Physics-based Character Animation

Daniele Reda

Who am I?

- PhD Candidate at University of British Columbia
- Reinforcement Learning for Physics Based Controllers
- Applied my research in indust
    - Wayve – Autonomous D
    - Inverted AI – A riving simul
    - y Labs – Virtual Reality
    - nctuary AI – Humanoid Robotics

physics-based characters

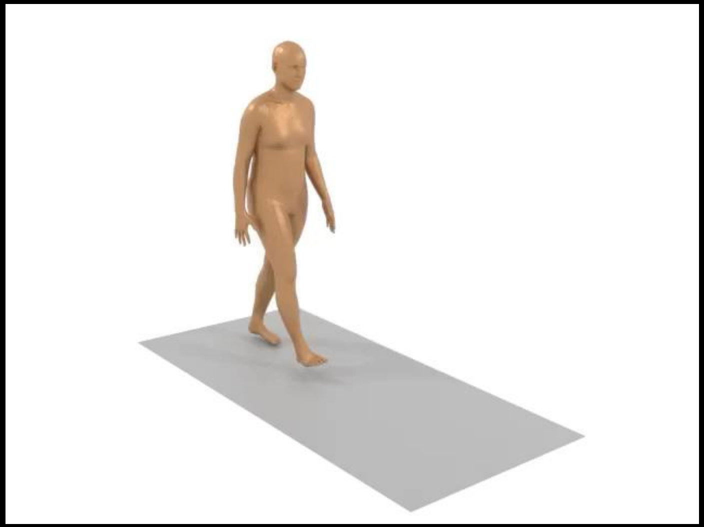
Outline

- How do we generate motions?
- Reinforcement Learning for Physics-Based Character Animation
- Why is RL hard?
- How can we use RL to learn motions?
- Real world applications of physics-based character animation

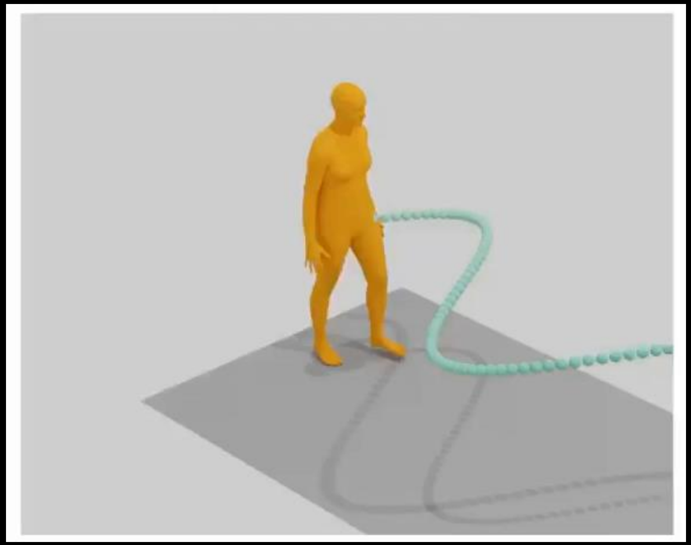
Generating motions

Artist-driven kinematic animations

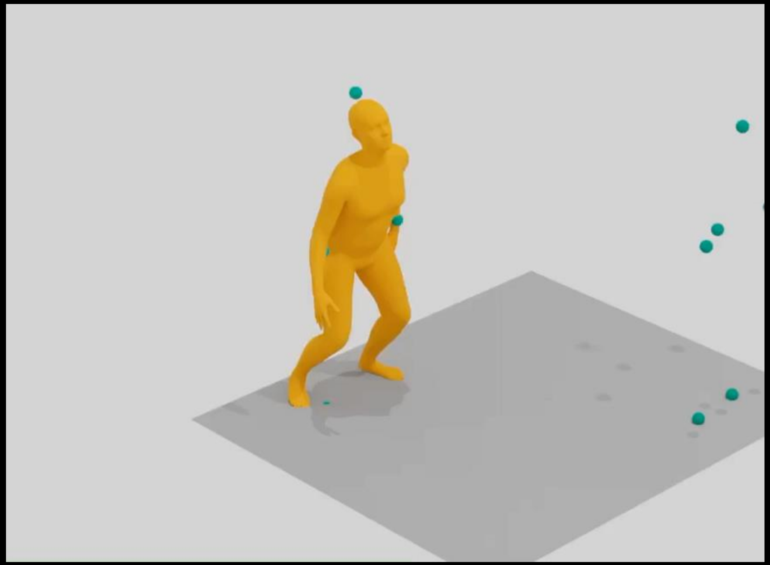
Data-driven kinematic animations



MDM [Tevet et al., 2022]



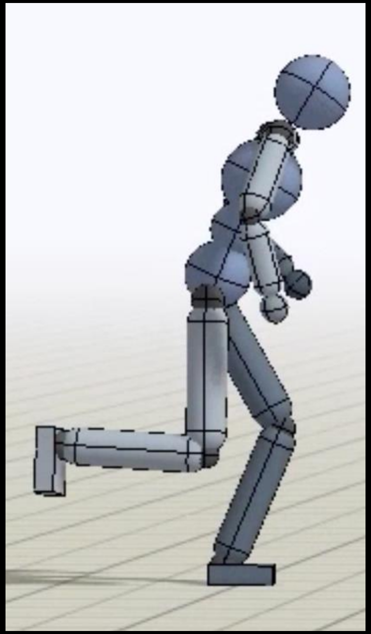
GMD [Karunratanakul et al., 2023]



OmniControl [Xie et al., 2024]

- models that act directly on poses
- Needs a lot of data to cover different scenarios
- Output might not respect physical properties:
  - self collisions, floor penetration, foot sliding, jittering

Physics-based Character Animation

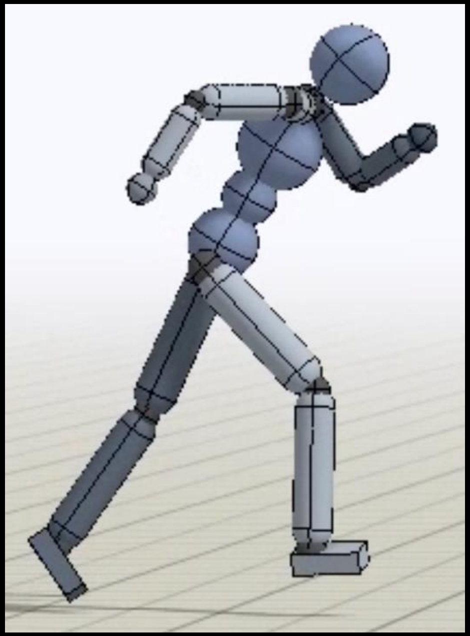


t

Control Policy

Acting on forces

Physics Simulator



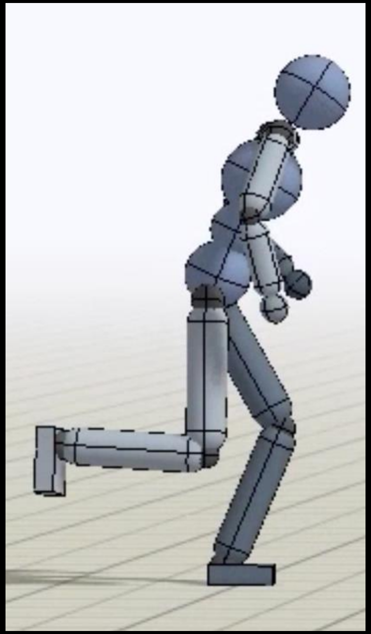
t+1

Character

Controller

Simulator

Physics-based Character Animation



t

Character

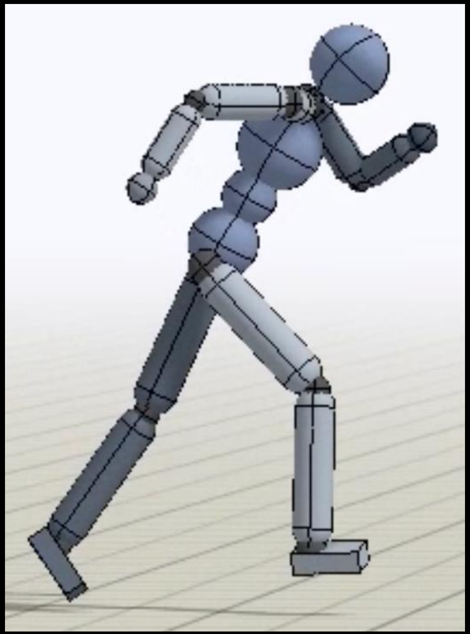
Control Policy

Acting on forces

Controller

Physics Simulator

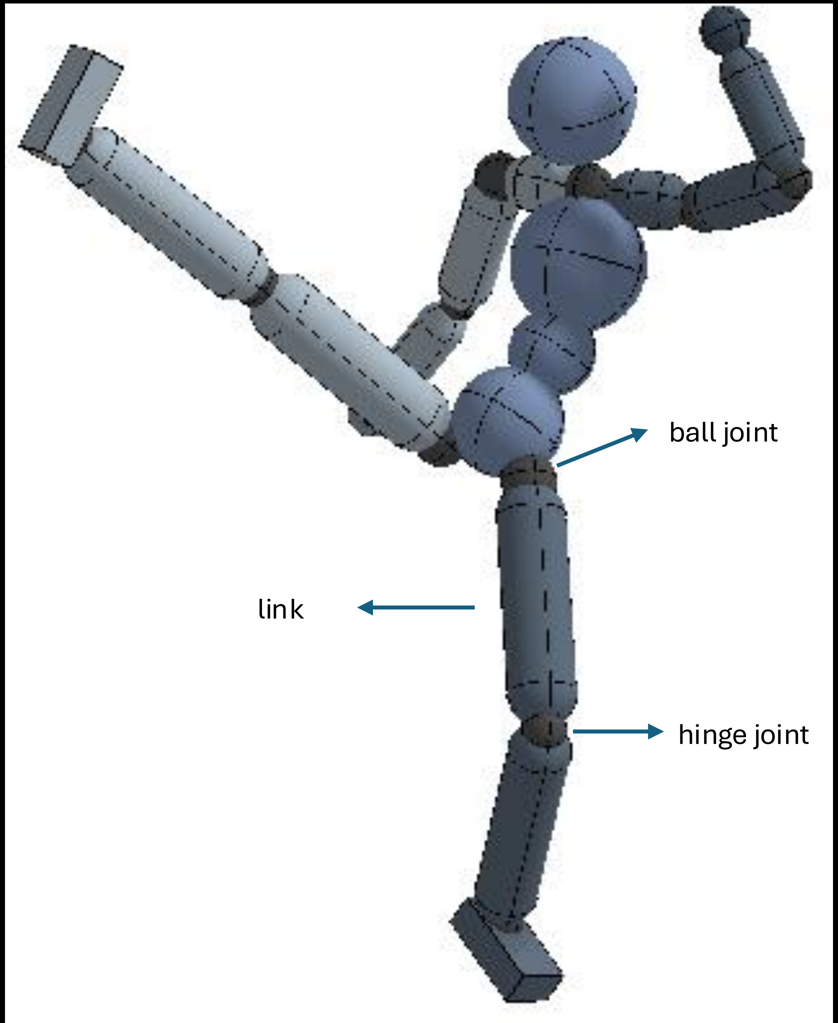
Simulator



t+1

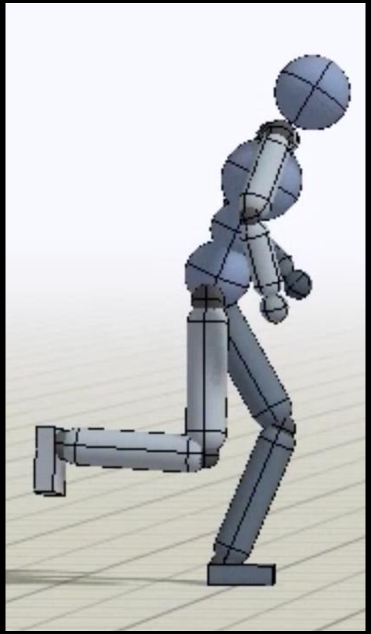
Physics-based character

Physics-based character



- Skeletons are described based on needs (more or less simplified)
- Composed of:
  - Body segments (links, bones, ...)
    - Mass information
    - Geometry Information
  - Joints
    - connects two adjacent links
    - Various types:
      - Ball joints (hips, shoulders)
      - Hinge joints (elbows, knees, ankles)
    - Constrained by joint limits
  - Actuators:
    - To generate forces and torques to move
    - one per active DOF

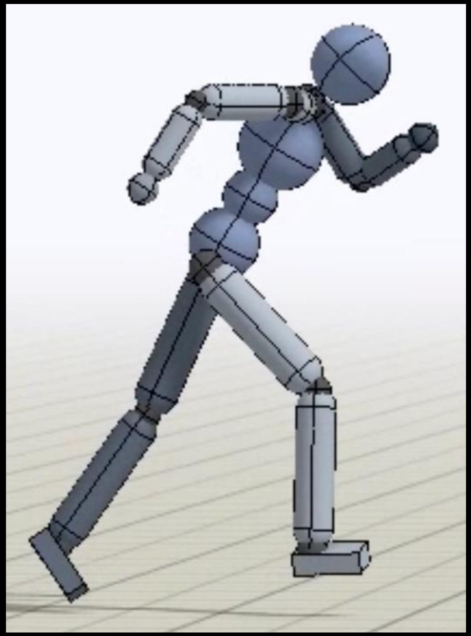
Physics-based Character Animation



Control Policy

Acting on forces

Physics Simulator



t

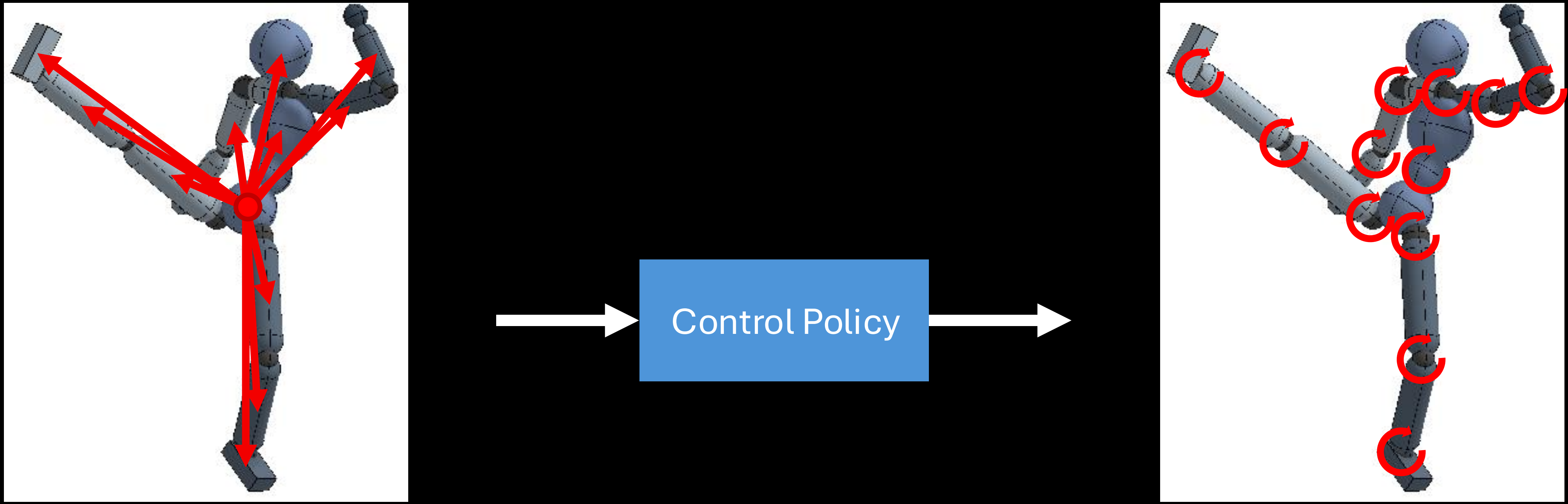
t+1

Character

Controller

Simulator

Motion Controller

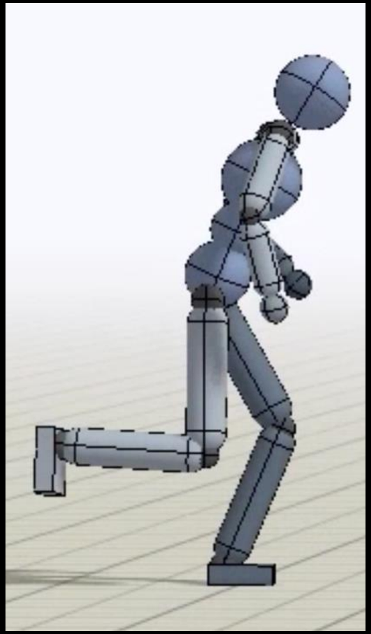


state

root position, orientation and velocities, joint angles and velocities

forces & torques

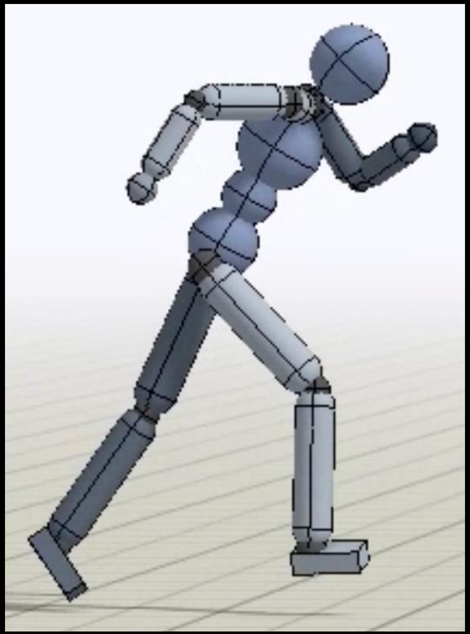
Physics-based Character Animation



Control Policy

Acting on forces

Physics Simulator



t

t+1

Character

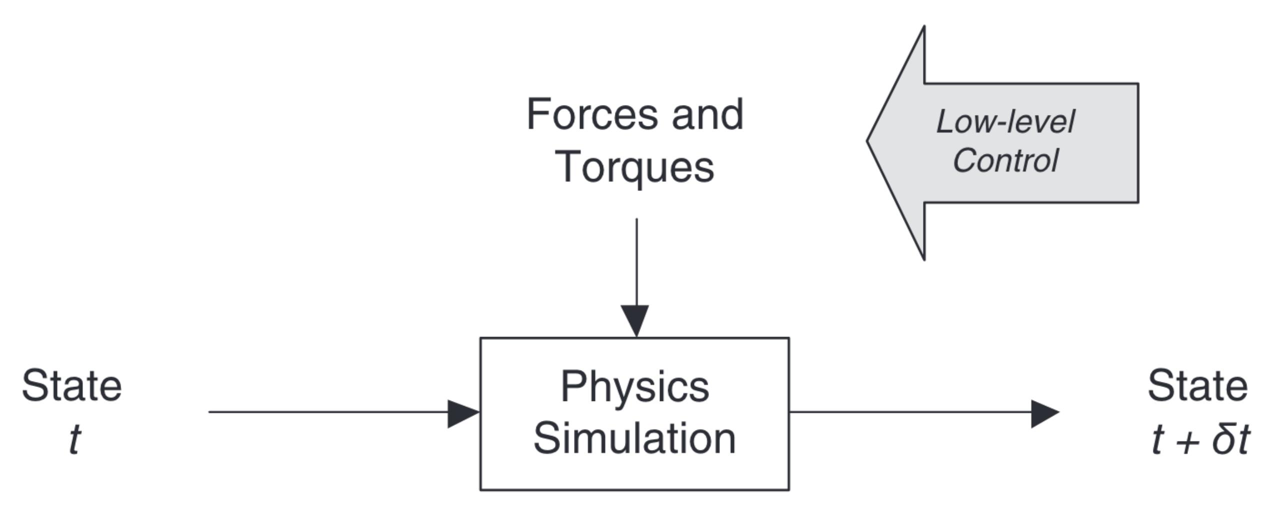
Controller

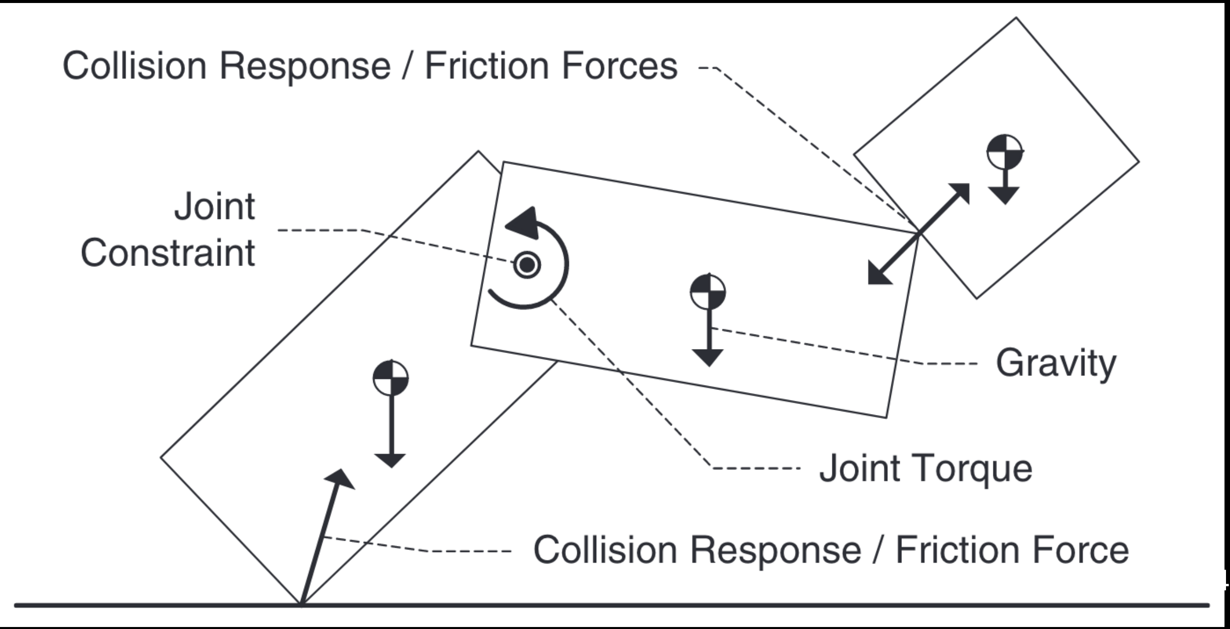
Simulator

Simulator

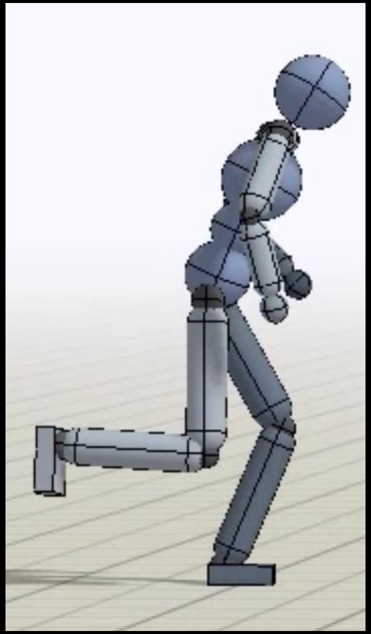
- Simulates physical properties of the world
- Control happens through forces and torques

- 3 steps:
  - Collision detection
  - Forward dynamics
  - Numerical Integration





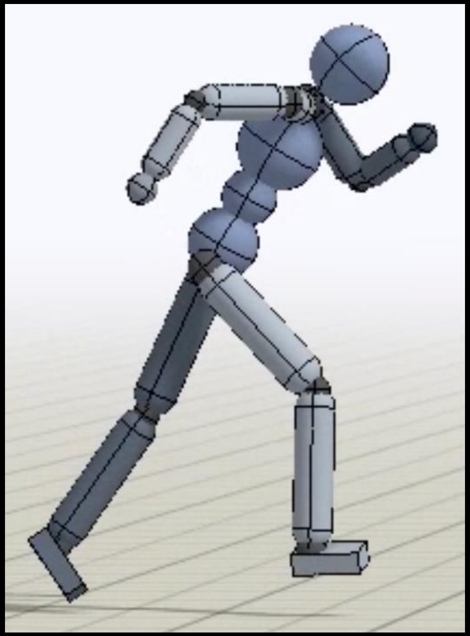
Physics-based Character Animation



Control Policy

Acting on forces

Physics Simulator



t

t+1

Character

Controller

Simulator

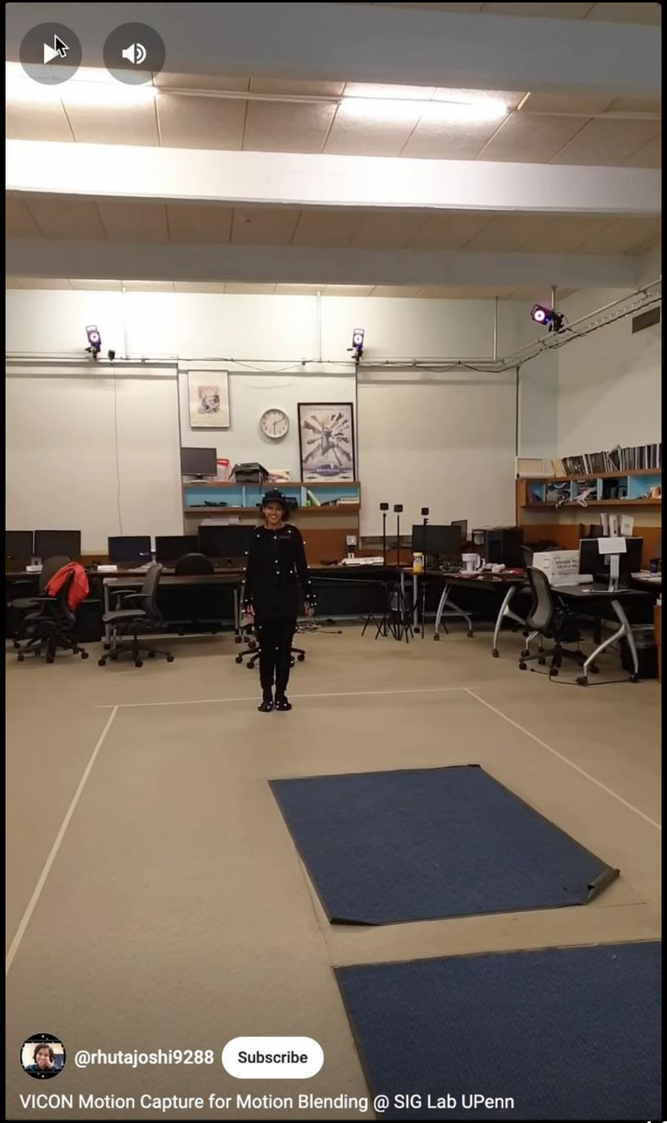
Some basic ideas to learn a controller

- How can we produce a controller?
- Let's start from a reference motion trajectory:

  trajectory = $\{s_0, s_1, s_2, \ldots, s_N\}$

- 3 basic concepts:
  - Inverse dynamics
  - PD Controllers
  - Behavioral Cloning

they are all solving an imitation task



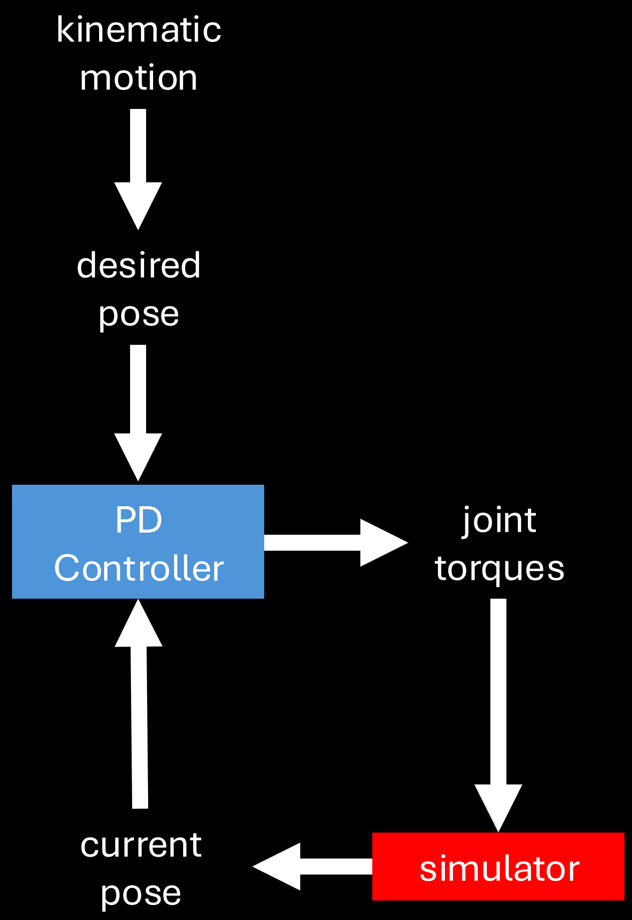
Inverse dynamics

- We can learn an inverse dynamics model to compute forces needed to go from s_t to s_{t+1}

$$\mathrm{a} = f^{-1}(s_t, \frac{ds}{dt})$$

- Overall hard to learn
  - requires access to the equation of motions (no access from the simulator)
  - under-determined (multiple solutions possible)
  - computationally expensive to be accurate
  - Very sensible to model errors -> instability and compounding errors

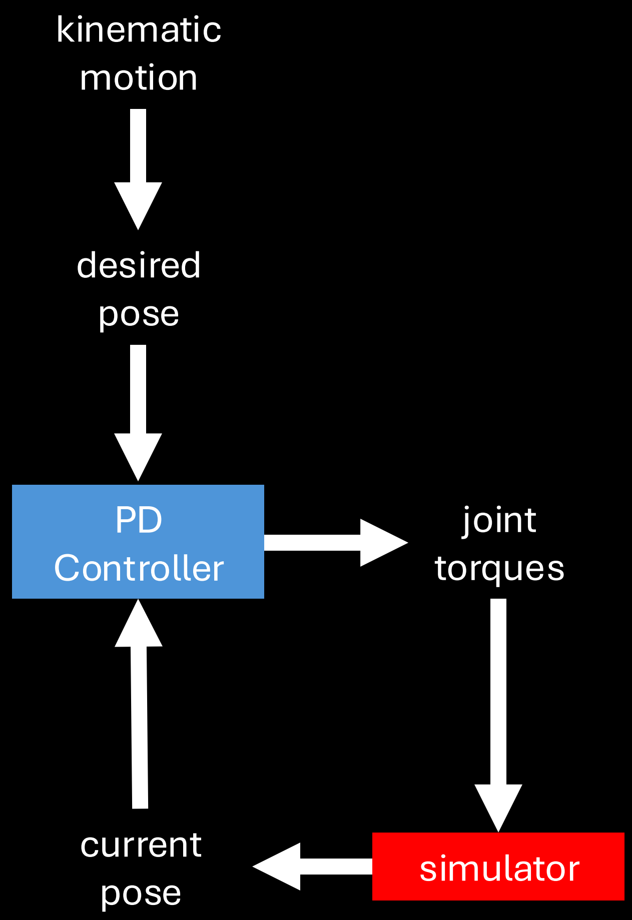
PD Controller



$$\tau = k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}).$$

- computes a joint torque linearly proportional to difference between current and desired state

PD Controller



- local feedback controllers operate on individual joints
- k_p and k_d require manual tuning and domain expertise
    - too low: controller lags behind motion
    - too high: stiff and unresponsive
- the controller relies on having a good desired motion to follow – how to produce it?

Behavioral Cloning

- Let's say we have a trajectory of both states and joint torques

  trajectory = {(s_0, a_0), (s_1, a_1), (s_2, a_2), … (s_N, a_N)}

$$\mathcal{D} = \langle s_i, a_i \rangle_{i=1}^{N}$$

- We could use supervised learning to learn a controller

$$\pi_\theta(s_t) : \mathcal{S} \rightarrow \mathcal{A}$$

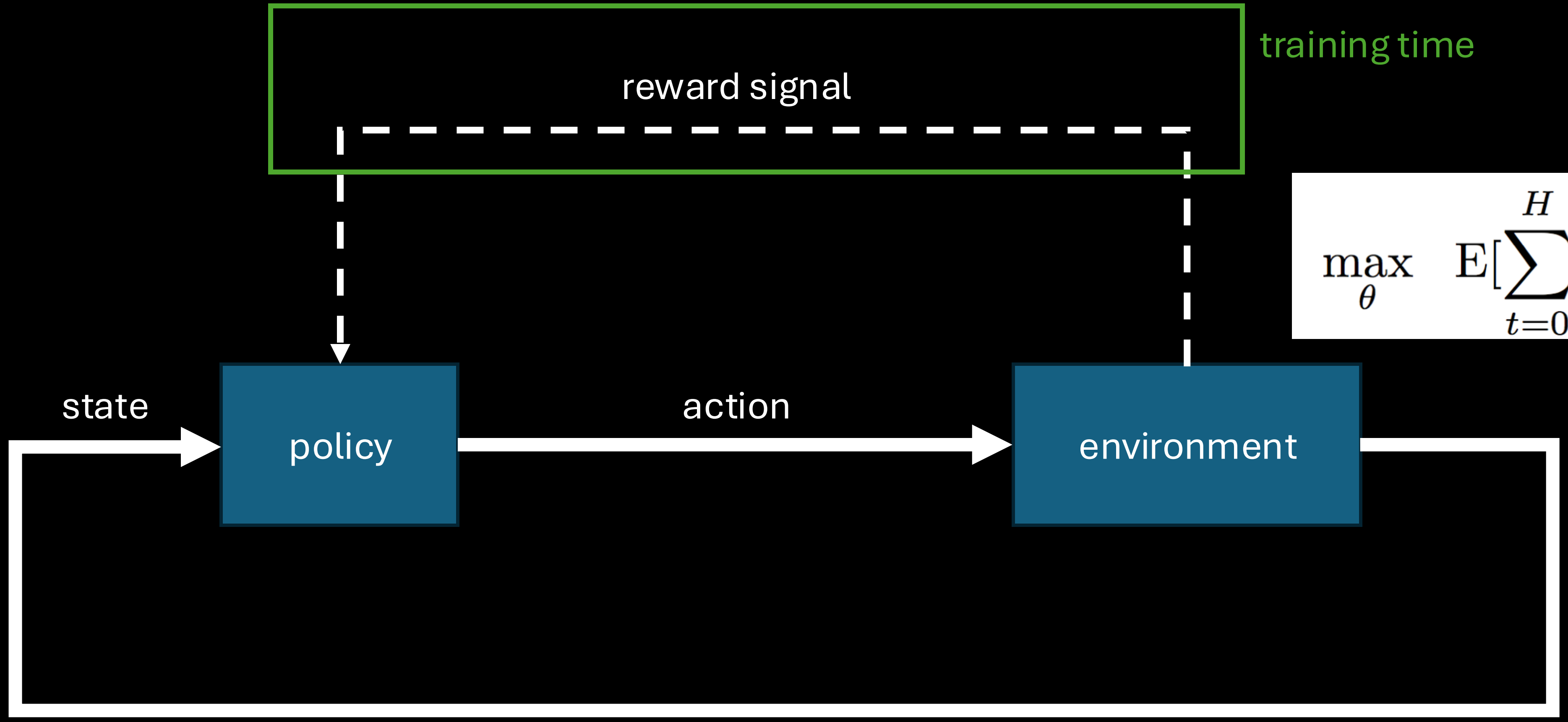
$$\min_\theta \sum_i \mathcal{L}(\pi_\theta(s_i), a_i)$$

- What could go wrong?

- Equivalent to an open loop controller:
  - Suffers from distribution shift and compounding error
  - It is very instable, not robust == it might get good training scores, but in closed loop it will not work
  - You can fix this: DAGGER [Ross et al, AISTATS 2011]

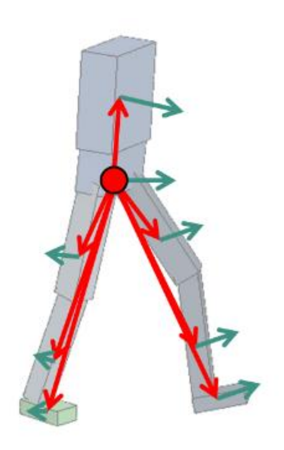
Reinforcement Learning

Reinforcement Learning



$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

Reinforcement Learning



positions, orientations, velocities, joint angles

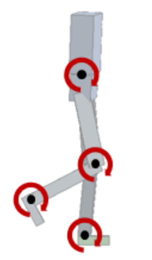
state

controller

training time

reward signal

action



joint torques

physics simulator

$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

In principle:

- specify rewards
- trial-and-error learning

Reinforcement Learning: In Practice

reward description?

ir

d

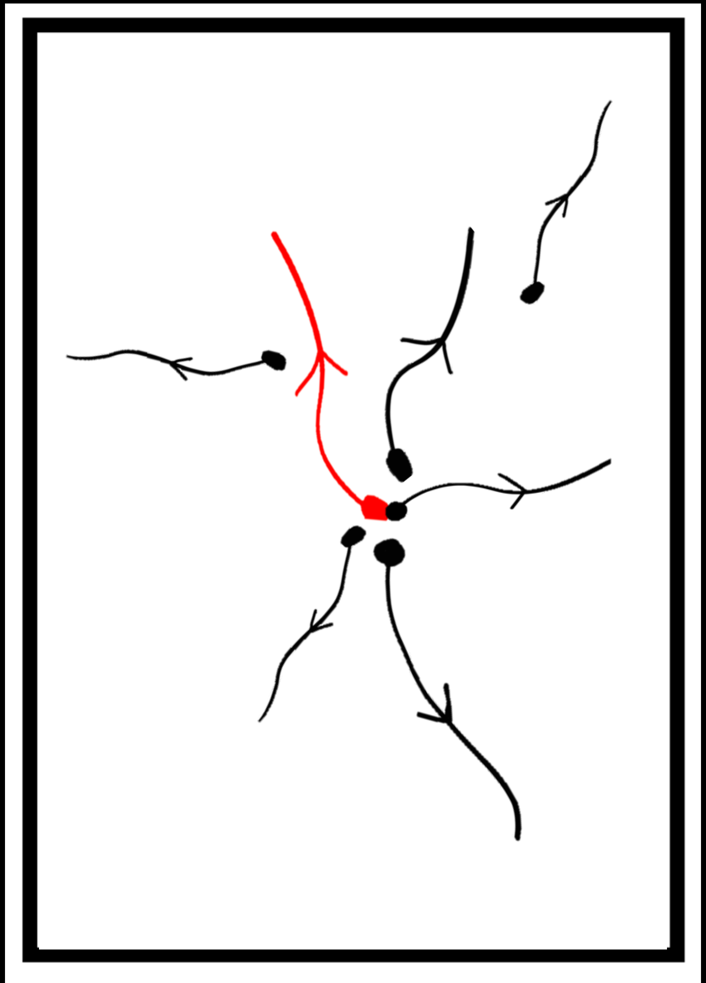
IT'S HARD

Reda et al., "Learning to locomote: Understanding how environment design matters for deep reinforcement learning." *MIG*, 2020

Why is RL hard? Exploration

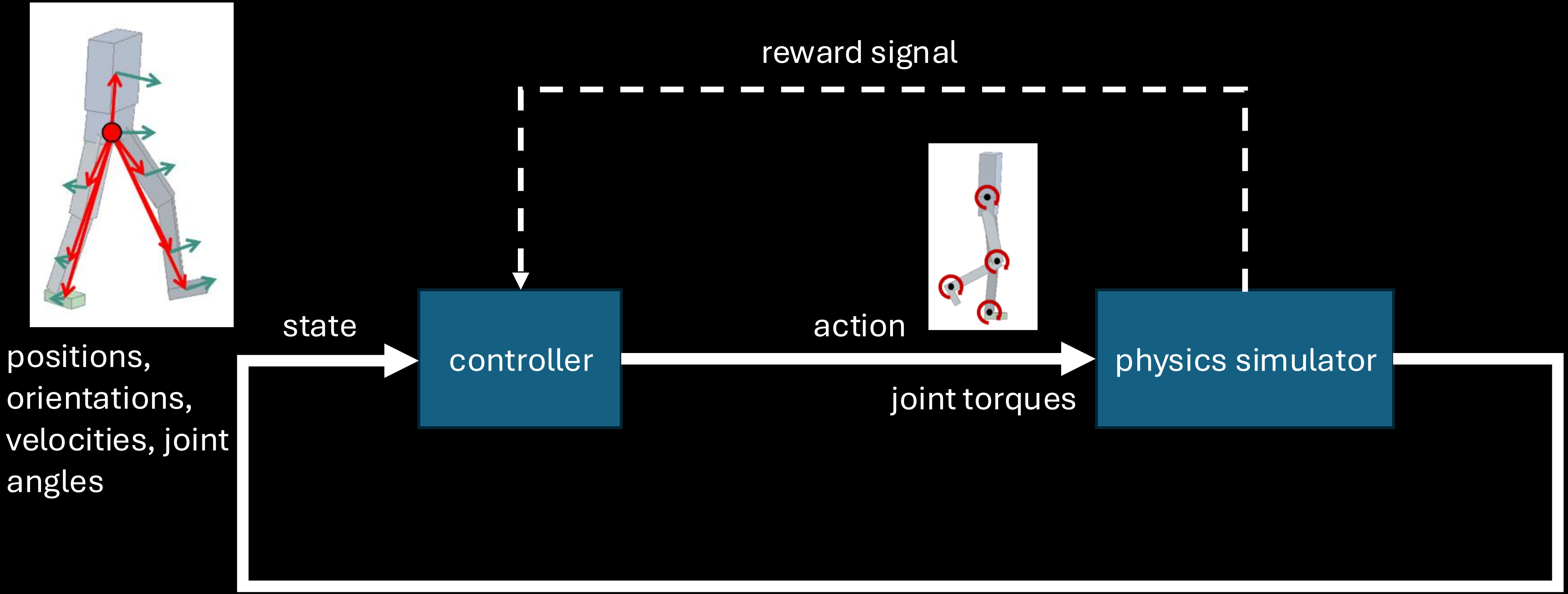
- Imagine trying to control a fighter jet with no knowledge
  - Hundreds of buttons and joysticks
  - Your feedback is just seeing where you are going
- Huge exploration dimensionality
- Reward assignment problem
- Exploration vs Exploitation

Why is RL hard? Exploration



RL for physics based animation

Reinforcement Learning



3D humanoid control from pure reward is one of the most difficult tasks in typical RL benchmark suites

Pure-objective reward

The reward function is defined as $0.05 r_{lv} + 0.05 r_{av} + 0.04 r_b + 0.01 r_{fc} + 0.02 r_{bc} + 0.025 r_s + 2 \cdot 10^{-5} r_\tau$. The individual terms are defined as follows.

$$r_{lv} := \begin{cases} \exp\left(-2.0(v_{pr} - 0.6)^2\right) & v_{pr} < 0.6 \\ 1.0 & v_{pr} \geq 0.6 \\ 0.0 & \text{zero command} \end{cases}$$

$$r_{av} := \begin{cases} \exp\left(-1.5(\omega_{pr} - 0.6)^2\right) & \omega_{pr} < 0.6 \\ 1.0 & \omega_{pr} \geq 0.6 \end{cases}$$

$$r_b := \exp(-1.5 v_o^2) + \exp(-1.5 ||({}^{B}_{IB}\omega)_{xy}||^2)$$

$$r_{fc} := \sum_{i \in I_{swing}} \left(\mathbb{1}_{\mathcal{F}_{clear}}(i) / |I_{swing}|\right) \in [0.0, 1.0].$$

$$r_{bc} := -|I_{c,body} \setminus I_{c,foot}|.$$

$$r_s := -||(r_{f,d})_t - 2(r_{f,d})_{t-1} + (r_{f,d})_{t-2}||.$$

$$r_\tau := -\sum_{i \in joints} |\tau_i|.$$

$$r_{target} = k_{target} \exp(-d/k_d),$$

$$r_{\text{energy}} = -4.5 \frac{1}{N_j} \sum_j | a_j \cdot v_j | - 0.225 \frac{1}{N_j} \sum_j | a_j |^2,$$

$$r_{\text{limit}} = -0.1 \sum_j \mathbb{1}_{j \notin [0.99 l_j, 0.99 u_j]}(j),$$

$$r_{\text{posture}} = -| \alpha_x | \mathbb{1}_{\alpha_x \notin [-0.4, 0.4]}(\alpha) - | \alpha_y | \mathbb{1}_{\alpha_y \notin [-0.2, 0.4]}(\alpha),$$

$$r_{\text{speed}} = -\max(\|v_{\text{root}}\|_2 - 1.6, 0),$$

$$r_{\text{alive}} = 2$$

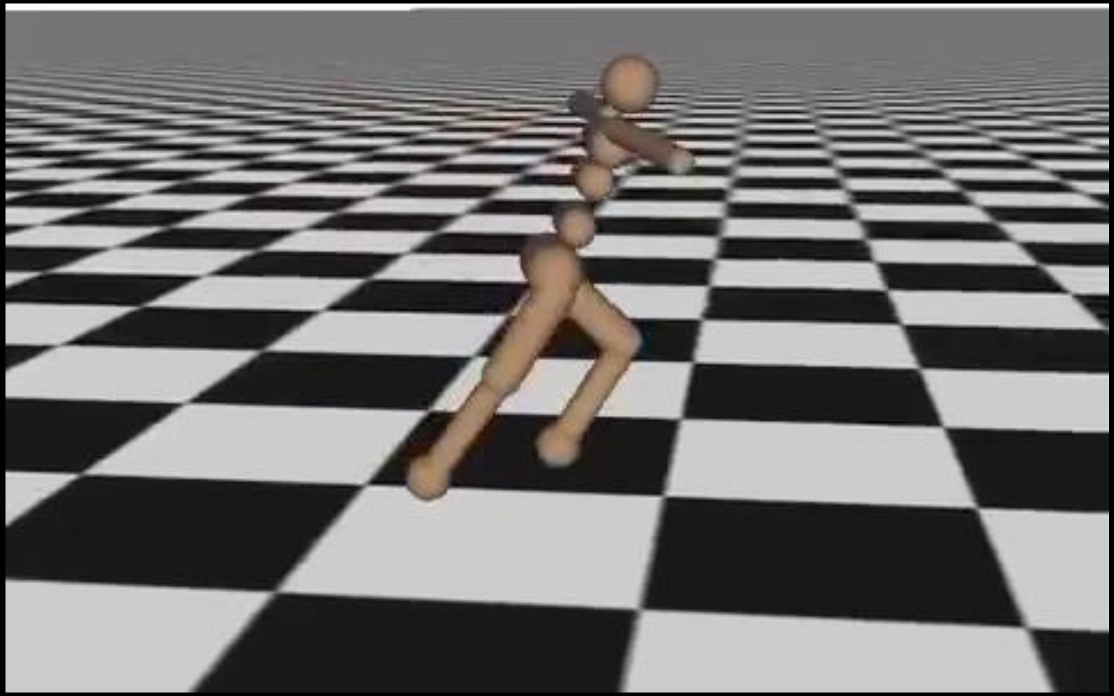
$$r_{progress} = (d_{t-1} - d_t) / dt$$

$$r_{addition} = r_{\text{energy}} + r_{\text{limit}} + r_{\text{posture}} + r_{\text{speed}} + r_{\text{alive}}.$$

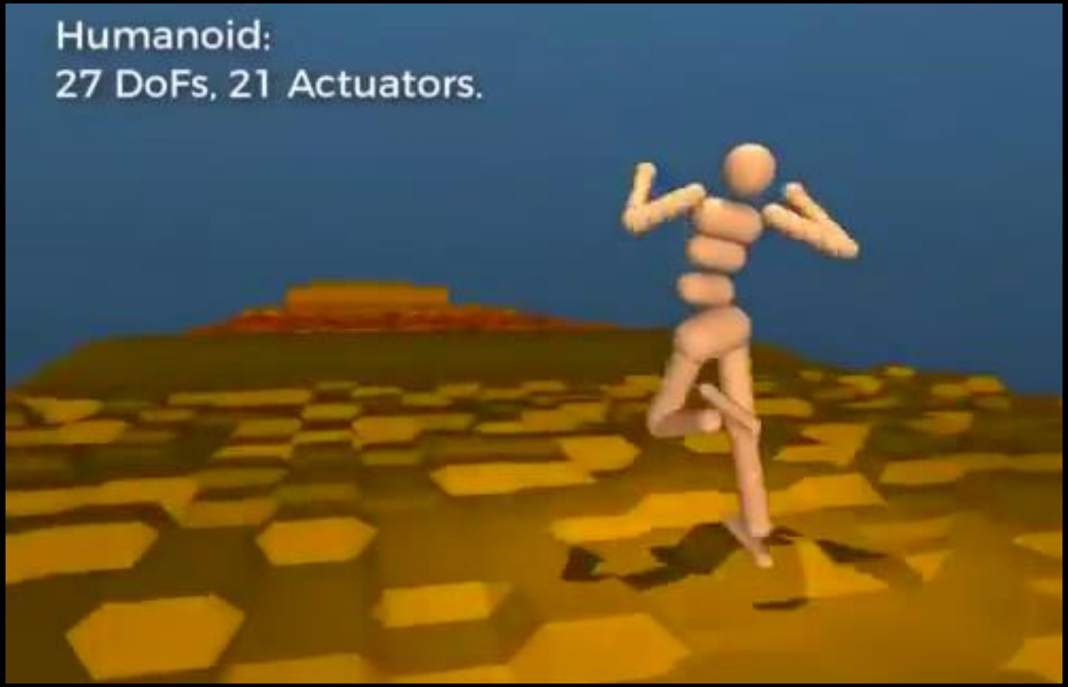
Lee, Joonho, et al. "Learning quadrupedal locomotion over challenging terrain." *Science robotics*, 2020

Xie, Zhaoming, et al. "Allsteps: curriculum-driven learning of stepping stone skills." SCA 2020

And after careful tuning you get this

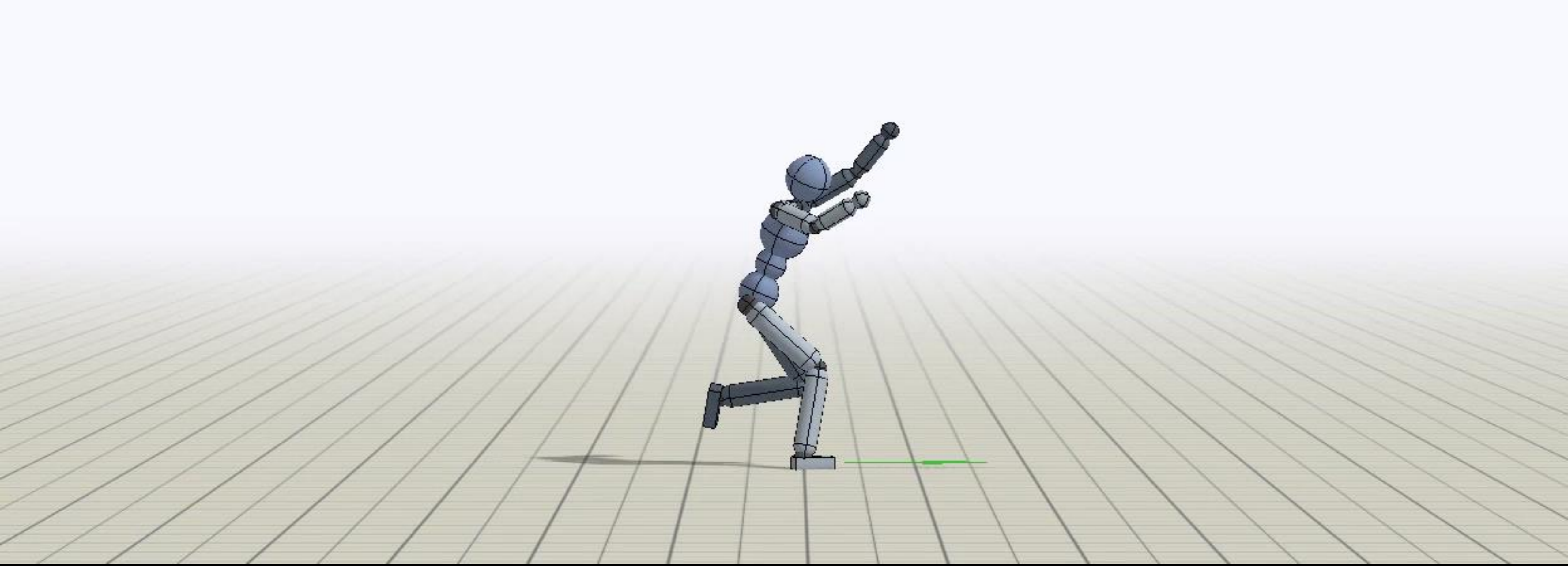


[Schulman et al. 2016]



[Heess et al. 2017]

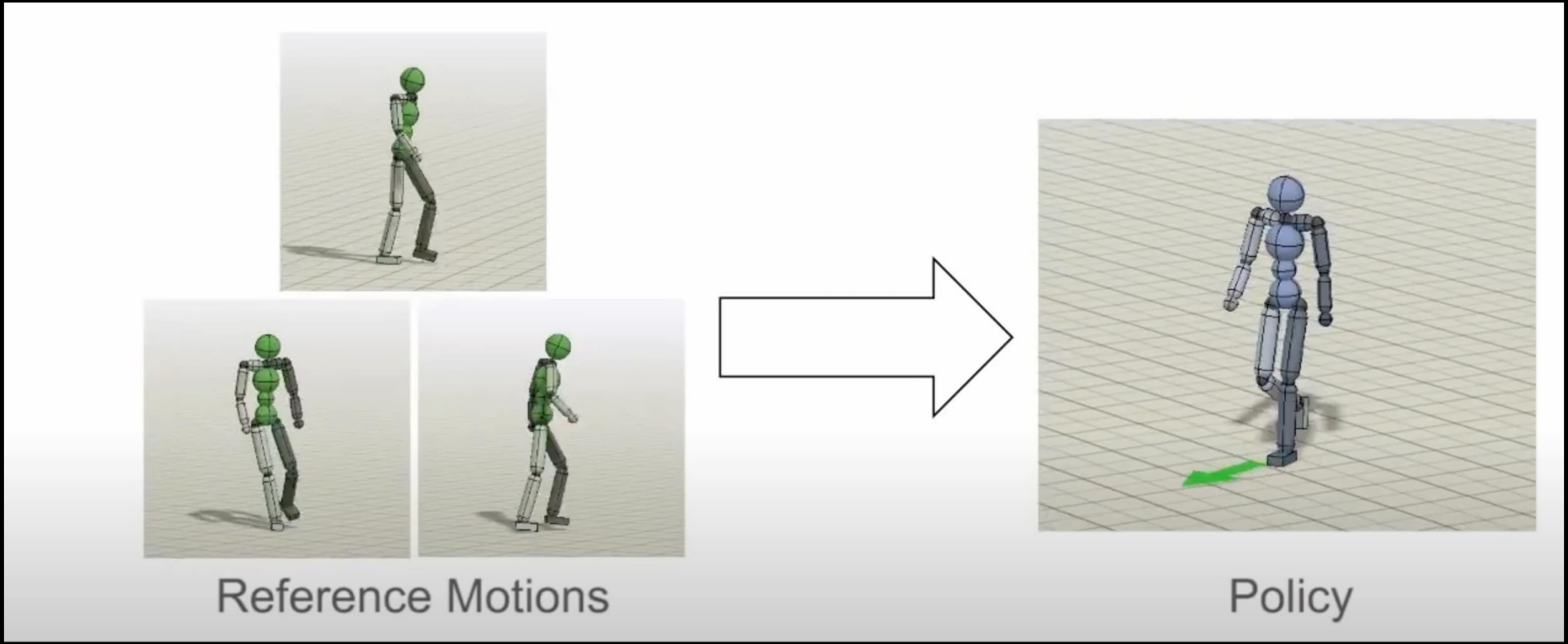
or this



Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

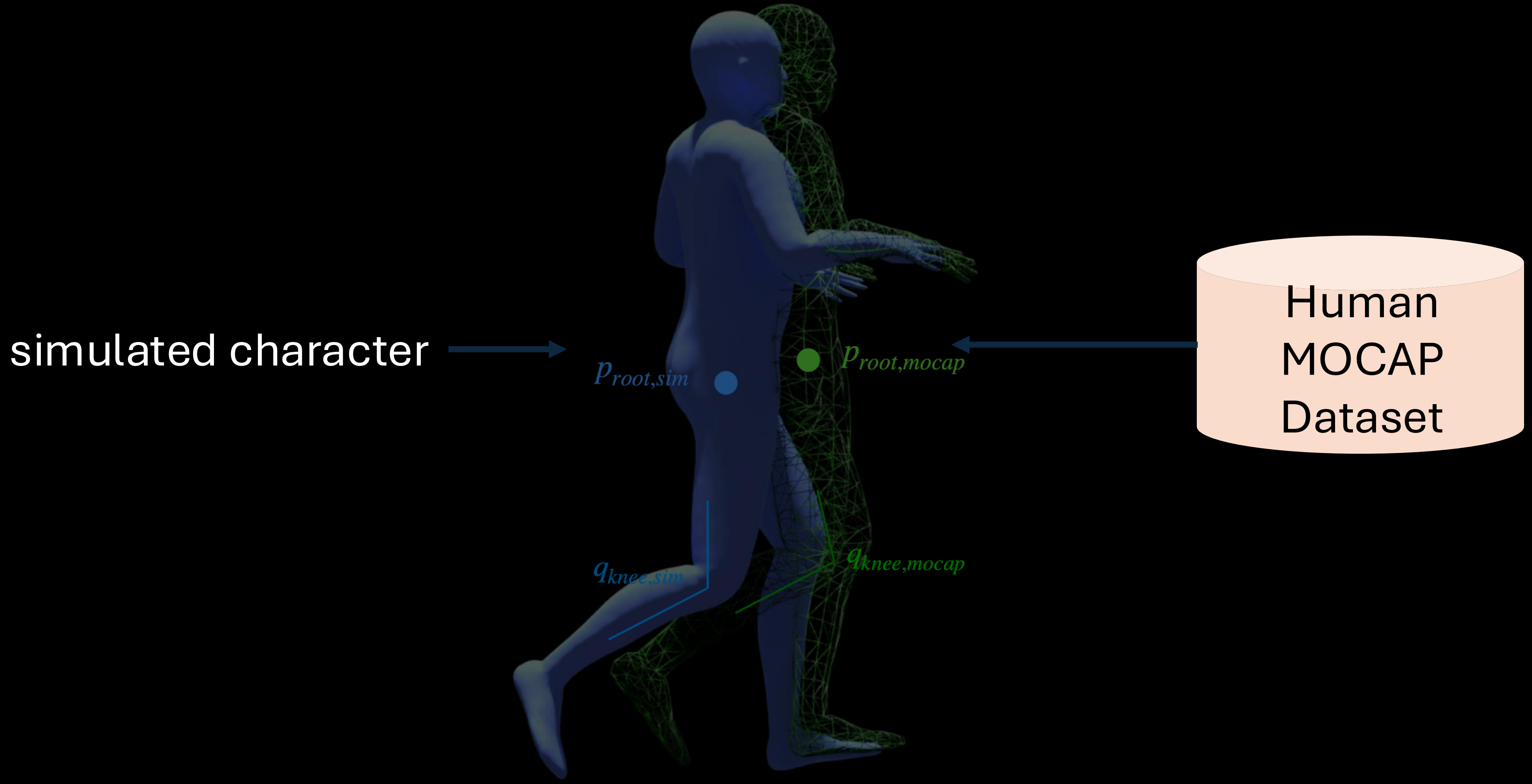
Ways to constrain exploration

Can we use motion capture data to constrain exploration?



Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

Can we use motion capture data to constrain exploration?

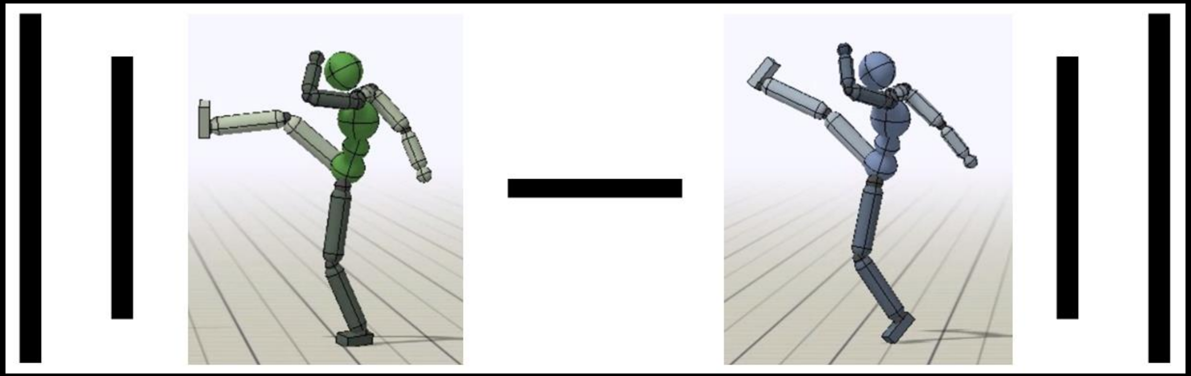


Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

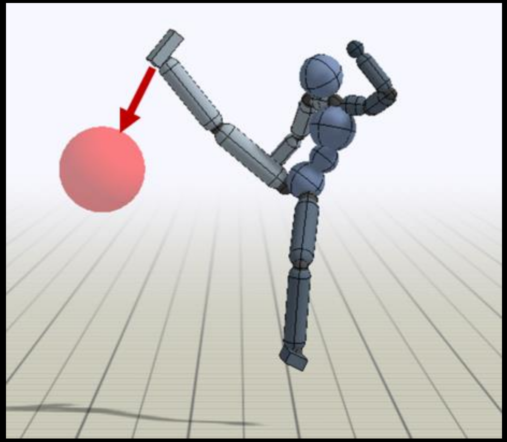
Can we use motion capture data to constrain exploration?

$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

Imitation Objective

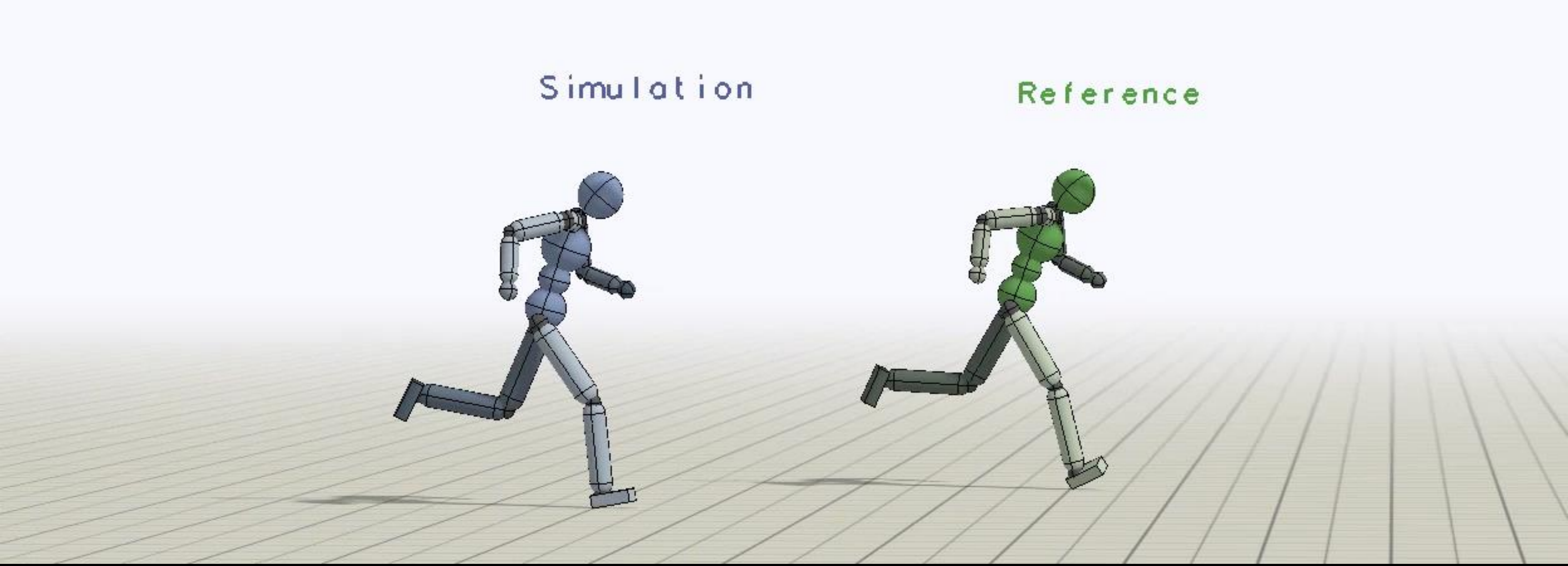


Task Objective



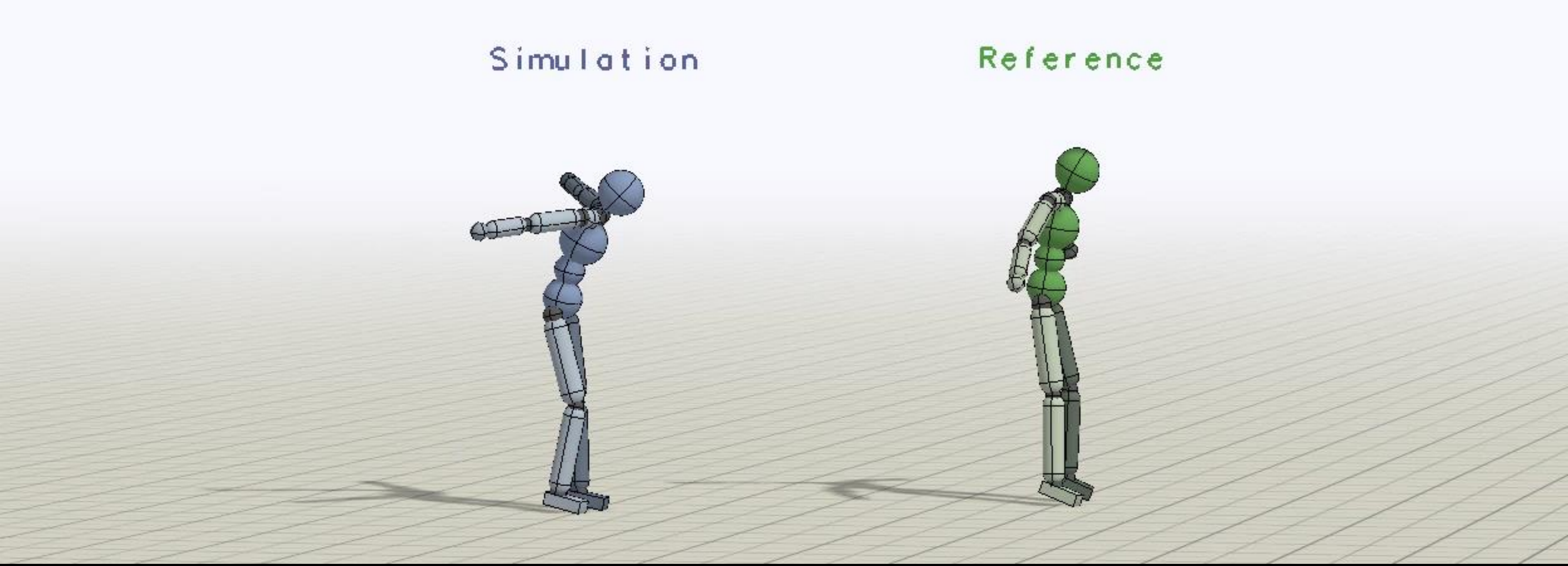
Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

Humanoid: running



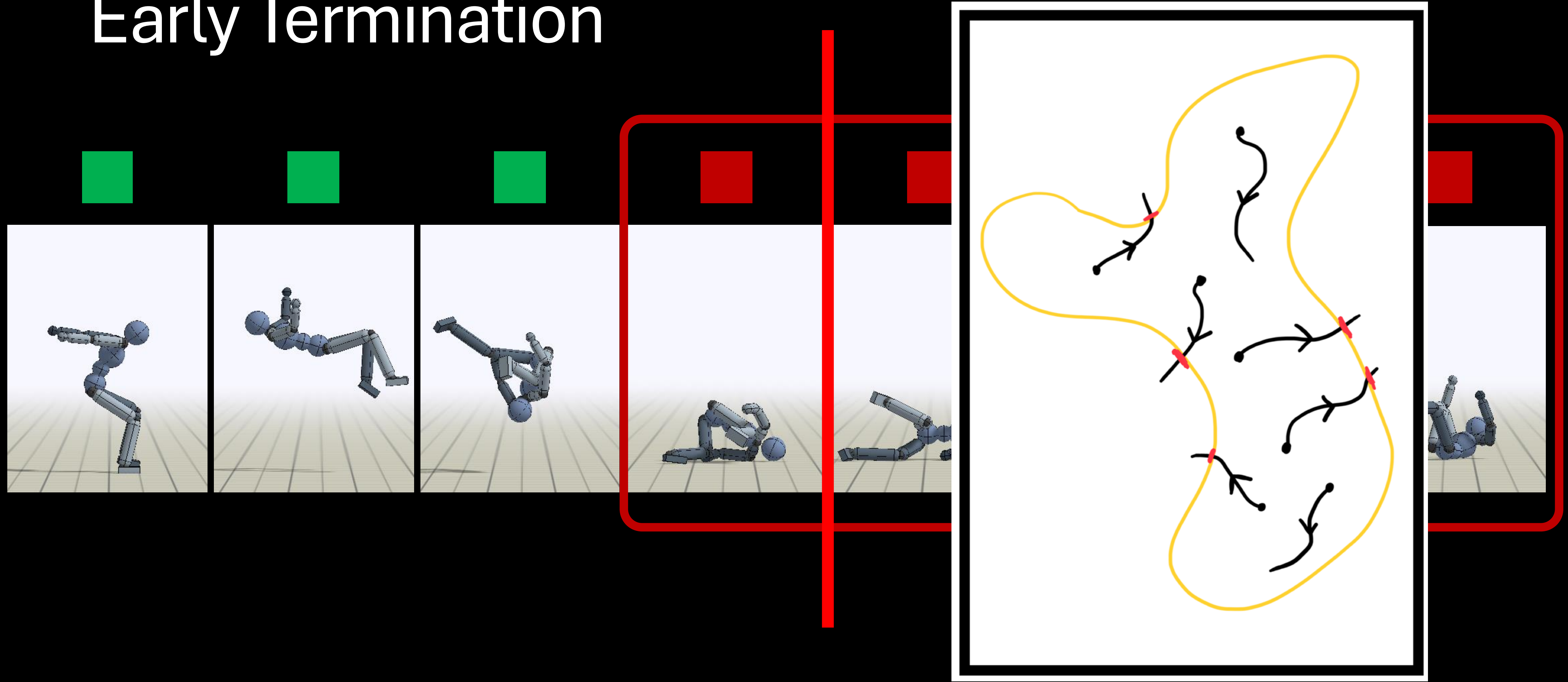
Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

What about a backflip motion?



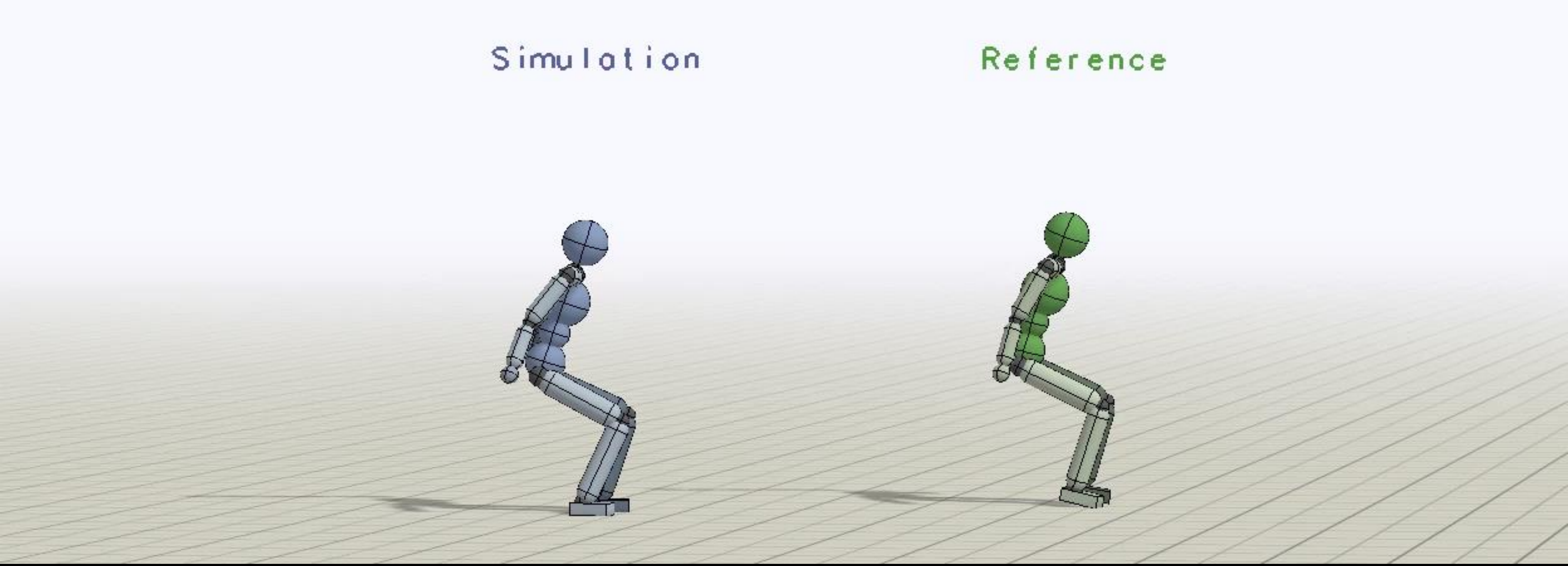
Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

Early Termination



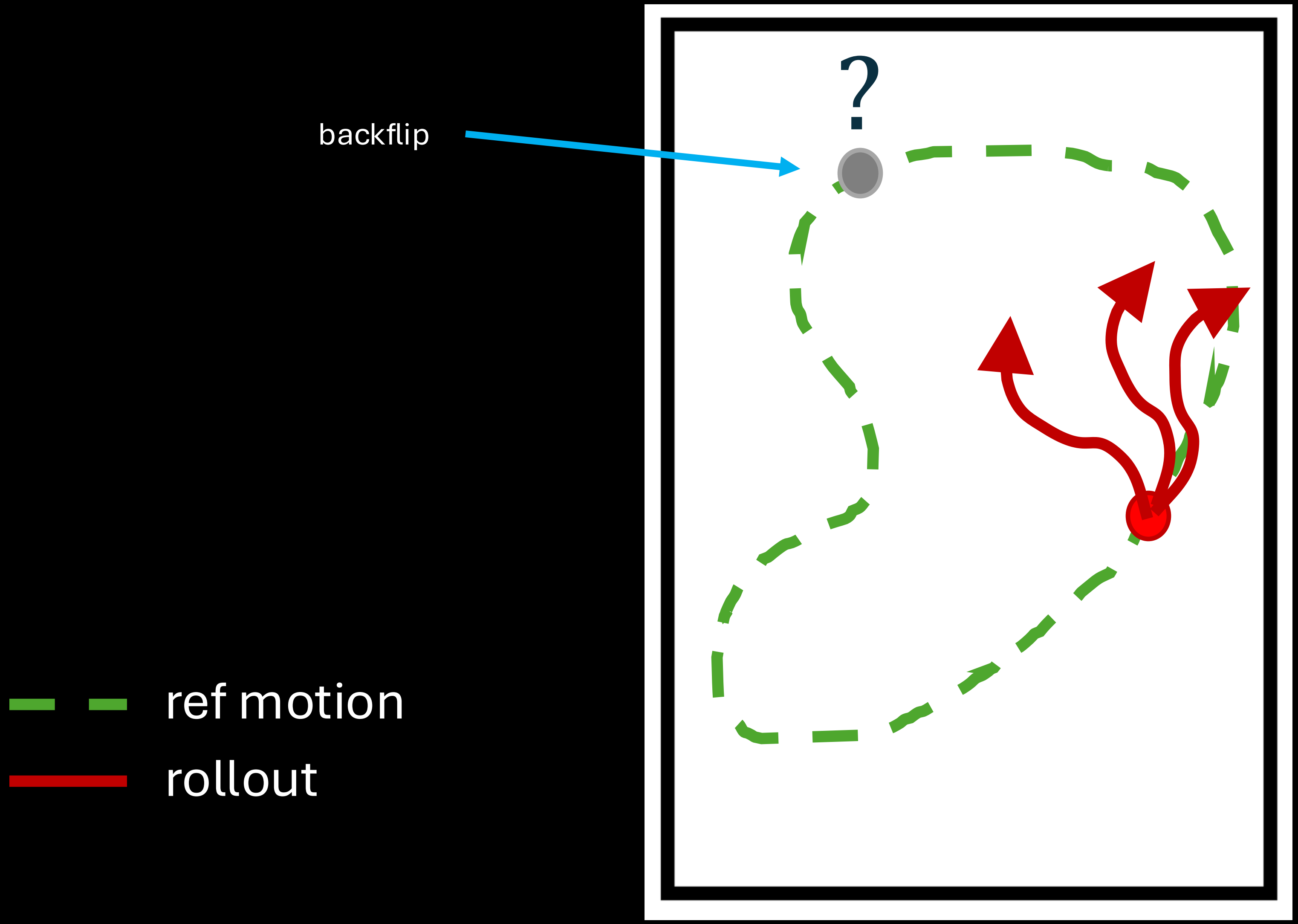
Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

With Early Termination

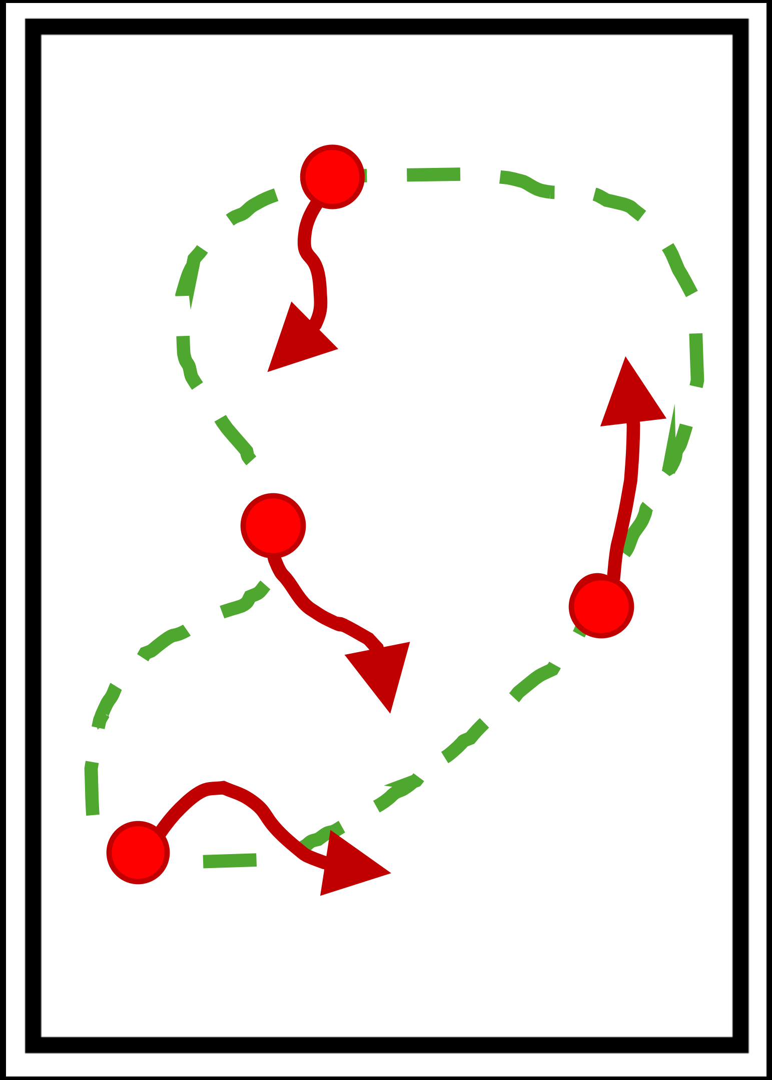


Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

State Initialization



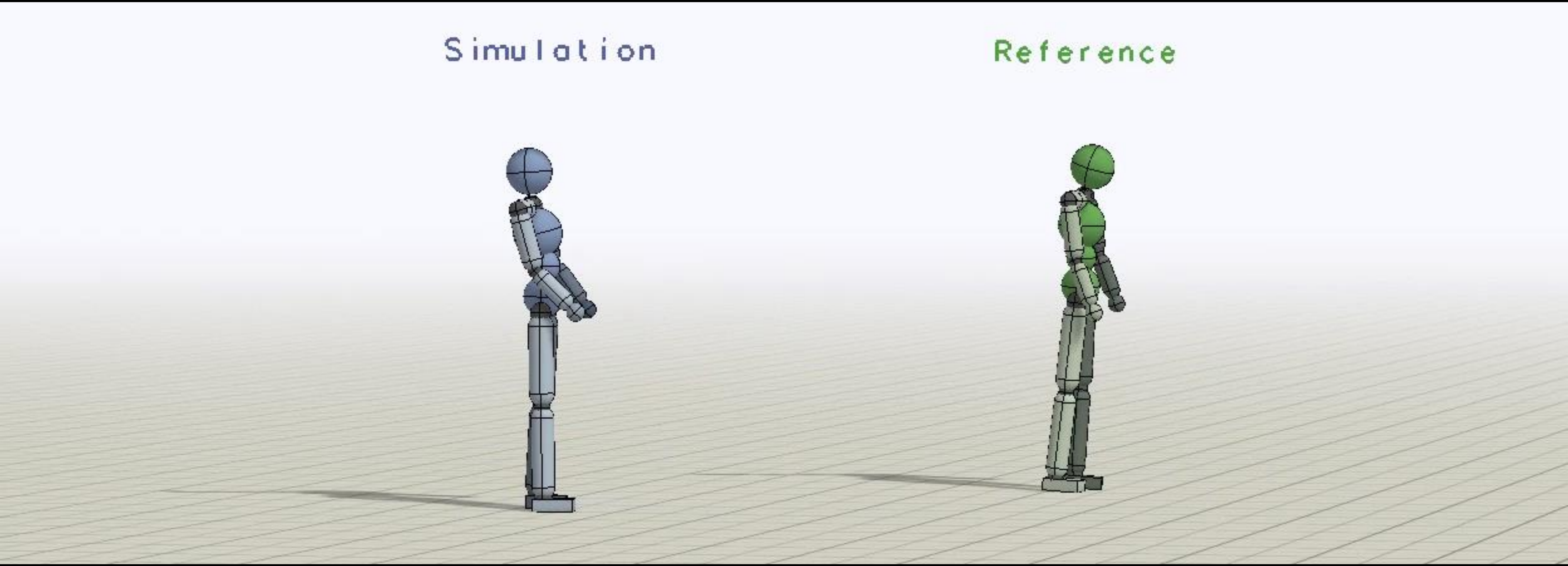
State Initialization



ref motion

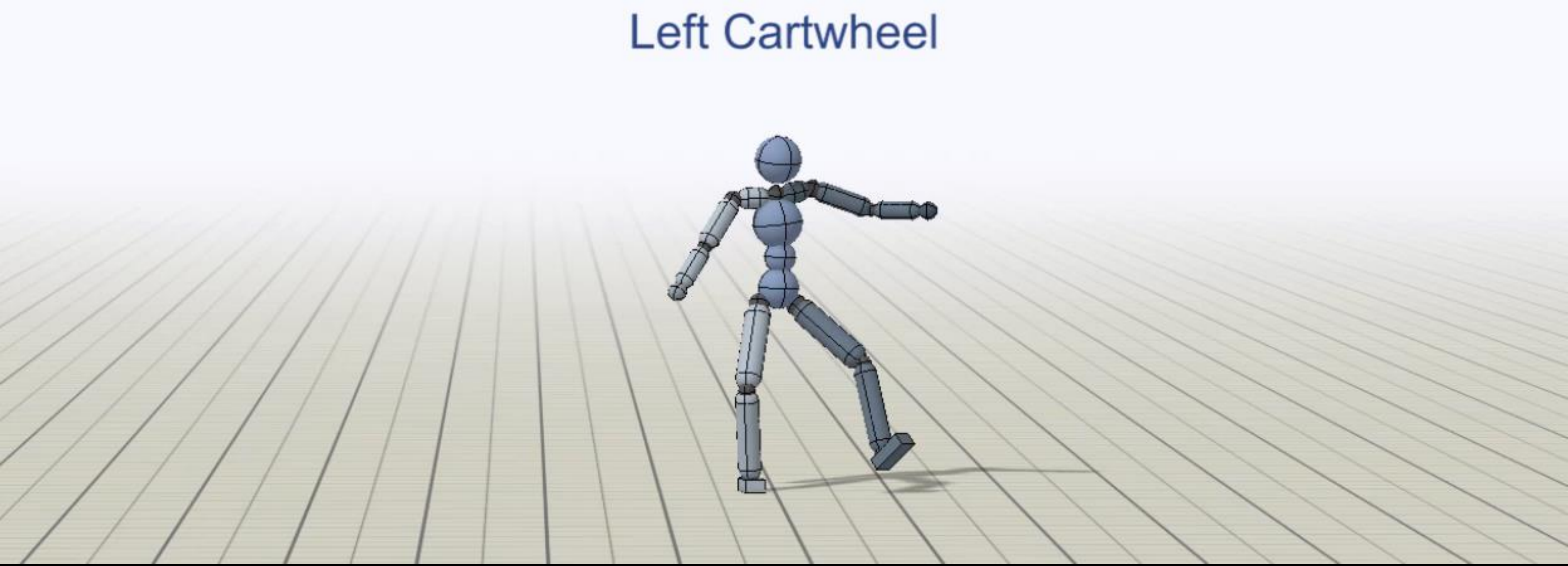
rollout

With Reference State Initialization



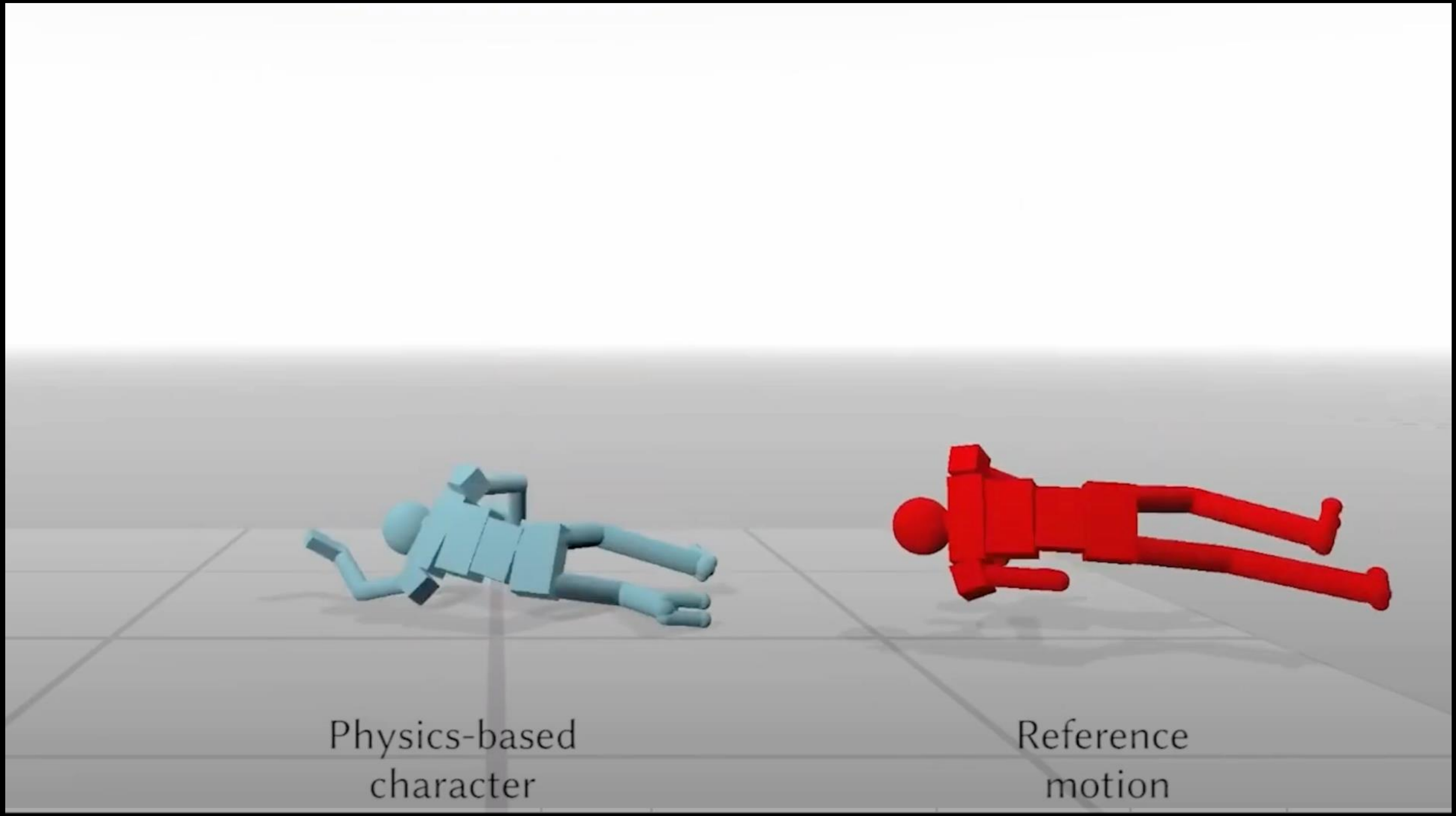
Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

RL gives robust motions



Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." ACM Transactions On Graphics (TOG), 2018

And mocap and physics can help learning good motions



Frezzato et al., Synthesizing Get-Up Motions for Physics-based Characters, SCA 2022

Even when the reference comes from a video

Video　　　　　　　　　　　　Simulation

Physics is a very good regularizer

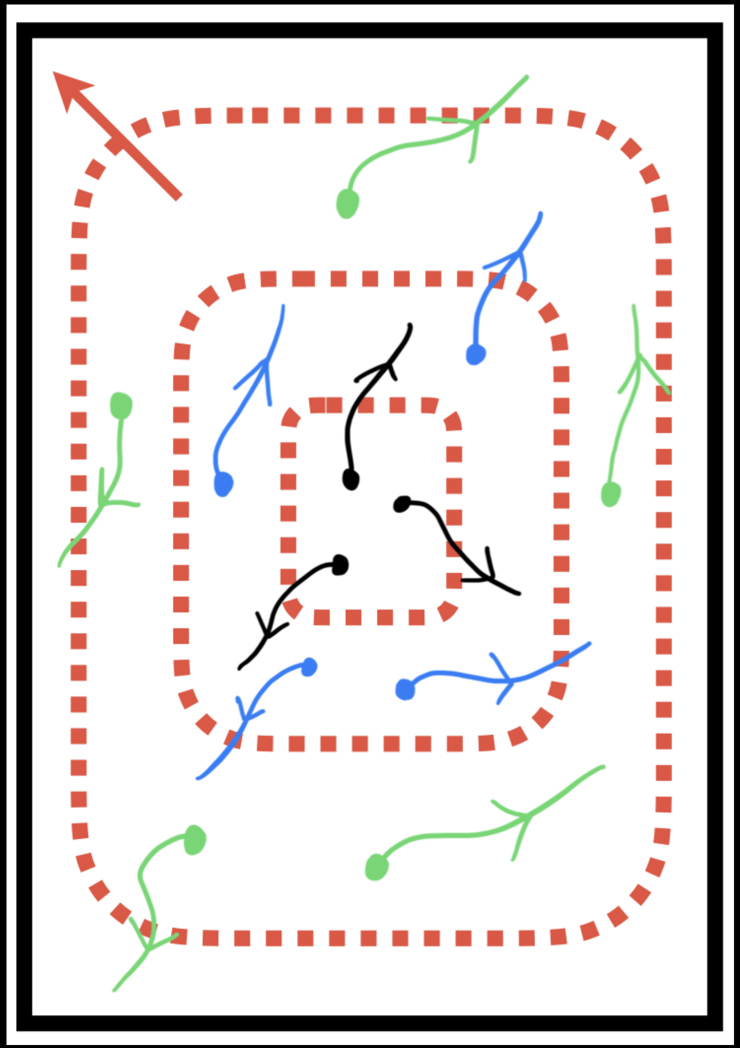
Peng, Xue Bin, et al. "Sfv: Reinforcement learning of physical skills from videos." ACM Transactions On Graphics 2018

What if we don’t have mocap?

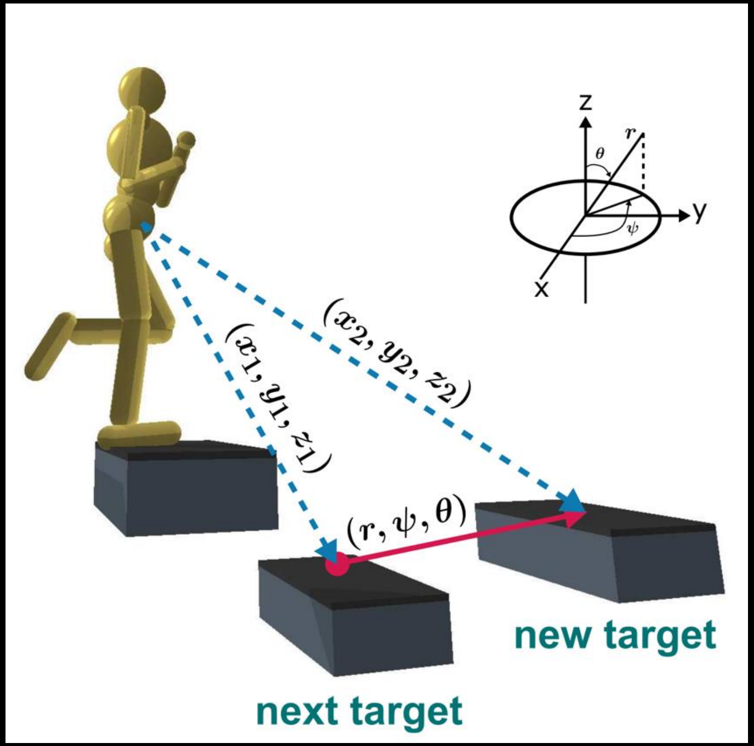
[STORROR Parkour]

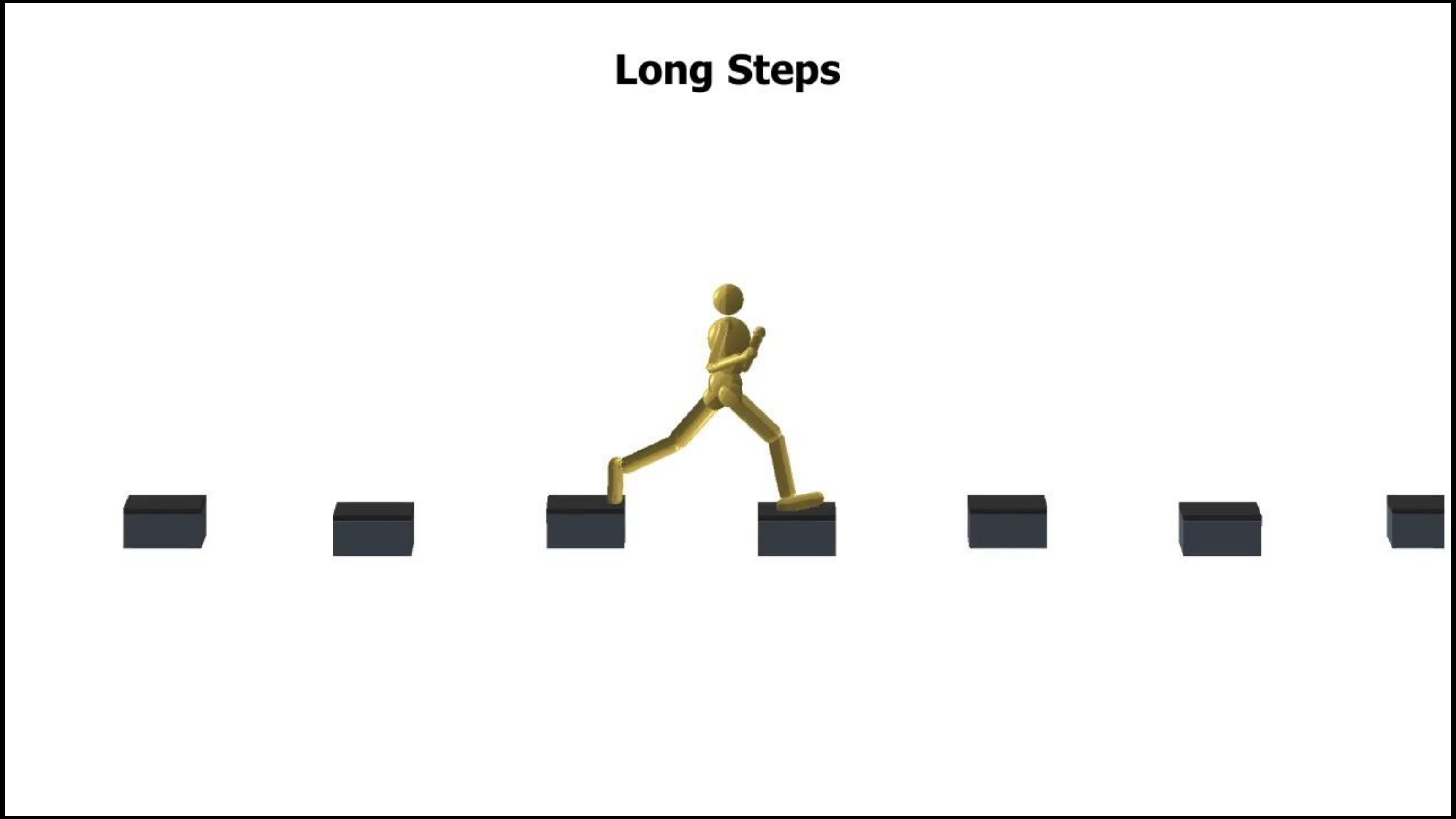
Curriculum Learning

- Take baby steps in the learning process
- You can start easy and progress as you learn the basic skills
- Don't start with the hardest task



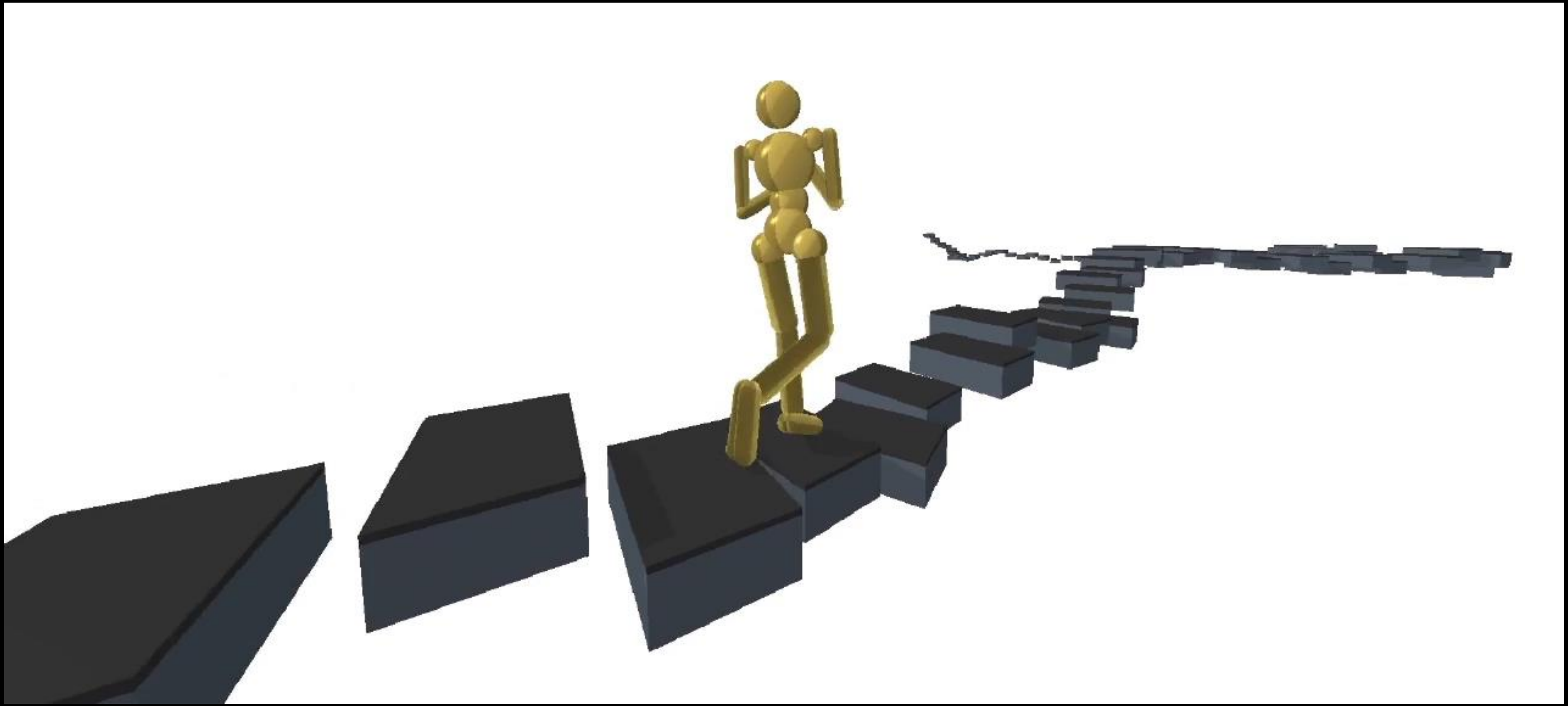
ALLSTEPS





Xie, Zhaoming, et al. "Allsteps: curriculum-driven learning of stepping stone skills." SCA 2020

ALLSTEPS



Xie, Zhaoming, et al. "Allsteps: curriculum-driven learning of stepping stone skills." SCA 2020

Brachiation

Source: https://youtu.be/U3JhwjNfx_g

Source: https://youtu.be/eV-gOL4t9Vk

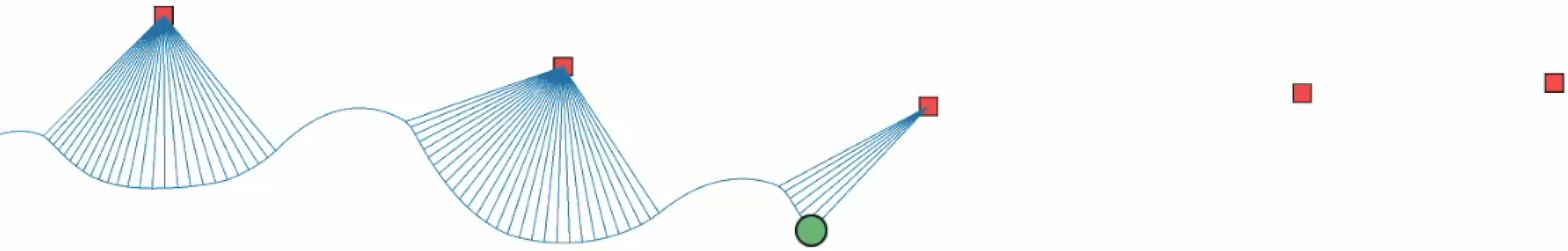
Reda, Daniele, Hung Yu Ling, and Michiel Van De Panne. "Learning to brachiate via simplified model imitation." SIGGRAPH 2022

Brachiation

Reda, Daniele, Hung Yu Ling, and Michiel Van De Panne. "Learning to brachiate via simplified model imitation." SIGGRAPH 2022

Simplified Model Imitation

Simplified Physical Model



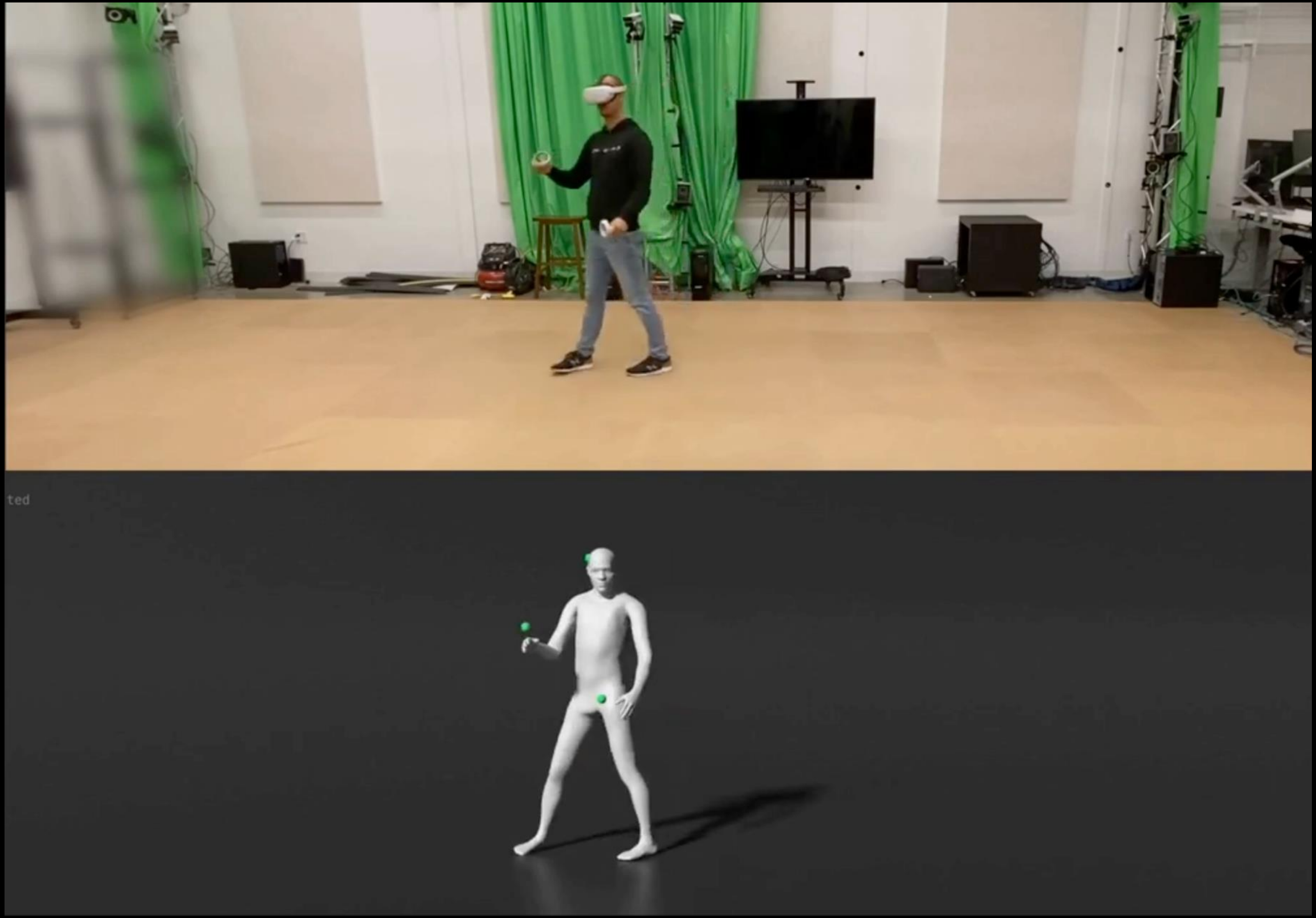
$$r_{simplified} = r_{task}$$

Reda, Daniele, Hung Yu Ling, and Michiel Van De Panne. "Learning to brachiate via simplified model imitation." SIGGRAPH 2022

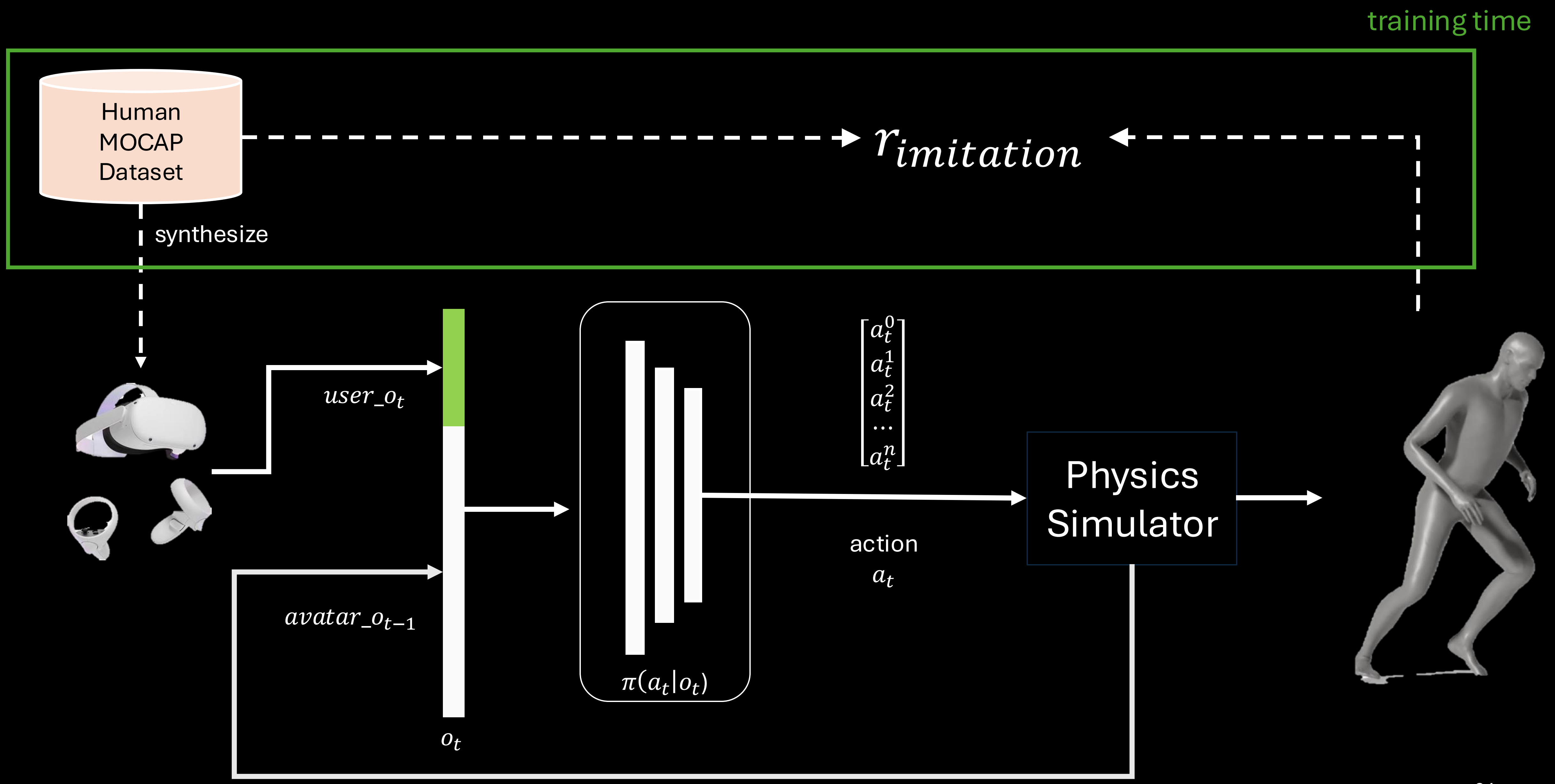
Real world applications

Virtual Reality

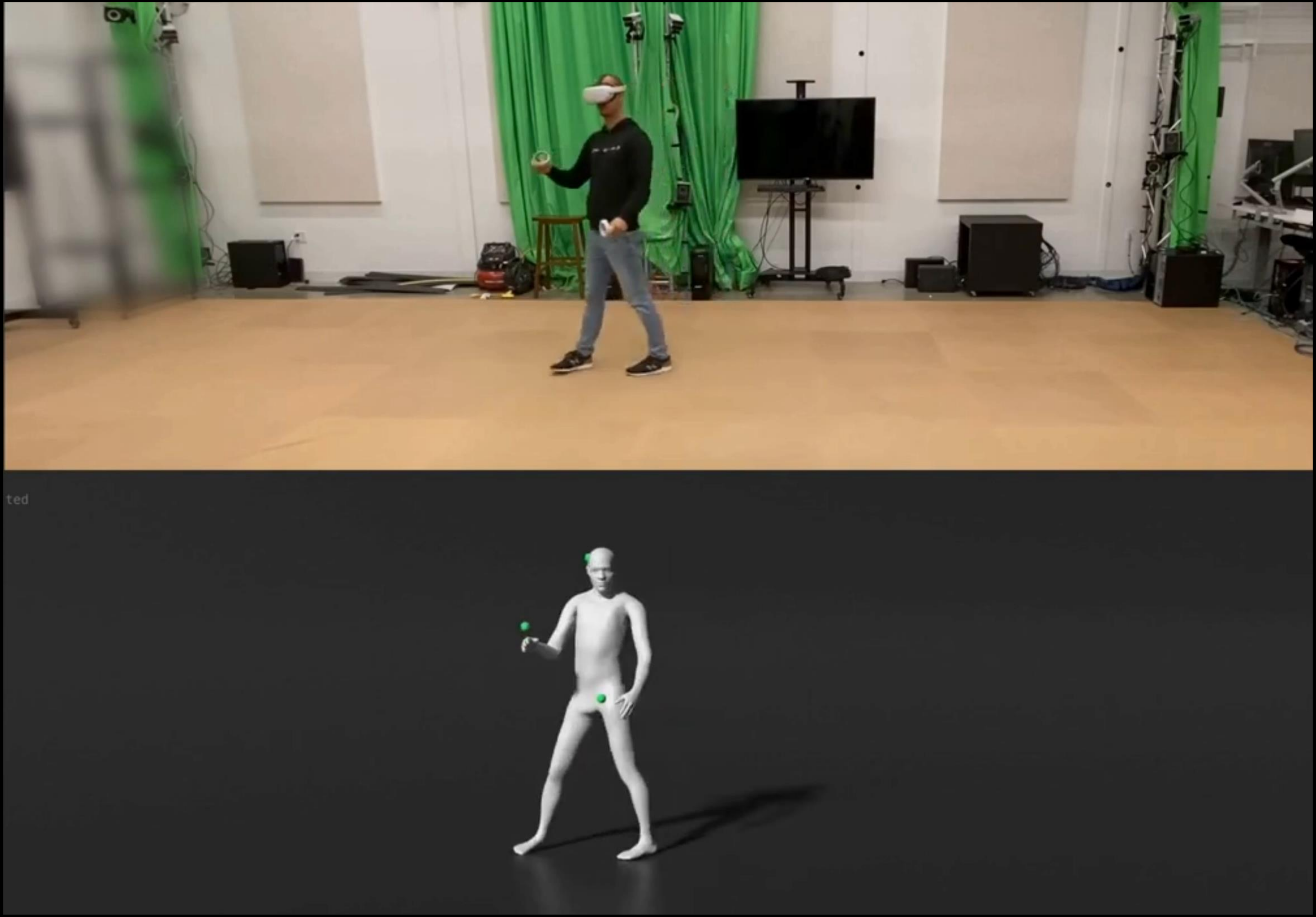
Physics can help predicting full body from VR



Winkler, Alexander, Jungdam Won, and Yuting Ye. "Questsim: Human motion tracking from sparse sensors with simulated avatars." SIGGRAPH Asia 2022



Winkler, Alexander, Jungdam Won, and Yuting Ye. "Questsim: Human motion tracking from sparse sensors with simulated avatars." SIGGRAPH Asia 2022

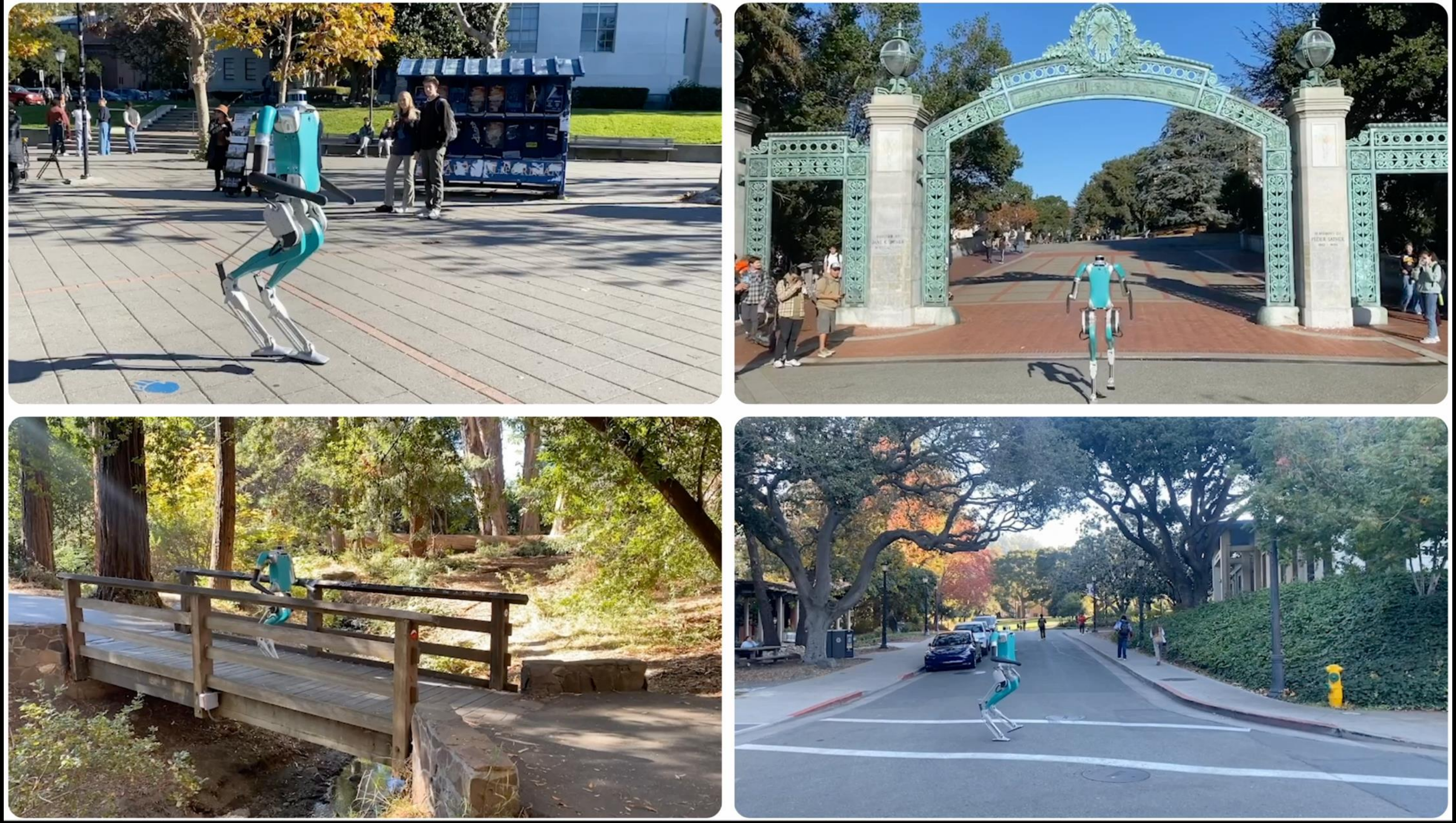


Winkler, Alexander, Jungdam Won, and Yuting Ye. "Questsim: Human motion tracking from sparse sensors with simulated avatars." SIGGRAPH Asia 2022

Reda, Daniele, et al. "Physics-based Motion Retargeting from Sparse Inputs." SCA 2023

Robotics

Beyond Simulation: Transfer to Real World



Radosavovic, Ilija, et al. "Real-world humanoid locomotion with reinforcement learning." Science Robotics 2024

Challenge: Sim2Real Gap

- When transferring simulated policies to the real world, the performances don't transfer
- Solutions:
  - Domain randomization
  - Dynamics randomization

[openAI, Rubik's Cube]

Gaming and VFX

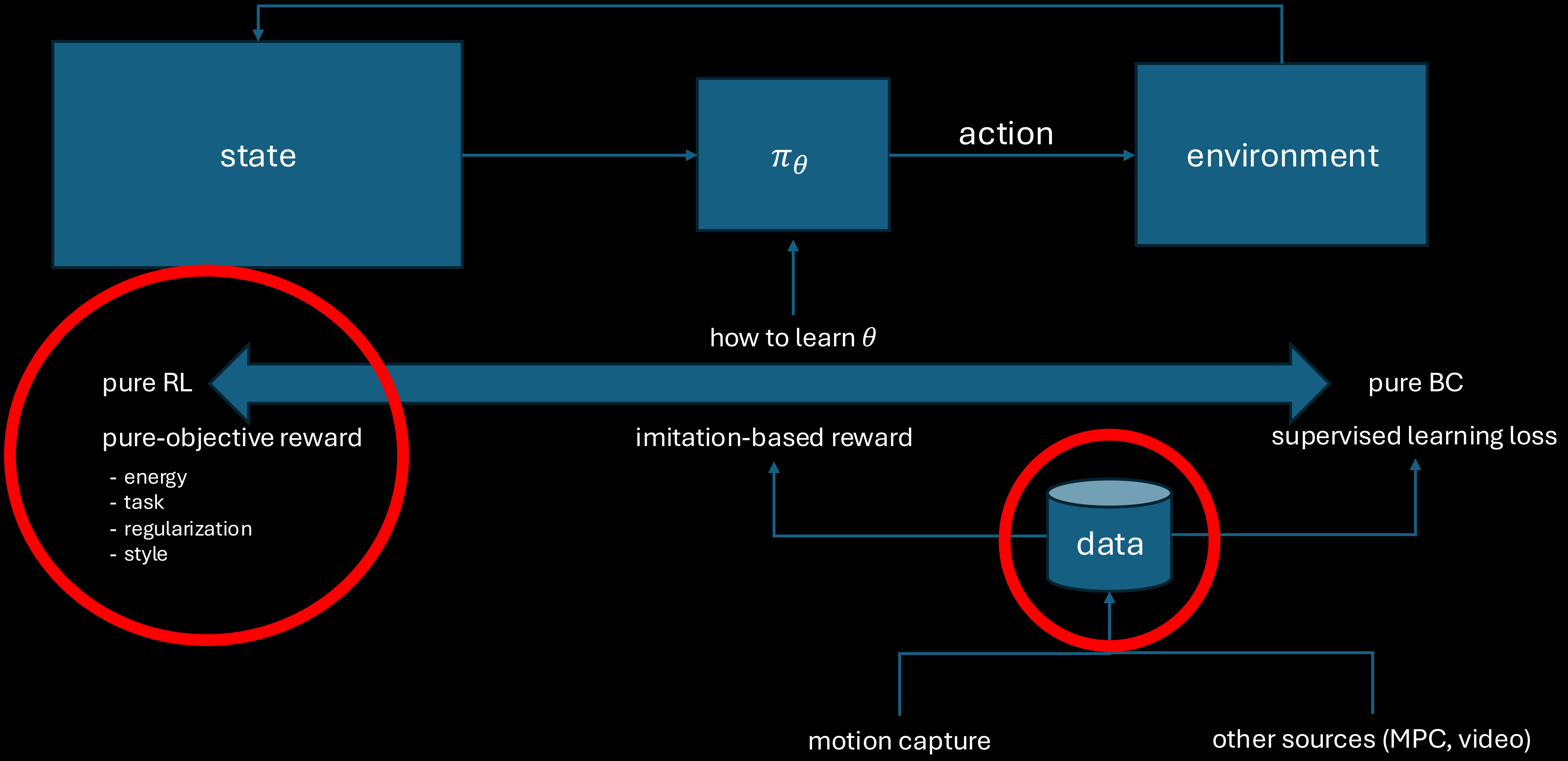
Nope…. not yet

Limitation: Controllability for virtual characters

[Assassin's Creed, Ubisoft]

To recap…

Recap: learning controllers with RL



- RL is powerful but hard to tune
- The exploration process needs some baby-sitting
- The outcome is very robust and general controllers
- Physics acts as a regularizer for hard tasks, and allows for realistic motions

What do we have now → Next

- Capabilities:
  - Robust controllers for different kind of terrains
  - Natural looking and general motions
  - Progress on sim2real
- Methodology:
  - Faster simulators
  - Better RL algorithms
  - Better reward function designs
  - Extracting good information from (a lot of) mocap data

- Capabilities:
  - Difficult and precise tasks:
    - Highly-dynamic
    - Very sparse rewards
    - High dimensional actions
    - High dimensional inputs (images)
  - Interaction with surrounding environments
- Methodology:
  - Using visual data instead of mocap and full states

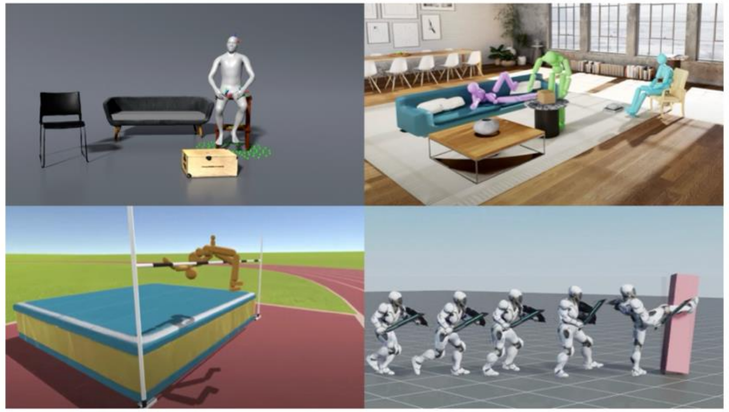
If you want to learn more:

rdednl.github.io/physics-based-controllers

Learning Physics-Based Character Controllers

A course presented at the SCA Summer School 2024

DANIELE REDA, University of British Columbia, Canada



Download slides here

Resources

Courses and Books

- **HuggingFace Reinforcement Learning Course**
- **Sutton & Barto, Reinforcement Learning: An Introduction**
- **Featherstone, Rigid Body Dynamics Algorithms**: Textbook on forward dynamics simulation

Surveys

- **Interactive Character Animation Using Simulated Physics: A State-of-the-Art Review**: A survey from 2012 summarizing old approaches (pre-deep learning) for learning physics-based controllers.
- **A Survey on Reinforcement Learning Methods in Character Animation**: A more recent survey giving a nice introduction on RL algorithms and a second section on how RL algorithms are used for learning character motions.
- **A Survey on Simulation Environments for Reinforcement Learning**: A 2021 survey summarizing different simulation environments used for RL pipelines.

Papers

- **Deepmimic**: This paper introduces the usage of motion capture data in reinforcement learning, as a way to formulate reward functions in the context of imitation. A lot of what this paper introduces is now common practice.
- **AMP**: This paper extends the concept of imitation-based reward function, with the usage of adversarial learning. It is an application of the paper GAIL to motion imitation.
- **ALLSTEPS**: This paper shows that through curriculum learning, a pure-objective reward approach is capable of learn robust controllers for difficult tasks.
- **Brachiation**: This paper shows that little signal is better than no signal. Using a simplified physical model, you can get imitation data for the center of mass for the full articulated character, which is enough to learn the full motion.