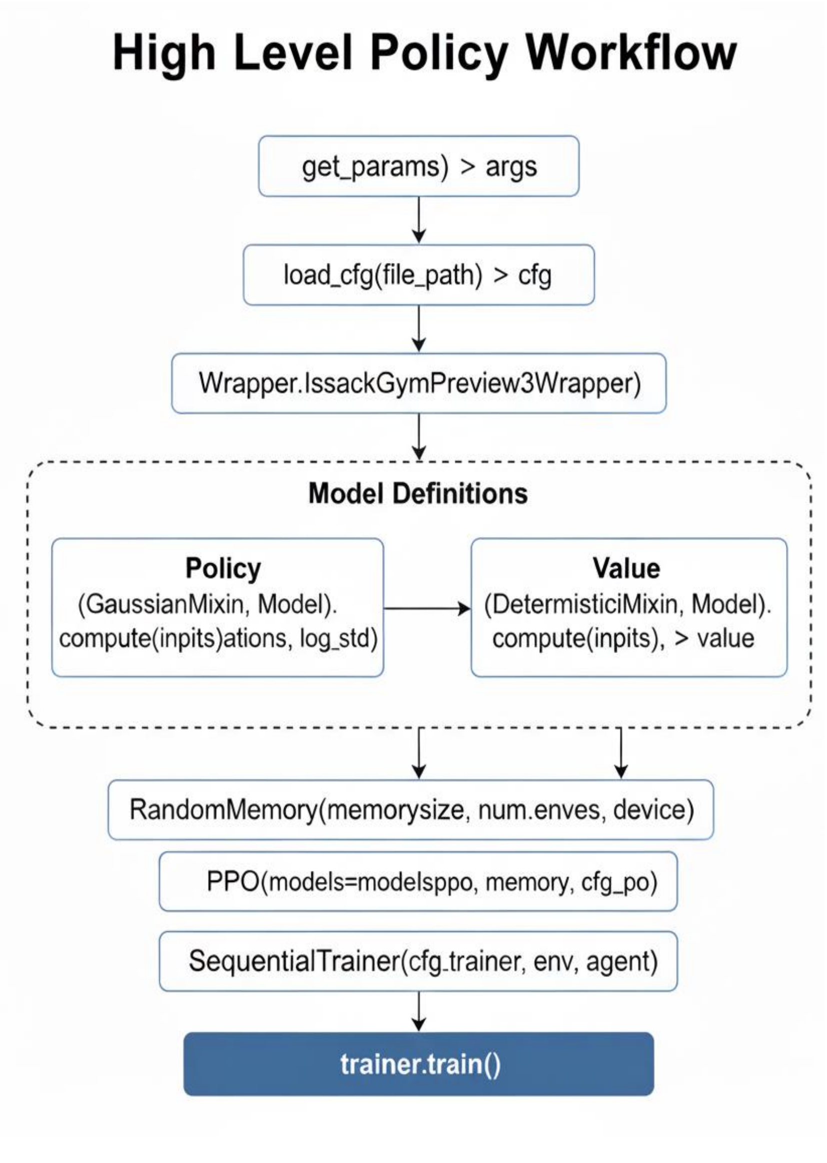
* x-axis : Number of Steps (No. of iterations = 45000) & y-axis : mentioned metric

Evaluation Metrics : Low Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Mean Episode length		
Mean Reward		

* x-axis : Number of Steps (No. of iterations = 45000) & y-axis : mentioned metric

High Level Policy (Teacher PPO & Student BC)



High Level Policy Workflow

get_params) > args

load_cfg(file_path) > cfg

Wrapper.IssackGymPreview3Wrapper)

Model Definitions

Policy
(GaussianMixin, Model).
compute(inpits)ations, log_std)

Value
(DetermisticiMixin, Model).
compute(inpits), > value

RandomMemory(memorysize, num.enves, device)

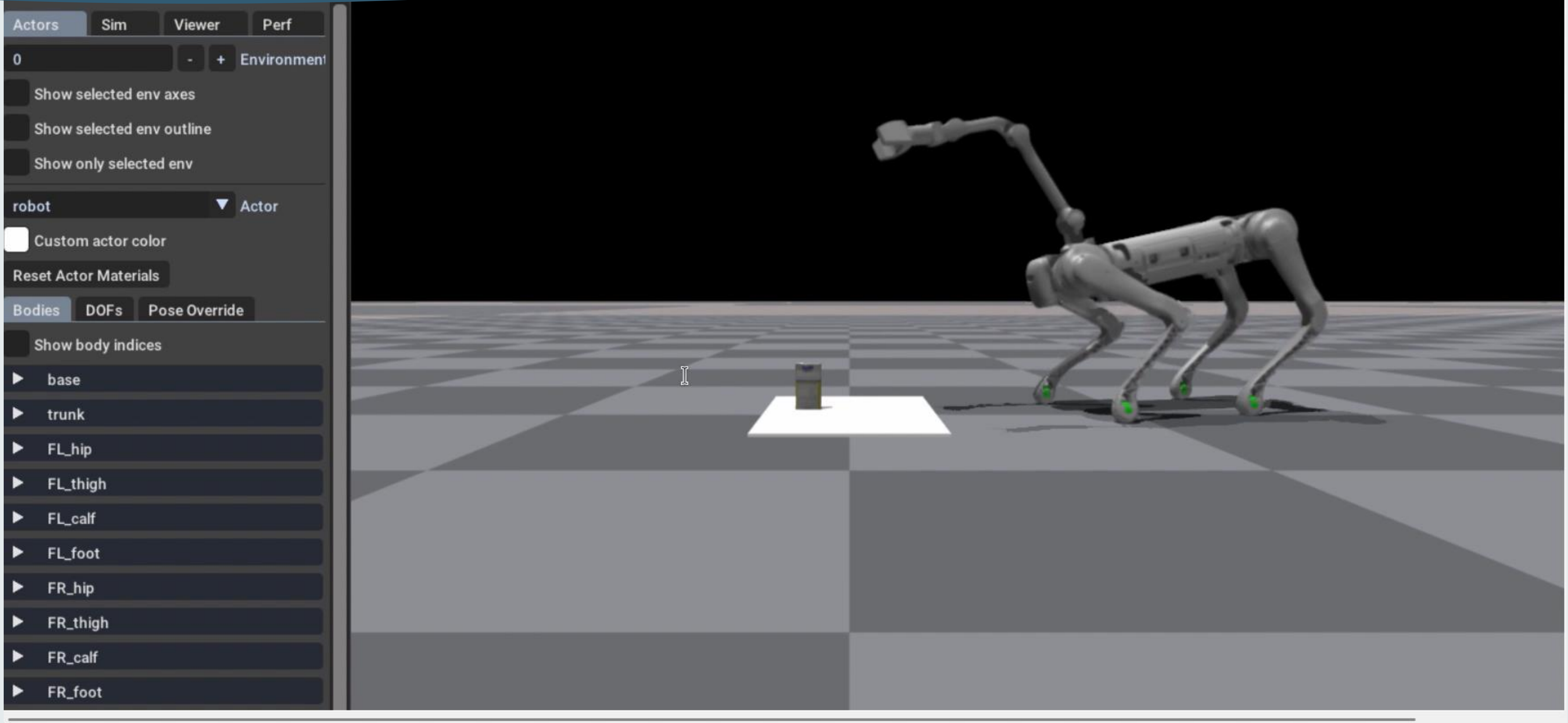
PPO(models=modelsppo, memory, cfg_po)

SequentialTrainer(cfg.trainer, env, agent)

trainer.train()

Aspect	Description
Purpose	Learns task-level control (object manipulation, whole-body coordination) using pre-trained low-level skills.
Entry Point	high-level/train_multistate.py
Functions	get_params(), load_cfg(), create_env(), Wrapper.IsaacGymPreview3Wrapper, Policy, Value, PPO, SequentialTrainer
Algorithm	Proximal Policy Optimization (**PPO**)
Input	Configuration file (.yaml), CLI arguments (e.g., --task, --timesteps, --rl_device, etc.)
Output	Abstract actions: target end-effector position, velocity, or skill command to low-level policy
Key Parameters	rollouts, learning_rate, discount_factor, ratio_clip, value_loss_scale, num_envs, controlFrequencyLow, wandb
Environment Wrapper	IsaacGymPreview3Wrapper — standardizes env interaction with PPO agent.
Policy Network	GaussianMixin, Model — outputs mean and log_std for stochastic actions.
Value Network	GaussianMixin, Model — outputs mean and log_std for stochastic actions.
Training Loop	SequentialTrainer(cfg_trainer, env, agent) → trainer.train()
Usage	Trained after the low-level controller; can use a vision-based **student policy for imitation (BC)**.

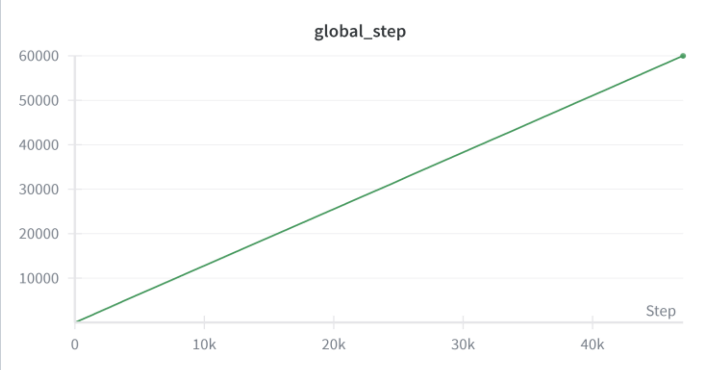
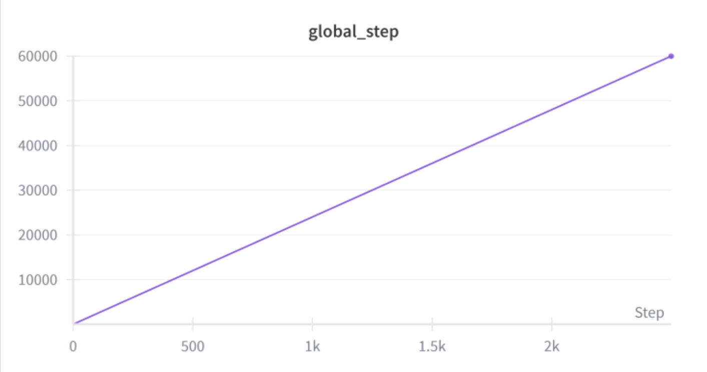
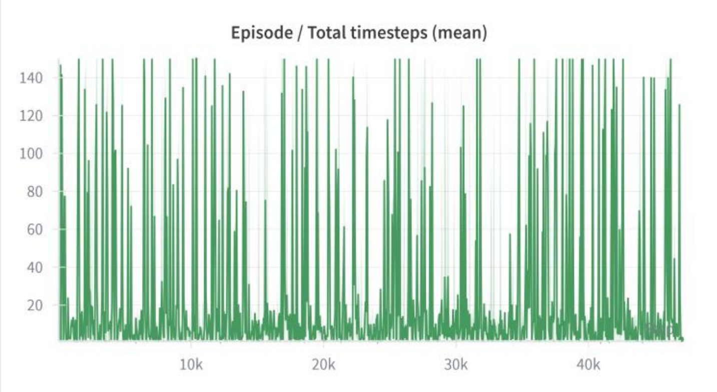
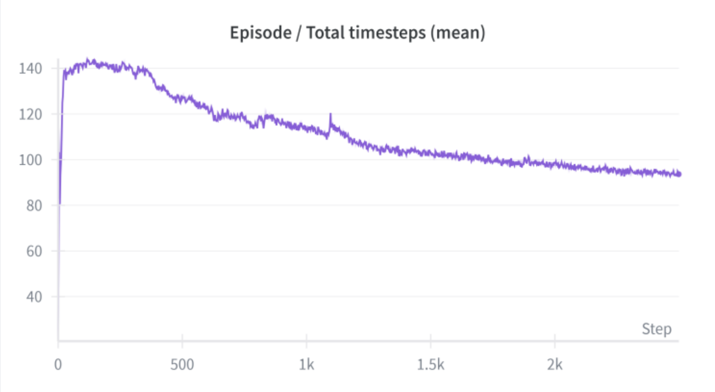
Demonstration (High-Level)



High-level-1 video • regenerated by our team.

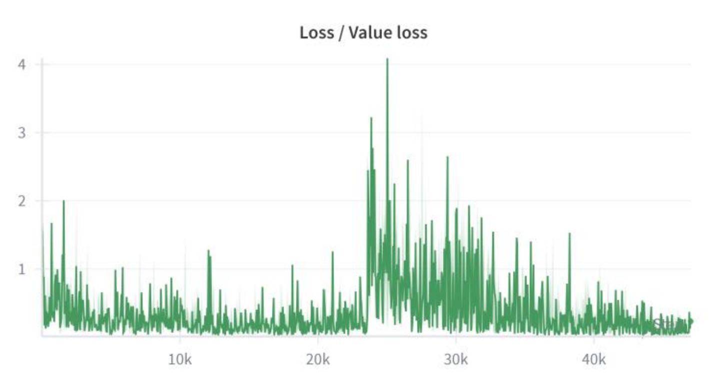
Evaluation Metrics: High Level Policy

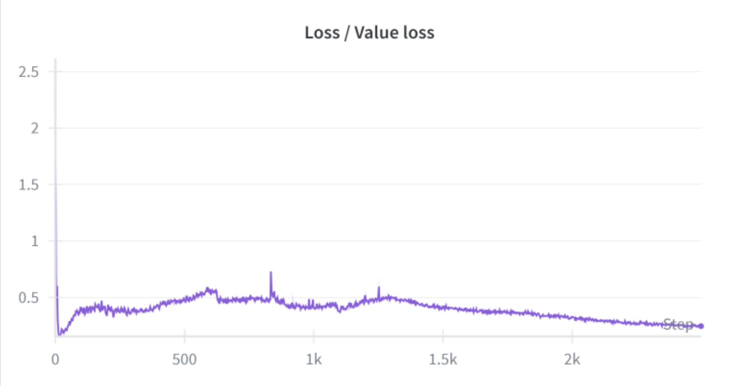
The model's **logs** were saved to **WandB**.

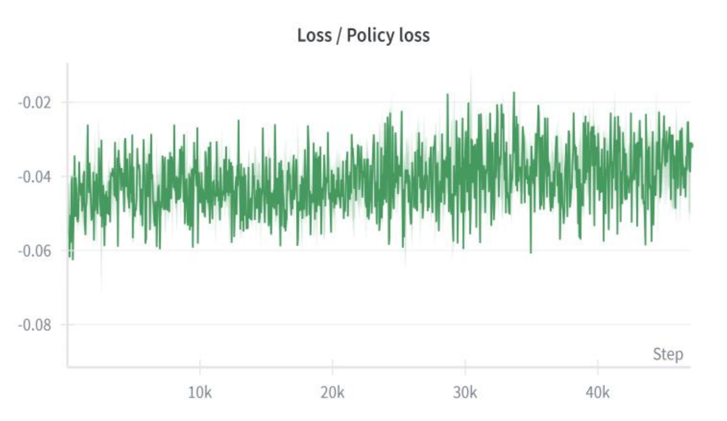
Metric	Actual (As from implemented code)	Expected (As given in git repository)
No. of iterations		
Episode/Total time steps (mean)		

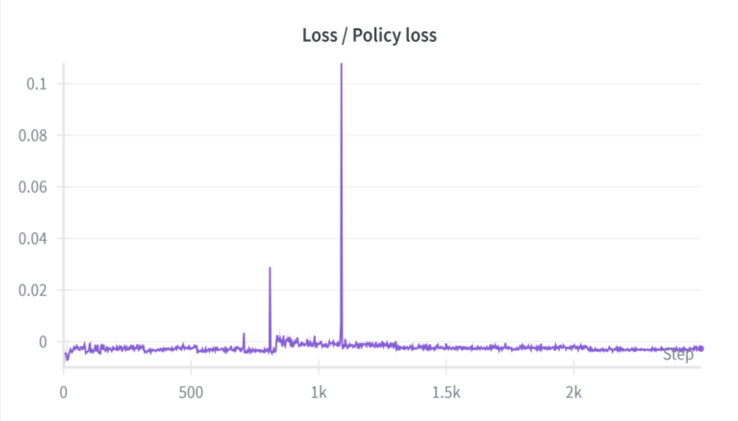
* x-axis : Number of Steps & y-axis : mentioned metric

Evaluation Metrics: High Level Policy

<table>
<tr><th>Metric</th><th>Actual
(As from implemented code)</th><th>Expected
(As given in git repository)</th></tr>
<tr><td>Loss
(Value loss)</td><td>

Loss / Value loss

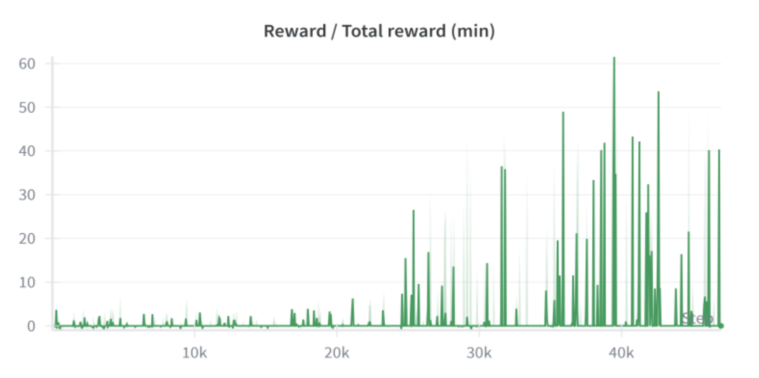
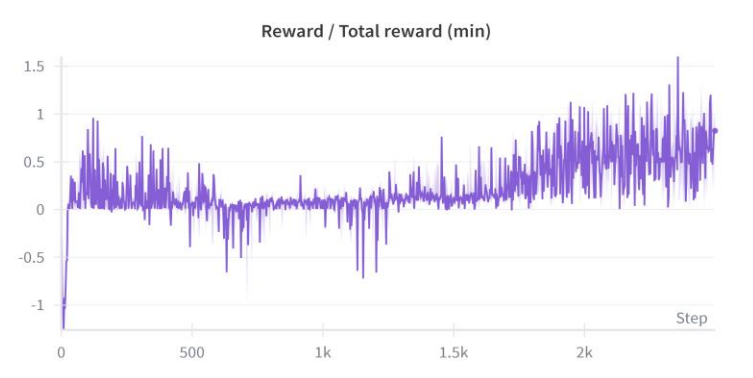
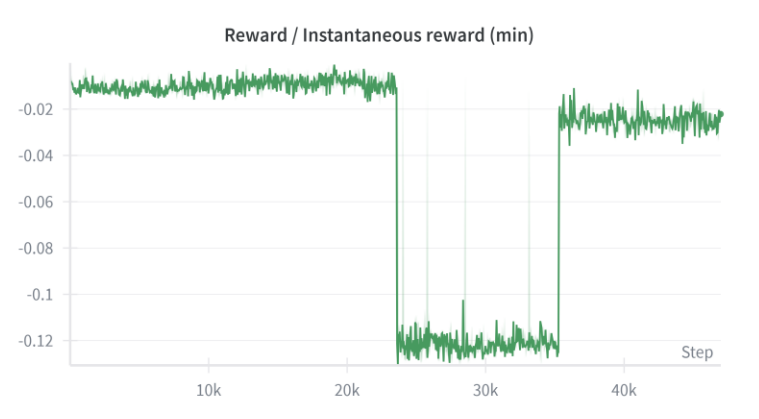
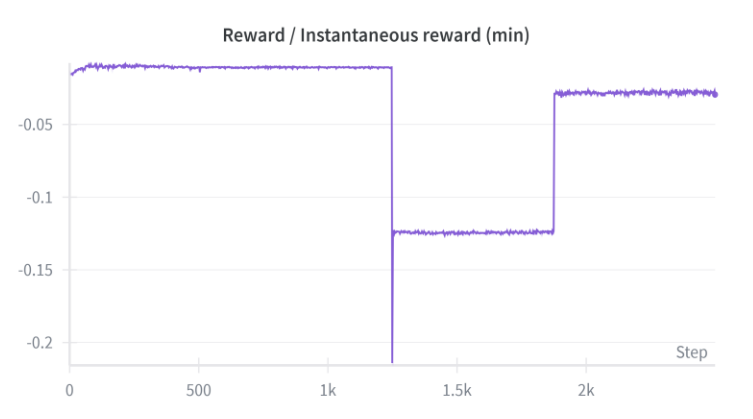
</td><td>

Loss / Value loss

</td></tr>
<tr><td>Loss
(Policy loss)</td><td>

Loss / Policy loss

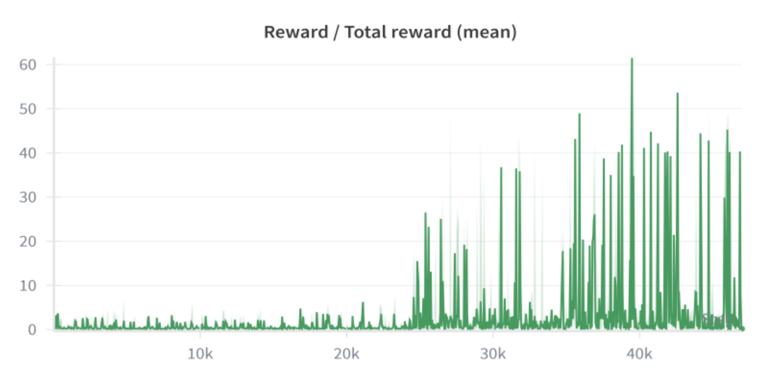
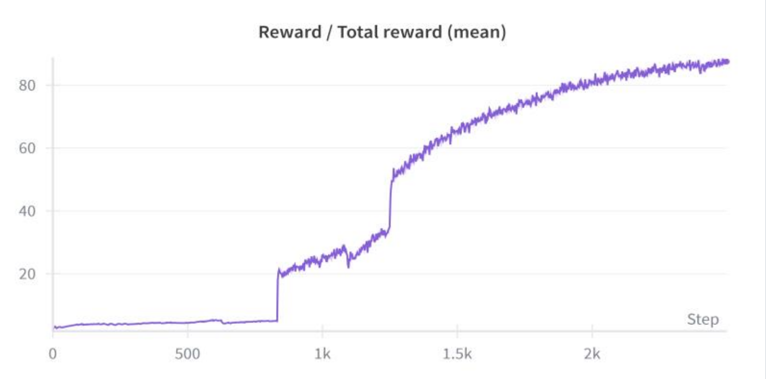
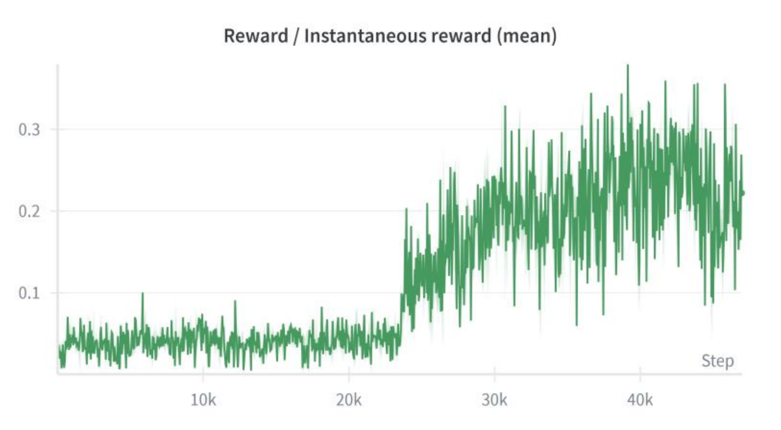
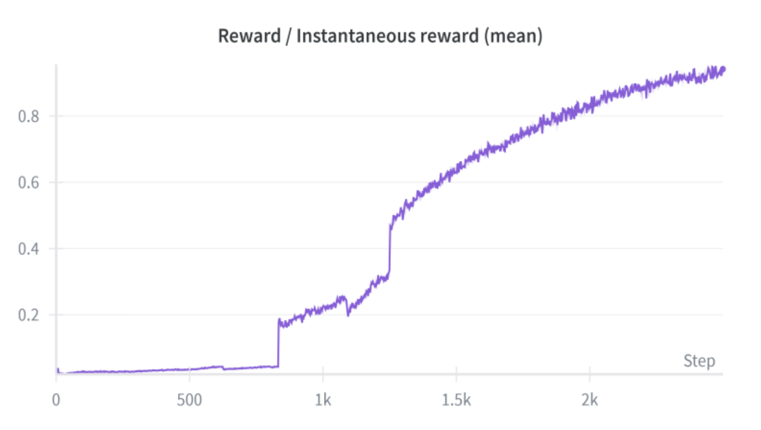
</td><td>

Loss / Policy loss

</td></tr>
</table>

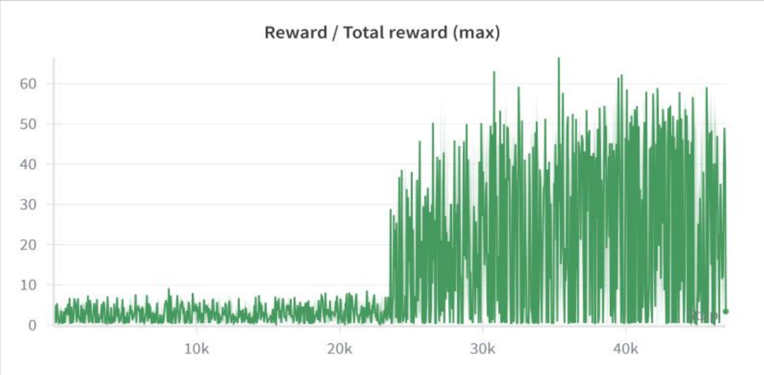
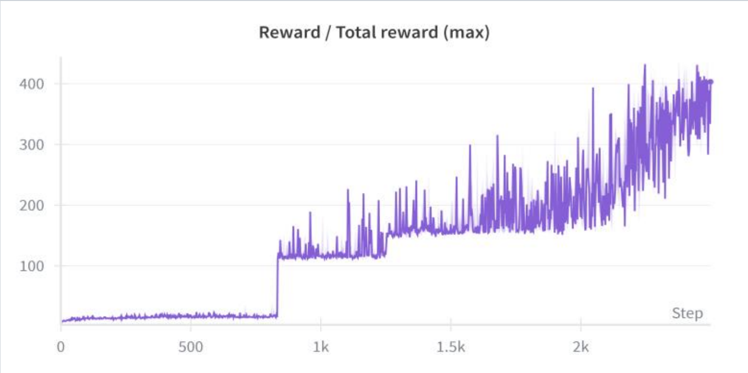
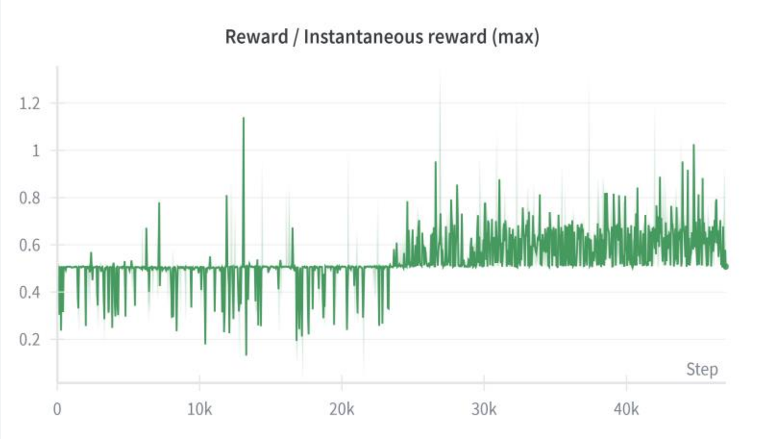
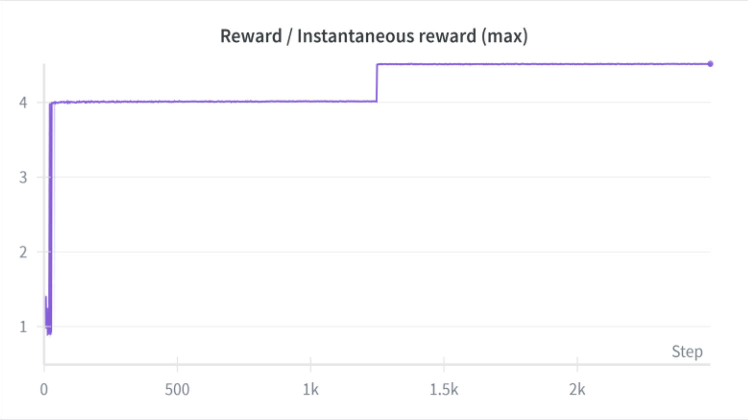
Evaluation Metrics: High Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Total Reward (min)		
Instantaneous Reward (min)		

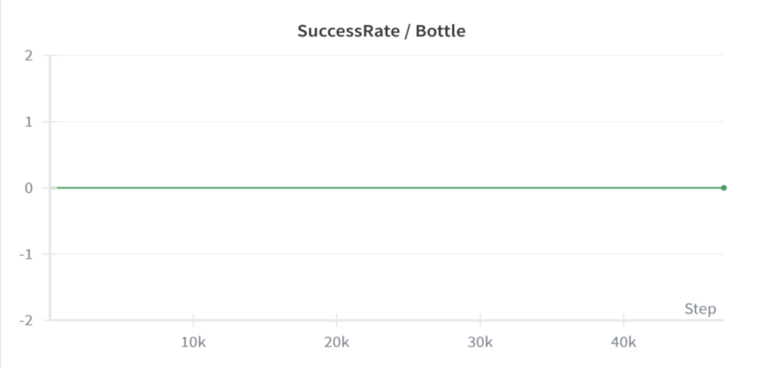
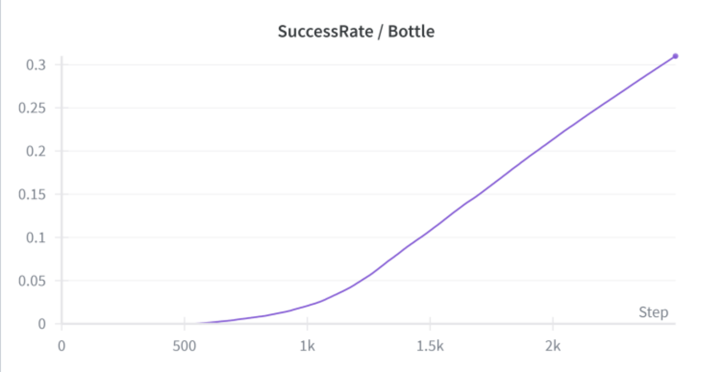
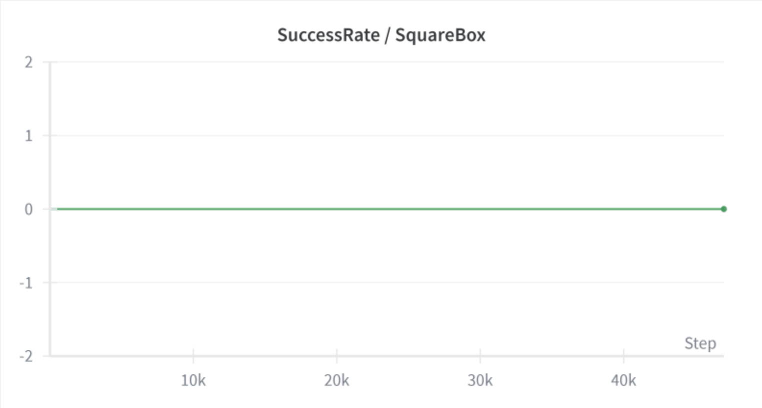
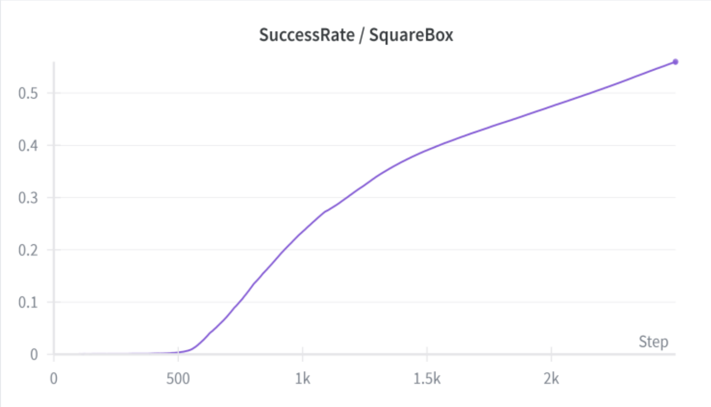
Evaluation Metrics: High Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Total Reward (mean)	 Reward / Total reward (mean)	 Reward / Total reward (mean)
Instantaneous Reward (mean)	 Reward / Instantaneous reward (mean)	 Reward / Instantaneous reward (mean)

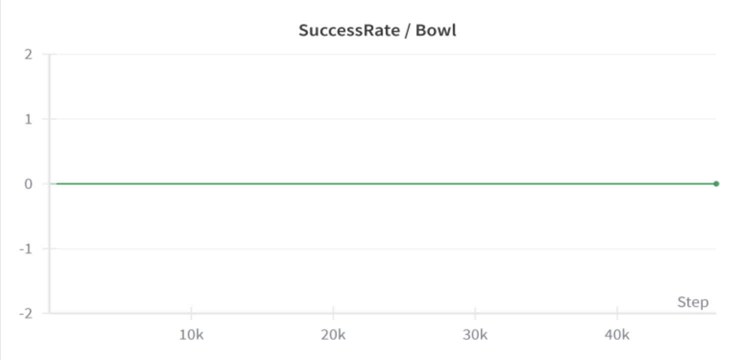
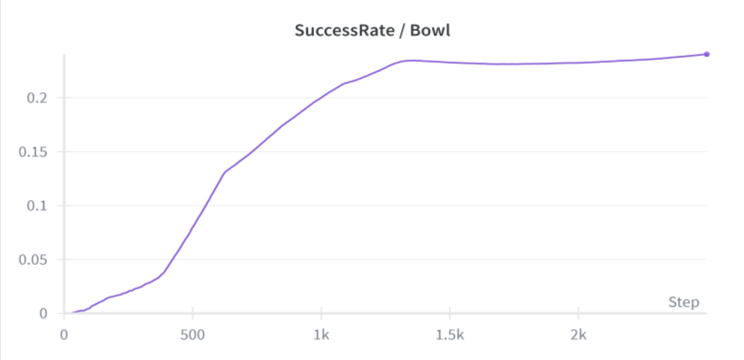
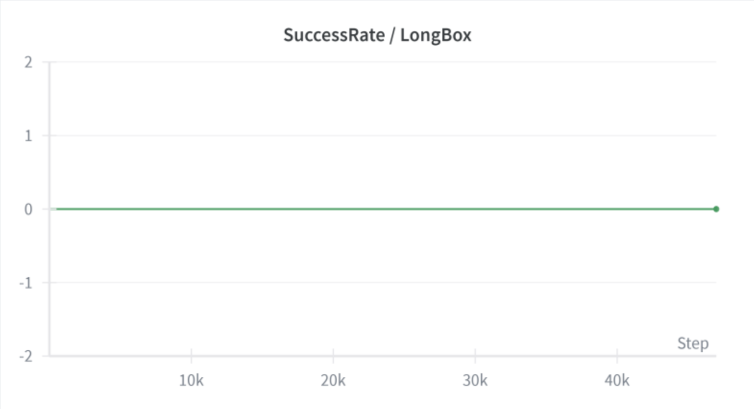
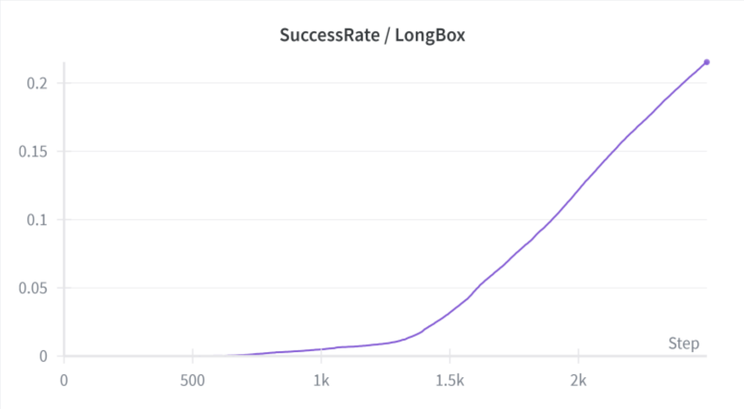
Evaluation Metrics: High Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Total Reward (max)		
Instantaneous Reward (max)		

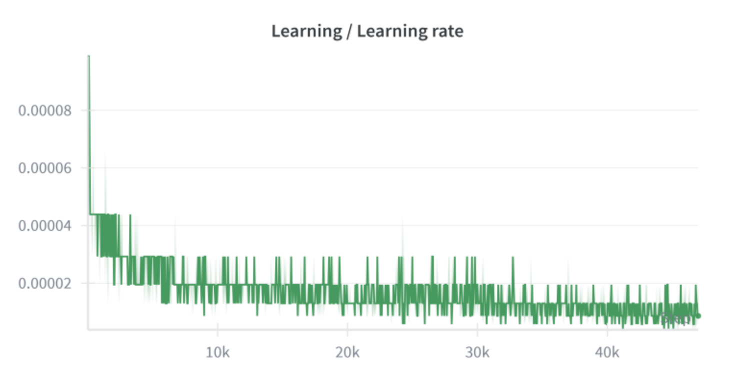
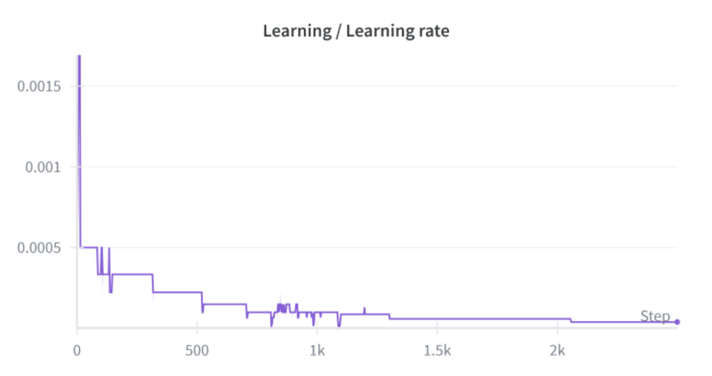
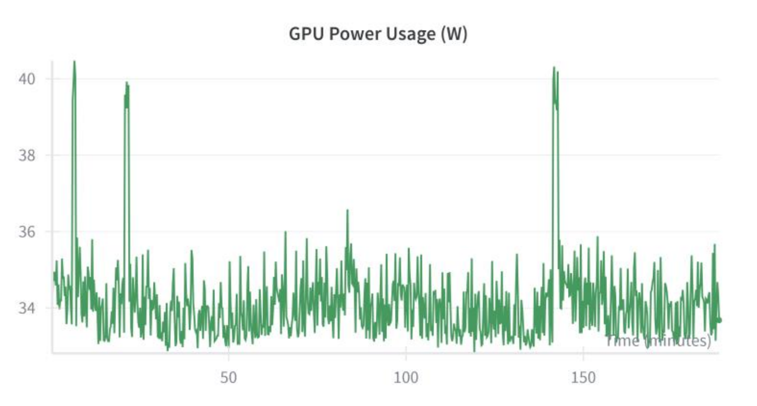
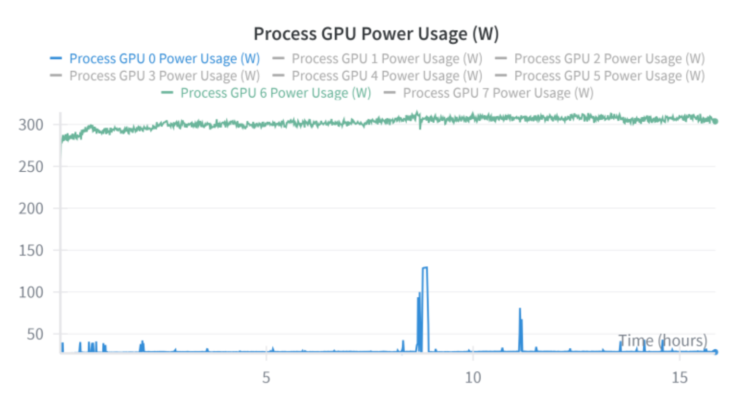
Evaluation Metrics: High Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Success Rate (Bottle)	 	
Success Rate (Square Box)	 	

Evaluation Metrics: High Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Success Rate (Bowl)		
Success Rate (Long Box)		

Evaluation Metrics: High Level Policy

Metric	Actual (As from implemented code)	Expected (As given in git repository)
Learning Rate		
System (GPU Usage in W)		

Reference

1. M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, and X. Wang. Visual Whole-Body Loco-Manipulation. *GitHub*.
   URL: https://github.com/Ericonaldo/visual_wholebody

2. M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, and X. Wang. Visual Whole-Body Control for Legged Loco-Manipulation. *Conference on Robot Learning (CoRL)*, 2024.
   URL: https://arxiv.org/pdf/2403.16967

3. M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, and X. Wang. VBC simulation metrics. Wandb output results:
   https://wandb.ai/ericonaldo/b1-pick-multi-teacher?nw=nwuserericonaldo

Thank You