REINFORCEMENT LEARNING

DR. NATALY KURITZ

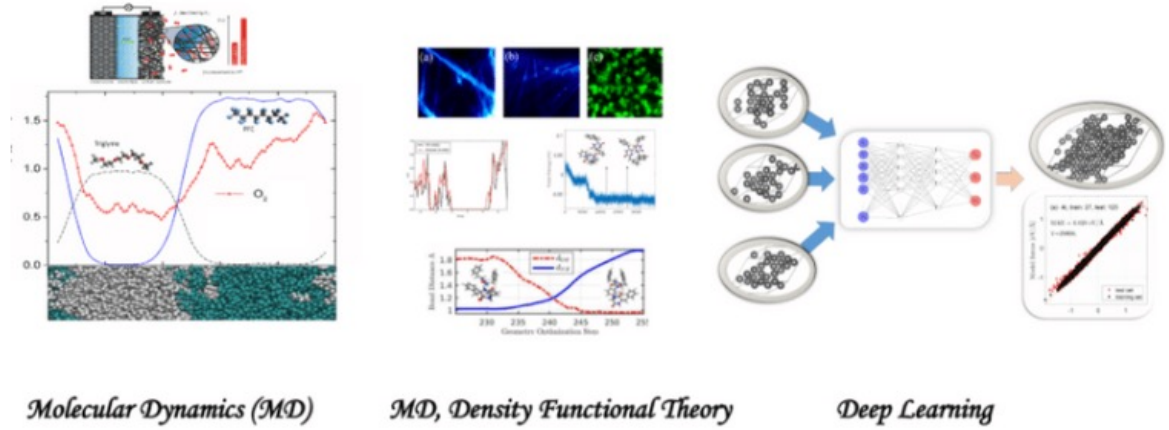
AI Decision Scientist

Anaplan

nataly17k@gmail.com

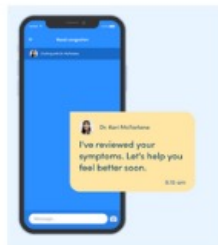
linkedin.com/in/nataly-kuritz/

DATA NIGHTS



Molecular Dynamics (MD) *MD, Density Functional Theory* *Deep Learning*

PHD ELECTRICAL ENG. FROM TAU



DATA SCIENTIST @ KHEALTH

EX-CEO&CO-FOUNDER OF NANOFLUTE

OUTLINE

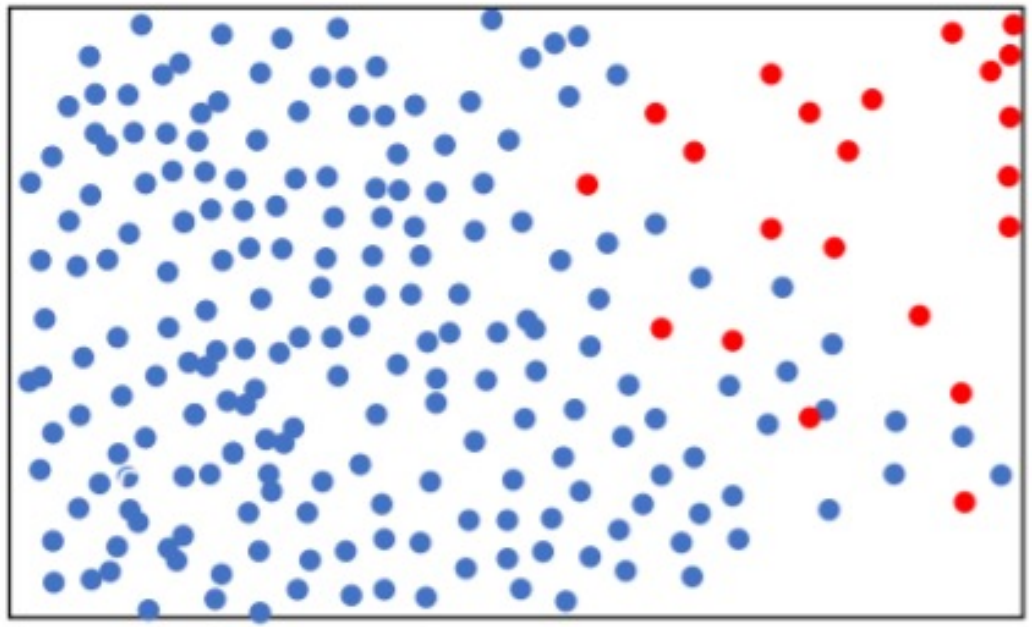
Background	Problem statement	Multi-armed-bandit	Real-world challenges	Summary
• Exploration need • Active learning • decision making • Use-cases and Landscape	• The Problem • terminology and definitions	• General Algo • Main sampling strategies	• Q-learner • Explainability	

WHAT'S IN COMMON?

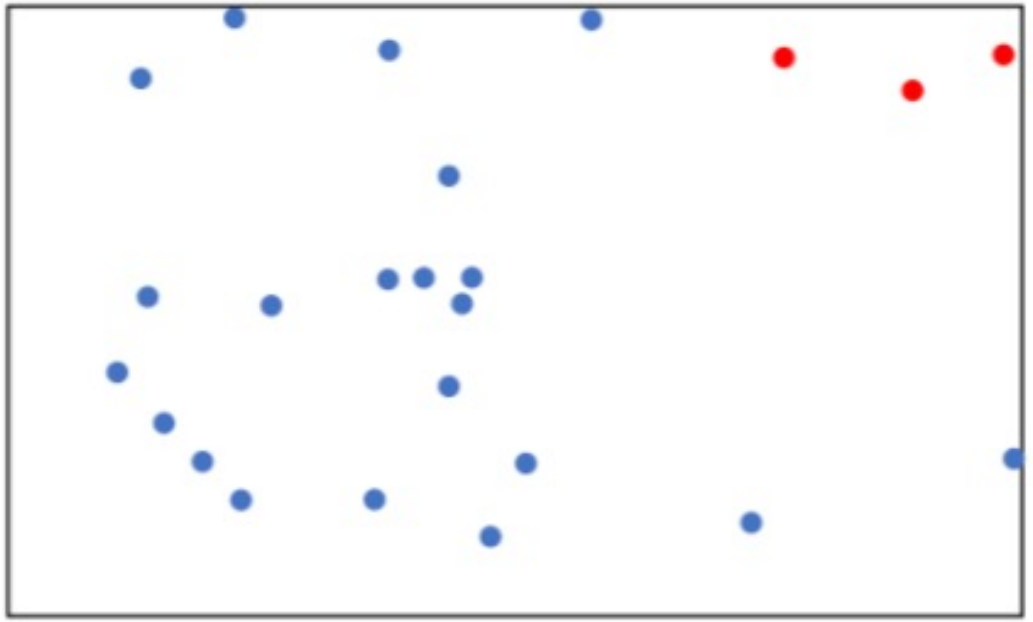
SPACE EXPLORATION

SUPERVISED, ACTIVE LERNING

EXPLORATION – EXPLOITATION BALANCE – FIRST MEETING



(a) Entire Data Distribution



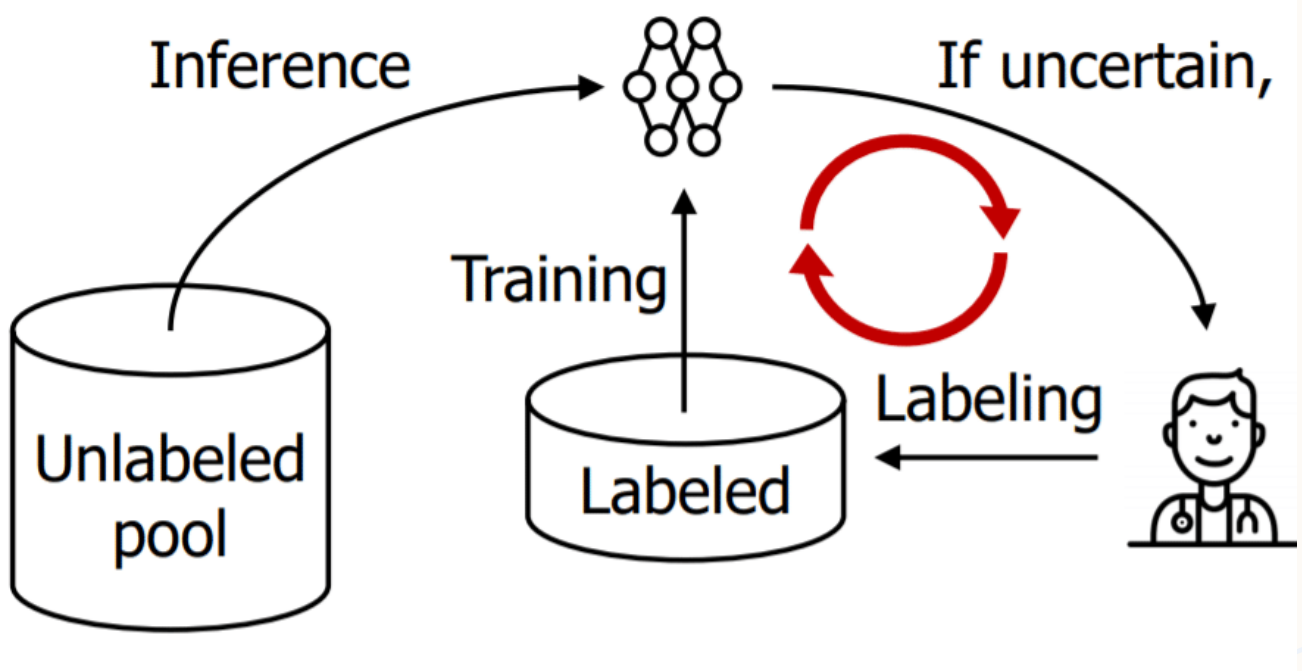
(b) Random Sampling

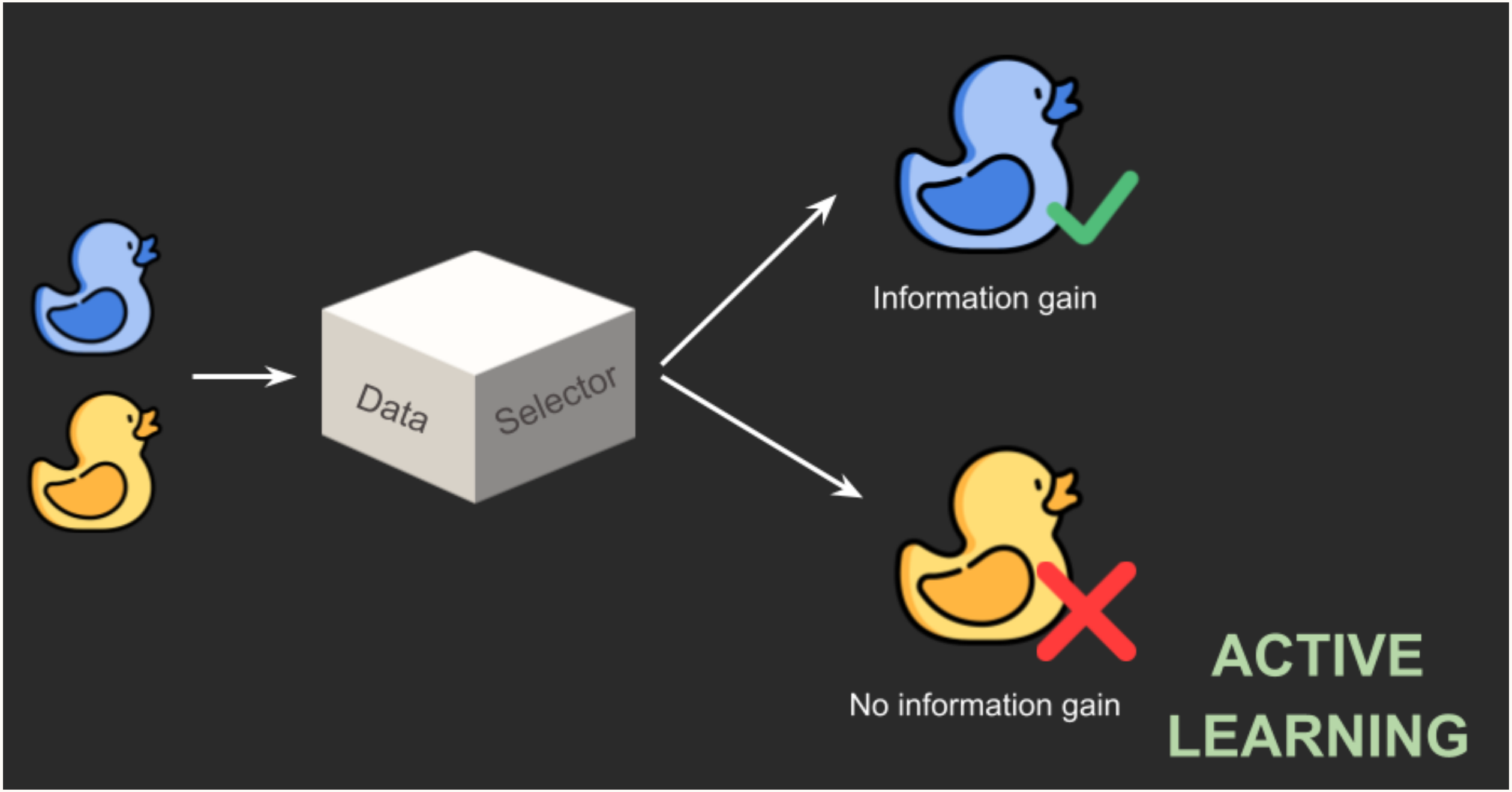
FEW WORDS ABOUT ACTIVE LEARNING

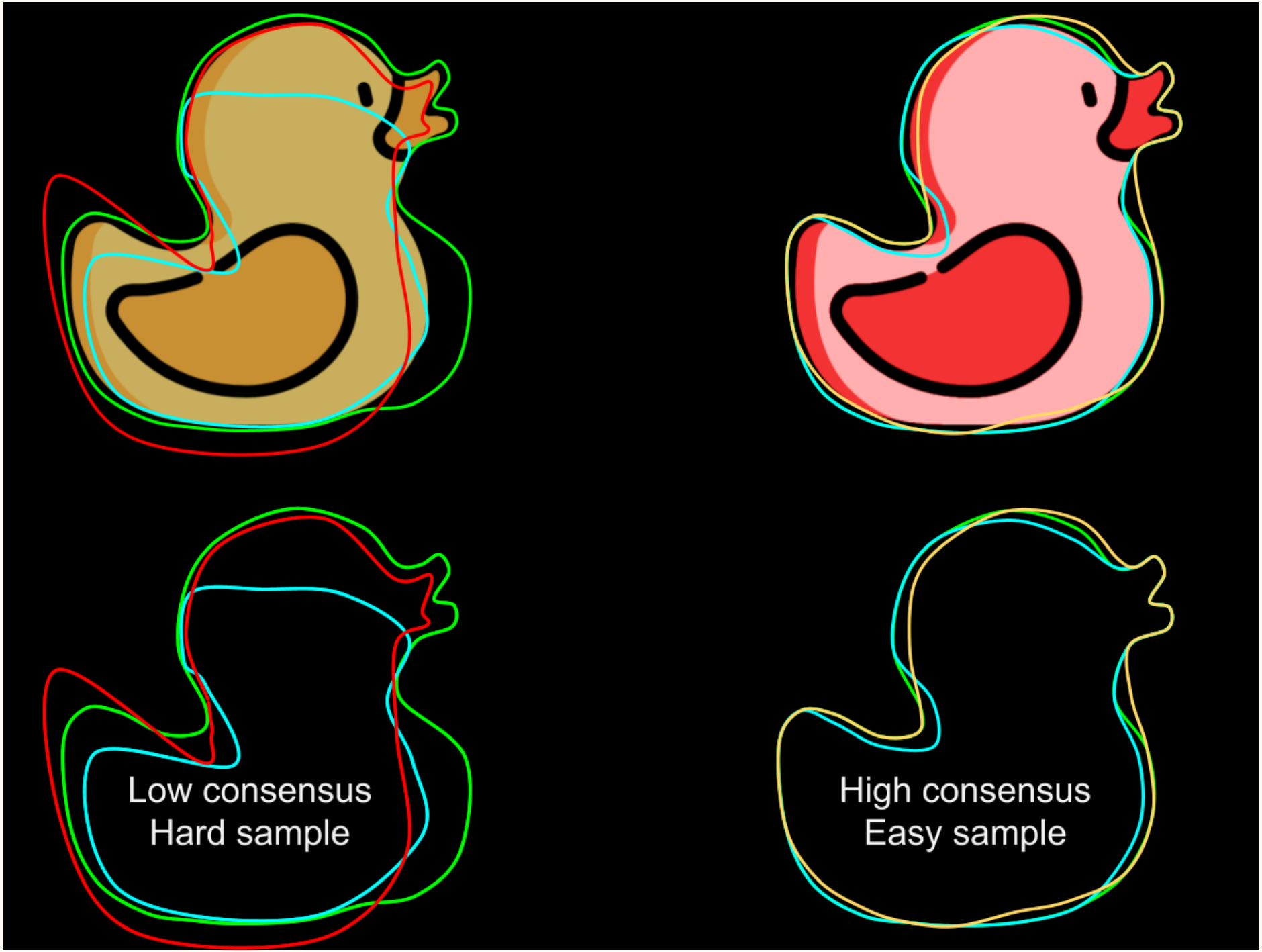
How do you choose your next sample to train

Effective learning

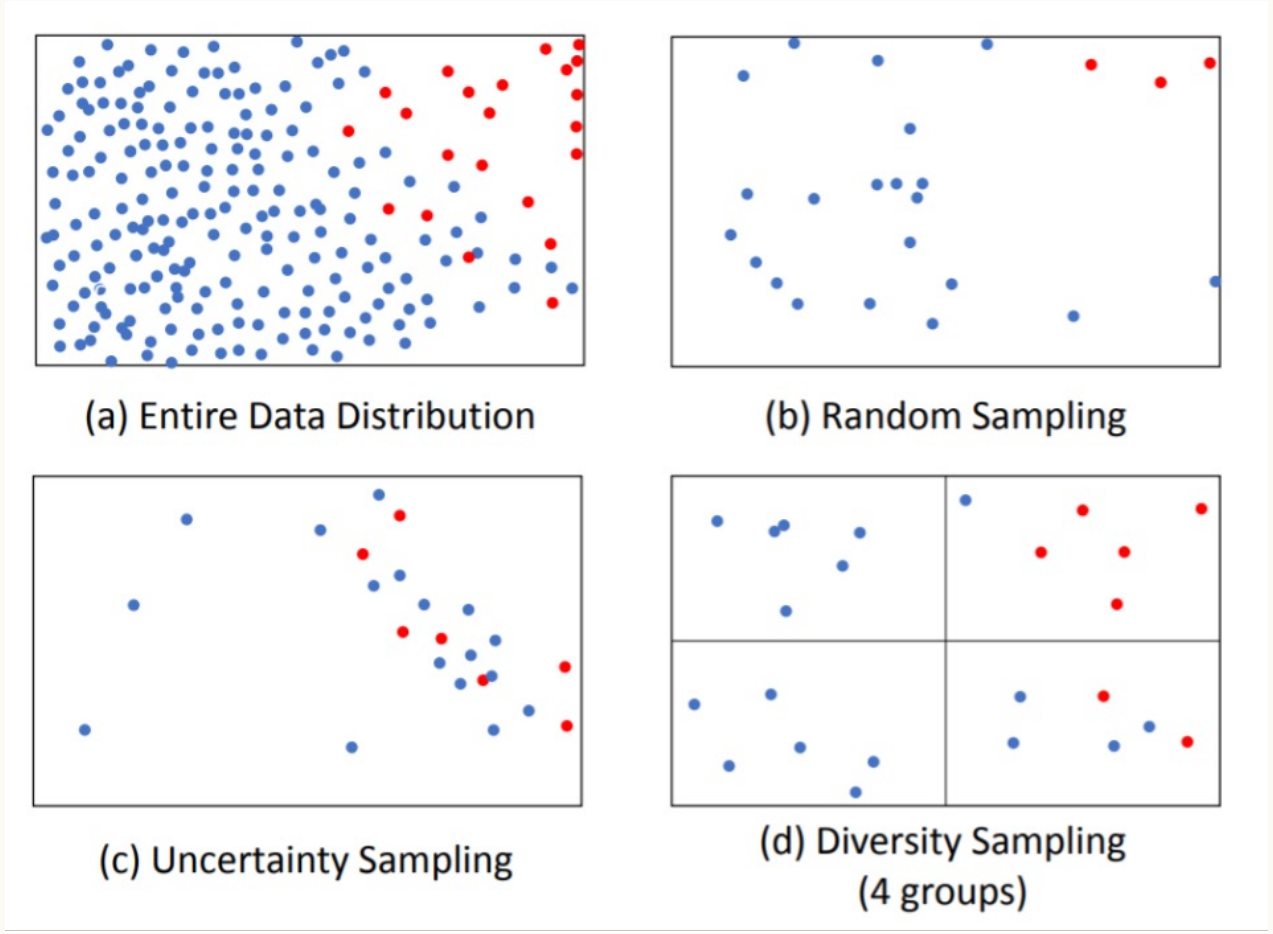
On-the-fly learning




Data
Selector
Information gain
No information gain
ACTIVE
LEARNING


Low consensus
Hard sample
High consensus
Easy sample

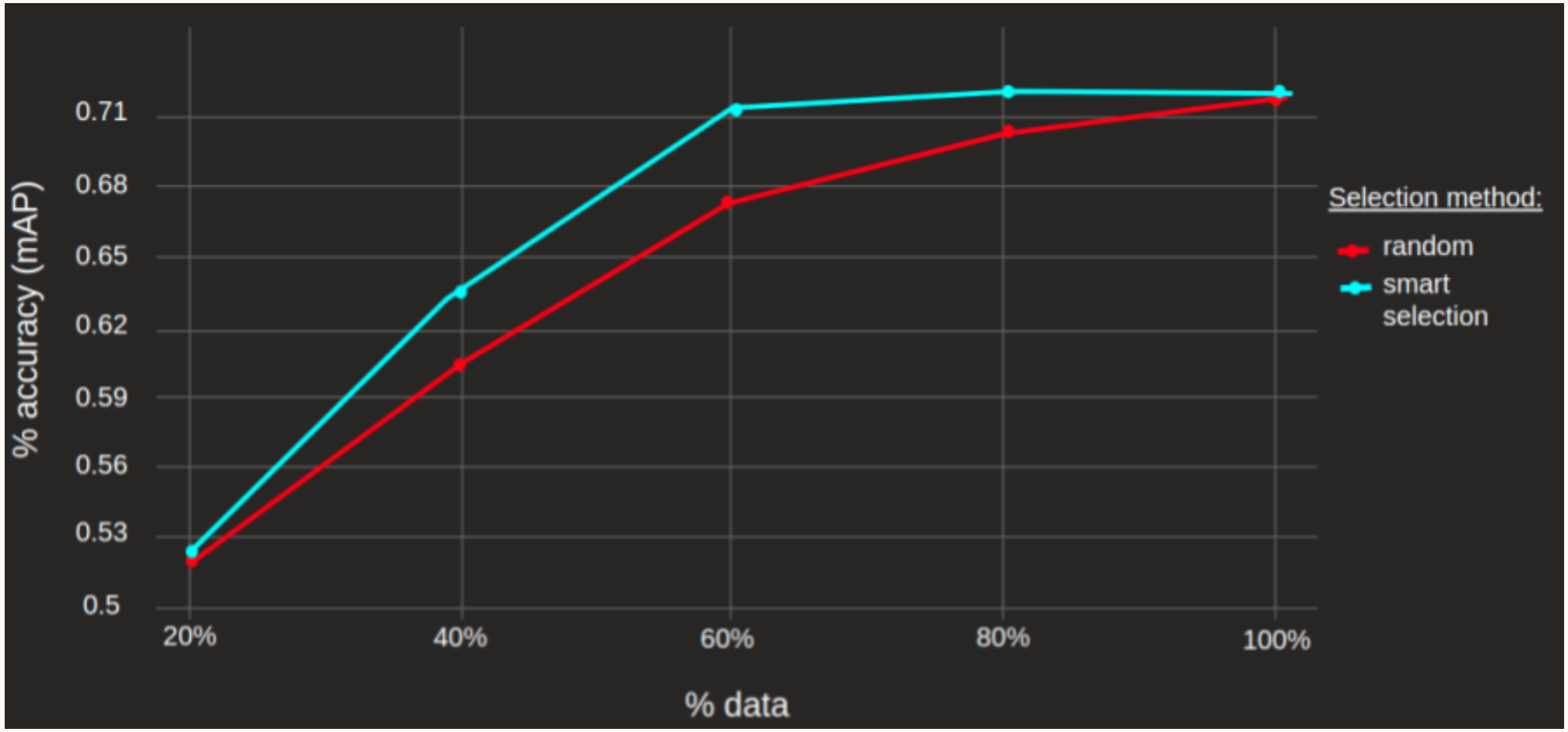
EXPLORATION – EXPLOITATION METHODS



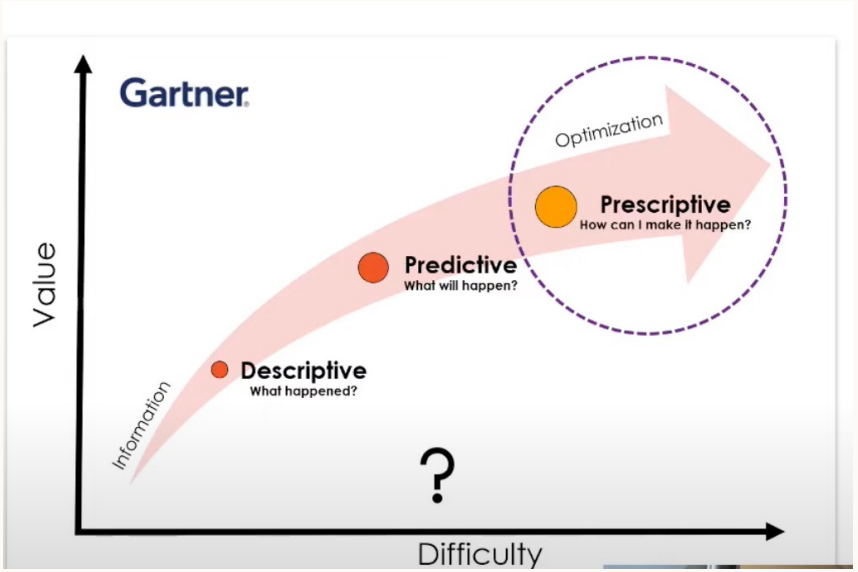
WHAT'S YOUR FAVORITE?

	Uncertainty approach	Diversity approach
Task agnostic/specific	Mostly task specific	Task agnostic
Output dataset	Redundant	Diverse
Model dependent	Mostly can be based on existing model	Need to implement new separate model

SMART SAMPLING



DEISION MAKING IS HARD


Gartner
Optimization
Prescriptive
How can I make it happen?
Predictive
What will happen?
Descriptive
What happened?
Value
Information
?
Difficulty

OUTPUT VS OUTCOME – BEYOND SUPERVISED

- **Structurization** of the data and essential insights/information
- Estimate the **uncertainty** quantitatively
- Describe/Analise —tell a **story** over facts
- get **insights** about inner behavior

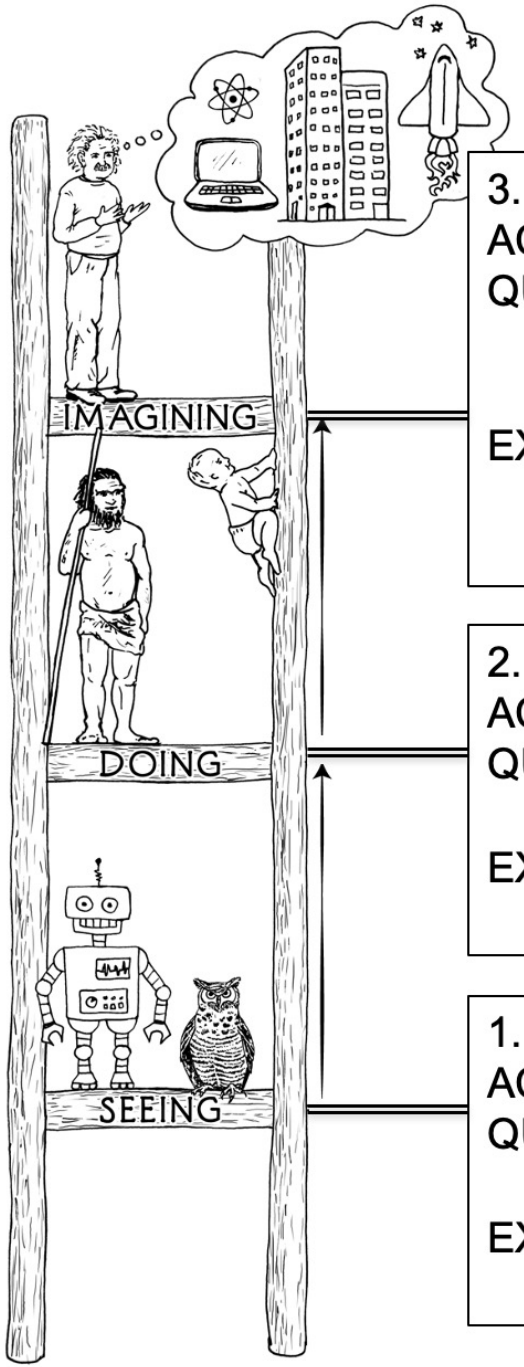
OUTPUT VS OUTCOME – BEYOND SUPERVISED

- Predict measurable
- Suggest anomaly
- Improve user experience
- Scale up

OUTPUT VS OUTCOME – BEYOND SUPERVISED

- Suggests **actions**
- Design **experiments**
- **Optimize** behavior
- **Decision** Making

3-LEVEL HIERARCHY



3. COUNTERFACTUALS

ACTIVITY: Imagining, Retrospection, Understanding

QUESTIONS: *What if I had done . . . ? Why?* (Was it X that caused Y? What if X had not occurred? What if I had acted differently?)

EXAMPLES: Was it the aspirin that stopped my headache? Would Kennedy be alive if Oswald had not killed him? What if I had not smoked the last 2 years?

2. INTERVENTION

ACTIVITY: Doing, Intervening

QUESTIONS: *What if I do . . . ? How?* (What would Y be if I do X?)

EXAMPLES: If I take aspirin, will my headache be cured? What if we ban cigarettes?

1. ASSOCIATION

ACTIVITY: Seeing, Observing

QUESTIONS: *What if I see . . . ?* (How would seeing X change my belief in Y?)

EXAMPLES: What does a symptom tell me about a disease? What does a survey tell us about the election results?

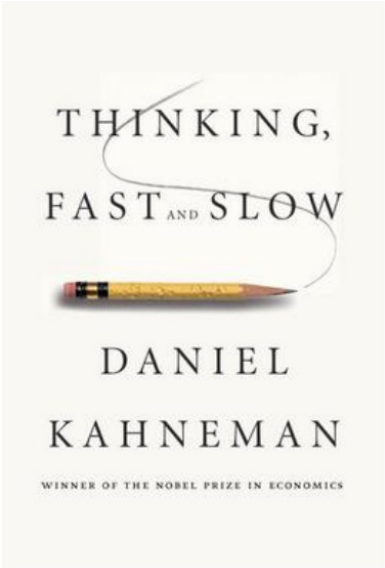
SYSTEM 1 VS. SYSTEM 2 COGNITION

2 systems (and categories of cognitive tasks):

Manipulates high-level / semantic concepts, which can be recombined combinatorially

System 1

- Intuitive, fast, **UNCONSCIOUS**, non-linguistic, habitual
- Current DL



System 2

- Slow, logical, sequential, **CONSCIOUS**, linguistic, algorithmic, planning, reasoning
- Future DL

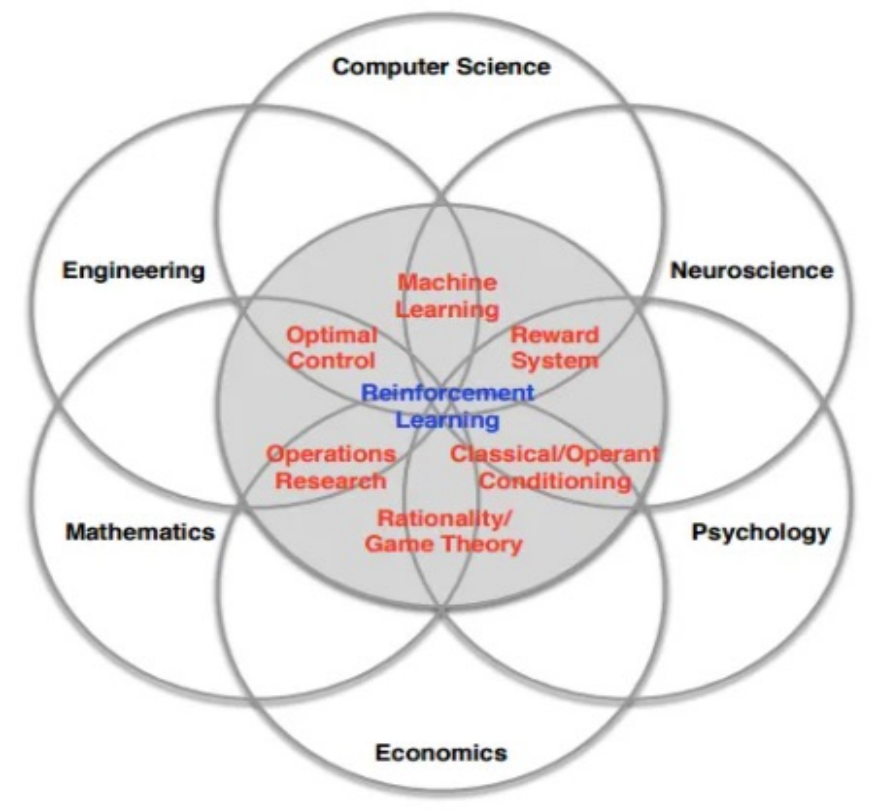
MISSING TO EXTEND DEEP LEARNING TO REACH **HUMAN-LEVEL AI**

- **Out-of-distribution generalization & transfer**
- **Higher-level cognition: system 1 → system 2**
  - *High-level semantic representations*
  - *Compositionality*
  - *Causality*
- **Agent perspective:**
  - *Better world models*
  - ***Causality***
  - *Knowledge-seeking*
- **Connections between all 3 above!**

WHAT WOULD WE GAIN

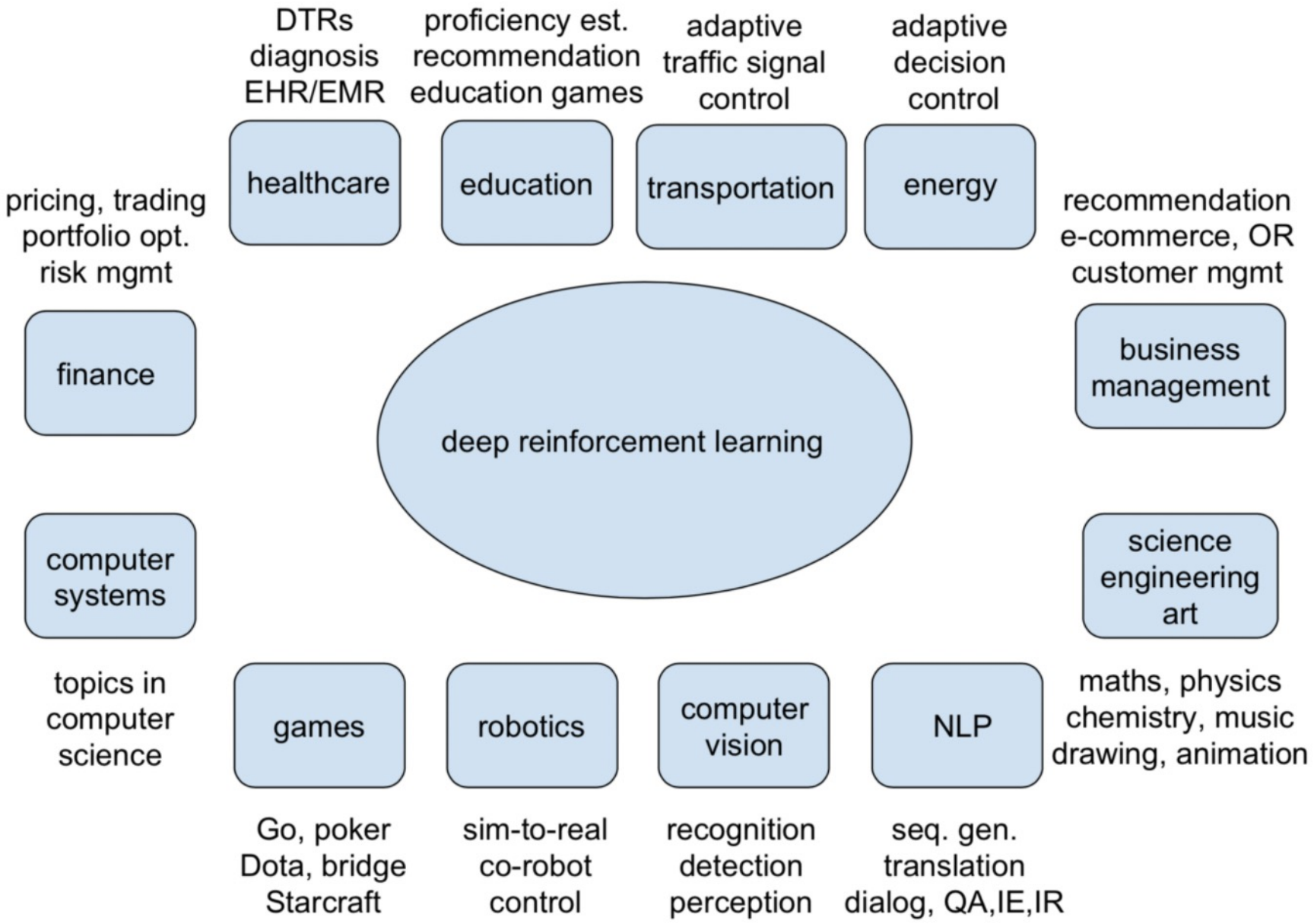
Mechanism	Disclose the mechanism
Reasoning	Reasoning
Optimization	Find best action
Decisions	Make decisions
React	React to dynamic environment (on-the-fly learning)
Auto-Pilot	Build an agent that can be better than you in taking hard decisions (beyond robust/scalable)

THE LANDSCAPE



TYPICAL USE CASES

Clinical trials	Smart robotics	Planning	Autonomous Driving
Ads placement on web pages	Recommendations	Traffic or communication networks	Game-playing



Yuxi Li, Deep Reinforcement Learning, arXiv, 2018

Q/A

Exploration need
Active Learning
Decision making
Could you think about other use cases?

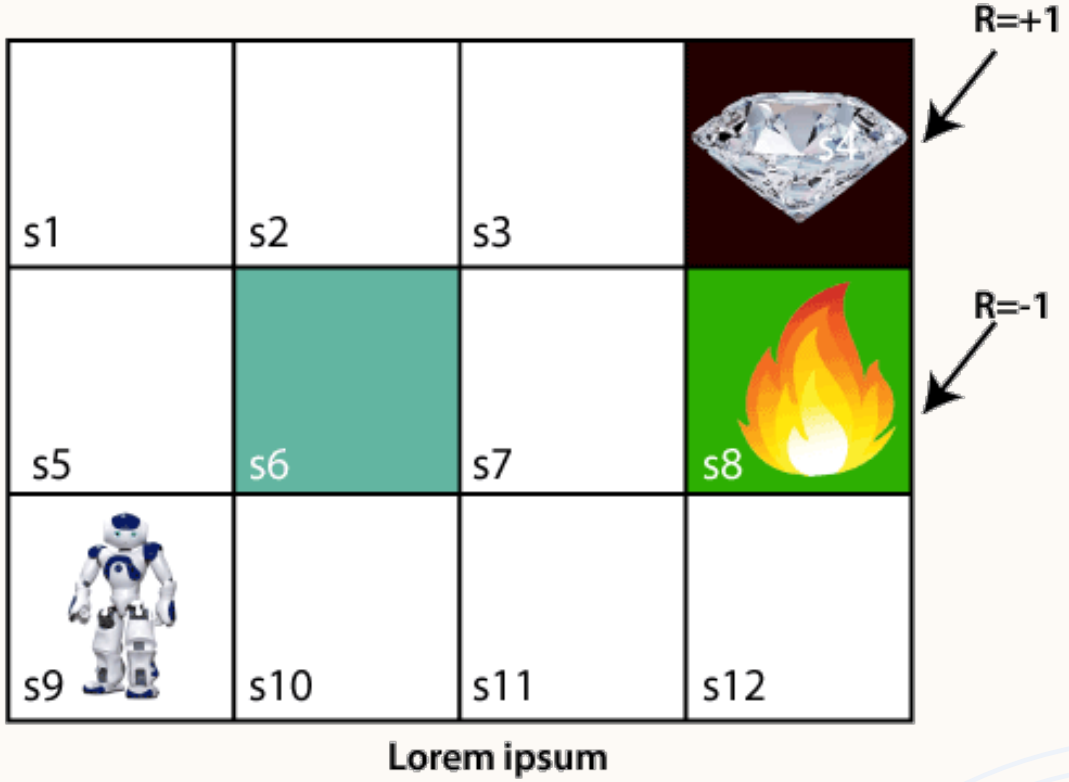
OUTLINE

Background	**Problem statement**	Multi-armed-bandit	Real-world challenges	Summary
• Exploration need • Active learning • decision making • Use-cases and Landscape	• **The Problem** • **Terminology and definitions**	• General Algo • Main sampling strategies	• Q-learner • Explainability	

GRID WORLD

Environment - The world

Agent - The brain

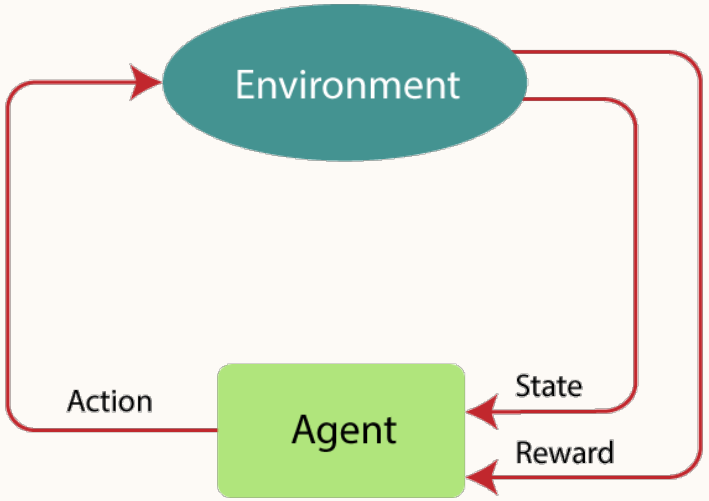


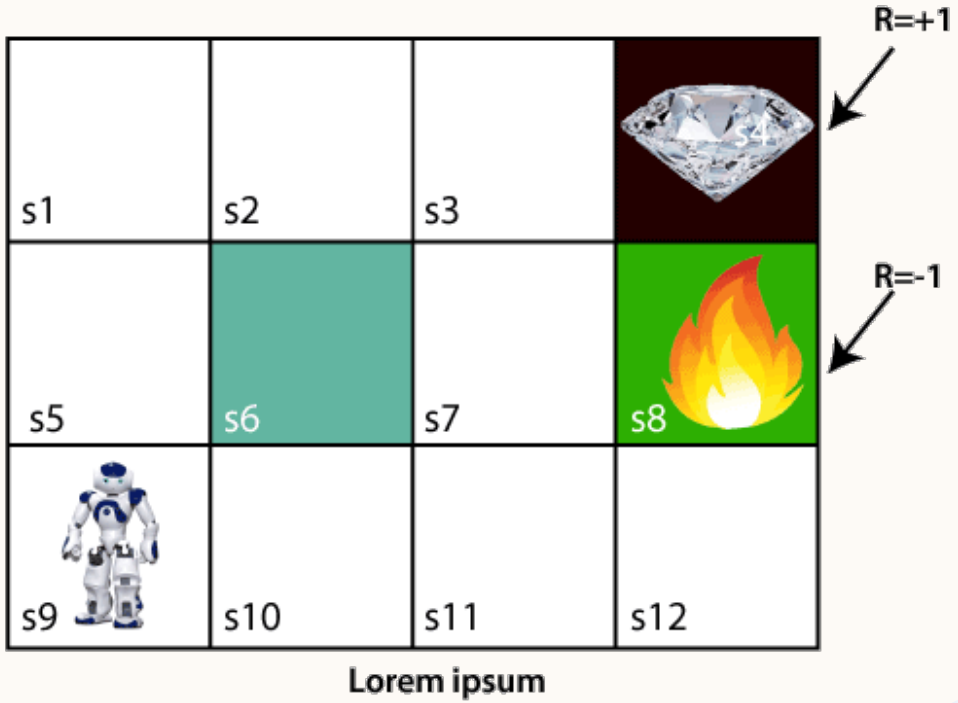
Lorem ipsum

GRID WORLD

Environment - The world

Agent - The brain





Lorem ipsum

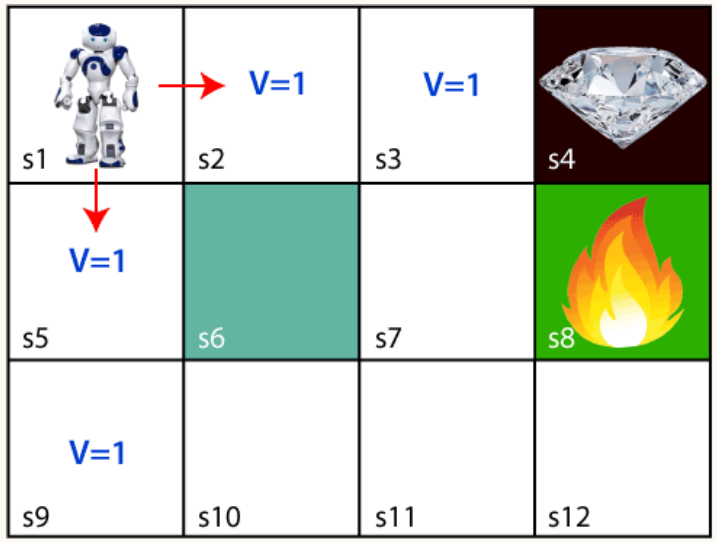
GRID WORLD

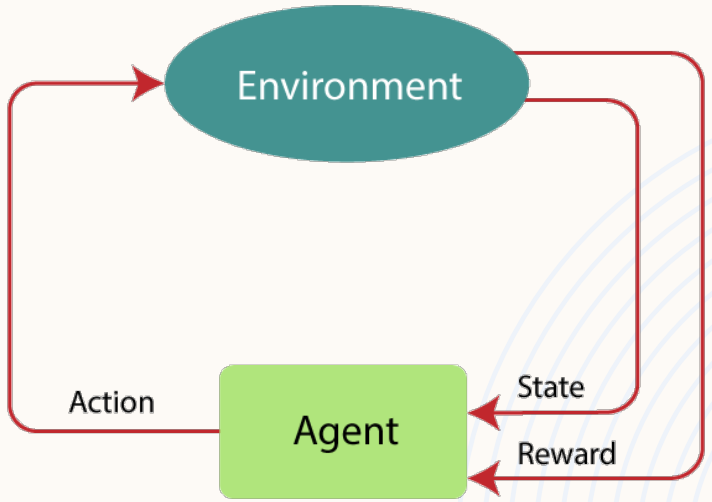
Environment - The world

Agent - The brain

Action – What Agent can do

State – any possible situation





GRID WORLD

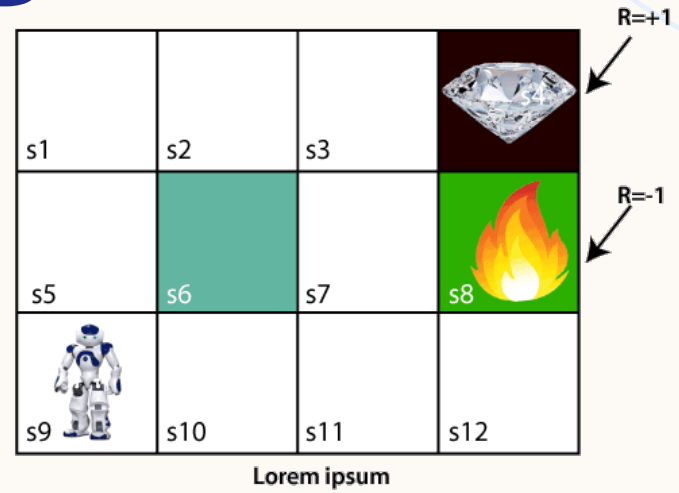
Environment - The world

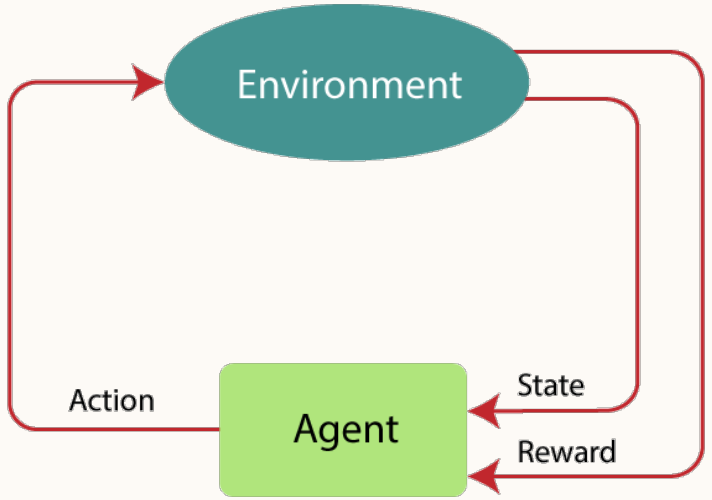
Agent - The brain

Action – What Agent can do

Reward – the goal

State – any possible situation/position





DISCOUNTED REWARDS

Rewards in the future are worth less than an immediate reward

- Because of uncertainty: who knows if/when you're going to get to that reward state

DISCOUNTED REWARDS

Discount factor $\gamma \leq 1$ (often $\gamma = 0.9$)

Assume reward n years in the future is only worth $(\gamma)^n$ of the value of immediate reward

- $(0.9\text{^}6) * 10{,}000 = 0.531 * 10{,}000 = 5310$

For each state, calculate a *utility* value equal to the *Sum of Future Discounted Rewards*

GRID WORLD

- **Environment** - The world
- **Agent** - The brain
- **Action** – What Agent can do
- **Reward** – the goal
- **Discount** - cost of an action/time
- **Value –** best state value
- **State** – any possible situation/position

$$V(s) = \max [R(s,a) + \gamma V(s`)]$$

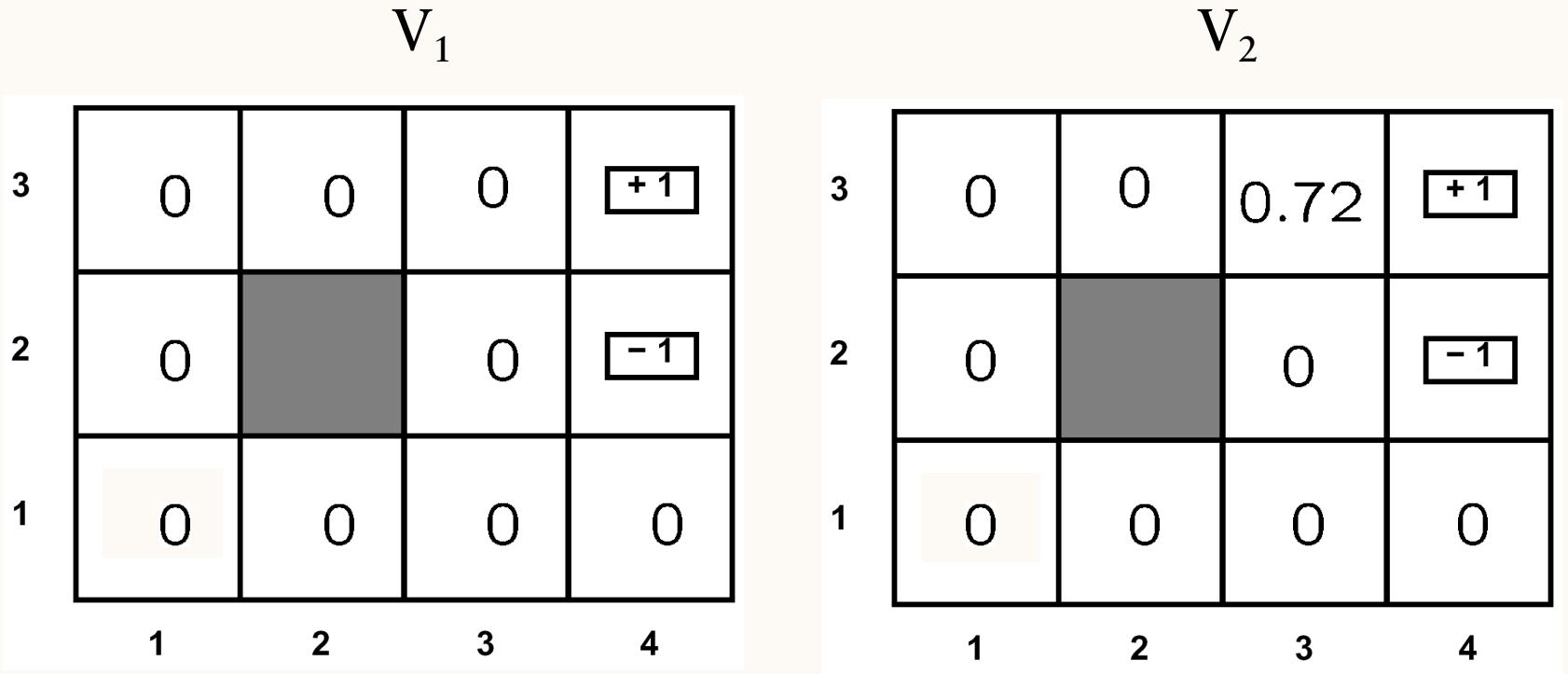
VALUE ITERATION

- Idea:
  - Start with $V_0^*(s) = 0$, which we know is right (why?)
  - Given V_i^*, calculate the values for all states for depth i+1:

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

  - This is called a value update or Bellman update
  - Repeat until convergence

EXAMPLE: VALUE ITERATION



- Information propagates outward from terminal states and eventually all states have correct value estimates

EXAMPLE: VALUE ITERATION

V_2

3	0	0	0.72	+1
2	0		0	−1
1	0	0	0	0
	1	2	3	4

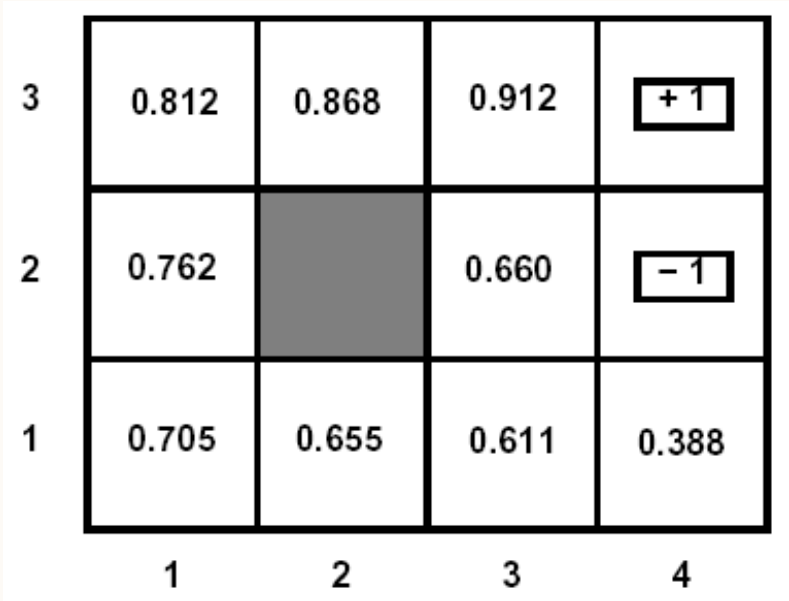
V_3

3	0	0.52	0.78	+1
2	0		0.43	−1
1	0	0	0	0
	1	2	3	4

- Information propagates outward from terminal states and eventually all states have correct value estimates

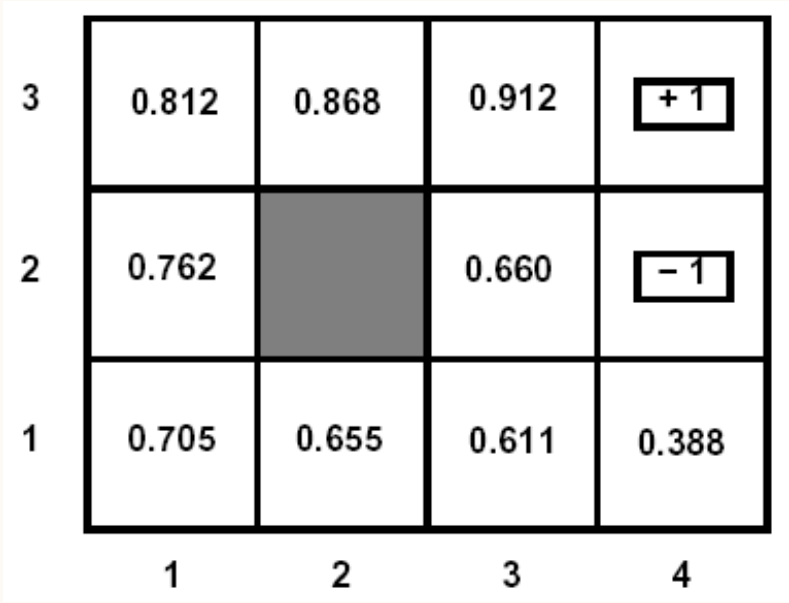
OPTIMAL UTILITIES

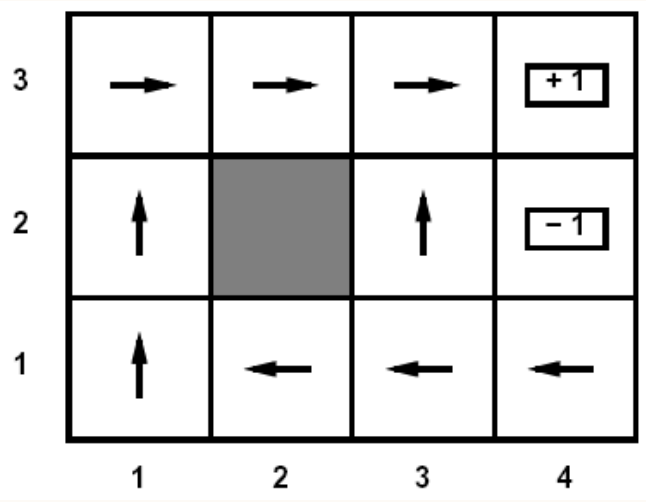
- Fundamental operation: compute the values (optimal expect) of states s



OPTIMAL UTILITIES

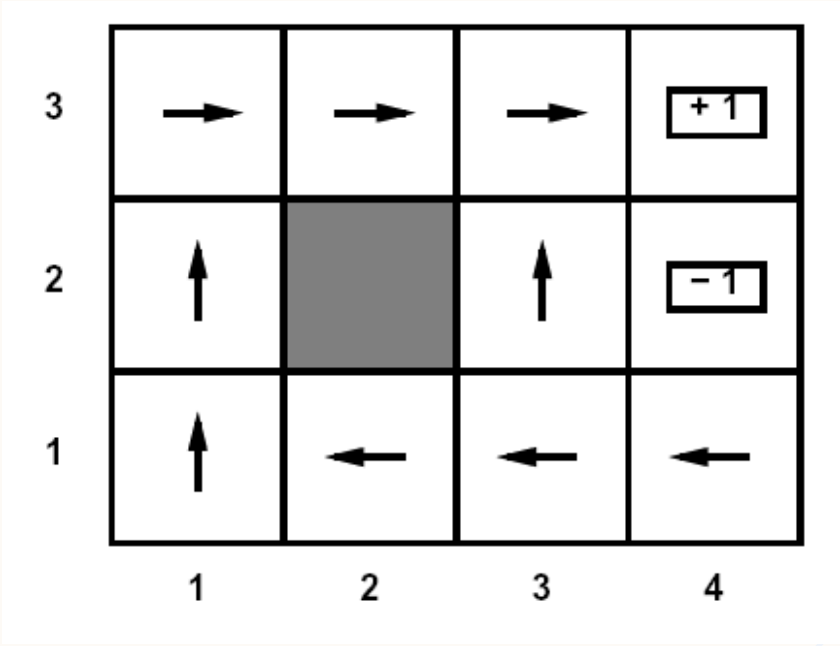
- Fundamental operation: compute the values (optimal expect) of states s





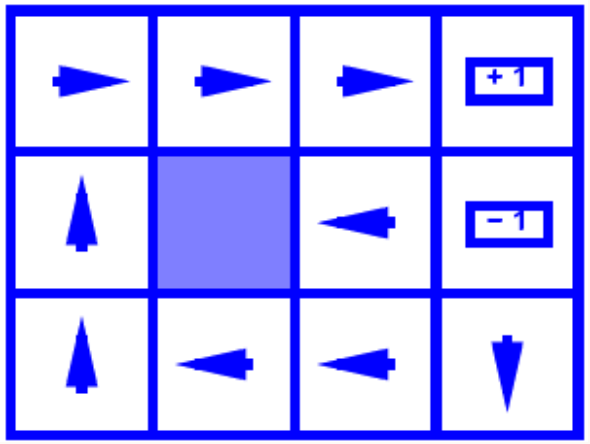
GRID WORLD

- **Environment** - The world
- **Agent** - The brain
- **Action** – What Agent can do
- **Reward** – the goal
- **State** – any possible situation
- **Policy** – the actionable strategy

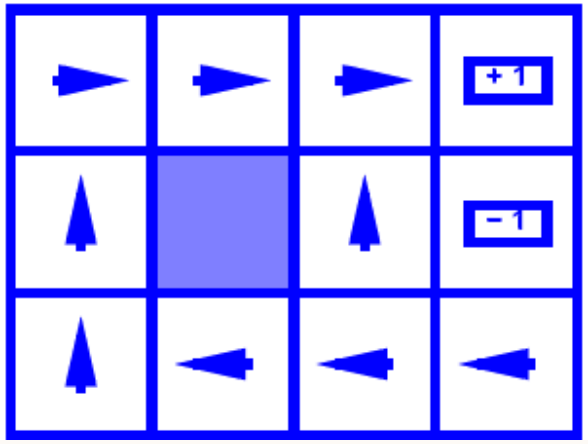


optimal policy $\pi^*: S \rightarrow A$

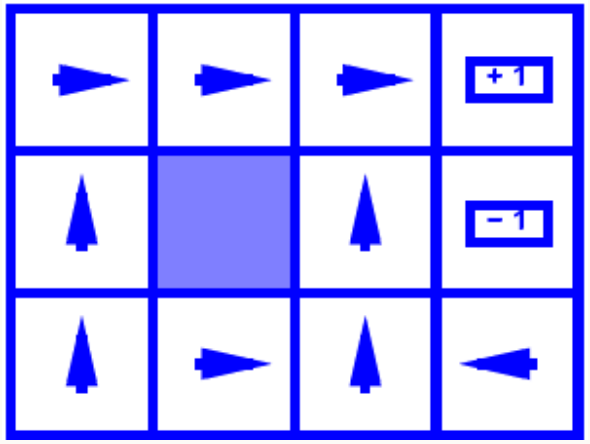
EXAMPLE OPTIMAL POLICIES



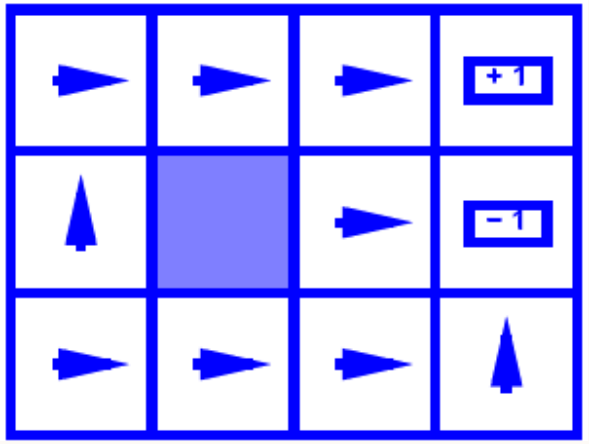
R(s) = -0.01



R(s) = -0.03



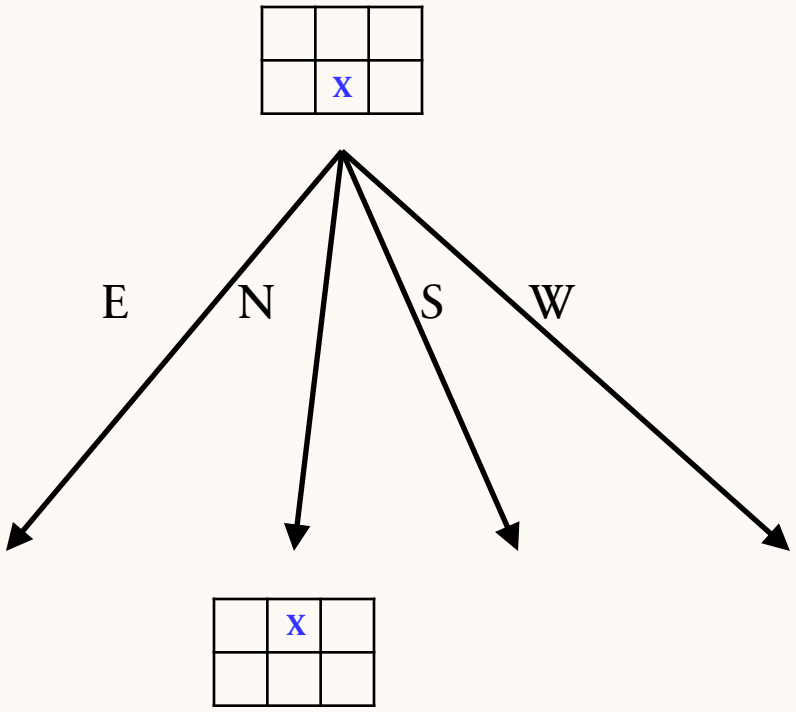
R(s) = -0.4



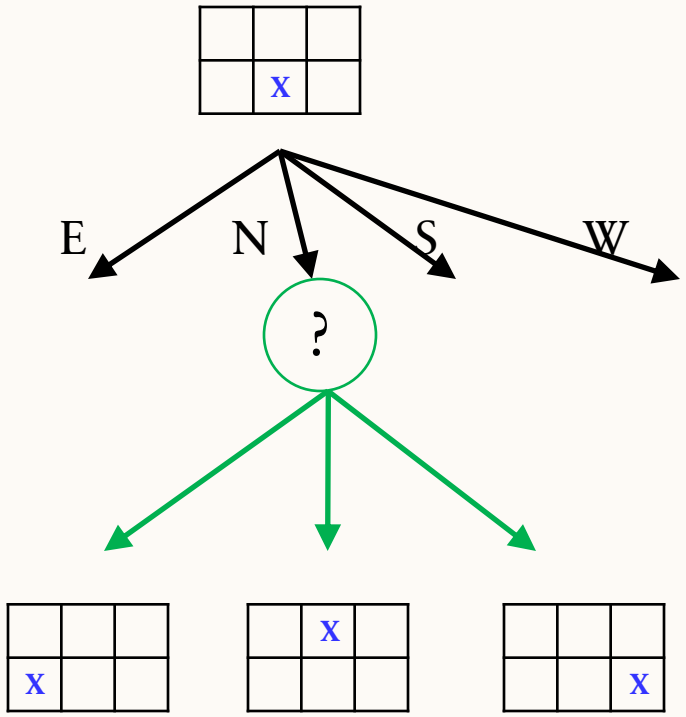
R(s) = -2.0

GRID FUTURES

Deterministic Grid World

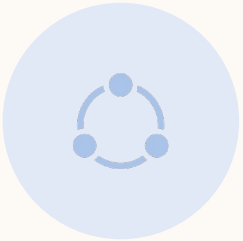


Stochastic Grid World



EVALUATION METRICS

#States generated



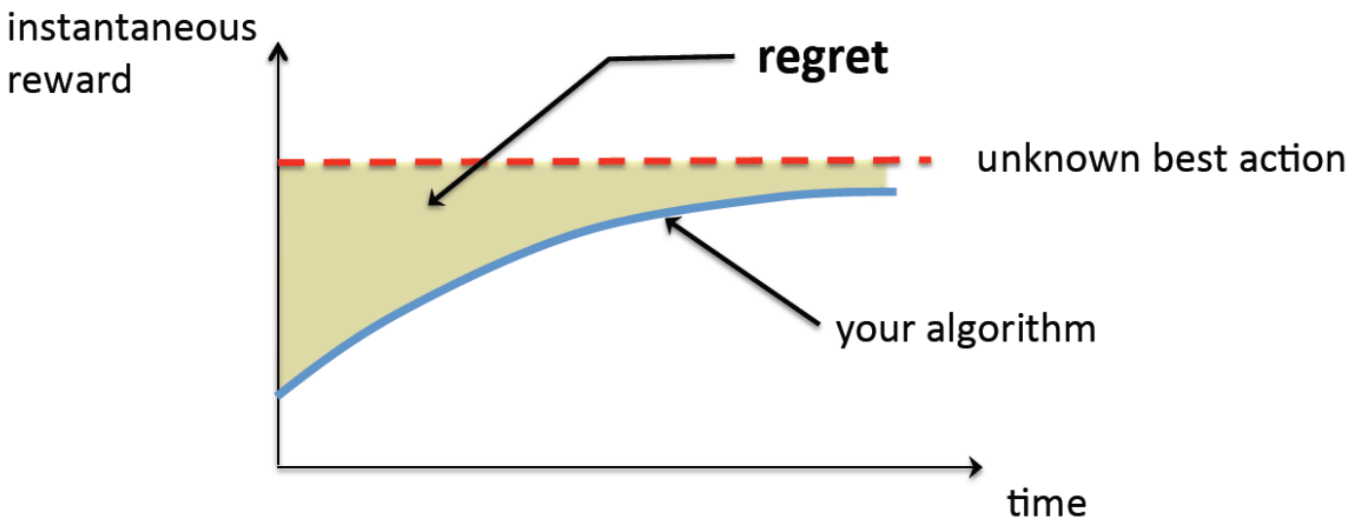
#Iterations to reach optimal policy

Rewards in terms of points earned

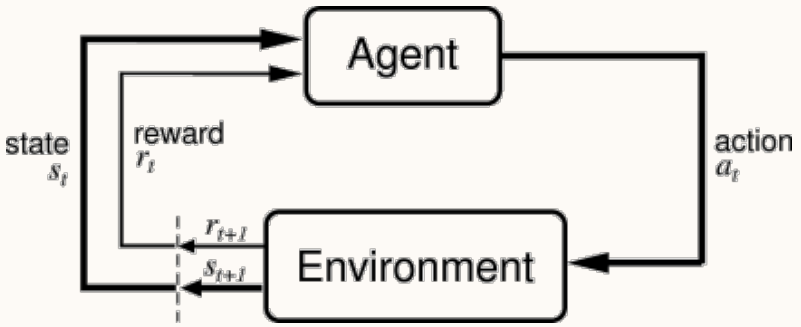
Rate of Convergence to optimal Policy

GOAL : REGRET AS PERFORMANCE METRIC

$$R(T) = \underbrace{\max_{x \in \mathcal{X}} \mathbb{E}\left[\sum_{n=1}^{T} r_n(x)\right]}_{\text{oracle}} - \underbrace{\mathbb{E}\left[\sum_{n=1}^{T} r_n(x_n)\right]}_{\text{your algorithm}}.$$



REINFORCEMENT LEARNING (RL)

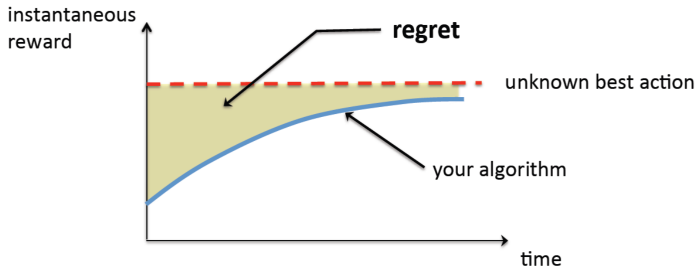


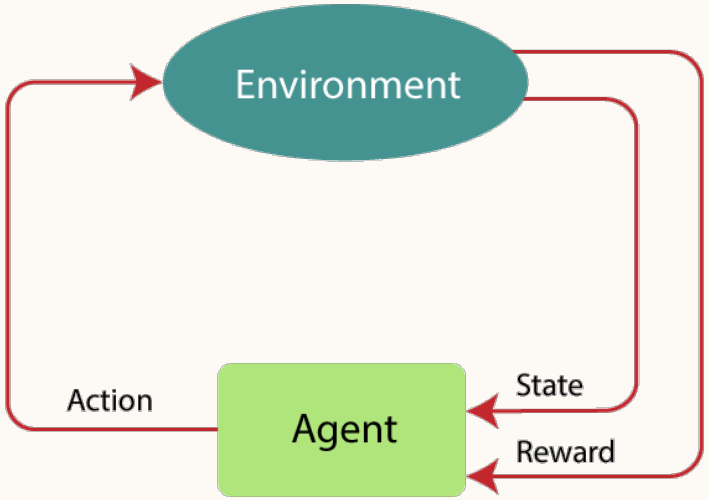
- RL algorithms attempt **to find a policy** for **maximizing** cumulative **reward** for the agent over the course of the problem.
- Typically represented by a **Markov Decision Process**
- RL differs from supervised learning in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected.

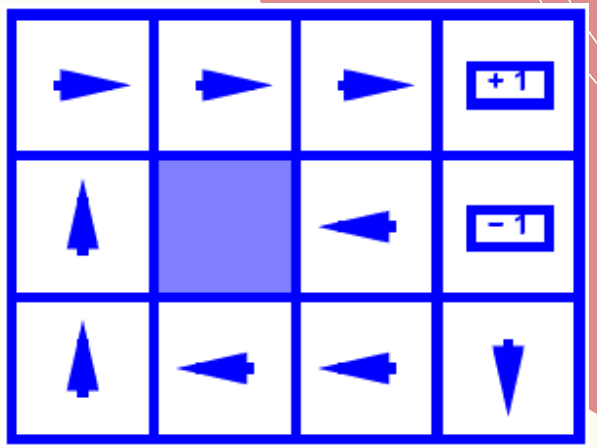
Q/A

Exploration need
Active Learning
Decision making
Problem Statement
Terminology

$$R(T) = \underbrace{\max_{x \in \mathcal{X}} \mathbb{E}\left[\sum_{n=1}^{T} r_n(x)\right]}_{\text{oracle}} - \underbrace{\mathbb{E}\left[\sum_{n=1}^{T} r_n(x_n)\right]}_{\text{your algorithm}}.$$





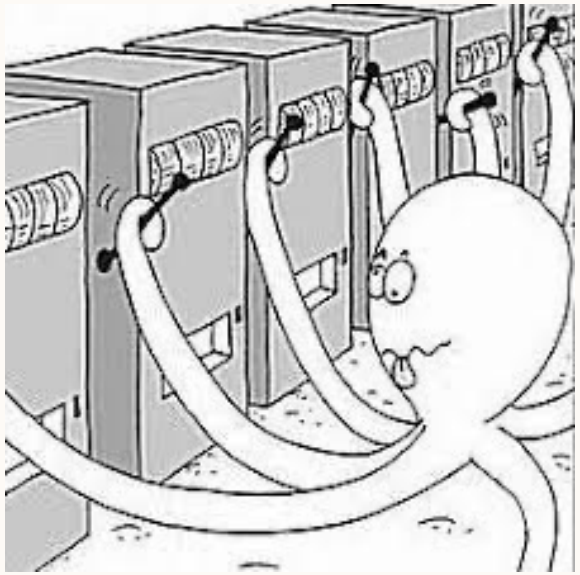


BREAK

OUTLINE

Background	Problem statement	**Multi-armed-bandit**	Real-world challenges	Summary
• Exploration need • Active learning • decision making • Use-cases and Landscape	• The Problem • Terminology and definitions	• **The task** • **General Algo** • **Main sampling strategies**	• Q-learner • Explainability	

MULTI-ARMED-BANDIT



OPTIMAL POLICY

$\mathbb{E}[a] = -\$0.5$

$\mathbb{E}[b] = -\$0.2$

$\mathbb{E}[c] = \$0.1$

$\mathbb{E}[d] = \$0.11$

LEARN A POLICY

$\mathbb{E}[a] = ?$

$\mathbb{E}[b] = ?$

$\mathbb{E}[c] = ?$

$\mathbb{E}[d] = ?$

LEARN A POLICY

-1, -1, 5

$\widehat{\mathbb{E}}[a] = 1$

-0.2, -0.2

$\widehat{\mathbb{E}}[b] = -0.2$

-0.5, -0.5, -0.5

$\widehat{\mathbb{E}}[c] = -0.5$

-2, -2

$\widehat{\mathbb{E}}[d] = -2$

LEARN A POLICY

-1, -1, 5, -1, -1, -1, -1, -1, 2, 6, -1, -1, -1, -1, -1

$\mathbb{E}[a] = -\$0.5$

-0.2, -0.2

$\mathbb{E}[b] = -\$0.2$

-0.5, -0.5, -0.5

$\mathbb{E}[c] = \$0.1$

-2, -2

$\mathbb{E}[d] = \$0.11$

NOTATION

k	Number of actions (arms)
t	Discreate time step or play number
$q_*(a)$	True value (expected reward) of action a
$Q_t(a)$	Estimate of $q_*(a)$ at time step t
$N_t(a)$	Number of times action a was selected before step t
R_t	The reward observed at time step t
A_t	The action chosen at time step t

LEARN A POLICY

$k=4$

$t=11$

$R = \{-1, -1, 5\}$

$N_{11}(a) = 3$

$q_*(a) = -\$0.5$

$Q_{11}(a) = 1$

$R = \{-0.2, -0.2\}$

$N_{11}(b) = 2$

$q_*(b) = -\$0.2$

$Q_{11}(b) = -0.2$

$R = \{-0.5, -0.5, -0.5\}$

$N_{11}(c) = 3$

$q_*(c) = \$0.1$

$Q_{11}(c) = -0.5$

$R = \{-2, -2\}$

$N_{11}(d) = 2$

$q_*(d) = \$0.11$

$Q_{11}(d) = -2$

TO EXPLORE OR TO EXPLOIT?

GREEDY ACTION = *EXPLOIT* CURRENT KNOWLEDGE

-1, -1, 5

$Q_{11}(a) = 1$

-0.2, -0.2

$Q_{11}(b) = -0.2$

-0.5, -0.5, -0.5

$Q_{11}(c) = -0.5$

-2, -2,

$Q_{11}(d) = -2$

OR *EXPLORE* FOR NEW KNOWLEDGE

-1, -1, 5

$Q_{11}(a) = 1$

-0.2, -0.2

$Q_{11}(b) = -0.2$

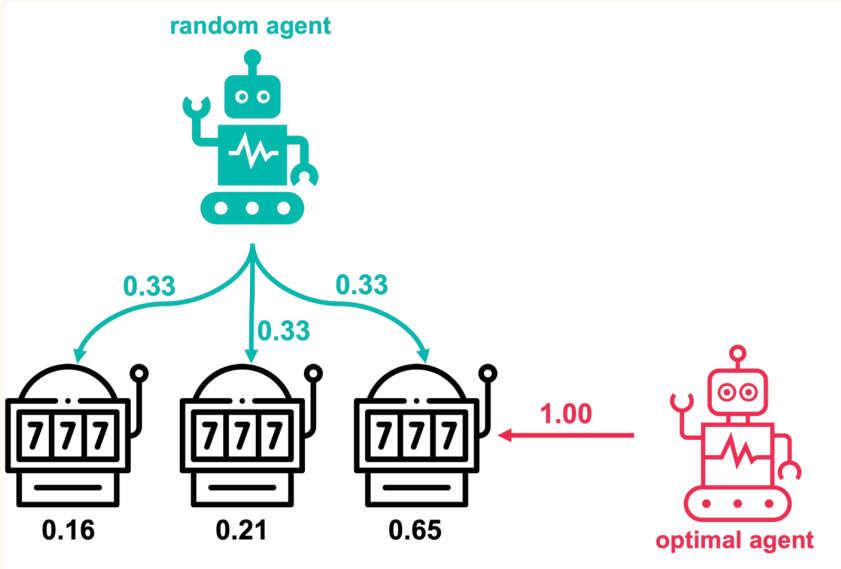
-0.5, -0.5, -0.5

$Q_{11}(c) = -0.5$

-2, -2,

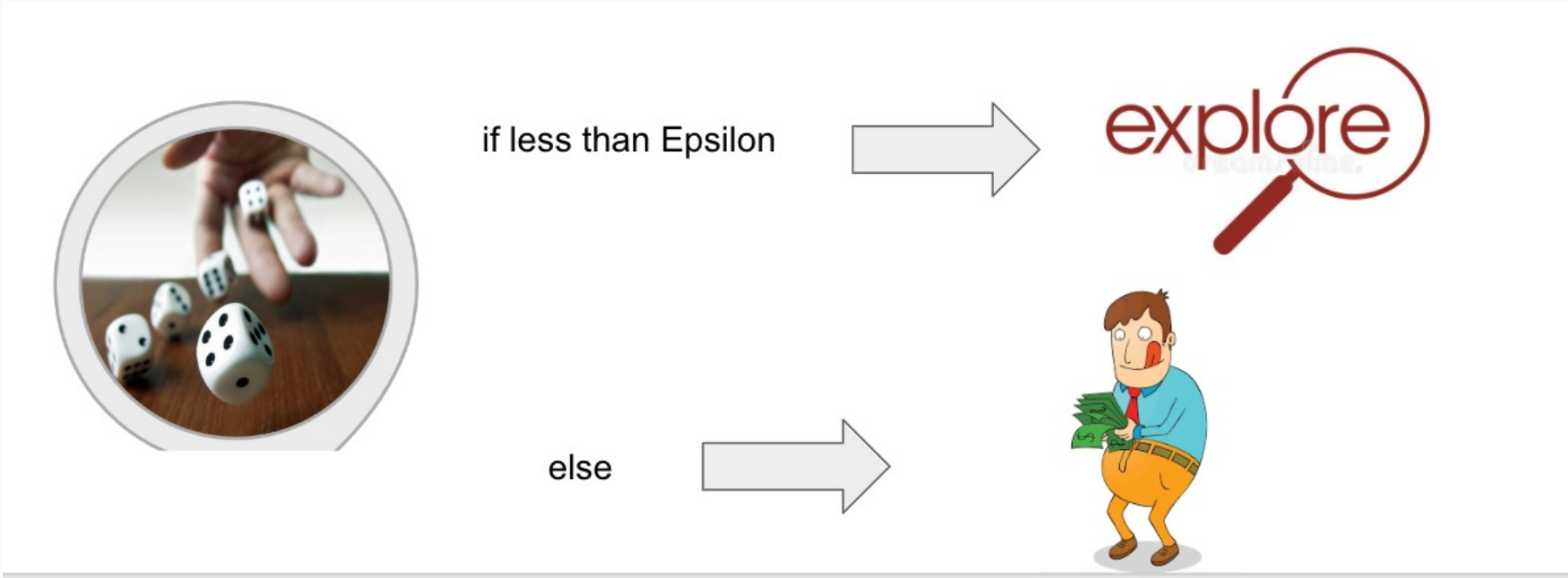
$Q_{11}(d) = -2$

RANDOM METHOD - EXPLORATION



EPSILON GREEDY

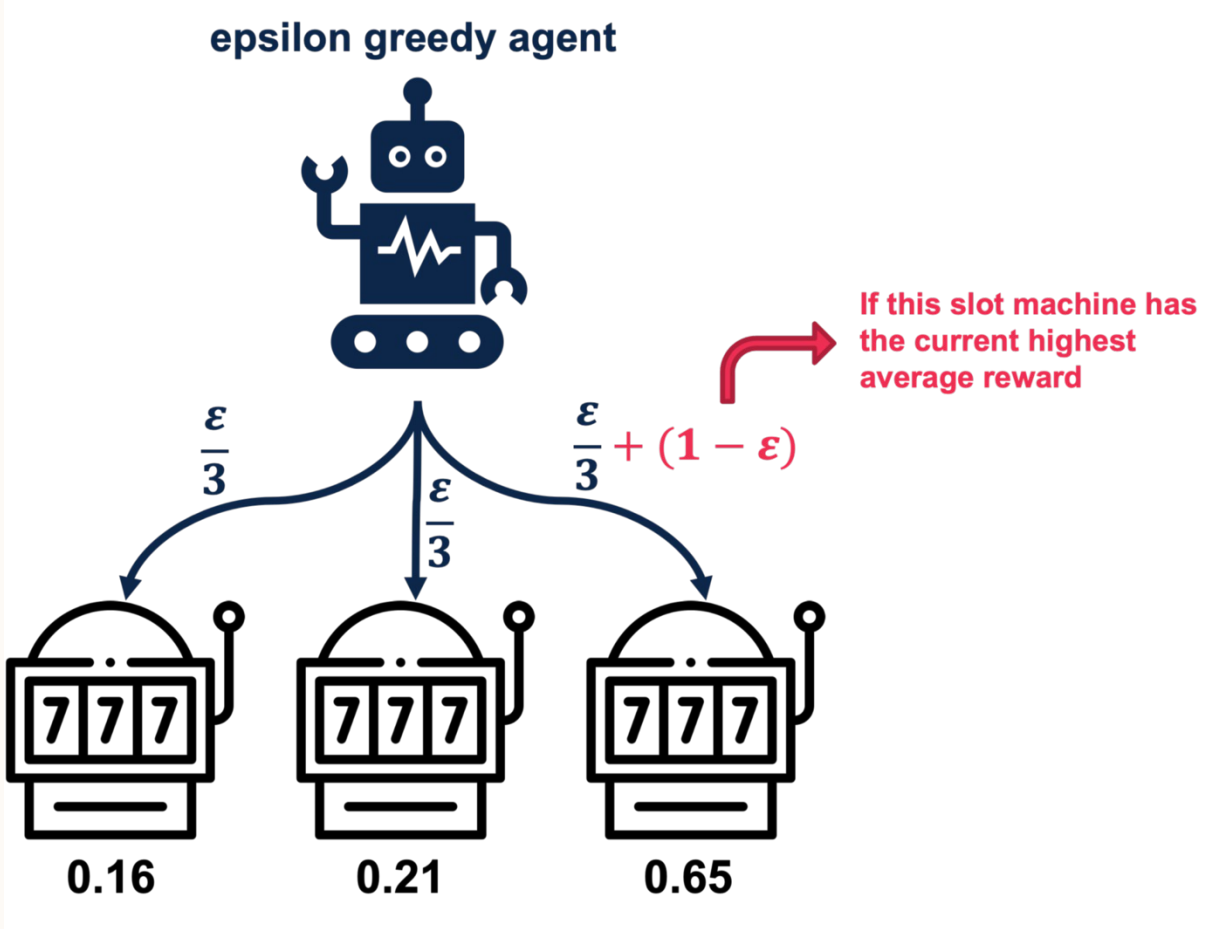
- Epsilon - the chance that we explore
- Greedy - taking the best (known) actions



FORMAL EPSILON GREEDY

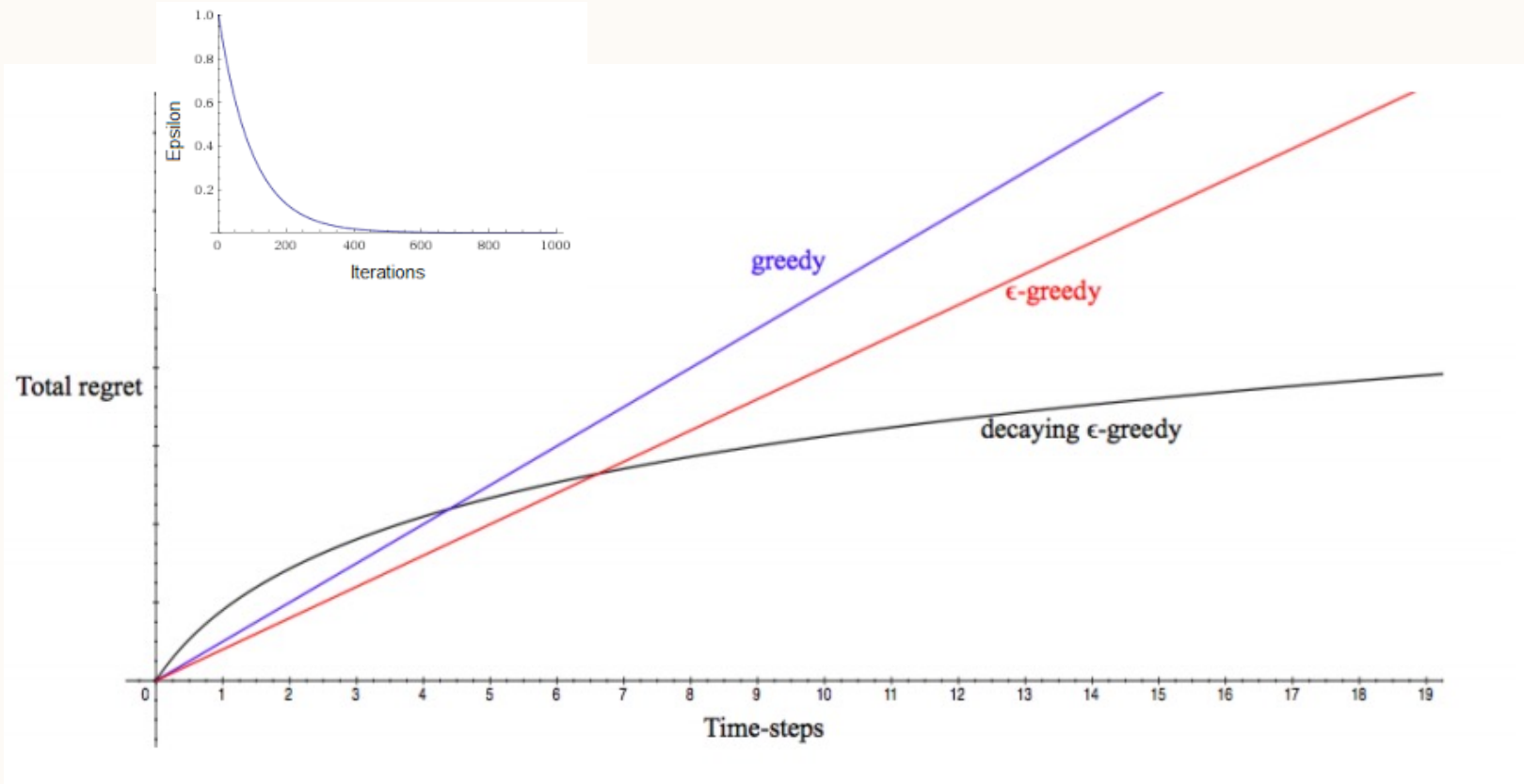
Action at time(t) $\begin{cases} \max Q_t(a) & \text{with probability } 1-\epsilon \\ \text{any action (a)} & \text{with probability } \epsilon \end{cases}$

EPSILON-GREEDY

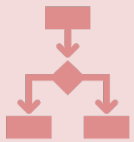


DECAYING EPSILON GREEDY

FOR EACH STEP WE WILL REDUCE THE PROBABILITY OF EXPLORATION



WEAKNESSES OF EPSILON ALGORITHMS

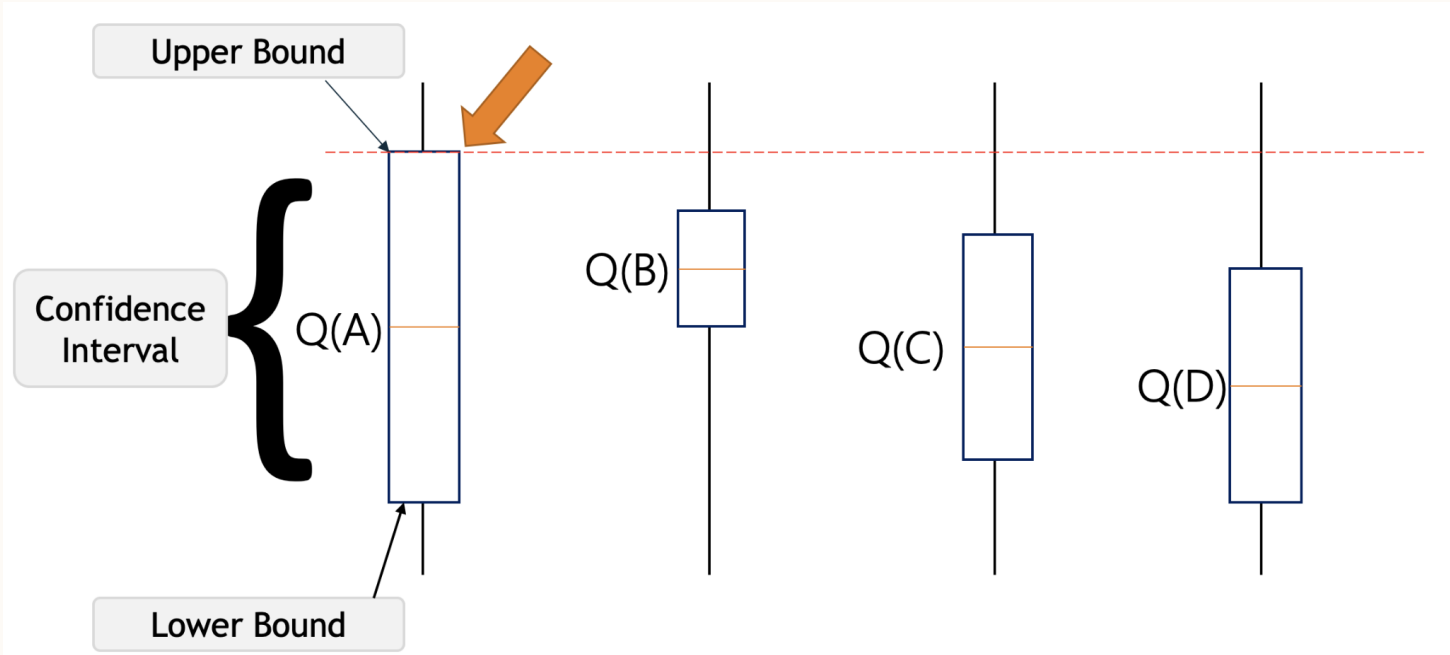


– Randomly selects an action to explore, does not explore more “promising” actions.

– Does not consider confidence interval. If an action has been taken many times, no need to explore it.

UPPER CONFIDENCE BOUND– FREQUENTIST APPROACH

- Add the confidence interval for each action taken.
- The more we take the action, the less exploration we need.



THE MATH

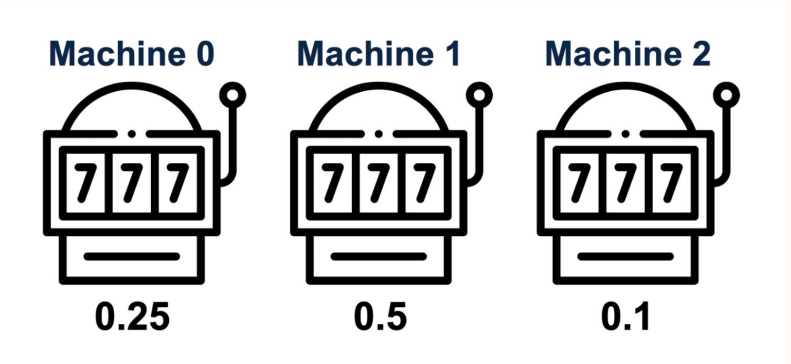
$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right]$$

where;

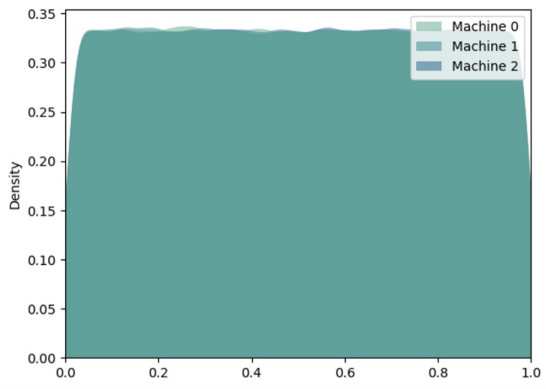
- $Q_t(a)$ *is the estimated value of action 'a' at time step 't'.*
- $N_t(a)$ *is the number of times that action 'a' has been selected, prior to time 't'.*
- *'c' is a confidence value that controls the level of exploration.*

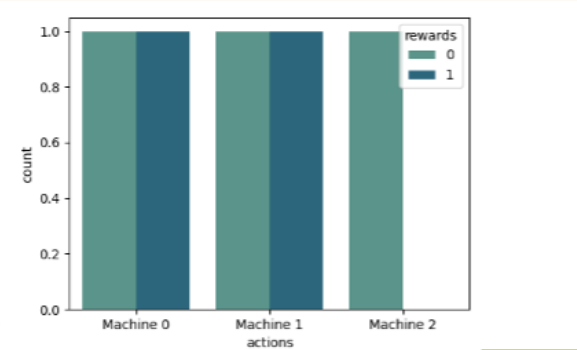
TOMPSON METHOD

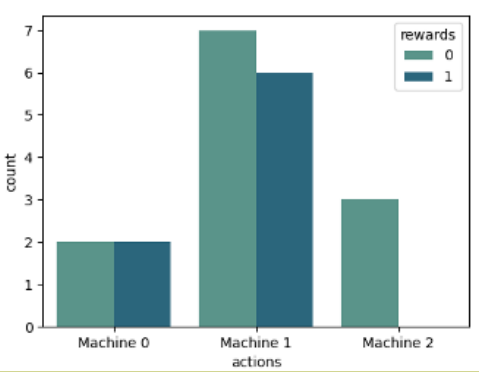
- Build a reward probability distribution (most commonly a beta distribution, for computational reasons)
- Sample according to assumed distribution
- Update the distribution according to reward

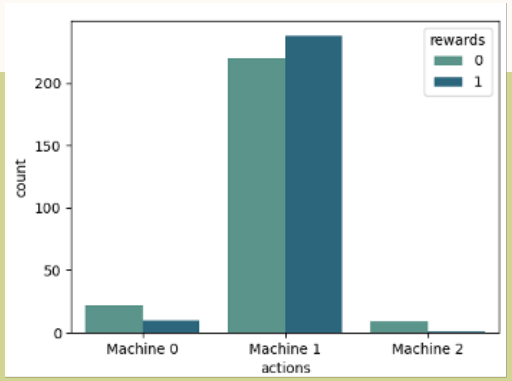


TOMPSON METHOD









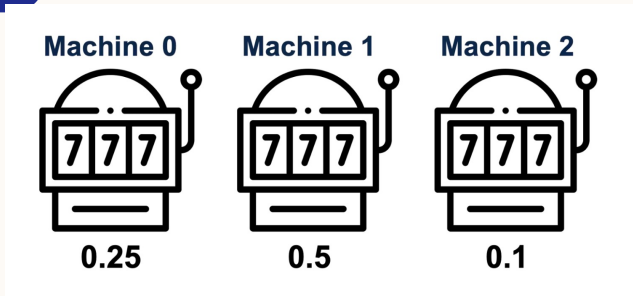
INITIAL

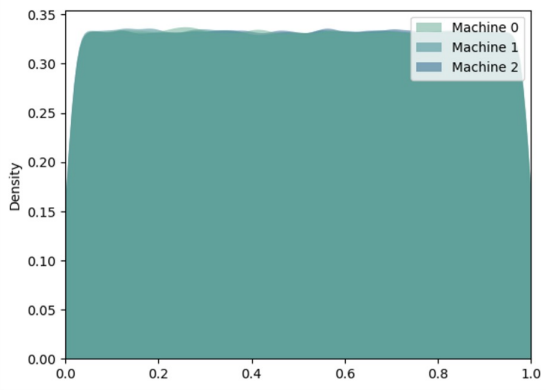
5 ITERATIONS

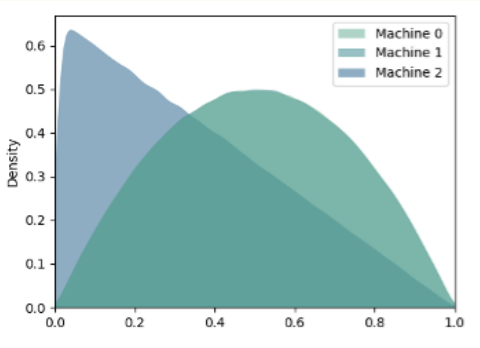
2 ITERATIONS

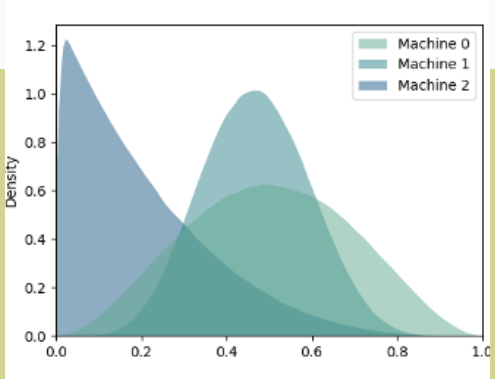
500 ITERATIONS

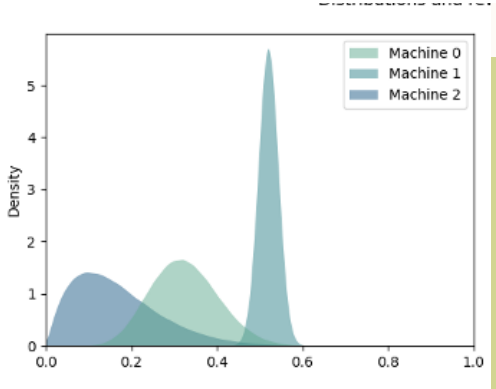
TOMPSON METHOD











INITIAL

5 ITERATIONS

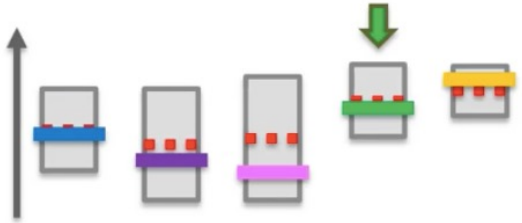
2 ITERATIONS

500 ITERATIONS

COMPARISON

Thompson Sampling Algorithm

UCB

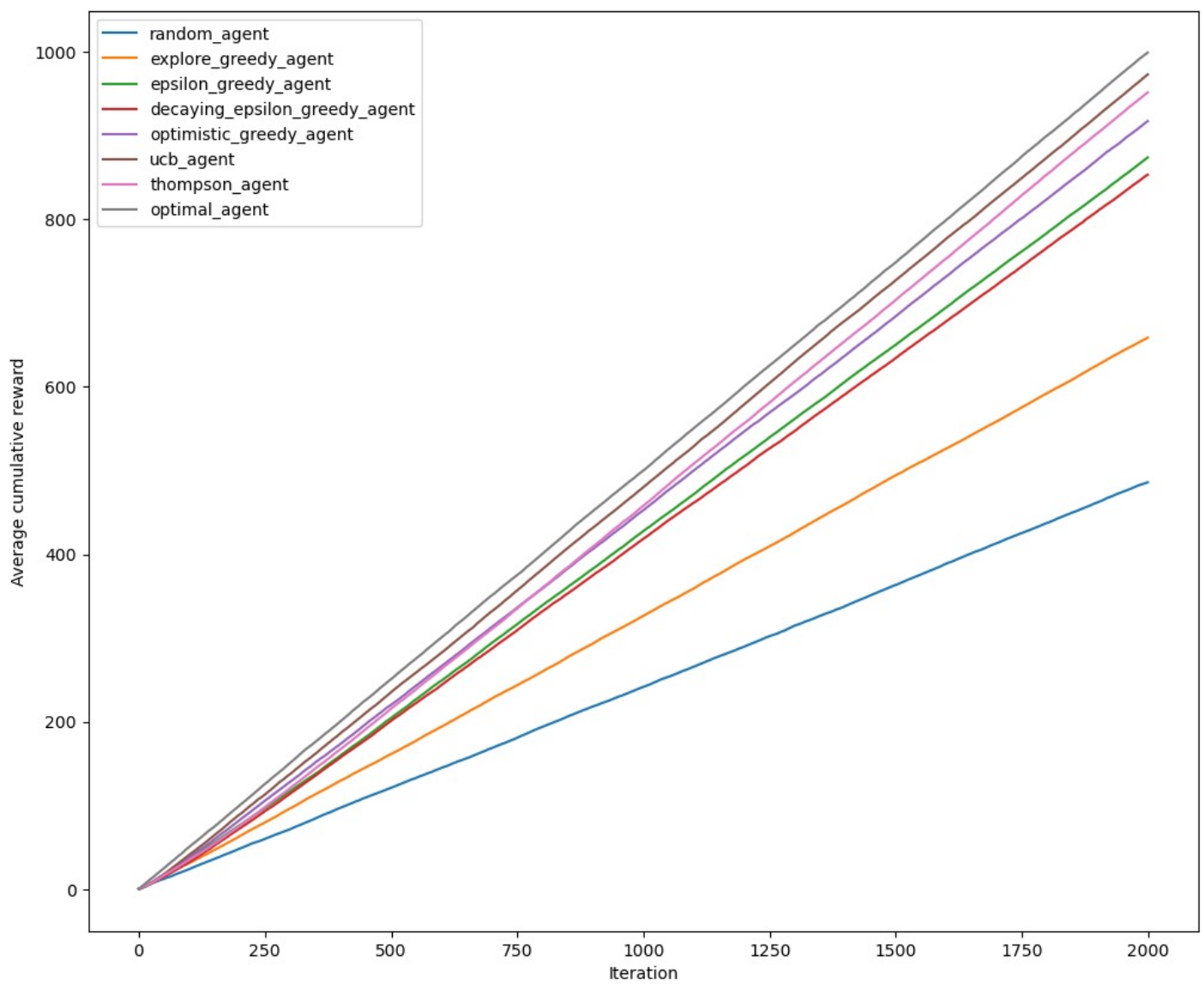


- Deterministic
- Requires update at every round

Thompson Sampling

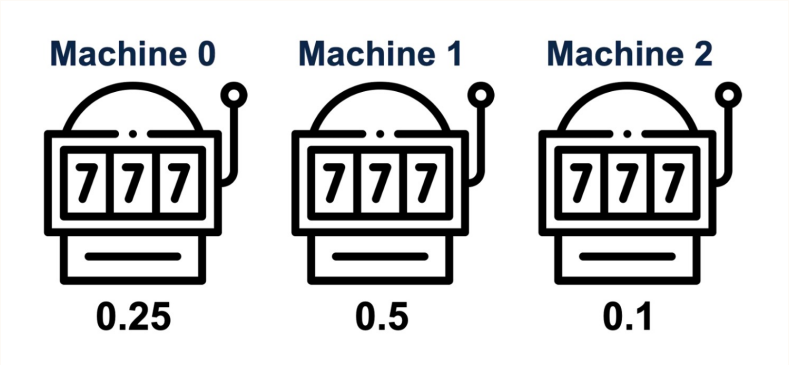
- Probabilistic
- Can accommodate delayed feedback
- Better empirical evidence

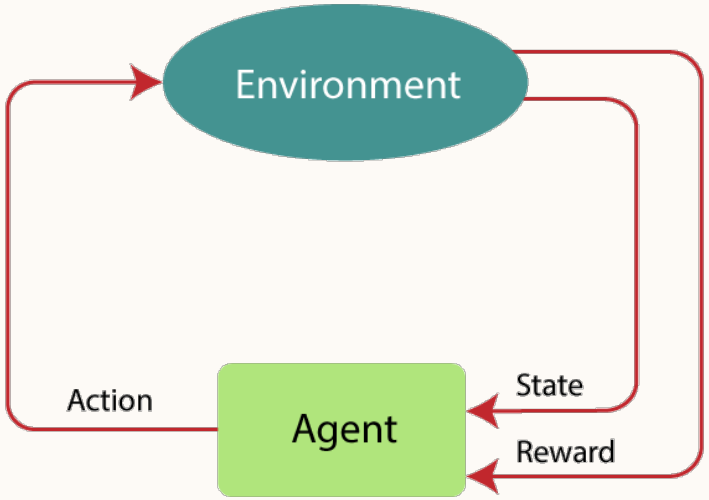
COMPARE

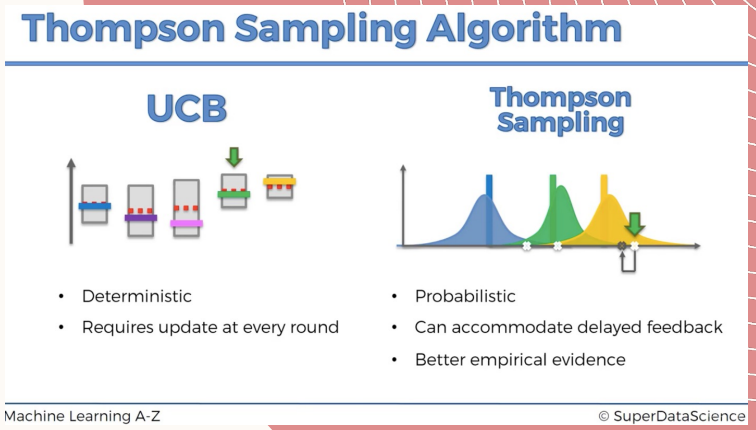

random_agent
explore_greedy_agent
epsilon_greedy_agent
decaying_epsilon_greedy_agent
optimistic_greedy_agent
ucb_agent
thompson_agent
optimal_agent
Average cumulative reward
Iteration
0
200
400
600
800
1000
0
250
500
750
1000
1250
1500
1750
2000

Q/A

Problem Statement
Terminology
UCB vs Tompson







LET'S PLAY

https://github.com/WhatIThinkAbout/BabyRobot/tree/master/Multi_Armed_Bandits

BREAK

OUTLINE

Background	Problem statement	Multi-armed-bandit	**Real-world challenges**	Summary
• Exploration need • Active learning • decision making • Use-cases and Landscape	• The Problem • Terminology and definitions	• The task • General Algo • Main sampling strategies	• **Too many parameters** • **Reward is not a loss** • **Explainability**	

STABLE ENVIROMENT – LARGE DATA

Q-LEARNER

•Q-learning is a popular model-free reinforcement learning algorithm based on the Bellman equation.

•**The main objective of Q-learning is to learn the policy which can inform the agent that what actions should be taken for maximizing the reward under what circumstances.**

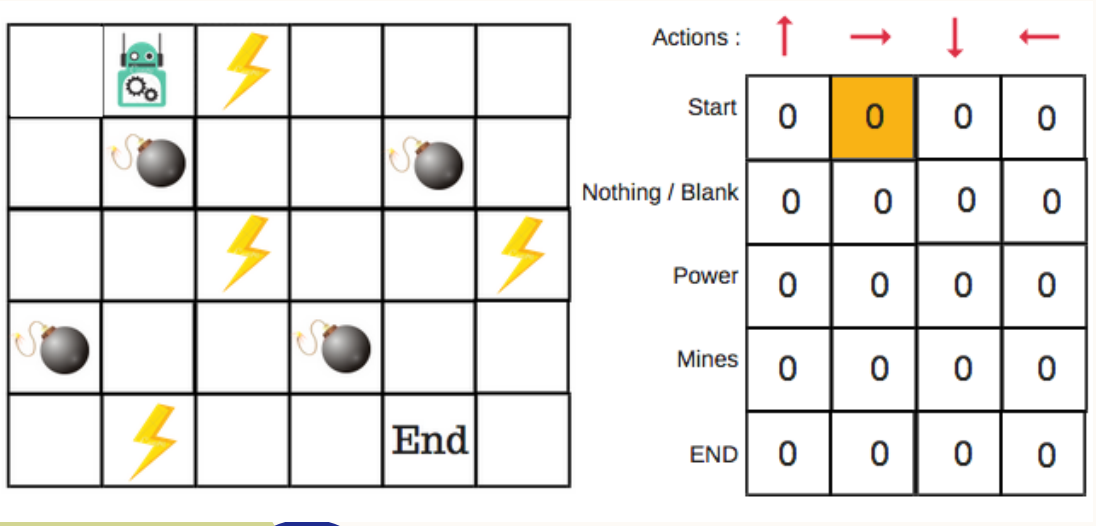
•It is an **off-policy RL** that attempts to find the best action to take at a current state.

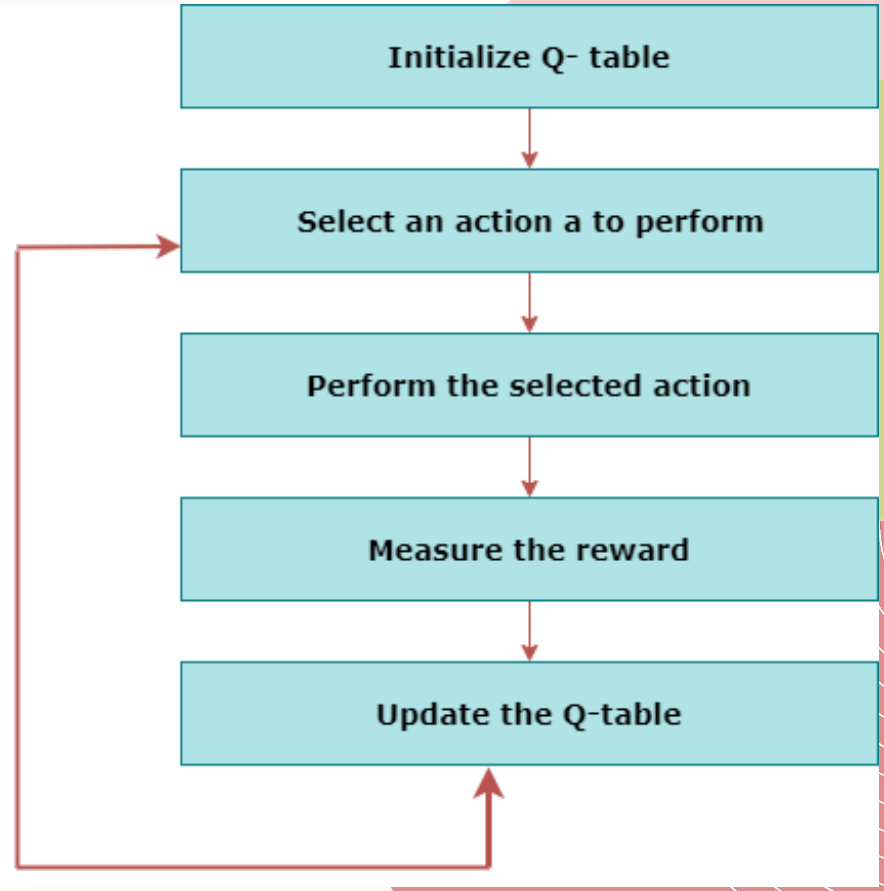
•The goal of the agent in Q-learning is to maximize the value of Q.

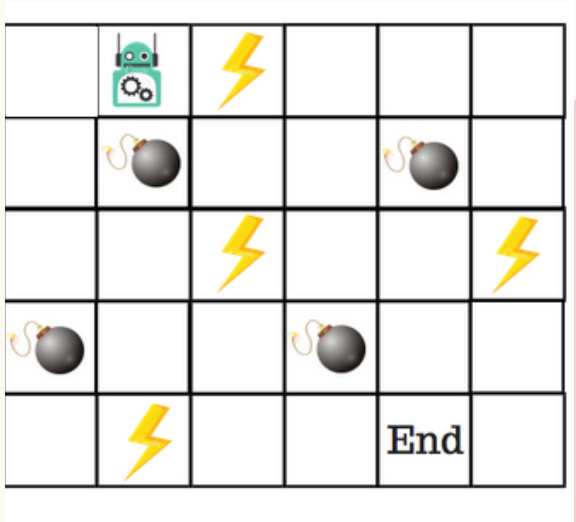
•The Q stands for **quality** in **Q-learning**, which means it specifies the quality of an action taken by the agent.

$$Q(S, a)= R(s, a)+ \gamma\sum_{s'}(P(s, a, s')\max Q(s',a`))$$

Q-LEARNER


End
Actions :
Start
Nothing / Blank
Power
Mines
END
0


Initialize Q- table
Select an action a to perform
Perform the selected action
Measure the reward
Update the Q-table



ML-LEARNING

In realistic situations, we cannot possibly learn about every single state!

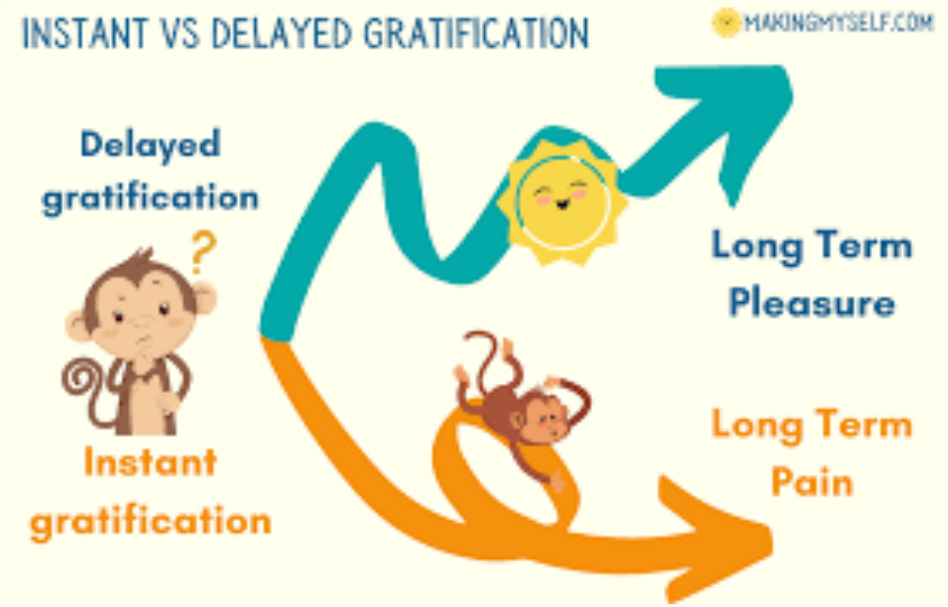
- **Too many states to visit them all in training**
- Too many states to hold the q-tables in memory

Instead, we want to generalize:

- Learn about some small number of training states from experience
- Find vectors to describe state/action
- Generalize that experience to new, similar states
- This is a fundamental idea in machine learning, and we'll see it over and over again

SPARSE REWARD

DELAYED REWARD



NEXT LEVEL - AGENT SUMMARIZATION

What should we optimize for in selecting state-action pairs for the summary?

NEXT LEVEL - AGENT SUMMARIZATION

Compute state importance using Q-values (Torrey & Taylor 2014)

$$I_s = Q_{s,a*} - Q_{s,a-}$$

Expected future rewards for best action

Expected future rewards for worst action

Important!

Not so important…

NEXT LEVEL – MANY AGENT SUMMARIZATION

- Disagreement-Based Summaries

WHY NOT ALWAYS REINFORCEMENT LEARNING???

Too Many things to learn

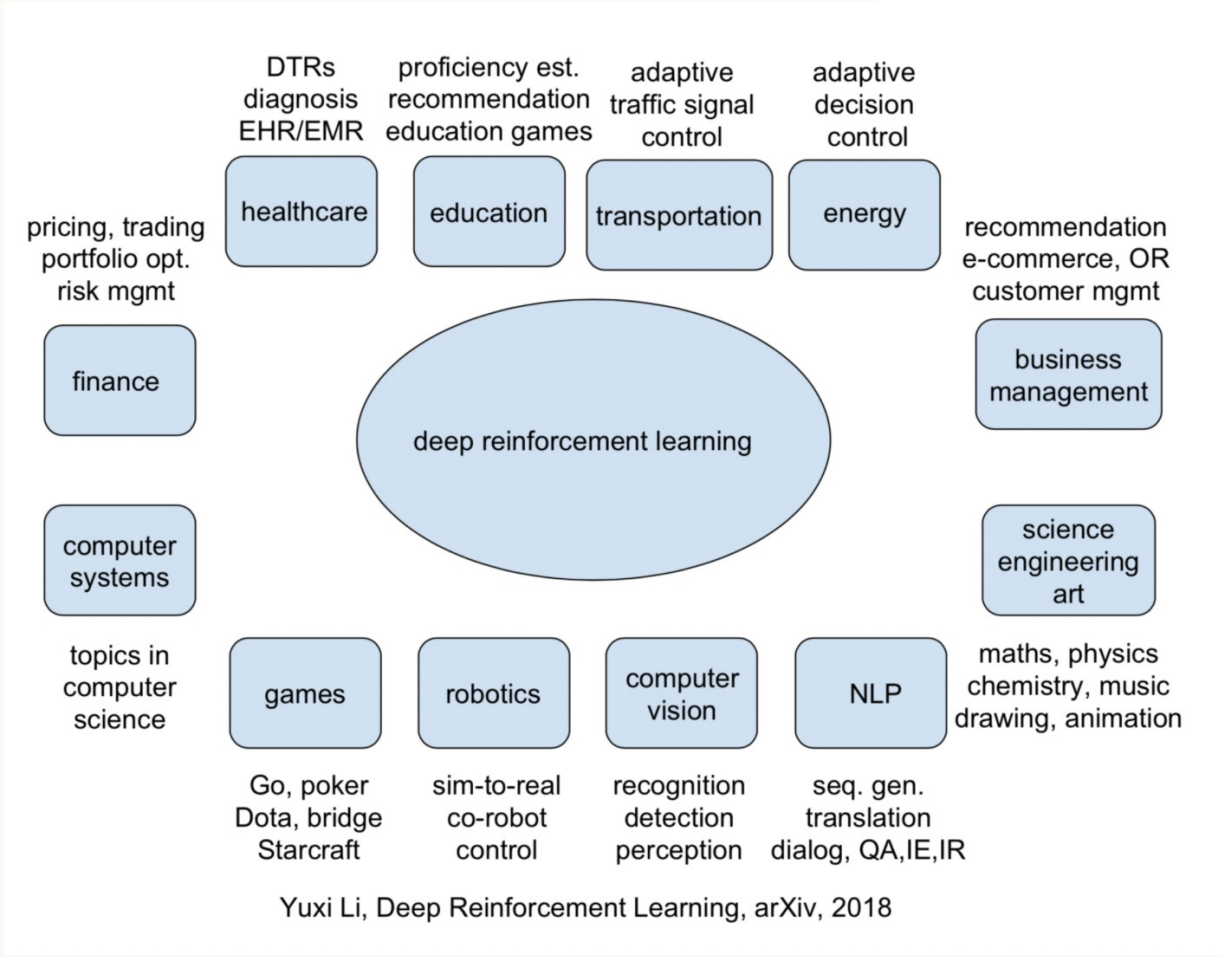
No need of exploration – stable environment

Delayed feedback (Would I survive drinking Coca-Cola?)

Sparse reward (long games)

Agent explainability

MANY USE-CASES

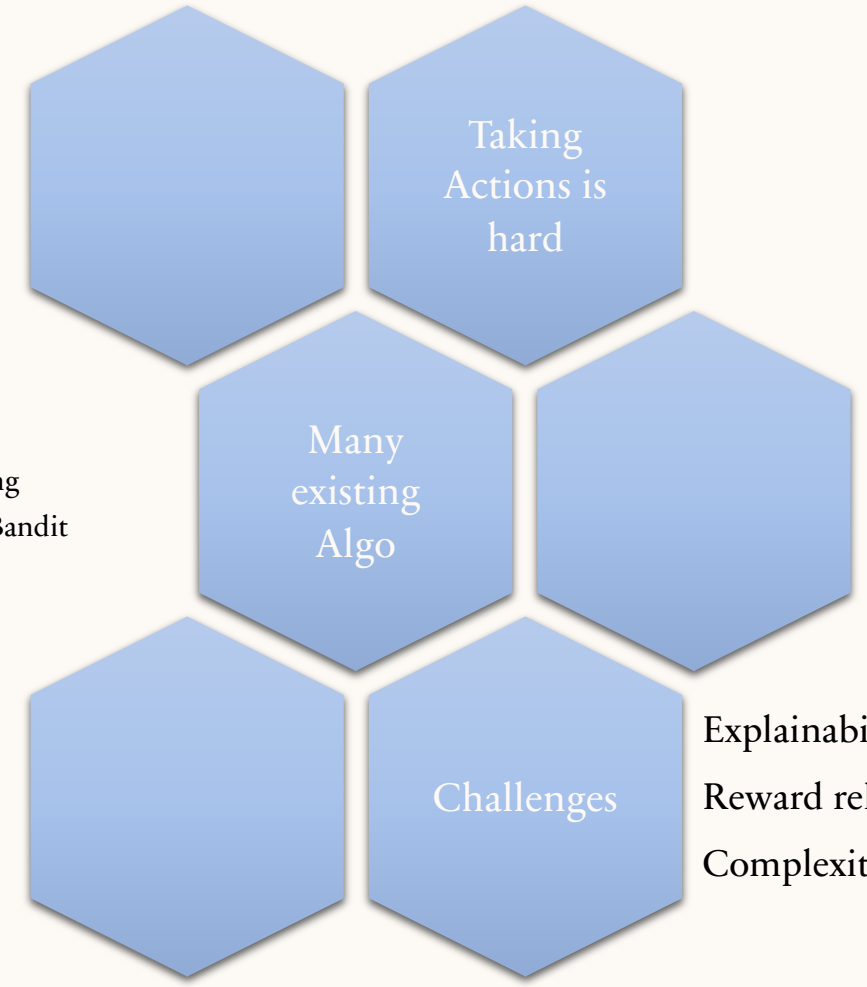


Yuxi Li, Deep Reinforcement Learning, arXiv, 2018

SUMMERY AND Q/A

We saw :

- Active Learning
- Multi-armed Bandit



Explainability

Reward related

Complexity

THANK YOU!


Happy Saturday everyone!Hope you guys have a great day and a wonderful Weekend!

REFERENCES

- https://www.youtube.com/watch?v=SgC6AZss478
- https://github.com/WhatIThinkAbout/BabyRobot/blob/master/Multi_Armed_Bandits/Part%202%20-%20The%20Bandit%20Framework.ipynb
- https://gist.github.com/mick001/f53dcd51ff59817783a4b505a4835453#file-q-learning-py
- https://firsttimeprogrammer.blogspot.com/2016/09/getting-ai-smarter-with-q-learning.html#more
- https://towardsdatascience.com/active-learning-behind-the-scenes-7e82f024a5db
- https://towardsdatascience.com/comparing-multi-armed-bandit-algorithms-on-marketing-use-cases-8de62a851831
- https://www.viralml.com/video-content.html?v=nSxaG_Kjw_w
- https://tech-ai.technion.ac.il/wp-content/uploads/2022/03/Agent-Policy-Summarization_Describing-Agent-Behavior-to-People_Ofra-Amir.pdf
- https://www.marktechpost.com/2021/03/03/introduction-to-reinforcement-learning/
- https://www.javatpoint.com/reinforcement-learning