CS234:
Reinforcement Learning

Emma Brunskill
Stanford University
Winter 2018

Today the 3rd part of the lecture is based on David Silver's introduction to RL slides

Welcome! Today's Plan

- Overview about reinforcement learning
- Course logistics
- Introduction to sequential decision making under uncertainty

Reinforcement Learning

Learn to make good sequences of decisions

Repeated Interactions with World

Learn to make good sequences of decisions

Reward for Sequence of Decisions

Learn to make good sequences of decisions

Don’t Know in Advance How World Works

Learn to make good sequences of decisions

Fundamental challenge in artificial intelligence and machine learning is learning to make good decisions under uncertainty

RL, Behavior & Intelligence

Childhood: primitive brain & eye, swims around, attaches to a rock

Adulthood: digests brain. Sits

Suggests brain is helping guide decisions (no more decisions, no need for brain?)

Example from Yael Niv

Atari

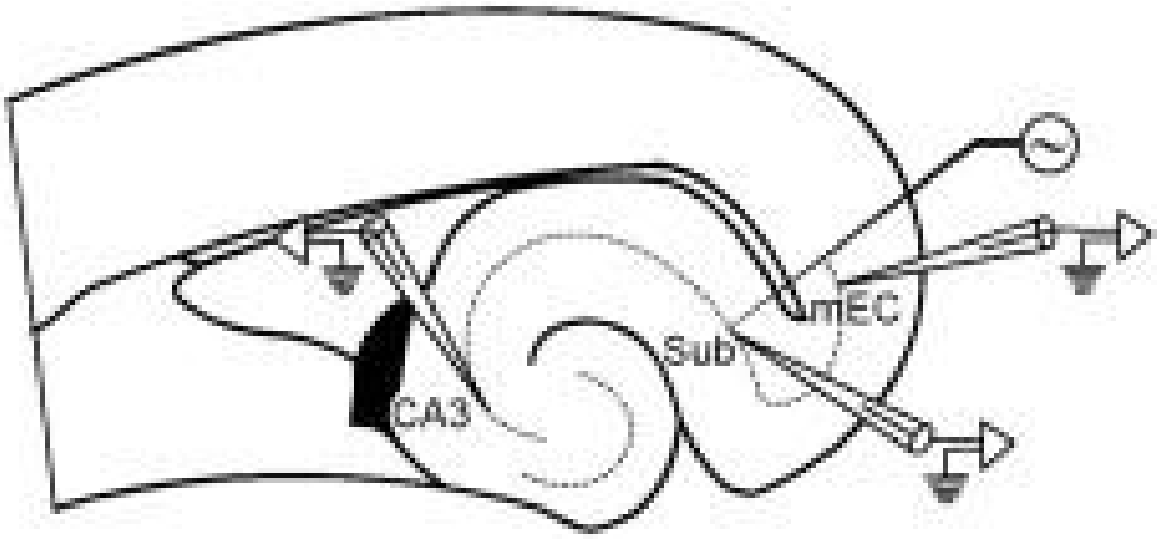
DeepMind Nature 2015

Robotics

Educational Games

RL used to optimize Refraction 1, Mandel, Liu, Brunskill, Popovic AAMAS 2014

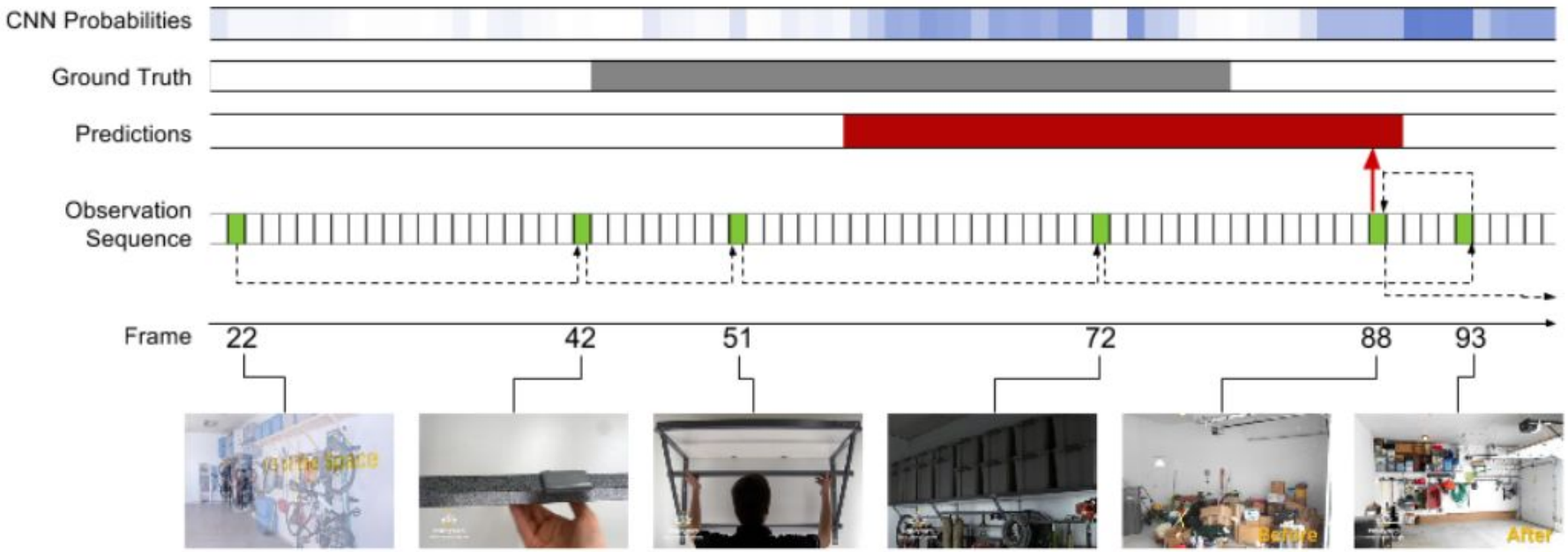
Healthcare



Adaptive control of epileptiform excitability in an in vitro model of limbic seizures.

Panuccio,Guez, Vincent,' Avoli' Pineau'

NLP, Vision, …



Yeung, Russakovsky, Mori, Li *2016*

Reinforcement Learning Involves

- Optimization
- Delayed consequences
- Exploration
- Generalization

Optimization

- Goal is to find an optimal way to make decisions
  - Yielding best outcomes
- Or at least very good strategy

Delayed Consequences

- Decisions now can impact things much later…
  - Saving for retirement
  - Finding a key in Montezuma's revenge
- Introduces two challenges
  1) When planning: decisions involve reasoning about not just immediate benefit of a decision but how its longer term ramifications
  2) When learning: temporal credit assignment is hard (what caused later high or low rewards?)

Exploration

- Learning about the world by making decisions
  - Agent as scientist
  - Learn to ride a bike by trying (and falling)
  - Finding a key in Montezuma's revenge
- Censored data
  - Only get a reward (label) for decision made
  - Don't know what would have happened if had taken red pill instead of blue pill (Matrix movie reference)
- Decisions impact what learn about
  - If choose going to Stanford instead of going to MIT, will have different later experiences…

- Policy is mapping from past experience to action
- Why not just pre-program a policy?

Generalization

- Policy is mapping from past experience to action
- Why not just pre-program a policy?

→ Go Up

Input: Image

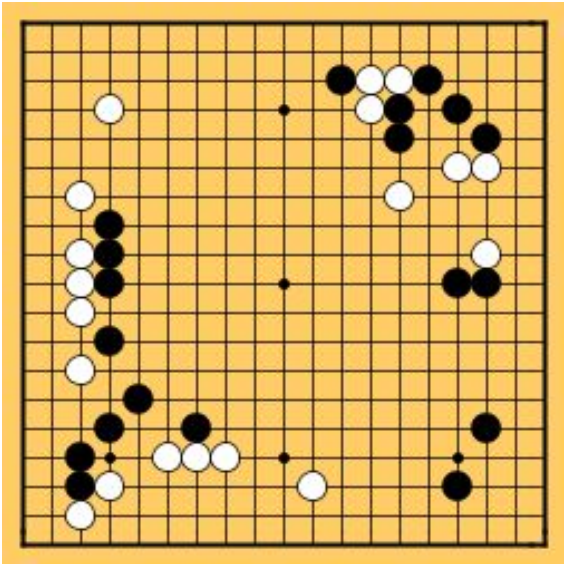
How many images are there? $(256^{100*200})^3$

Reinforcement Learning Involves

- Optimization
- Generalization
- Exploration
- Delayed consequences

AI Planning (vs RL)

- Optimization
- Generalization
- Exploration
- Delayed consequences



- Computes good sequence of decisions
- But given model of how decisions impact world

Supervised Machine Learning (vs RL)

- Optimization
- Generalization
- Exploration
- Delayed consequences

- Learns from experience
- But provided correct labels

Unsupervised Machine Learning (vs RL)

- Optimization
- Generalization
- Exploration
- Delayed consequences

- Learns from experience
- But no labels from world

Imitation Learning

- Optimization
- Generalization
- Exploration
- Delayed consequences

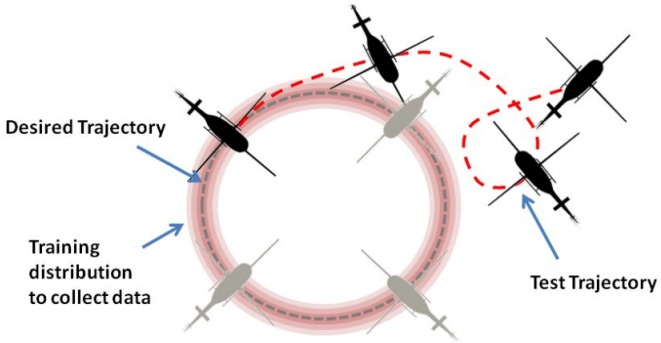
- Learns from experience… of others
- Assumes input demos of good policies

Imitation Learning

Abbeel, Coates and Ng helicopter team, Stanford

Imitation Learning

- Reduces RL to supervised learning
- Benefits
  - Great tools for supervised learning
  - Avoids exploration problem
  - With big data lots of data about outcomes of decisions
- Limitations
  - Can be expensive to capture
  - Limited by data collected
- Imitation learning + RL promisi



Ross & Bagnell 2013

How Do We Proceed?

- Explore the world
- Use experience to guide future decisions

Other issues

- Where do rewards come from?
  - And what happens if we get it wrong?
- Robustness / Risk sensitivity
- We are not alone…
  - Multi agent RL

Today's Plan

- Overview about reinforcement learning
- Course logistics
- Introduction/review of sequential decision making under uncertainty

Basic Logistics

- Instructor: Emma Brunskill
- CAs: Alex Jin (head CA), Anchit Gupta, Andrea Zanette, James Harrison, Luke Johnson, Michael Painter, Rahul Sarkar, Shuhui Qu, Tian Tan, Xinkun Nie, Youkow Homma
- Time: MW 11:50am-1:20pm
- Location: Nvidia
- Additional information
  - Course webpage: http://cs234.stanford.edu
  - Schedule, Piazza link, lecture slides,

Prerequisites

- Python proficiency
- Basic probability and statistics
- Multivariate calculus and linear algebra
- Machine learning or AI (e.g. CS229 or CS221)
- The terms loss function, derivative, and gradient descent should be familiar
- Have heard of Markov decision processes and RL before in an AI or ML class
  - We will cover the basics, but quickly

Our Goal is that by the End of the Class You Will Be Able to:

- Define the key features of reinforcement learning that distinguish it from AI and non-interactive machine learning (as assessed by the exam)
- Given an application problem (e.g. from computer vision, robotics, etc) decide if it should be formulated as a RL problem, if yes be able to define it formally (in terms of the state space, action space, dynamics and reward model), state what algorithm (from class) is best suited to addressing it, and justify your answer. (as assessed by the project and the exam)
- Implement (in code) common RL algorithms including a deep RL algorithm (as assessed by the homeworks)
- Describe (list and define) multiple criteria for analyzing RL algorithms and evaluate algorithms on these metrics: e.g. regret, sample complexity, computational complexity, empirical performance, convergence, etc. (as assessed by homeworks and the exam)
- Describe the exploration vs exploitation challenge and compare and contrast at least two approaches for addressing this challenge (in terms of performance, scalability, complexity of implementation, and theoretical guarantees) (as assessed by an assignment and the exam)

Grading

- Assignment 1 10%
- Assignment 2 20%
- Assignment 3 15%
- Midterm 25%

Grading

- Assignment 1 10%
- Assignment 2 20%
- Assignment 3 15%
- Midterm 25%
- Quiz 5%
  - 4.5% individual, 0.5% group

Grading

- Assignment 1 10%
- Assignment 2 20%
- Assignment 3 15%
- Midterm 25%
- Quiz 5%
  - 4.5% individual, 0.5% group
- Final Project 25%
  - Proposal 1%
  - Milestone 3%
  - Poster presentation 5%
  - Paper 16%

Communication

- We believe students often learn an enormous amount from each other as well as from us, the course staff.
- Therefore we use Piazza to facilitate discussion and peer learning
  - Please use for all questions related to lectures, homeworks, and projects.

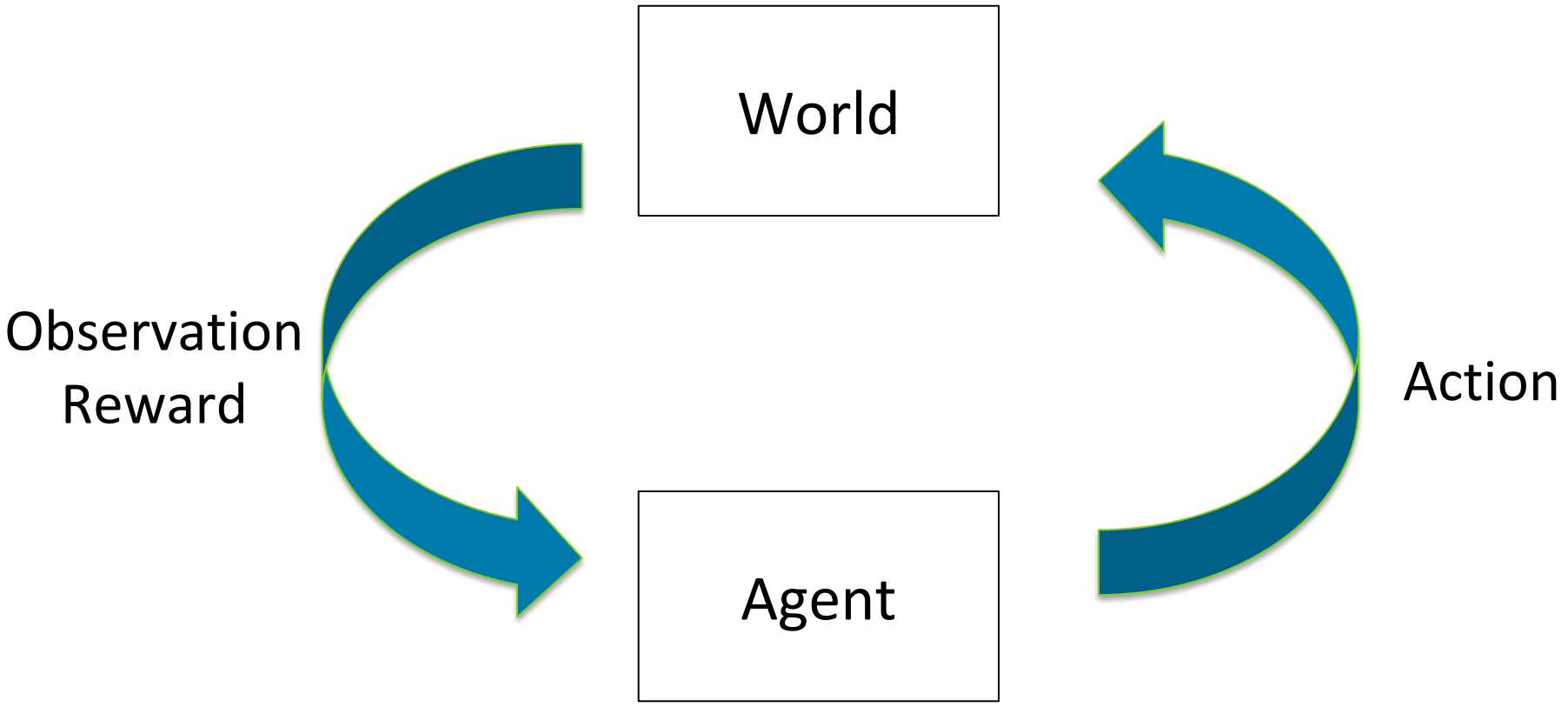
Grading

- Late policy
  - 6 free late days
  - See webpage for details on how many per assignment/project and penalty if use more
- Collaboration: see webpage and just reach out to us if you have any questions about what is considered allowed collaboration

Today's Plan

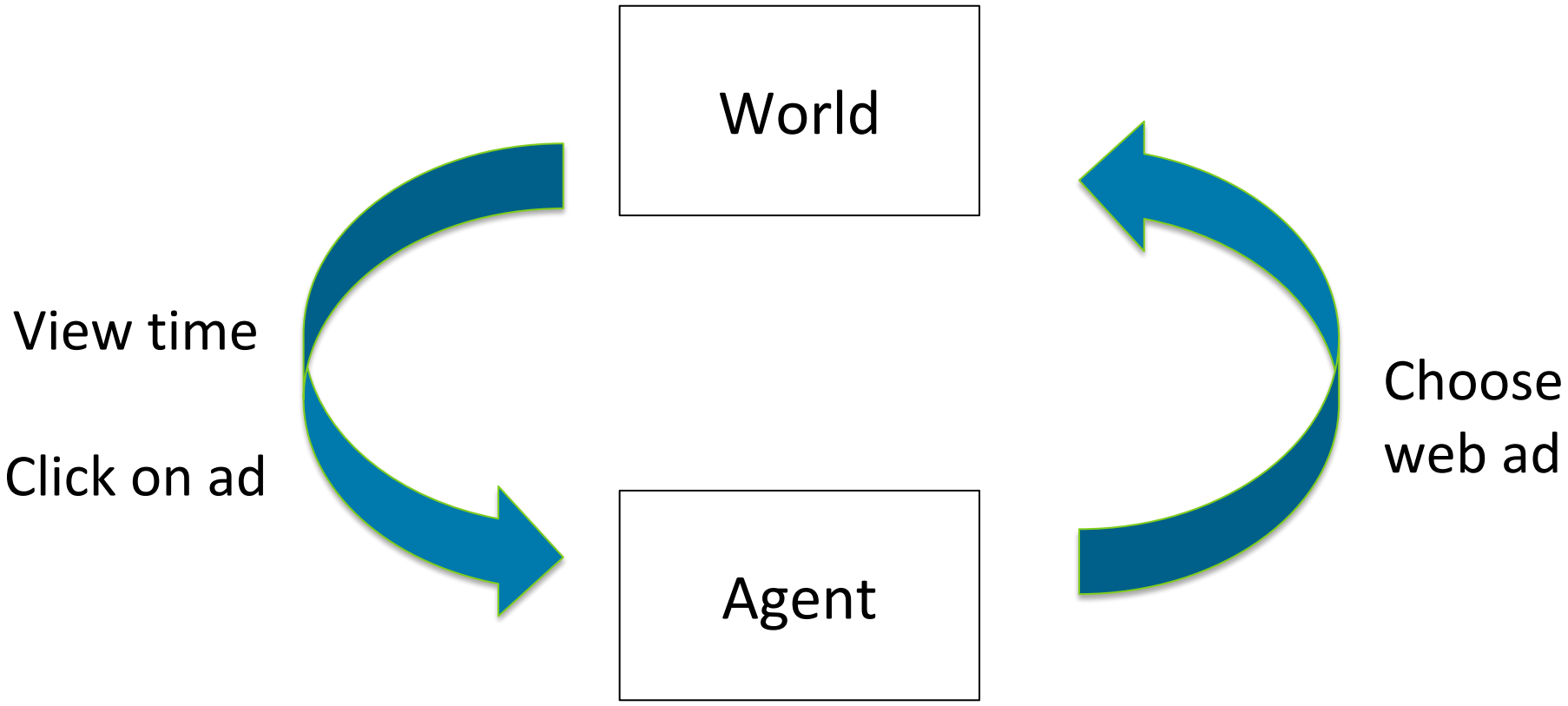
- Overview about reinforcement learning
- Course logistics
- Introduction/review of sequential decision making under uncertainty

Sequential Decision Making



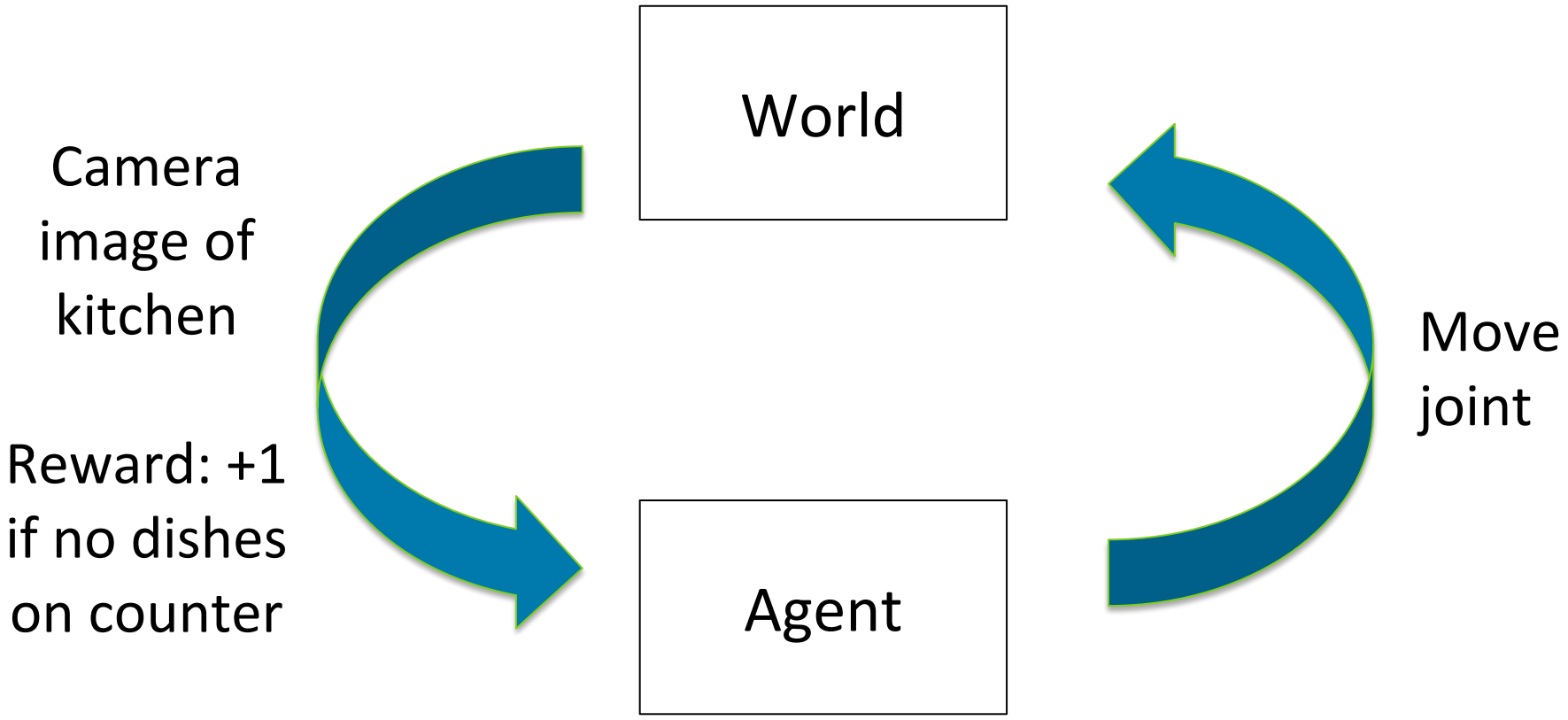
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards
- May require strategic behavior to achieve high rewards

Ex. Web Advertising



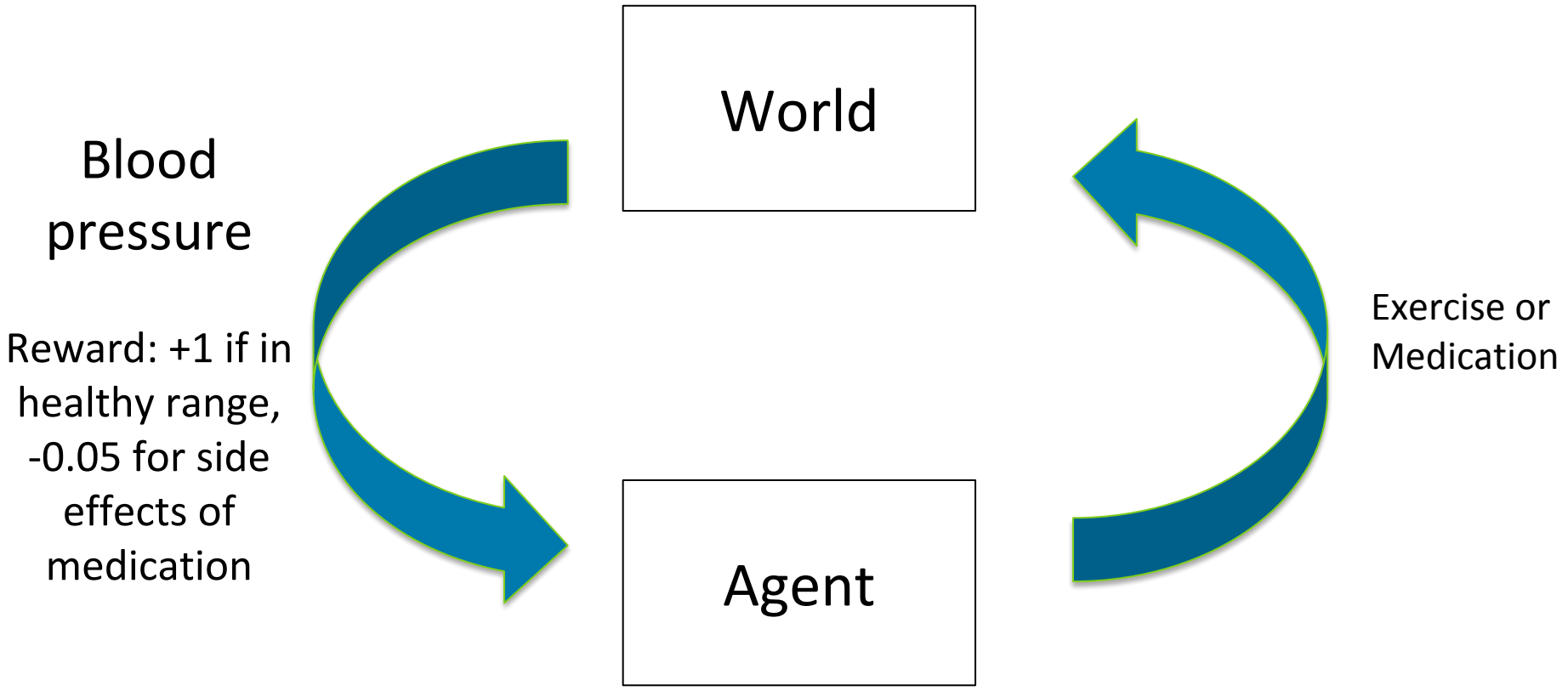
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards
- May require strategic behavior to achieve high rewards

Ex. Robot Unloading Dishwasher



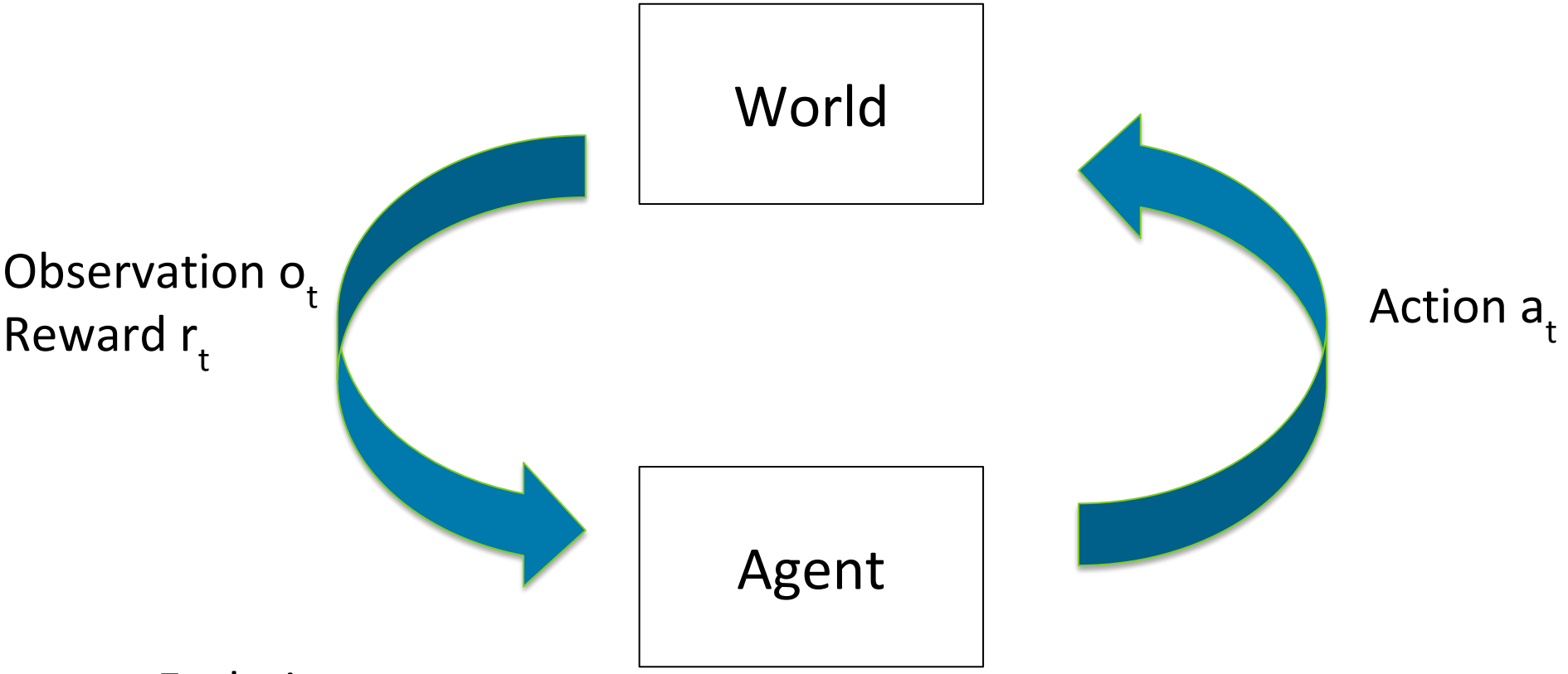
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards
- **May require strategic behavior to achieve high rewards**

Ex. Blood Pressure Control



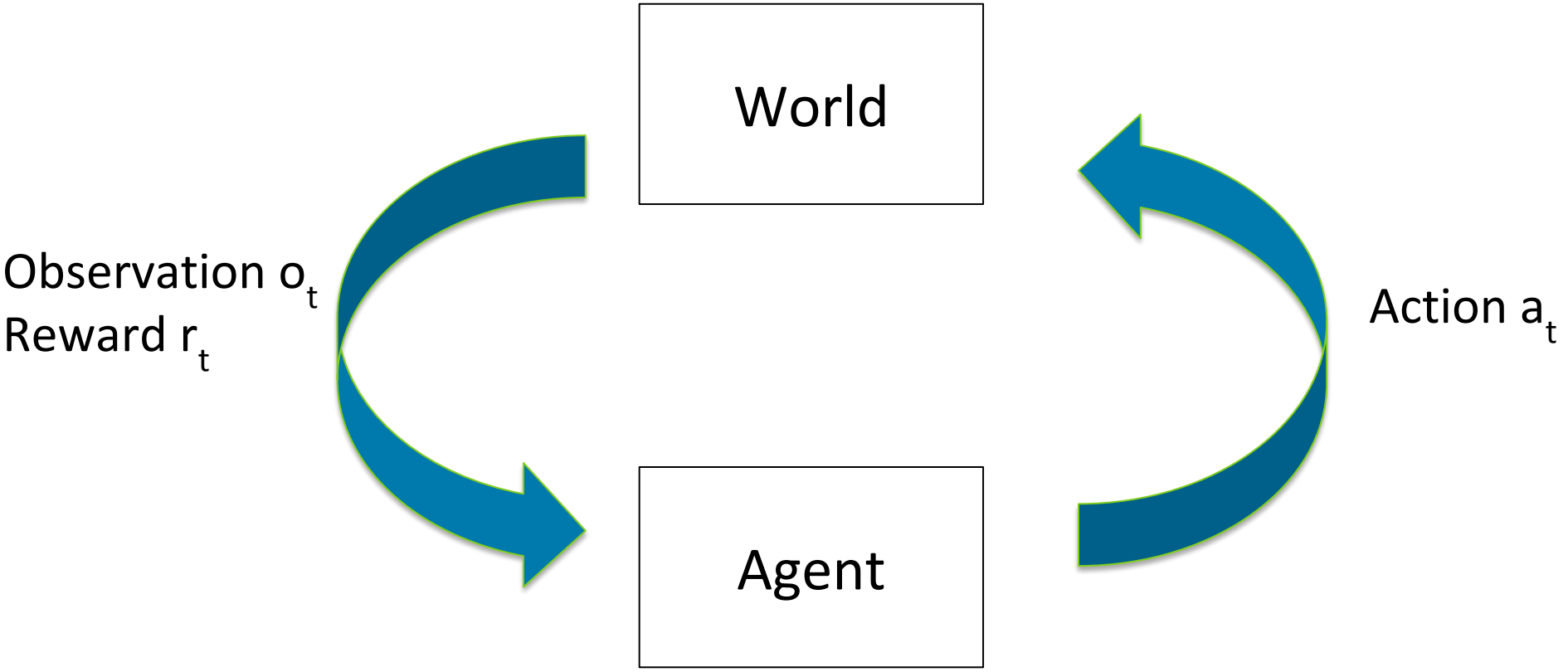
- Goal: Select actions to maximize total expected future reward
- **May require balancing immediate & long term rewards**
- **May require strategic behavior to achieve high rewards**

Sequential Decision Process: Agent & the World (Discrete Time)



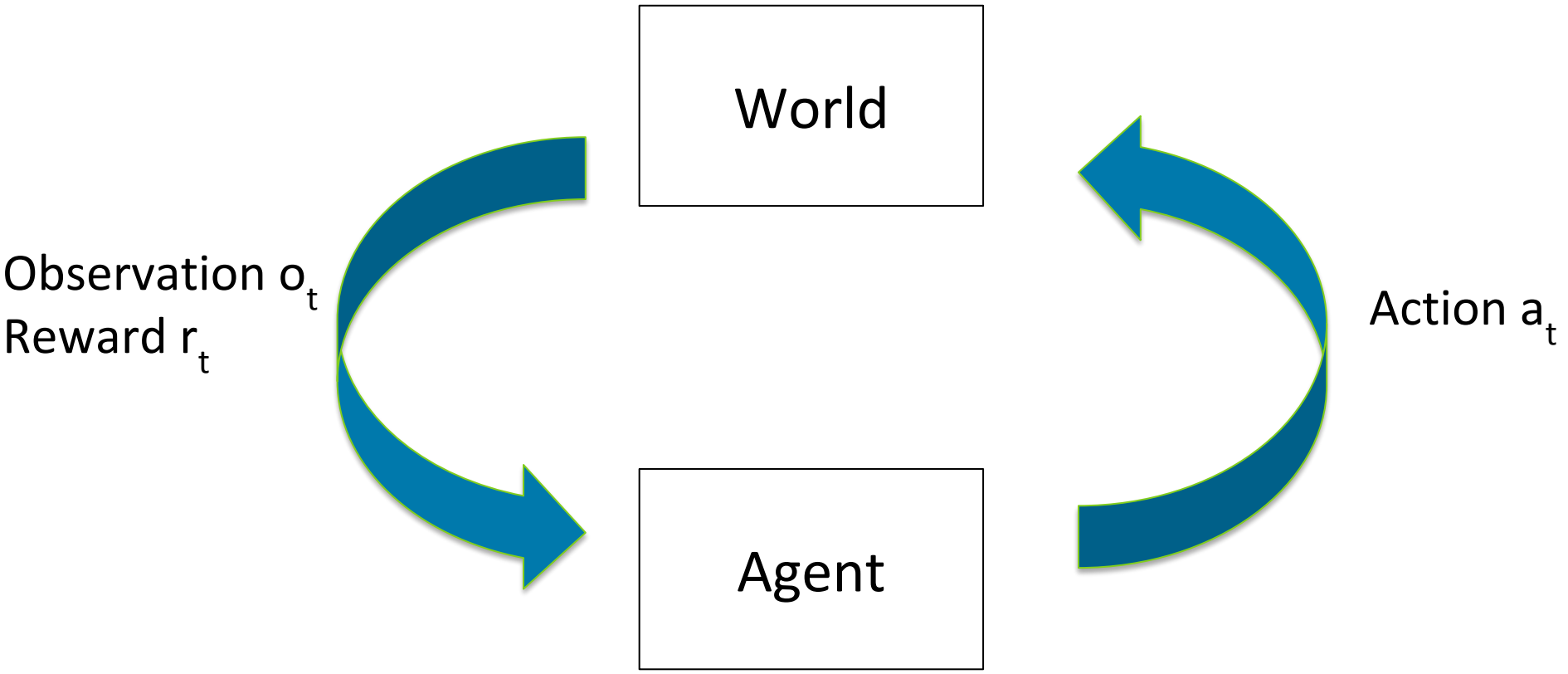
- Each time step t:
  - Agent takes an action a_t
  - World updates given action a_t, emits observation o_t, reward r_t
  - Agent receives observation o_t and reward r_t

History: Sequence of Past Observations, Actions & Rewards



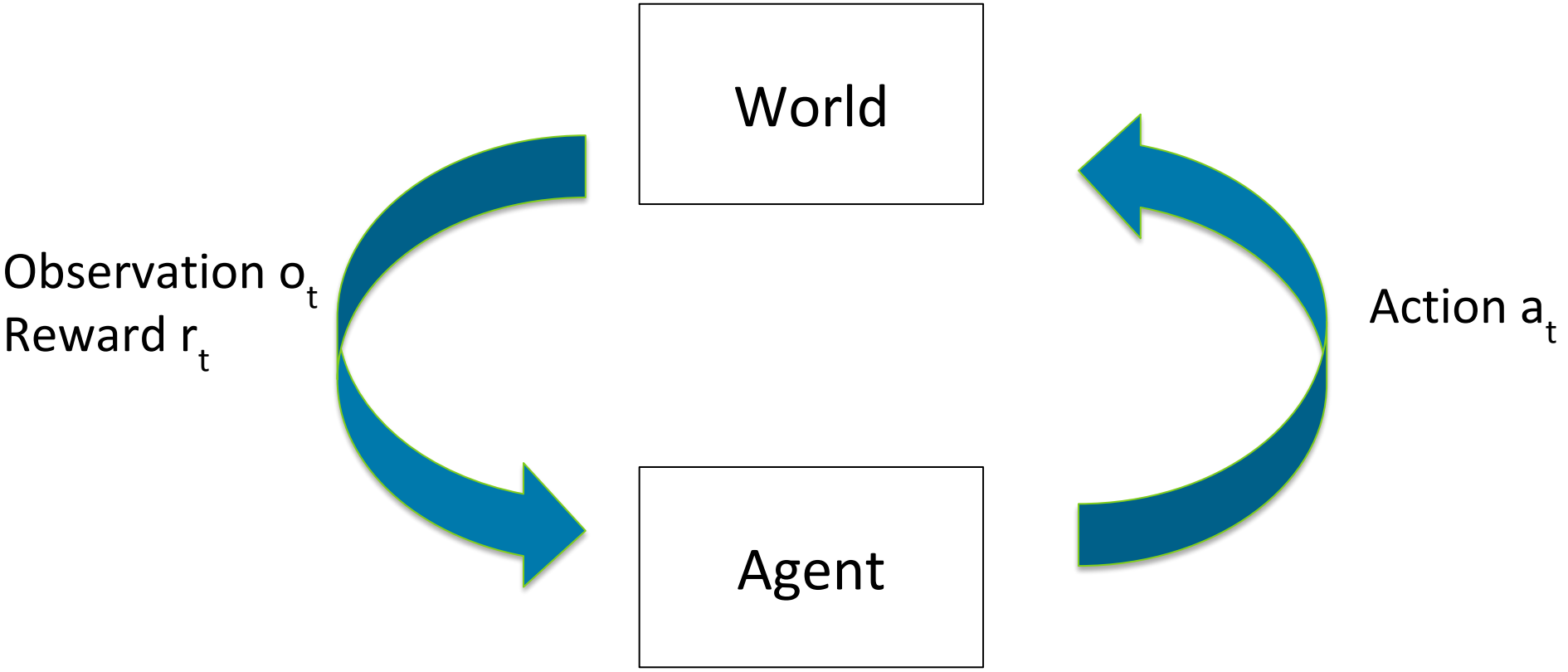
- HIstory $h_t=(a_1, o_1, r_1, \ldots a_t, o_t, r_t)$
- Agent chooses action based on history,
- State is information assumed to determine what happens next
  - Function of history: $s_t=(h_t)$

World State



- This is true state of the world used to determine how world generates next observation and reward
- Often hidden or unknown to agent
- Even if known may contain information not needed by agent

Agent State: Agent's Internal Representation



- What the agent / algorithm uses to make decisions about how to act
- Generally a function of the history: $s_t=(h_t)$
- Could include meta information like state of algorithm (how many computations executed, etc) or decision process (how many decisions left until an episode ends)

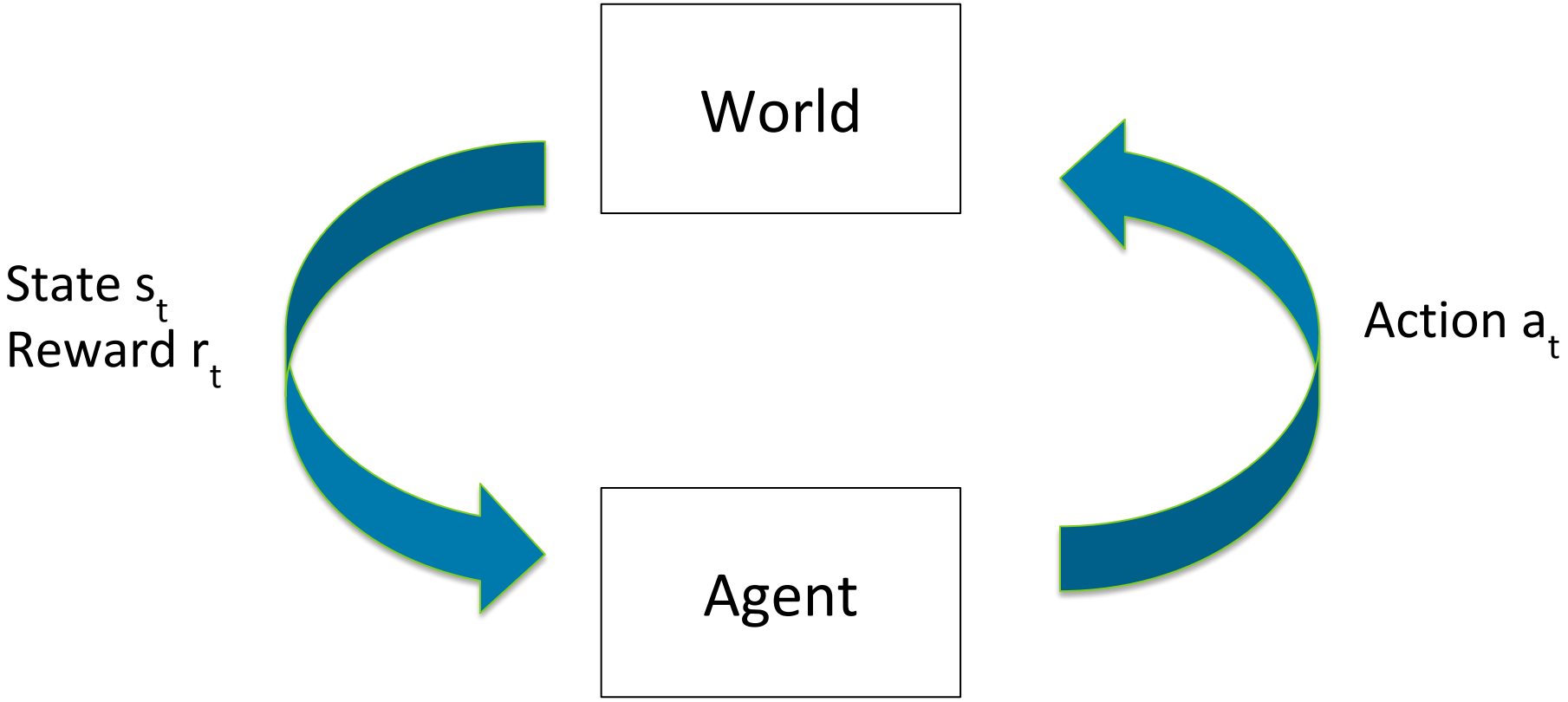
Markov

- Information state: sufficient statistic of history
- **Definition:**
  - State s_t is Markov if and only if (iff):
    - $p(s_{t+1}|s_t,a_t) = p(s_{t+1}|h_t,a_t)$
- Future is independent of past given present

Why is Markov Assumption Popular?

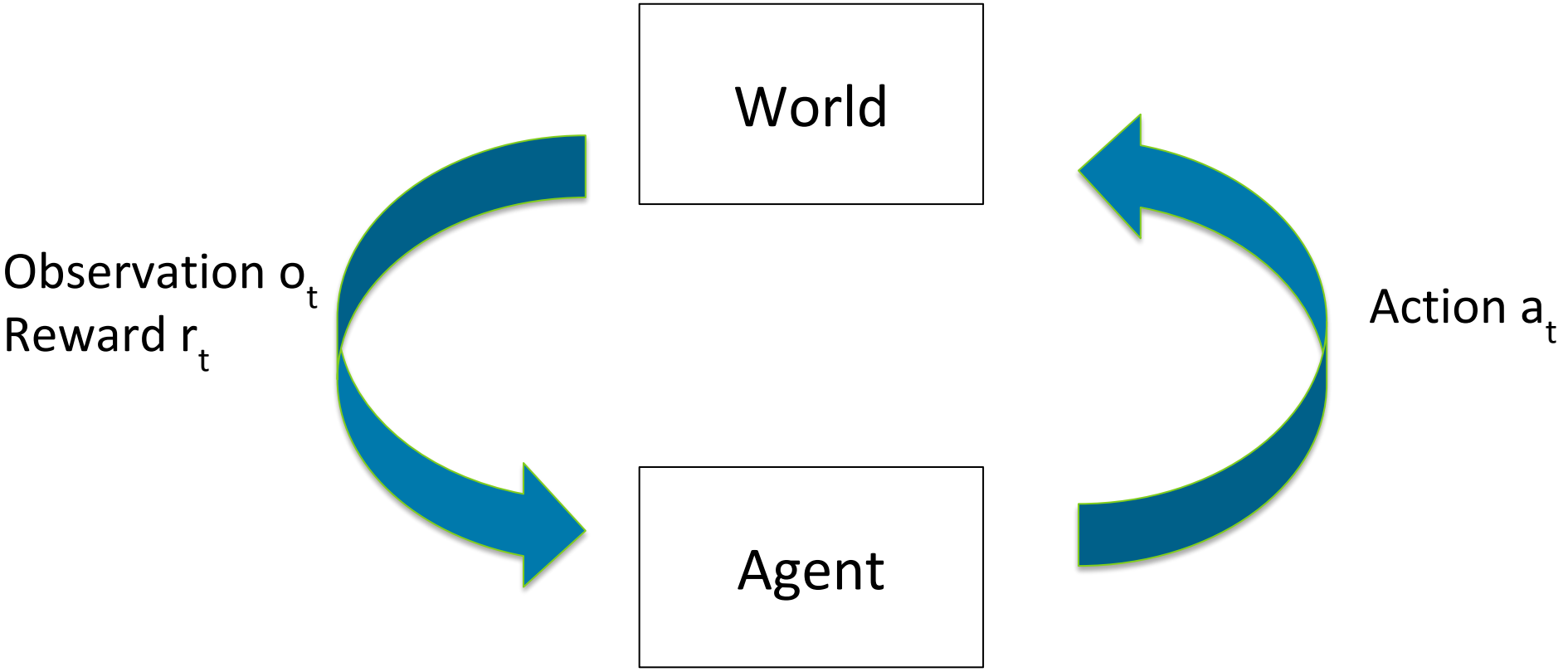
- Can always be satisfied
  - Setting state as history always Markov: $s_t = h_t$
- In practice often assume most recent observation is sufficient statistic of history $s_t = o_t$
- State representation has big implications for
  - computational complexity
  - data required
  - resulting performance
- when learning to make good sequences of decisions

Full Observability / Markov Decision Process (MDP)



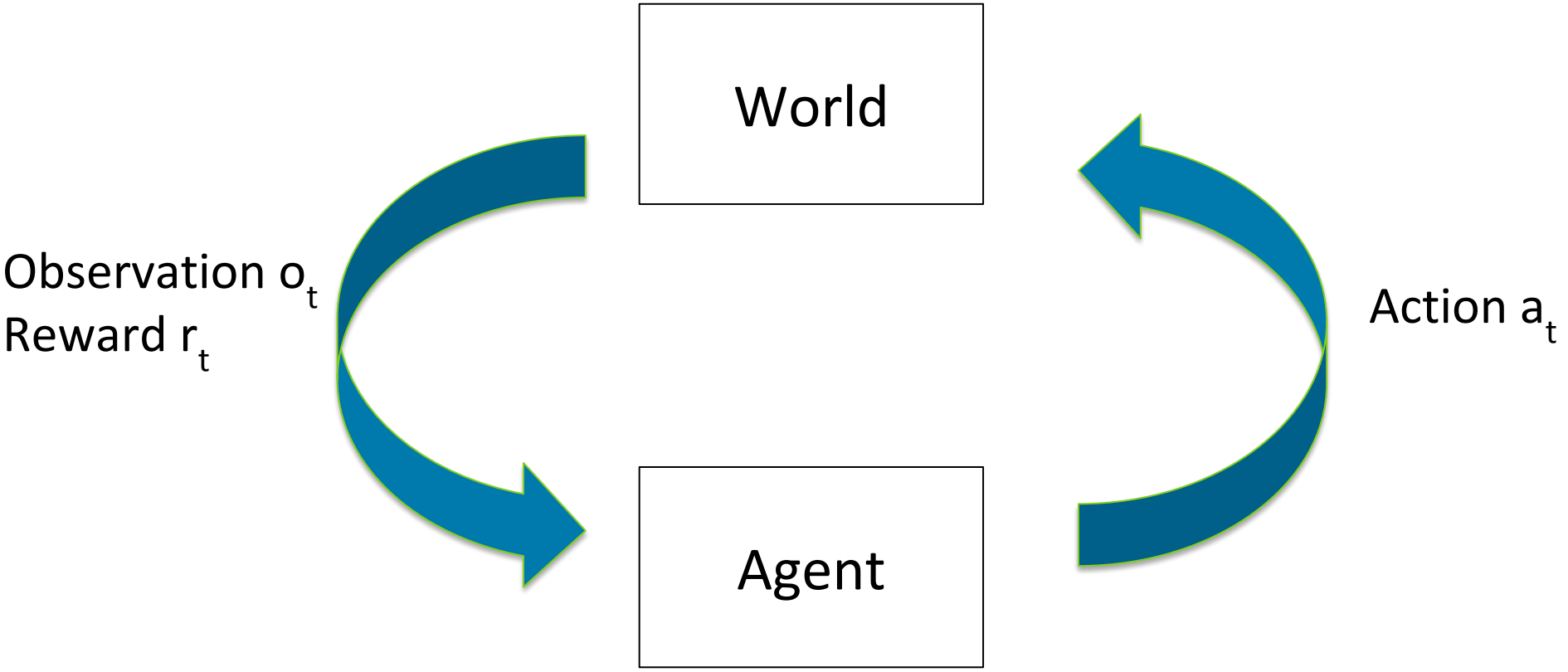
- Environment and World State $s_t = o_t$

Partial Observability /
Partially Observable Markov Decision Process (POMDP)



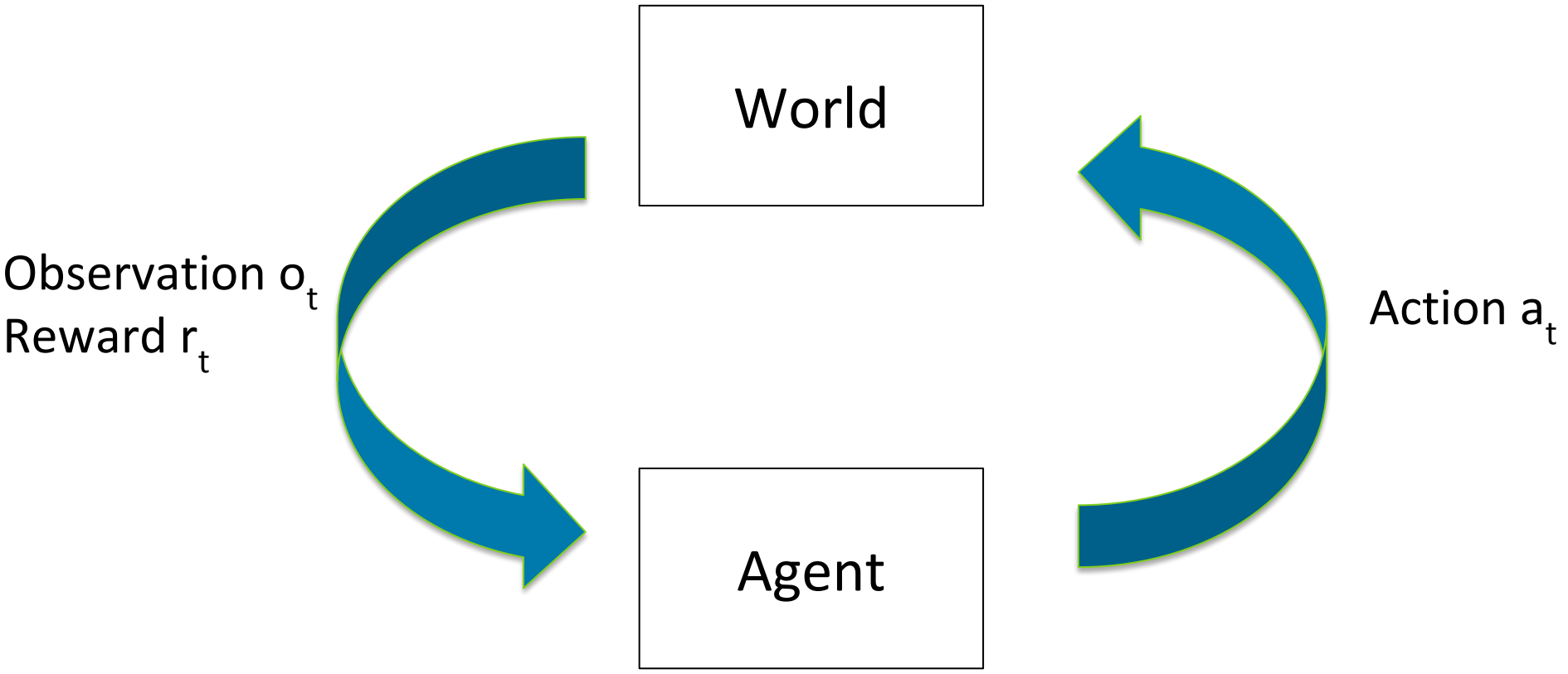
- Agent state is not the same as the world state
- Agent constructs its own state, e.g.
  - Use history $s_t = h_t$, or beliefs of world state, or RNN, ...

Partial Observability Examples: Poker player (only see own cards), Healthcare (don't see all physiological processes)....



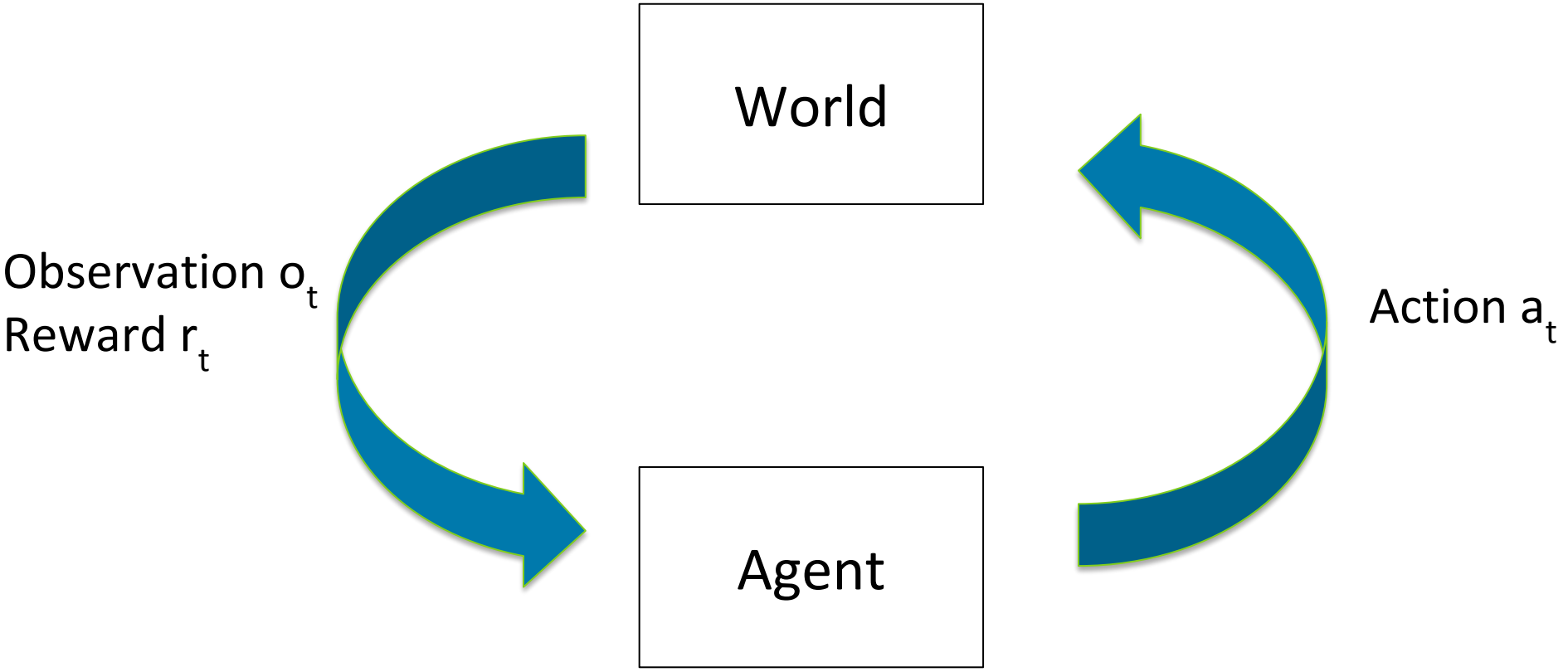
- Agent state is not the same as the world state
- Agent constructs its own state, e.g.
  - Use history $s_t = h_t$, or beliefs of world state, or RNN, ...

Types of Sequential Decision Processes: Bandits



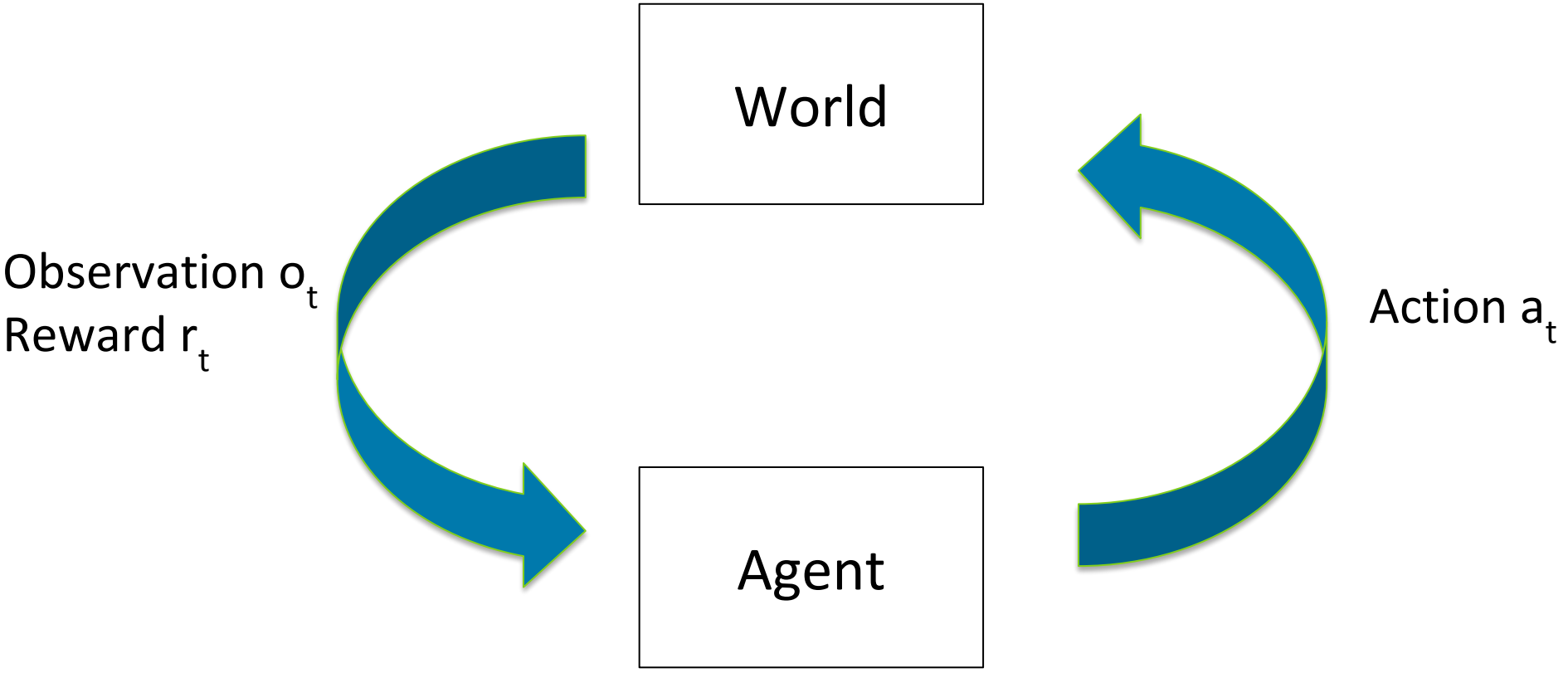
- Bandits: actions have no influence on next observations
- No delayed rewards

Types of Sequential Decision Processes: MDPs and POMDPs



- Actions influence future observations
- Credit assignment and strategic actions may be needed

Types of Sequential Decision Processes: How the World Changes



- **Deterministic**: Given history and action, single observation & reward
  - Common assumption in robotics and controls
- **Stochastic:** Given history and action, many potential observations & reward
  - Common assumption for customers, patients, hard to model domains

RL Agent Components

- Often include one or more of:
  - **Model:** Agent's representation of how the world changes in response to agent's action
  - **Policy:** function mapping agent's states to action
  - **Value function:** future rewards from being in a state and/or action when following a particular policy

Model

- Agent's representation of how the world changes in response to agent's action
- Transition / dynamics model predicts next agent state
  - $P(s_{t+1} = s' \mid s_t = s, a_t = a)$
- Reward model predicts immediate reward
  - $R(s_t = s, a_t = a) = \mathbb{E}[r_t \mid s_t = s, a_t = a]$

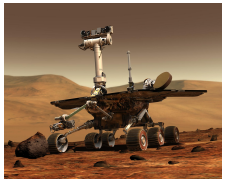
Policy

- Policy π determines how the action chooses actions
- $\pi: S \rightarrow A$, mapping from state to action
- Deterministic policy $\pi(s) = a$
- Stochastic policy $\pi(a|s) = P(a_t = a | s_t = s)$

Value

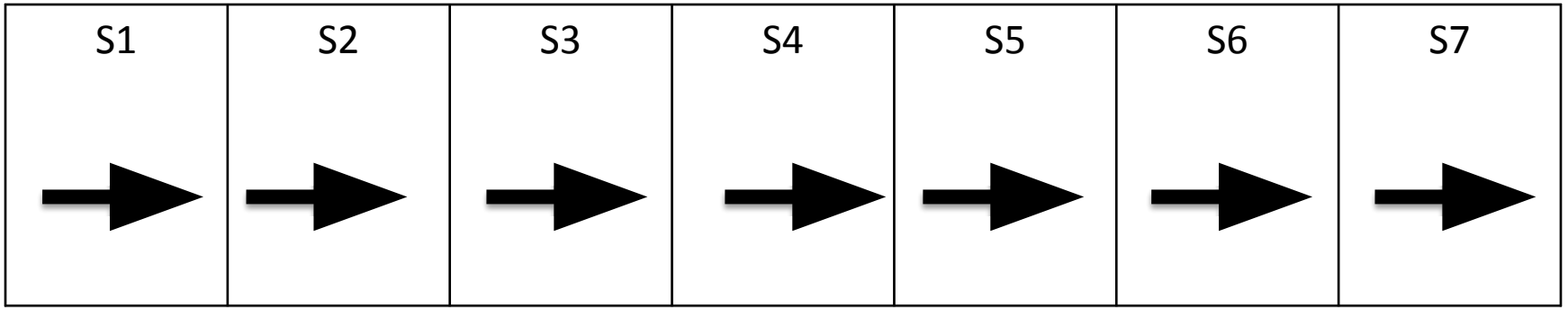
- Value function V^{π}: expected discounted sum of future rewards under a particular policy π
  - $V^{\pi}(s_t=s) = \mathbb{E}_{\pi}\,[r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+1} + \ldots \mid s_t = s]$
- Discount factor γ weighs immediate vs future rewards
- Can be used to quantify goodness/badness of states and actions
- And decide how to act by comparing policies

Example: Simple Mars Rover Decision Process

S1	S2	S3	S4	S5	S6	S7
						

- States: Location of rover (S1… S7)
- Actions: TL, TR
- Rewards
  - +1 in state S1
  - +10 in state S7
  - 0 in all other states

Example: Simple Mars Rover Policy



- Policy represented by arrows
- $\pi(S1)=\pi(S2)=...\pi(S7)=TR$

Example: Simple Mars Rover Value Function

S1	S2	S3	S4	S5	S6	S7
+1	0	0	0	0	0	+10

- γ=0
- π(S1)=π(S2)=...π(S7)=TR
- Numbers show value $V^{\pi}(s)$ for this policy and this discount factor

Example: Simple Mars Rover Model

S1	S2	S3	S4	S5	S6	S7
0	0	0	0	0	0	0

- Agent can construct its own estimate of the world models (dynamics and reward)
- In the above the numbers show the agent's estimate of the reward model
- Agent's transition model
  - P(S1|S1,TR)=0.5 = P(S2|S1,TR) ...
- Model may be wrong

Types of RL Agents: What the Agent (Algorithm) Learns

- Value based
  - Explicit: Value function
  - Implicit: Policy (can derive a policy from value function)
- Policy based
  - Explicit: policy
  - No value function
- Actor Critic
  - Explicit: Policy
  - Explicit: Value function

Types of RL Agents

- Model Based
  - Explicit: model
  - May or may not have policy and/or value function
- Model Free
  - Explicit: Value function and/or Policy Function
  - No model

RL Agents

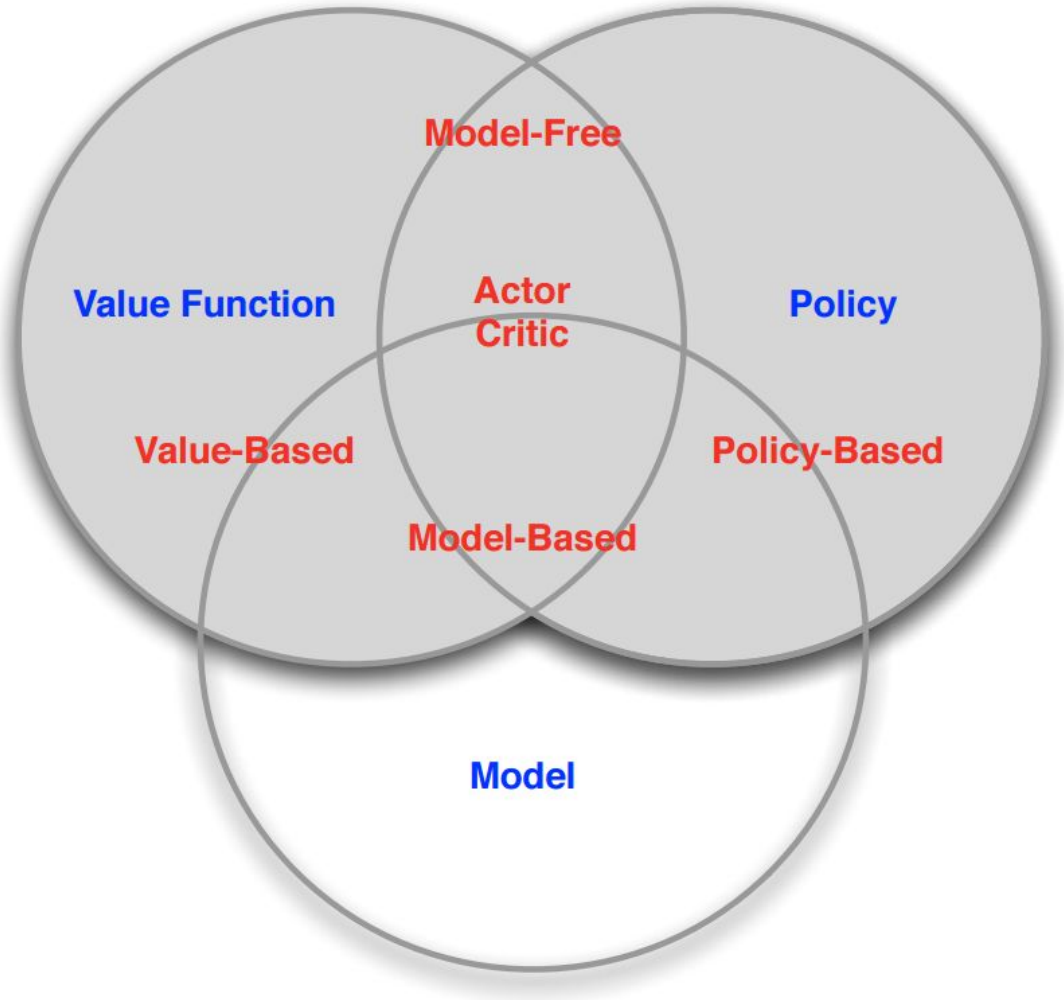


Figure from David Silver

Key Challenges in Learning to Make Sequences of Good Decisions

- Planning (Agent's internal computation)
  - Given model of how the world works
    - Dynamics and reward model
  - Algorithm computes how to act in order to maximize expected reward
    - With no interaction with real environment

Key Challenges in Learning to Make Sequences of Good Decisions

- Planning (Agent's internal computation)
  - Given model of how the world works
    - Dynamics and reward model
  - Algorithm computes how to act in order to maximize expected reward
    - With no interaction with real environment
- Reinforcement learning
  - Agent doesn't know how world works
  - Interacts with world to implicitly/explicits learn how world works
  - Agent improves policy (may involve planning)

Planning Example

- Solitaire: single player card game
- Know all rules of game / perfect model
- If take action a from state s
  - Can compute probability distribution over next state
  - Can compute potential score
- Can plan ahead to decide on optimal action
  - E.g. dynamic programming, tree search, …

Reinforcement Learning Example

- Solitaire with no rule book
- Learn directly by taking actions and seeing what happens
- Try to find a good policy over time (that yields high reward)

Exploration and Exploitation

- Agent only experiences what happens for the actions it tries
  - Mars rover trying to drive left learns the reward and next state for trying to drive left, but not for trying to drive right.
  - Obvious! But leads to a dilemma

Exploration and Exploitation

- Agent only experiences what happens for the actions it tries
- How balance should a RL agent balance
  - Exploration -- trying new things that enable agent to make better decisions in the future
  - Exploitation -- choosing actions that are expected to yield good reward given past experience
- Often there may be an exploration-exploitation tradeoff
  - May have to sacrifice reward in order to explore & learn about potentially better policy

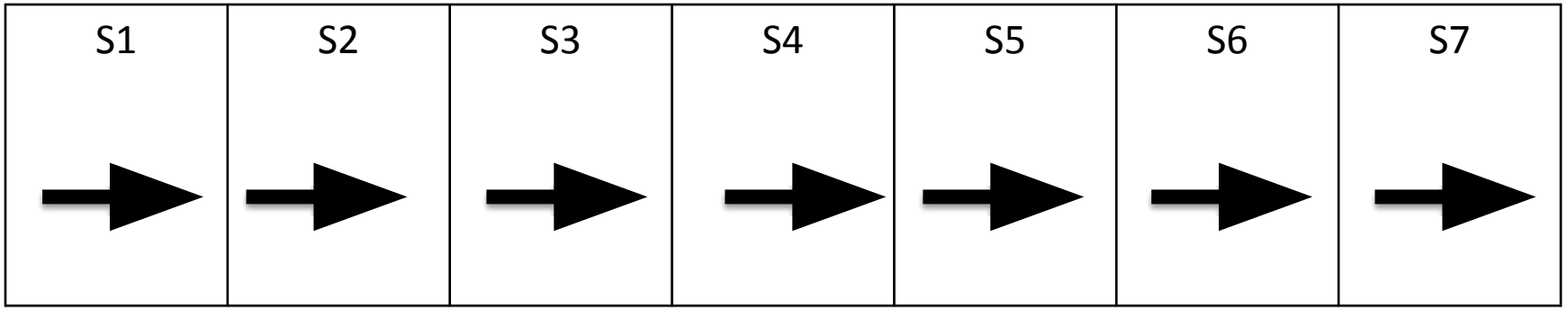
Exploration and Exploitation Examples

- Movies
  - Exploitation: Watch a favorite movie you've seen
  - Exploration: Watch a new movie
- Advertising
  - Exploitation: Show most effective ad so far
  - Exploration: Show a different ad
- Driving
  - Exploitation: Try fastest route given prior experience
  - Exploration: Try a different route

Evaluation and Control

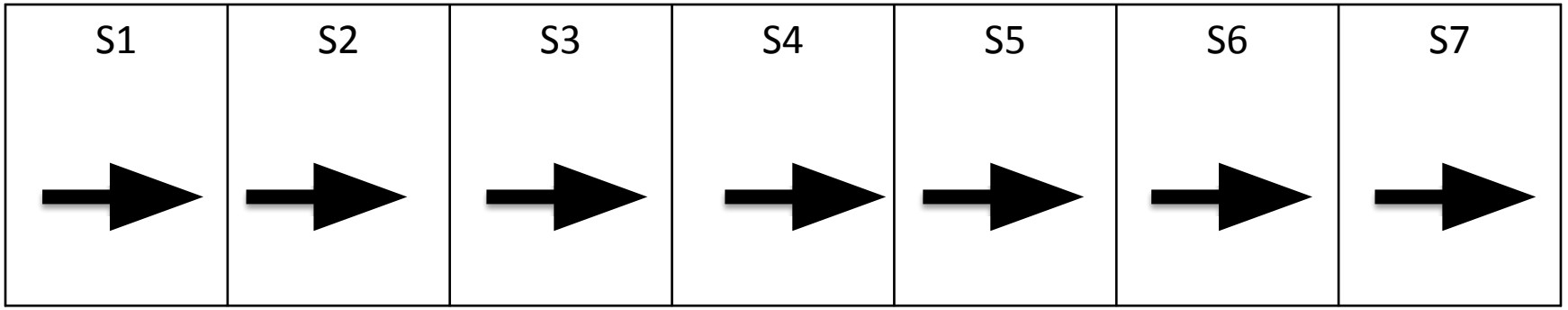
- Evaluation
  - Estimate/Predict the expected rewards from following a given policy
- Control
  - Optimization: find the best policy

Example: Simple Mars Rover Policy Evaluation



- Policy represented by arrows
- π(S1)=π(S2)=...π(S7)=TR
- γ=0
- What is the value of this policy?

Example: Simple Mars Rover Policy Control



- γ=0
- What is the policy that optimizes the expected discounted sum of rewards?

Course Outline

- Markov decision processes & planning
- Model-free policy evaluation
- Model-free control
- Value function approximation & Deep RL
- Policy Search
- Exploration
- Advanced Topics
- See website for more details

Summary

- Overview about reinforcement learning
- Course logistics
- Introduction to sequential decision making under uncertainty