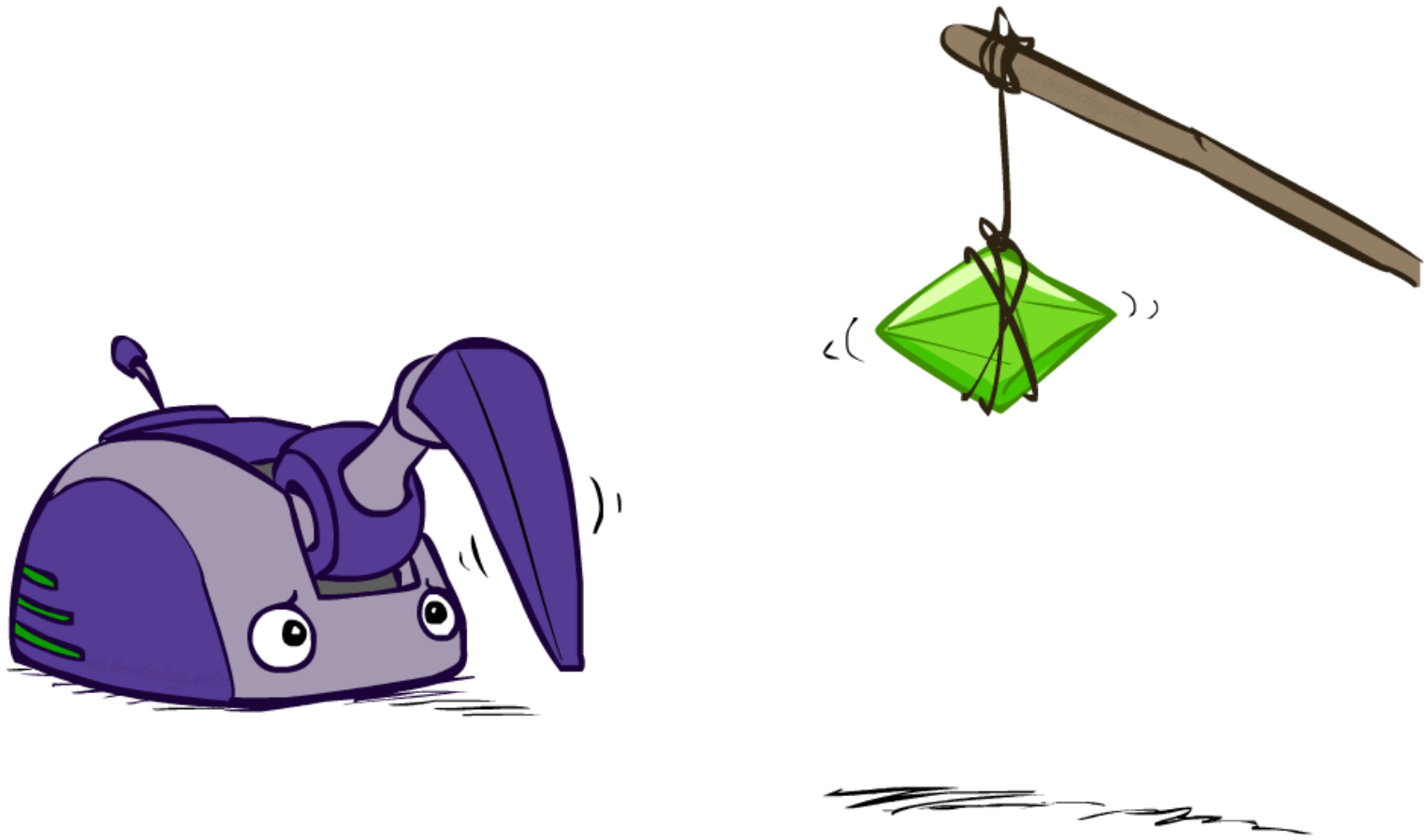
CS 188: Artificial Intelligence

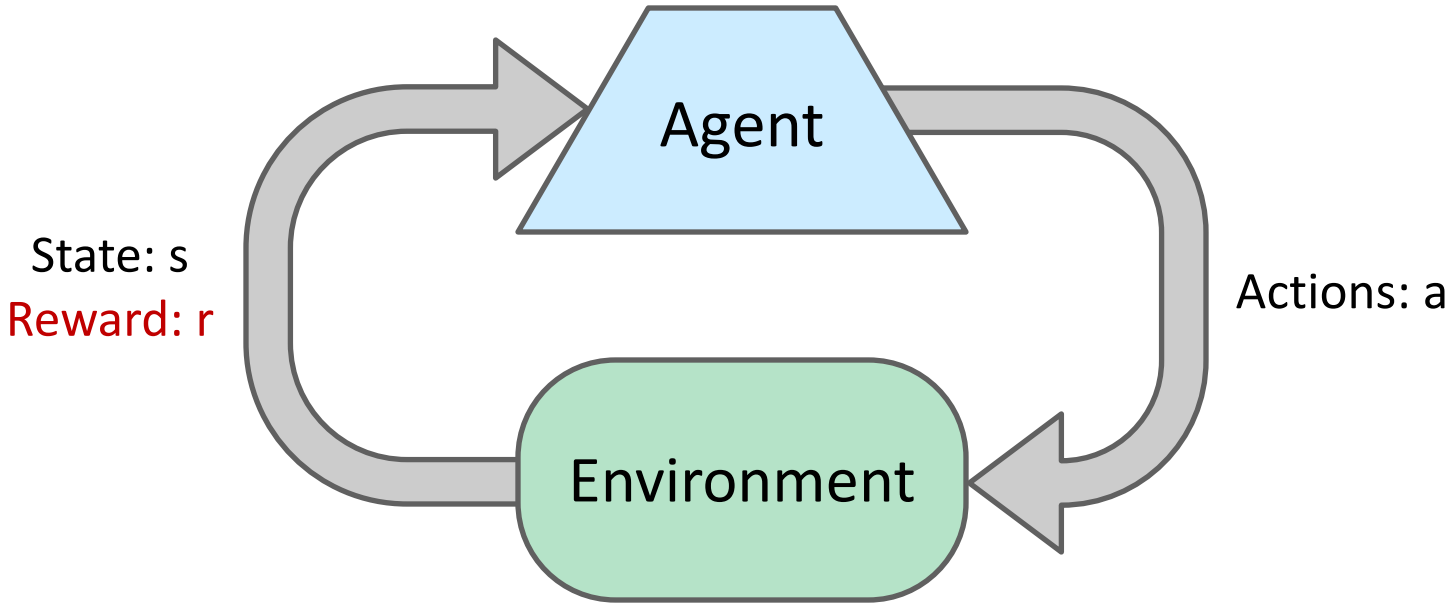
Reinforcement Learning

University of California, Berkeley

Reinforcement Learning



Reinforcement Learning



- Basic idea:
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the reward function
  - Must (learn to) act so as to maximize expected rewards
  - All learning is based on observed samples of outcomes!

Example: Learning to Walk

Initial

A Learning Trial

After Learning [1K Trials]

[Kohl and Stone, ICRA 2004]

Example: Learning to Walk

Initial

[Kohl and Stone, ICRA 2004]

[Video: AIBO WALK – initial]

Example: Learning to Walk

Training

[Kohl and Stone, ICRA 2004]

[Video: AIBO WALK – training]

Example: Learning to Walk

Finished

[Kohl and Stone, ICRA 2004]

[Video: AIBO WALK – finished]

Example: Sidewinding

[Andrew Ng]

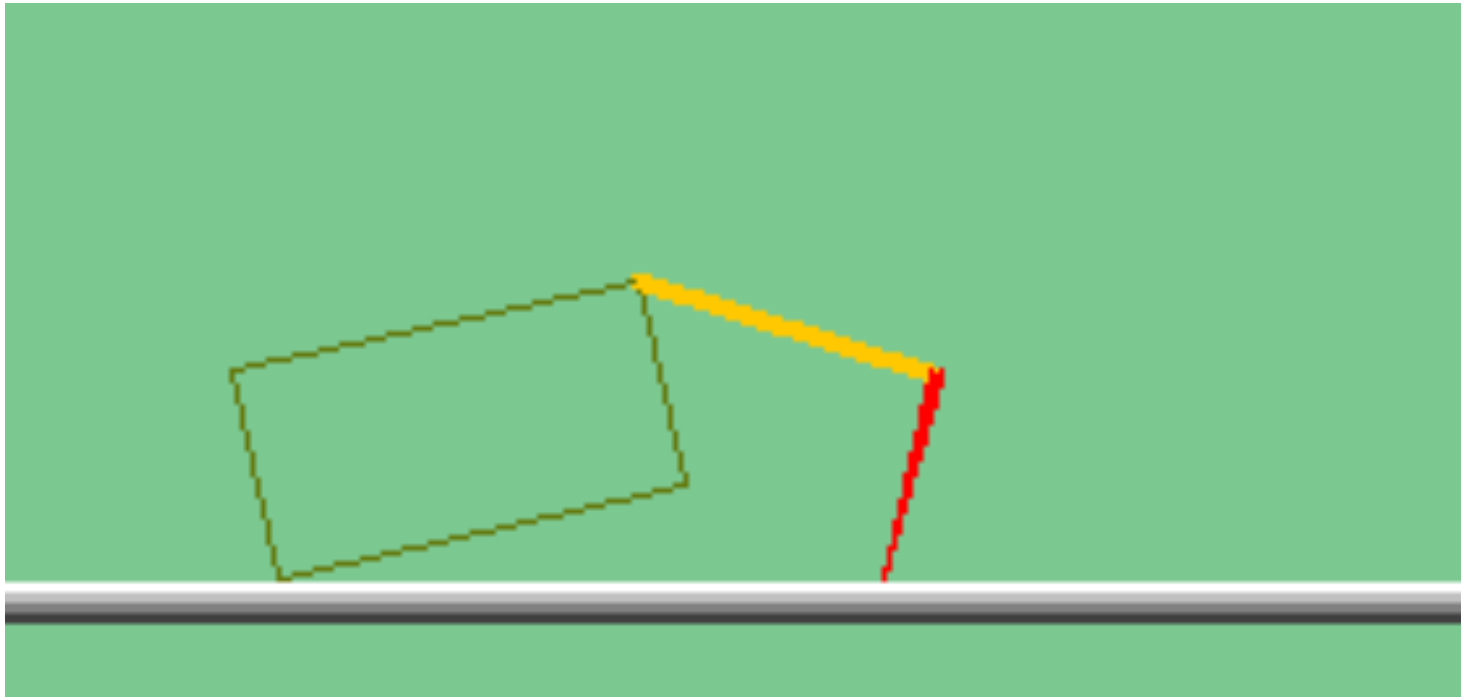
[Video: SNAKE – climbStep+sidewinding]

Example: Toddler Robot

[Tedrake, Zhang and Seung, 2005]

[Video: TODDLER – 40s]

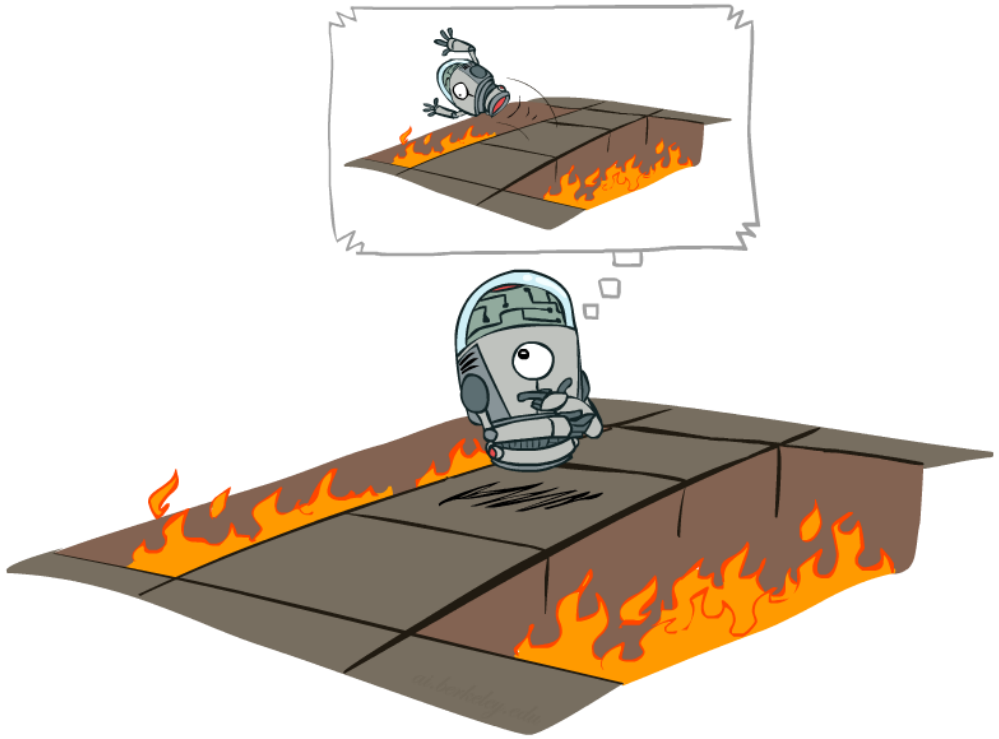
The Crawler!



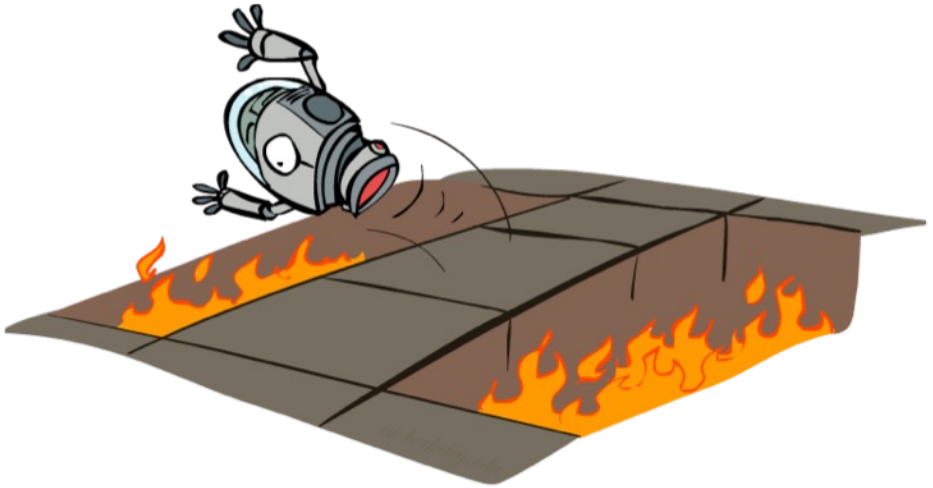
[Demo: Crawler Bot (L10D1)] [You, in Project 3]

Video of Demo Crawler Bot

Offline (MDPs) vs. Online (RL)



Offline Solution



Online Learning

Reinforcement Learning

- Still assume a Markov decision process (MDP):
  - A set of states $s \in S$
  - A set of actions (per state) A
  - A model T(s,a,s')
  - A reward function R(s,a,s')
- Still looking for a policy $\pi(s)$

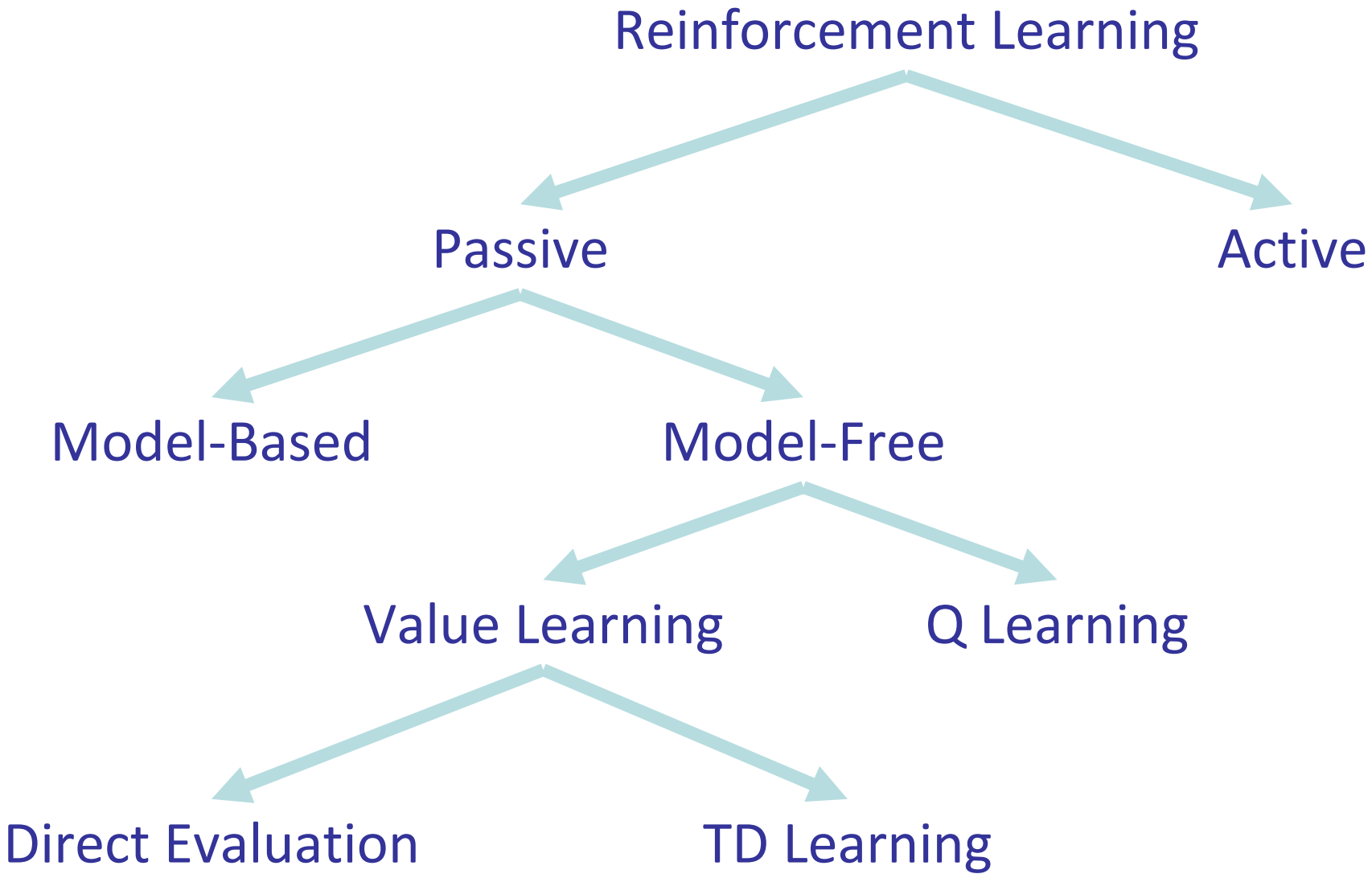
Cool

Warm

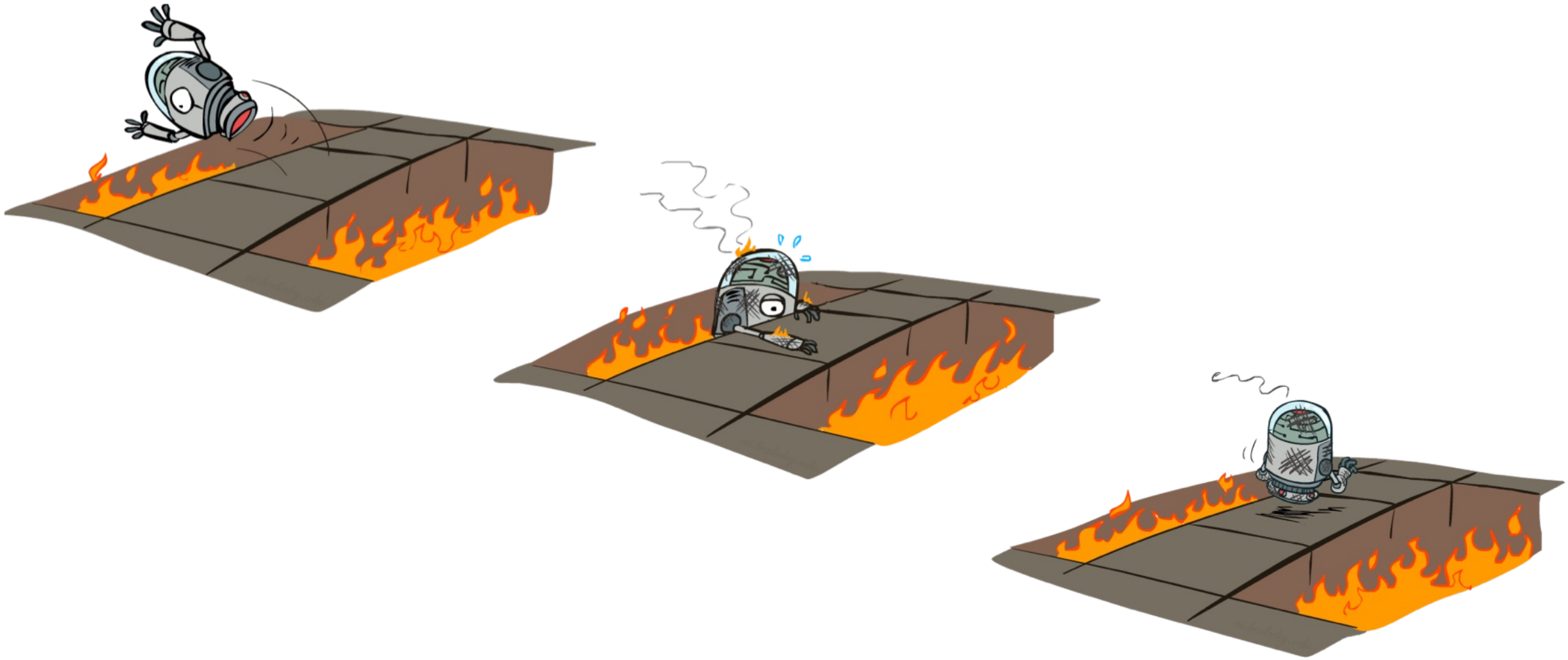
Overheated

- New twist: don't know T or R
  - I.e. we don't know which states are good or what the actions do
  - Must actually try out actions and states to learn

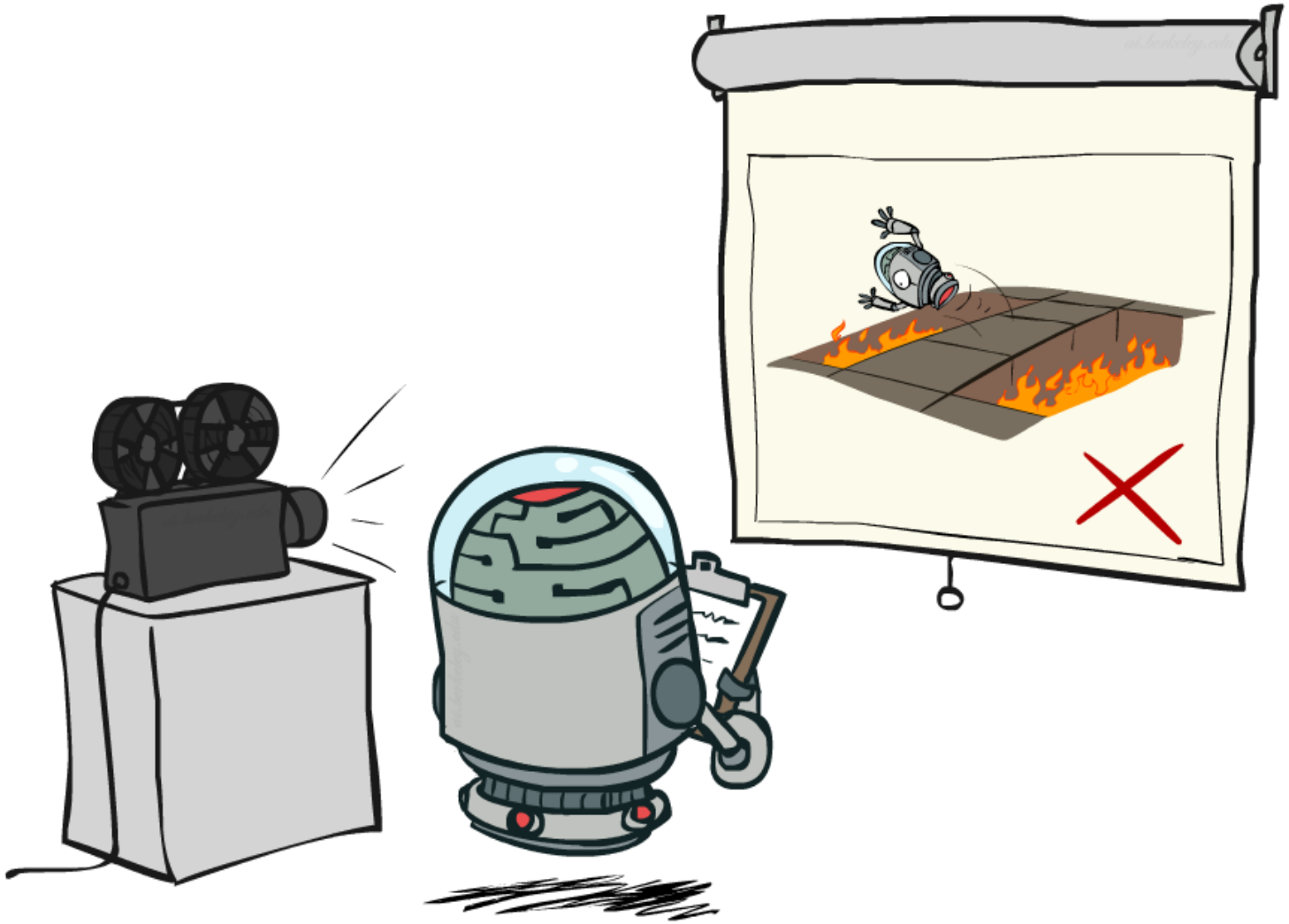
Reinforcement Learning Taxonomy



Active Reinforcement Learning

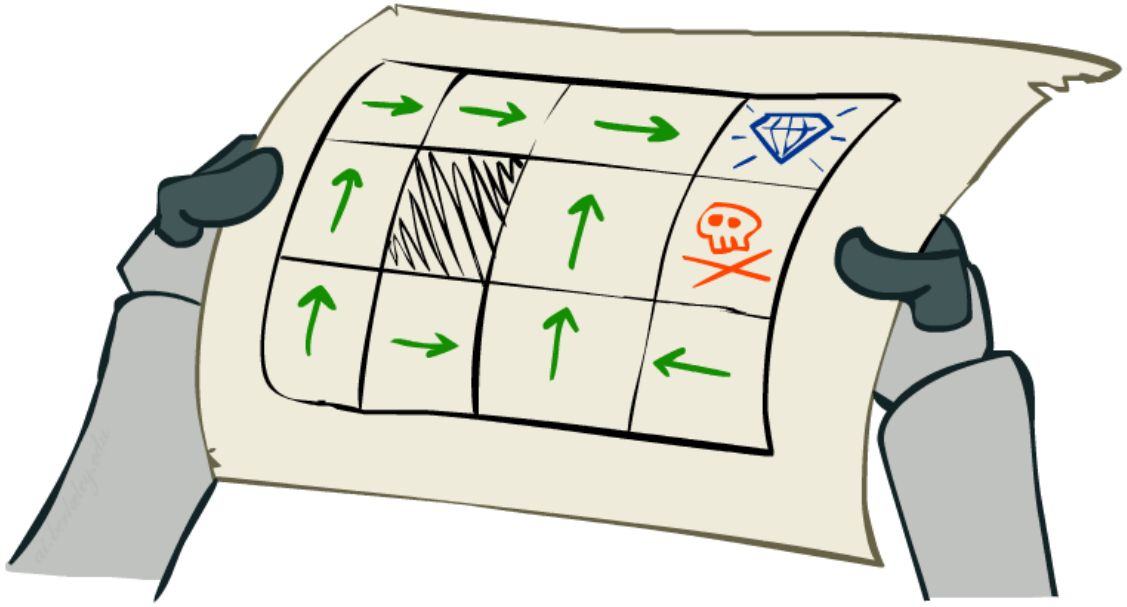


Passive Reinforcement Learning



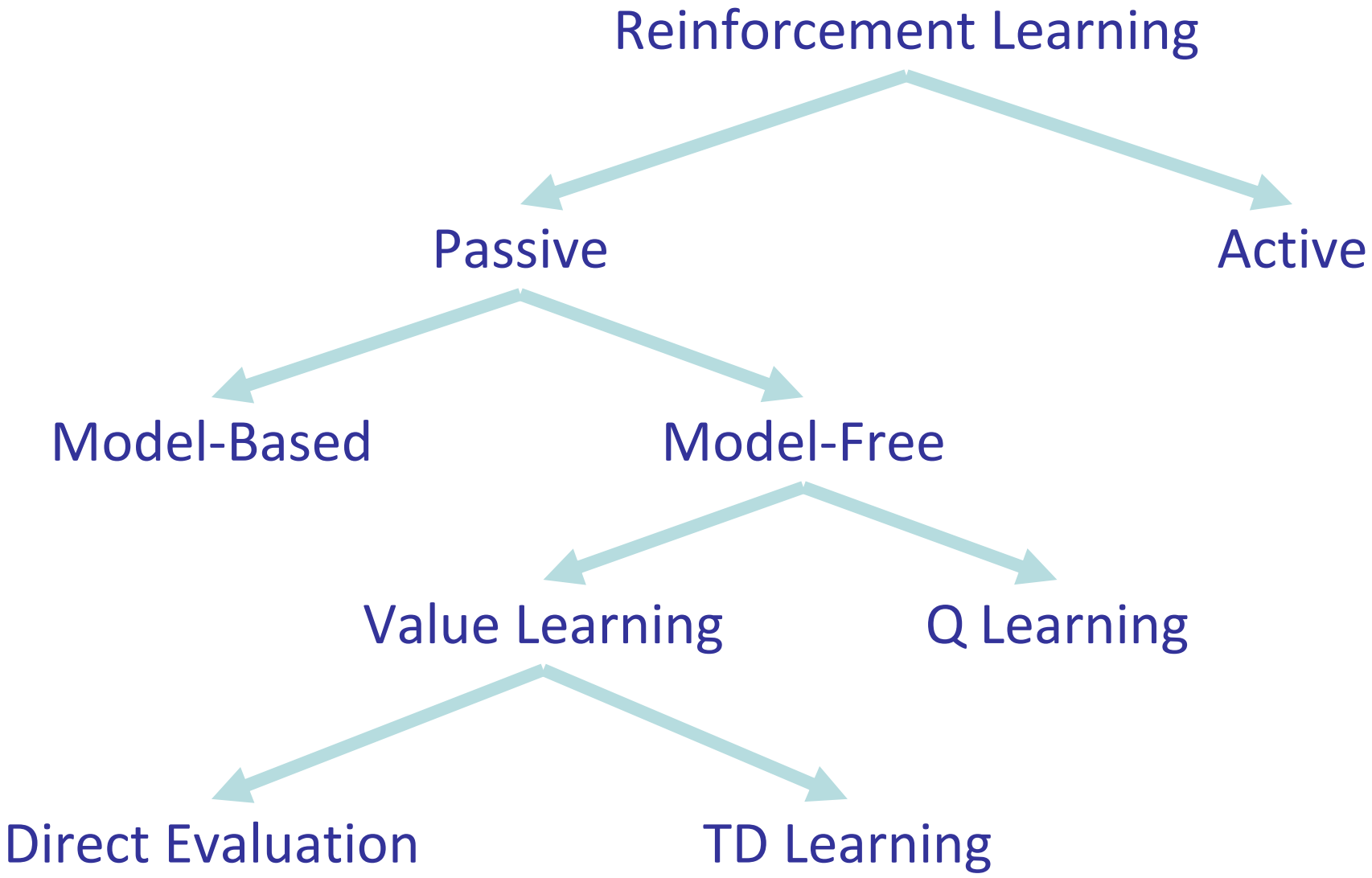
Passive Reinforcement Learning

- Simplified task: policy evaluation
  - Input: a fixed policy $\pi(s)$
  - You don't know the transitions T(s,a,s')
  - You don't know the rewards R(s,a,s')
  - Goal: learn the state values

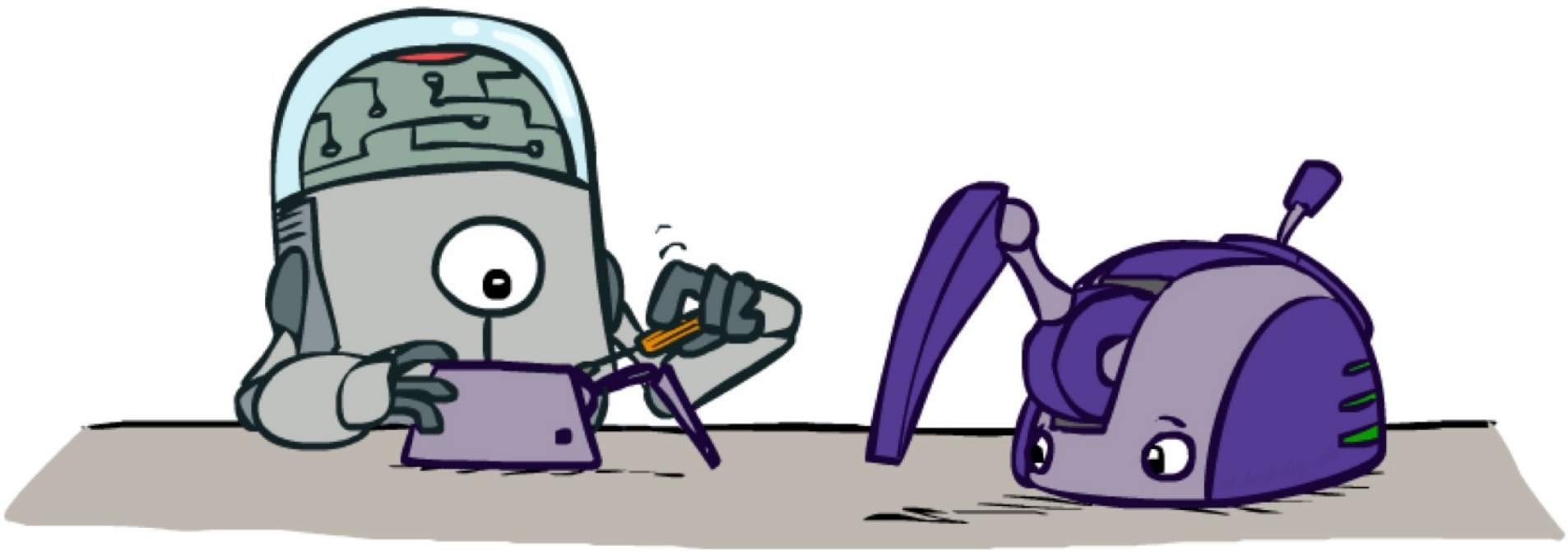


- In this case:
  - Learner is "along for the ride"
  - No choice about what actions to take
  - Just execute the policy and learn from experience
  - This is NOT offline planning! You actually take actions in the world.

Reinforcement Learning Taxonomy



Model-Based Learning



Model-Based Learning

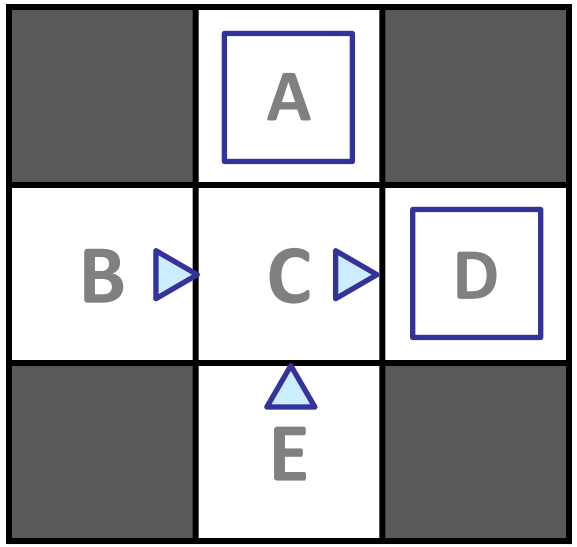
- Model-Based Idea:
  - Learn an approximate model based on experiences
  - Solve for values as if the learned model were correct

- Step 1: Learn empirical MDP model
  - Count outcomes s’ for each s, a
  - Normalize to give an estimate of $\hat{T}(s, a, s')$
  - Discover each $\hat{R}(s, a, s')$ when we experience (s, a, s’)

- Step 2: Solve the learned MDP
  - For example, use value iteration, as before

Example: Model-Based Learning

Input Policy π



Assume: $\gamma = 1$

Observed (s, a, s', R) Transitions

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Learned Model

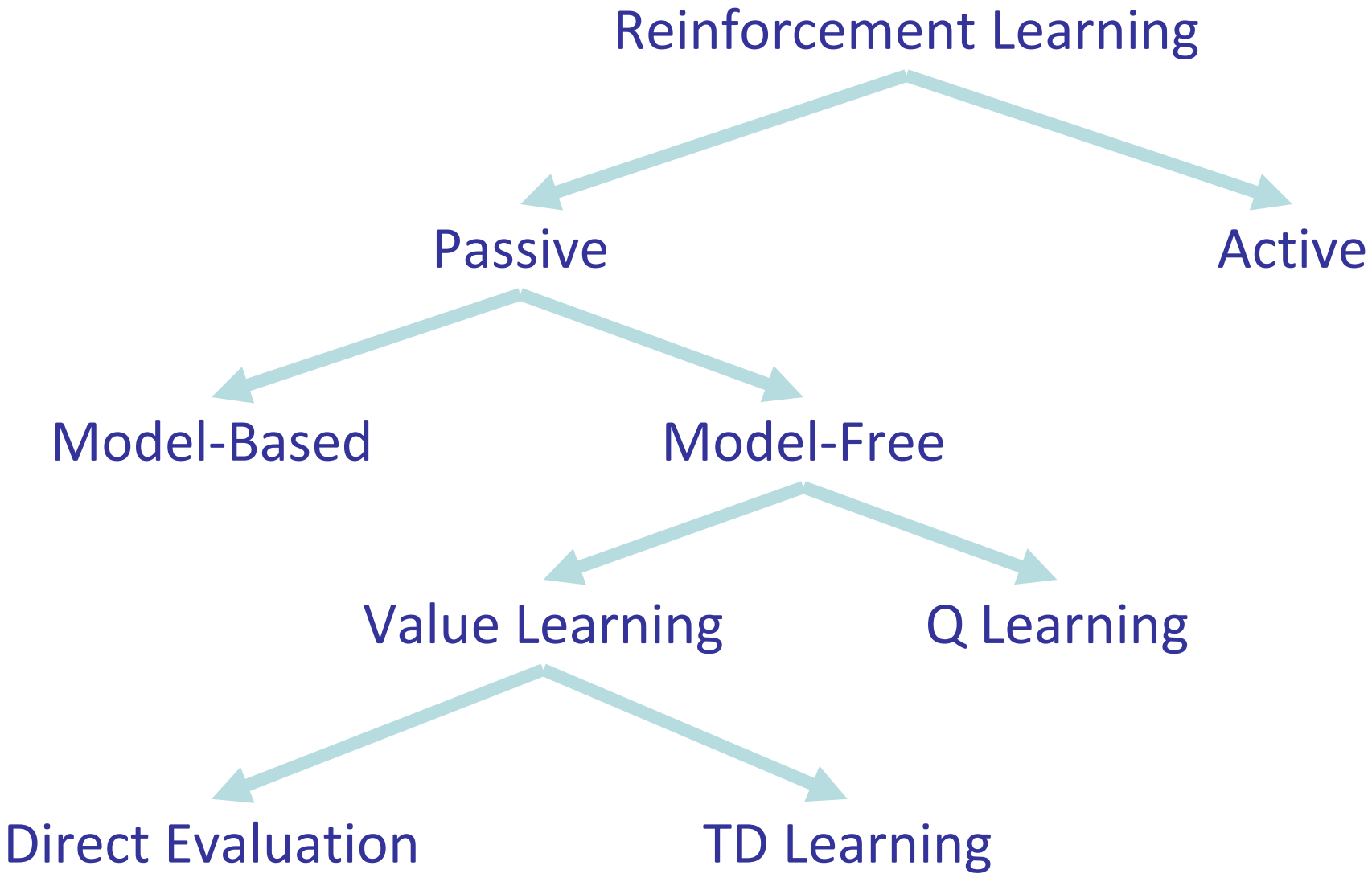
$\hat{T}(s, a, s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
...

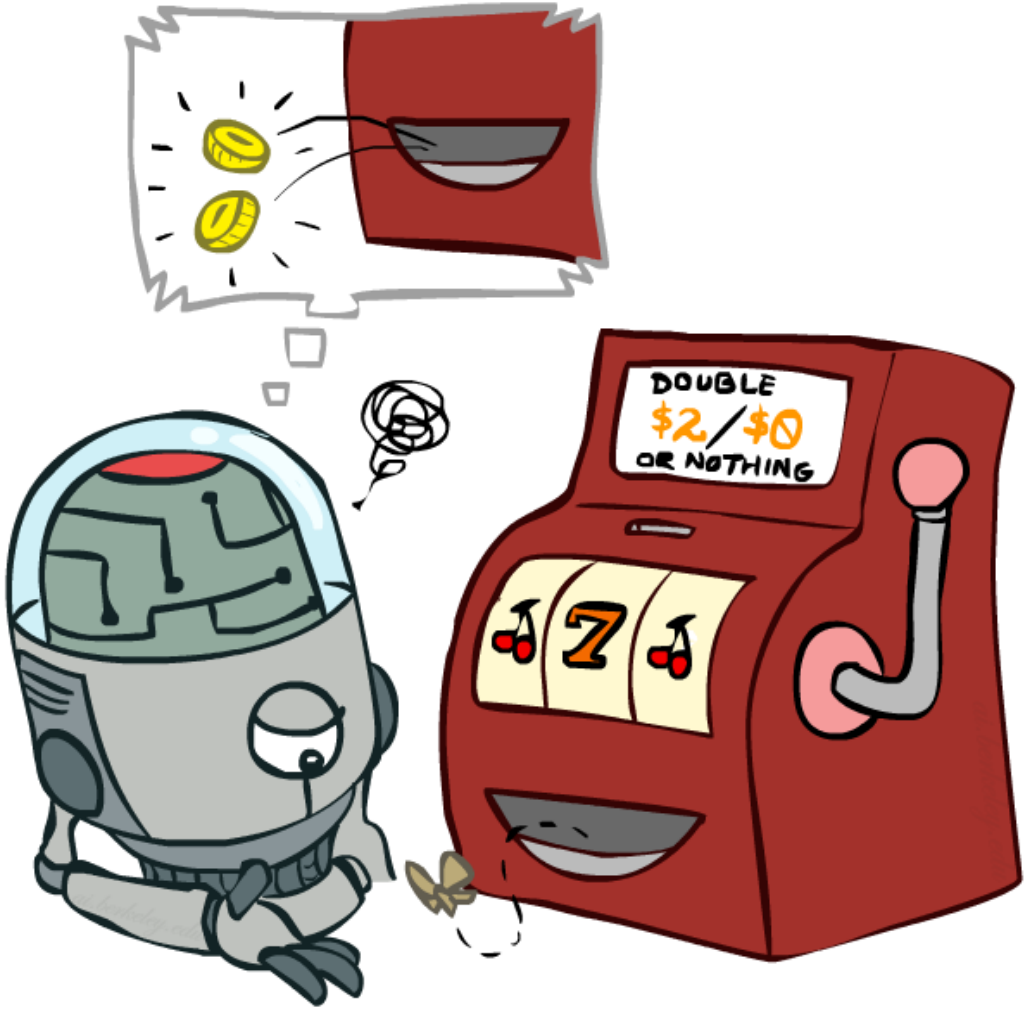
$\hat{R}(s, a, s')$

R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
...

Reinforcement Learning Taxonomy



Model-Free Learning



Direct Evaluation

- Goal: Compute values for each state under π

- Idea: Average together observed sample values
  - Act according to π
  - Every time you visit a state, write down what the sum of discounted rewards turned out to be:

$$sample_i(s) = \sum_t \gamma^t R^t$$

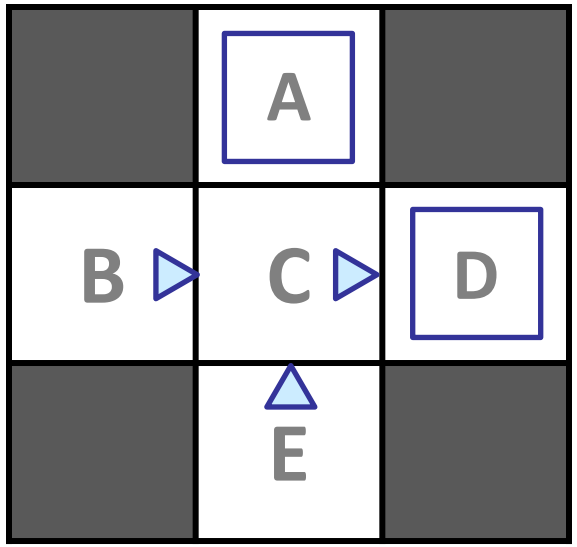
  - Average those samples:

$$V(s) \approx \frac{1}{N} \sum_i sample_i(s)$$

- This is called direct evaluation

Example: Direct Evaluation

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

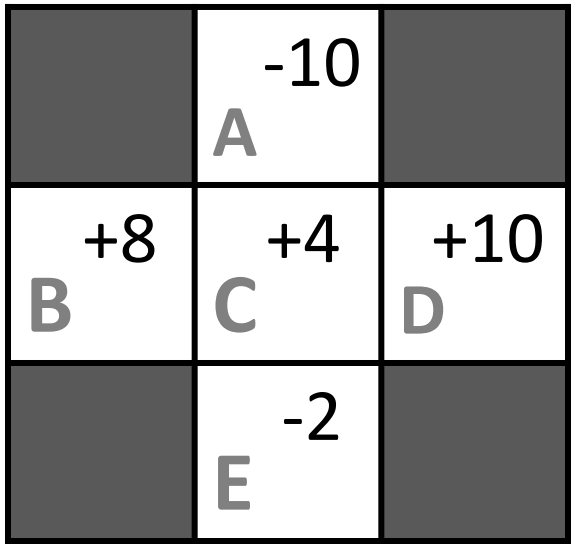
Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values



$$sample_i(s) = \sum_t \gamma^t R^t$$

$$V(s) \approx \frac{1}{N} \sum_i sample_i(s)$$

Quiz: Direct Evaluation

Observed (s, a, s', R) Transitions

Episode 1

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

C, east, A, -1
A, exit, x, -5

Episode 3

B, east, C, -1
C, east, D, -1
D, exit, x, +10

$$sample_i(s) = \sum_t \gamma^t R^t$$

$$V(s) \approx \frac{1}{N} \sum_i sample_i(s)$$

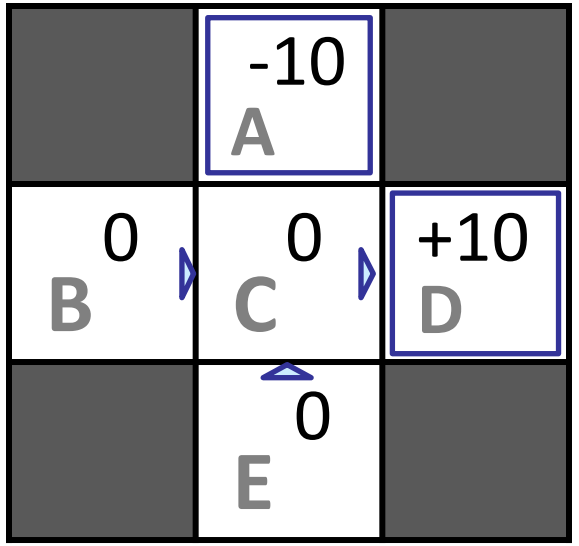
What is value of state C via Direct Evaluation?

Assume: $\gamma = 1$

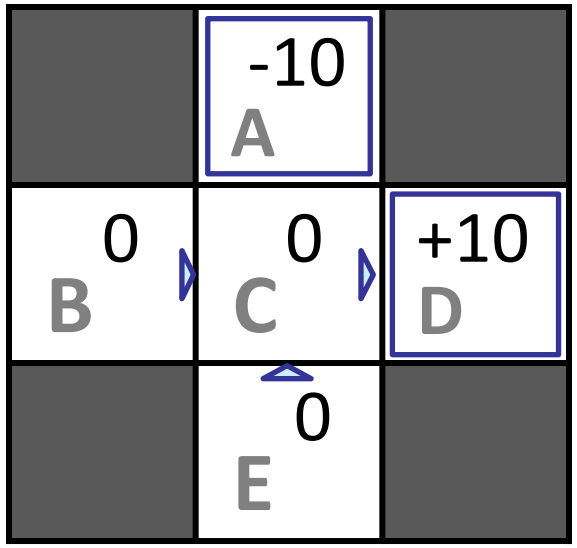
Problems with Direct Evaluation

- What's good about direct evaluation?
  - It's easy to understand
  - It doesn't require any knowledge of T, R
  - It eventually computes the correct average values, using just sample transitions
- What bad about it?
  - It wastes information about state connections
  - Need to have all episodes ahead of time (cannot "stream" in transitions)

Output Values



Problems with Direct Evaluation



Observed (s, a, s', R) Transitions:

Episode 1

E (home),	study,	C (know material),	0
C (know material),	go to exam,	D (pass exam),	0
D (pass exam),	exit,	x,	+10

Episode 2

B (library),	study,	C (know material),	0
C (know material),	go to exam,	A (miss bus & fail exam),	0
A (fail exam),	exit,	x,	-10

Is studying in the library a bad idea?

Direct Evaluation

- Goal: Compute values for each state under π

- Idea: Average together observed sample values
  - Act according to π
  - Every time you visit a state, write down what the sum of discounted rewards turned out to be:

$$sample_i(s) = \sum_t \gamma^t R^t$$

  - Average those samples:

$$V(s) \approx \frac{1}{N} \sum_i sample_i(s)$$

- This is called direct evaluation

Exponential Moving Average

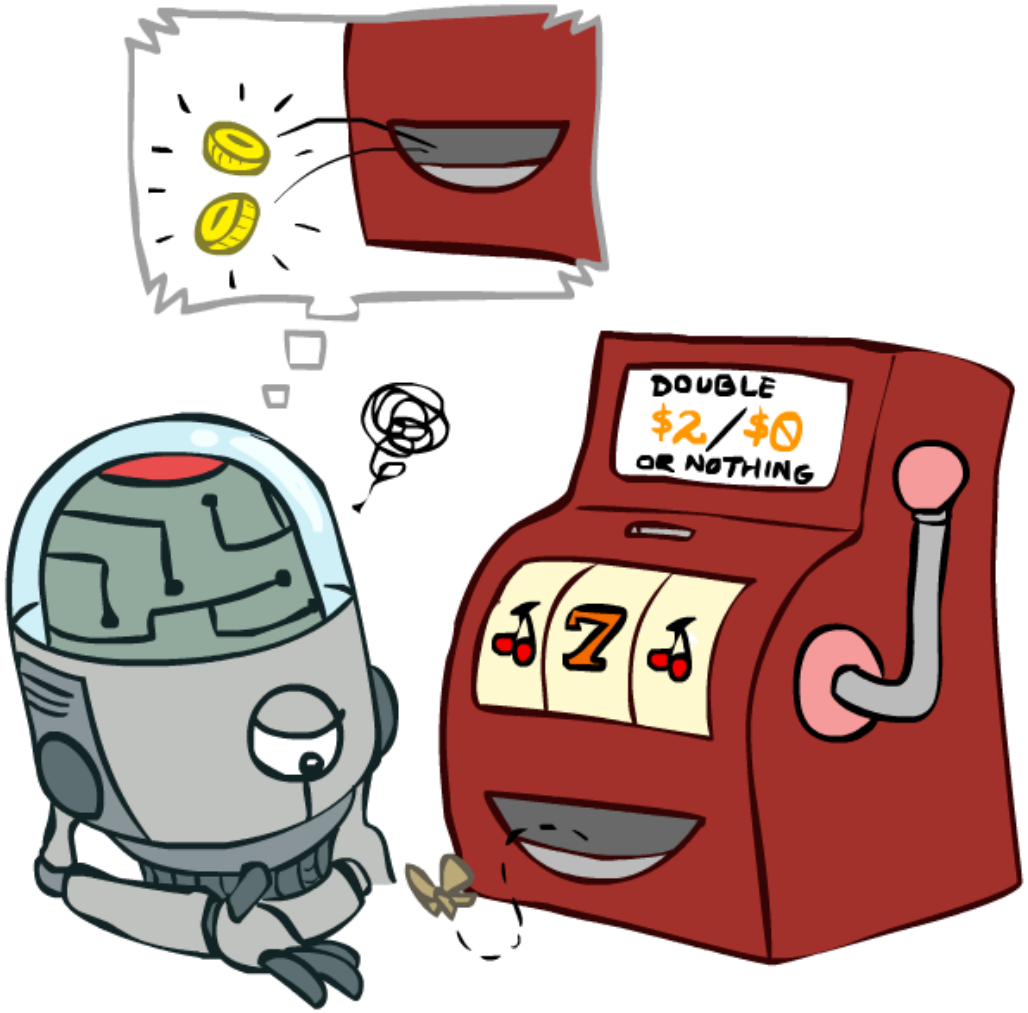
- Traditional Average: $AVG(x) = \frac{1}{N}\sum_{n} x_n$
  - Need to have all N samples at once (cannot "stream" in samples)

- Exponential moving average
  - The running interpolation update: $\bar{x}_n = (1-\alpha)\cdot\bar{x}_{n-1} + \alpha\cdot x_n$
  - Makes recent samples more important:

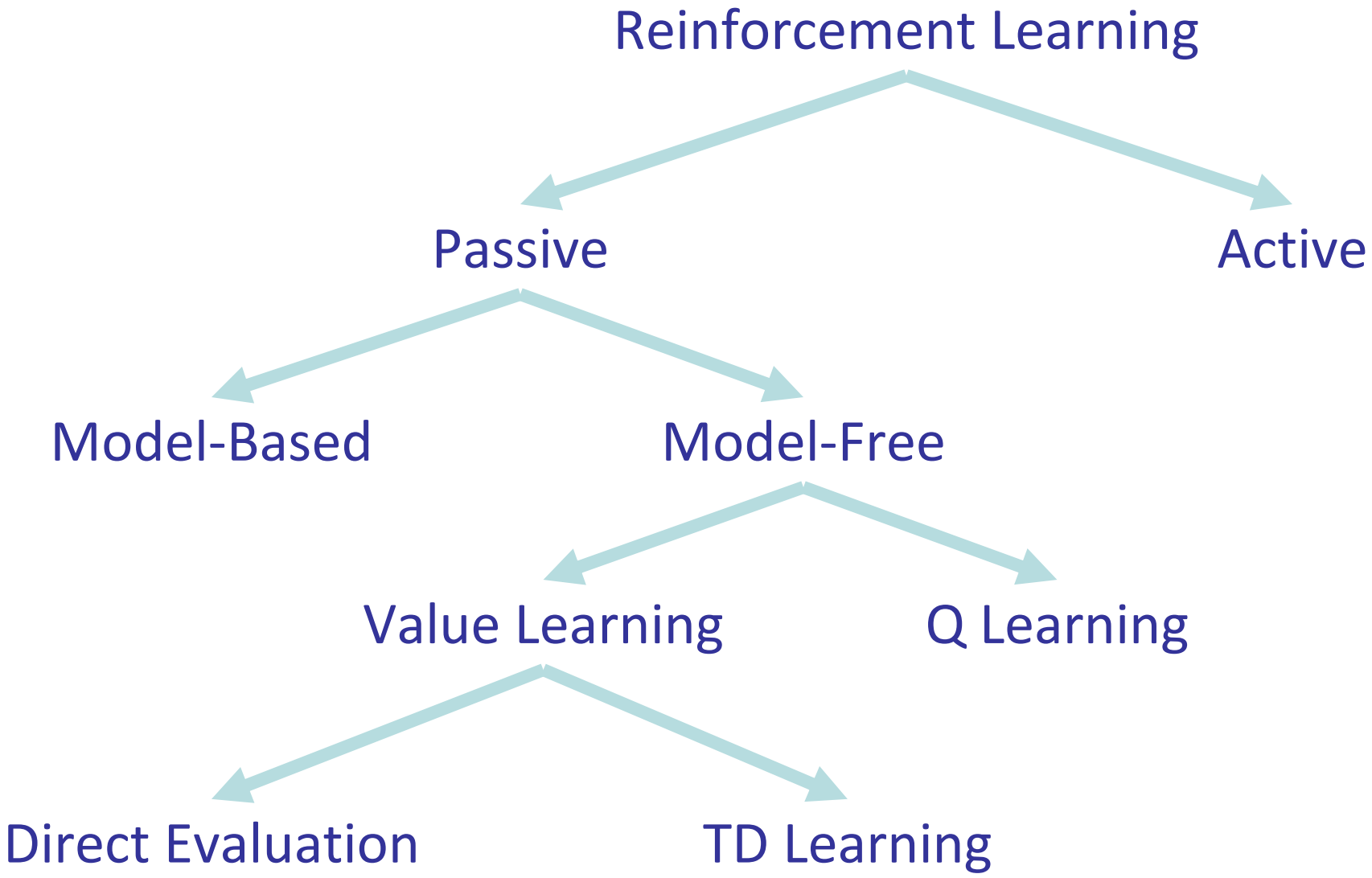
$$\bar{x}_n = \frac{x_n + (1-\alpha)\cdot x_{n-1} + (1-\alpha)^2\cdot x_{n-2} + \dots}{1 + (1-\alpha) + (1-\alpha)^2 + \dots}$$

  - Forgets about the past samples (how quickly depends on α)
- Decreasing learning rate (alpha) can give converging averages

Temporal Difference Learning

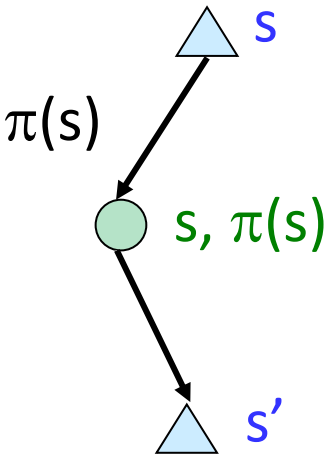


Reinforcement Learning Taxonomy



Temporal Difference Learning

- Big idea: learn from every experience!
  - Update V(s) each time we experience a transition (s, a, s', r)
  - Likely outcomes s' will contribute updates more often



- Temporal difference learning of values
  - Policy still fixed, still doing evaluation!
  - Move values toward value of whatever successor occurs: running average

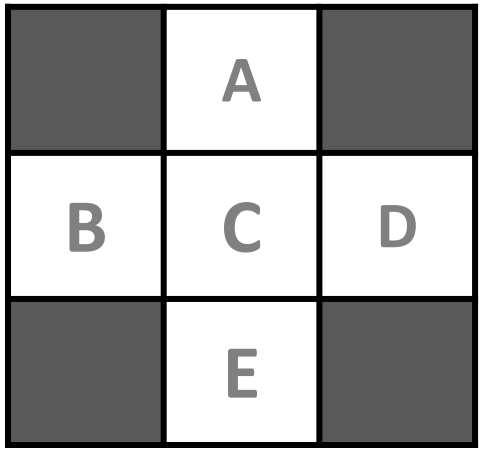
Sample of V(s): $sample = R(s, \pi(s), s') + \gamma V^{\pi}(s')$

Update to V(s): $V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + (\alpha)sample$

Same update: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$

Example: Temporal Difference Learning

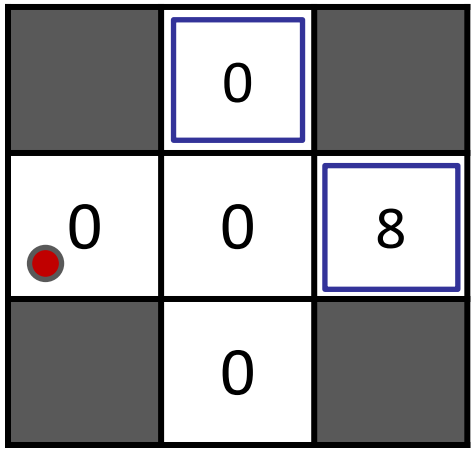
States



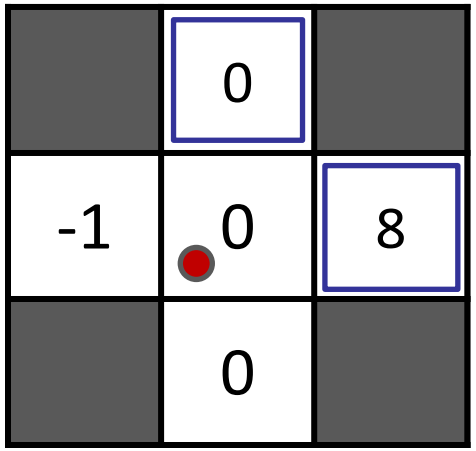
Assume: $\gamma = 1$, $\alpha = 1/2$

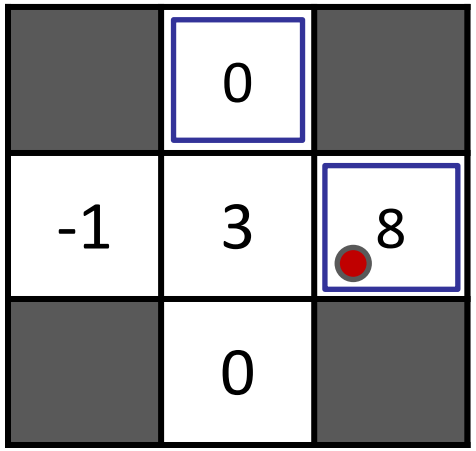
Observed Transitions

B, east, C, -2



C, east, D, -2





$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

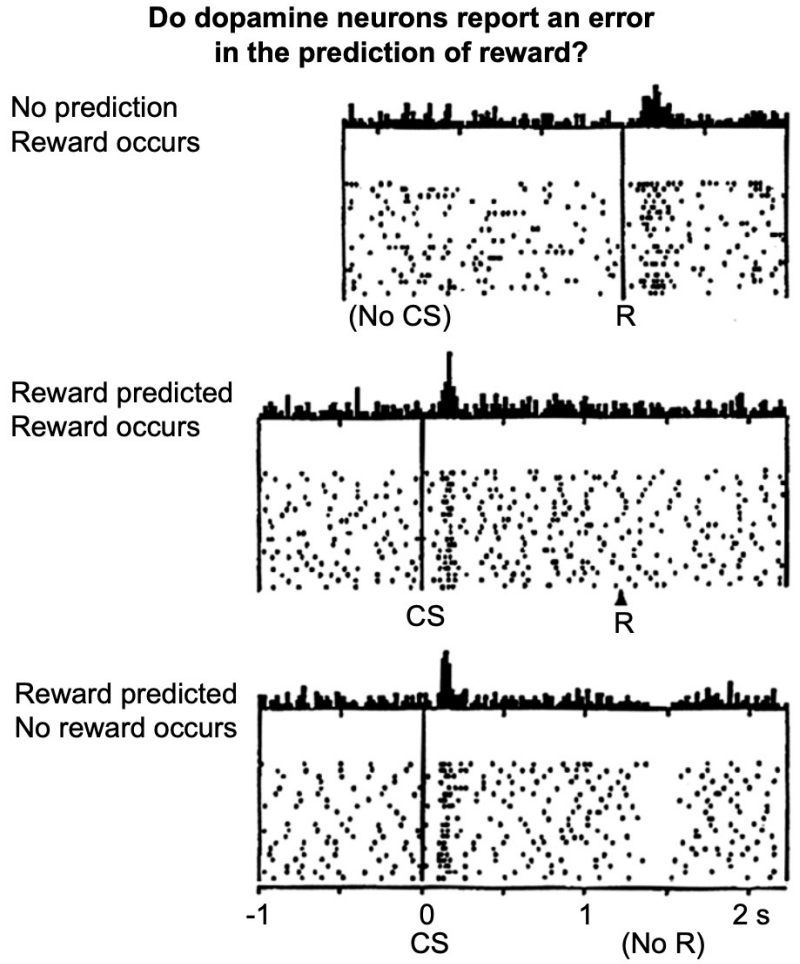
TD Learning in the Brain

- Neurons transmit Dopamine to encode reward or value prediction error

$$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$$

- Example of Neuroscience & RL informing each other

- For more examples, see *[AI and Neuroscience: A virtuous circle]*
  - https://www.deepmind.com/blog/ai-and-neuroscience-a-virtuous-circle



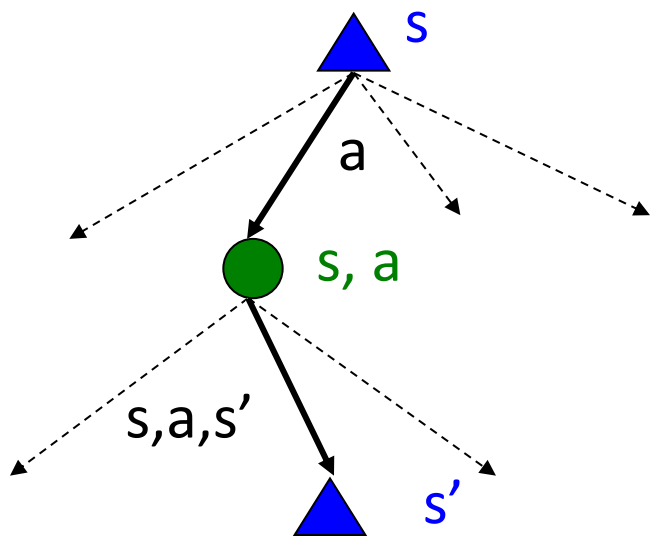
[A Neural Substrate of Prediction and Reward. Schultz, Dayan, Montague. 1997]

Problems with TD Value Learning

- TD value leaning is a model-free way to do **policy evaluation**, mimicking Bellman updates with running sample averages
- However, if we want to turn values into a (new) policy, we're stuck:

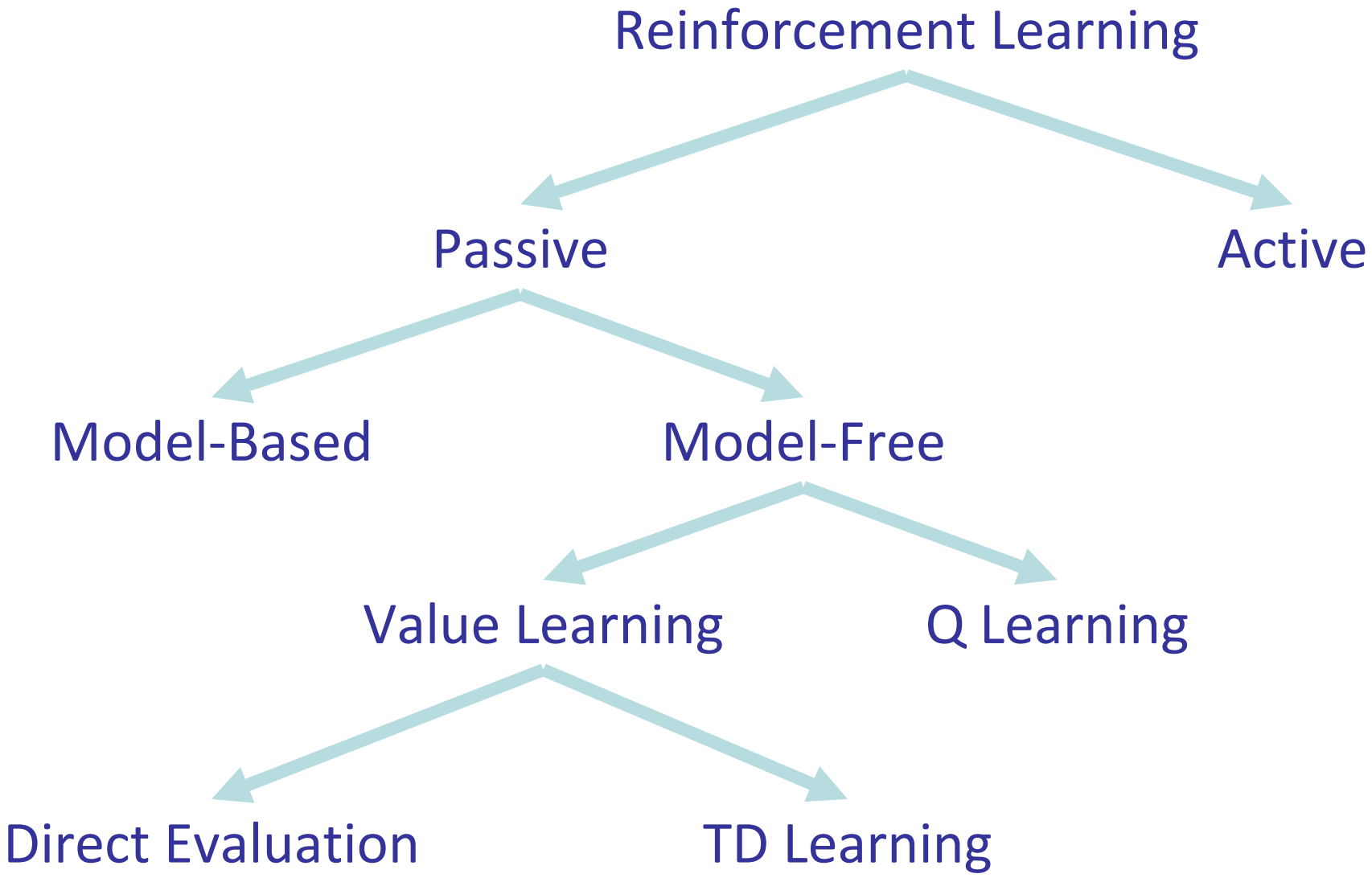
$$\pi(s) = \arg\max_a Q(s,a)$$

$$Q(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s') \right]$$



- What can we do?
  - Learn Q-values, not values
  - Makes action selection model-free too!

Reinforcement Learning Taxonomy



Q-Value Iteration

- Value iteration: find successive (depth-limited) values
  - Start with $V_0(s) = 0$, which we know is right
  - Given V_k, calculate the depth k+1 values for all states:

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

- But Q-values are more useful, so compute them instead
  - Start with $Q_0(s,a) = 0$, which we know is right
  - Given Q_k, calculate the depth k+1 q-values for all q-states:

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

Q-Learning

- Q-Learning: sample-based Q-value iteration

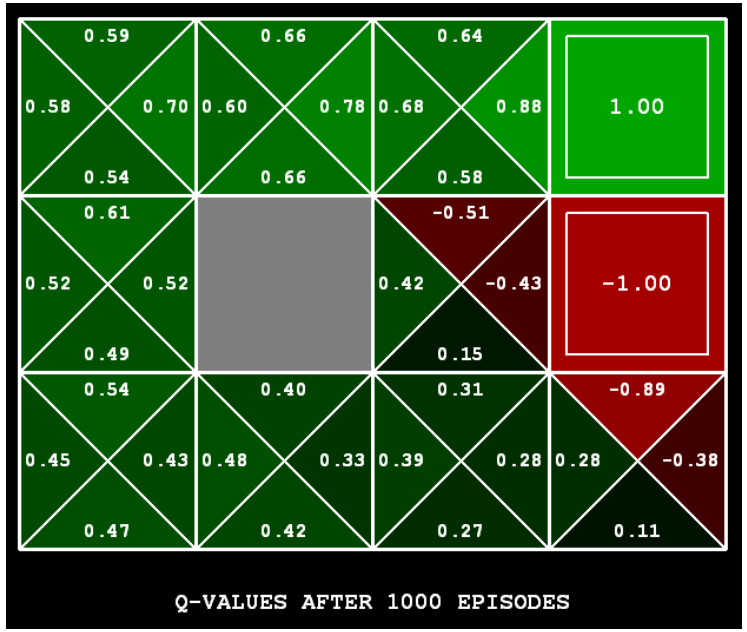
$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

- Learn Q(s,a) values as you go
  - Receive a sample (s,a,s',r)
  - Consider your old estimate: $Q(s,a)$
  - Consider your new sample estimate:

$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

  - Incorporate the new estimate into a running average:

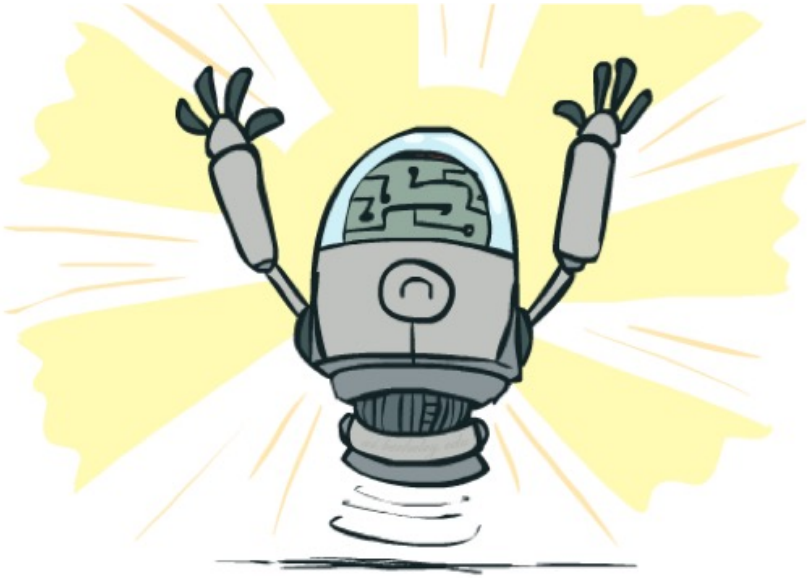
$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\,[sample]$$



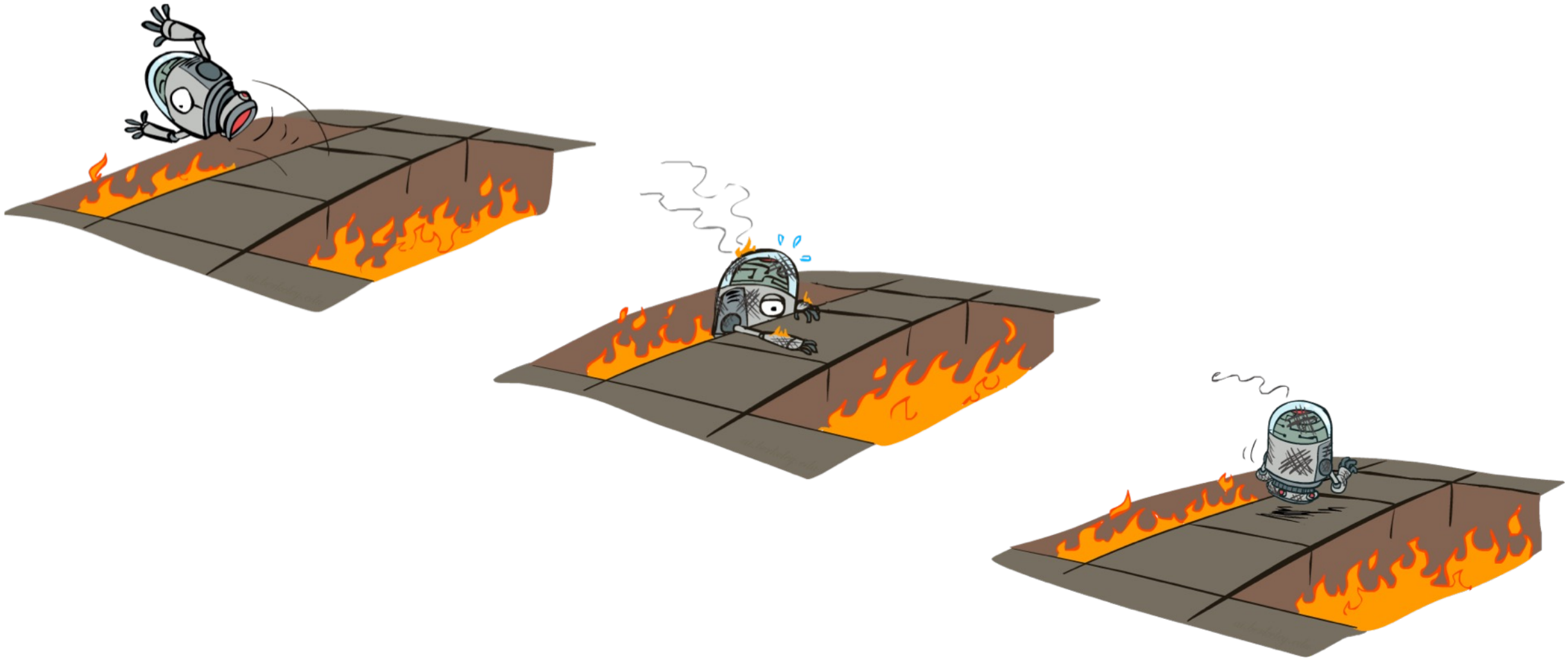
[Demo: Q-learning – gridworld (L10D2)]
[Demo: Q-learning – crawler (L10D3)]

Q-Learning Properties

- Amazing result: Q-learning converges to optimal policy -- even if you're acting suboptimally!
- This is called off-policy learning
- Caveats:
  - You have to explore enough
  - You have to eventually make the learning rate small enough
  - … but not decrease it too quickly
  - Basically, in the limit, it doesn't matter how you select actions (!)



Active Reinforcement Learning

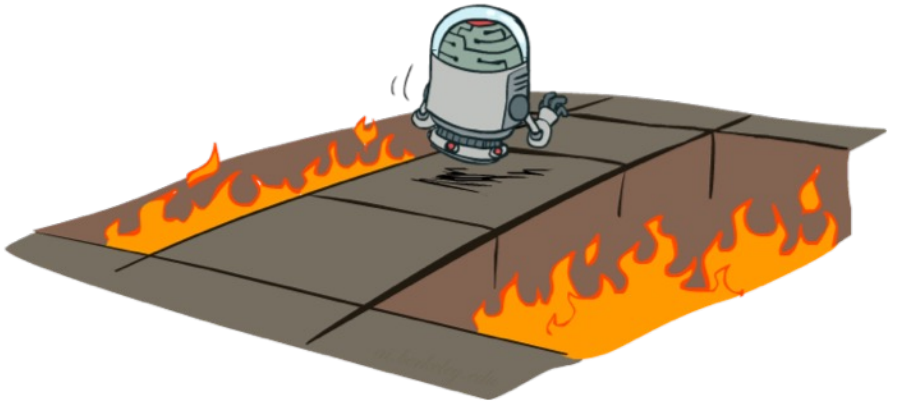


Video of Demo Q-Learning -- Gridworld

Video of Demo Q-Learning -- Crawler

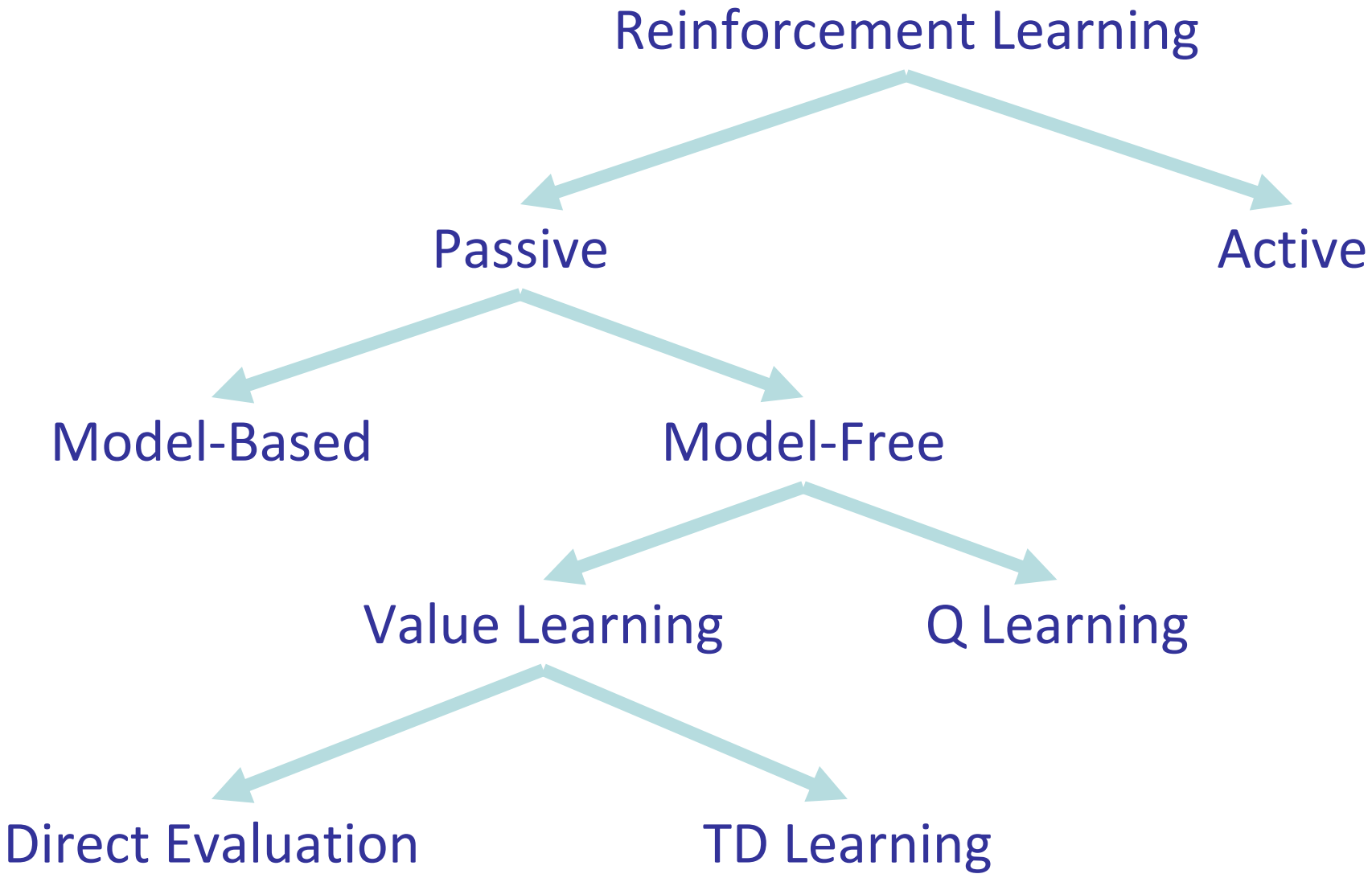
Active Reinforcement Learning

- Full reinforcement learning: optimal policies (like value iteration)
  - You don't know the transitions T(s,a,s')
  - You don't know the rewards R(s,a,s')
  - You choose the actions now
  - Goal: learn the optimal policy / values



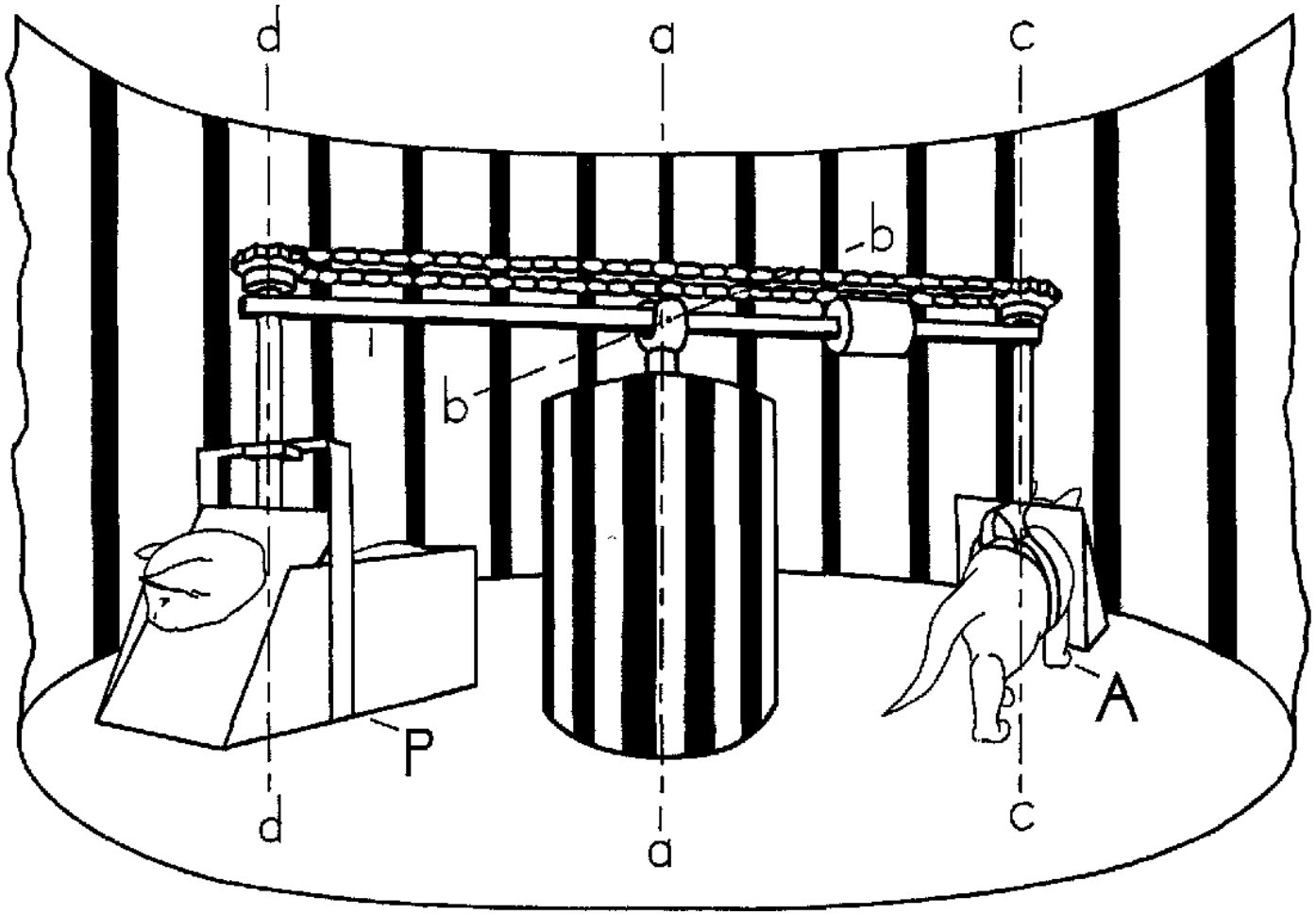
- In this case:
  - Learner makes choices!
  - Fundamental tradeoff: exploration vs. exploitation
  - This is NOT offline planning! You actually take actions in the world and find out what happens…

Reinforcement Learning Taxonomy



Next Time: More RL Ideas and Part 1 Conclusion!

Passive vs Active Reinforcement Learning



[Movement-produced stimulation in the development of visually guided behavior. Held and Hein. 1963]