REINFORCEMENT LEARNING

TODAY'S CLASS

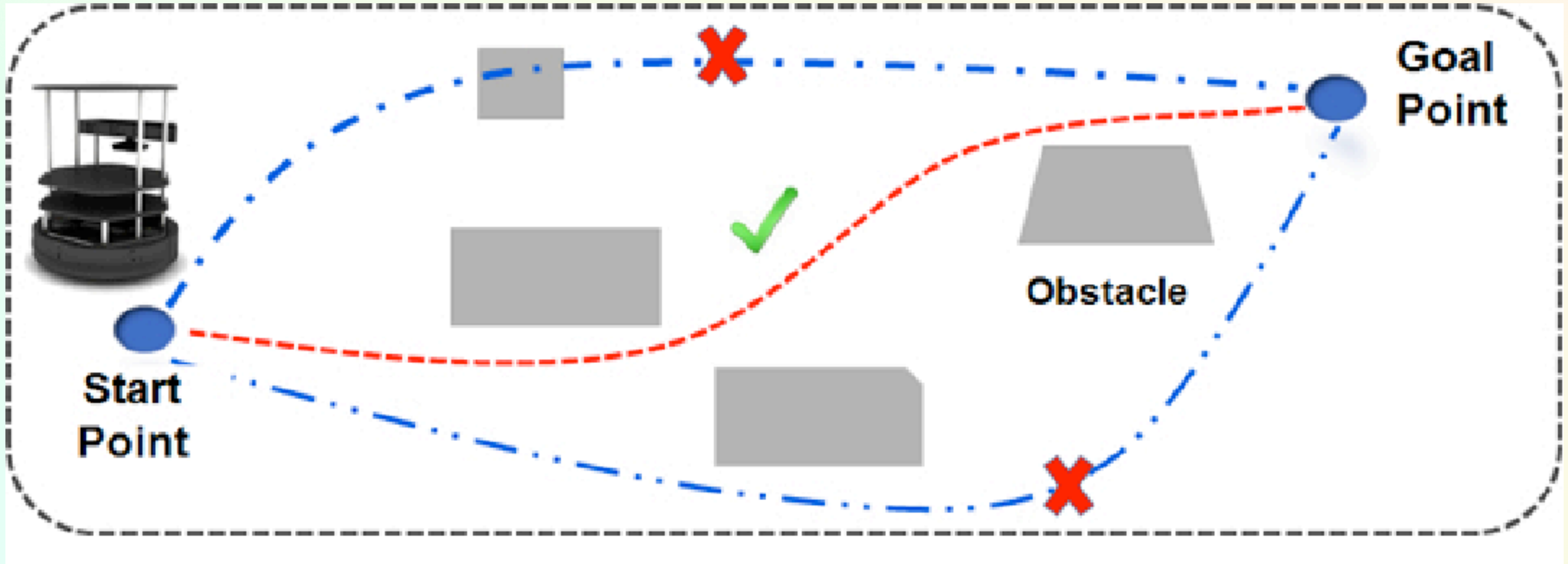
GOALS

1. Define Sequential Decision Making Problems
2. Imitation Learning (copying an expert)
3. Reinforcement Learning
4. Policy Gradient Methods

DECISION MAKING PROBLEMS

EXAMPLES

Robot navigation

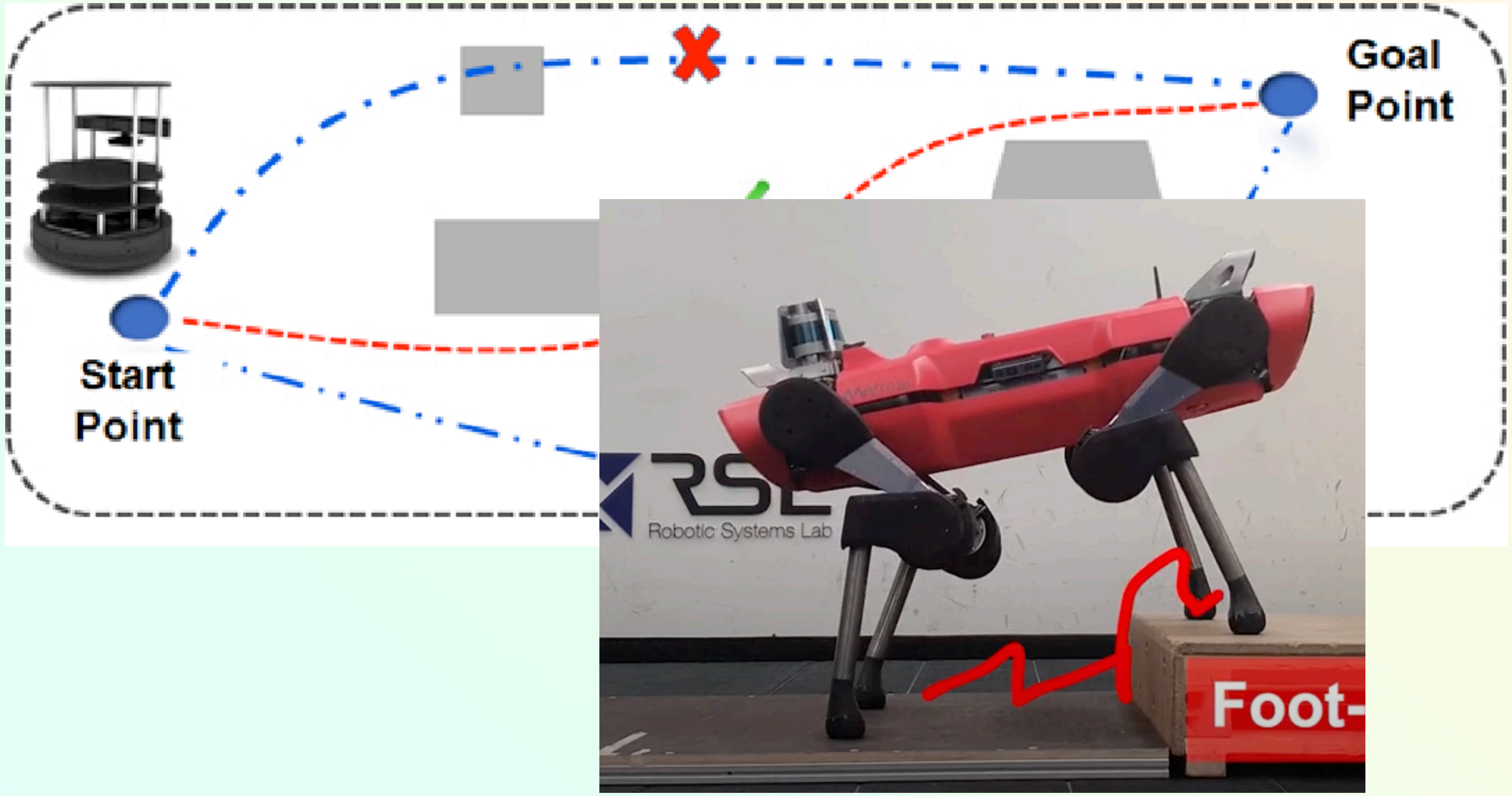


DECISION MAKING PROBLEMS

EXAMPLES

Robot navigation

Robot locomotion



https://www.youtube.com/watch?v=9j2a1oAHDL8

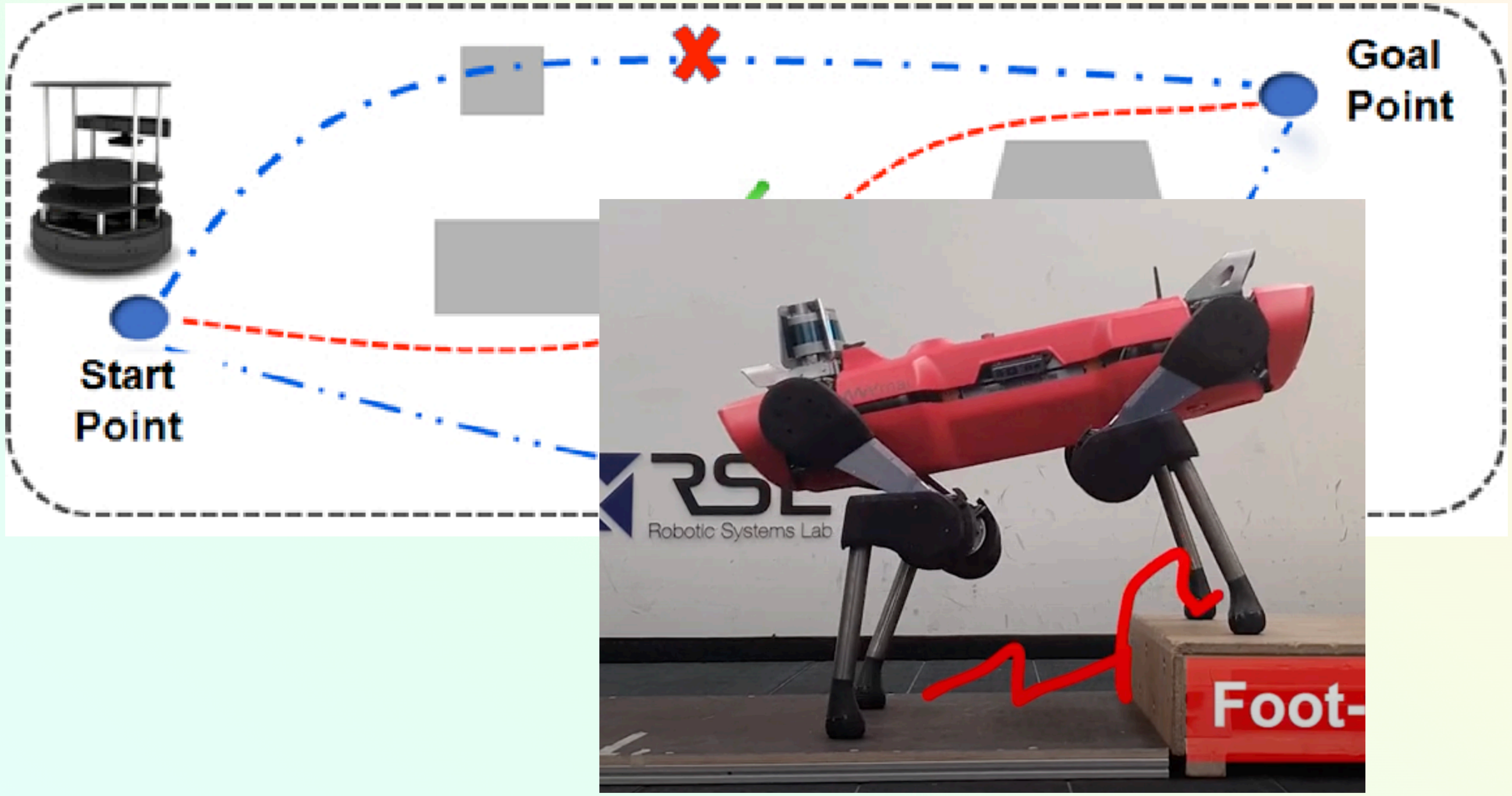
DECISION MAKING PROBLEMS

EXAMPLES

Robot navigation

Robot locomotion

Recommendation (Ads, YouTube, etc)



DECISION MAKING PROBLEMS

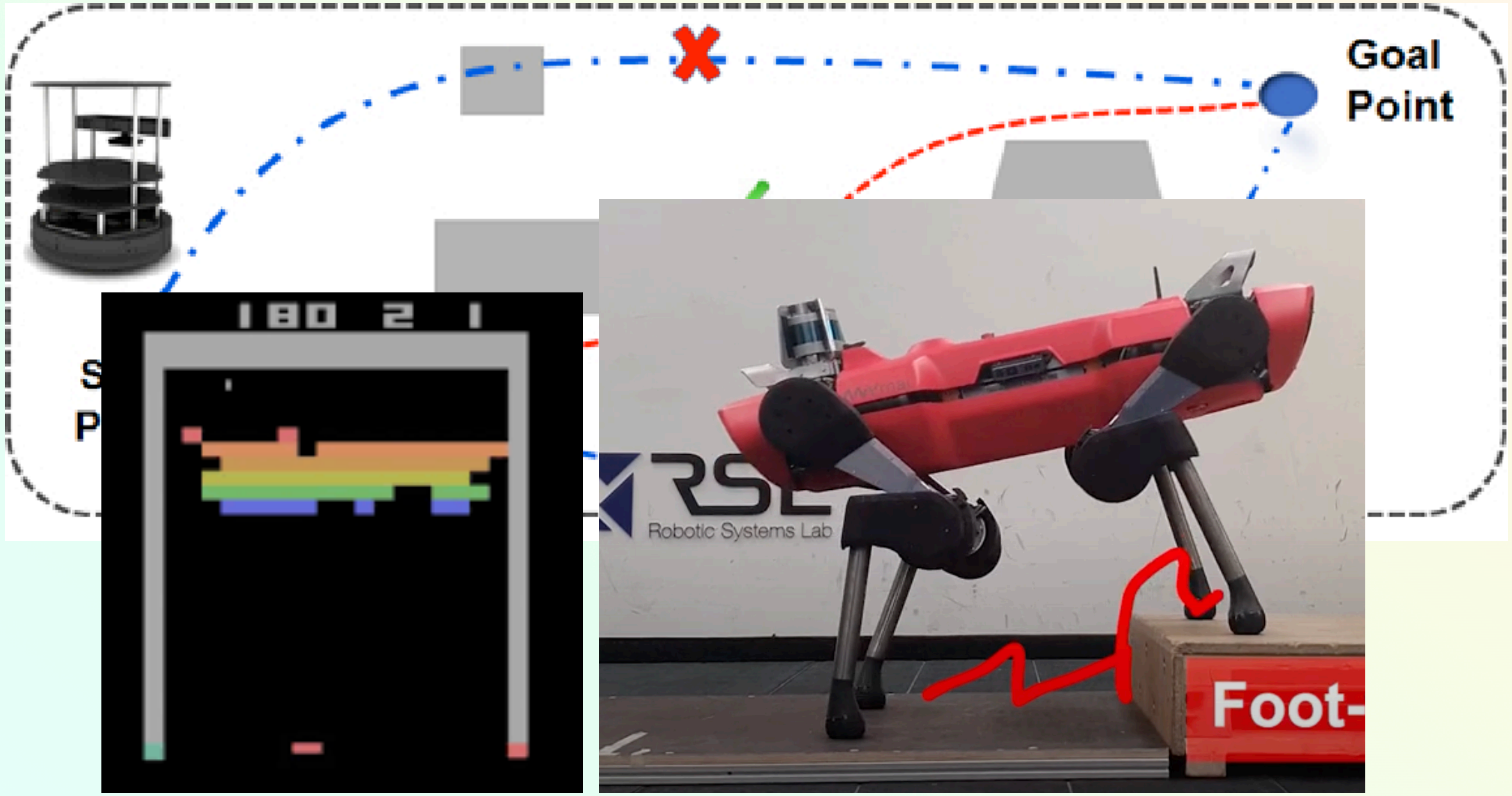
EXAMPLES

Robot navigation

Robot locomotion

Recommendation (Ads, YouTube, etc)

Game Playing



DECISION MAKING PROBLEMS

EXAMPLES

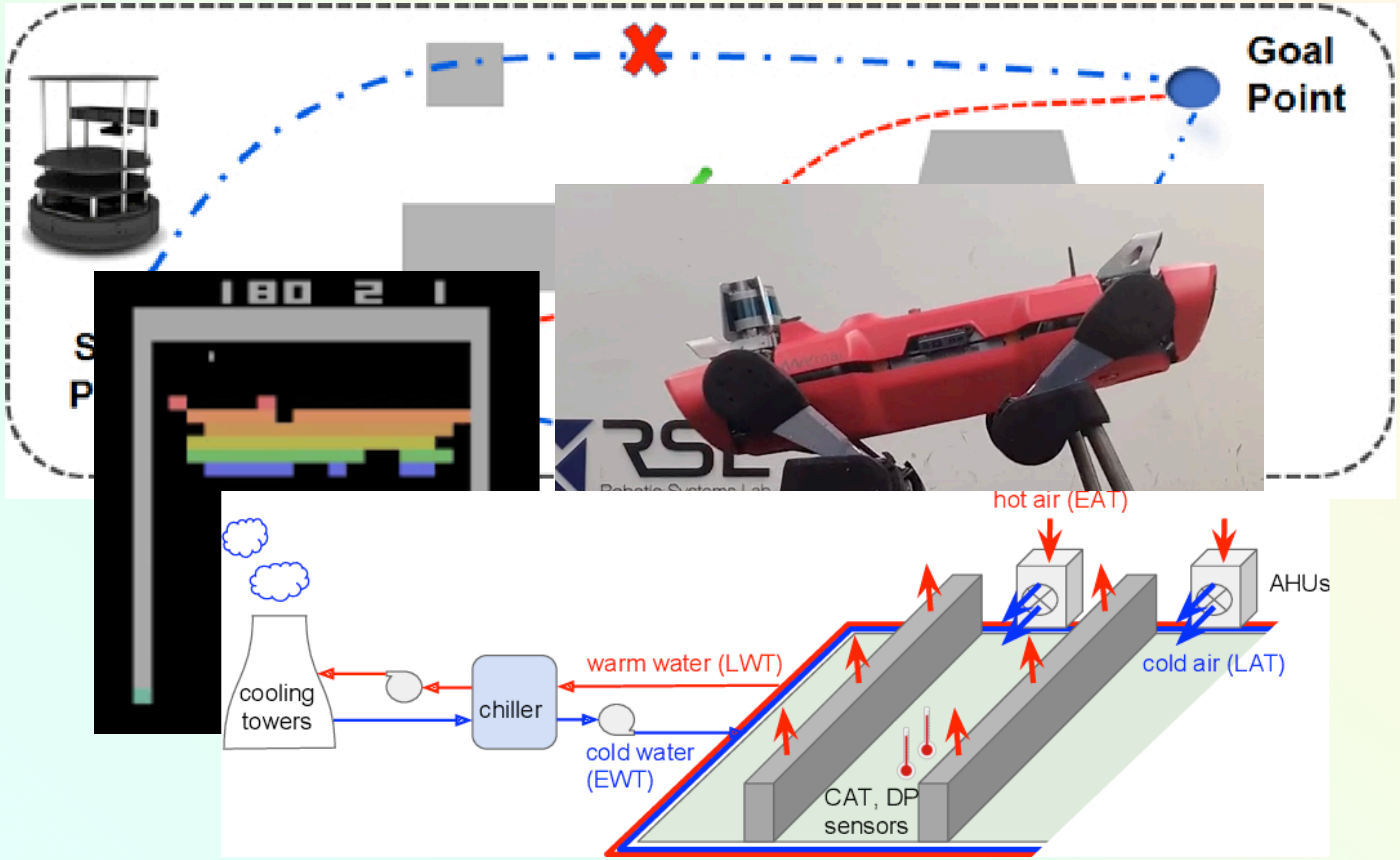
Robot navigation

Robot locomotion

Recommendation (Ads, YouTube, etc)

Game Playing

Data Center Cooling



Lazic, N., Boutilier, C., Lu, T., Wong, E., Roy, B., Ryu, M., & Imwalle, G.P. (2018). Data center cooling using model-predictive control. Neural Information Processing Systems.

DECISION MAKING PROBLEMS

EXAMPLES

Robot navigation

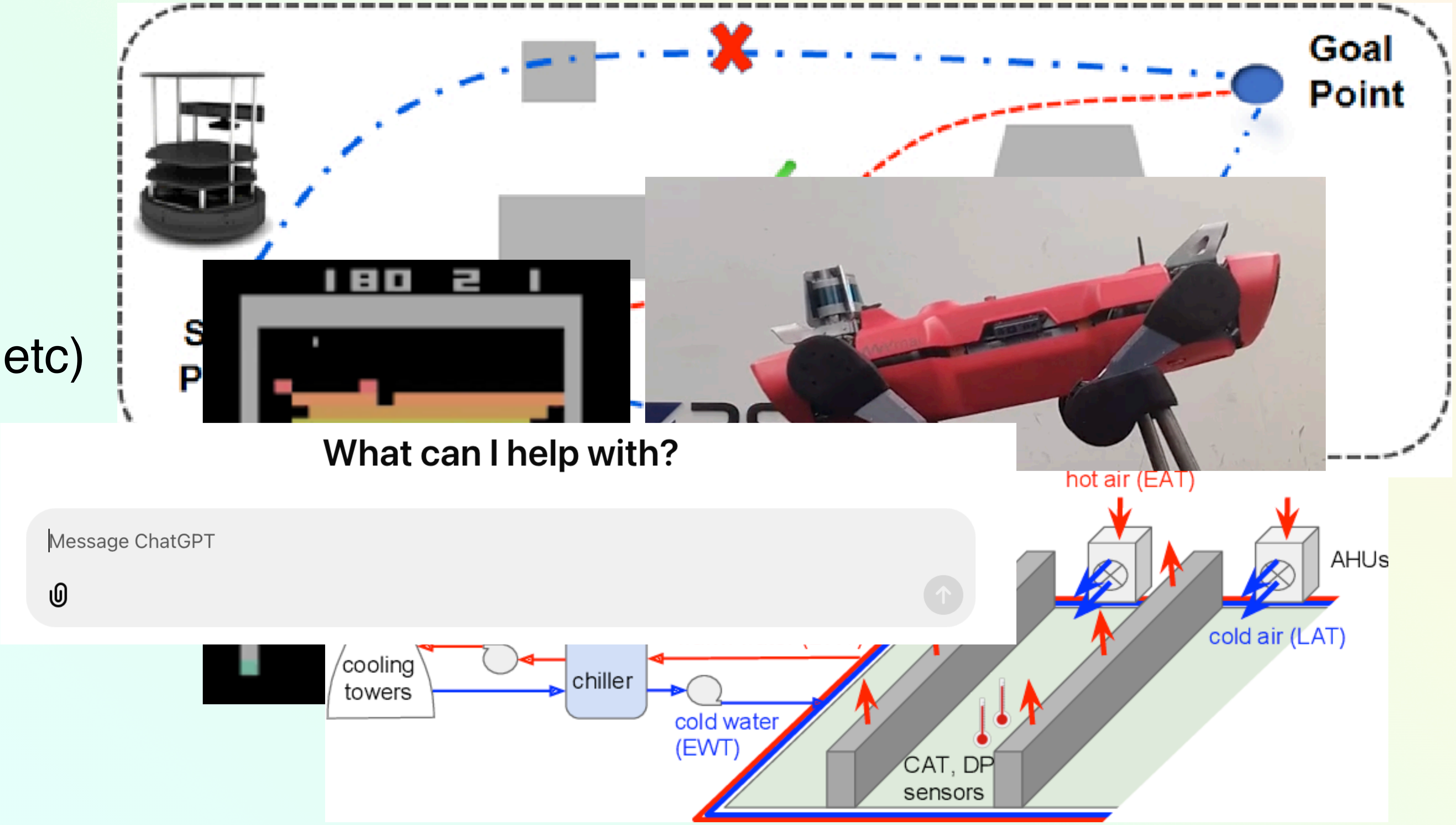
Robot locomotion

Recommendation (Ads, YouTube, etc)

Game Playing

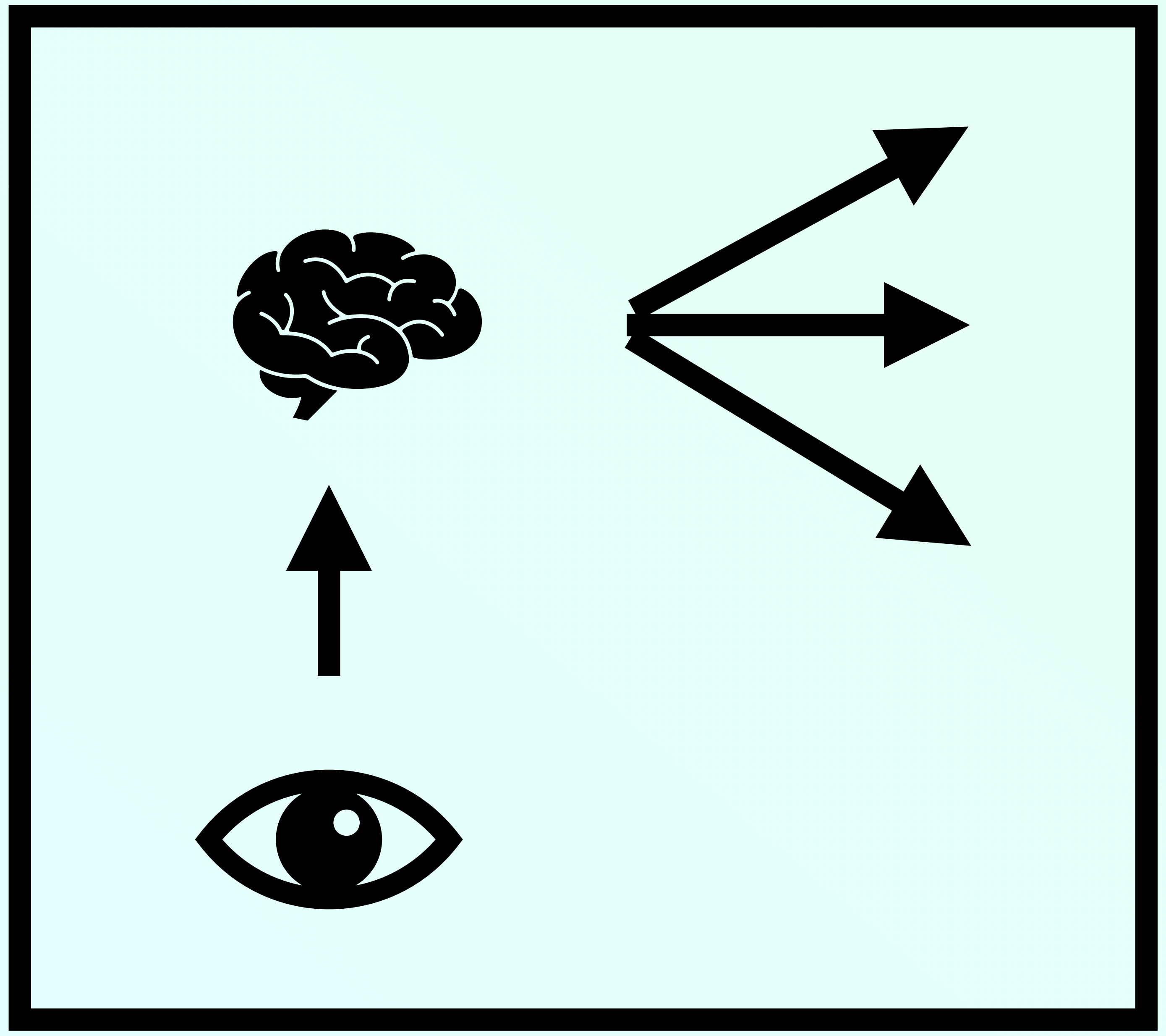
Data Center Cooling

Chatbots/LLMs



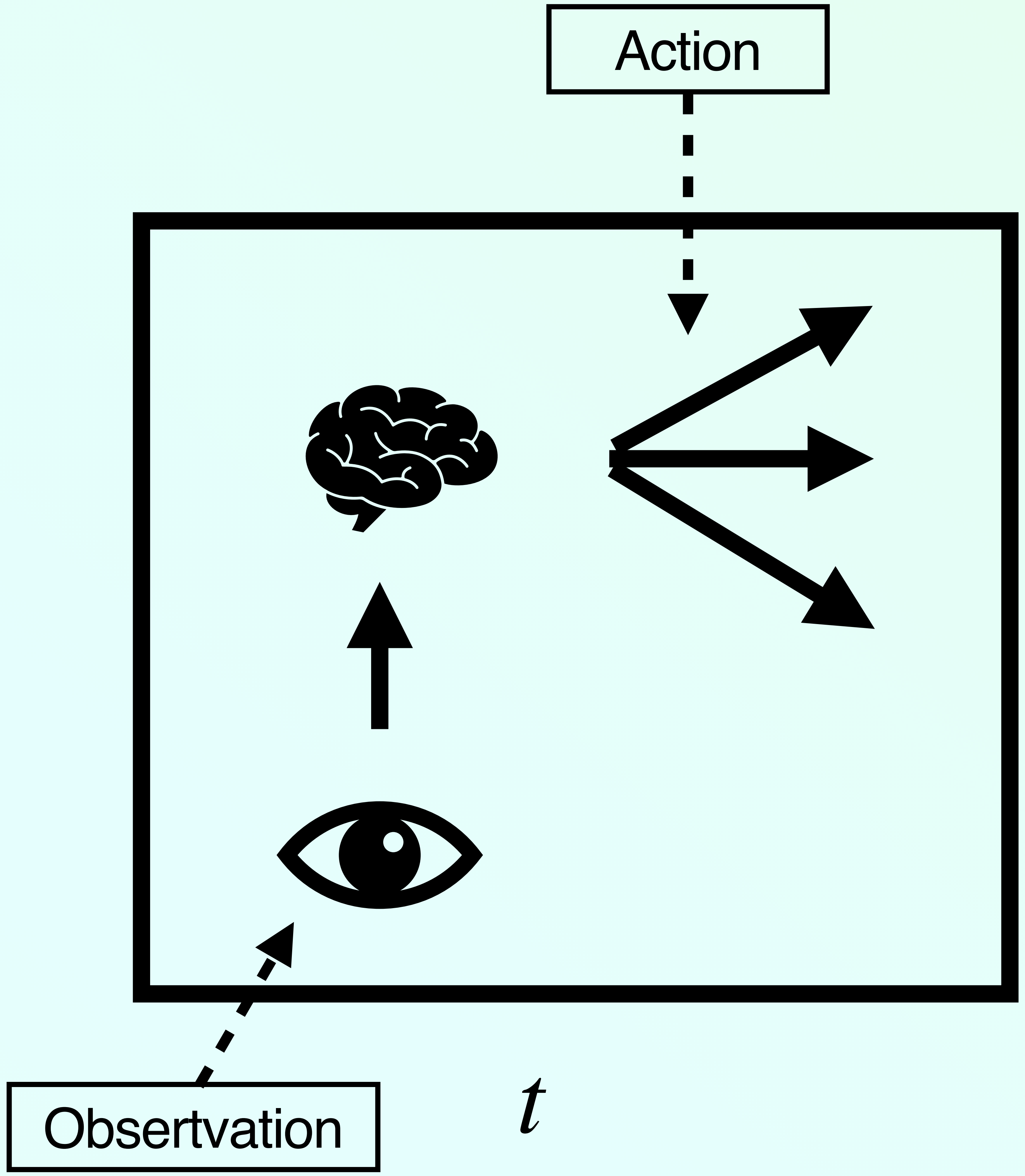
https://chatgpt.com/

SEQUENTIAL DECISION MAKING

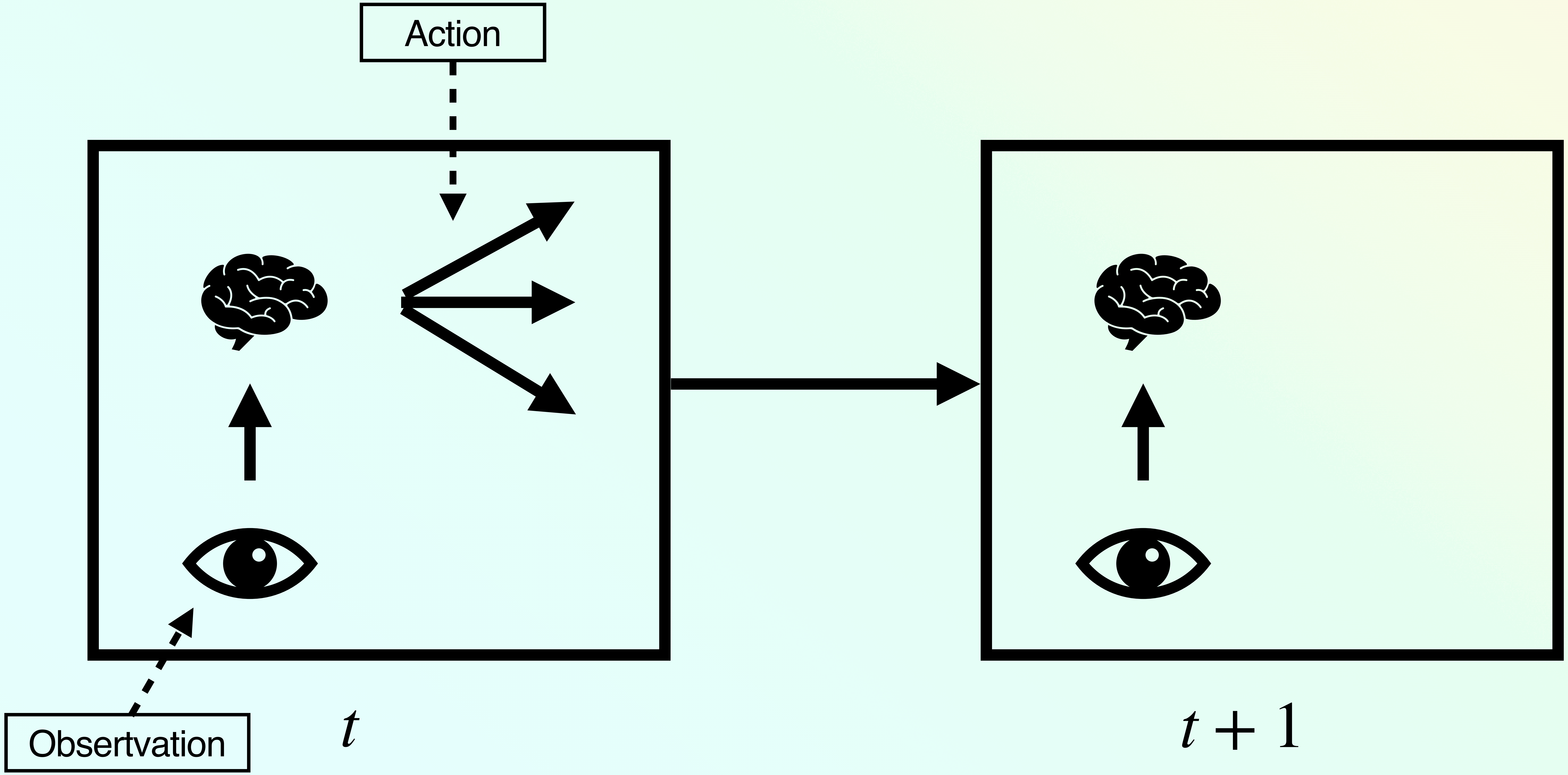


t

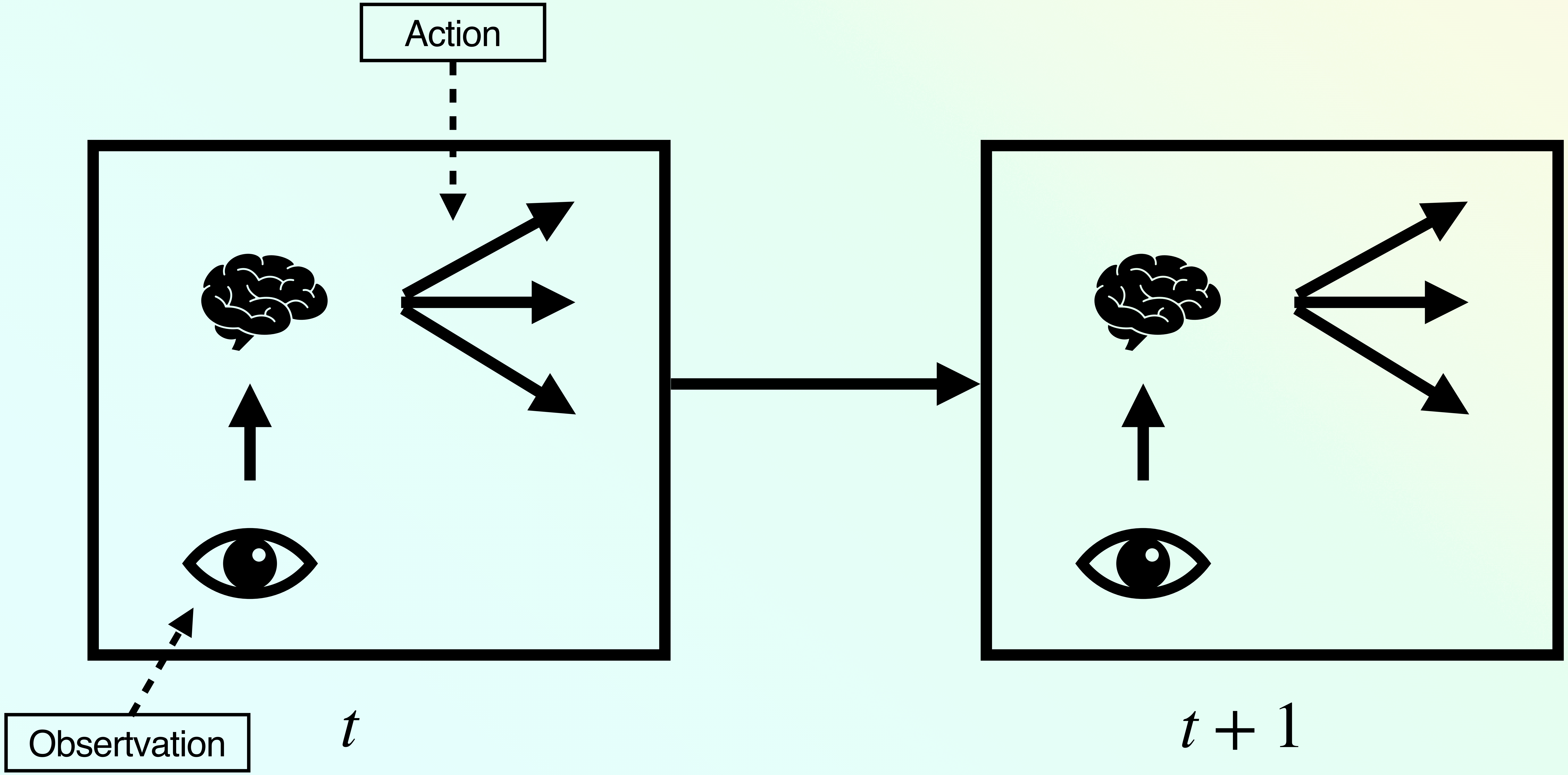
SEQUENTIAL DECISION MAKING



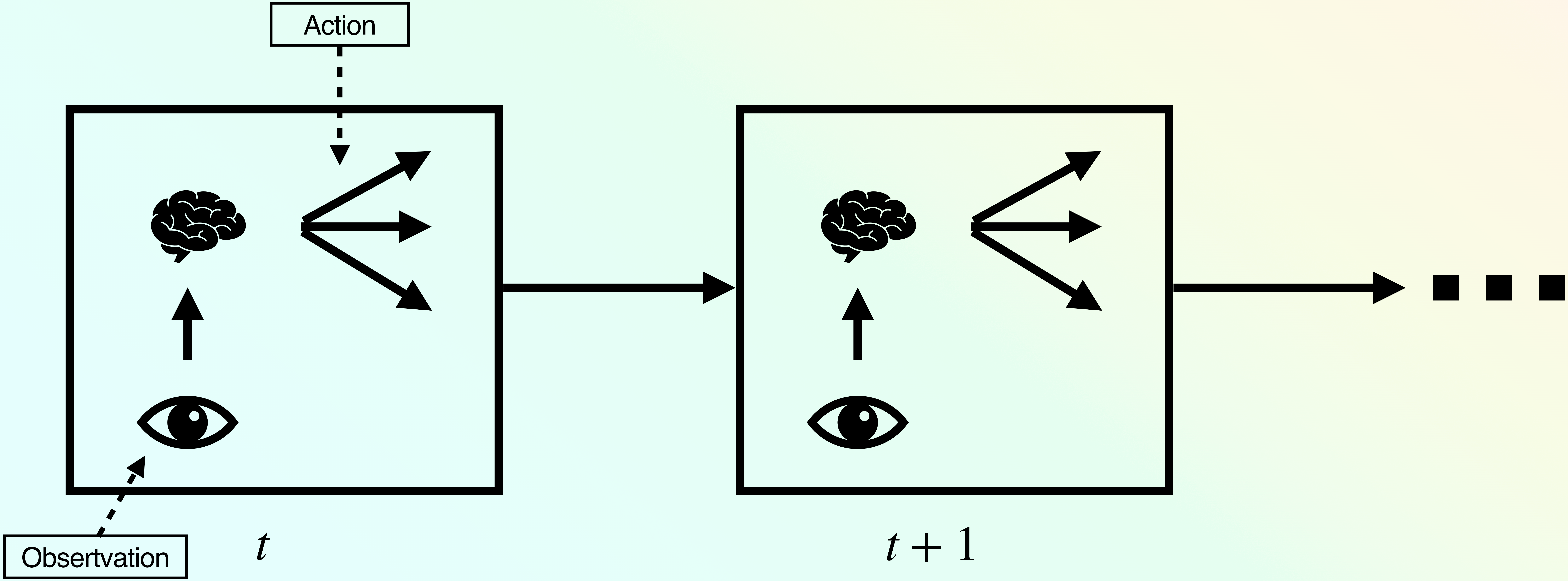
SEQUENTIAL DECISION MAKING



SEQUENTIAL DECISION MAKING



SEQUENTIAL DECISION MAKING



SEQUENTIAL DECISION MAKING

PROBLEM FORMULATION (IMITATION LEARNING)

Can we frame this problem as a function approximation problem?

X_t — observation at time t

A_t — action (agent's decision) at time t

$f_*(X_t)$ — expert's probability distribution over actions given X_t

$f(X_t)$ — agent's probability distribution over actions given X_t

$$\arg\min_f \mathbf{E}\left[\sum_{t=1}^{\infty} d\left(f(X_t), f_*(X_t)\right)\right]$$

SEQUENTIAL DECISION MAKING

PROBLEM FORMULATION (IMITATION LEARNING)

Collect data from an expert:

$$D = \{x_1, x_2, \cdots, x_m\}, \{a_1, a_2, \cdots, a_m\}$$

Classification (or regression) to fit f to D

$$l_D(\theta) = -\sum_{i=1}^{m} \ln \Pr(A_t = a_i \,|\, X_t = x_i) = -\sum_{i=1}^{m} \ln f(x_i, \theta)_{a_i}$$

IMITATI

PROBLEM FORMUL

https://www.youtube.com/watch?v=anOI0xZ3kGM

IMITATION LEARNING

PROPERTIES

- Approximation has errors that does not match the expert
  - Errors lead to new observations, not in the data set

IMITATION LEARNING

PROPERTIES

- Approximation has errors that does not match the expert
  - Errors lead to new observations, not in the data set
- Assume perfect approximation on D
  - The environment may have noise (cannot exactly replicate the same sequences in D)
  - Tiny changes in observation may lead to new states

IMITATION LEARNING

PROPERTIES

- Imitation needs:
  - Lots of expert data
  - Collect expert decisions from observation that an expert won't see, but the agent might

IMITATION LEARNING

PROPERTIES

- Imitation needs:
  - Lots of expert data
  - Collect expert decisions from observation that an expert won't see, but the agent might

An agent can only be as good as an expert!

It is not always possible to have a good enough expert!

REINFORCEMENT LEARNING

DESIRED PROPERTIES FOR DECISION-MAKING AGENT

The agent should learn from its interactions with the environment

— collect its own data (not always necessary)

The agent needs to determine what is the best action

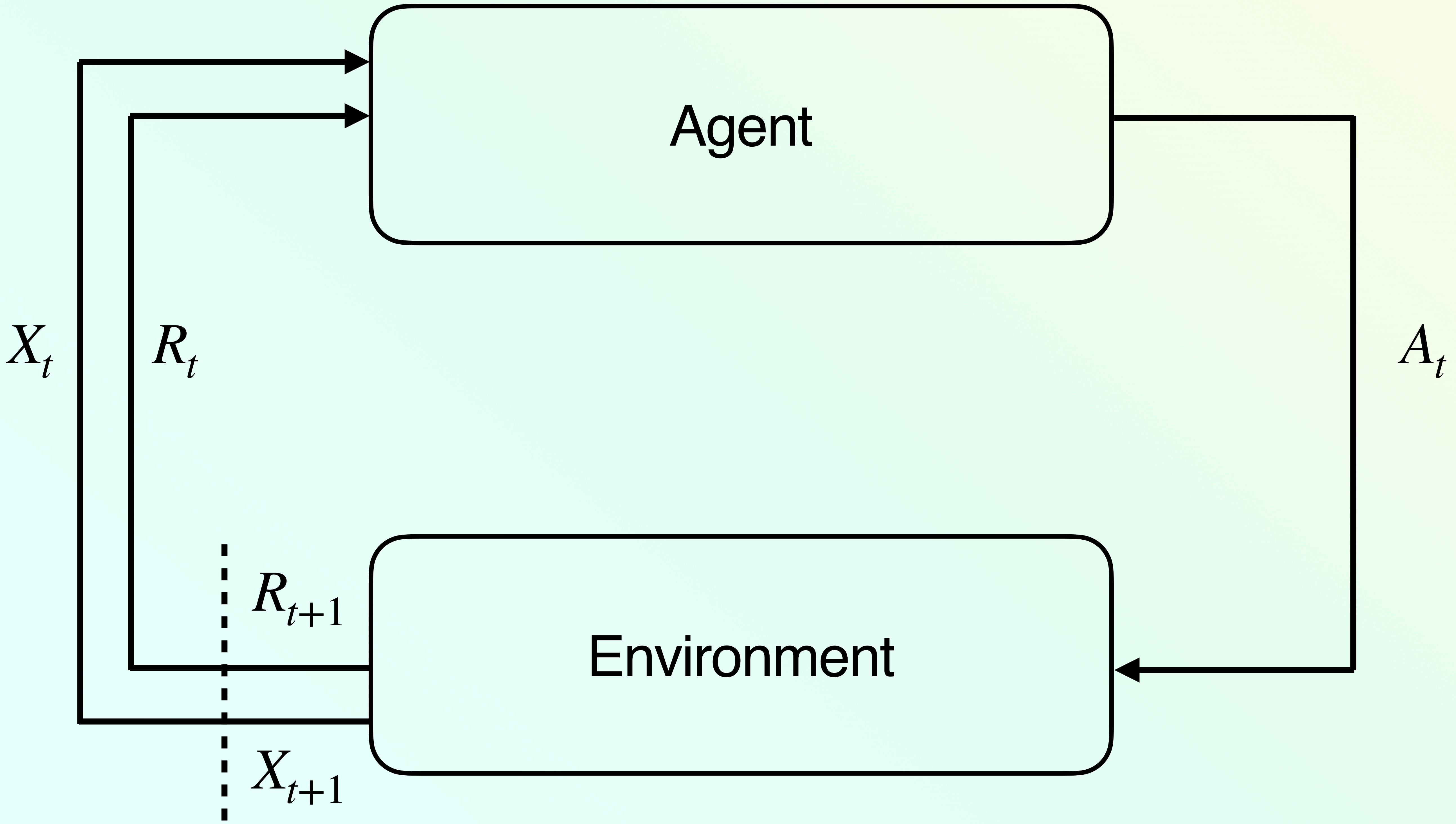
— Agent has to search for the best action (we cannot tell it what to do)

The agent needs to know how good its decision was.

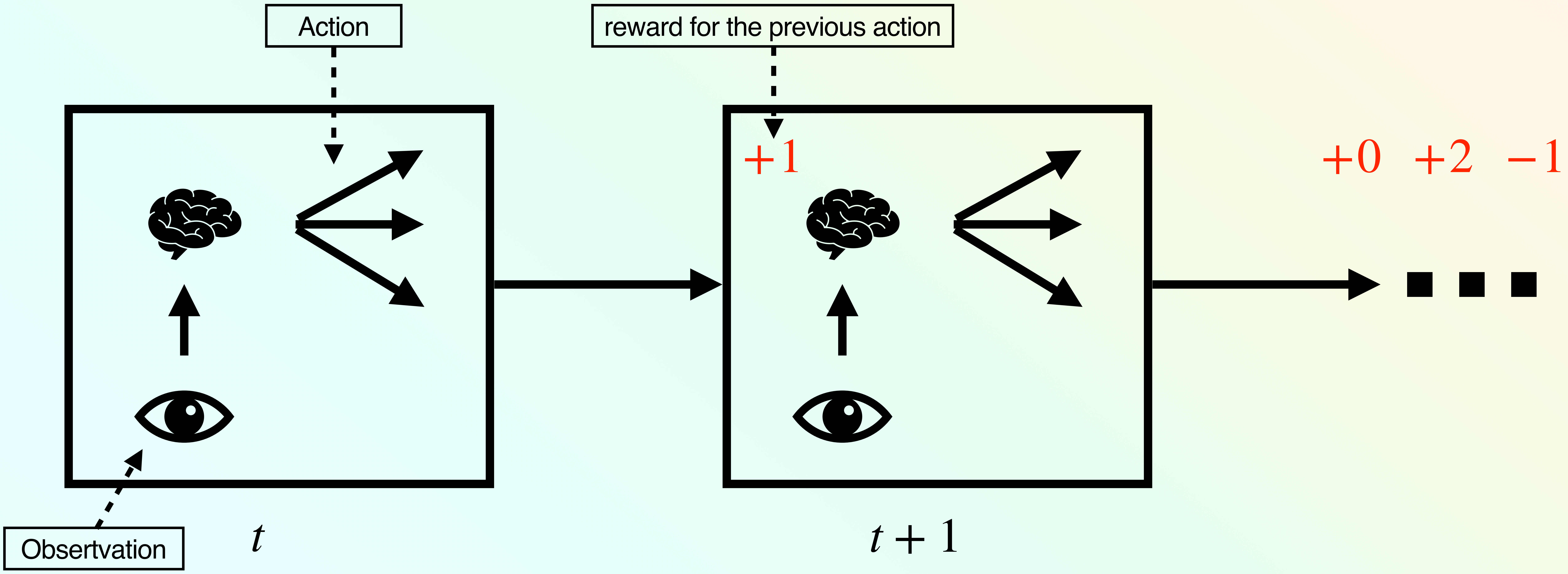
— Provide a score (reward) indicating the quality of decisions

REINFORCEMENT LEARNING

AGENT-ENVIRONMENT INTERACTION



SEQUENTIAL DECISION MAKING



SEQUENTIAL DECISION MAKING

REINFORCEMENT LEARNING

OBJECTIVE

Finite Episodic Problem:

The agent interacts with the environment for T time steps

The initial configuration of the environment is sampled from some distribution d_0

$$d_0(x) = \Pr(X_1 = x)$$

Agent receives a reward R_{t+1} for taking action A_t in X_t

After T time steps, the environment resets to an initial configuration

We call one sequence of time steps $t = 1$ to $t = T$ an *episode*

REINFORCEMENT LEARNING

OBJECTIVE

Agent samples actions from $f(X_t, \theta)$, i.e., $A_t \sim f(X_t, \theta)$

The objective function (expectation of the sum of all rewards):

$$\rho(\theta) \doteq \mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right]$$

Randomness comes from initial state X_1 and randomness in actions A_t

REINFORCEMENT LEARNING

OBJECTIVE

Agent samples actions from $f(X_t, \theta)$, i.e., $A_t \sim f(X_t, \theta)$ — call f a *policy*

The objective function (expectation of the sum of all rewards):

$$\rho(\theta) \doteq \mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right]$$

Randomness comes from initial state X_1 and randomness in actions A_t

The agent's goal is to find parameters θ that maximize ρ

$$\arg\max_{\theta} \rho(\theta)$$

RL PROBLEMS

EXAMPLES

Maximize score

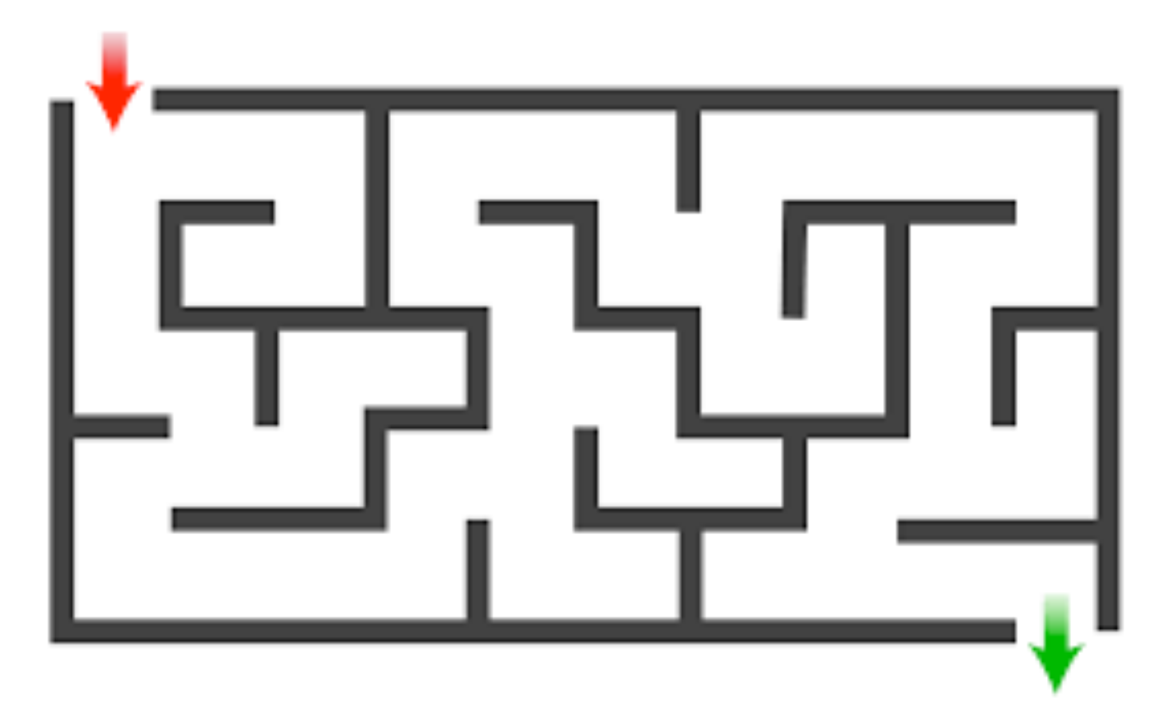
$X_t =$ image of the game

A_t = controller movement

R_{t+1} = points for breaking bricks

RL PROBLEMS

EXAMPLES



Get out the maze as fast as possible

$X_t =$ Position in the maze

A_t = direction to move

$R_{t+1} = -1$

Agent penalized for being in the maze every time step

RL PROBLEMS

EXAMPLES

LLM

$X_t =$ token sequence/context up to point t

A_t = next token

$R_{t+1} = ?$ – 0 until the end of the episode

R_{T+1} – score representing the quality of generated sequence

Human prefered output

Make LLM output human prefer text

Many reward/scoring functions people use

RL VS SUPERVISED LEARNING

SOME DIFFERENCES

Supervised Learning:

- Instructive feedback — predict this label
- Assumes training data comes from the same distribution that the model will be used in

Reinforcement learning:

- Evaluative feedback — how good was the decision (not what was the best decision)
  - — This makes RL have to search for good decisions through trial and error
- Data distribution changes — improving the decisions will change the observations

QUIZ

REINFORCEMENT LEARNING

OPTIMIZATION

$$\rho(\theta) \doteq \mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right]$$

Gradient ascent:

$$\theta \leftarrow \theta + \eta \nabla \rho(\theta)$$

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \frac{\partial}{\partial \theta} \mathbf{E} \left[\sum_{t=1}^{T} R_{t+1} \right]$$

POLICY GRADIENT

EXPRESSION

$$\rho(\theta) = \mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right]$$

$$G_t = \sum_{k=0}^{T-t} R_{t+k+1}$$

$$H_{1:T} = \{X_1, A_1, X_2, R_2, A_2, \cdots, X_T, A_T, R_{T+1}\}$$

$$\tau_{1:T} = \{x_1, a_1, x_2, r_2, \cdots, x_T, a_T, r_{T+1}\}$$

POLICY GRADIENT

EXPRESSION

$$\rho(\theta) = \mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right] = \mathbf{E}\left[G_1\right] = \sum_{\tau} \Pr(H_{1:T} = \tau_{1:T}) G_1$$

$$G_t = \sum_{k=0}^{T-t} R_{t+k+1}$$

$$H_{1:T} = \{X_1, A_1, X_2, R_2, A_2, \cdots, X_T, A_T, R_{T+1}\}$$

$$\tau_{1:T} = \{x_1, a_1, x_2, r_2, \cdots, x_T, a_T, r_{T+1}\}$$

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \frac{\partial}{\partial \theta} \mathbf{E} \left[\sum_{t=1}^{T} R_{t+1} \right] = \frac{\partial}{\partial \theta} \mathbf{E} \left[G_1 \right] = \frac{\partial}{\partial \theta} \sum_{\tau} \Pr \left(H_{1:T} = \tau \right) G_1$$

POLICY GRADIENT

EXPRESSION

$$
\begin{aligned}
\nabla \rho(\theta) =\ & \frac{\partial}{\partial \theta} \mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right] = \frac{\partial}{\partial \theta} \mathbf{E}\left[G_1\right] = \frac{\partial}{\partial \theta} \sum_{\tau} \Pr\left(H_{1:T} = \tau\right) G_1 \\
=\ & \sum_{\tau} \frac{\partial \Pr\left(H_{1:T} = \tau\right) G_1}{\partial \theta} = \sum_{\tau} \left(\frac{\partial \Pr\left(H_{1:T} = \tau\right)}{\partial \theta} G_1 + \frac{\partial G_1}{\partial \theta} \Pr\left(H_{1:T} = \tau\right) \right) \\
=\ & \sum_{\tau} \frac{\partial \Pr\left(H_{1:T} = \tau\right)}{\partial \theta} G_1
\end{aligned}
$$

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \frac{\partial}{\partial\theta}\mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right] = \frac{\partial}{\partial\theta}\mathbf{E}\left[G_1\right] = \frac{\partial}{\partial\theta}\sum_{\tau}\Pr\left(H_{1:T}=\tau\right)G_1$$

$$= \sum_{\tau}\frac{\partial \Pr\left(H_{1:T}=\tau\right)G_1}{\partial\theta} = \sum_{\tau}\left(\frac{\partial \Pr\left(H_{1:T}=\tau\right)}{\partial\theta}G_1 + \frac{\partial G_1}{\partial\theta}\Pr\left(H_{1:T}=\tau\right)\right)$$

$$= \sum_{\tau}\frac{\partial \Pr\left(H_{1:T}=\tau\right)}{\partial\theta}G_1$$

$$= \sum_{\tau}\Pr\left(H_{1:T}=\tau\right)G_1\frac{\partial \ln \Pr\left(H_{1:T}=\tau\right)}{\partial\theta}$$

$$\frac{d}{dx}p(x) = p(x)\frac{d}{dx}\ln p(x)$$

POLICY GRADIENT

EXPRESSION

$$
\begin{aligned}
\nabla \rho(\theta) &= \frac{\partial}{\partial \theta}\mathbf{E}\left[\sum_{t=1}^{T} R_{t+1}\right] = \frac{\partial}{\partial \theta}\mathbf{E}\left[G_1\right] = \frac{\partial}{\partial \theta}\sum_{\tau} \Pr\left(H_{1:T} = \tau\right) G_1 \\
&= \sum_{\tau} \frac{\partial \Pr\left(H_{1:T} = \tau\right) G_1}{\partial \theta} = \sum_{\tau}\left(\frac{\partial \Pr\left(H_{1:T} = \tau\right)}{\partial \theta} G_1 + \frac{\partial G_1}{\partial \theta}\Pr\left(H_{1:T} = \tau\right)\right) \\
&= \sum_{\tau} \frac{\partial \Pr\left(H_{1:T} = \tau\right)}{\partial \theta} G_1 \\
&= \sum_{\tau} \Pr\left(H_{1:T} = \tau\right) G_1 \frac{\partial \ln \Pr\left(H_{1:T} = \tau\right)}{\partial \theta} \\
&= \mathbf{E}\left[G_1 \frac{\partial \ln \Pr\left(H_{1:T} = \tau\right)}{\partial \theta}\right]
\end{aligned}
$$

$$
\frac{d}{dx}p(x) = p(x)\frac{d}{dx}\ln p(x)
$$

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \mathbf{E}\left[G_1 \frac{\partial \ln \Pr\left(H_{1:T}\right)}{\partial \theta}\right]$$

G_1 — How good was the episode $H_{1:T}$

$\frac{\partial \ln \Pr(H_{1:T})}{\partial \theta}$ — direction to change θ to make the episode $H_{1:T}$ more likely

$\nabla \rho(\theta)$ — makes episodes more likely proportionally to how good they are

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \mathbf{E}\left[G_1 \frac{\partial \ln \Pr\left(H_{1:T}\right)}{\partial \theta}\right]$$

G_1 — How good was the episode $H_{1:T}$

$\dfrac{\partial \ln \Pr(H_{1:T})}{\partial \theta}$ — direction to change θ to make the episode $H_{1:T}$ more likely

$\nabla \rho(\theta)$ — makes episodes more likely proportionally to how good they are

$\nabla \rho(\theta) = \mathbf{E}\left[(G_1 - \rho(\theta)) \dfrac{\partial \ln \Pr\left(H_{1:T}\right)}{\partial \theta}\right]$ — make episodes that are better than average more likely

QUIZ

POLICY DERIVATIVE

MAKE EPISODE MORE LIKELY

$$\frac{\partial \ln \Pr\left(H_{1:T} = \tau\right)}{\partial \theta}$$

$f(X_t, \theta)$ controls action distribution

Different actions control the rewards and the subsequent observations X_{t+1}

$$p(x, a, x', r) \doteq \Pr(X_{t+1} = x', R_{t+1} = r \,|\, X_t = x, A_t = a)$$

POLICY DERIVATIVE

MAKE EPISODE MORE LIKELY

$$\frac{\partial \ln \Pr\left(H_{1:T} = \tau\right)}{\partial \theta}$$

$f(X_t, \theta)$ controls action distribution

Different actions control the rewards and the subsequent observations X_{t+1}

$$p(x, a, x', r) \doteq \Pr(X_{t+1} = x', R_{t+1} = r \,|\, X_t = x, A_t = a)$$

Assume

$$\Pr(X_{t+1} = x', R_{t+1} = r \,|\, H_{1:t} = \tau_{1:t}) = \Pr(X_{t+1} = x', R_{t+1} = r \,|\, X_t = x_t, A_t = a_t)$$

EPISODE PROBABILITIES

$$\Pr(H_{1:T} = \tau_{1:T}) = d_0(x_1)f(x_1, \theta)_{a_1}p(x_1, a_1, x_2, r_2)f(x_2, \theta)_{a_2}p(x_2, a_2, x_3, r_3)\cdots p(x_T, a_T, x_{T+1}, r_{T+1})$$

$$= d_0(x_1)\prod_{t=1}^{T} f(x_t, \theta)_{a_t}p(x_t, a_t, x_{t+1}, r_{t+1})$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1) = \Pr(A_1 = a_1 \,|\, X_1 = x_1) \Pr(X_1 = x_1)$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1) = \Pr(A_1 = a_1 | X_1 = x_1) \Pr(X_1 = x_1) = f(x_1, \theta)_{a_1} d_0(x_1)$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1) = \Pr(A_1 = a_1 \,|\, X_1 = x_1) \Pr(X_1 = x_1) = f(x_1, \theta)_{a_1} d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1, X_2 = x_2, R_2 = r_2) = \Pr(X_2 = x_2, R_2 = r_2 \,|\, X_1 = x_1, A_1 = a_1) \Pr(X_1 = x_1, A_1 = a_1)$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1) = \Pr(A_1 = a_1 \mid X_1 = x_1)\Pr(X_1 = x_1) = f(x_1, \theta)_{a_1} d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1, X_2 = x_2, R_2 = r_2) = \Pr(X_2 = x_2, R_2 = r_2 \mid X_1 = x_1, A_1 = a_1)\Pr(X_1 = x_1, A_1 = a_1)$$

$$= p(x_1, a_1, x_2, r_2) f(x_1, \theta)_{a_1} d_0(x_1)$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1) = \Pr(A_1 = a_1 \,|\, X_1 = x_1)\Pr(X_1 = x_1) = f(x_1, \theta)_{a_1} d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1, X_2 = x_2, R_2 = r_2) = \Pr(X_2 = x_2, R_2 = r_2 \,|\, X_1 = x_1, A_1 = a_1)\Pr(X_1 = x_1, A_1 = a_1)$$

$$= p(x_1, a_1, x_2, r_2) f(x_1, \theta)_{a_1} d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1, X_2 = x_2, R_2 = r_2, A_2 = a_2) = \Pr(A_2 = a_2 \,|\, X_2 = x_2)\Pr(X_2 = x_2, R_2 = r_2, X_1 = x_1, A_1 = a_1)$$

EPISODE PROBABILITIES

$$\Pr(X_1 = x_1) = d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1) = \Pr(A_1 = a_1 \,|\, X_1 = x_1)\Pr(X_1 = x_1) = f(x_1, \theta)_{a_1} d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1, X_2 = x_2, R_2 = r_2) = \Pr(X_2 = x_2, R_2 = r_2 \,|\, X_1 = x_1, A_1 = a_1)\Pr(X_1 = x_1, A_1 = a_1)$$

$$= p(x_1, a_1, x_2, r_2) f(x_1, \theta)_{a_1} d_0(x_1)$$

$$\Pr(X_1 = x_1, A_1 = a_1, X_2 = x_2, R_2 = r_2, A_2 = a_2) = \Pr(A_2 = a_2 \,|\, X_2 = x_2)\Pr(X_2 = x_2, R_2 = r_2, X_1 = x_1, A_1 = a_1)$$

$$= f(x_2, \theta)_{a_2} p(x_1, a_1, x_2, r_2) f(x_1, \theta)_{a_1} d_0(x_1)$$

EPISODE PROBABILITIES

$$\Pr(H_{1:T} = \tau_{1:T}) = d_0(x_1) f(x_1, \theta)_{a_1} p(x_1, a_1, x_2, r_2) f(x_2, \theta)_{a_2} p(x_2, a_2, x_3, r_3) \cdots p(x_T, a_T, x_{T+1}, r_{T+1})$$

$$= d_0(x_1) \prod_{t=1}^{T} f(x_t, \theta)_{a_t} p(x_t, a_t, x_{t+1}, r_{t+1})$$

EPISODE DERIVATIVE

MAKE EPISODE MORE LIKELY

$$
\begin{aligned}
\frac{\partial}{\partial\theta} \ln \Pr(H_{1:T} = \tau_{1:T}) &= \frac{\partial}{\partial\theta} \ln\left(d_0(x_1) \prod_{t=1}^{T} f(x_t, \theta)_{a_t} p(x_t, a_t, x_{t+1}, r_{t+1}) \right) \\
&= \frac{\partial}{\partial\theta} \left[\ln d_0(x_1) + \sum_{t=1}^{T} \ln f(x_t, \theta)_{a_t} + \ln p(x_t, a_t, x_{t+1}, r_{t+1}) \right] \\
&= \frac{\partial}{\partial\theta} \ln d_0(x_1) + \sum_{t=1}^{T} \frac{\partial}{\partial\theta} \ln f(x_t, \theta)_{a_t} + \frac{\partial}{\partial\theta} \ln p(x_t, a_t, x_{t+1}, r_{t+1}) \\
&= \sum_{t=1}^{T} \frac{\partial}{\partial\theta} \ln f(x_t, \theta)_{a_t}
\end{aligned}
$$

EPISODE DERIVATIVE

MAKE EPISODE MORE LIKELY

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\ln \Pr(H_{1:T}=\tau_{1:T}) &= \frac{\partial}{\partial\theta}\ln\left(d_0(x_1)\prod_{t=1}^{T} f(x_t,\theta)_{a_t} p(x_t,a_t,x_{t+1},r_{t+1})\right)\\
&= \frac{\partial}{\partial\theta}\left[\ln d_0(x_1)+\sum_{t=1}^{T}\ln f(x_t,\theta)_{a_t}+\ln p(x_t,a_t,x_{t+1},r_{t+1})\right]\\
&= \frac{\partial}{\partial\theta}\ln d_0(x_1)+\sum_{t=1}^{T}\frac{\partial}{\partial\theta}\ln f(x_t,\theta)_{a_t}+\frac{\partial}{\partial\theta}\ln p(x_t,a_t,x_{t+1},r_{t+1})\\
&= \sum_{t=1}^{T}\frac{\partial}{\partial\theta}\ln f(x_t,\theta)_{a_t}
\end{aligned}
$$

Do not need to know p to take the derivative!

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \mathbf{E}\left[(G_1 - \rho(\theta))\frac{\partial \ln \Pr\left(H_{1:T}\right)}{\partial \theta}\right]$$

POLICY GRADIENT

EXPRESSION

$$\nabla \rho(\theta) = \mathbf{E}\left[(G_1 - \rho(\theta)) \sum_{t=1}^{T} \frac{\partial \ln f(X_t)_{A_t}}{\partial \theta}\right]$$

The agent doesn't need to model the world to improve its decision.

REINFORCE

SIMPLE STOCHASTIC POLICY GRADIENT ALGORITHM

Idea: sample an episode τ using actions from $f(X_t, \theta)$ and compute the sample estimate of the gradient.

REINFORCE

SIMPLE STOCHASTIC POLICY GRADIENT ALGORITHM

Collect_episode(θ):

$X_1 \sim d_0$

states = [], actions = [], rewards = []

for $t \in \{1, \ldots, T\}$

$A_t \sim f(X_t, \theta)$

states.append(X_t), actions.append(A_t)

$X_{t+1}, R_{t+1} \sim$ environment(X_t, A_t)

rewards.append(R_{t+1})

return states, actions, rewards

Sample_gradient(states, actions, rewards, θ, $\hat{\rho}$)

$G_1 \leftarrow$ sum(rewards)

return $(G_1 - \hat{\rho}) \sum_{t=1}^{T} \frac{\partial \ln f(X_t, \theta)_{A_t}}{\partial \theta}$

REINFORCE

SIMPLE STOCHASTIC POLICY GRADIENT ALGORITHM

REINFORCE(θ, η, β, max_iters)

$\hat{\rho} \leftarrow 0$

for itr in 1:max_iters

states, actions, rewards = collect_episode(θ)

$\widehat{\nabla} \leftarrow$ sample_gradient(states, actions, rewards, $\theta, \hat{\rho}$)

$\theta \leftarrow \theta + \eta \widehat{\nabla}$

$\hat{\rho} \leftarrow \beta\hat{\rho} + (1 - \beta)G_1$

REINFORCE

PROPERTIES

High variance gradient estimates:

- Observe one sampled episode
- The cumulative reward G_1 is high-variance
- $R_{t'<t}$ has no impact on A_t

Solutions:

- Sample multiple episodes
- Use G_t for each $\frac{\partial \ln f(X_t, \theta)_{A_t}}{\partial \theta}$
- Predict G_t give X_t

GRADIENT ESTIMATE

A MORE COMMON GRADIENT ESTIMATE

Sample_gradient(states, actions, rewards, θ, v)

$G_1, G_2, \ldots, G_T \leftarrow$ cumulative_rewards(rewards)

$$\widehat{\nabla} \leftarrow \sum_{t=1}^{T} (G_t - v(X_t)) \frac{\partial \ln f(X_t, \theta)_{A_t}}{\partial \theta}$$

return $\widehat{\nabla}$

POLICY GRADIENT METHODS

SOME GOOD THINGS TO KNOW

They mix exploration (trying new actions) with exploitation (trying actions that worked well)

If a large step is taken (usually with large η) the following can happen:

- The agent oversteps and the policy can diverge (same as in supervised learning)
- The agent overcommits to sub-optimal actions, i.e.,
  - it won't try any new actions anymore
  - and cannot keep improving.

Policy gradient methods often get stuck in flat gradient spaces and stop trying new actions.

RL FOR TRAINING LANGUAGE MODELS

GENERAL SET UP

Supervised pretraining

Minimize negative log-likelihood on next token prediction over a large data set

Goal: have the model generate text that is preferable to the user.

Problem: next token prediction does not mean generating useful responses

Solution: Sample text outputs and score the outputs. Make the outputs that score better more likely!

This is the policy gradient process!

RL FOR TRAINING LANGUAGE MODELS

MORE SPECIFIC

$X_{1:k}$ initial context

$\hat{X}_{k+1:t} \sim f(X_{1:k}, \theta)$ generated output

$G = r(\hat{X}_{1:t})$ score for the entire text

$\rho(\theta) = \mathbf{E}[G]$

SCORING TEXT

What is deep learning?

Deep learning is a mathematical framework for constructing a layered composition of functions that can be trained via gradient descent.

SCORING TEXT

What is deep learning?

Deep learning is a subset of machine learning that focuses on using artificial neural networks to model and solve complex problems. It is inspired by the structure and function of the human brain, particularly how neurons connect and communicate with one another.

Key Features of Deep Learning:

1. **Neural Networks**: Deep learning uses multi-layered neural networks, often called deep neural networks, where each layer extracts increasingly abstract features from the data.
2. **Large Data Requirements**: Deep learning algorithms typically require large datasets to achieve high accuracy, as they need extensive examples to generalize well.
3. **Representation Learning**: It automatically discovers the representations needed for feature detection or classification from raw data, reducing the need for manual feature extraction.
4. **Complex Problem Solving**: It excels at tasks involving complex data like images, audio, text, and video.

Common Architectures in Deep Learning:

1. **Convolutional Neural Networks (CNNs)**: Used primarily for image and video recognition.
2. **Recurrent Neural Networks (RNNs)**: Effective for sequential data like time series and natural language.
3. **Transformers**: Revolutionized natural language processing (NLP) with models like GPT and BERT.
4. **Autoencoders**: Used for unsupervised learning, such as anomaly detection or data compression.
5. **Generative Adversarial Networks (GANs)**: Generate new data similar to the training data, such as synthetic images

…

SCORING TEXT

What score should we assign to each one?

What is deep learning?

Deep learning is a mathematical framework for constructing a layered composition of functions that can be trained via gradient descent.

What is deep learning?

Deep learning is a subset of machine learning that focuses on using artificial neural networks to model and solve complex problems. It is inspired by the structure and function of the human brain, particularly how neurons connect and communicate with one another.

Key Features of Deep Learning:

1. **Neural Networks**: Deep learning uses multi-layered neural networks, often called deep neural networks, where each layer extracts increasingly abstract features from the data.
2. **Large Data Requirements**: Deep learning algorithms typically require large datasets to achieve high accuracy, as they need extensive examples to generalize well.

SCORING TEXT

What score should we assign to each one?

Extremely difficult to construct a scoring system!

Rewarding intuitive things (e.g., length) can produce undesired consequences (e.g., generating super-long answers).

Humans have preferences, e.g.,

- Rank one output as better than another

What is deep learning?

Deep learning is a mathematical framework for constructing a layered composition of functions that can be trained via gradient descent.

What is deep learning?

Deep learning is a subset of machine learning that focuses on using artificial neural networks to model and solve complex problems. It is inspired by the structure and function of the human brain, particularly how neurons connect and communicate with one another.

Key Features of Deep Learning:

1. **Neural Networks**: Deep learning uses multi-layered neural networks, often called deep neural networks, where each layer extracts increasingly abstract features from the data.
2. **Large Data Requirements**: Deep learning algorithms typically require large datasets to achieve high accuracy, as they need extensive examples to generalize well.

APPROXIMATING HUMAN PREFERENCES

$r(\theta_r, X_{1:k}, \hat{X}_{k+1:t}) \approx$ human's preferences between two samples $\hat{X}_{k+1:t}$ and $\hat{X}'_{k+1:t}$

Let $X'_{k+1:t}$ be the human prefered generation

$$l(\theta_r) = \mathbf{E}\left[\ln\left(\sigma\left(r(\theta_r, X_{1:k}, X'_{k+1:t}) - r(\theta_r, X_{1:k}, X_{k+1:t})\right)\right)\right]$$

RLHF

$$X_{k+1:t} \sim f(X_{1:k}, \theta)$$

$$\rho(\theta) = \mathbf{E}\left[r(\theta_r, X_{1:k}, X_{k+1:t})\right]$$

RLHF

$$X_{k+1:t} \sim f(X_{1:k}, \theta)$$

$$\rho(\theta) = \mathbf{E}\left[r(\theta_r, X_{1:k}, X_{k+1:t})\right]$$

$\arg\max_{\theta} \rho(\theta)$ will overfit to learned preferences $r(\theta_r, \ldots)$

RLHF

Idea: keep generated outputs close to that from the supervised learning model.

θ_{SL} — weights learned from supervised learning

$$\rho(\theta) = \mathbf{E}\left[r(\theta_r, X_{1:k}, X_{k+1:t}) + \beta \ln \frac{f(X_{1:k}, \theta)_{X_{k+1:t}}}{f(X_{1:k}, \theta_{SL})_{X_{k+1:t}}}\right]$$

RLHF

SUMMARY

Iterative process:

1. Sample model outputs
2. Collect human rankings of outputs
3. Model the ranking (learning r)
4. Optimize the model ($\arg\max_{\theta} \rho(\theta)$)
5. Repeat

RLHF

Still an active area of research!

Many problems:

- Estimating r is very difficult
- Optimizing θ of leads to overfitting and not producing diverse text
- Collecting human feedback

RLHF

Still an active area of research!

Many problems:

- Estimating r is very difficult
- Optimizing θ of leads to overfitting and not producing diverse text
- Collecting human feedback

Human feedback data is collected from poorer parts of the world where workers are in harsh conditions!

Human feedback does not mean the model is correct. Humans are wrong all the time and contain their own biases.

NEXT CLASS

Reinforcement Learning from Human Feedback