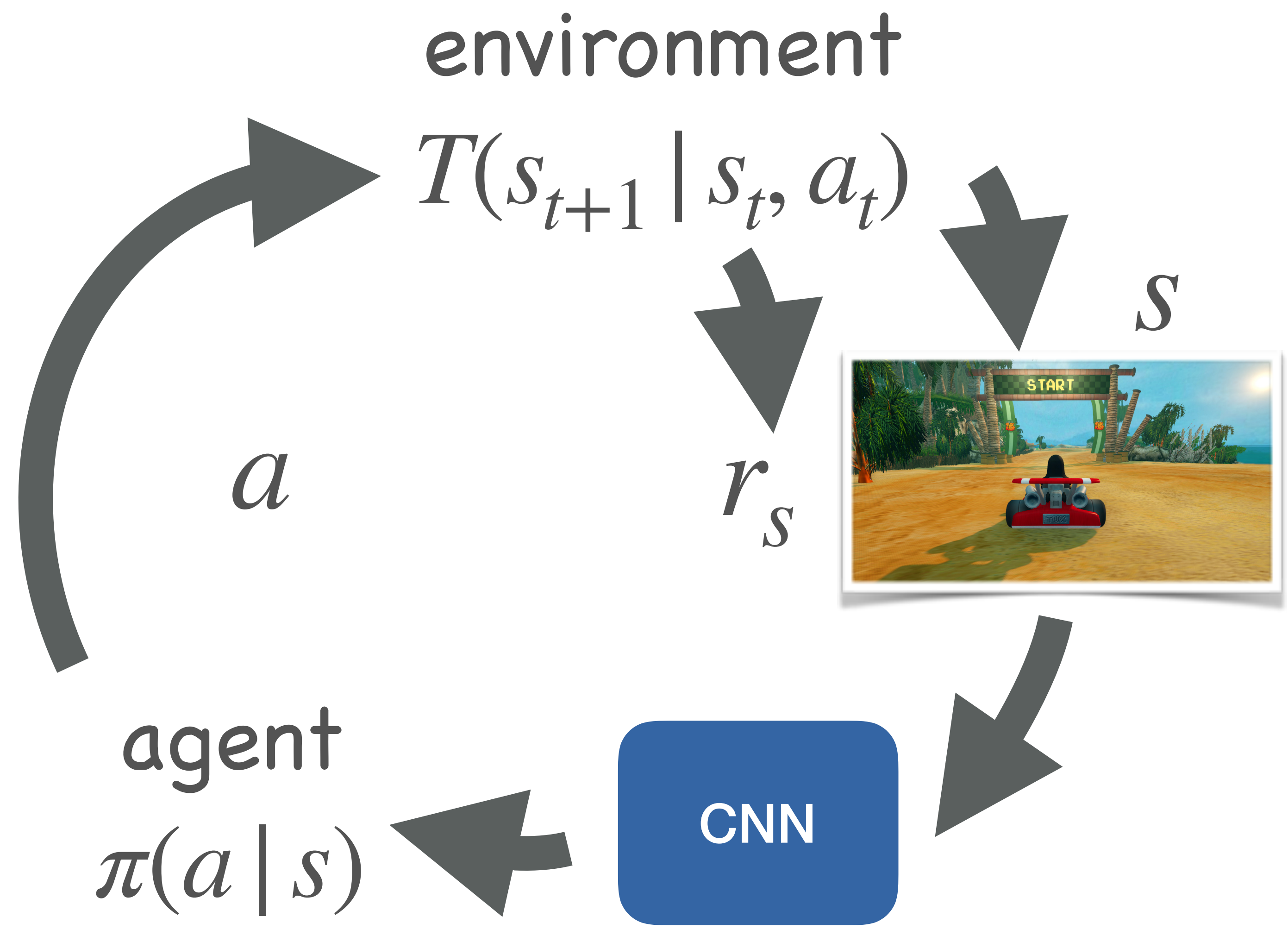
REINFORCE

Non-differentiability

- Compute gradient of
  $\mathbb{E}_{\tau \sim P_{\pi,T}}[R(\tau)]$
  $= \sum_{\tau} P_{\pi,T}(\tau) R(\tau)$



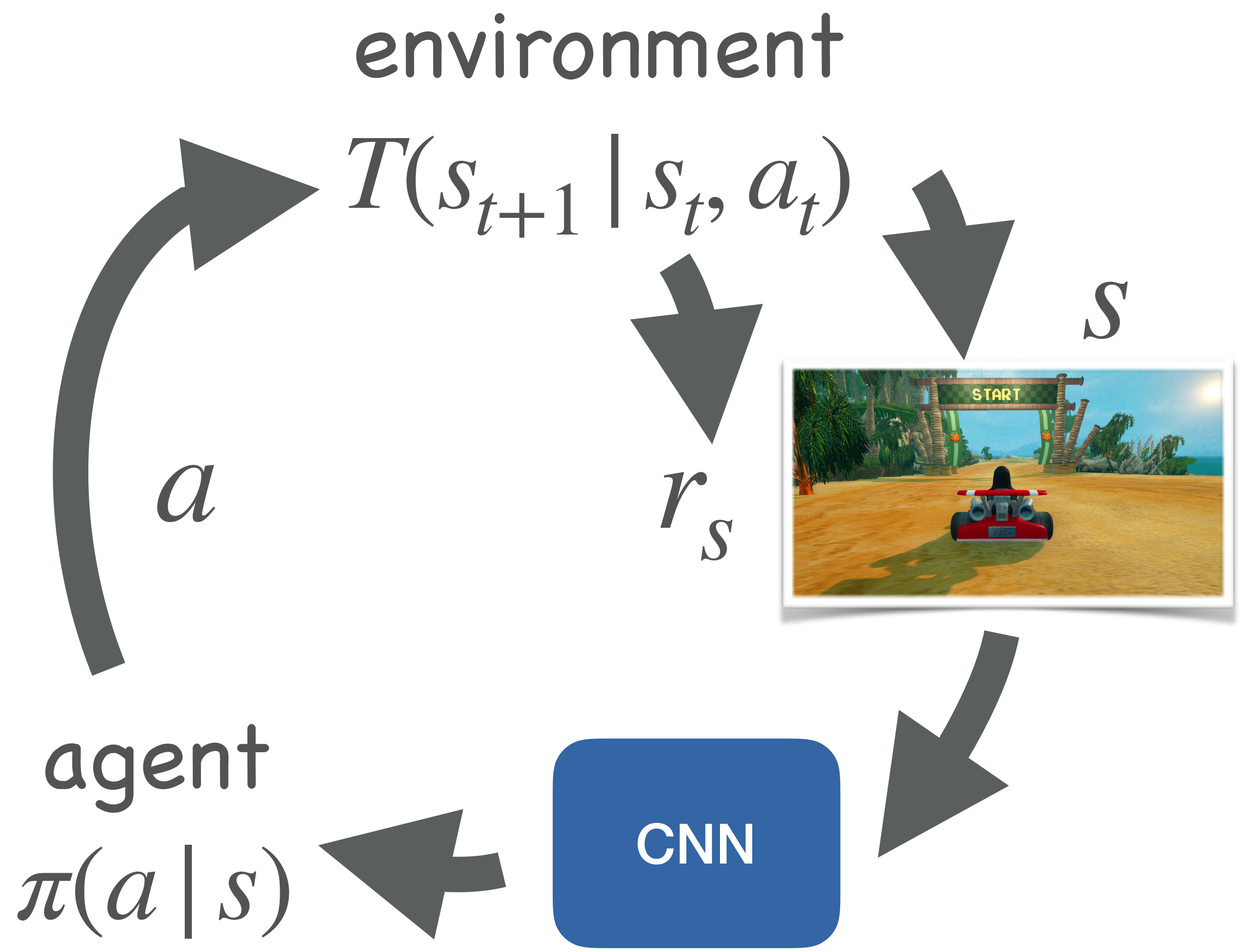
The log-derivative trick

- Simple chain rule

$$\nabla_\theta p_\theta(x) = p_\theta(x)\, \nabla_\theta \log p_\theta(x)$$

- Gradient of expected return

$$\nabla \mathbb{E}_{\tau \sim P_{\pi,T}}[R(\tau)] = \sum_\tau P_{\pi,T}(\tau) R(a)\, \nabla \log P_{\pi,T}(\tau)$$
$$= \mathbb{E}_{\tau \sim P_{\pi,T}}\left[R(\tau)\, \nabla \log P_{\pi,T}(\tau)\right]$$



REINFORCE

- Compute gradient using Monte Carlo sampling

$$\mathbb{E}_{\tau \sim P_{\pi,T}} \left[R(\tau) \nabla \log P_{\pi,T}(\tau) \right]$$

$$\approx \frac{1}{N} \sum_{\tau \sim P_{\pi,T}} \left[R(\tau) \nabla \log P_{\pi,T}(\tau) \right]$$

Simple statistical gradient-following algorithms for connectionist reinforcement learning, Williams, Machine learning 1992

REINFORCE issues

$$\frac{1}{N} \sum_{\tau \sim P_{\pi,T}} \left[R(\tau) \nabla \log P_{\pi,T}(\tau) \right]$$

- Needs lots of samples for a good gradient
  - High-variance gradient estimator
  - Cannot reuse rollouts (τ)