香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

# Reinforcement Learning for Smart Building Control

Zhe (Walter) Wang

Department of Civil & Environmental Engineering
HKUST

# Acknowledgement

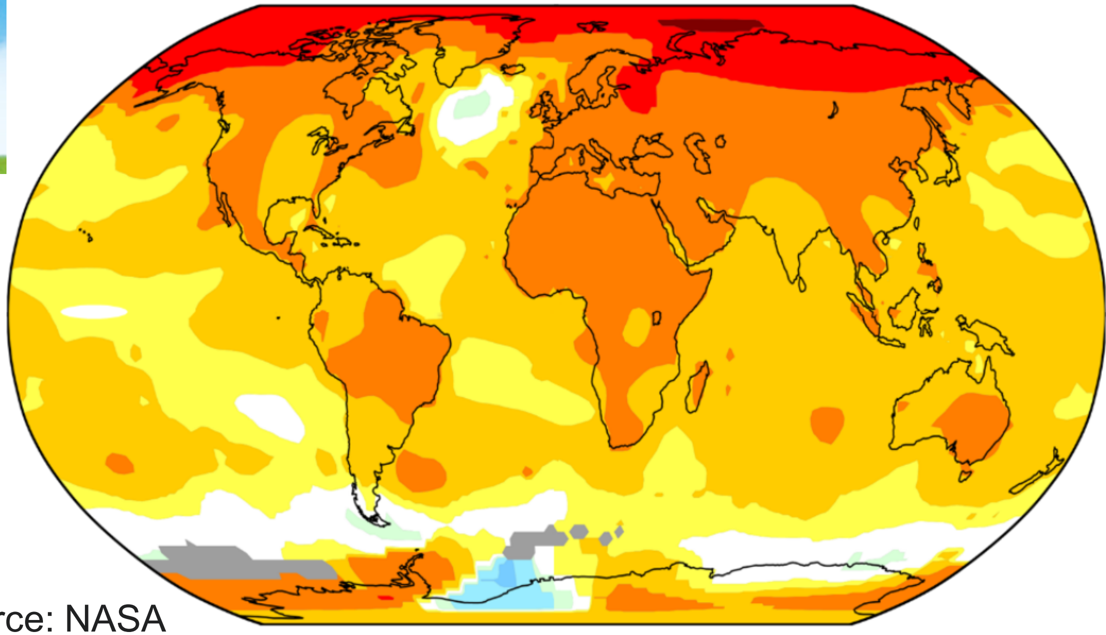
- I was inspired a lot from the following resources
  - David Silver, Reinforcement Learning, University College London COMPM050/COMPGI13, https://www.davidsilver.uk/teaching/
  - Sergey Levine, Deep Reinforcement Learning, University of California Berkeley CS285, https://rail.eecs.berkeley.edu/deeprlcourse/

- *All resources (videos and slides) for the above two courses are open source and free for download*
- *I strongly recommend you to take a look if you are interested in this topic*

# Building Energy System

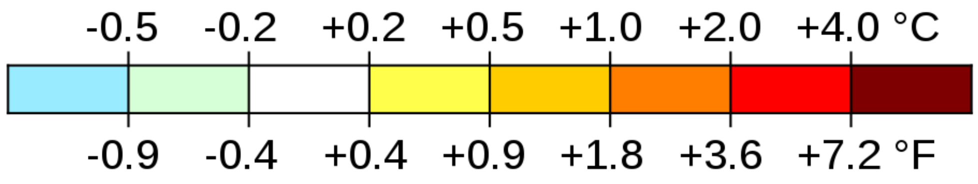
- Building is important
  - Building is a significant energy consumer and carbon emitter
    - 40% in U.S./U.K.
    - 30% in China
  - A source of enormous untapped efficiency potential

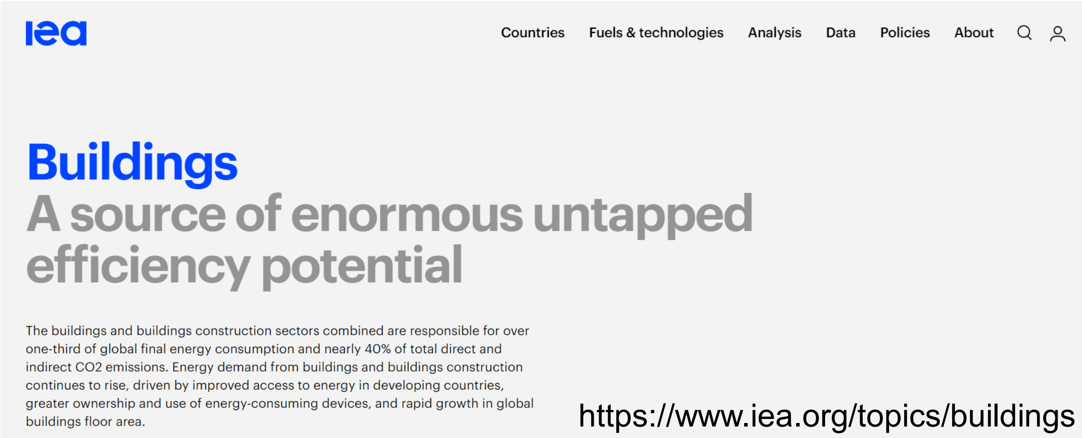
Temperature change in the last 50 years



Source: NASA

2011-2020 average vs 1951-1980 baseline



iea

Countries Fuels & technologies Analysis Data Policies About

## Buildings

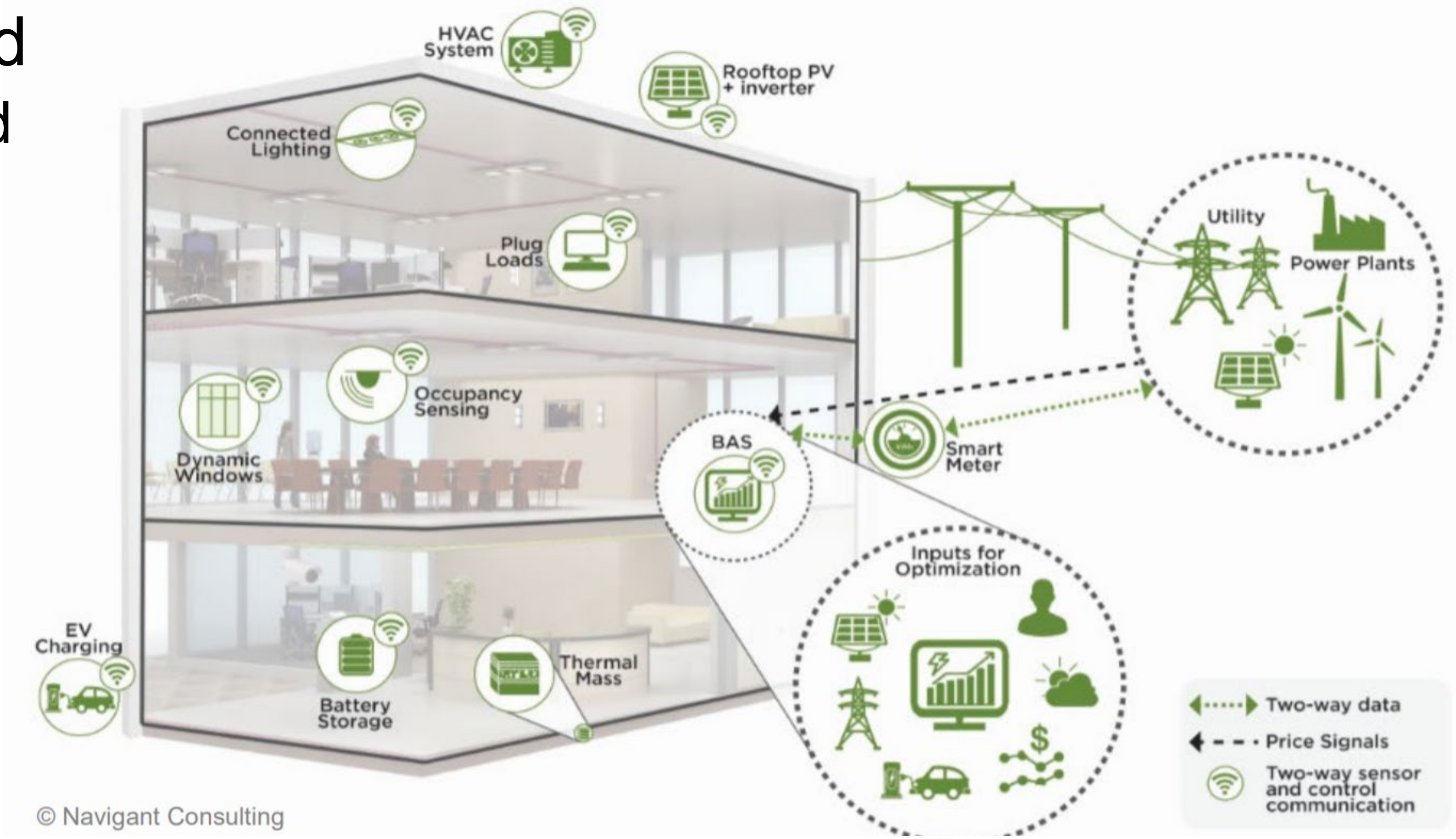
### A source of enormous untapped efficiency potential

The buildings and buildings construction sectors combined are responsible for over one-third of global final energy consumption and nearly 40% of total direct and indirect CO2 emissions. Energy demand from buildings and buildings construction continues to rise, driven by improved access to energy in developing countries, greater ownership and use of energy-consuming devices, and rapid growth in global buildings floor area.

https://www.iea.org/topics/buildings

# Building Energy System

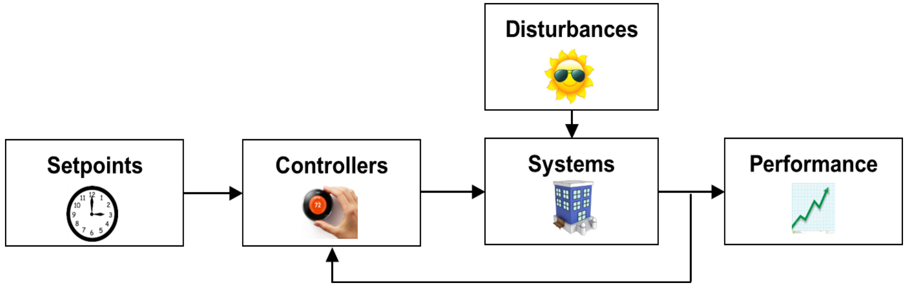
- Building is complicated
  - Complex electrical and thermal systems
    - HVAC
    - Electrical vehicle
    - Battery
    - PV
    - Human interaction




Neukomm, M., Nubbe, V. and Fares, R., 2019. Grid-interactive efficient buildings technical report series: Overview of research challenges and gaps.

# Building control

- Operate the building as we wish
  - Guarantee comfort
  - Enhance energy efficiency
  - Grid-interactive




Scheduled setpoints and PID

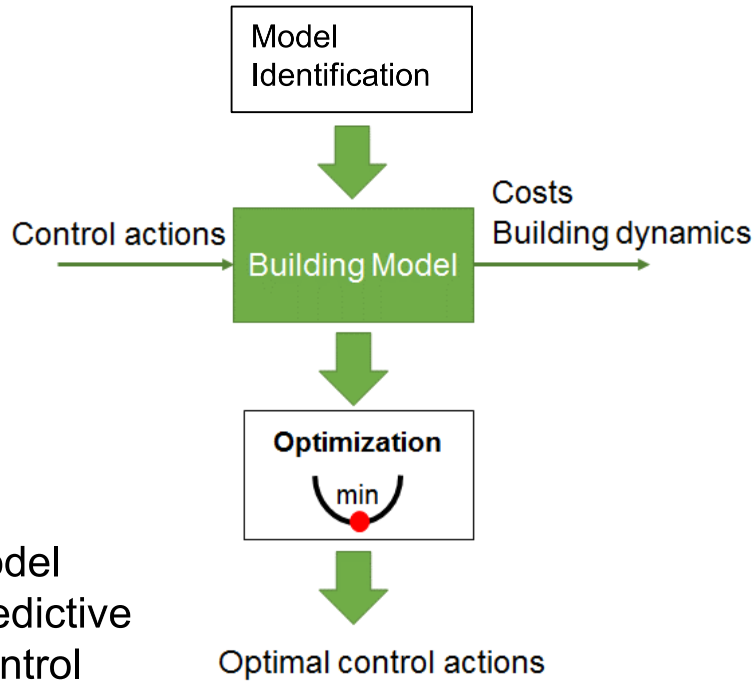
1900s

**Rule-based control (RBC)**

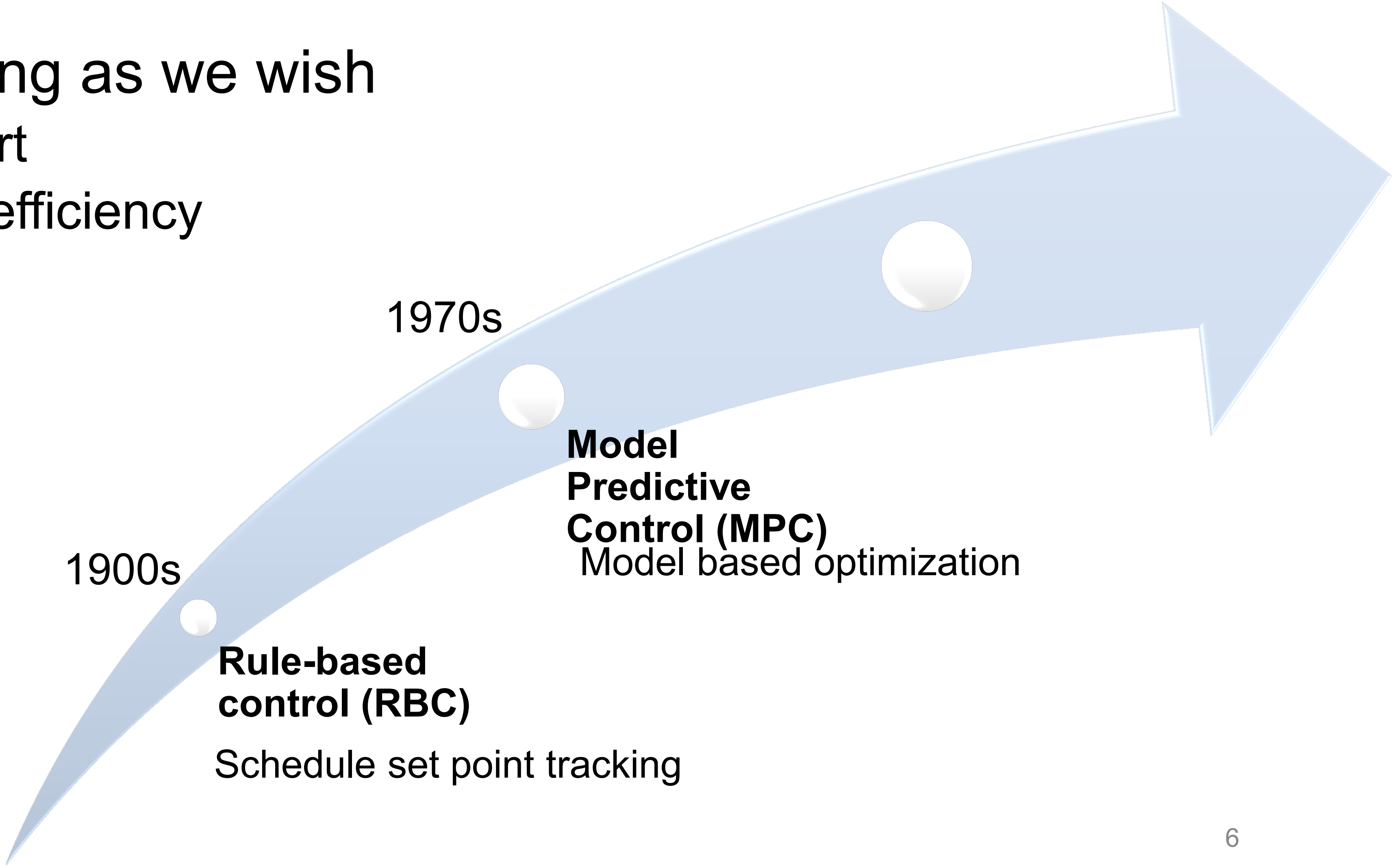
Schedule set point tracking

# Building control

- Operate the building as we wish
  - Guarantee comfort
  - Enhance energy efficiency
  - Grid-interactive



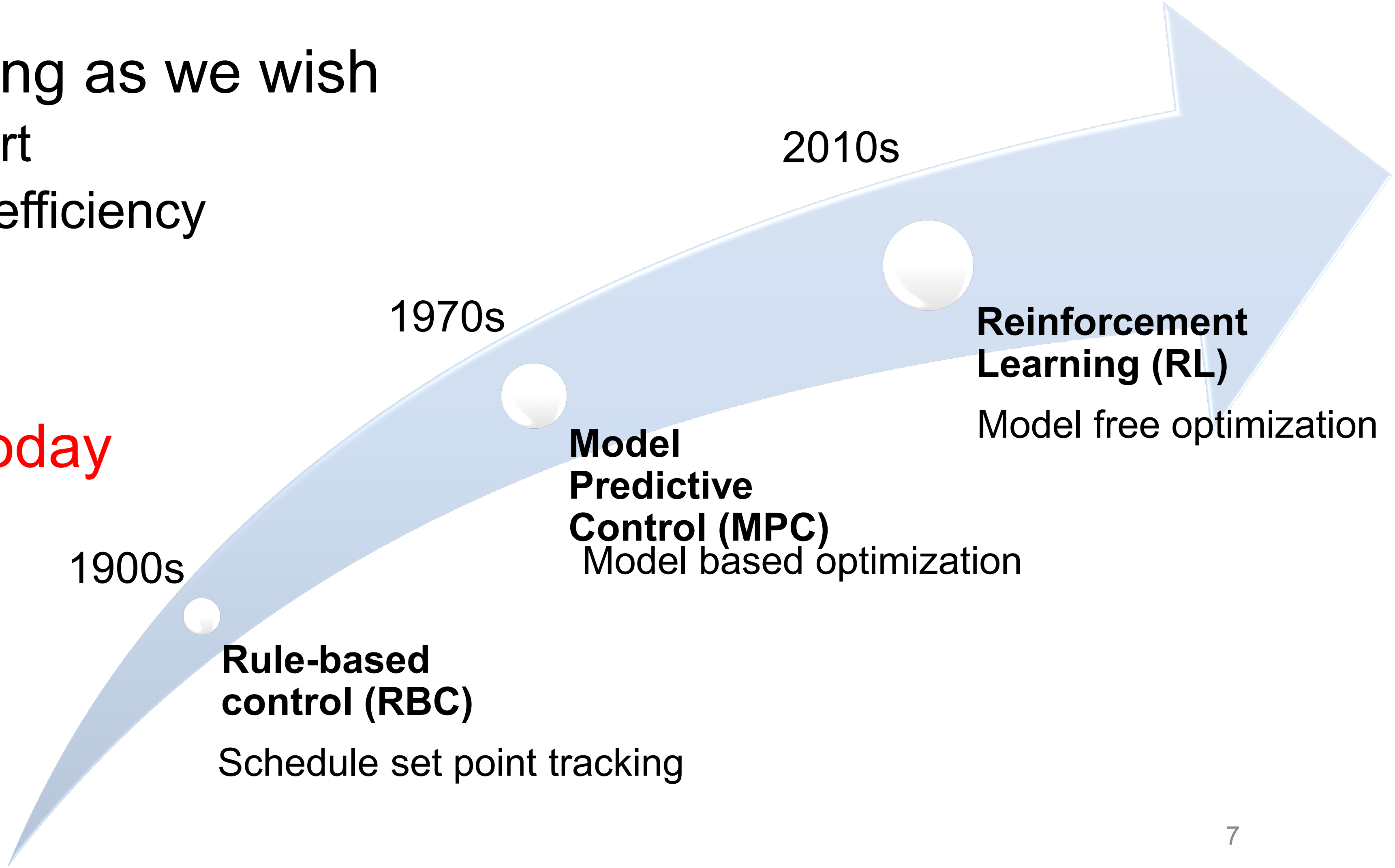

Model Predictive Control



# Building control

- Operate the building as we wish
  - Guarantee comfort
  - Enhance energy efficiency
  - Grid-interactive

- RL: The topic of today



# Content

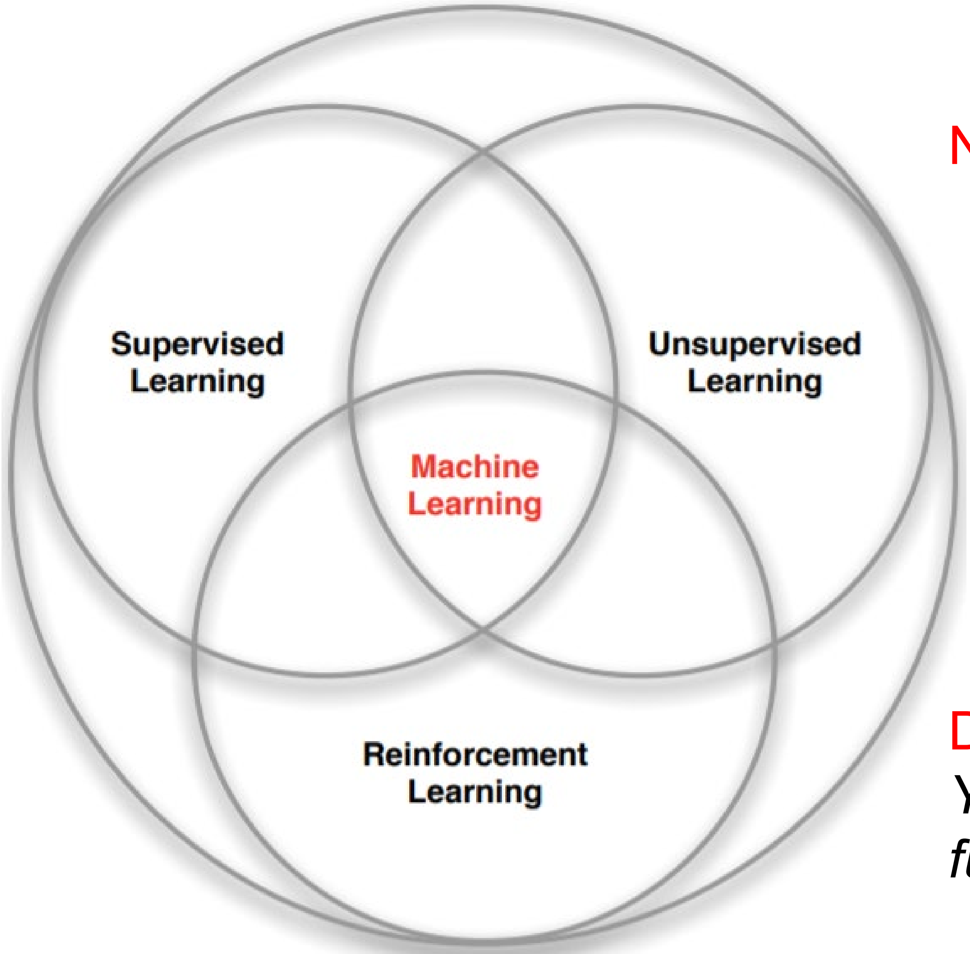
- What is Reinforcement Learning
- Framework of RL
- Math behind RL
  - Markovian Decision Process
  - Bellman Equation
- Q-learning

# Reinforcement Learning

- A branch of Machine Learning for dynamic decision making

Immediate rewards

No rewards

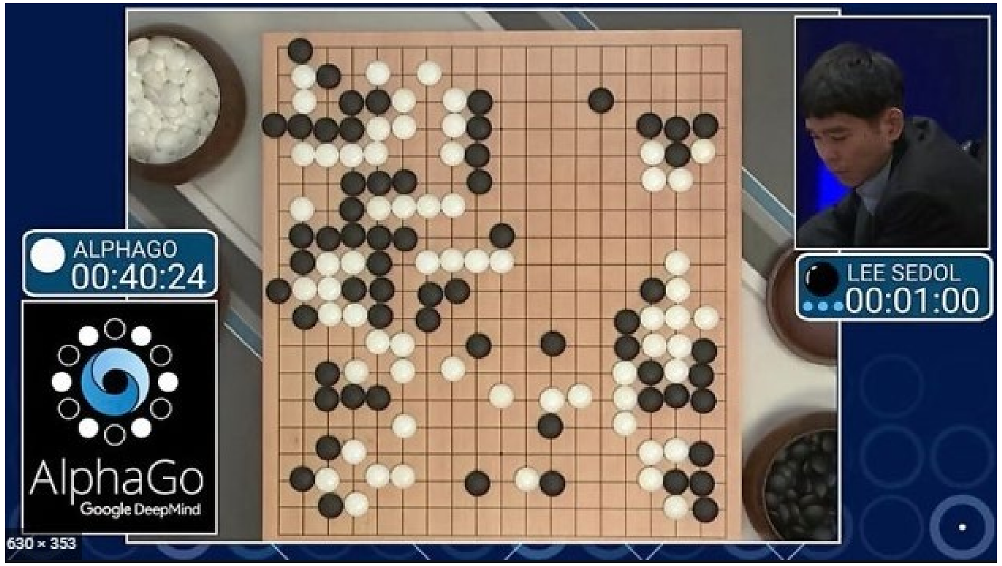



Delayed rewards
*Your action will affect future states*

# Reinforcement Learning

- A branch of Machine Learning for dynamic decision making
- Successful application in many fields

*AlphaGo* beat *Lee Sedol*, 2016

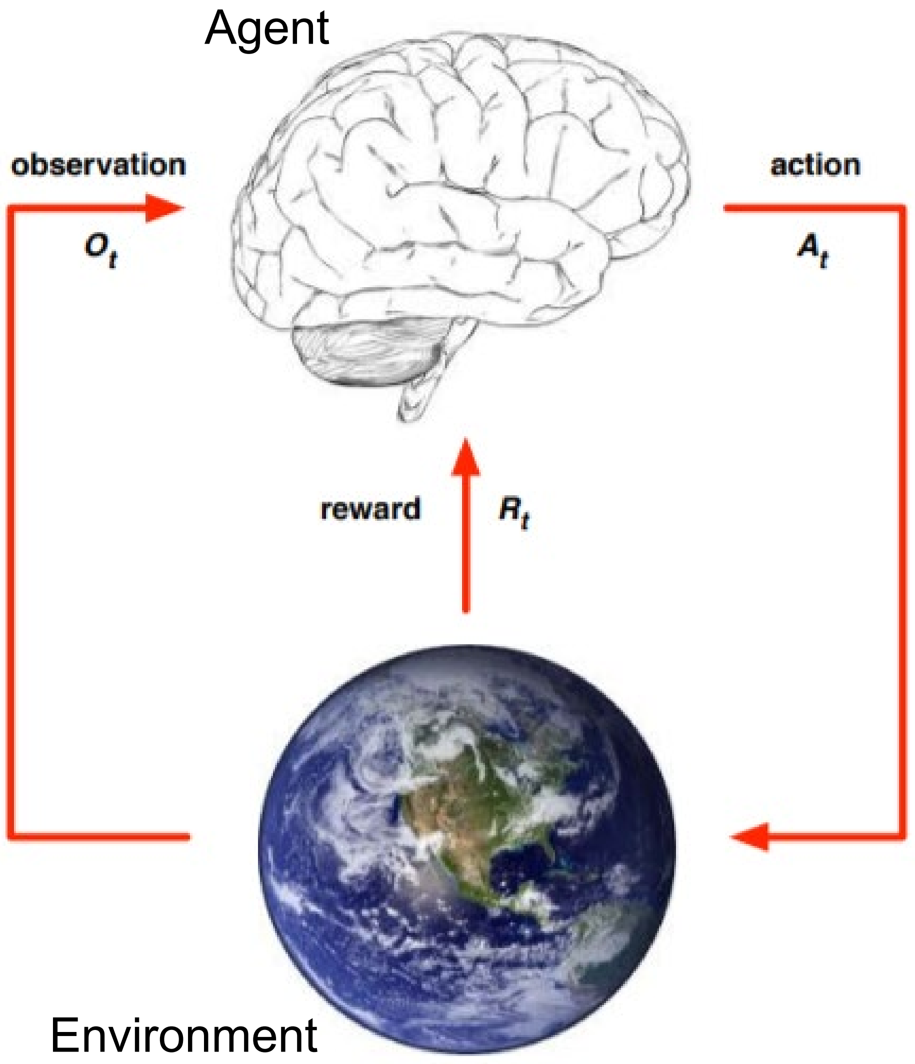



*OpenAI Five* beat *OG*, 2019


https://www.youtube.com/watch?v=tfb6aEUMC04

# Reinforcement Learning

- At each time step
  - The agent
    - Observe the state $S_t$
    - Calculate the action $A_t$
  - The environment
    - Execute the action $A_t$
    - Emit the reward $R_t$
    - Emit the new state $S_{t+1}$ ($S'$)
- Move to the next time step
  - Very typical *for* loop




David Silver, Reinforcement Learning, University College London COMPM050/COMPGI13

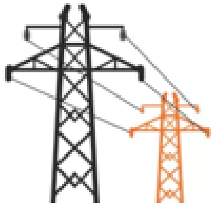
# Reinforcement Learning for building control

***Control actions***
- HVAC system

***Target building***
- NERSC data center

***Rewards***
- Energy consumption
- Utility bills

Wang Hall, LBNL

Smart grid
- Utility price
- DR signals

Energy storage

***DRL Controller***
- Different learning algorithms and approaches

***States from observation***
- Indoor environment
- Outdoor environment
- Grid signals

# Markov Property

- The future is independent of the past given the present

**Definition**

A state $S_t$ is *Markov* if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

- The state captures all relevant information from the history
- Once the state is known, the history may be thrown away
- The state is a sufficient statistic of the future

# Markov Process (MP)

- A Markov Process (or Markov Chain) is memoryless random process, i.e. a sequence of random states with the Markov property

**Definition**

A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$

# Markov Process (MP)


- Is the following state a Markov Process?
  - $[x_t]$
  - $[x_t, v_t, a_t]$
- The process will become a Markov Process if you can measure all the relevant information
- What if you cannot measure all the relevant information?
  - Partially Markov Process
    - Very common in practice
      - The problem is not properly formed
      - Some states are not measurable
    - Out of the scope of this lecture

# Markov Reward Process (MRP)

- A Markov Reward Process is a Markov chain with values

**Definition**

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}\left[R_{t+1} \mid S_t = s\right]$
- $\gamma$ is a discount factor, $\gamma \in [0, 1]$

# Total Return

**Definition**

The *return* $G_t$ is the total discounted reward from time-step $t$.

$$G_t = R_{t+1} + \gamma R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Why discount?
  - Current reward worth more than future reward (interests rate)
  - Future is associated with uncertainty
  - Mathematically stable

# Markov Decision Process (MDP)

- A Markov decision process (MDP) is a Markov reward process with decisions.

**Definition**

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
- $\gamma$ is a discount factor $\gamma \in [0, 1]$.

# Policy

**Definition**

A *policy* $\pi$ is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}\left[A_t = a \mid S_t = s\right]$$

- Fully define the behavior of an agent

# Value function

- v-function
  - Defines on state

**Definition**

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state $s$, and then following policy $\pi$

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$

- q-function
  - Defines on state, action pair

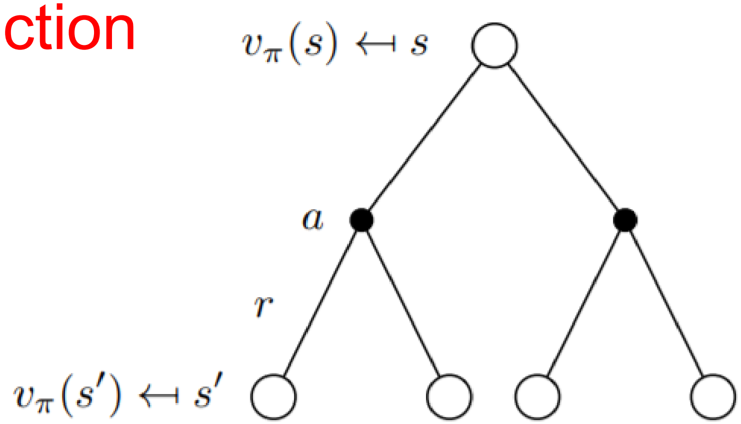
**Definition**

The *action-value function* $q_\pi(s, a)$ is the expected return starting from state $s$, taking action $a$, and then following policy $\pi$

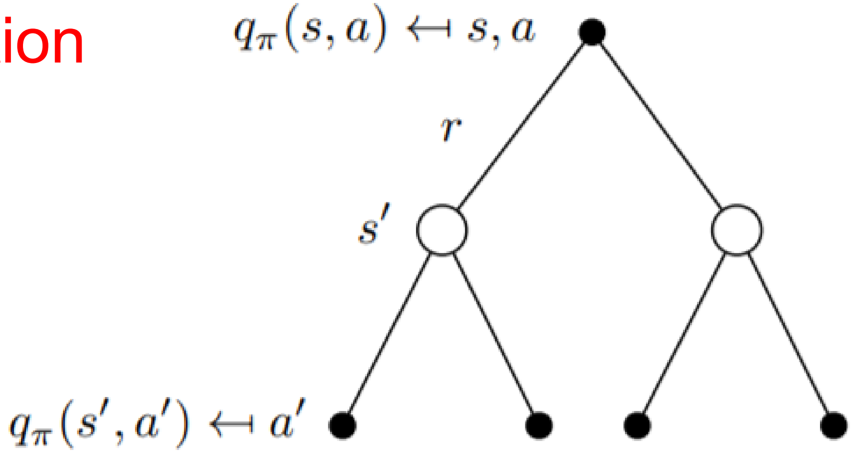
$$q_\pi(s, a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

# Bellman Equation

V-function




Q-function




$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left( \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s',a')$$

# Q-function

- Why we bother to learn MDP and BE?
  - We want to know the policy
- If we know V-function, can we extract policy?
  - No!
- If we know Q-function, can we extract policy?
  - Yes!
  - Just select the action with the largest Q-value

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_*(s,a) \\ 0 & \textit{otherwise} \end{cases}$$

- The next question
  - How can we solve the BE to learn Q-function?

# Bellman Optimality Equation

- BE $$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s',a')$$
- BOE $$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} q_*(s',a')$$
- Many iterative solution methods
  - Value Iteration
  - Policy Iteration
  - Q-learning
  - SARSA

# Q-learning

- How to represent the Q function
  - Table: Q-table
  - Neural Network: Deep Q learning
- How to learn the Q function
  - Bellman Optimality Equation

$$q_*(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} q_*(s', a')$$

# Q-learning

- Key: Update the Q-function towards the "true" value

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize $S$
    Repeat (for each step of episode):
        Choose $A$ from $S$ using policy derived from $Q$ (select the best action)
        Take action $A$, observe $R$, $S'$
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
        $S \leftarrow S'$
    until $S$ is terminal

$$(1-\alpha)Q(S,A) + \alpha\boxed{\left(R + \gamma \max_a Q(S',a)\right)}$$

"True" Q-value

- The immediate reward is true, contains more information

# Q-learning

- The maze problem
  - Exit the maze as soon as possible
    - The reward of each step is -1
    - Ignore discount factor

| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | 0 |
| Right | 0 | | 0 |
| Left | | 0 | |
| Down | 0 | 0 | |

| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | 0 |
| Right | -1 | | 0 |
| Left | | -2 | |
| Down | -1 | -1 | |

| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | 0 |
| Right | -1 | | 0 |
| Left | | -2 | |
| Down | -1 | -1 | |

# Q-learning

New episode

| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | 0 |
| Right | -1 | | 0 |
| Left | | -2 | |
| Down | -1 | -1 | |

| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | -2 |
| Right | -2 | | -1 |
| Left | | -2 | |
| Down | -1 | -1 | |

| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | -2 |
| Right | -2 | | -1 |
| Left | | -2 | |
| Down | -1 | -1 | |

New episode

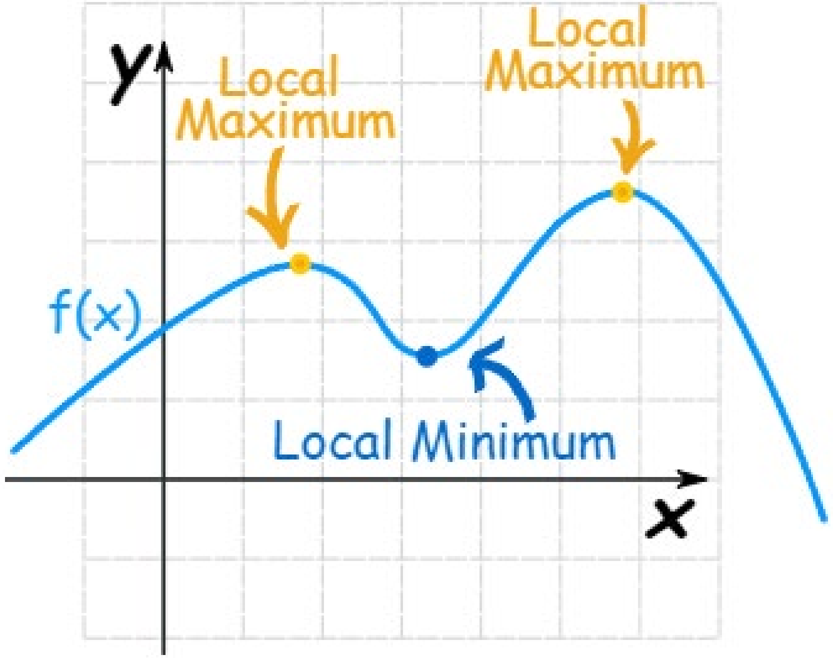
| | |
|---|---|
| S | 1 |
| 2 | E |

| | S | 1 | 2 |
|---|---|---|---|
| Up | | | -3 |
| Right | -2 | | -1 |
| Left | | -3 | |
| Down | -2 | -1 | |

This Q table can guide you out of this maze!

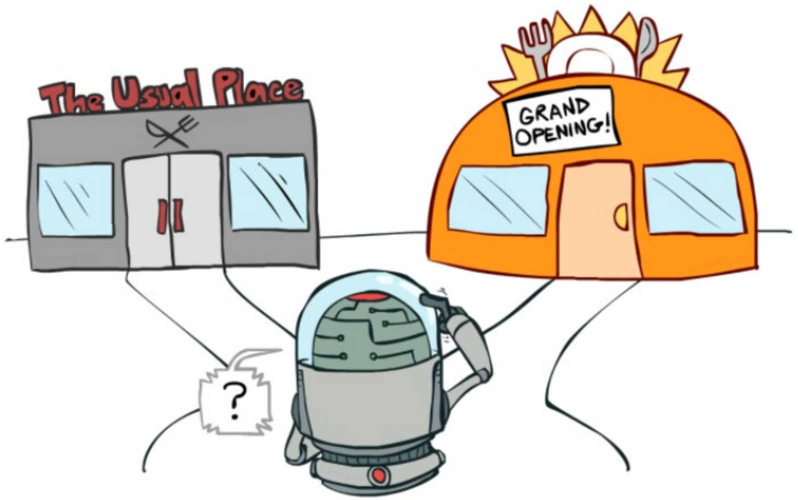
# Exploration versus Exploitation

- Exploit
  - Select the best action you have tested
  - Make fully use of the previous experience
  - Problem: stuck in local maximum
- Explore
  - Explore untested actions might help you make better selection in the future
  - Even if it reduces your immediate rewards
- Epsilon-greedy




$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a) \\ \epsilon/m & \text{otherwise} \end{cases}$$

# Exploration versus Exploitation




- Exploit
  - Select the best action you have tested
  - Make fully use of the previous experience
  - Problem: stuck in local maximum
- Explore
  - Explore untested actions might help you make better selection in the future
  - Even if it reduces your immediate rewards
- Epsilon-greedy

$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a) \\ \epsilon/m & \text{otherwise} \end{cases}$$

Go to your favorite restaurant

Try new restaurant

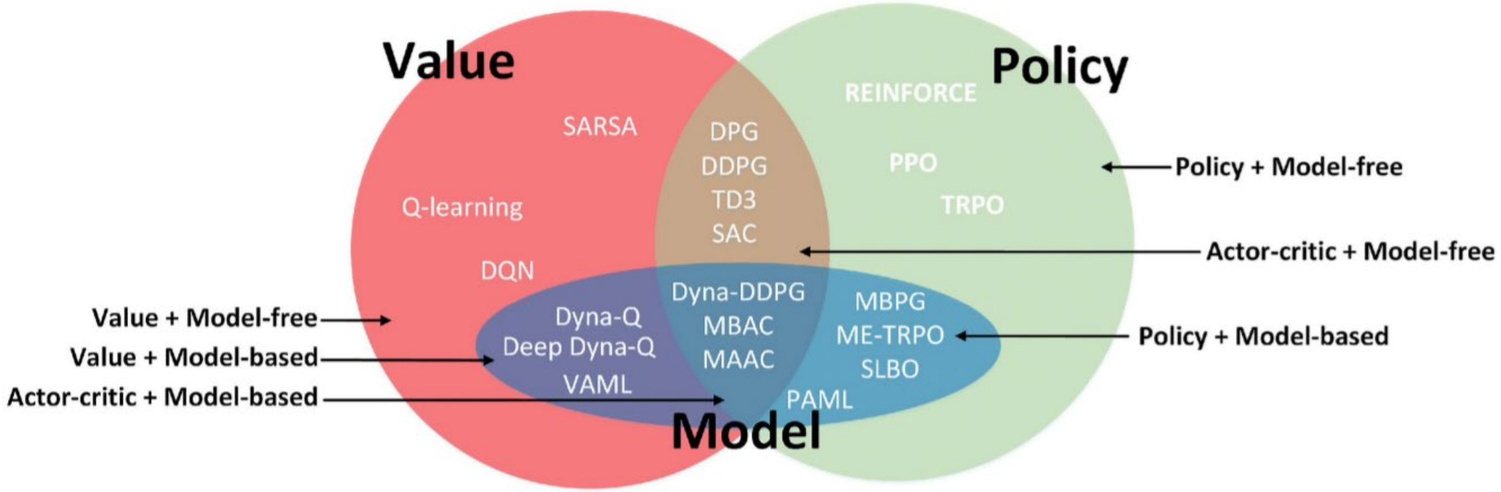
For the most time, go to your favorite;
Try new restaurant from time to time

# Summary

- Three generation of building control
- Reinforcement Learning: framework
- MDP and Bellman Equation
- Q-table learning
- Exploration versus exploitation
  - Epsilon-greedy

# Summary

- RL is a very fancy and fast evolving subject
- Q-table learning is one of the many RL algorithms




- Epsilon-greedy is one of many methods for exploration versus exploitation