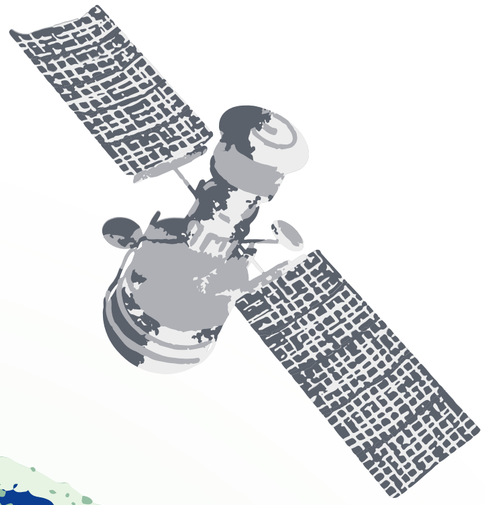
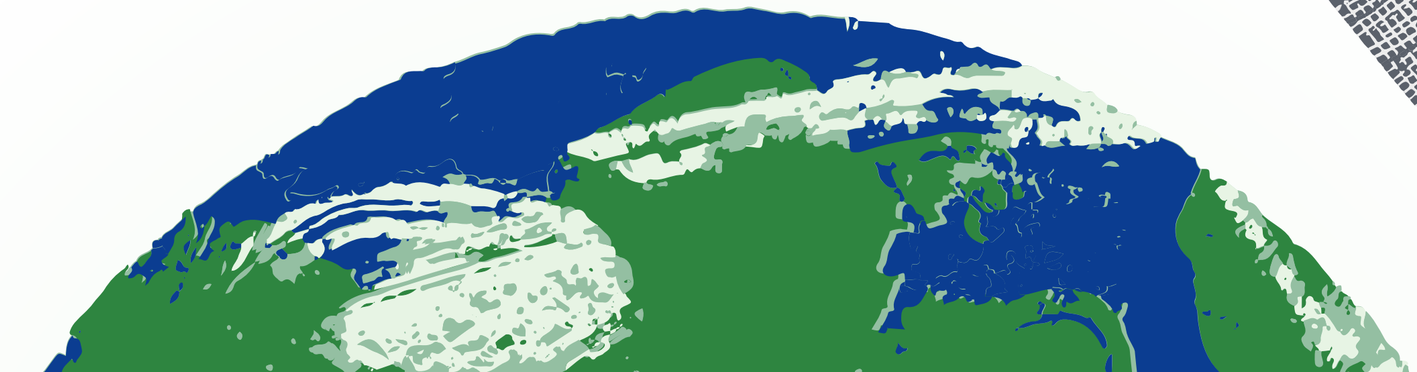
SATELLITE CONTROL USING PARALLEL DEEP REINFORCEMENT LEARNING



NVIDIA IsaacGym - SKRL Integration for smooth control



MOTIVATION

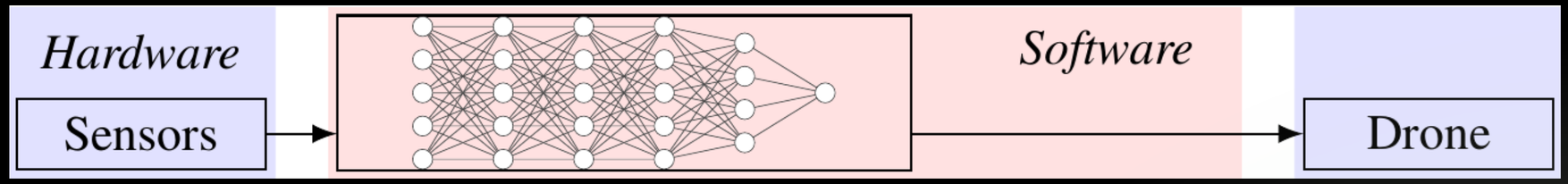
Motivi chiave per cui scegliere un framework **parallelizzato** e basato su **Deep Reinforcement Learning**

- **Autonomia** onboard per gestire la latenza di comunicazione
- **Inferenza** a **bassa latenza** su **hardware vincolato** in performance e consumi
- **Zero-Shot Transfer: Adattamento** a dinamiche **variabili/sconosciute** o **guasti**
- **Allenamento ottimizzato** in simulazioni **parallele**

AUTONOMOUS CONTROL

Model-Based Approach

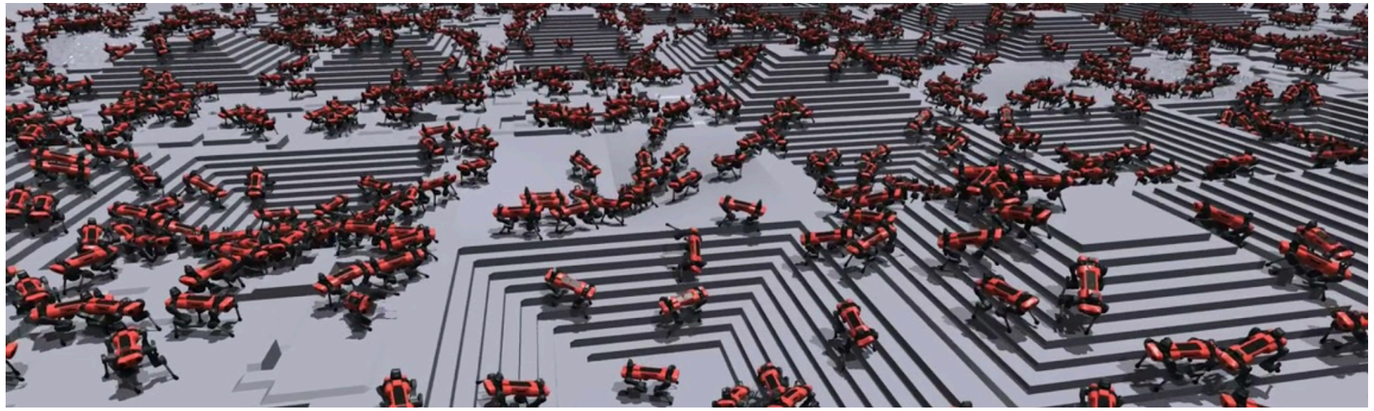
Learning-Based Approach



PARALLELIZED ENVIRONMENT

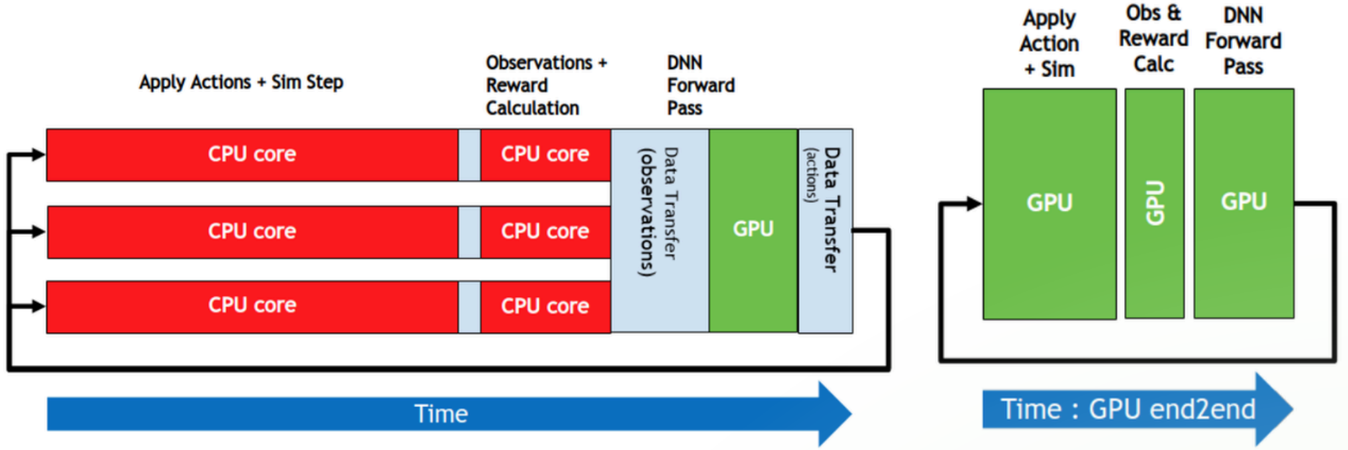
IsaacGym

- **Legacy open-source** tool
- Simulazione fisica e addestramento su **GPU ottimizzato su NVIDIA**



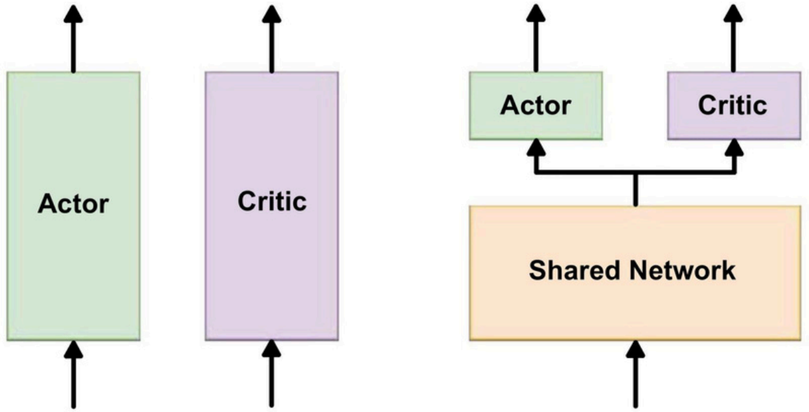
SKRL

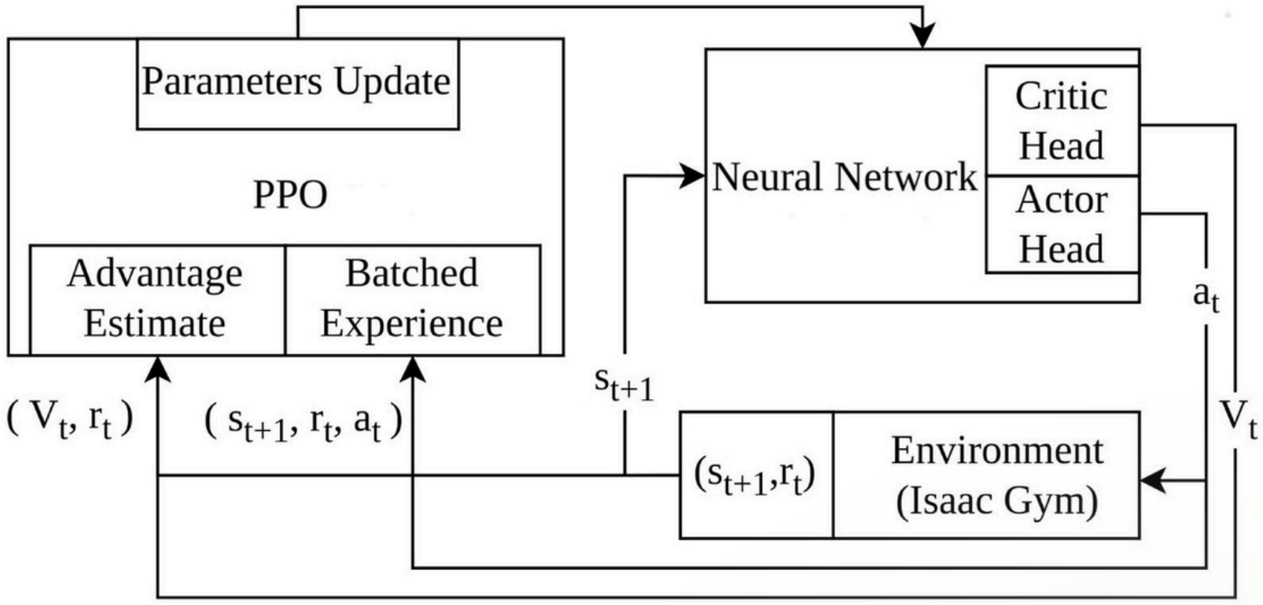
- Libreria **open-source**
- **Integrato** con IsaacGym
- **Agent, gestione memoria, rumore, scheduler, pre-processor**



CONTROL LOOP IN SKRL

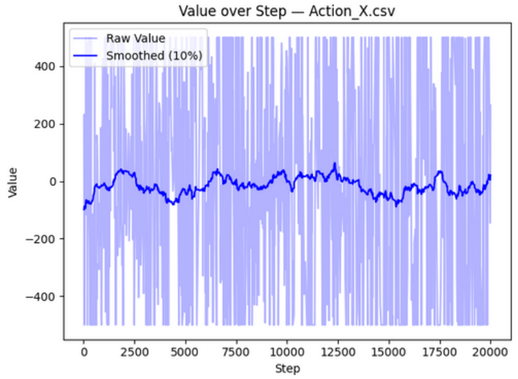
- **Agente**: in base allo stato degli ambienti ottimizza parametri di simulazione e dei modelli
- **Modelli** di rete neurale: **Attore** (Policy: azione da stato) e **Critico** (Value: stima dello stato)
- Reti neurali **asimmetriche** o **parzialmente condivise**

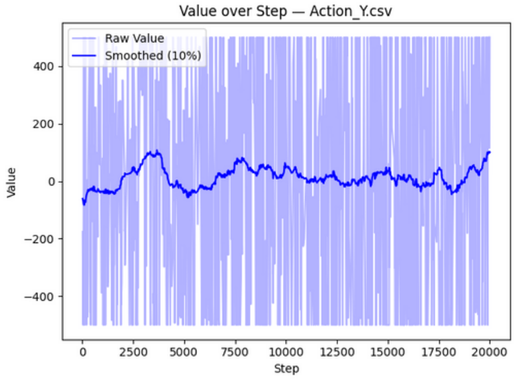


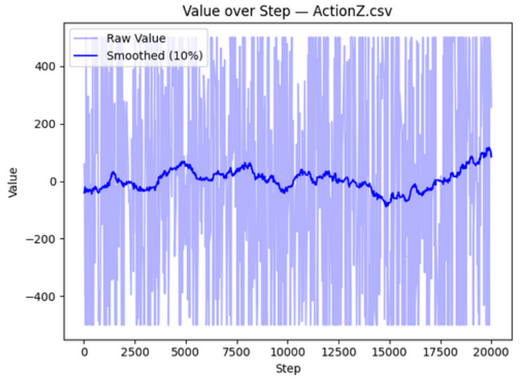


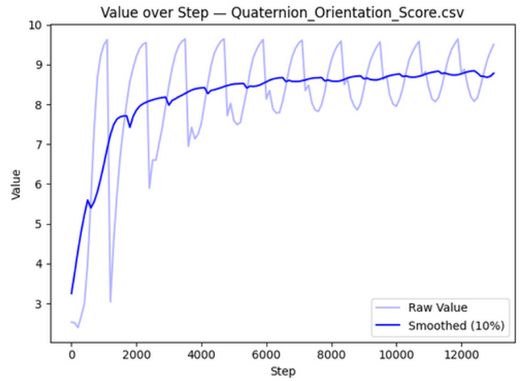
TRAINING RESULTS

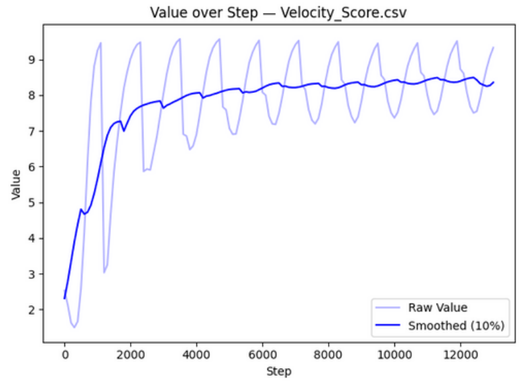
- Tool per il **monitoring dell'esecuzione** degli **ambienti** e del **training**

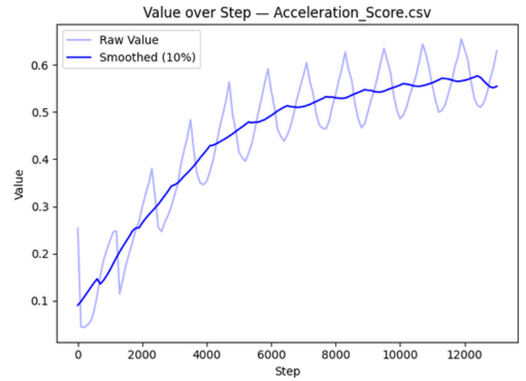


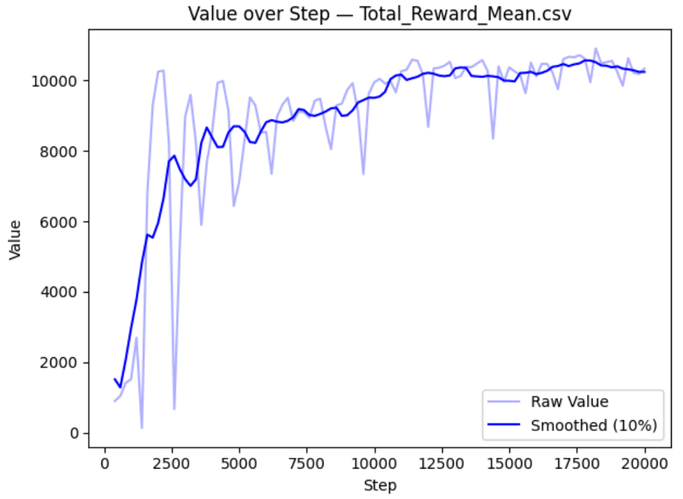






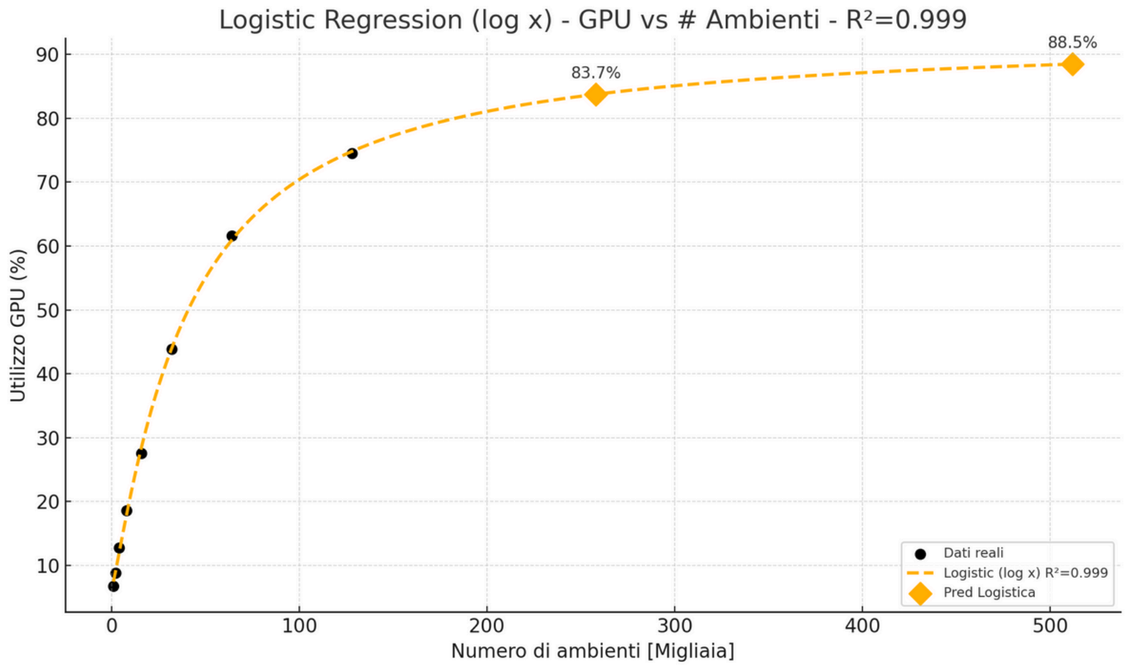


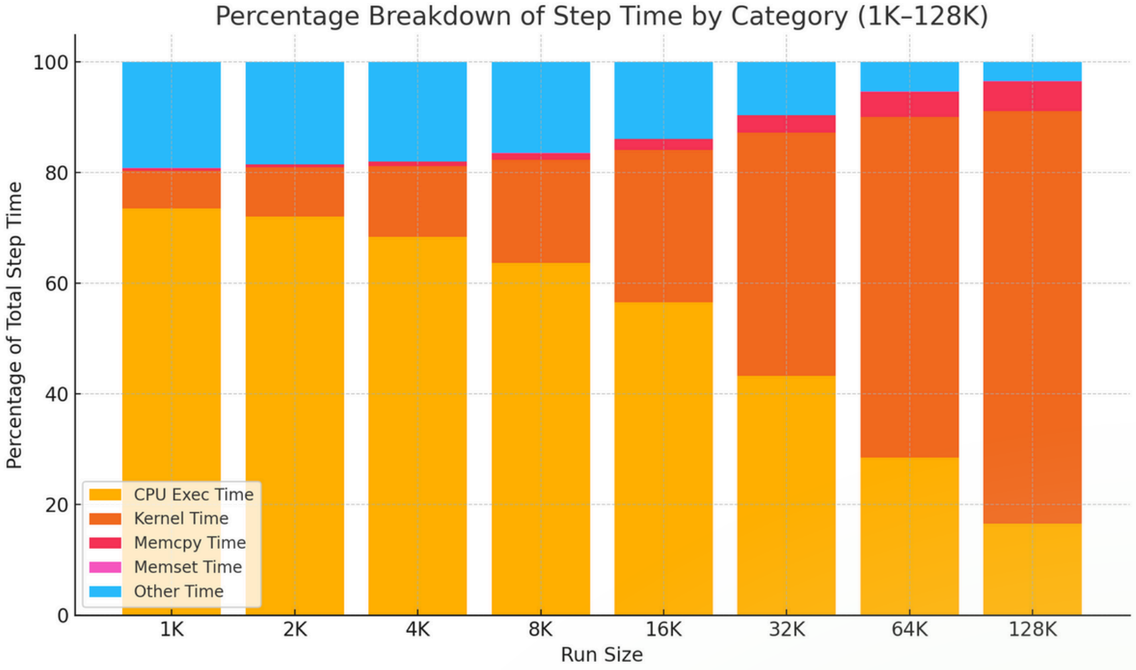




PROFILING PERFORMANCE

- Tool per l'**analisi** dell'utilizzo di **CPU, GPU e memoria**
- Dimostrazione della corretta **vettorizzazione** e perciò **scalabilità** nell'**utilizzo delle risorse**

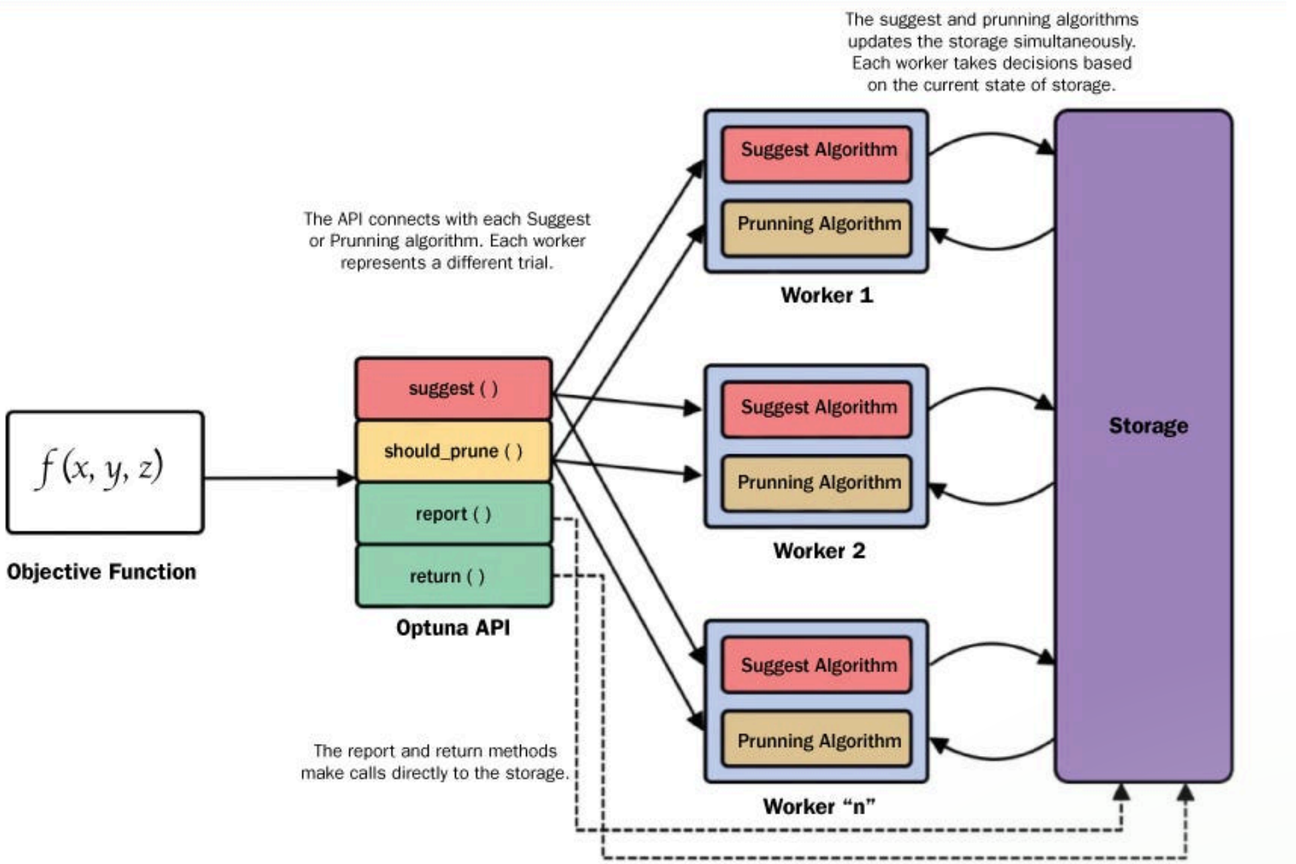




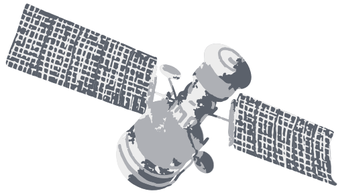
HYPERPARAMETERS OPTIMIZATION

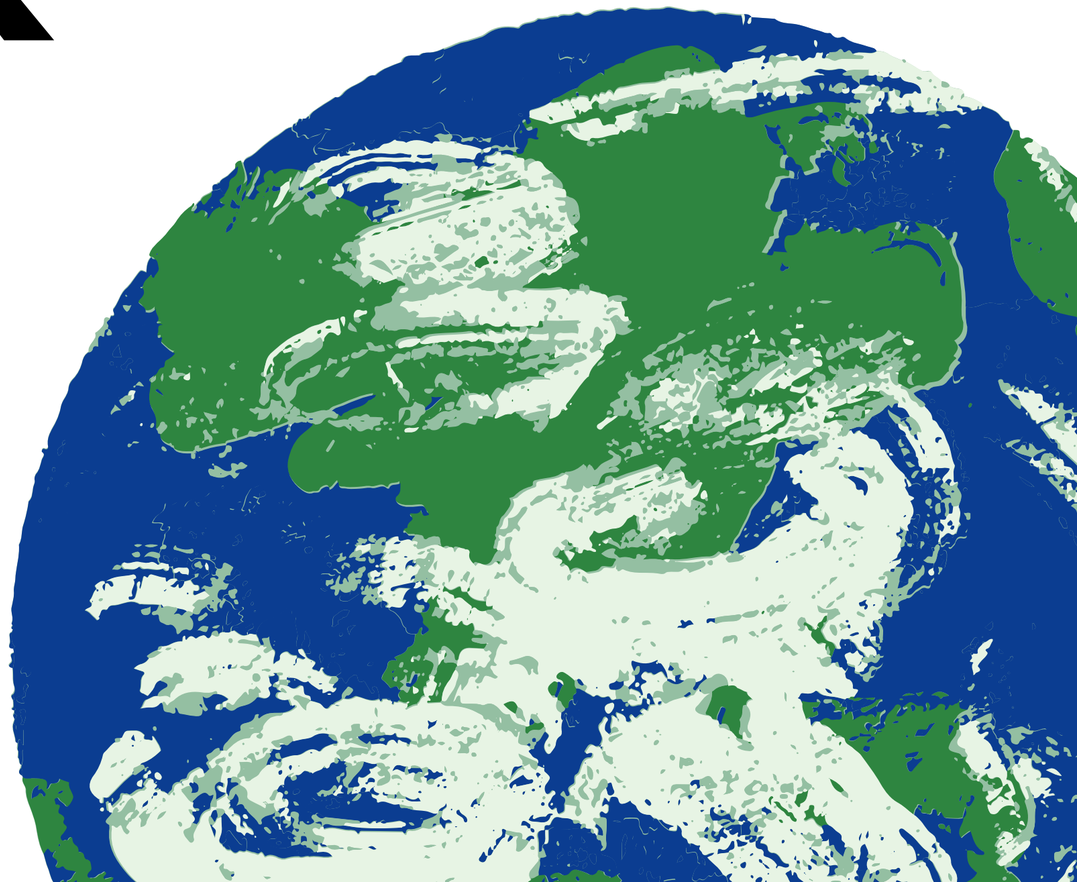
Optuna

- **Framework** open-source
- **Pruning automatico** degli ambienti meno performanti basata sulla reward
- **Parallelizzazione** automatica degli ambienti su più **thread**



FUTURE WORK





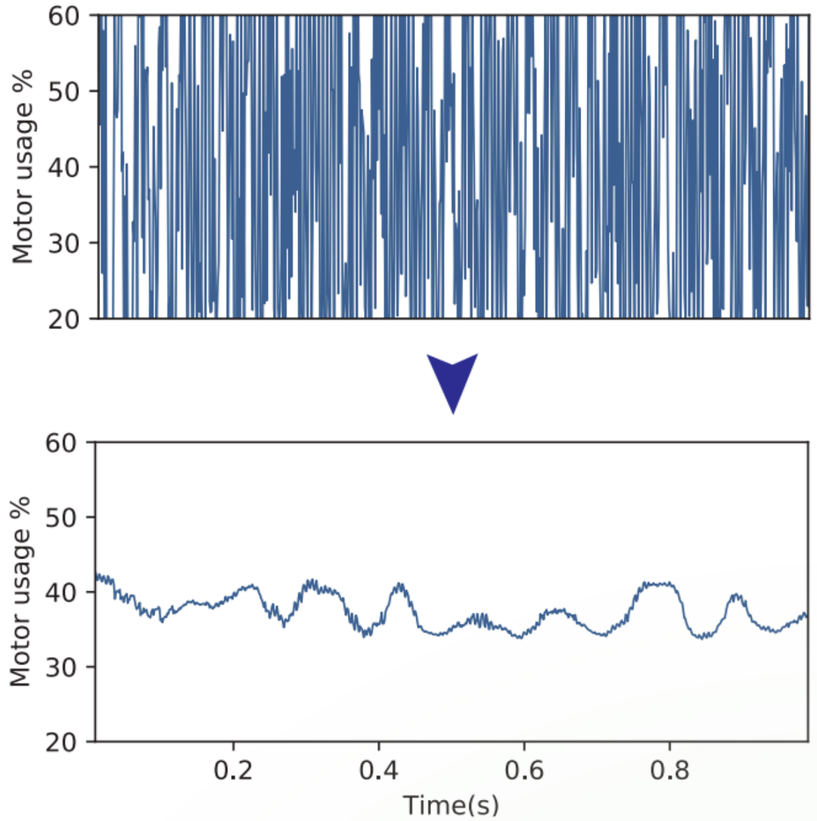
REGULARIZING TRAINING FOR SMOOTH CONTROL

Conditioning for **A**ction **P**olicy **S**moothness

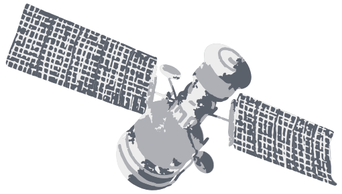
- **Regolarizzazione** per controller RL che garantisce **azioni fluide**
- **Riduce consumo energetico** attuatori

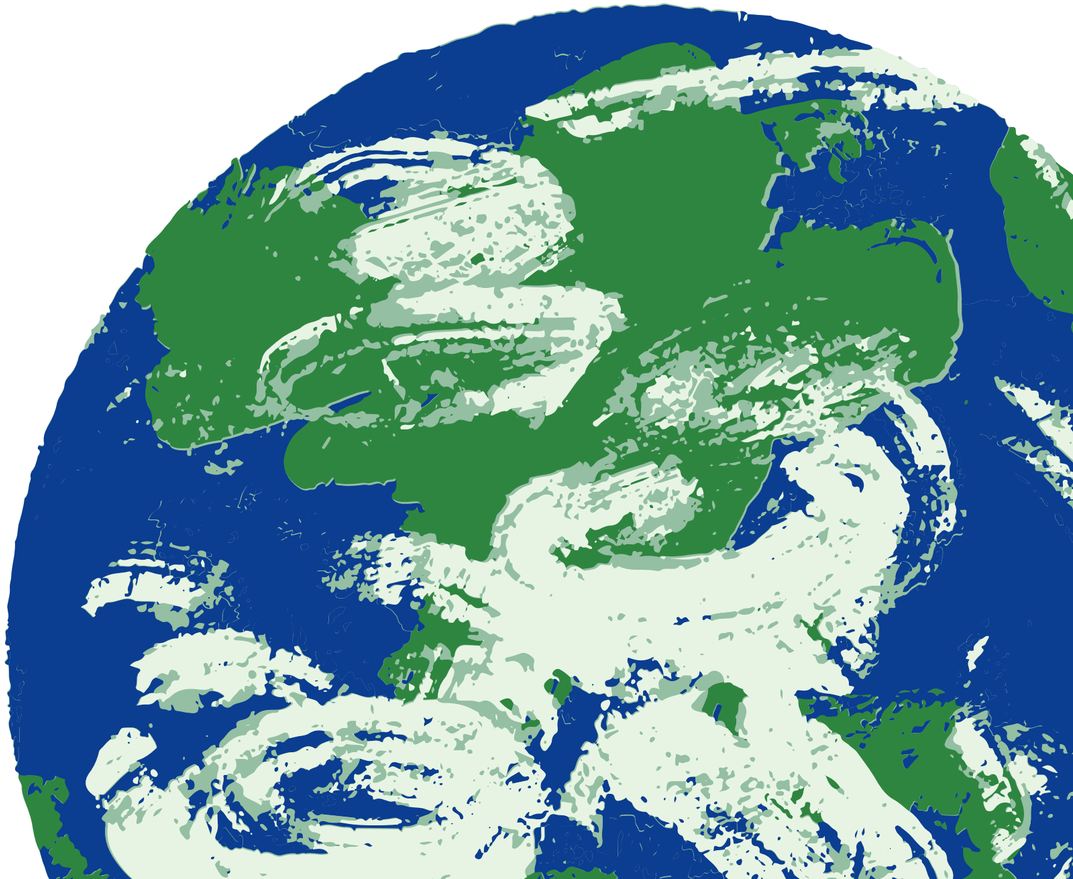
$$J_{\pi_\theta}^{CAPS} = J_{\pi_\theta} - \lambda_T L_T - \lambda_S L_S$$
$$L_T = D_T\left(\pi_\theta(s_t), \pi_\theta(s_{t+1})\right)$$
$$L_S = D_S\left(\pi_\theta(s_t), \pi_\theta(\bar{s}_t)\right) \quad \text{where } \bar{s} \sim \phi(s_t)$$

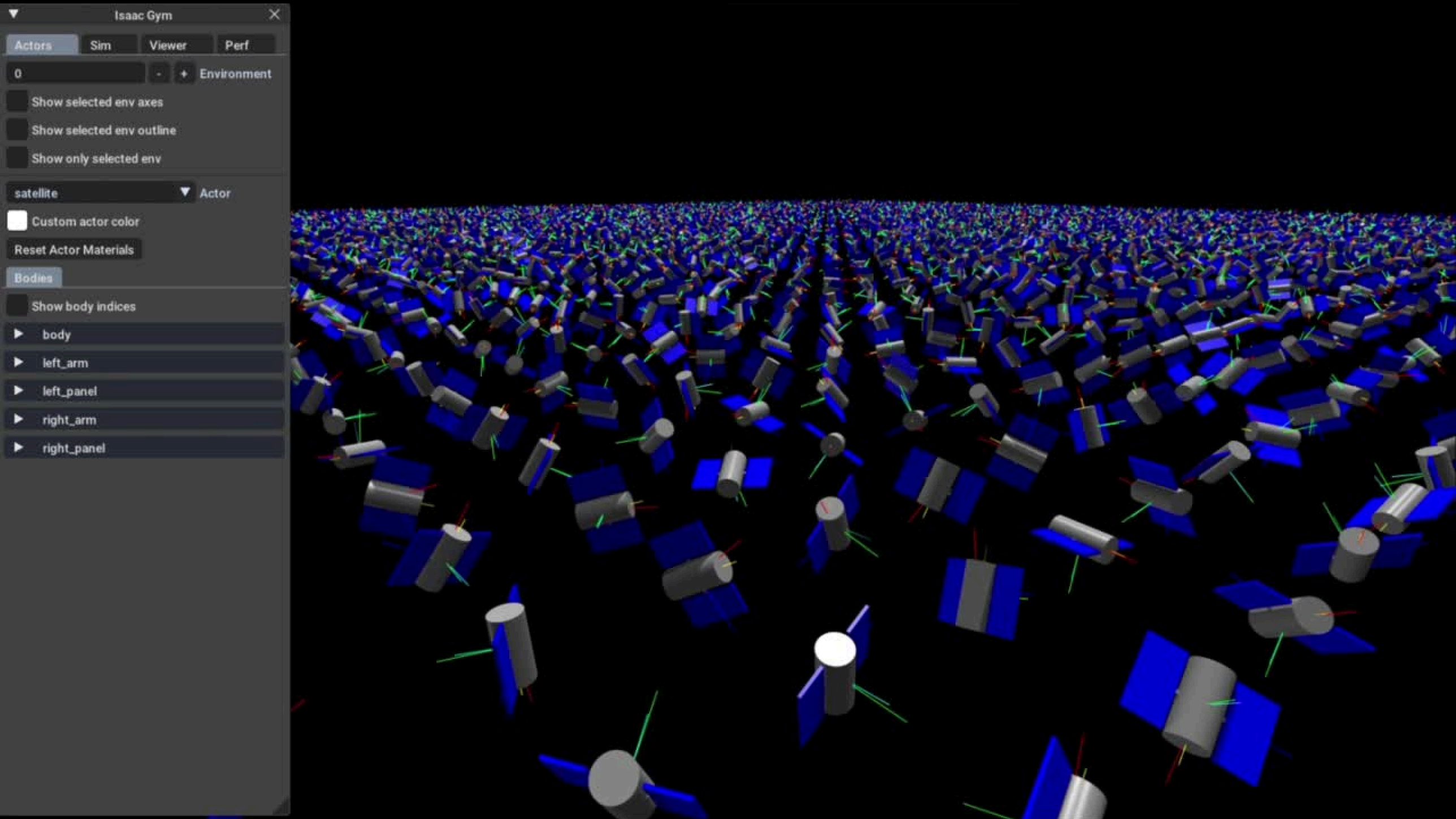
L_T penalizza la policy se cambia drasticamente l'azione da uno stato a quello successivo
L_S penalizza la policy se piccole variazioni nello stato portano ad azioni molto differenti



DEMO






Isaac Gym
Actors
Sim
Viewer
Perf
0
-
+
Environment
Show selected env axes
Show selected env outline
Show only selected env
satellite
Actor
Custom actor color
Reset Actor Materials
Bodies
Show body indices
body
left_arm
left_panel
right_arm
right_panel

GRAZIE