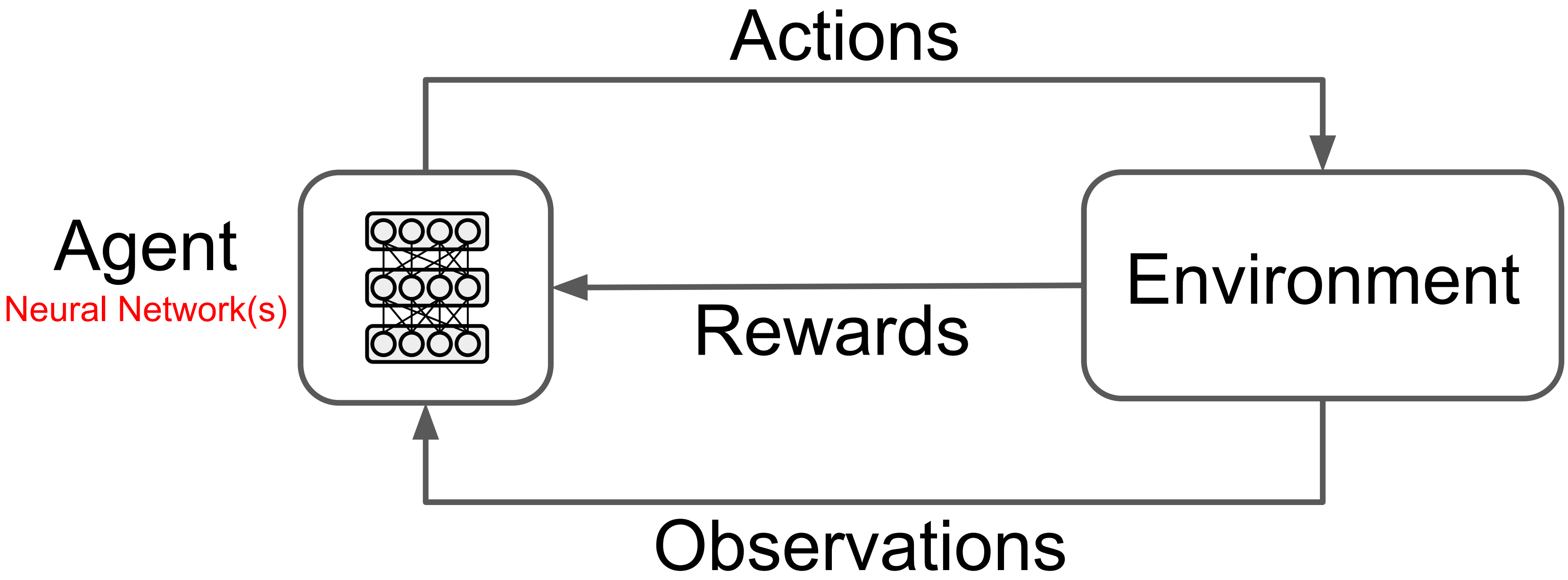
# Recent advances in model-free and model-based reinforcement learning

Timothy Lillicrap
Research Scientist, DeepMind & UCL

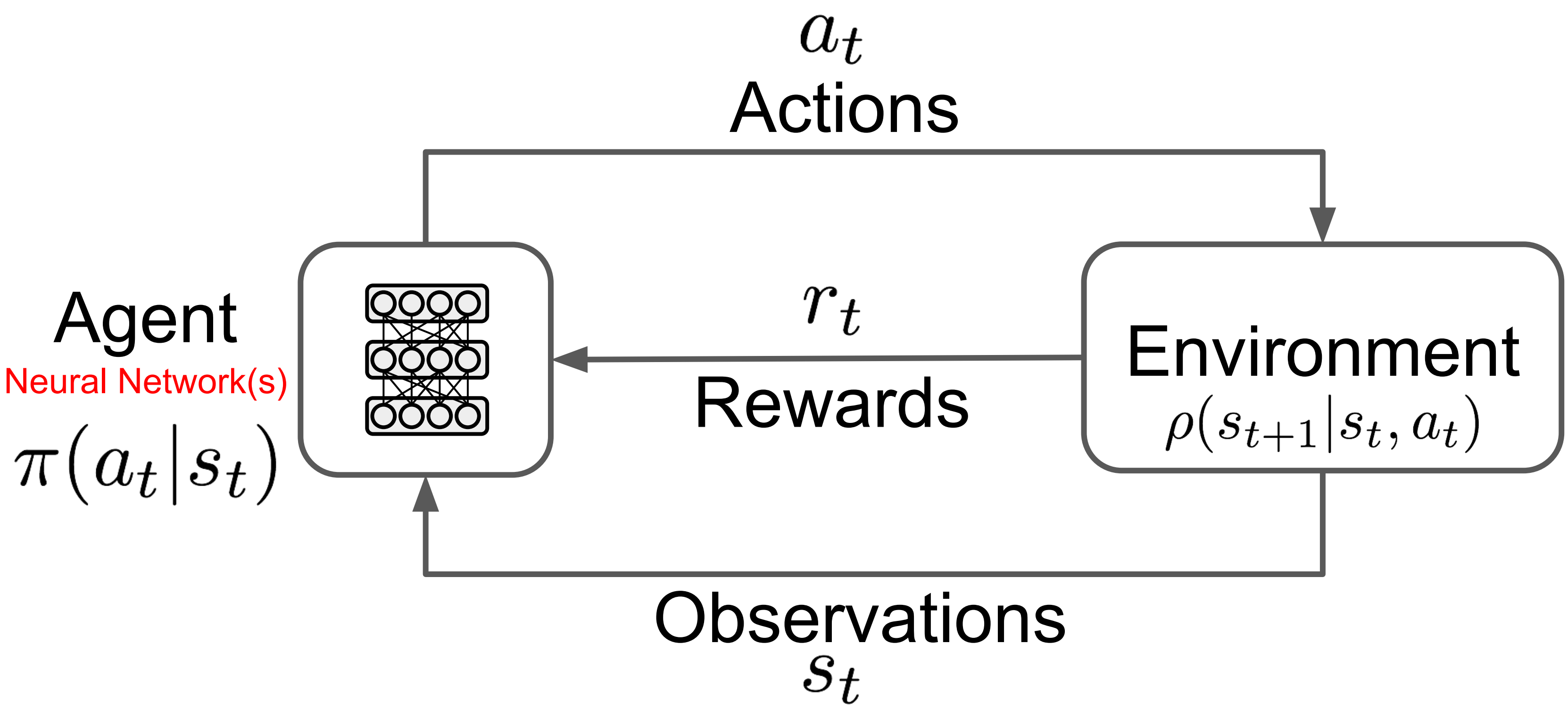
20 18 IMAG Futures

DeepMind

# What is Deep Reinforcement Learning?



# Formalizing the Agent-Environment Loop



# Measuring Outcomes

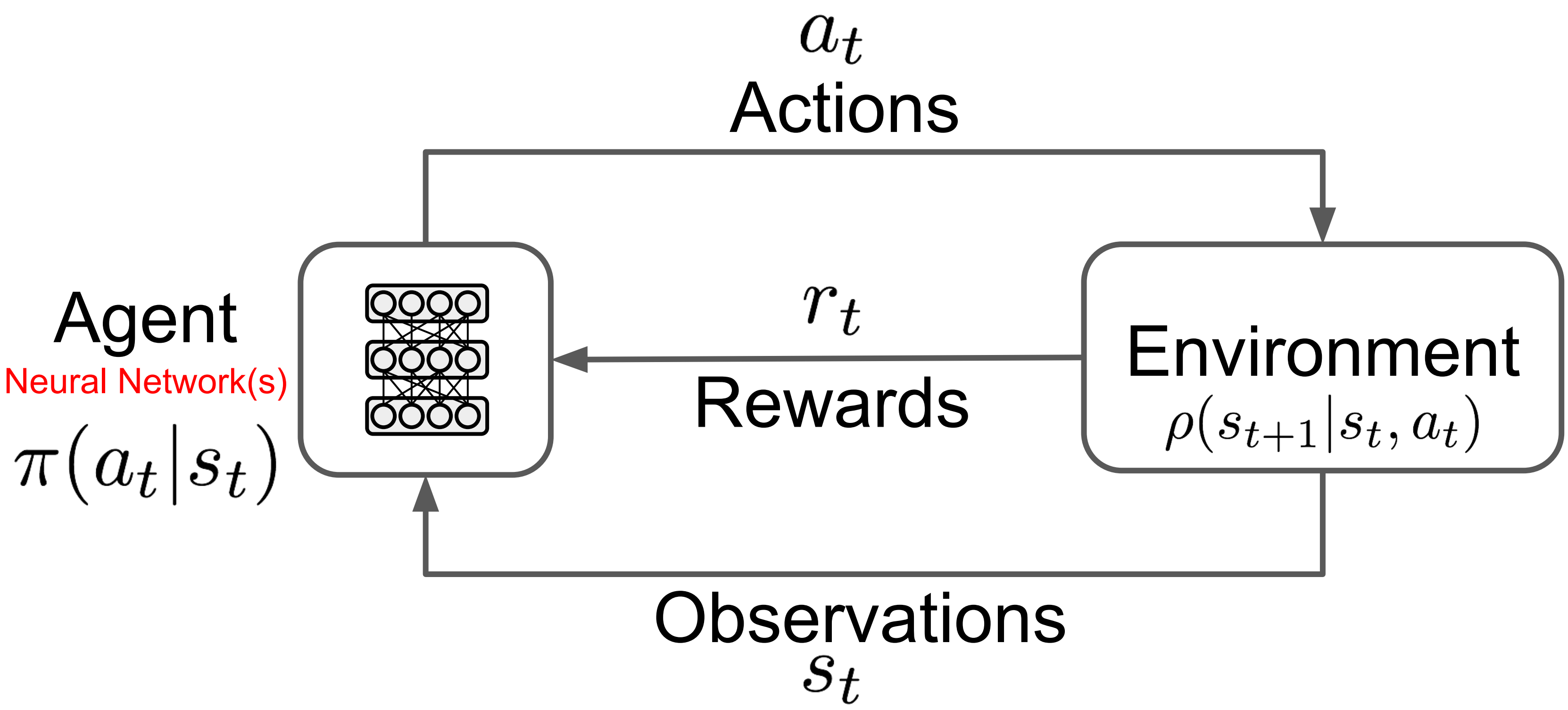
Return for a single trial:

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$$

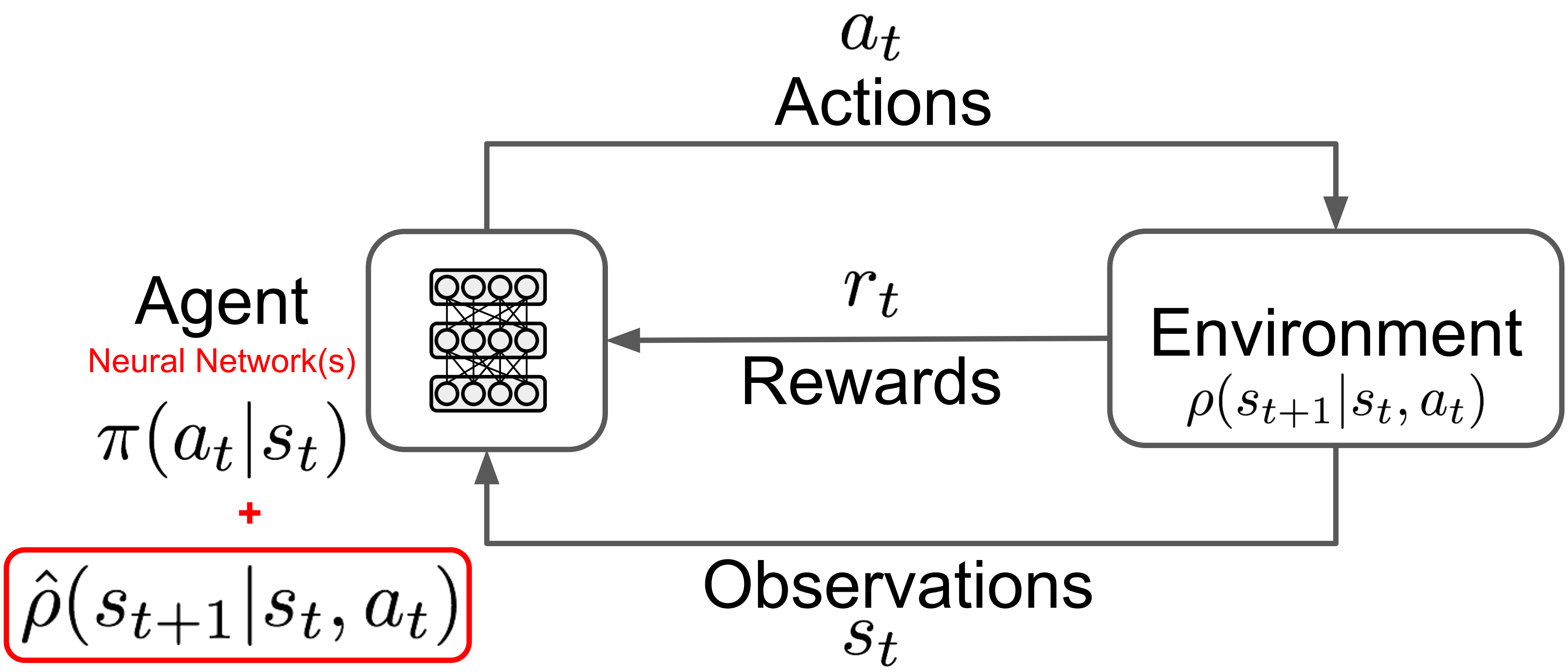
Objective function:

$$J(\theta) = \int_{\mathbb{T}} p_\theta(\tau) R(\tau) d\tau$$

# Formalizing the Agent-Environment Loop



# Model-free versus model-based RL



# A Single Trial

$$
\begin{array}{ccccc}
r_0, & r_1, & r_2, & \ldots, & r_T \\
a_0, & a_1, & a_2, & \ldots, & a_T \\
\pi(a_0|s_0), & \pi(a_1|s_1), & \pi(a_2|s_2), & \ldots, & \pi(a_T|s_T) \\
s_0, & s_1, & s_2, & \ldots, & s_T \\
 & \rho(s_1|s_0, a_0), & \rho(s_2|s_1, a_1), & \ldots, & \rho(s_T|s_{T-1}, a_{T-1})
\end{array}
$$

Time $\longrightarrow$

Probability of trajectory $\tau$

$$p_\theta(\tau) = \rho(s_0) \prod_{t=0}^{T} \rho(s_{t+1}|s_t, a_t) \pi_\theta(a_t|s_t)$$

# The Policy Gradient

Derivative of the log function

+

Chain Rule

$$\nabla_x \log f(x) = \frac{1}{f(x)} \nabla_x f(x)$$

$$\Longrightarrow$$

$$\nabla_x f(x) = f(x) \nabla_x \log f(x)$$

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int_{\mathbb{T}} \boxed{\nabla_\theta p_\theta(\tau)} R(\tau) d\tau \\
&= \int_{\mathbb{T}} \boxed{p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)} R(\tau) d\tau \\
&= \mathbb{E}_{\pi_\theta} \left[ \nabla_\theta \log p_\theta(\tau) R(\tau) \right]
\end{aligned}$$

# The Policy Gradient

$$
\begin{array}{rrcl}
 & p_\theta(\tau) & = & \rho(s_0)\prod_{t=0}^{T}\rho(s_{t+1}|s_t,a_t)\pi_\theta(a_t|s_t) \\
\Longrightarrow & \log p_\theta(\tau) & = & \log\rho(s_0)+\sum_{t=0}^{T}\log\rho(s_{t+1}|s_t,a_t)+\sum_{t=0}^{T}\log\pi_\theta(a_t|s_t) \\
\Longrightarrow & \nabla_\theta\log p_\theta(\tau) & = & \sum_{t=0}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)
\end{array}
$$

$$
\begin{aligned}
\Longrightarrow \ \nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log p_\theta(\tau) R(\tau)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)R(\tau)\right]
\end{aligned}
$$

The environment dynamics disappear from the policy gradient!

# Update Parameters with the Policy Gradient

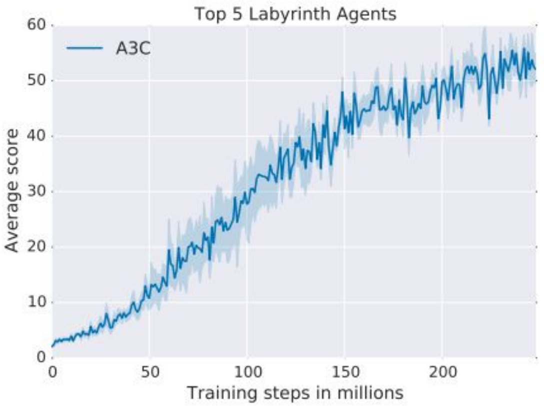
1. Sample a trajectory by rolling out the policy:

$$\tau \sim \begin{array}{ccccc} r_0, & r_1, & r_2, & \ldots, & r_T \\ a_0, & a_1, & a_2, & \ldots, & a_T \\ \pi(a_0|s_0), & \pi(a_1|s_1), & \pi(a_2|s_2), & \ldots, & \pi(a_T|s_T) \\ s_0, & s_1, & s_2, & \ldots, & s_T \\ & \rho(s_1|s_0,a_0), & \rho(s_2|s_1,a_1), & \ldots, & \rho(s_T|s_{T-1},a_{T-1}) \end{array}$$

2. Compute an estimate of the policy gradient and update network parameters:

$$\theta_{i+1} = \theta_i + \eta \nabla_\theta J(\hat{\theta})|_{\theta=\theta_i}$$

# Training Neural Networks with Policy Gradients




100s of Millions of steps!

The policy gradient has high variance.


Mnih et al., *ICML* 2016

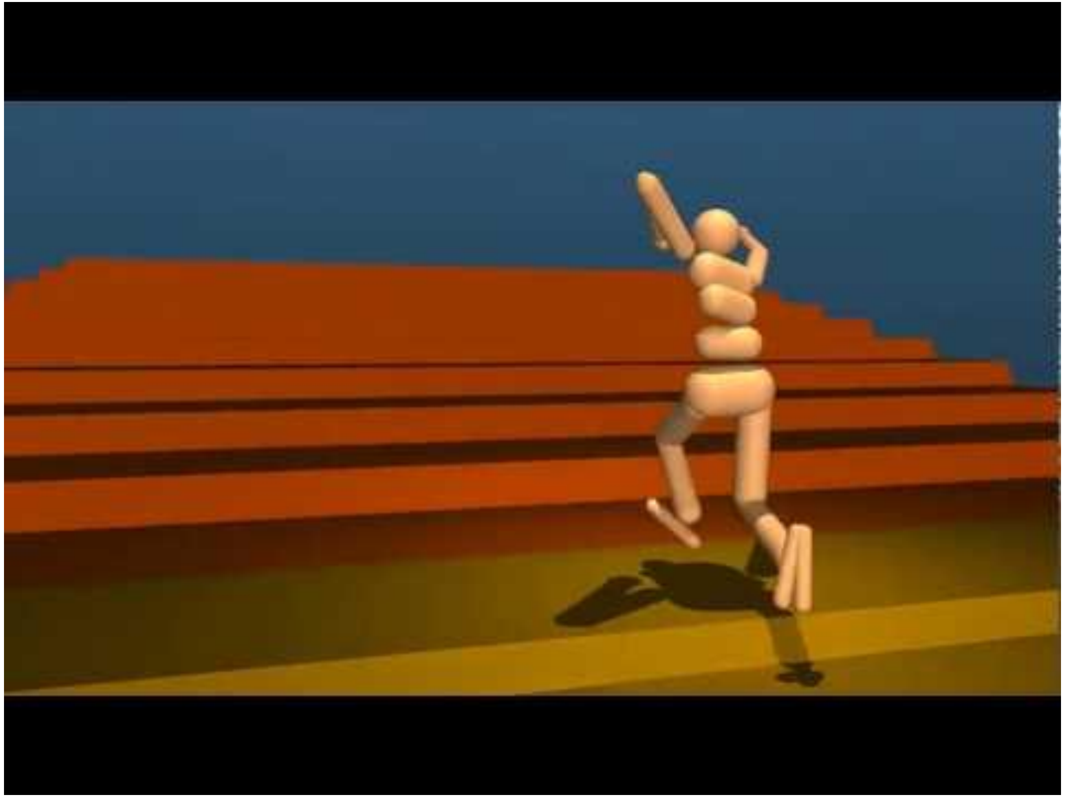
# Combating Variance with Value Functions

$$V^\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=\tau}^{T} \gamma^t r_t | s_t = s \right]$$

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{t=\tau}^{T} \gamma^t r_t | s_t = s, a_t = a \right]$$

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$$

# Proximal Policy Gradient for Flexible Behaviours

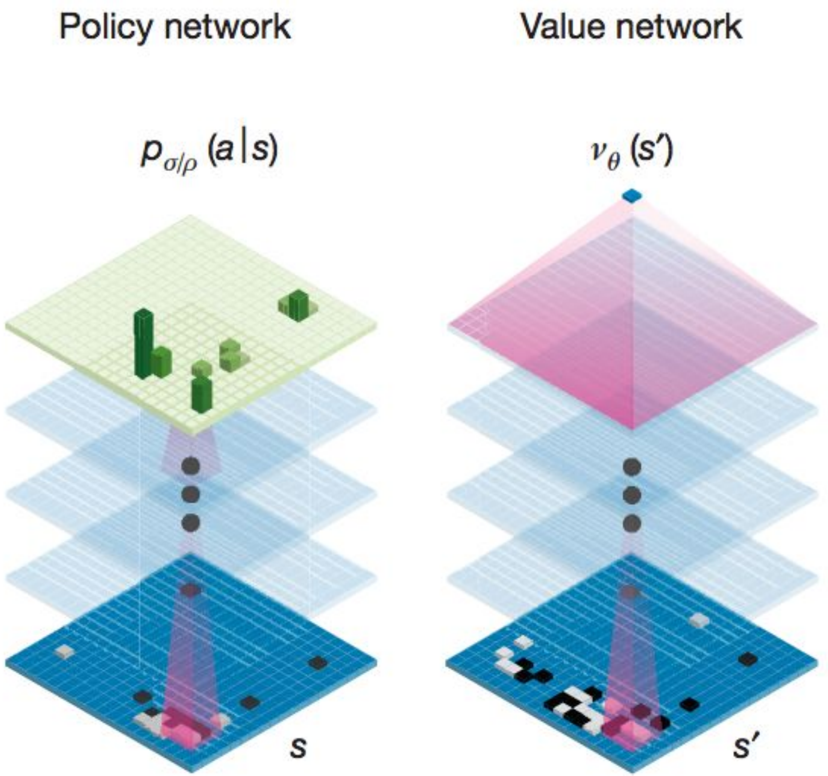


Heess et al., 2017

# Playing Go with Deep Networks and Planning

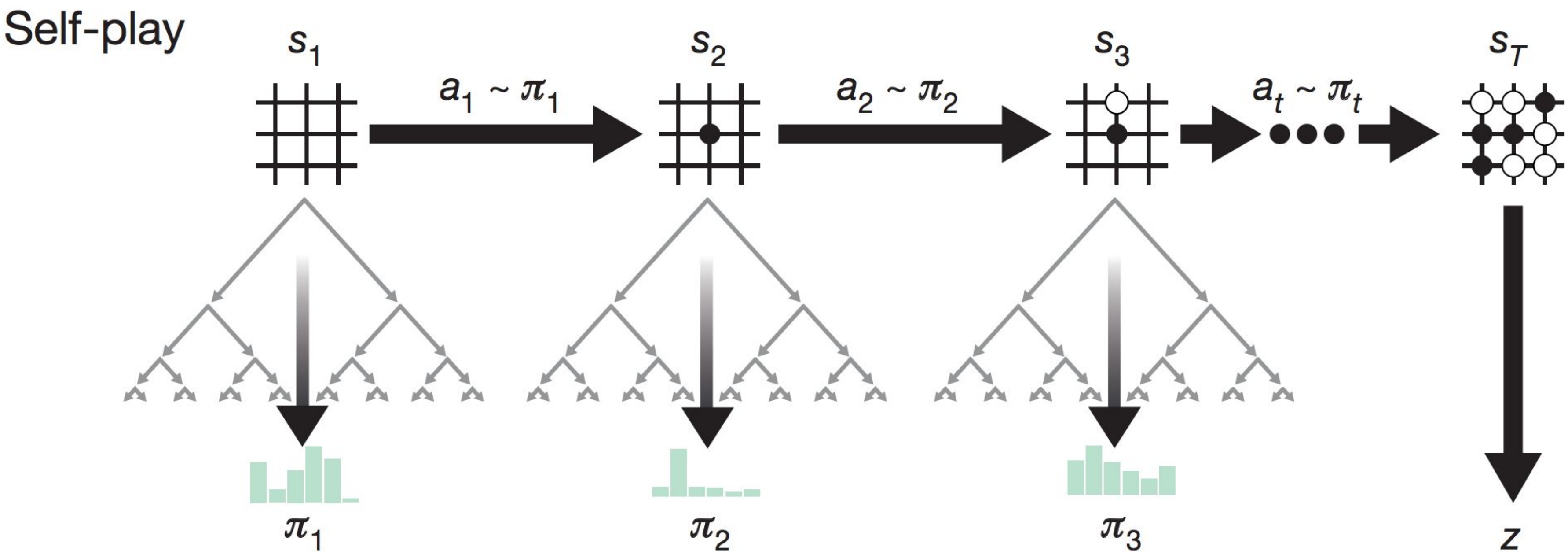
$$\rho(s_{t+1} | s_t, a_t)$$

Use environment model
in order to plan!



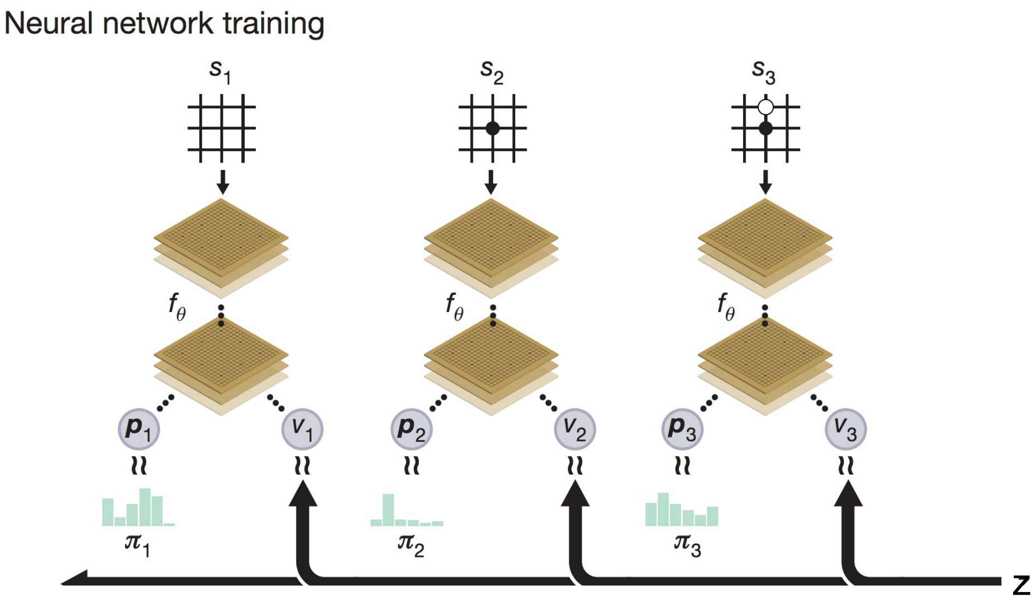

Silver, Huang et al., *Nature*, 2016

# Playing Go with Without Human Knowledge




Silver, Schrittwieser, Simonyan, et al. *Nature*, 2017

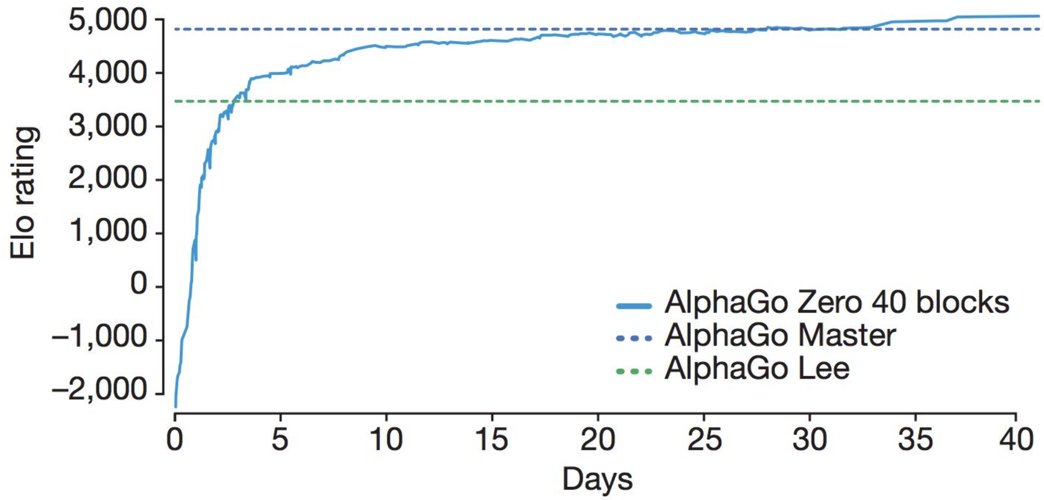
# Playing Go with Without Human Knowledge

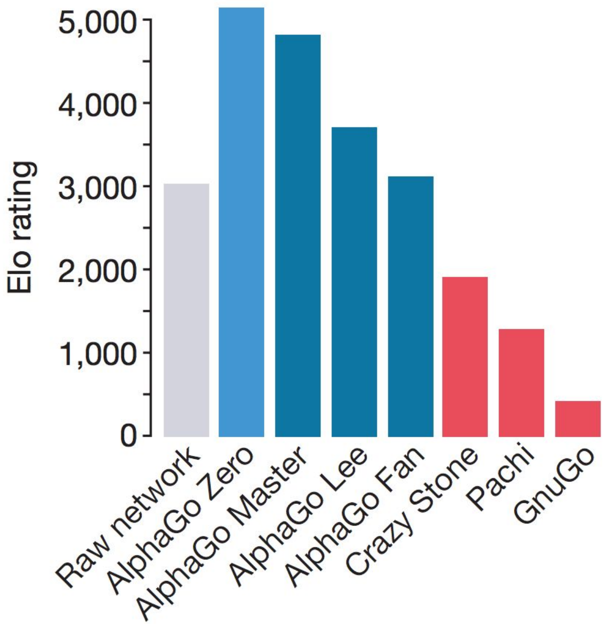



$$(\boldsymbol{p}, v) = f_\theta(s) \quad \text{and} \quad l = (z - v)^2 - \boldsymbol{\pi}^{\mathrm{T}} \log \boldsymbol{p} + c\|\theta\|^2$$

Silver, Schrittwieser, Simonyan, et al. *Nature*, 2017

# Playing Go with Without Human Knowledge

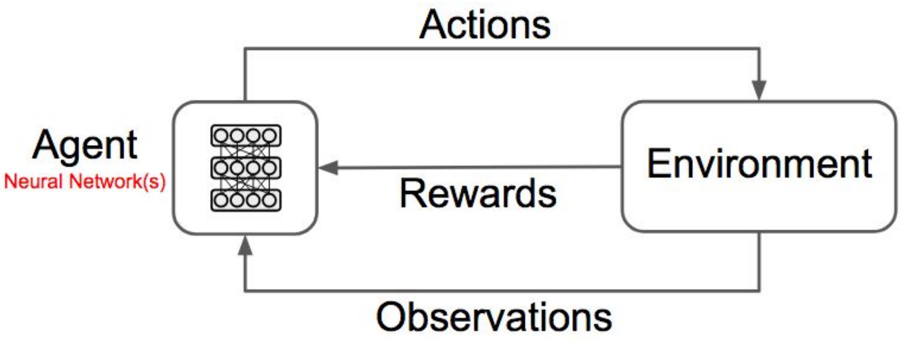



Silver, Schrittwieser, Simonyan, et al. *Nature*, 2017

# Model-based RL in unknown environments

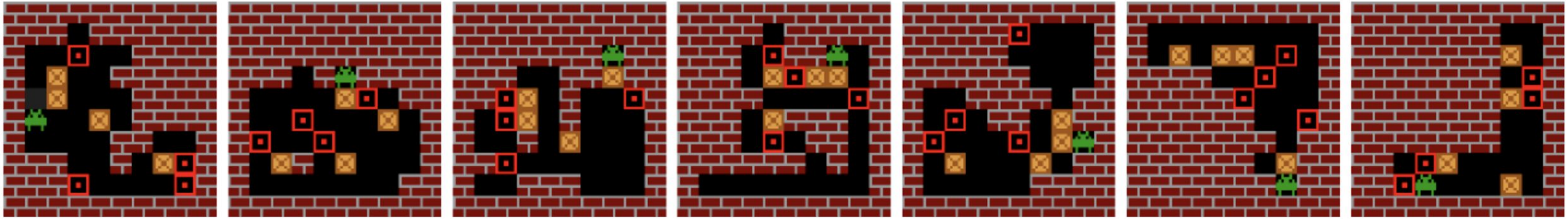
- Learned / imperfect dynamics models are difficult to leverage for benefits in complex environments.
- In part this is because planners will exploit model imperfections.
- Modelling uncertainty is one possible solution.
- Another is to allow a model-free component to decide *when* to trust a causal model.

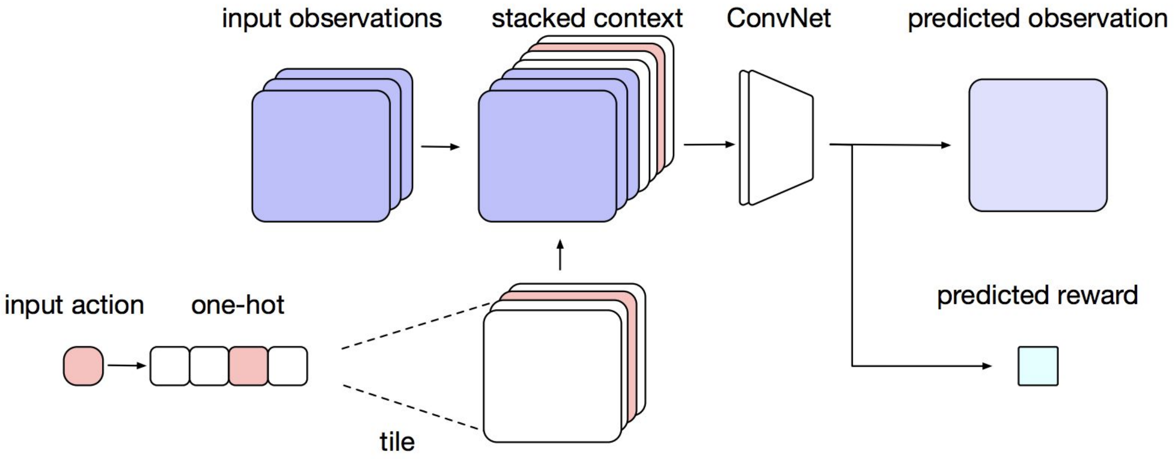



$$\boxed{\hat{\rho}(s_{t+1} | s_t, a_t)}$$

+ planning

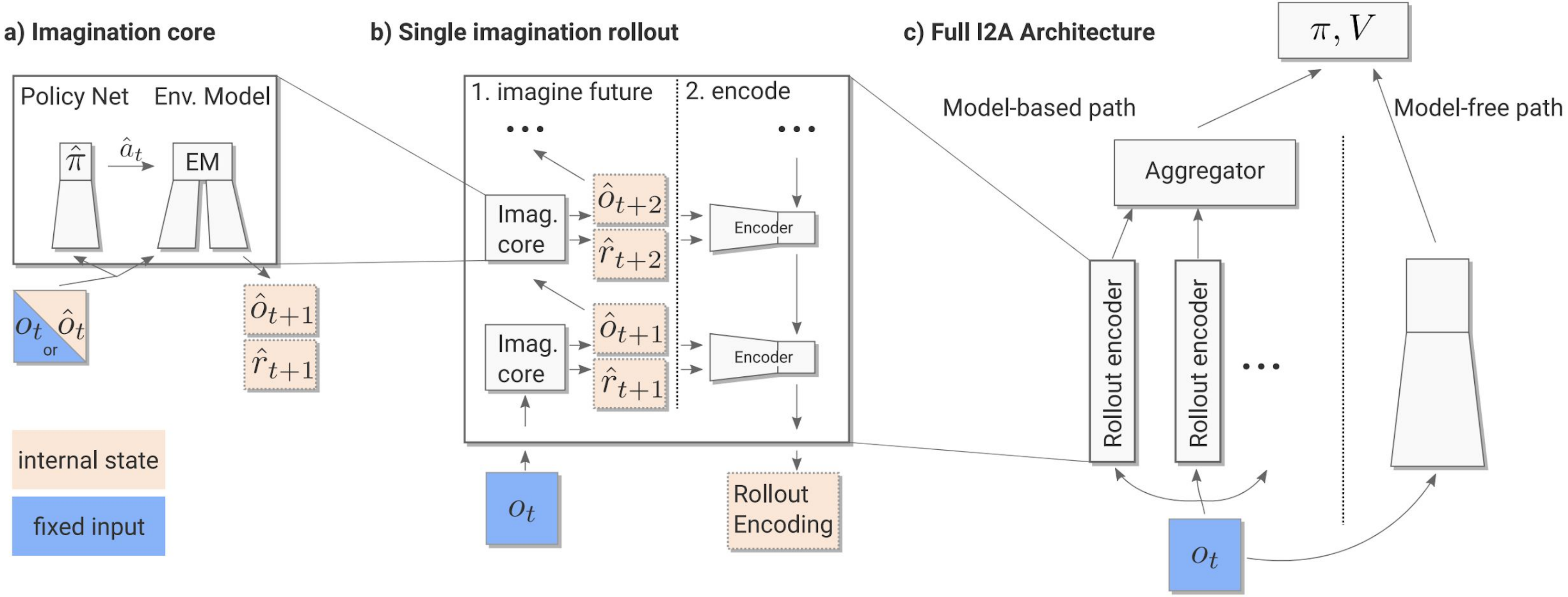
# Merging model-based and model-free approaches





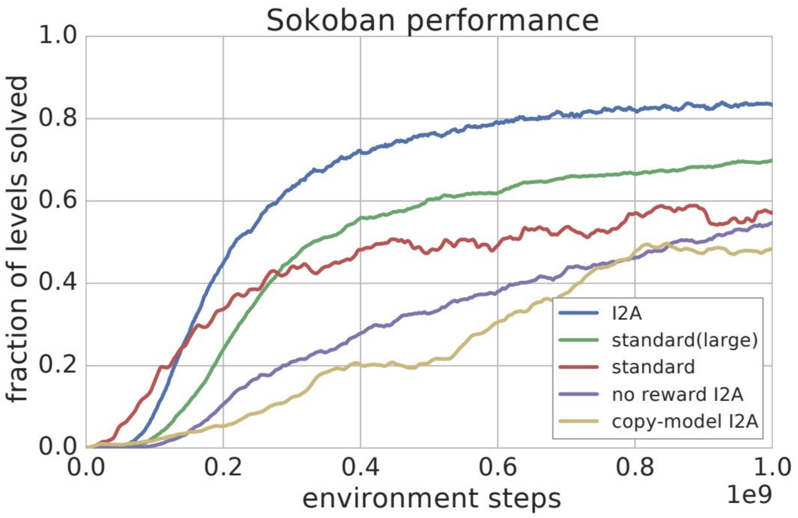

Weber et al., arXiv 2018

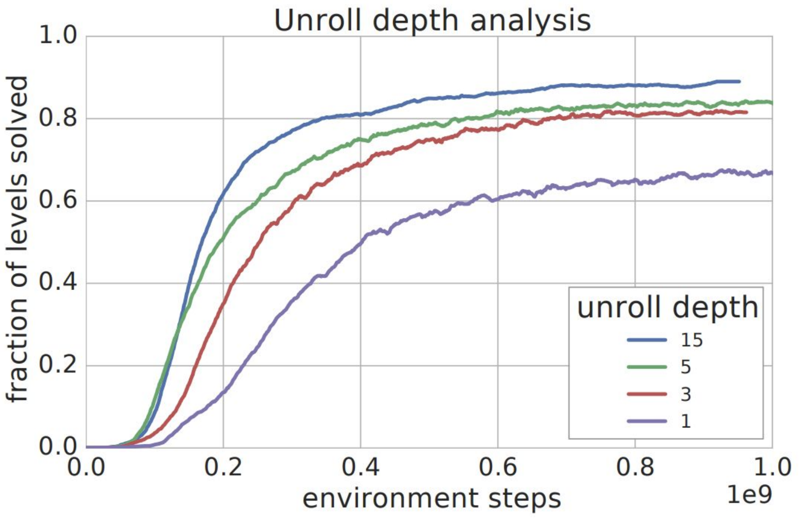
# Merging model-based and model-free approaches




Weber et al., arXiv 2018

# Merging model-based and model-free approaches




Weber et al., arXiv 2018

# Questions?