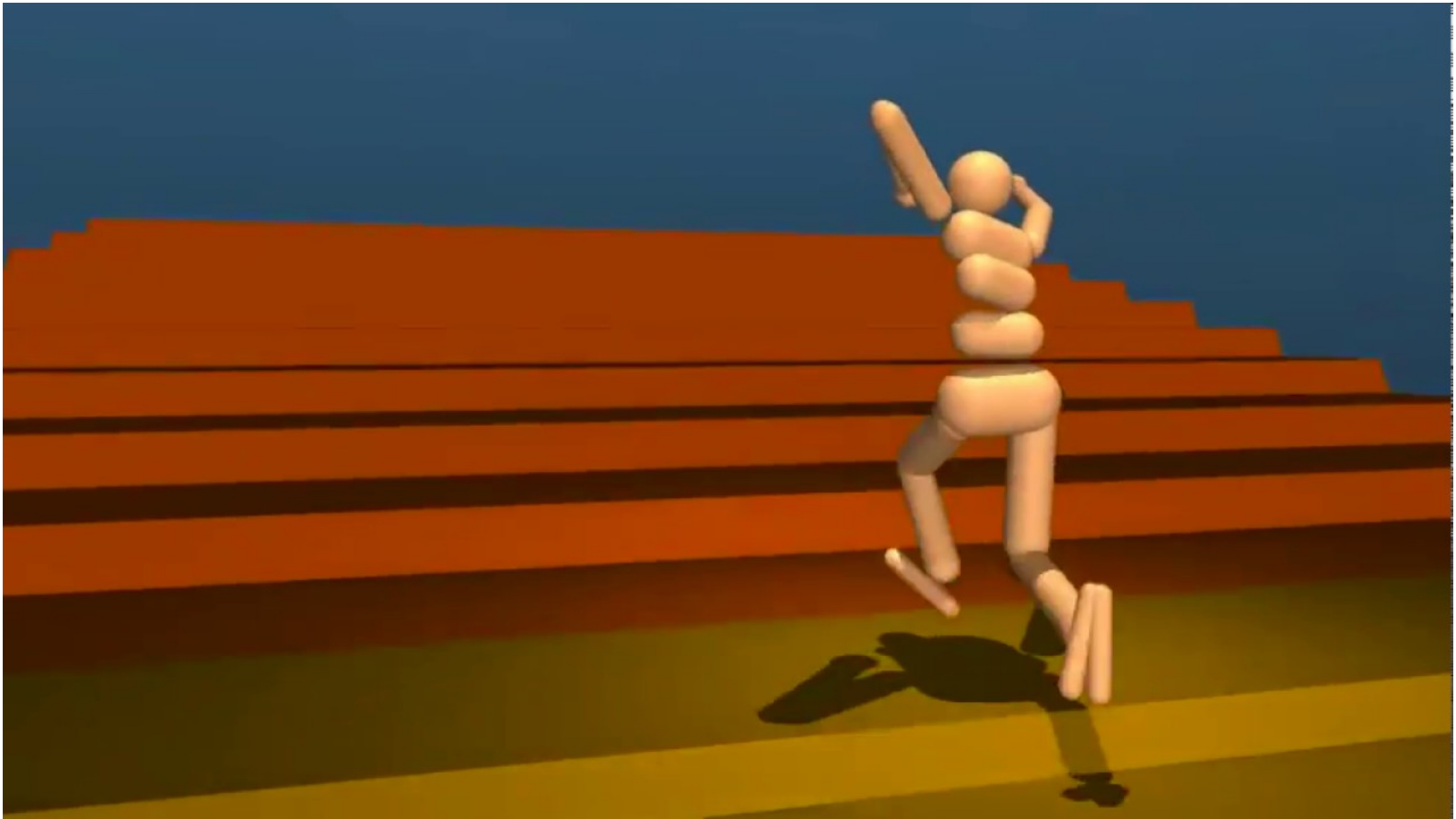
Master Reinforcement Learning 2022 Lecture 4: Policy Based Methods

Aske Plaat

Different Approaches

- Model-free
  - Value-based [2,3]
  - Policy-based [4]
- Model-based
  - Learned [5]
  - Perfect; Two-Agent [6]
- Multi-agent [7]
- Hierarchical Reinforcement Learning (Sub-goals) [8]
- Meta Learning [9]

Motivation



Overview

- Continuous Action Space
  - Robotics, Games
  - MuJoCo
- REINFORCE

- AC
  - Bootstrapping
  - Baseline
  - Trust Region
  - Entropy Exploration

- Environments
  - Locomotion
  - Visuo-motor Interaction

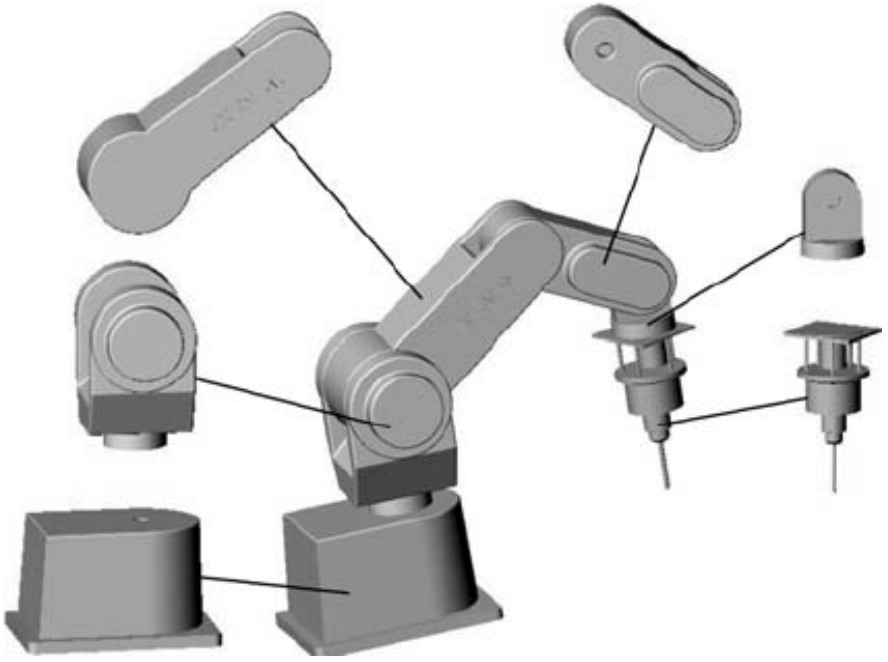
Indirect Discrete Actions

- Value based is 2-step (indirect policy)
- Value-based: getting to best action via value function
  $\pi(s) = \text{argmax}_a\ Q(s,a)$
- Certain games: chess, checkers, Go
  Pieces move to discrete locations

Direct Continuous Actions

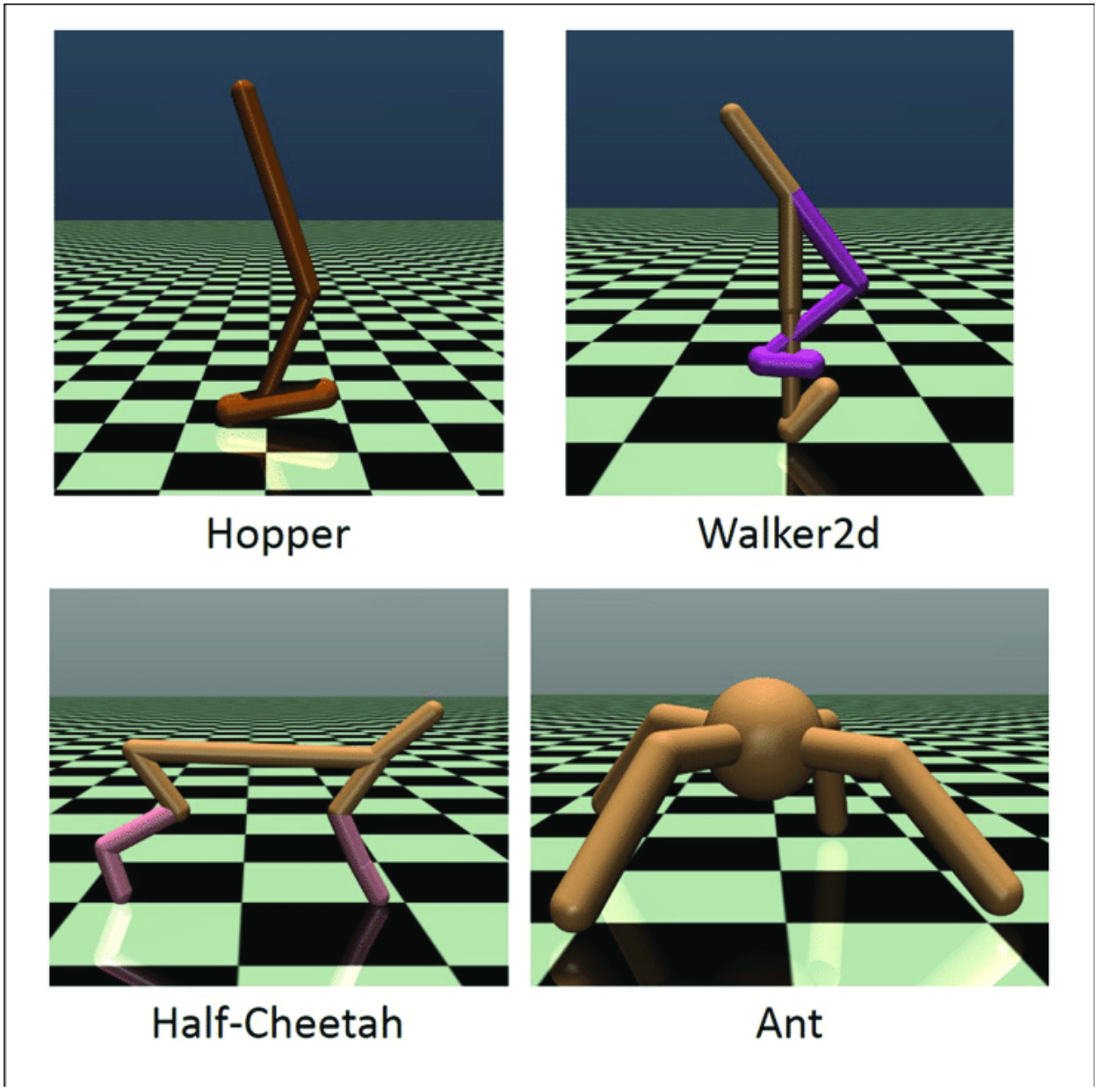
- When actions are continuous, argmax is difficult/unstable

- Certain games, robots, cars, etc
  Bet any amount in poker, move a joint any degrees



MuJoCo

- Software Physics simulator
- Prevents wear on real robots
- Model-free - millions of trials



Policy-based Algorithms

- (vanilla) REINFORCE
- Actor Critic
  - TD Bootstrap
  - Advantage: A3C
  - Trust Region: TRPO, **PPO**
  - Entropy: SAC
  - DQN-based: DDPG

REINFORCE

- take parameterized policy π_{θ_0}
- sample an episode τ with parameters θ_1
- if it is better, then push parameters in that direction 1
- if not, then push parameters the other way
- (aka: vanilla policy gradient)

Policy-gradient Theorem

$$Policy\ gradient : E_{\pi}[\nabla_{\theta}(log\pi(s,a,\theta))R(\tau)]$$

Policy function — Score function

$$Update\ rule : \Delta\theta = \alpha * \nabla_{\theta}(log\pi(s,a,\theta))R(\tau)$$

Change in parameters — Learning rate

REINFORCE

function REINFORCE

Initialise θ arbitrarily

for each episode $\{s_1, a_1, r_2, ..., s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**

for $t = 1$ to $T - 1$ **do**

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$$

end for

end for

return θ

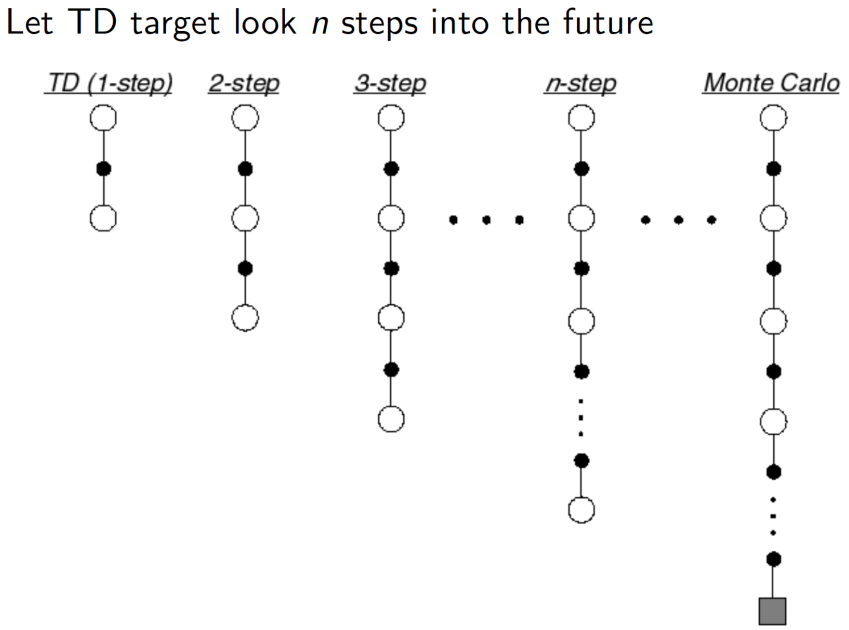
end function

Policy-gradient Theorem

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}[R(\tau)] \\
&= \nabla_\theta \int_\tau P(\tau|\theta) R(\tau) && \text{Expand expectation} \\
&= \int_\tau \nabla_\theta P(\tau|\theta) R(\tau) && \text{Bring gradient under integral} \\
&= \int_\tau P(\tau|\theta) \nabla_\theta \log P(\tau|\theta) R(\tau) && \text{Log-derivative trick} \\
&= \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}[\nabla_\theta \log P(\tau|\theta) R(\tau)] && \text{Return to expectation form} \\
\therefore \nabla_\theta J(\pi_\theta) &= \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau)\right] && \text{Expression for grad-log-prob}
\end{aligned}
$$

Adv./Disadv.

- Continuous policies
- Stochastic policies
- Direct policy (no in-between value step)
- Full Episode policy sample:
  - Low bias
  - High Variance
    - (Slow convergence & Bad performanc
  - [Single stepping Value-based is high-bia



Can we Combine the advantages of value-based with policy-based?

Actor Critic

- Actor = policy
- Critic = value
- 2 ideas to reduce variance
  - temporal difference bootstrapping
  - baseline subtraction

Bootstrapping

- TD computes values step-wise (low variance)

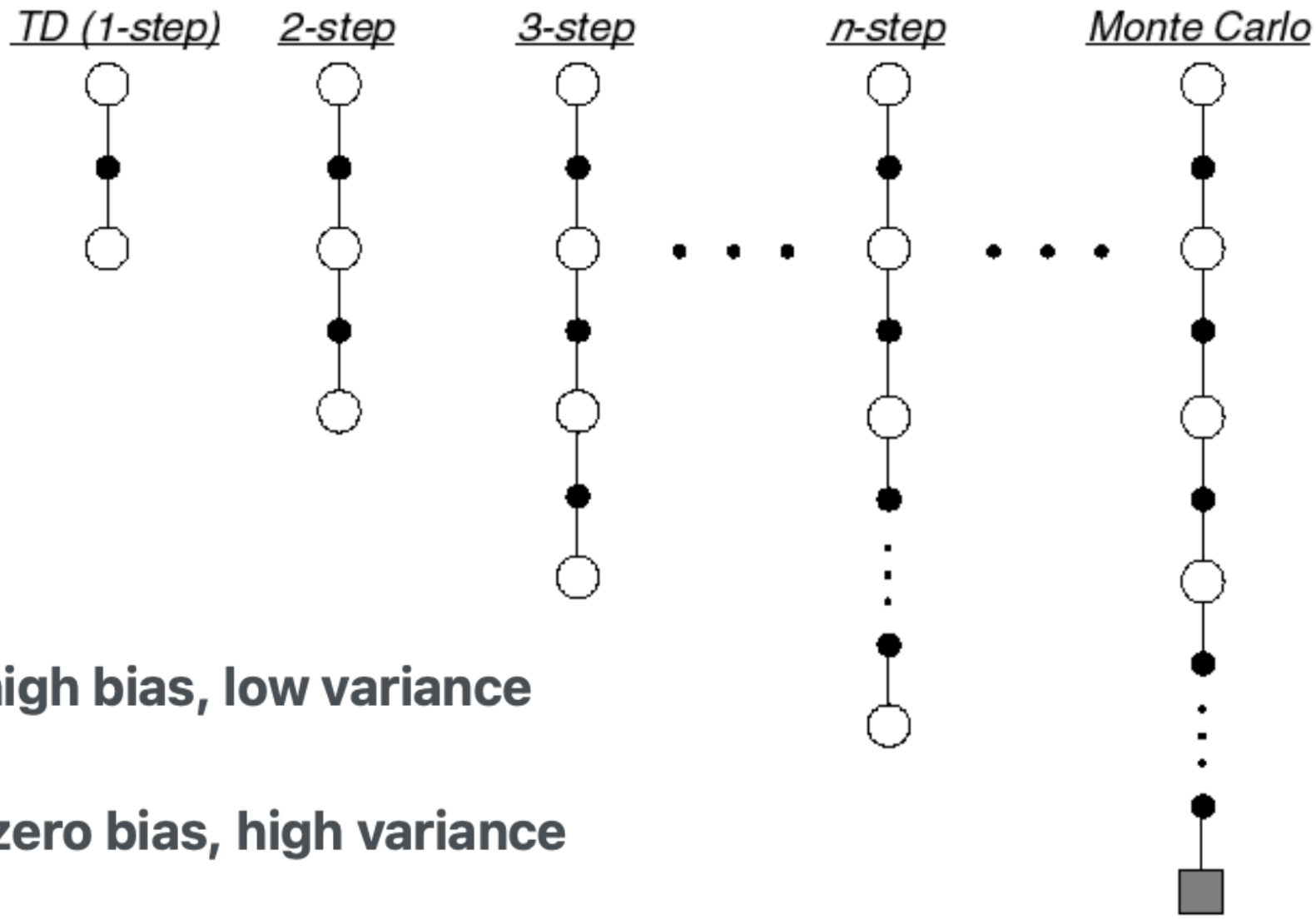
$$V(s) \leftarrow V(s) + \alpha[R' + \gamma V(s') - V(s)]$$

- We can also use n-step values

- TD bootstrapping reduces variance of Monte Carlo at the cost of more bias

Bias/Variance

Let TD target look *n* steps into the future



TD – high bias, low variance

MC – zero bias, high variance

Baseline Subtraction

- Rewards are often skewed, such as, all positive, leading to high variance.
  Centering around zero would reduce variance

- When a baseline function is added to a function, the Expectation does not change, and the variance is reduced

- The Value function is such a function for the Q function

- $A(s,a) = Q(s,a) - V(s)$

Advantage Variants

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau_0 \sim p_\theta(\tau_0)} \Big[\sum_{t=0}^{n} \Psi_t \nabla_\theta \log \pi_\theta(a_t|s_t) \Big]$$

- Targets:

$$\Psi_t = \hat{Q}_{MC}(s_t, a_t) = \sum_{i=t}^{\infty} \gamma^i \cdot r_i \qquad \text{Monte Carlo target}$$

$$\Psi_t = \hat{Q}_n(s_t, a_t) \quad = \sum_{i=t}^{n-1} \gamma^i \cdot r_i + \gamma^n V_\theta(s_n) \qquad \text{bootstrap } (n\text{-step target})$$

$$\Psi_t = \hat{A}_{MC}(s_t, a_t) = \sum_{i=t}^{\infty} \gamma^i \cdot r_i - V_\theta(s_t) \qquad \text{baseline subtraction}$$

$$\Psi_t = \hat{A}_n(s_t, a_t) \quad = \sum_{i=t}^{n-1} \gamma^i \cdot r_i + \gamma^n V_\theta(s_n) - V_\theta(s_t) \qquad \text{baseline + bootstrap}$$

$$\Psi_t = Q_\phi(s_t, a_t) \qquad \text{Q-value approximation}$$

Trust Regions

Trust Region

- Vanilla REINFORCE is high variance (AC helps)
- parameters θ are pushed wildly
- Normally, we would mitigate variance by reducing step size of function input (θ)
- Trust Region is an approach using function output (π) to mitigate step size, using KL divergence (TRPO), or simple clipping (PPO)

Trust Region

- TRPO compares old and new policy (output)

$$L(\theta) = \mathbb{E}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot A_t \right]$$

- limiting KL divergence (measure of distance of distributions)

$$\mathbb{E}_t \left[\text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)) \right] \leq \delta$$

- A small divergence allows larger step size, a large divergence contracts

- PPO just clips L to a small range $[1-\epsilon, 1+\epsilon] \cdot A_t$

Trust Region

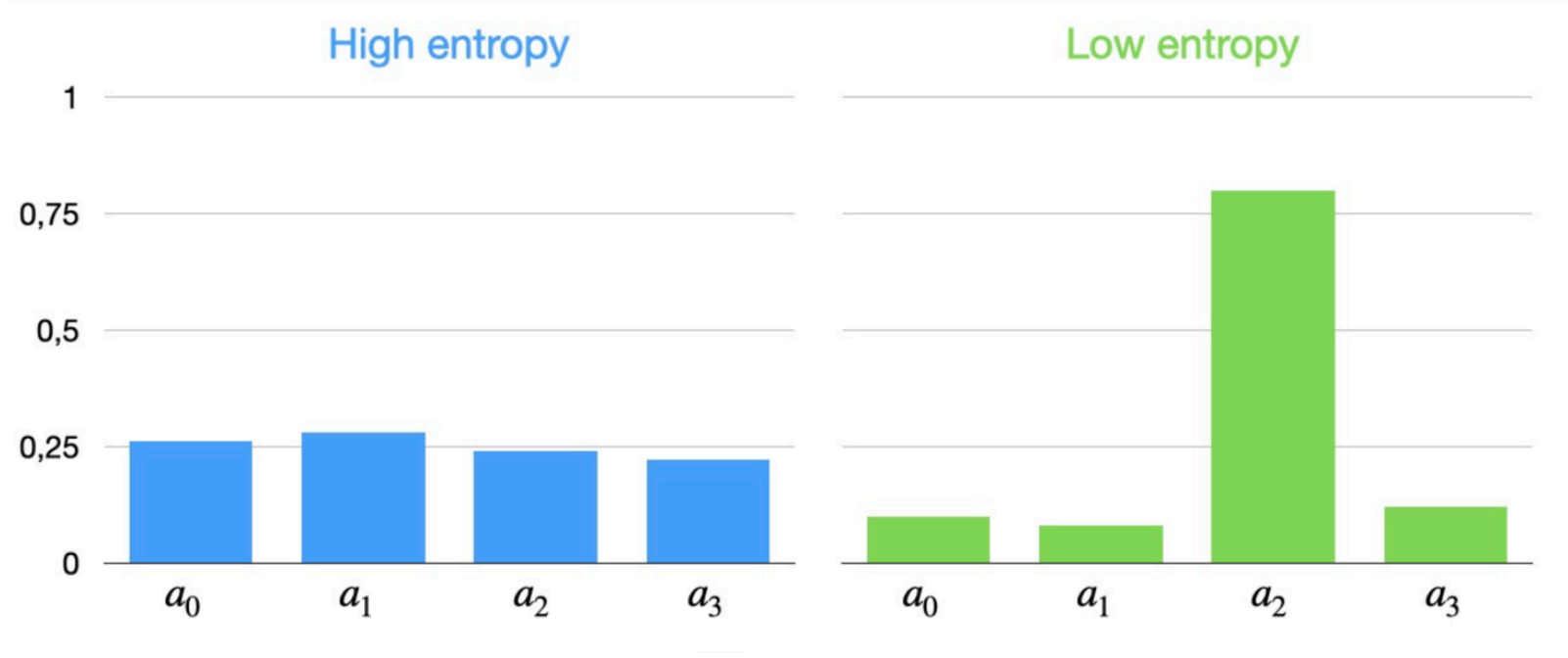
- Trust Region Policy Optimization TRPO
- Proximal Policy Optimization PPO
- TRPO and PPO keep new policy closer to old policy
- Reduce variance

Variance
Entropy
Exploration

Entropy/Exploration

- Too little exploration leads to local optima.
- Entropy is randomness. High entropy policies favor exploration
- Soft Actor Critic adds entropy *H* to the loss function

$$\theta_{t+1} = \theta_t + R \cdot \nabla_\theta \log \pi_\theta(a_t|s_t) + \eta \nabla_\theta H[\pi_\theta(a|s)]$$

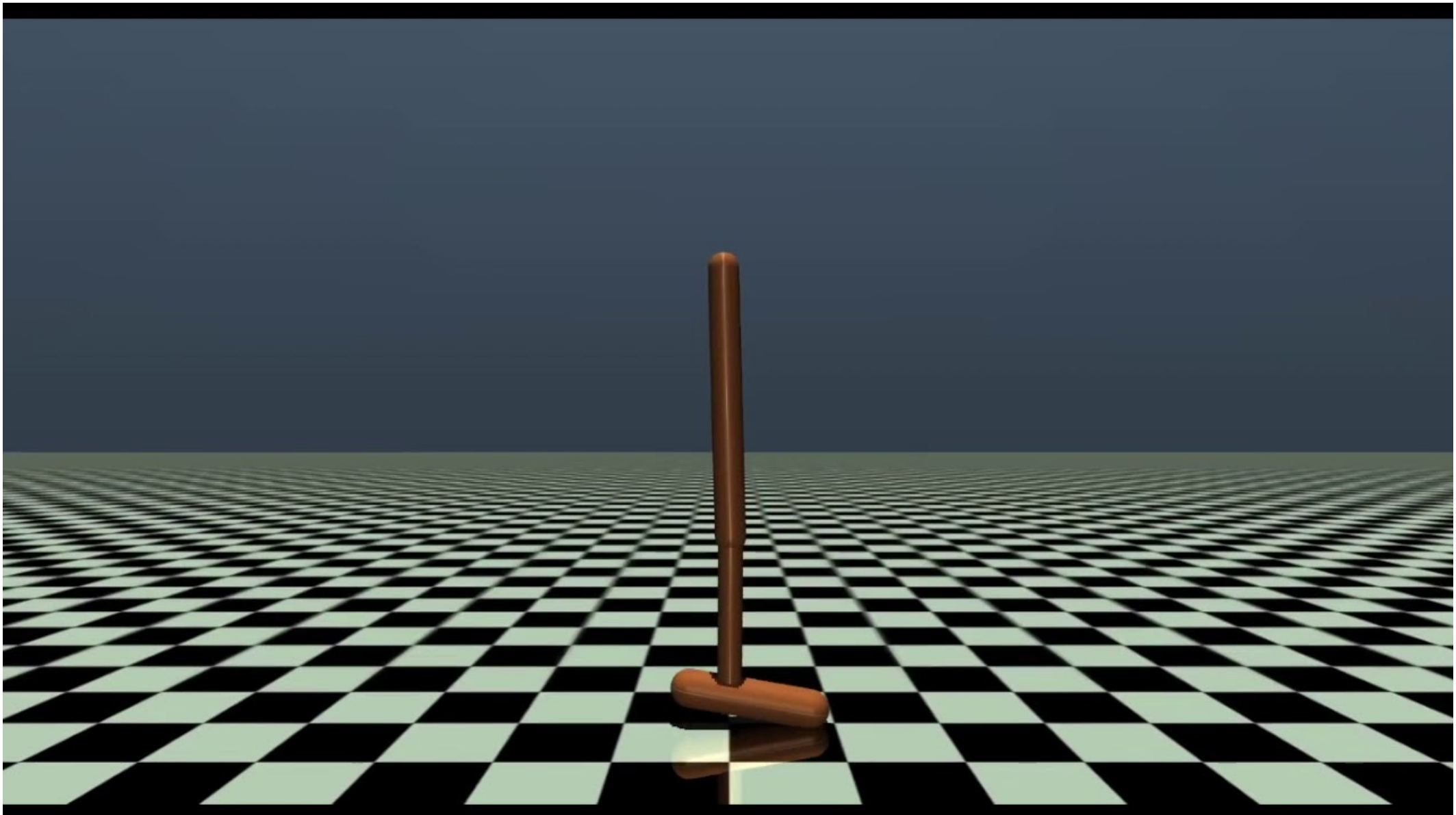


Policy-based Algorithms

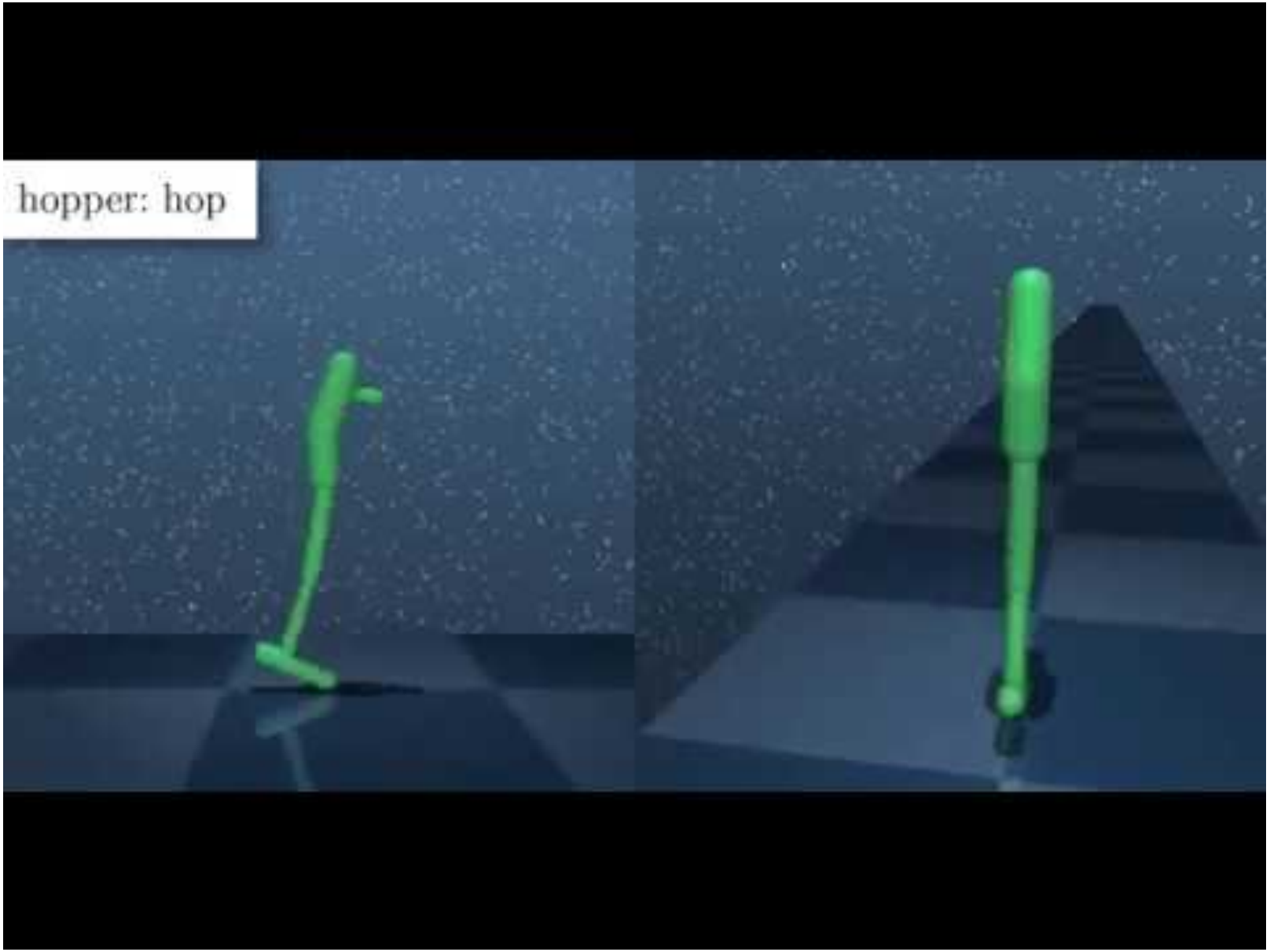
- (vanilla) REINFORCE - directly improving continuous policy, but high variance
- Actor Critic
  - TD Bootstrap
  - Advantage: A3C
- Trust Region: TRPO, **PPO**
- Entropy: SAC
- DQN-based: DDPG (read book)

Results

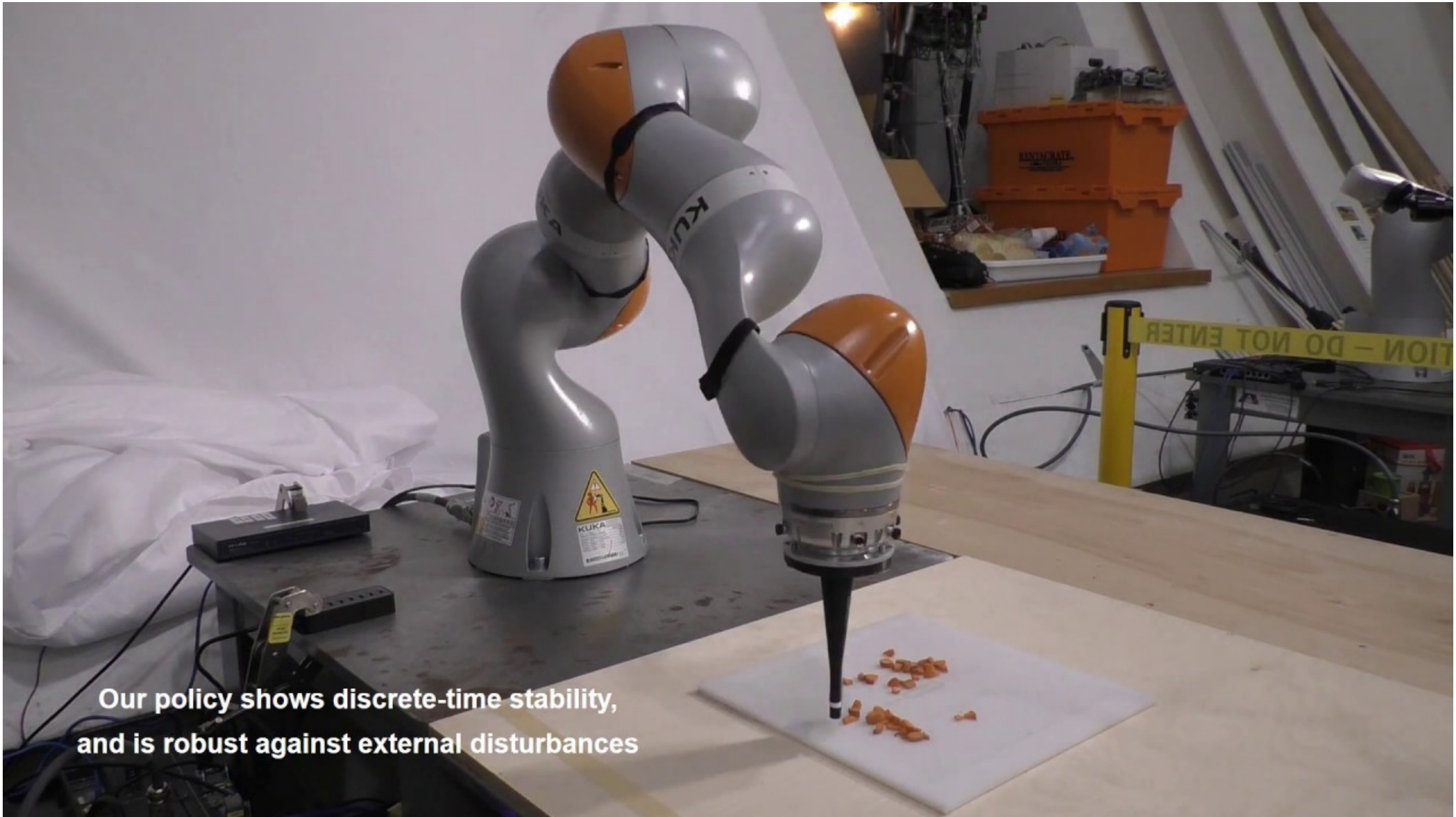
Locomotion



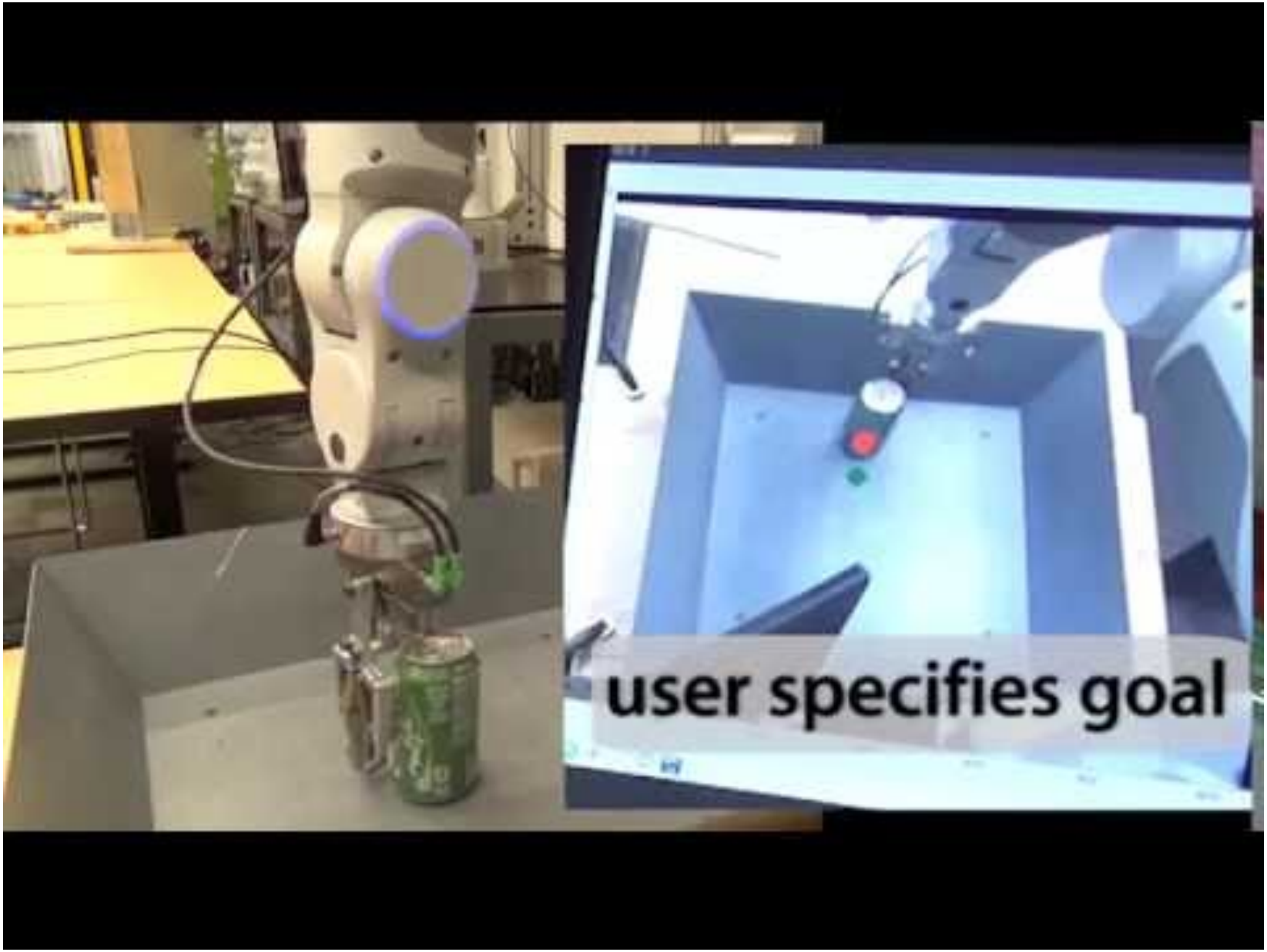
DeepMind Control Suite



Visuo-motor


Our policy shows discrete-time stability,
and is robust against external disturbances

Visuo-motor



Questions?