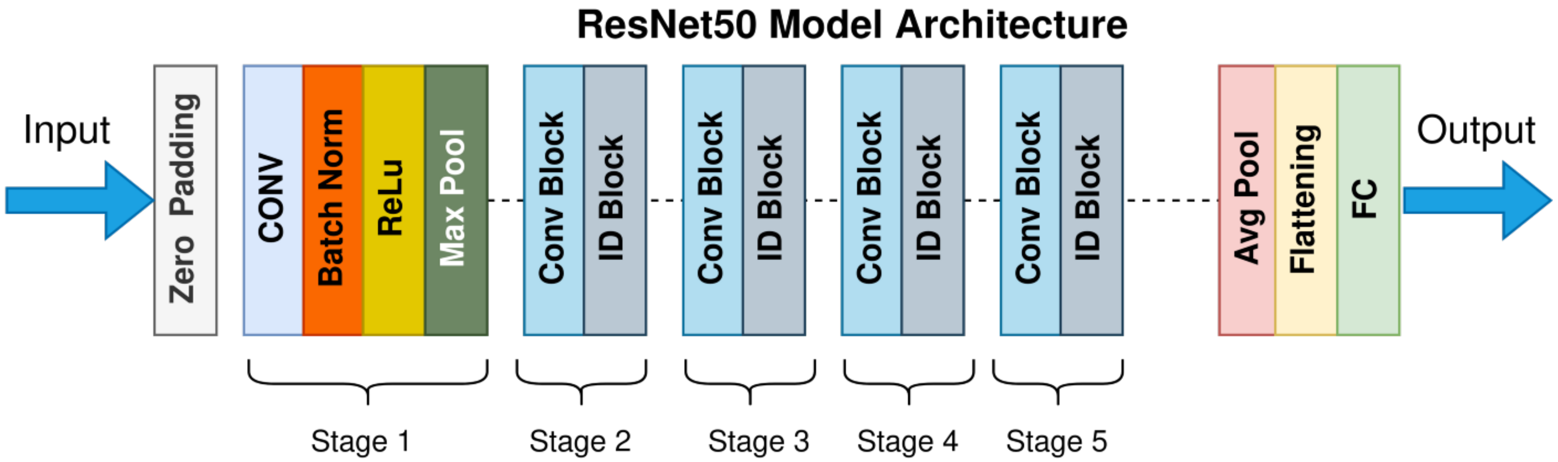
Offline Reinforcement Learning

Assurance for high-stakes applications

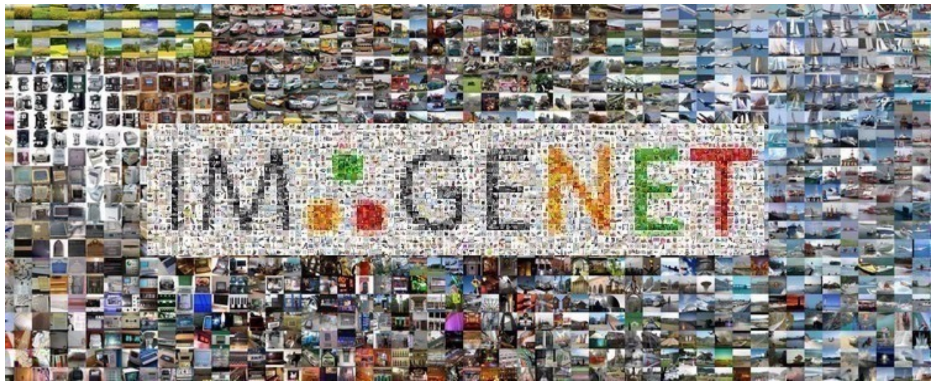
Thanh Nguyen-Tang & Raman Arora
Department of Computer Science, Whiting School of Engineering, JHU
{nguyent,arora}@cs.jhu.edu

What makes modern machine learning work?

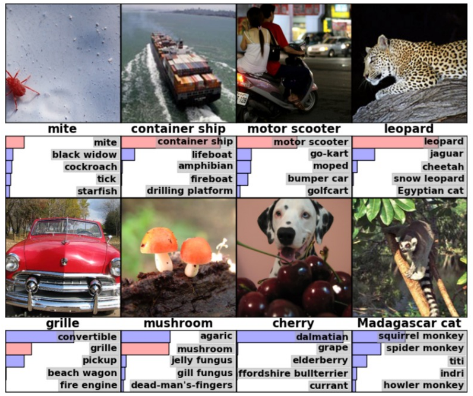
- Big models



- Big data

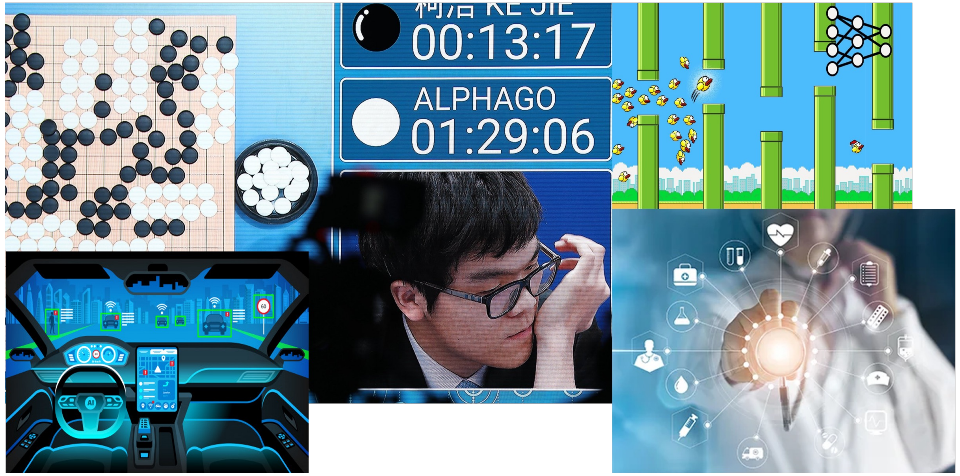


From Prediction to Decision-Making



Supervised learning:

- i.i.d. data
- Ground truth supervision
- Objective: to predict the right label

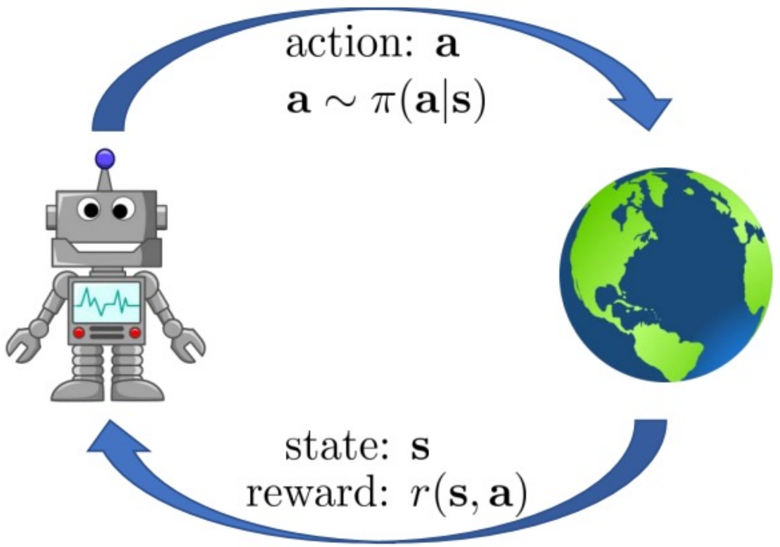


Reinforcement Learning:

- Each decision changes the future inputs
- No supervision, only abstract goal with delayed feedbacks
- Objective: to accomplish the task

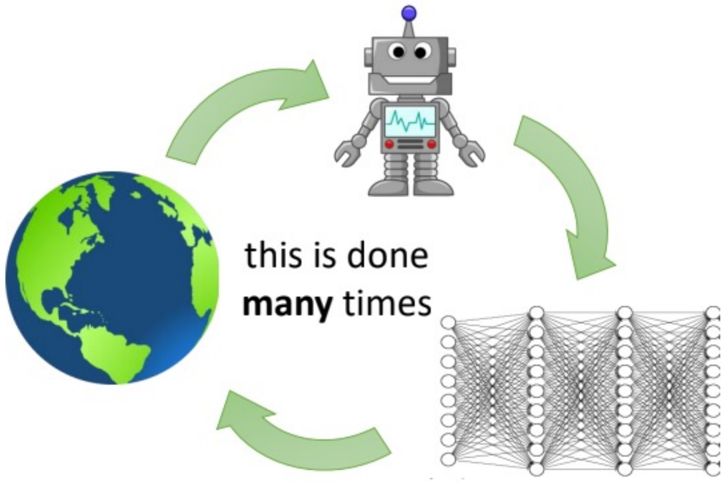
What is Reinforcement Learning (RL)?

at deployment time:



1. deploy the trained policy
2. interact
3. iterate

at training time:

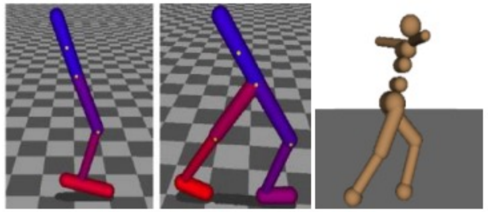


1. try a candidate policy
2. collect data
3. train
4. iterate

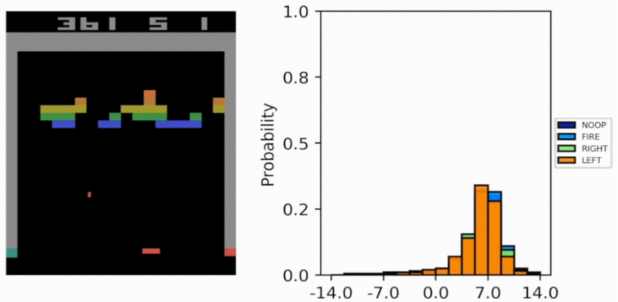
Does (online) RL work?

Yes! But only when **online interaction** is feasible and plentiful!

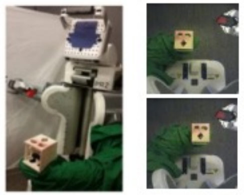
Mnih et al. '13



Schulman et al. '14 & '15



Nguyen-Tang et al. '21

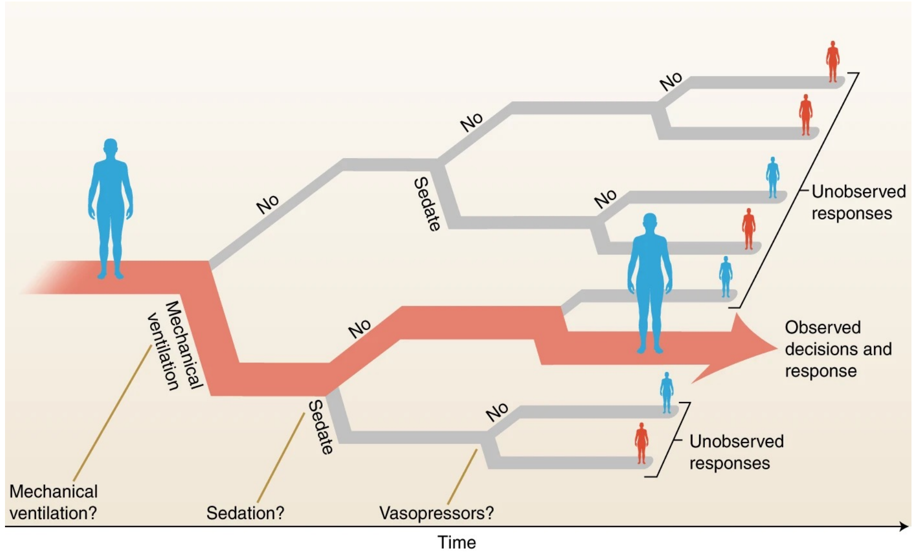


Levine*, Finn*, et al. '16

Kalashnikov et al. '18

Safety and Assurance in AI

- Online RL may be risky, unethical or prohibitive in high-stakes applications such as **self-driving cars, financial investment** and **clinical diagnosis**

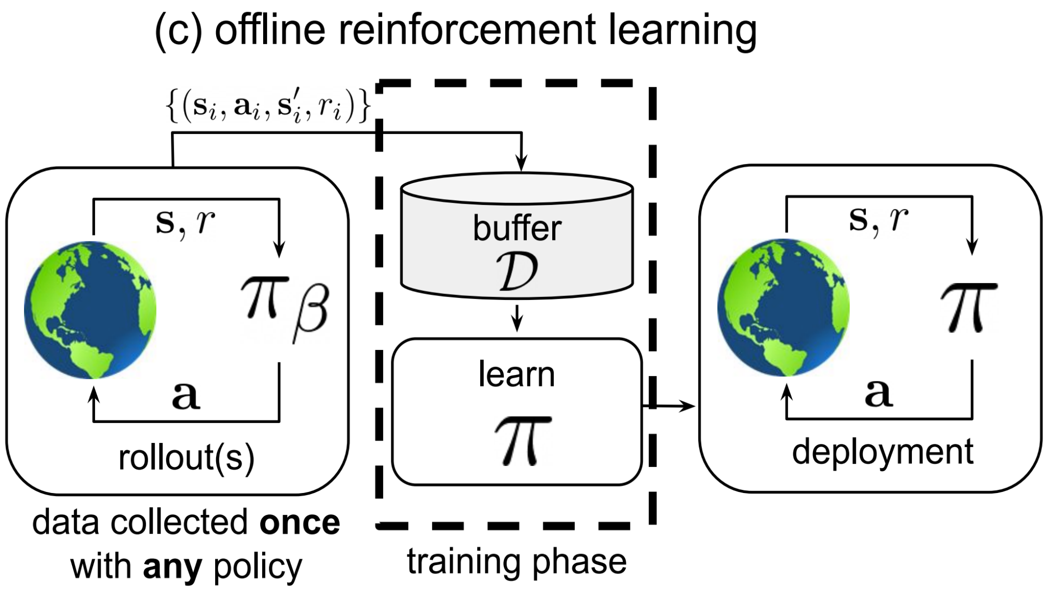


E.g. In **dynamical treatment regimes**, it is regarded unethical to actively collect the treatment effects of a potential treatment policy in patients.

Figure from Gottesman et al. "*Guidelines for reinforcement learning in healthcare*". Nature Medicine, 2019

Offline RL

1. <u>Leverage</u> ***observational*** dataset
   - Past experiences that have been collected previously
2. <u>Train</u> an offline RL algorithm in this observational dataset to learn a policy
3. <u>Deploy</u> the learned policy in the real world



Advantages:

- No exploration
- Flexibility to incorporate big data (and big models)

Levine, Kumar, Tucker, Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. ‘20

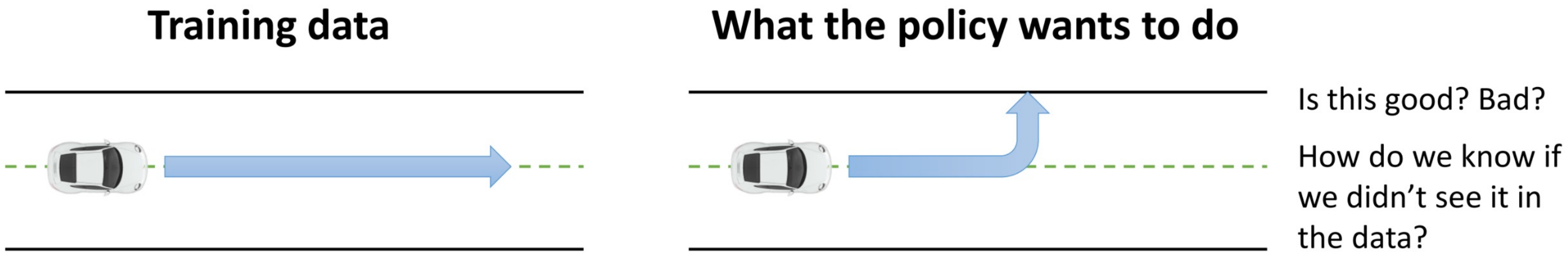
Offline RL as **Assurance** in high-stakes applications

- **Healthcare**
  - The iterative process of diagnosing and treating a patient is a RL problem
  - Evaluating a treatment policy in question in patients is dangerous
  - Offline RL: use historical treatment data to refine a new treatment policy

- **Financial investment**
  - RL can iteratively learn from a stock market
  - Online RL requires the large number of interactions with the market to learn a useful trading policy
    - By the time it learns a useful strategy, it already incurs a huge financial loss for its exploration
  - Offline RL: leverage the big historical trading data to learn a useful trading strategy
    - Minimize the risk of financial loss just for exploration

What is **hard** about offline RL?

Fundamental problem: decision making under distributional shifts (i.e., counterfactual learning)



- **Online** RL does not have this problem
- **Offline** RL must
  - account for **out-of-distribution** actions
  - **generalize** from the best behavior seen in the observational data

Key Results / Contribution I

- **NT**, **Arora**. "*Provably efficient neural offline RL via perturbed rewards*". Under review for ICLR 2023.
  - For **generalization** to unseen states in a large state space, we use deep neural network to approximate value functions
  - **Computational efficiency**: we give a polynomial time algorithm based on perturbing rewards in offline data and training multiple deep neural networks using SGD
  - **Statistical efficiency**: our approach finds an optimal policy using a polynomial number of samples, without a uniform data coverage assumption

Key Results / Contributions II

- **NT**, Yin, Gupta, Venkatesh, **Arora**. "*On instance-dependent bounds for offline RL with linear function approximation*". Under review for AAAI 2023.
  - Under a gap assumption (reward associated with an optimal action is well separated from that of sub-optimal arms), we show faster rates of convergence (from $\frac{1}{\sqrt{K}}$ to $\frac{1}{\mathrm{K}}$ where K is the number of offline samples) using a computationally efficient algorithm

Thank you