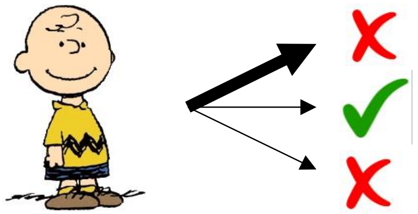
Summary

Chen-Yu Wei

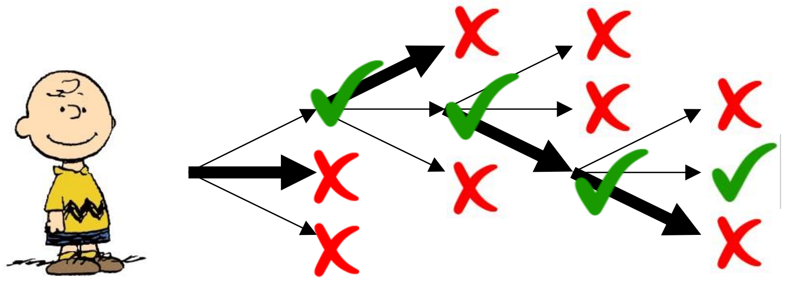
What is Reinforcement Learning?

- Learning to act from reward feedback?
- Learning to make sequential decisions?

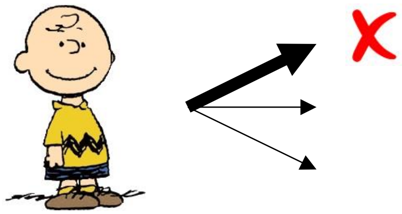
After this course, there should be a deeper understanding about it — RL (or this course) is just “supervised” learning techniques with **partial feedback** (or **weaker supervision**).



Full-information learning w/o long-term effect

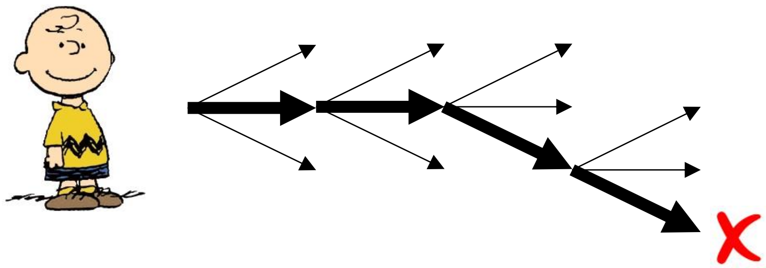


Full-information learning with long-term effect



Bandit-information learning w/o long-term effect

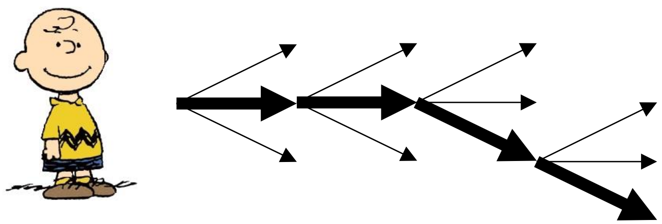
Need exploration



Bandit-information learning with long-term effect

Need exploration & credit assignment

Classification on Full-Information Long-Term Problems



Full-information about $Q^\star(s,a)\ \forall a$ or $\underset{a}{\operatorname{argmax}}\, Q^\star(s,a)$

The training data has already done credit assignment

$\Rightarrow$ Can just imitate the expert (Car driving)

Full-information about $P(\cdot\,|s,a)$ and $R(s,a)\ \forall a$

The learner still has to perform credit assignment.

$\Rightarrow$ 1. Full-information VI/PI (might be computationally infeasible)

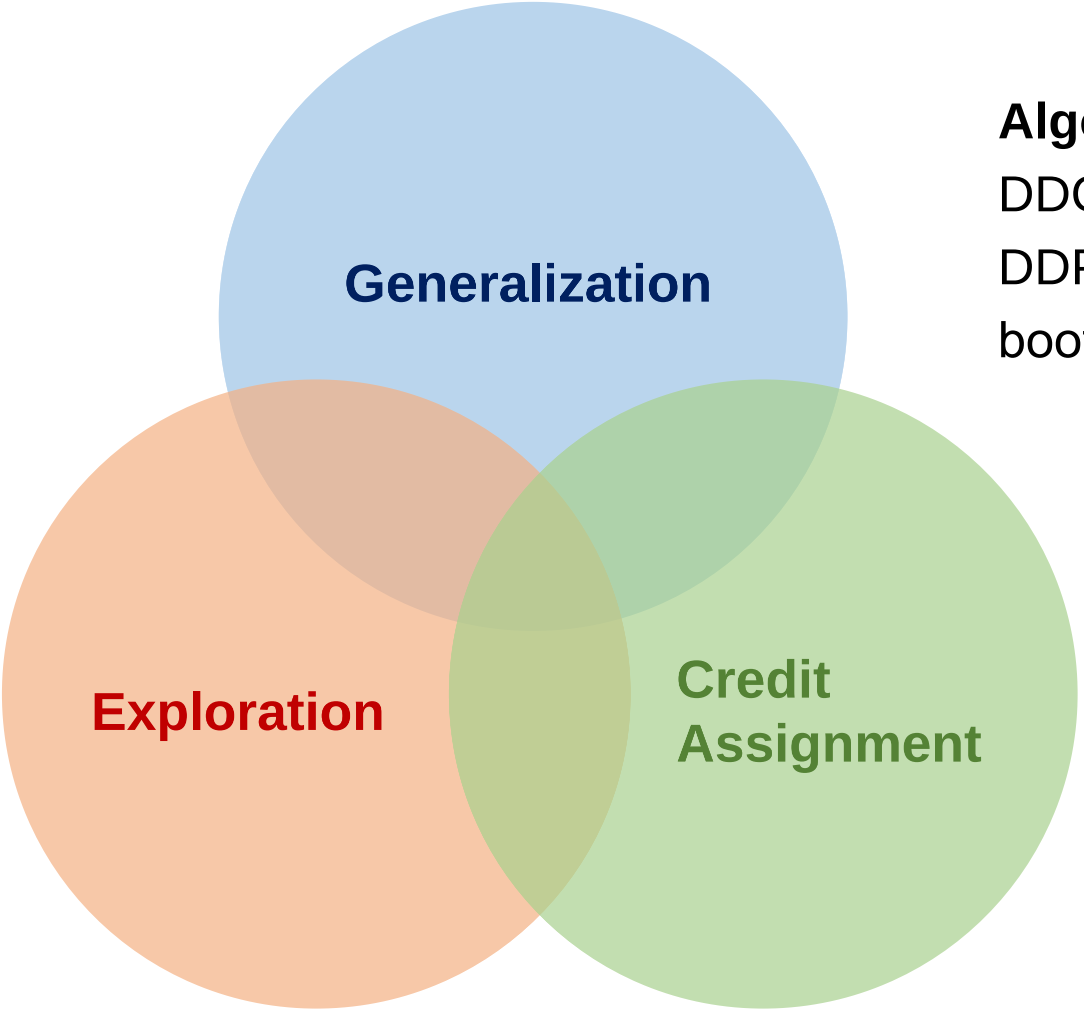
2. Approximate VI/PI (RL)

(Go)

Function approximation for functions of states, contexts, or actions

Action space:
- EG, BE, IGW
- UCB, TS
- Inverse weighting and baseline
- One-point unbiased gradient estimator

State space:
- UCB, TS
- Information-directed sampling
- Several bonus design



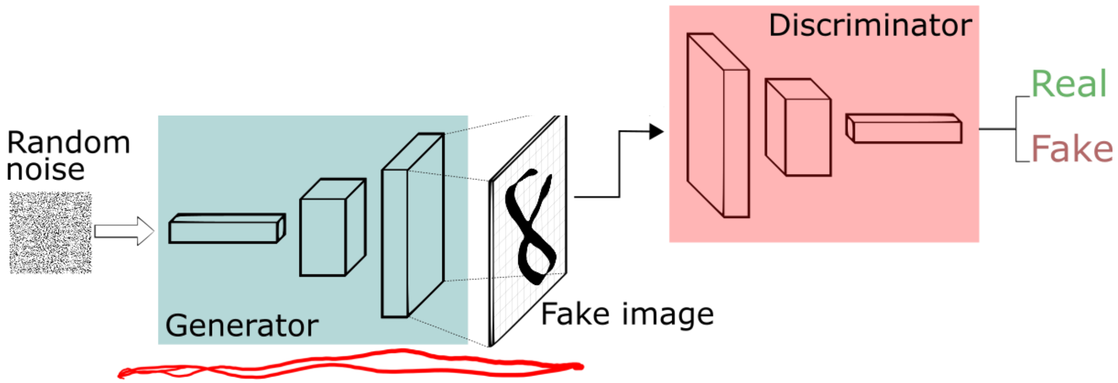
Algorithms: EXP3, DQN, DDQN, PPO, PG, A2C, DDPG, TD3, SAC, bootstrapped DQN….

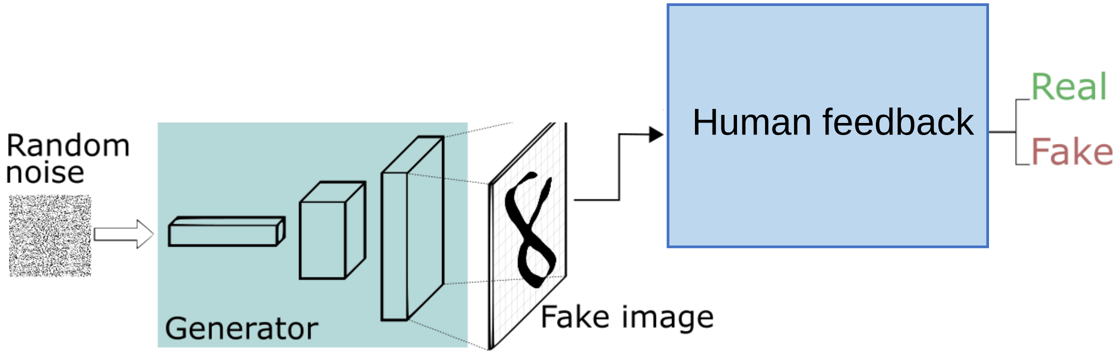
Dynamic programming
- (Approximate) value iteration, policy iteration
- Target network
- GAE

When and How to Use Reinforcement Learning?

- Analyze the problem
  - What **information** do we have in our problem? (full-information or bandit)
  - Full-information: $\underset{a}{\operatorname{argmax}}\, Q^\star(s,a)$, or $P(\cdot\,|s,a)$ & $R(s,a)$?
- Use RL only when needed
  - (Useful) supervision signal is bandit in nature (information consideration)
  - Problem is too big so we cannot perform full VI/PI (computational consideration)
- Integrate it with supervised learning or other machine learning techniques
  - There could be multiple sources of supervision signals: full-information and bandit
  - Some supervision signal could give a better initialization of VI/PI

Example 1: GAN





Action: fake image
Reward: discriminator output

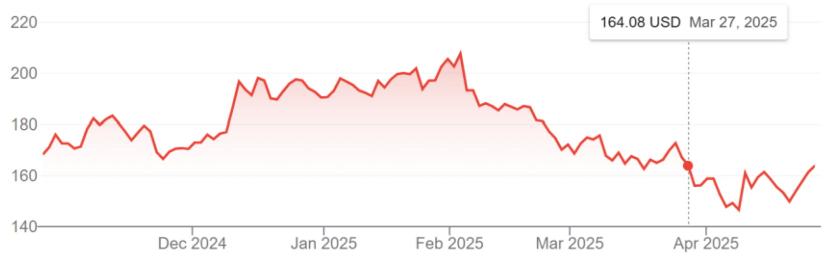
Do we have access to $\nabla_a r(a_t)$? **Yes**

Exploration is *not* needed.

Do we have access to $\nabla_a r(a_t)$? **No**

Exploration is needed. We may use value-based or policy-based approaches.

Example 2. Learning to Trade in a Stock Market



State: All available information
Action: {Sell, Hold, Buy}
Reward: Profit

$Q^*(s,a)$

What information do we have?

Full information (though noisy) about P(s'|s,a) and R(s,a): we know the consequence of taking a particular action even if we did not take that action.

Still making sense to use RL techniques, but there is potential to improve **data efficiency**:

Value-based method: in replay buffer we may add (s,a,r,s') for actions that we did not take before.

Policy-based method: we may be able to evaluate $Q^\pi(s,a)$ more accurately (less variance) by rolling out π from s, a multiple times.

Example 3. Go

State: current placement of the stones
Action: next placement
Reward: win/lose (revealed at the end)

Full information about P(s'|s,a) and R(s,a)

In theory, one can perform VI to find the maximin policy. But the large state space (3^{361}) disallow us to do so.

The full knowledge, again, equip the learner with advantage to repeatedly rolling out trajectories from a particular state (MCTS).

Also, it makes data from the "real world" very cheap.

Role of RL in “Learning to Act” Problems

Learning to Act

- **Reward Maximization Problems:** the ultimate goal is to maximize a golden reward
  - Go, Chess
  - Driving
  - Answering math questions, code generation

- **Imitation Problems:** the ultimate goal is to behave like human
  - Language model
  - Household robot
  - Image generation

Approaches to Learning to Act

- **Reward Maximization Problems:** the ultimate goal is to maximize a golden reward
  - Reinforcement Learning
  - Behavior Cloning (supervised learning with expert demonstration): used a lot in complex problems like driving, Go
  - A common practice: start with BC, and then perform RL
- **Imitation Problems:** the ultimate goal is to behave like human
  - Behavior Cloning
  - Distribution matching (GAN, diffusion models)
  - Inverse Reinforcement Learning:
    1. Infer an MDP such that human behavior appears approximately optimal on it.
    2. Perform *Reinforcement Learning* on the inferred MDP.

For language modeling, “DPO” and “RLHF” correspond to BC and IRL respectively.

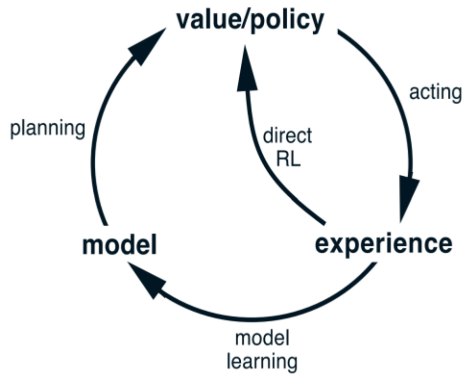
Topics We Did Not Cover

Topics We Did Not Cover

- Model-Based RL $\boxed{P, R}$.
- Offline RL
- Reward Design
- Robustness / Sim-to-Real

Model-Based Reinforcement Learning

$\mathcal{D}^{(1)} = \{(s, a, r, s')\}$ $\quad \mathcal{D}^{(2)}$ $\quad \cdots \quad \mathcal{D}^{(k-1)}$ $\quad \cdots$



(s, a, r, s')

$$\mathcal{D} = \mathcal{D}^{(1)} \cup \cdots \cup \mathcal{D}^{(k-1)}$$

$$\phi_k \leftarrow \underset{\theta}{\operatorname{argmin}} \ \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - r - \gamma \max_{a'} Q_{\phi_{k-1}}(s', a')\right)^2\right]$$

Model-free

$s \rightarrow$ **model** $\rightarrow \hat{P}(\cdot\,|s,a)$

$a \rightarrow$ **model** $\rightarrow \hat{R}(s,a)$

Trained with $\mathcal{D}$

Loop: Interact with environment → model training → planning

Planning: Find a good policy using the trained model

Model-based

Offline Reinforcement Learning

$(A, 1) \rightarrow 4/10$

$(A, 2) \rightarrow ?$

- The learner does not interact with the environment, but purely learn from existing data collected by other policies. After learning, the policy might be directly deployed.
- Difference with imitation learning: we do not assume the data is from expert. The goal of offline RL, like online RL, is to **maximize reward**.
- We do not need to design exploration strategy anymore. But we have to worry about the consequence of insufficient coverage of data.
  - Goal of exploration: to **resolve uncertainty**.
  - In offline RL, uncertainty may not be resolved completely.
  - Therefore, we usually **avoid uncertainty** when outputting policy in offline RL.

Offline Reinforcement Learning

Pessimistic Value Iteration (for offline RL to generate the final policy):

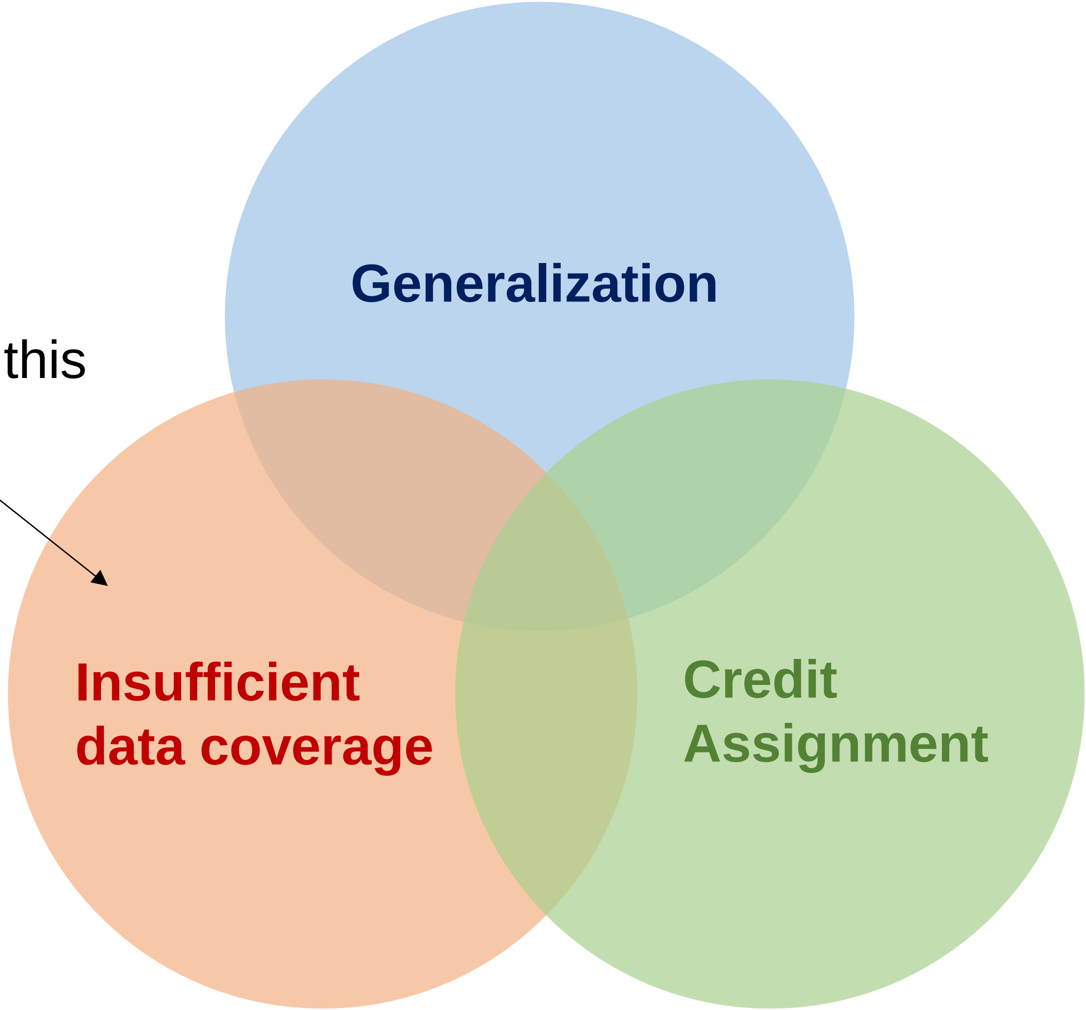
$$\tilde{Q}(s,a) \leftarrow \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}(s',a') - \text{ Uncertainty of } \hat{P}(\cdot\,|\mathrm{s},\mathrm{a}), \hat{R}(s,a)$$

cf. Optimistic Value Iteration (for online RL to generate the next policy):

$$\tilde{Q}(s,a) \leftarrow \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}(s',a') + \text{ Uncertainty of } \hat{P}(\cdot\,|\mathrm{s},\mathrm{a}), \hat{R}(s,a)$$

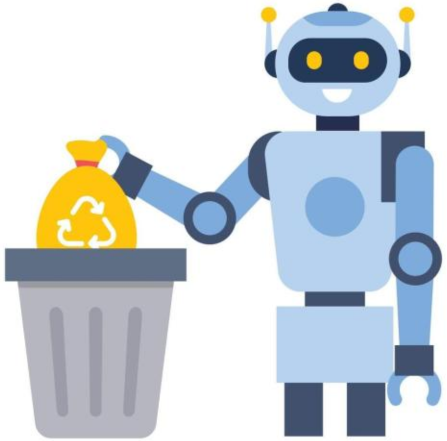
Offline Reinforcement Learning

Like exploration for online RL, this is due to **bandit information**



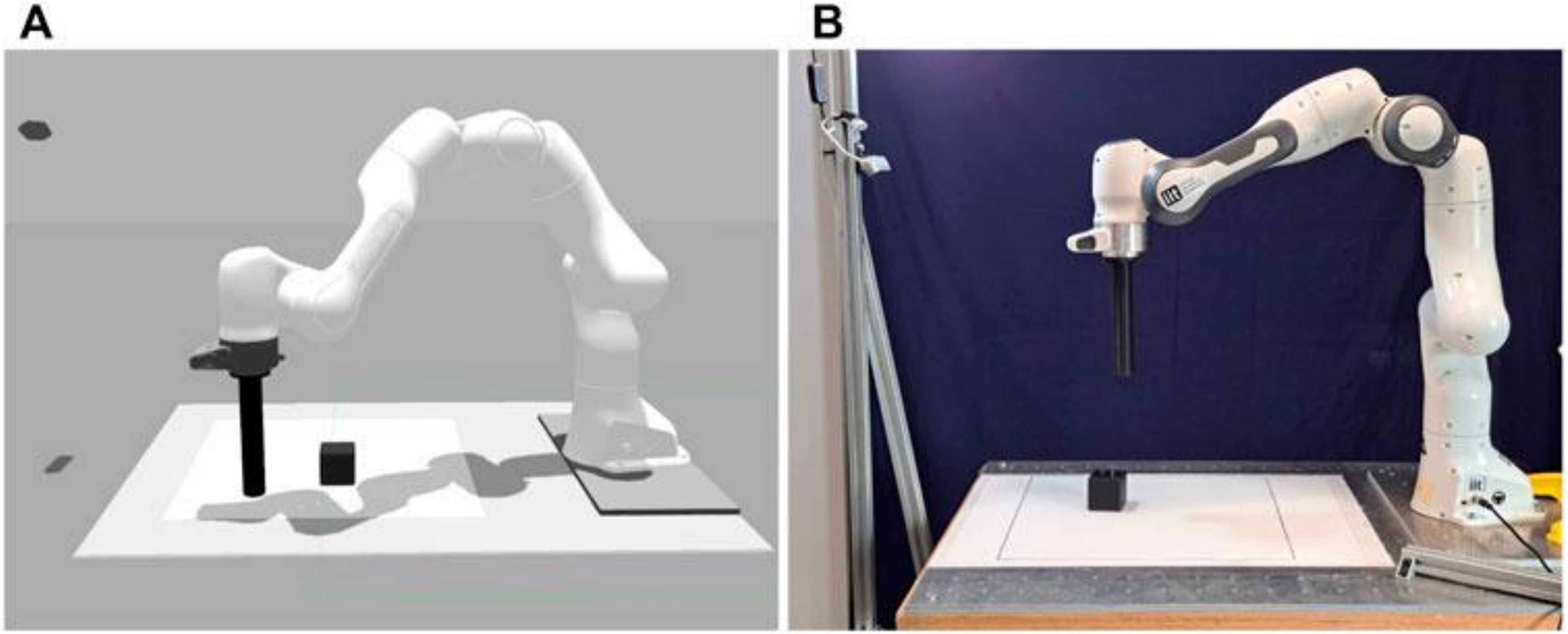
Reward Design

- Sparse reward: hard for typical RL algorithm to learn
- Reward hacking / misalignment



Robustness / Sim-to-Real

- How to minimize the performance degradation of a simulator-trained agent in a real environment



Simulated scenario   Real-world scenario

Final Reminders

Reminders

- Deadline of submitting final presentation: 11:59pm this Wednesday (April 30)
  - Ensure you have access to create video in Panopto (see my previous piazza announcement)
- From April 30 to May 8
  - Please engage in discussion about others / your groups' presentation on Panopto
  - This will give you extra points ranging from 0 to 5.
- Final report: May 5
  - Summarize what you have (for works you haven't done, mention them as future work)
- HW4: May 8
- Course evaluation