Geometry of Neural Reinforcement Learning in Continuous State and Action Spaces

ICLR 2025 ORAL

Saket Tiwari (on job market for PostDoc)

Omer Gottesman

George Konidaris

April, 2025

Data Geometry and Manifold Hypothesis

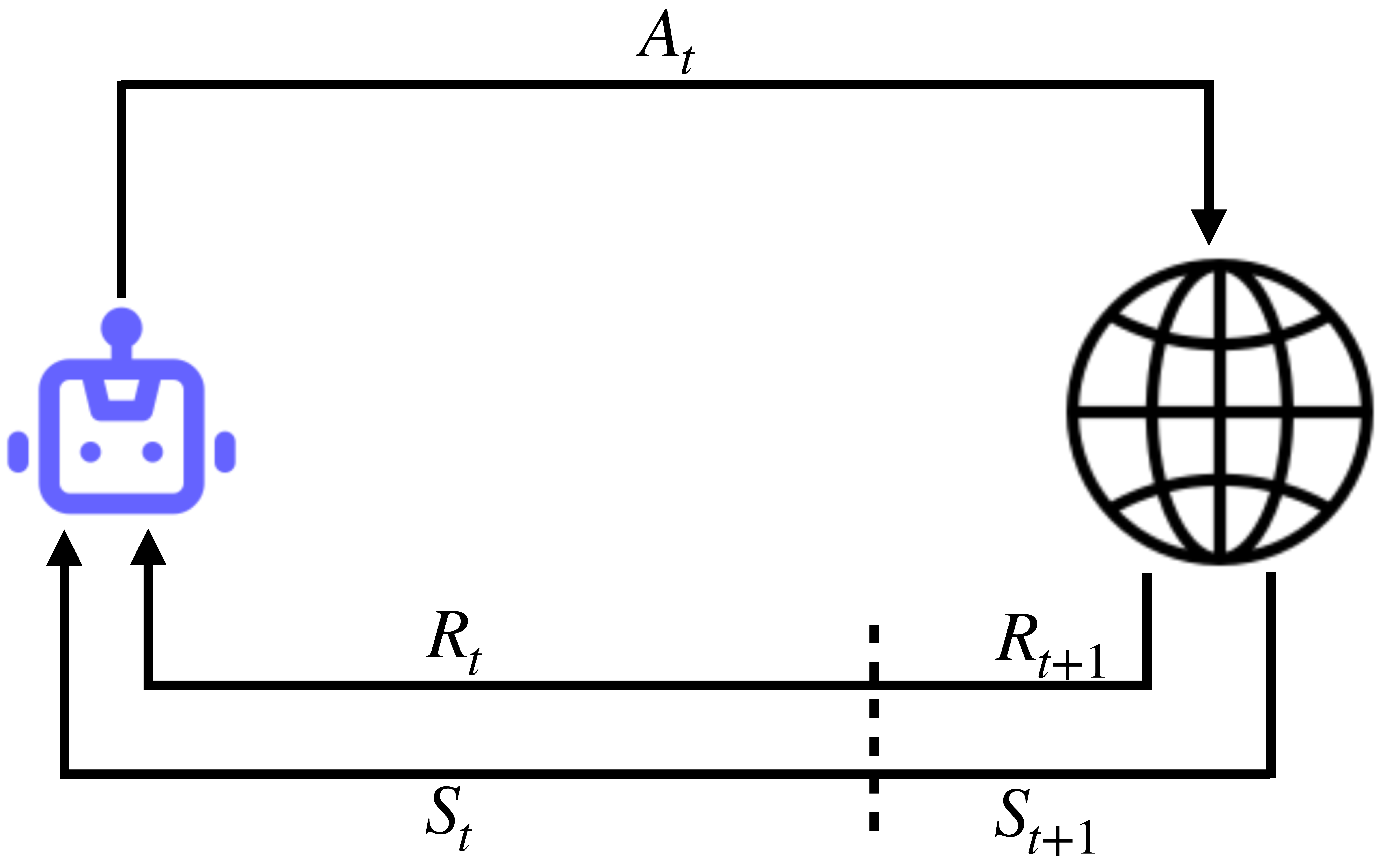
Tenenbaum, et al 2000

The Manifold Hypothesis

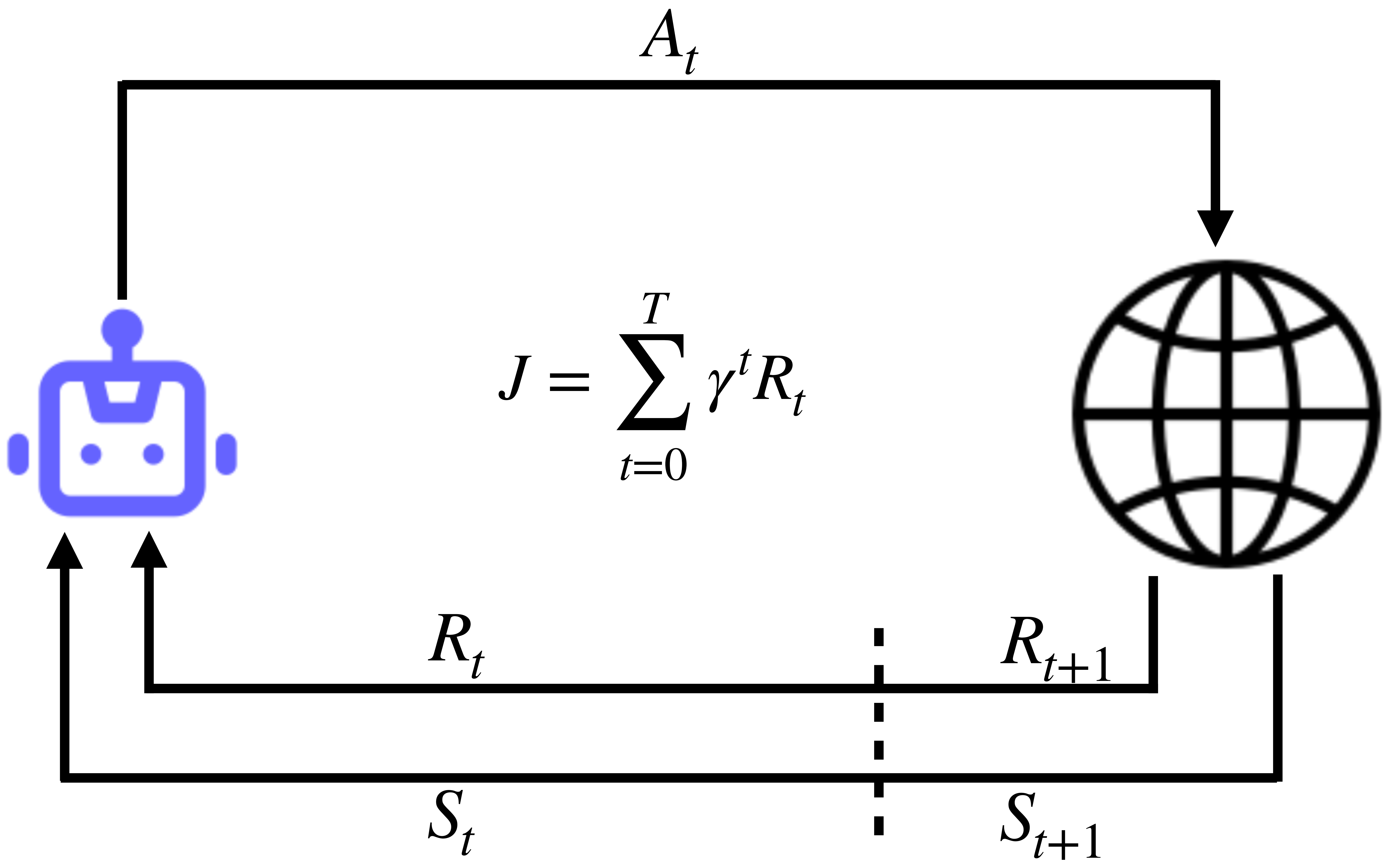
Natural data lies close to lower-dimensional manifolds in its embedding space.

2D Manifolds embedded in 3D spaces: Torus, Sphere and Cylinder

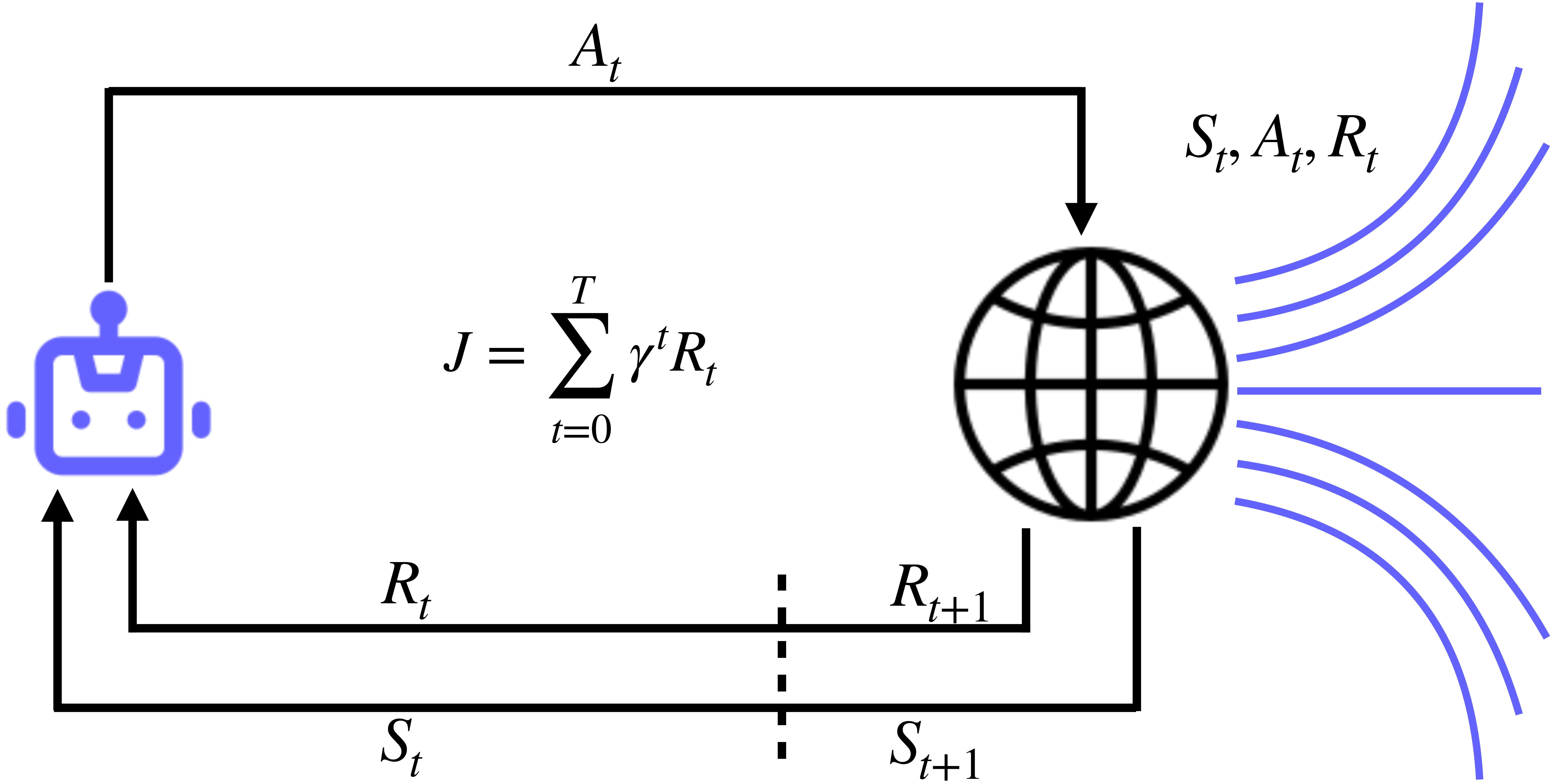
Reinforcement Learning



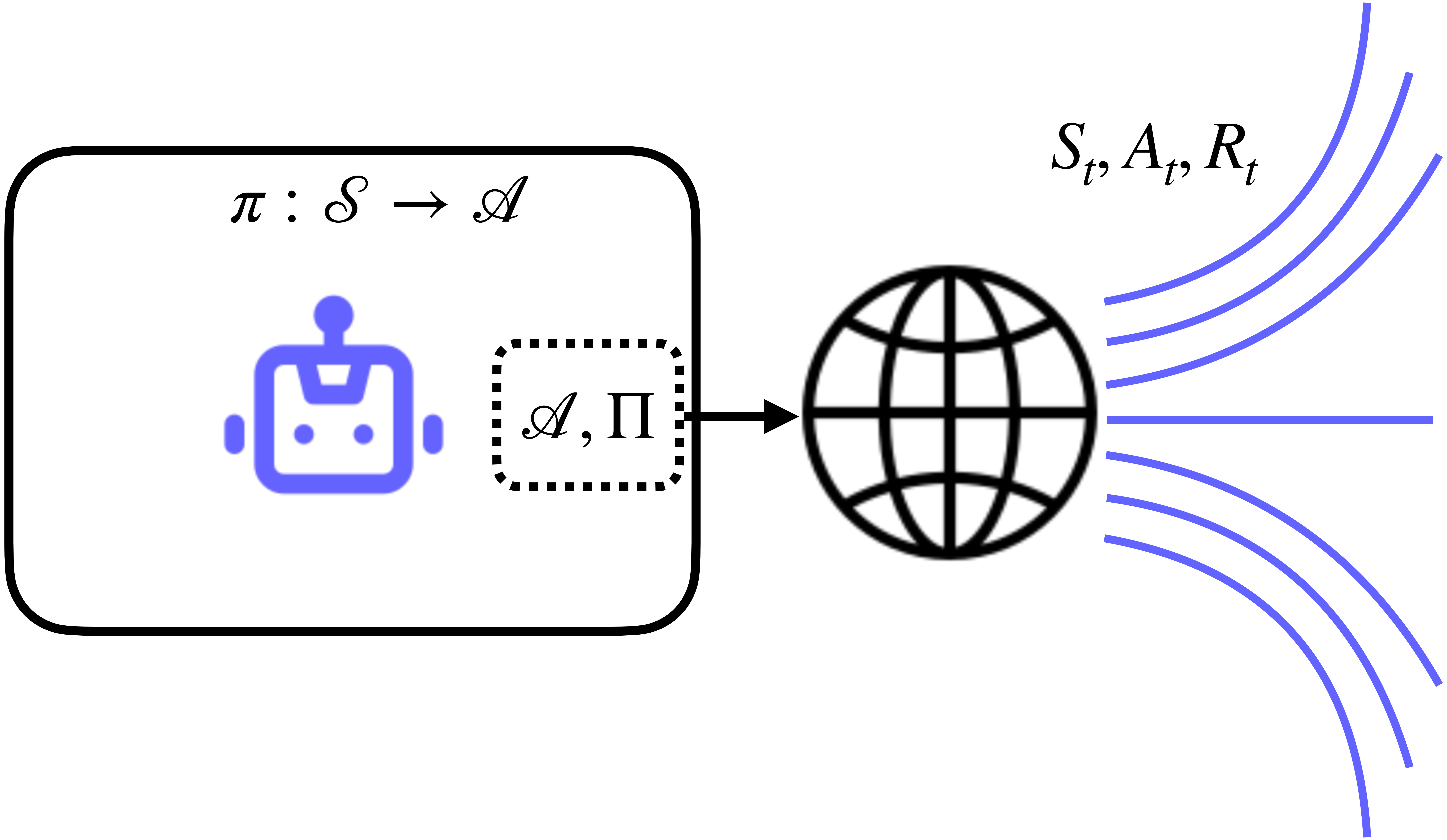
Reinforcement Learning



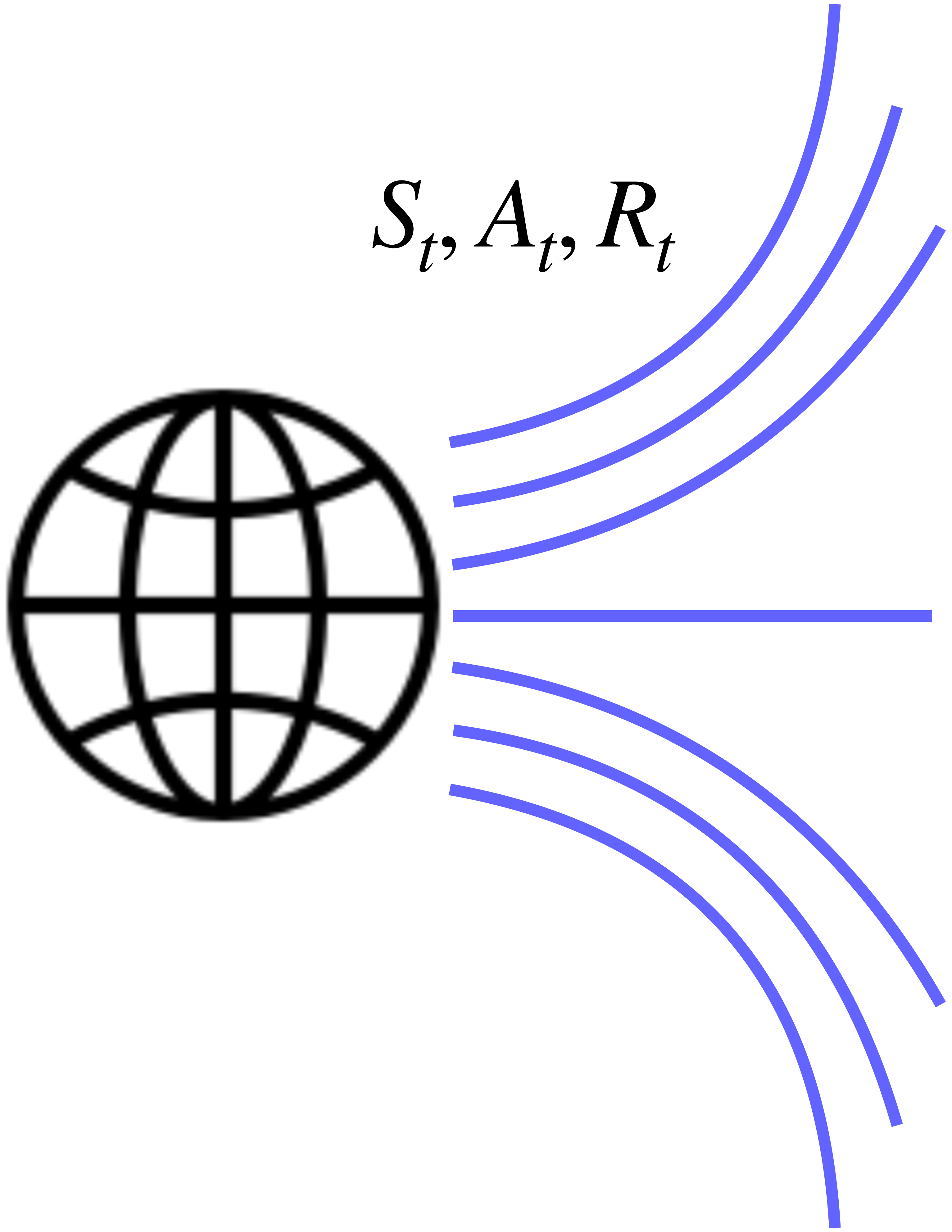
Reinforcement Learning

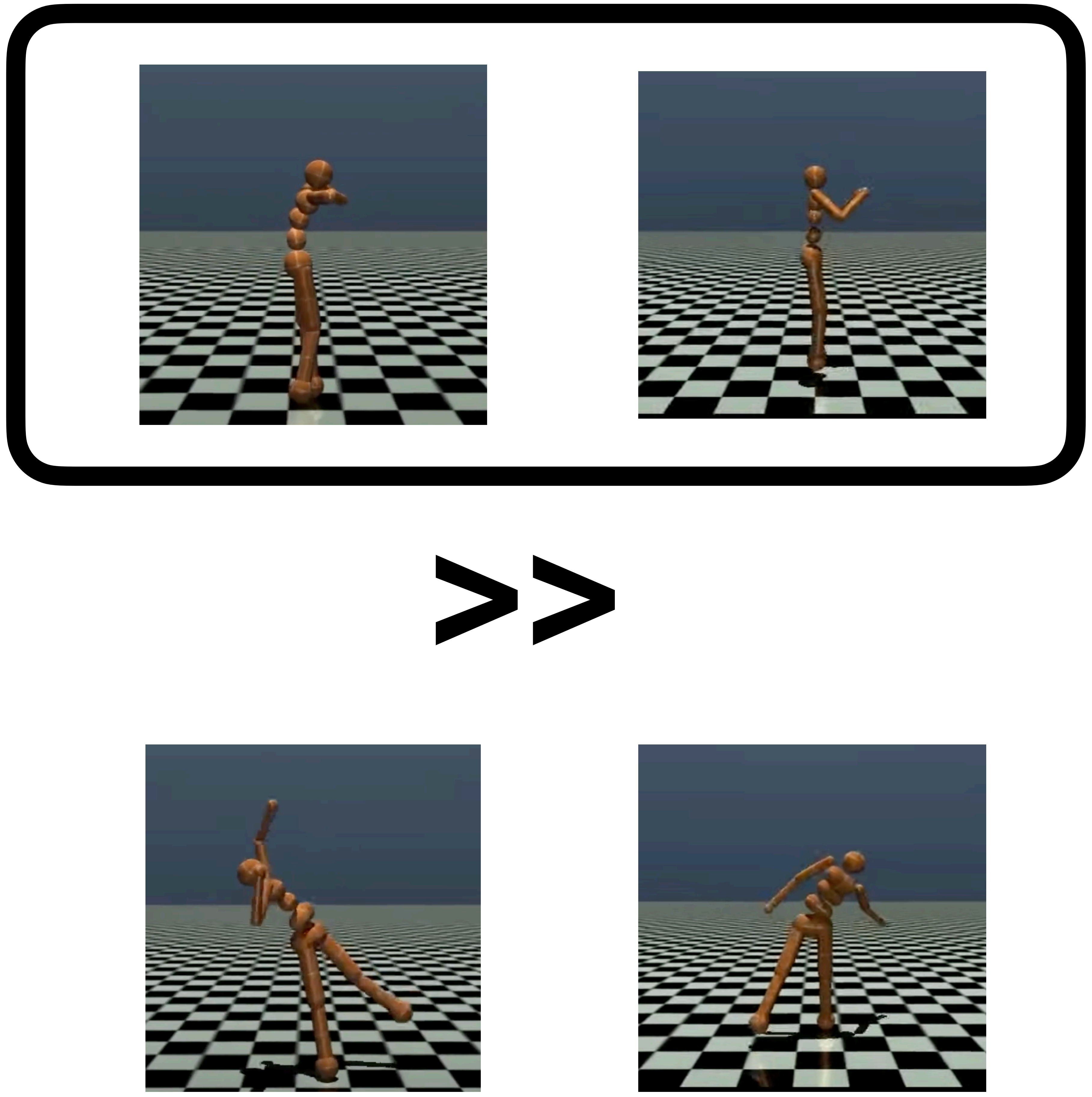


Reinforcement Learning



Reinforcement Learning





Reinforcement Learning: Policy Gradient Ascent

$$J(\theta) = \sum_{t=0}^{T} \gamma^t R_t$$

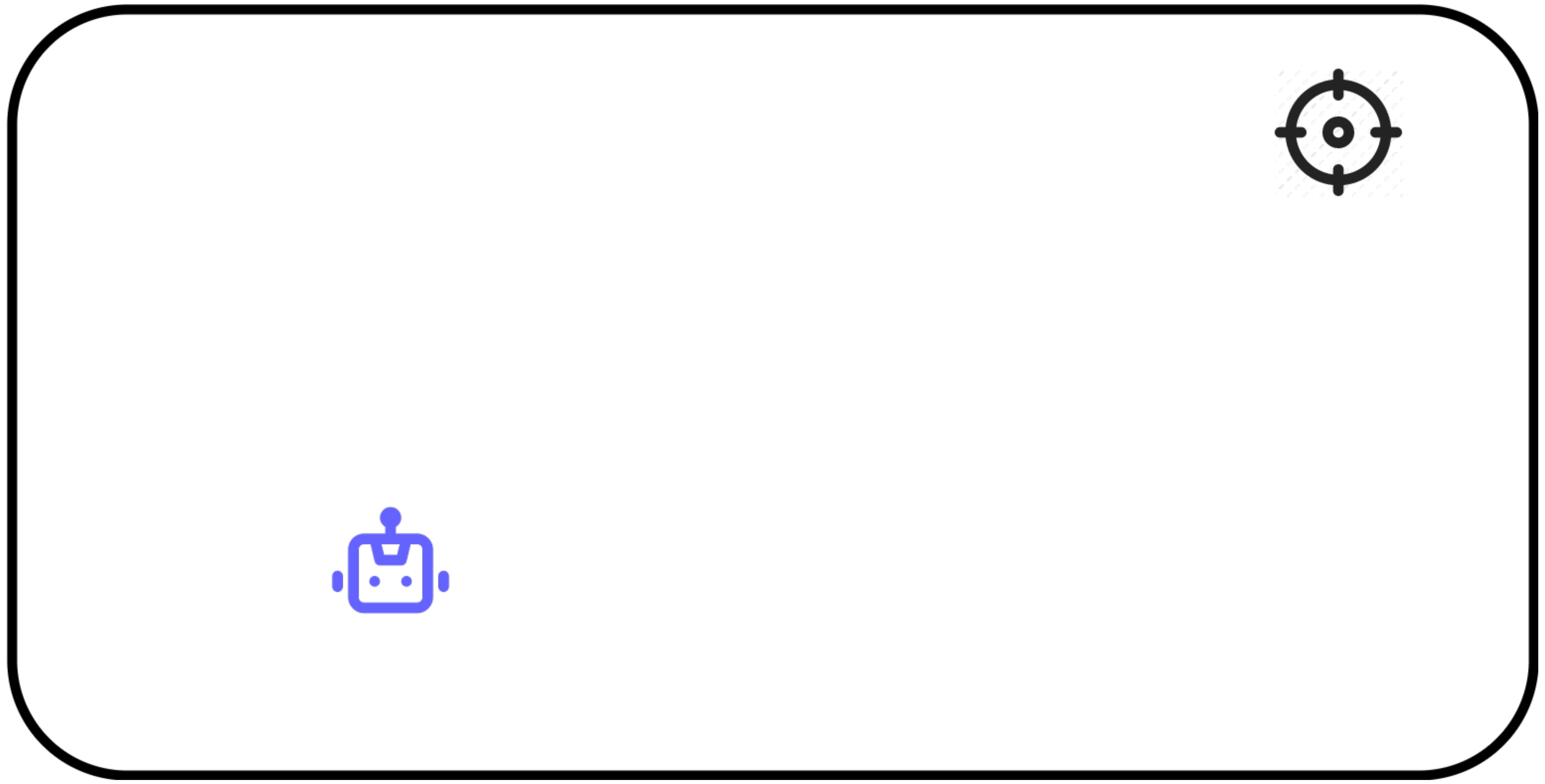
$$\pi(s; \theta_\pi)$$

$$\theta_\pi^1 \leftarrow \theta_\pi^0 + \alpha \nabla_\theta J(\theta) \,|_{\theta=\theta_\pi^0}$$

$$\theta_\pi^0 \rightarrow \theta_\pi^1 \ \ldots \rightarrow \theta_\pi^\tau \ \ldots \rightarrow \theta_\pi^*$$

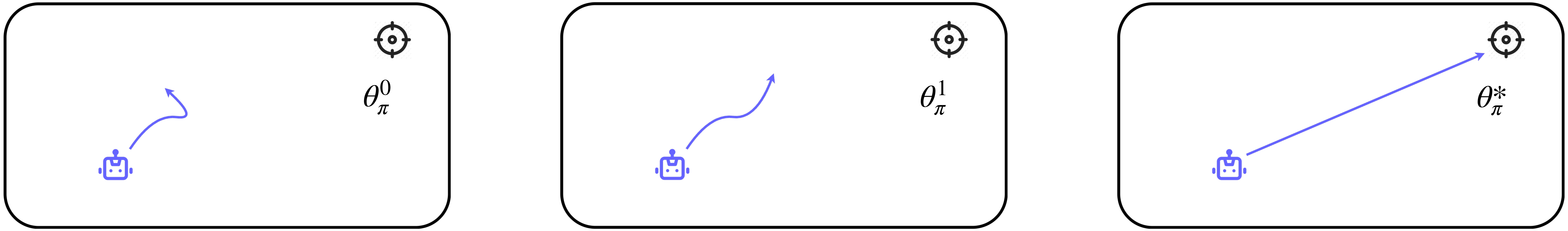
Reinforcement Learning: Policy Gradient Ascent

$$\theta_\pi^0 \rightarrow \theta_\pi^1 \ \ldots \rightarrow \theta_\pi^\tau \ \ldots \rightarrow \theta_\pi^*$$



Reinforcement Learning: Policy Gradient Ascent

$$\theta_\pi^0 \rightarrow \theta_\pi^1 \;\ldots \rightarrow \theta_\pi^\tau \;\ldots \rightarrow \theta_\pi^*$$

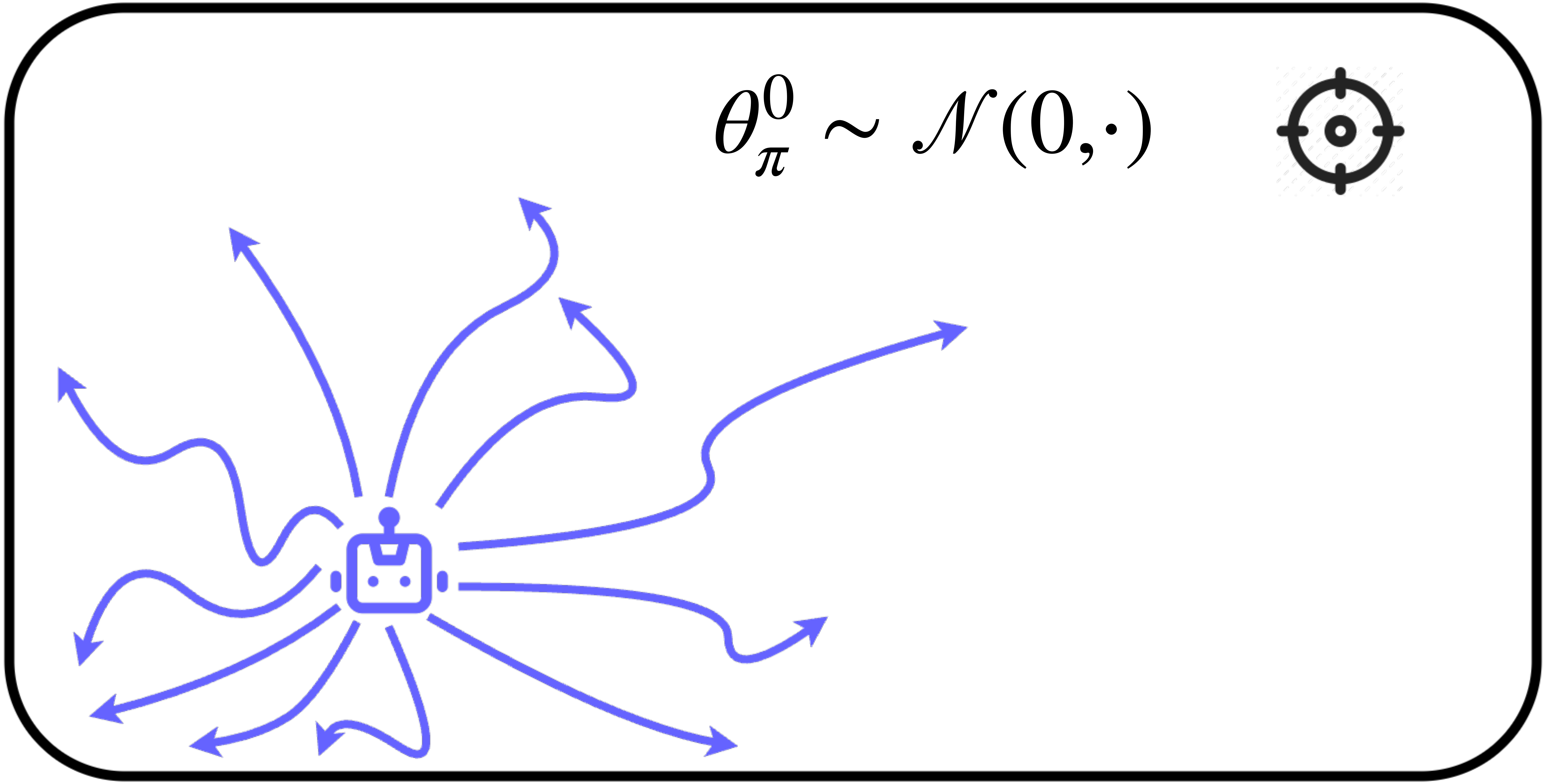


Deep Reinforcement Learning

$$\theta_\pi^0 \sim \mathcal{N}(0,\cdot)$$

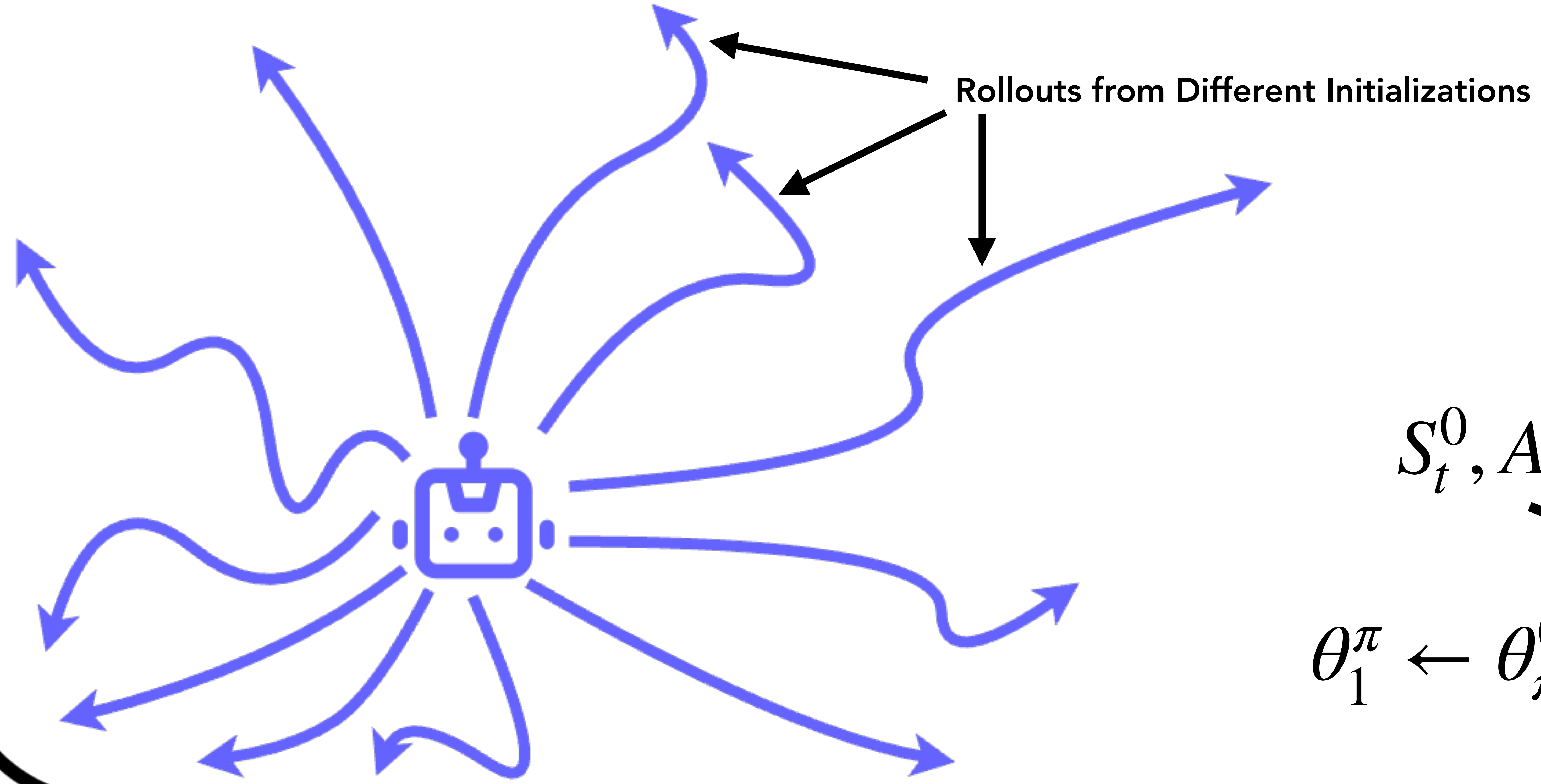
Deep Reinforcement Learning

$$\theta_\pi^0 \sim \mathcal{N}(0,\cdot)$$



Deep Reinforcement Learning

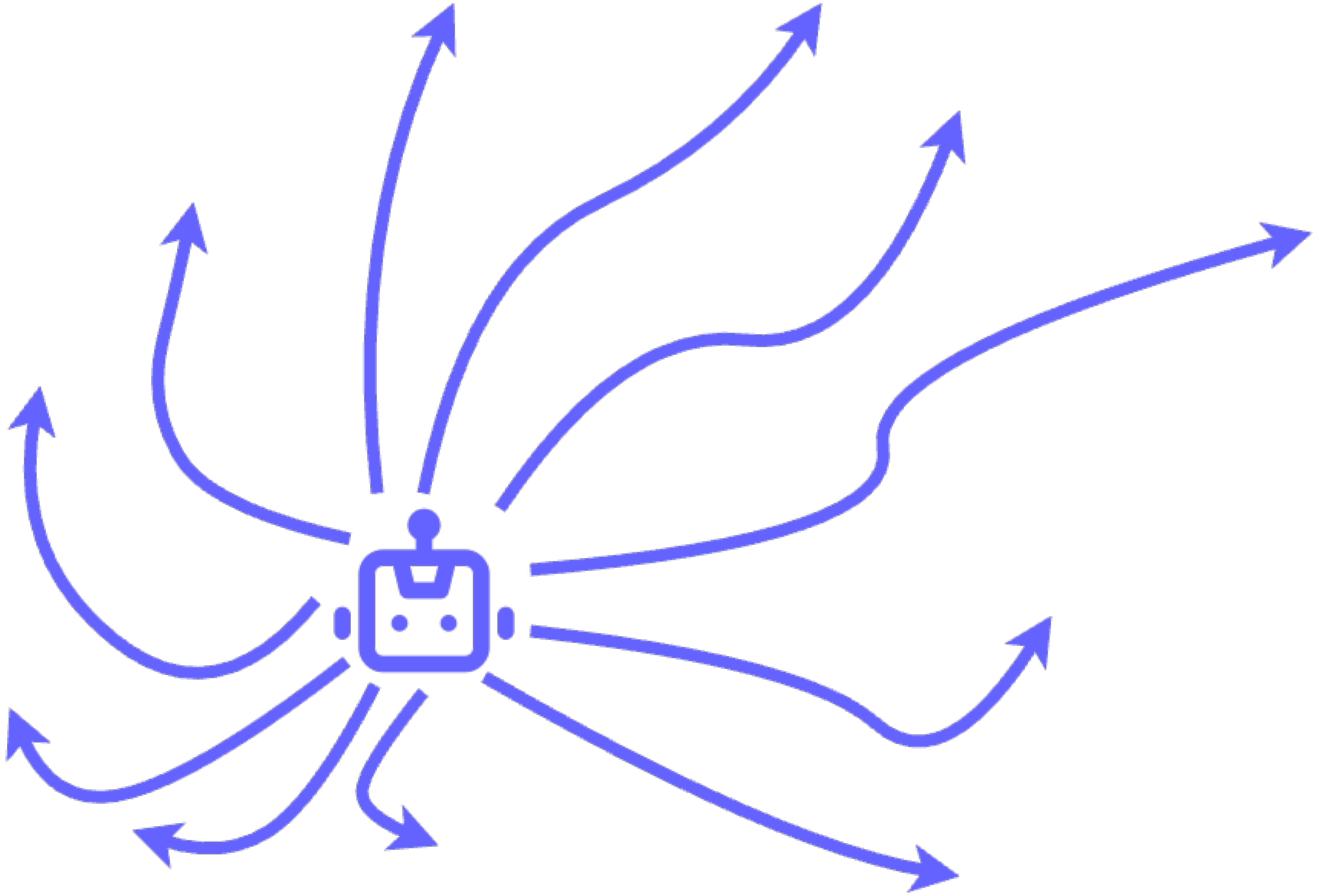
$$\theta_\pi^0 \sim \mathcal{N}(0,\cdot)$$

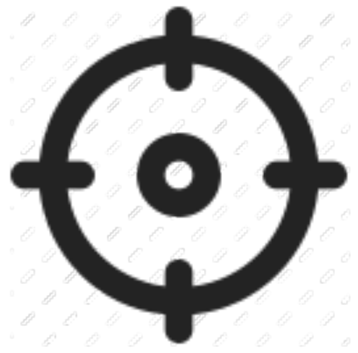


$$S_t^0, A_t^0, R_t^0 \leftarrow \theta_\pi^0$$

$$\theta_1^\pi \leftarrow \theta_\pi^0 + \alpha \nabla_{\theta_\pi} J(\theta_\pi)$$

Deep Reinforcement Learning

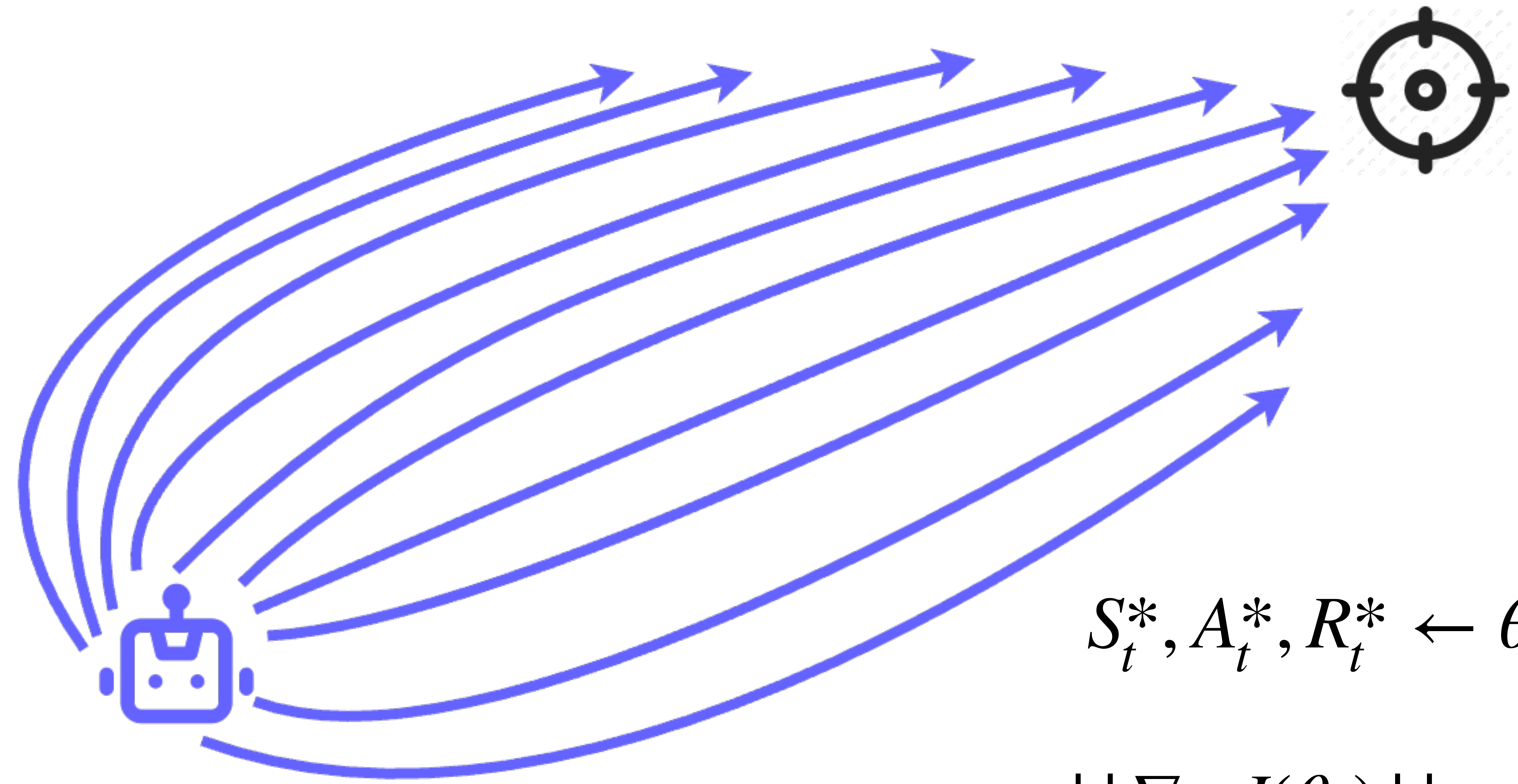




$$S_t^1, A_t^1, R_t^1 \leftarrow \theta_\pi^1$$

$$\theta_\pi^2 \leftarrow \theta_\pi^1 + \alpha \nabla_{\theta_\pi} J(\theta_\pi)$$

Deep Reinforcement Learning



$$S_t^*, A_t^*, R_t^* \leftarrow \theta_\pi^*$$

$$||\nabla_{\theta_\pi} J(\theta_\pi)|| \approx 0$$

Deep Reinforcement Learning

Distribution of parameters

$$\mathcal{N}(0,\cdot) = \mathbf{W}_{\pi}^{\mathbf{0}} \rightarrow \mathbf{W}_{\pi}^{\mathbf{1}} \ \ldots \rightarrow \mathbf{W}_{\pi}^{\tau} \ \ldots \rightarrow \mathbf{W}_{\pi}^{*}$$

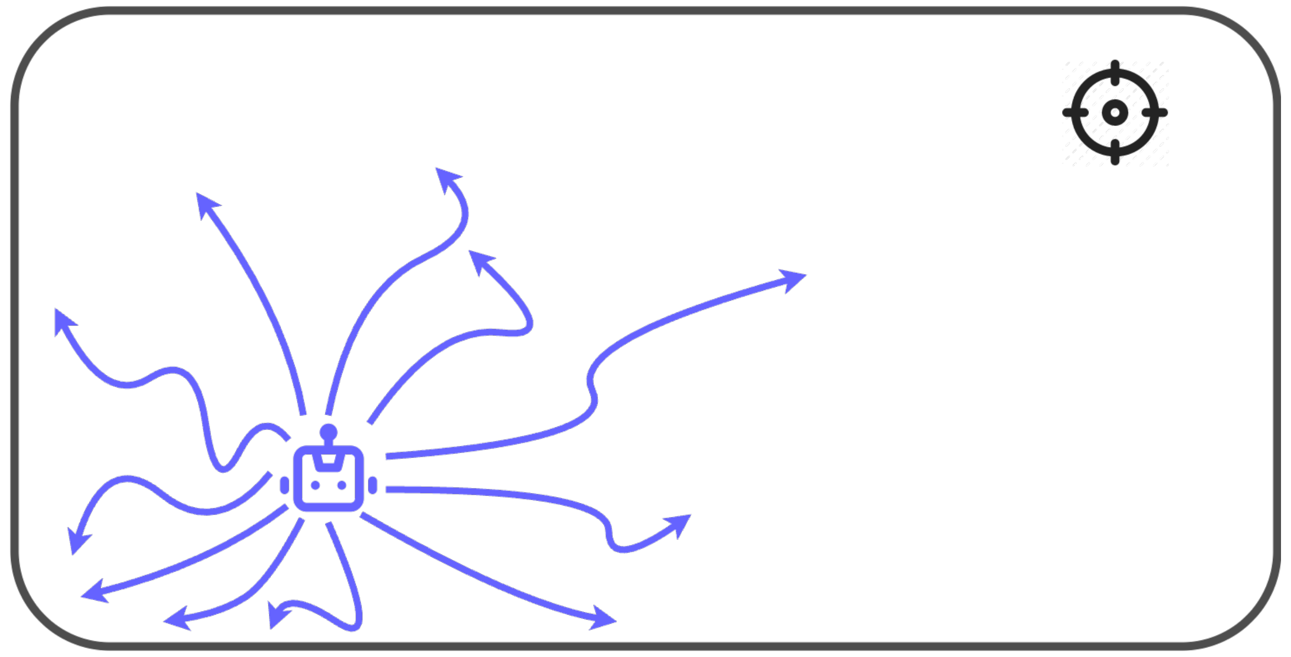
Deep Reinforcement Learning

Distribution of parameters

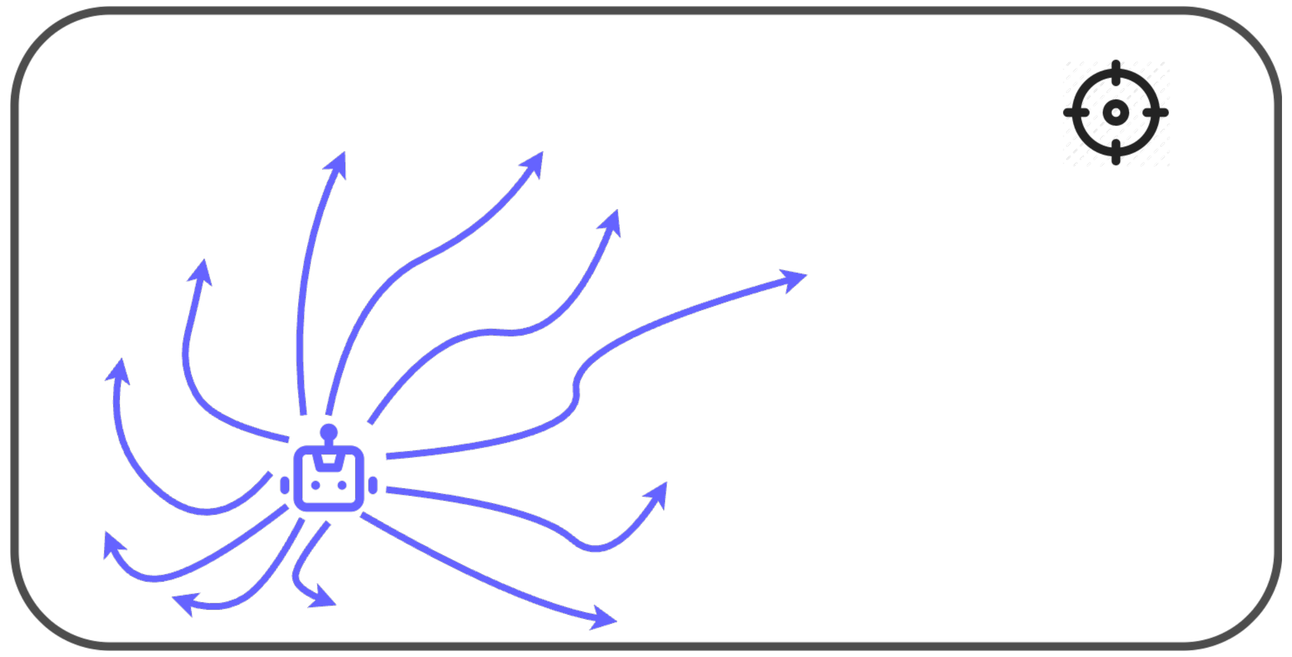
$$\mathcal{N}(0,\cdot) = \mathbf{W}_\pi^{\mathbf{0}} \rightarrow \mathbf{W}_\pi^{\mathbf{1}} \ \ldots \rightarrow \mathbf{W}_\pi^{\tau} \ \ldots \rightarrow \mathbf{W}_\pi^{*}$$

$$\boxed{S_t^{\tau}}$$

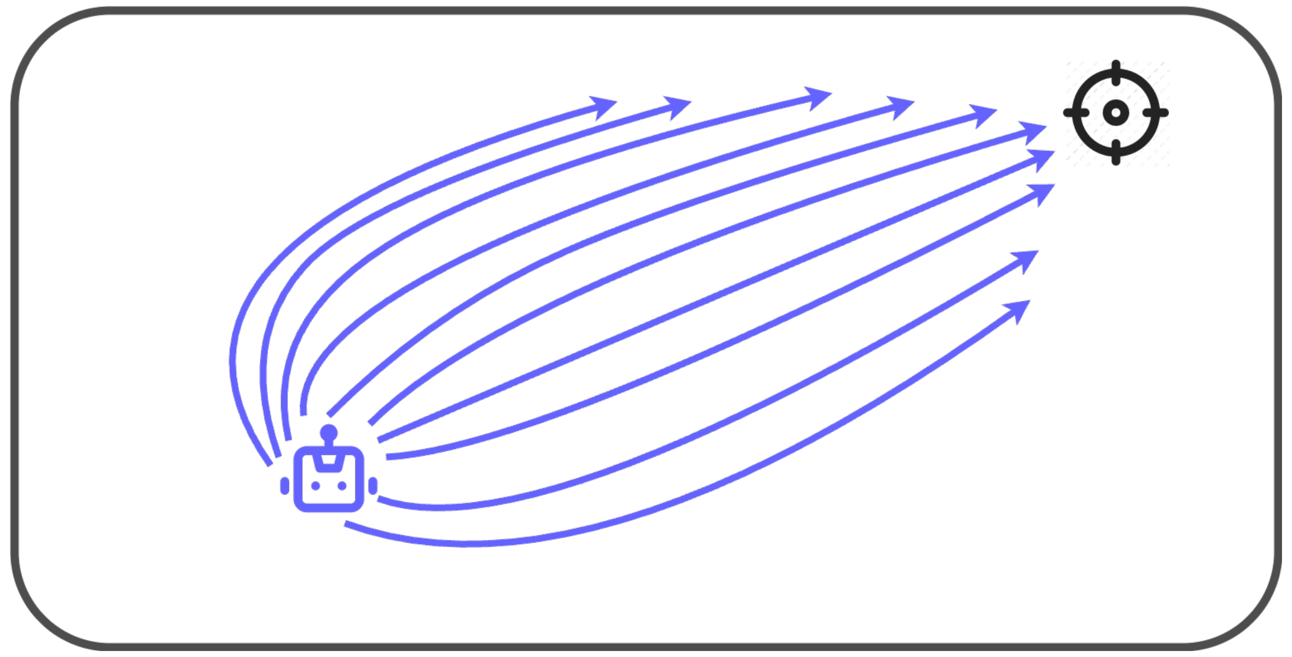
$\tau = 0$



$\tau = 1$



$\tau \rightarrow \infty$



Low-Dimensional Data in RL for Deterministic Environments

Theoretical Setting

- Deterministic transitions
- Continuous states, actions, time
- Neural networks in the infinite width limit

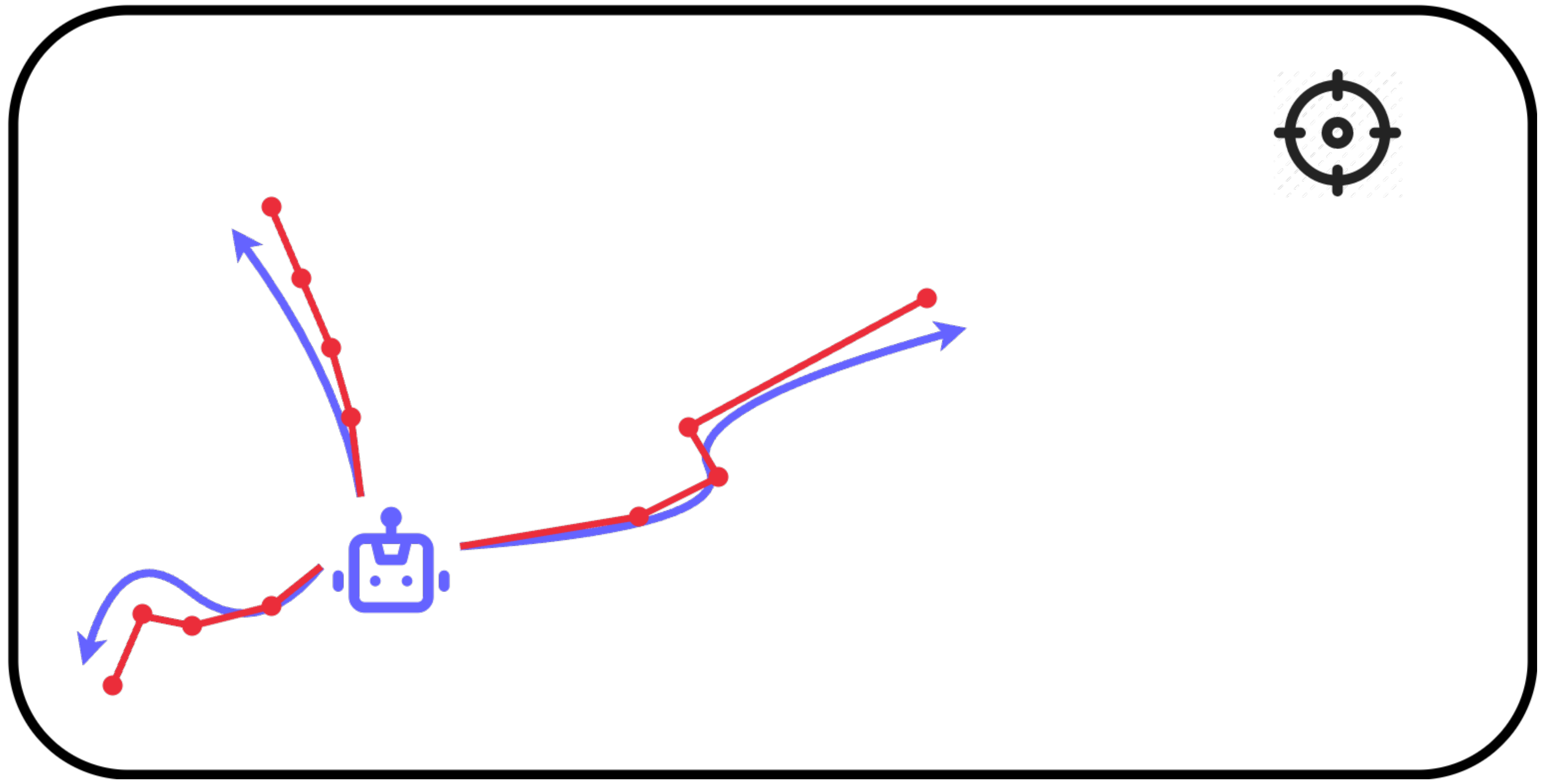
Continuous time

Why? Gradient Descent vs Gradient Flow

$$x_{t+1} \leftarrow x_t - \alpha \nabla_x L(x) \qquad \text{VS} \qquad \frac{dx(t)}{dt} = -\nabla L(x(t))$$

Continuous time

Continuous Time vs Discrete Time RL



Baird, 1994
Doya, 2000
Munos, 2006
Wang et al. 2020

Continuous state, action, time

Continuous Time Deterministic MDP

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, (g, H), r, \gamma, T, s_0 \rangle$$

Continuous state, action, time

Continuous Time Deterministic MDP

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, (g, H), r, \gamma, T, s_0 \rangle$$

$$\mathcal{S} \subseteq \mathbb{R}^{d_s} \quad \mathcal{A} \subseteq \mathbb{R}^{d_a} \qquad r(s_t) \qquad s(0) = s_0$$

Continuous state, action, time

Continuous Time Deterministic MDP

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \overbrace{(g, H)}^{\text{Transition Dynamics}}, r, \gamma, T, s_0 \rangle$$

$$\frac{ds_t}{dt} = \underbrace{g(s_t)}_{\mathbb{R}^{d_s}} + \underbrace{H(s_t)}_{\mathbb{R}^{d_s \times d_a}}\pi(s_t)$$

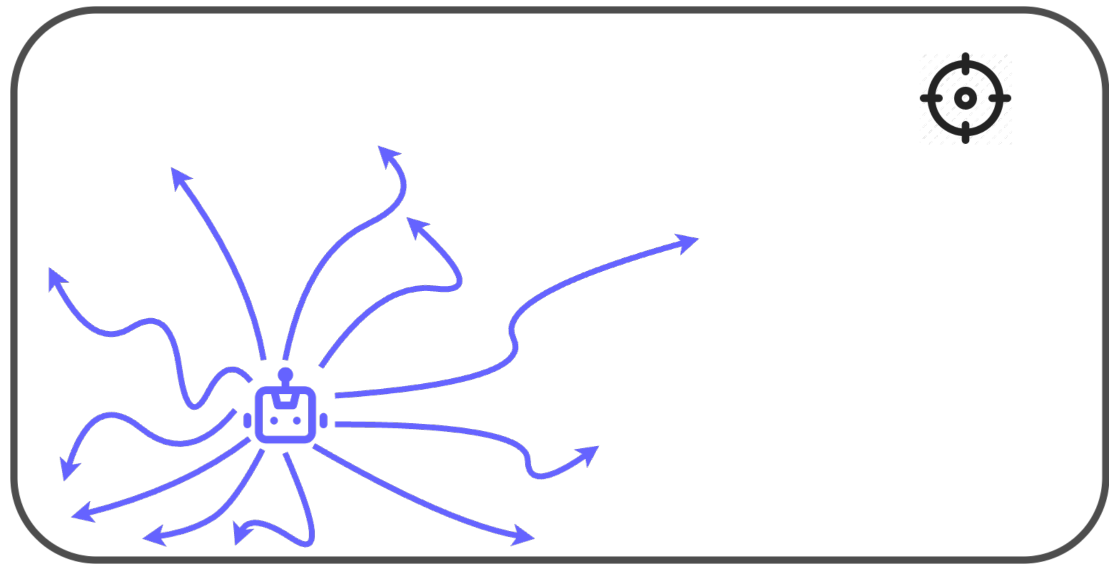
Low-Dimensional Data in RL for Deterministic Environments

$$\frac{ds_t}{dt} = g(s_t) + H(s_t)\,\Phi(s_t, \mathbf{W}_\pi^{\mathbf{0}})\mathbf{W}_\pi^\tau$$

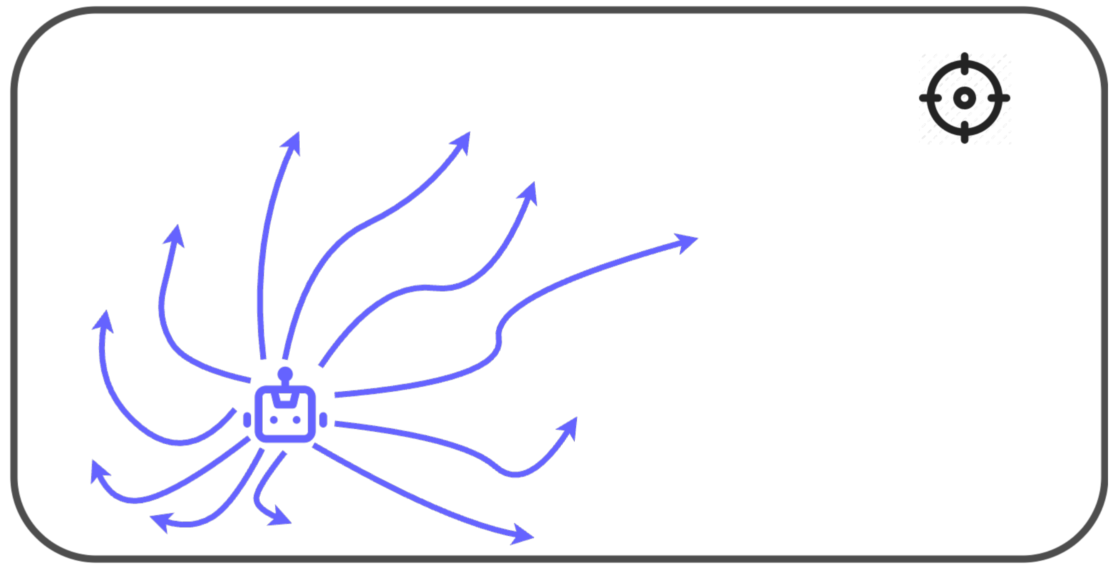
Infinite Width Neural Network

$$\boxed{S_t^\tau}$$

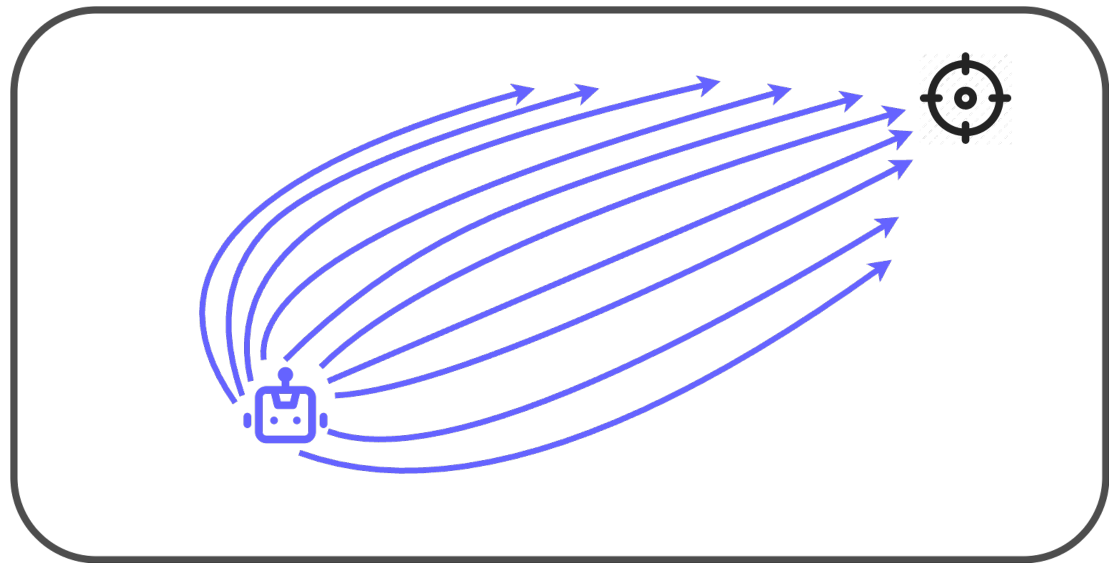
$\tau = 0$



$\tau = 1$



$\tau \to \infty$



Low-Dimensional Data in RL for Deterministic Environments

Main result:

For $t \in (0,\delta)$, starting from a fixed state s :

$$S_t^\tau(n) \xrightarrow[\text{In probability}]{\text{width } n \to \infty} \hat{S}_t^\tau + O(\delta^3)$$

Low-Dimensional Data in RL for Deterministic Environments

Main result:

For $t \in (0,\delta)$, starting from a fixed state s :

$$S_t^\tau(n) \xrightarrow[\text{In probability}]{\text{width } n \to \infty} \hat{S}_t^\tau + O(\delta^3)$$

$$\Pr(\text{distance}(\hat{S}_t^\tau, M) \geq D) \leq e^{-\mathscr{R}_\tau D}$$

$$\dim(M) \leq 2d_a + 1$$

Low-Dimensional Data in RL for Deterministic Environments

Informal main result:

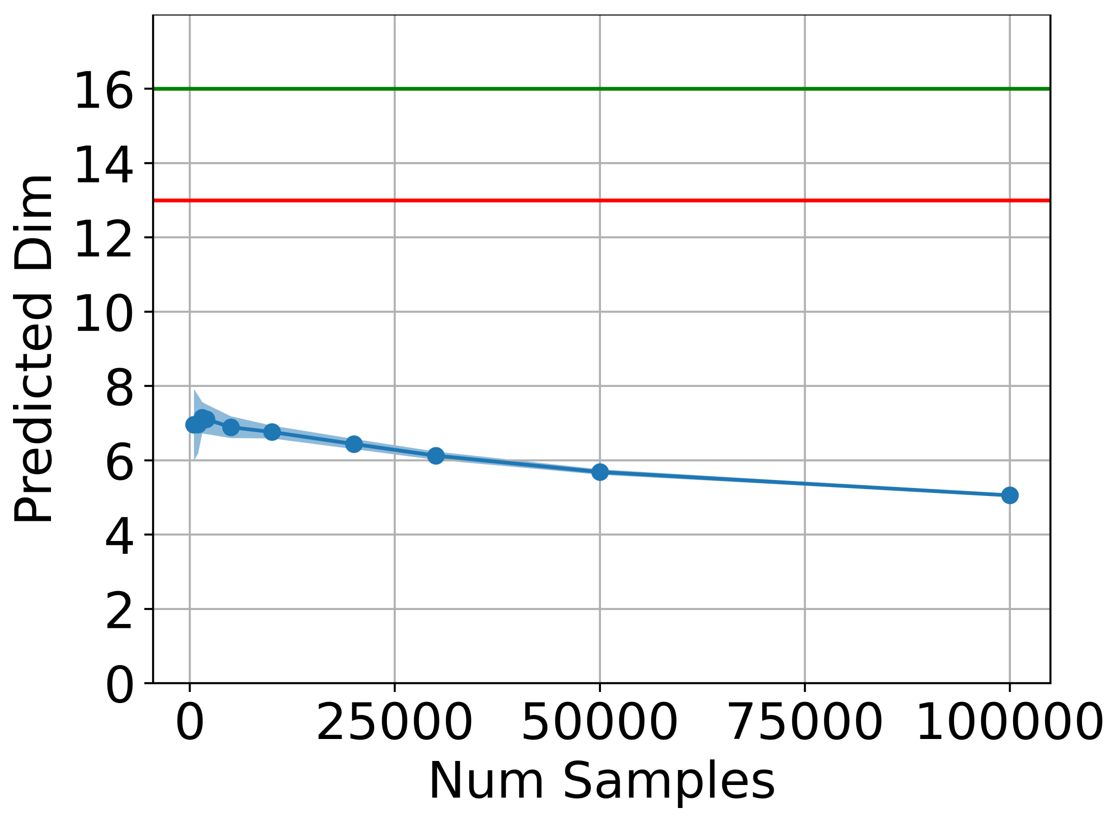
For a "well behaved" policy gradient algorithm with wide neural network policy and small learning rates: the agent's states are concentrated around a low dimensional manifold.

Empirical evidence: dimensionality estimation

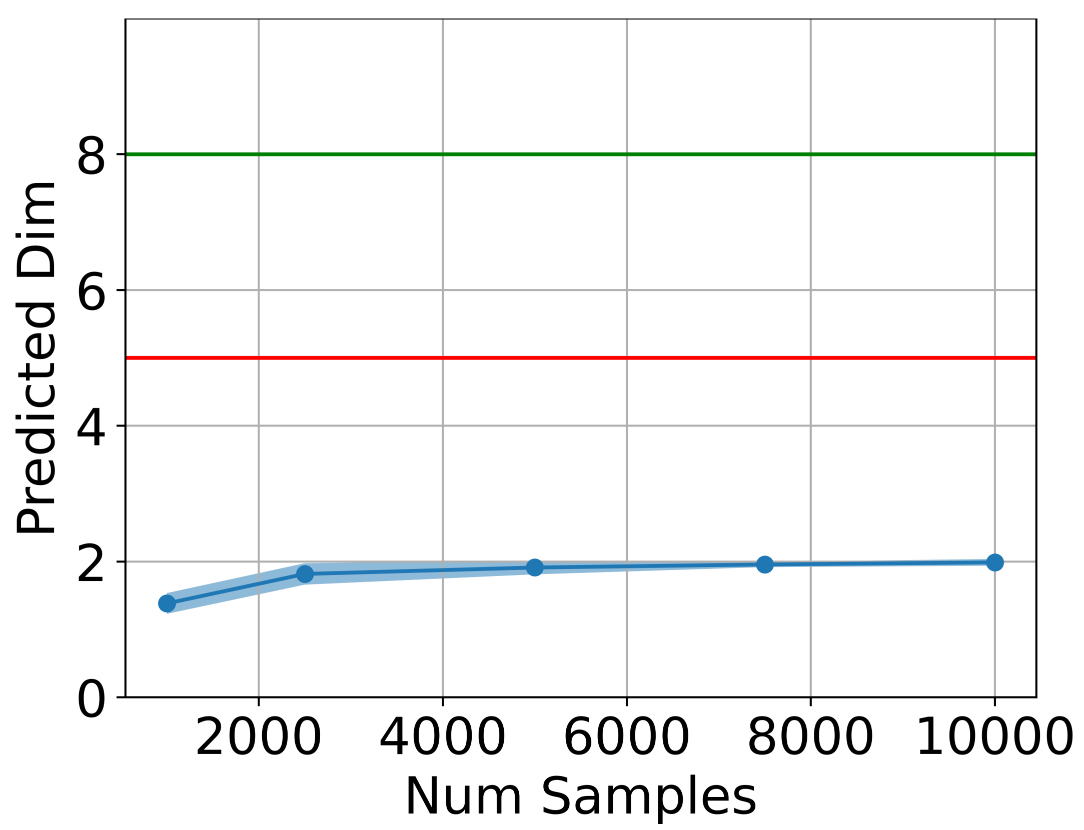
Facco et al 2017: empirical estimates

Green $= d_s$, Red $= 2d_a + 1$

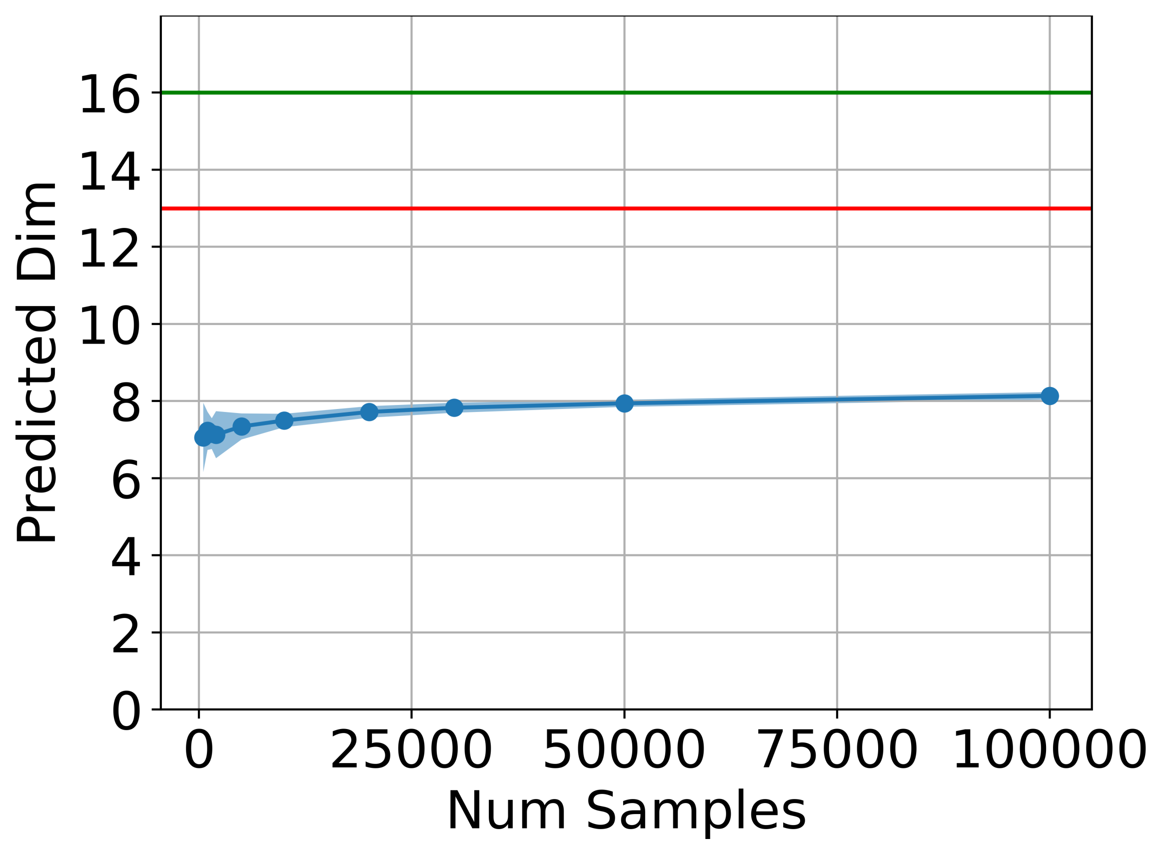
Walker



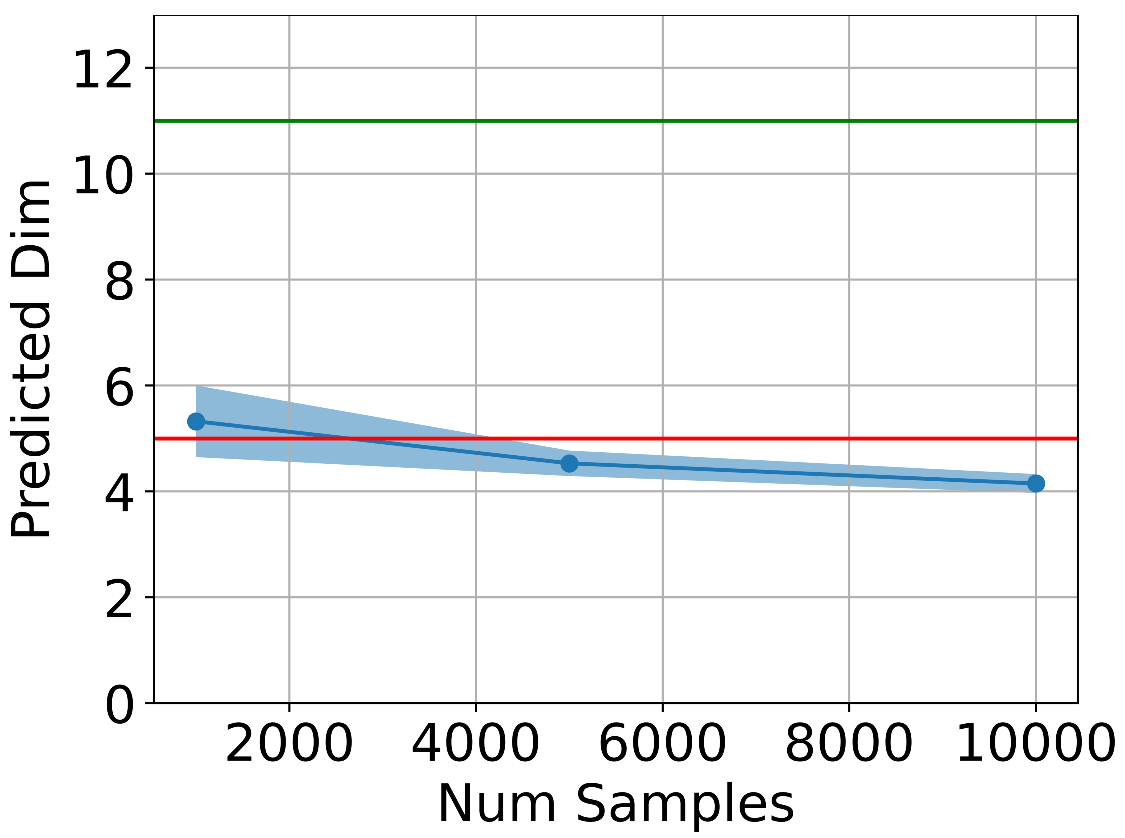
Pendulum



Cheetah



Reacher

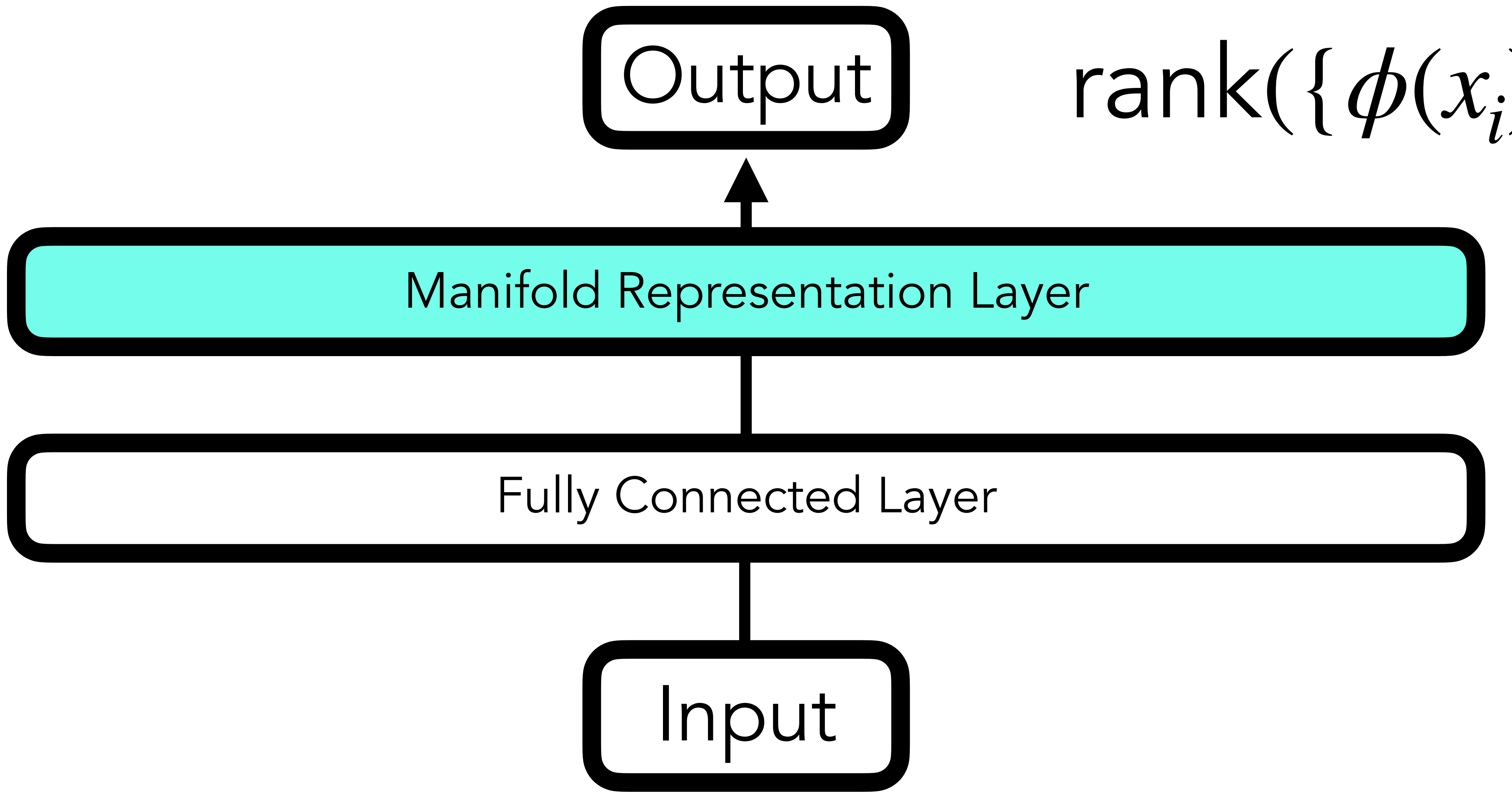


Application: Exploit the Underlying Structure

Manifold Representation Layer

Can we learn low-dimensional representations as theory suggests?

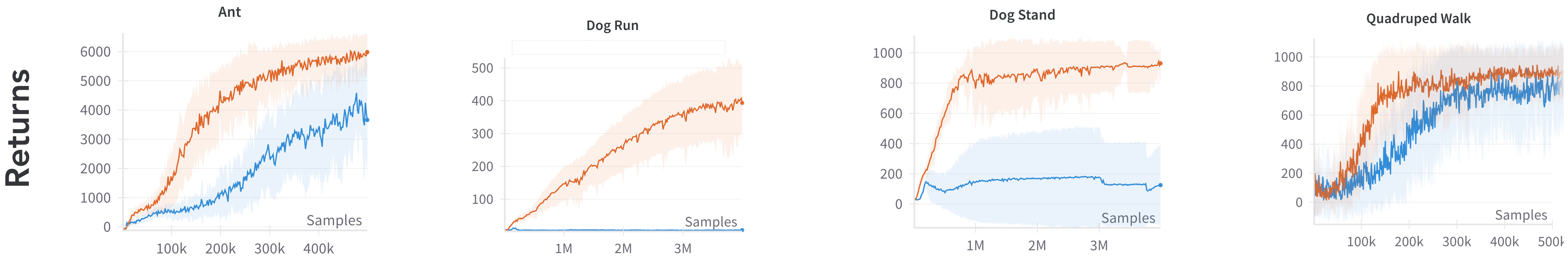
Yu et al. 2023, White box transformer



$$\mathrm{rank}(\{\phi(x_i)\}_{i=1}^{D}) = m, \quad x_i \in M$$

Manifold Representation Layer

Red = Ours, Blue = Baseline



There is a **low-dimensional structure** to data in reinforcement learning and neural networks are well suited to **exploit this structure.**

Hire me for postdoc!

saket.ai