ETH zürich

Reinforcement Learning & Imitation Learning: an overview

Deep Learning Seminar
17.05.2022

Eugene Bykovets, D-INFK

Problem class

The problem is **to train** an intelligent **agent to achieve** a particular **goal** in a simulated environment or the real world. How can we do that?

Outline

- **Reinforcement learning**
  - Concept
  - Application domains
  - Notable problems
  - Reward misspecification
- **Imitation learning**
  - Behavioural cloning
  - Direct policy learning
  - Adversarial imitation learning
- **Inverse reinforcement learning**
  - Preference reward learning
  - Adversarial inverse reinforcement learning

Outline

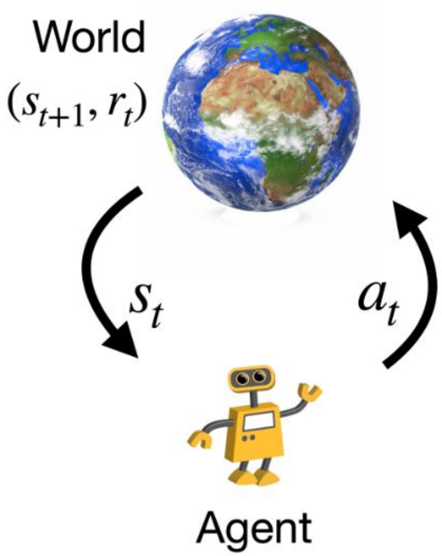
- **Reinforcement learning**
  - Concept
  - Application domains
  - Notable problems
  - Reward misspecification

- **Imitation learning**
  - Behavioural cloning
  - Direct policy learning
  - Adversarial imitation learning

- **Inverse reinforcement learning**
  - Preference reward learning
  - Adversarial inverse reinforcement learning

Reinforcement Learning: Concept

1. We model environment as a Markov Decision Process $\boldsymbol{MDP} = (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{\gamma}, \boldsymbol{P_0})$, where:
   - $\boldsymbol{S}$ is state space
   - $\boldsymbol{A}$ is action space
   - $\boldsymbol{P}(\boldsymbol{s'}|\boldsymbol{s}, \boldsymbol{a})$ is transition probability
   - $\boldsymbol{R}(\boldsymbol{s'}, \boldsymbol{a}, \boldsymbol{s})$ is reward signal
   - $\boldsymbol{\gamma}$ is discount factor
   - $\boldsymbol{P_0}$ is initial state distribution

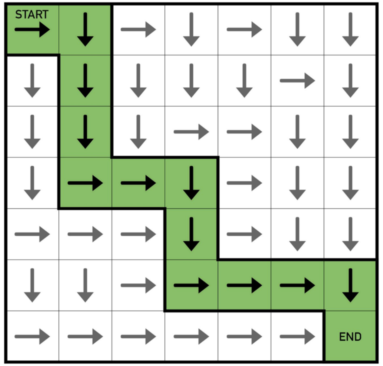


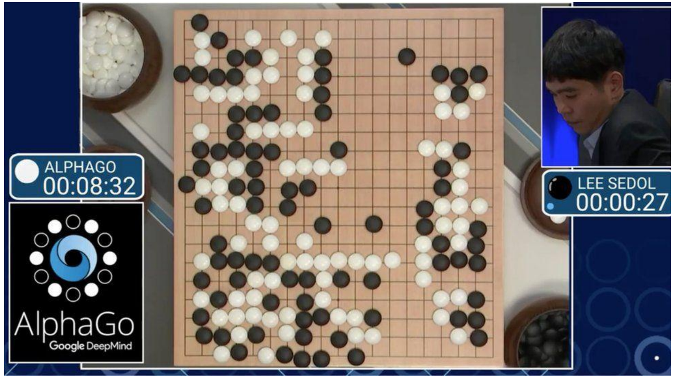
- Agent uses reward signal $\boldsymbol{R}(\boldsymbol{s'}, \boldsymbol{a}, \boldsymbol{s})$ from the environment as a guidance to achieve goal

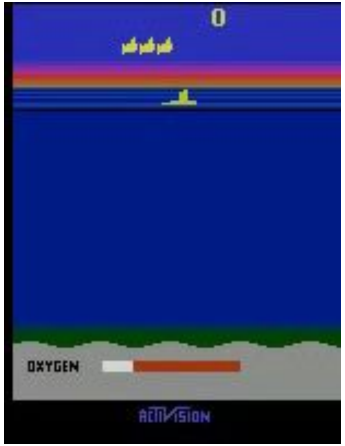
- Goal is to train the agent behavior (aka policy) $\pi(a|s)$ which maximizes expected discounted reward:

$$\boldsymbol{E}[\boldsymbol{\Sigma}_{t=0}^{\infty} \boldsymbol{\gamma}^t \boldsymbol{R}(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{s_{t+1}})]$$

Reinforcement Learning: Simulated domains

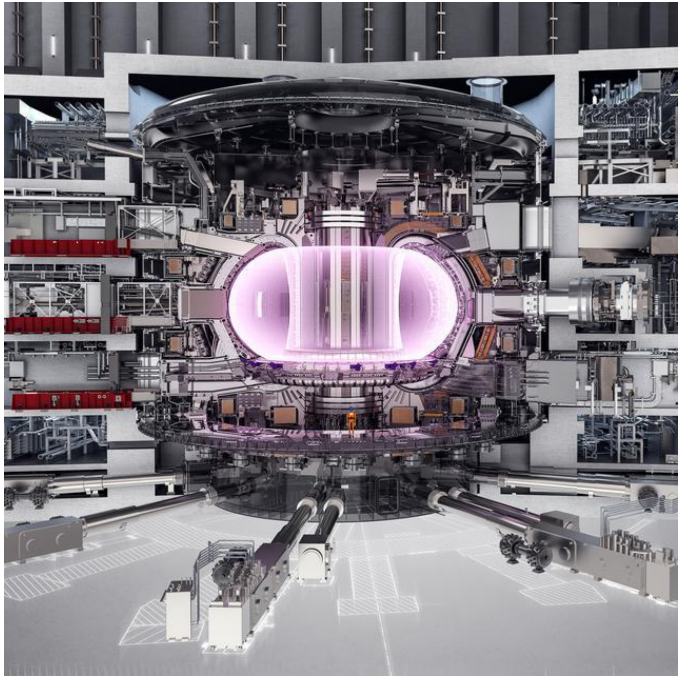






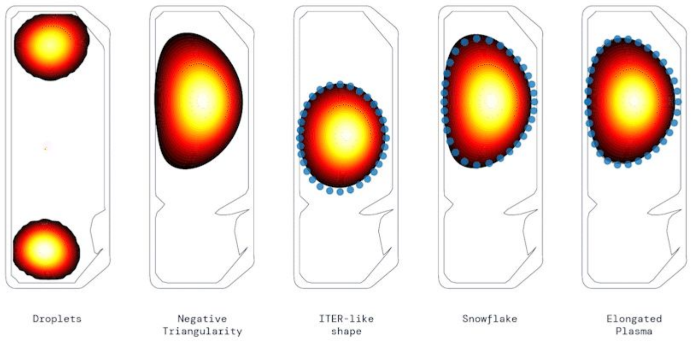
Reinforcement Learning: Real-world domains (plasma control)

- Researchers have long sought a source of clean, limitless energy
- One **contender is nuclear fusion** which is process of smashing and fusing hydrogen releases huge amounts of energy
- One way scientists have recreated these extreme conditions is by using a **tokamak**, a doughnut-shaped vacuum surrounded by magnetic coils, that **is used to contain a plasma of hydrogen in extremely high temperature**



Reinforcement Learning: Real-world domains (plasma control)

- Plasma in these machines are inherently unstable, as a result **sustaining the process is a complex challenge**
- **Control system** needs to coordinate the tokamak's many magnetic coils and **adjust the voltage on them thousands of times per secon**d to ensure the plasma never touches the walls of the vessel, which would result in heat loss and possibly damage
- Swiss Plasma Center at EPFL and DeepMind managed to **train controller with reinforcement learning** in simulated environment and apply in real tokamak



Reinforcement Learning: Problems

- Markovian nature of data is a challenge for optimizers
  - Most of the results are obtained given i.i.d assumption (SGD gradient are biased [1]) all the SGD-like optimizers still affected
- Training domain shift
  - RL can be approximated supervised as i.i.d problem with continuous training domain shift by using experience replay buffer
- Sample efficiency
  - In the worst case, we see all states only once. There is a remedy: the usage large experience reply buffer, but it is not a panacea
- Safety problem
  - Training in real-world can be damaging for agents and the environment
- Reward specification
  - Could we really design a reward that corresponds to our intentions?

Reinforcement Learning: reward misspecification

- CoastRunners game
  - **The goal** is to **gain** as much **score** as possible **by collecting items and win the race** (as human understands)

- Reward is:
  - **Sparse:** not every step towards the finish is encouraged
  - **Misspecified:** Item gathering has superior encouragement

- **The result** is **misbehavior!**

Reinforcement Learning: reward misspecification

Maybe we should try to **train** an **agent without a pre-designed reward signal?**

Outline

- **Reinforcement learning**
  - Concept
  - Algorithms taxonomy
  - Application domains
  - Notable problems
  - Reward misspecification

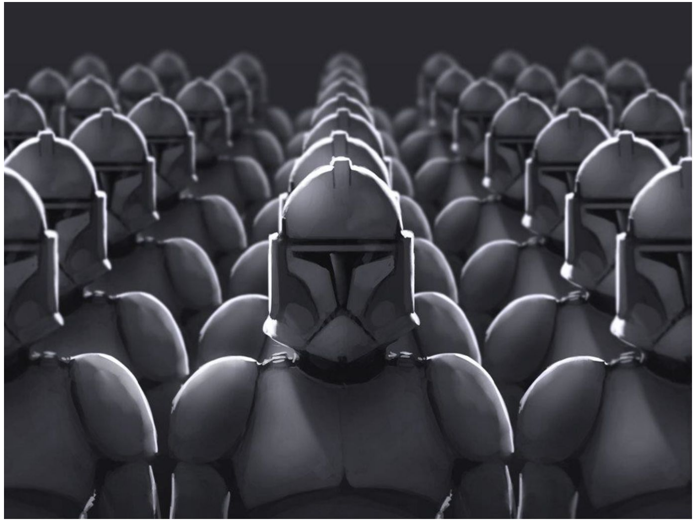
- **Imitation learning**
  - Behavioral cloning
  - Direct policy learning
  - Adversarial imitation learning

- **Inverse reinforcement learning**
  - Preference reward learning
  - Adversarial inverse reinforcement learning

- **Offline reinforcement learning**

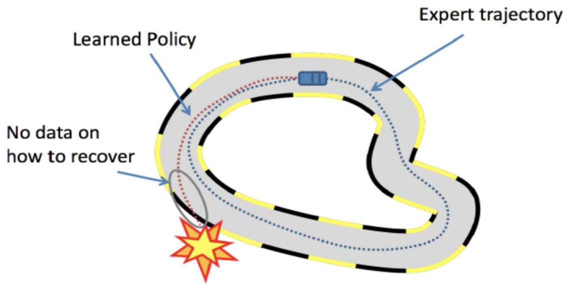
Behavioral cloning: Concept

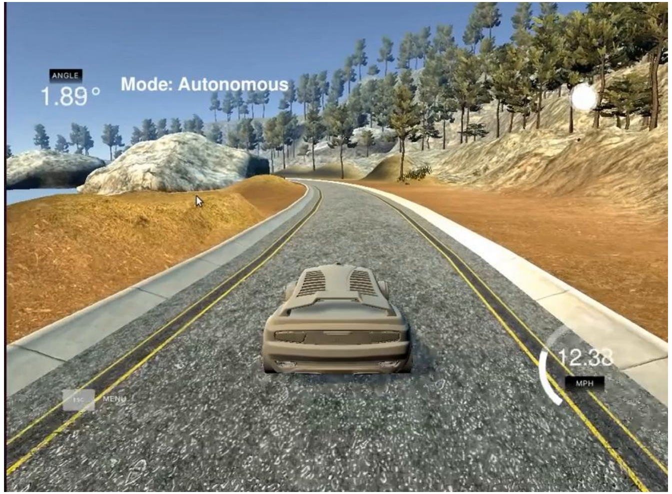
1. Collect demonstrations τ^* trajectories from expert
2. Treat the expert demonstrations as i.i.d. state-action pairs: $(s_0^*, a_0^*), (s_1^*, a_1^*), \ldots$
3. Learn $\pi(a|s)$ policy using supervised learning by minimizing the loss function $L(a^*, \pi_\theta(s))$



Behavioral cloning: Problems

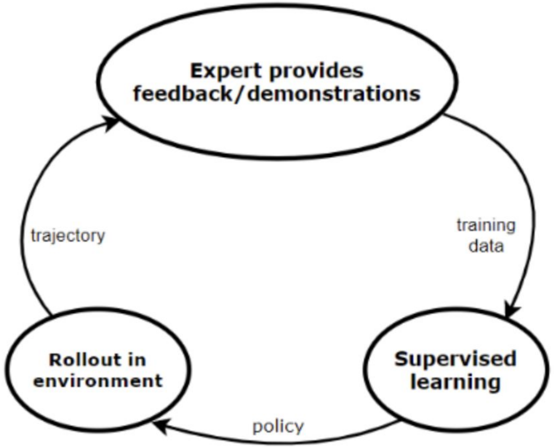
- **Markovian** data means the next state depends on the current one
- **Misbehavior** in the **current state** leads to the **accumulation of the error** in all the **next steps**
- **Misbehavior** is very **likely** in states which **different from expert**
- **We need an oracle!**





Direct policy learning: Concept

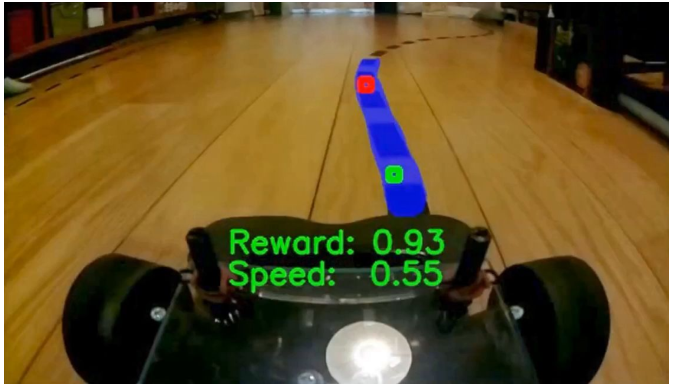
- **Improved concept** of behavioral cloning
- Assume the **presence** of an interactive expert-level demonstrator **(oracle)**
- The main idea is to get **more suboptimal trajectories** to **improve behavioral robustness** in states that are far from that contained in expert data



Direct policy learning: Data/Policy Aggregation

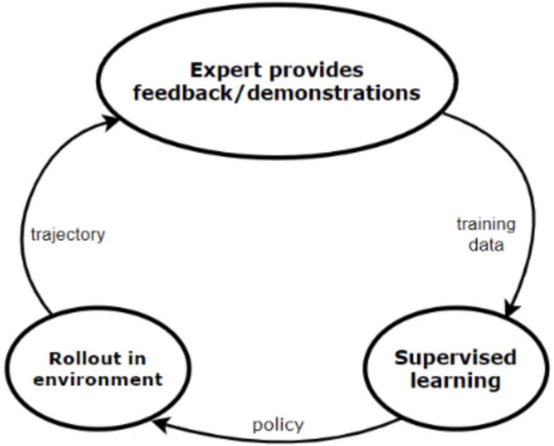
1.

- Initial predictor π_0
- For $m = 1$:
  - Collect trajectories τ^* by rolling out π_{m-1}
  - Estimate state distribution P_m using $s \in \tau^*$
  - Collect interactive feedback $\{\pi^*(s) \mid s \in \tau^*\}$
  - Data Aggregation (e.g. Dagger)
    - Train π_m on $P_1 \cup \cdots \cup P_m$
  - (Alternative is policy aggregation, e.g. SEARN)
    - Train π'_m on P_m
    - $\pi_m = \beta\pi'_m + (1 - \beta)\,\pi_{m-1}$

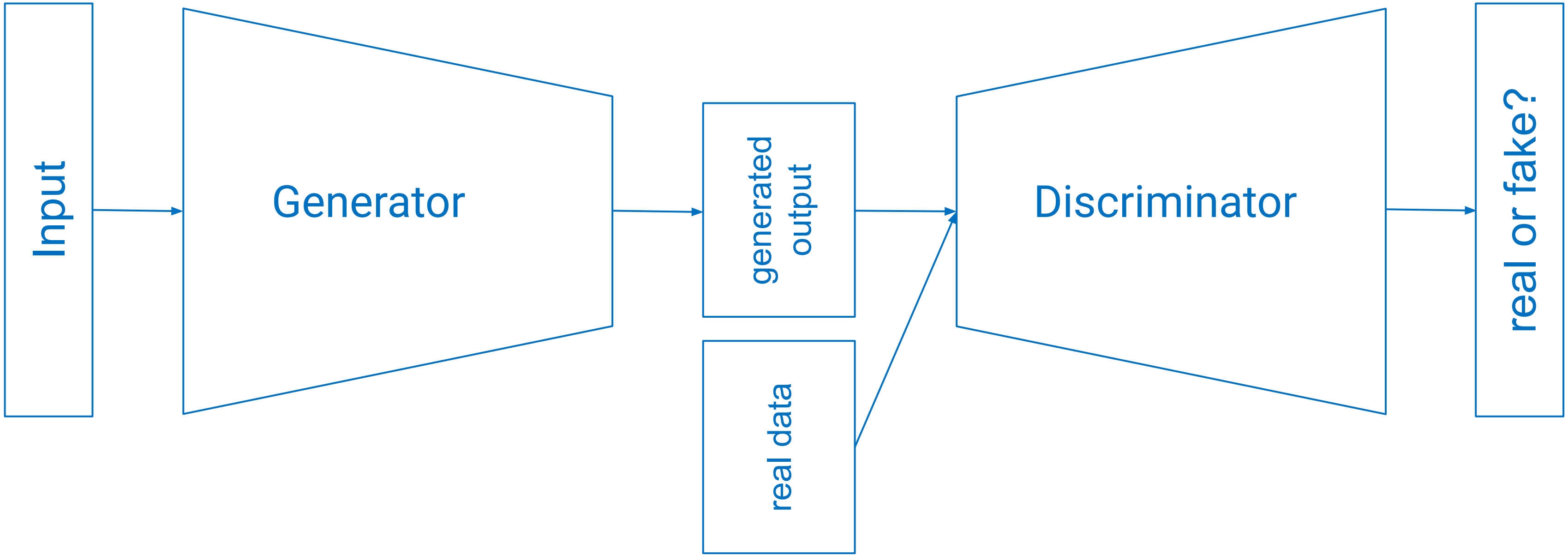


Direct policy learning: Problem

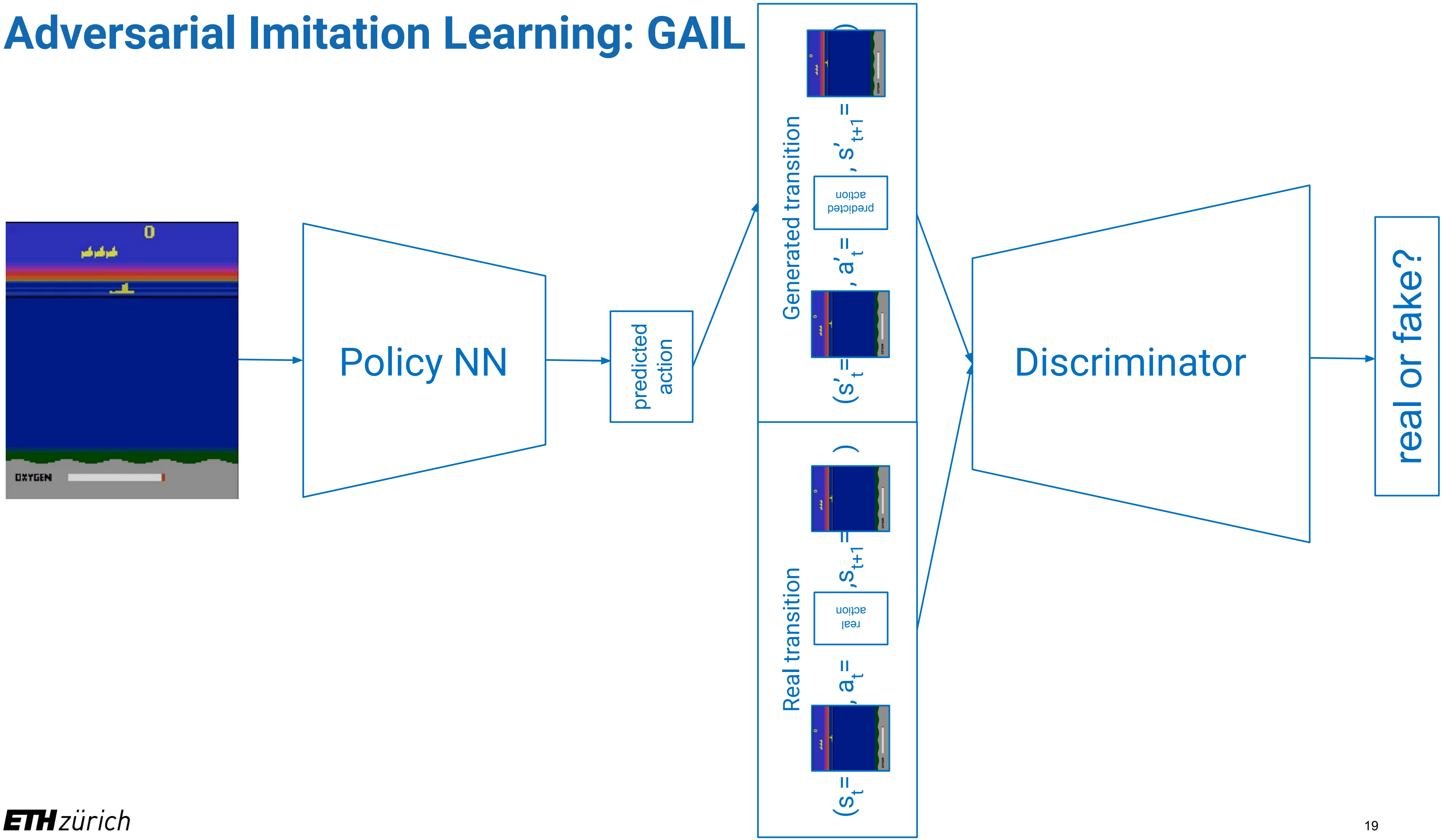
- Improved concept of behavioral cloning
- **Assume the presence of an interactive expert-level demonstrator (oracle) expensive!**
- The main idea is to get more suboptimal trajectories to improve behavioral robustness in states that are far from that contained in expert data



Adversarial Imitation Learning: GAN

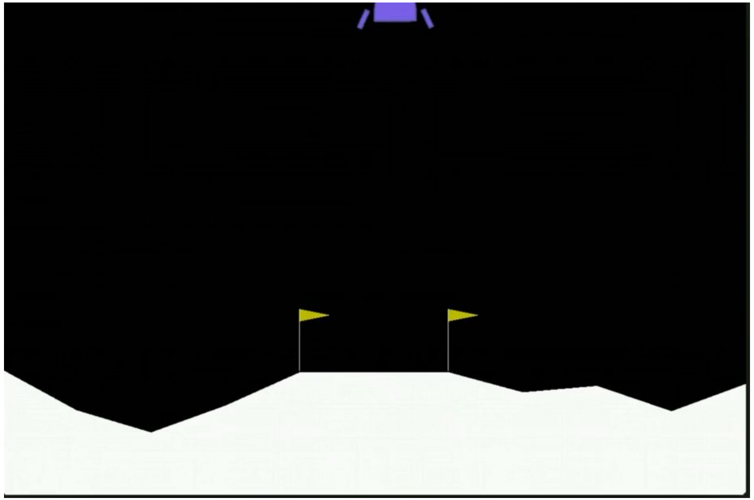


Adversarial Imitation Learning: GAIL



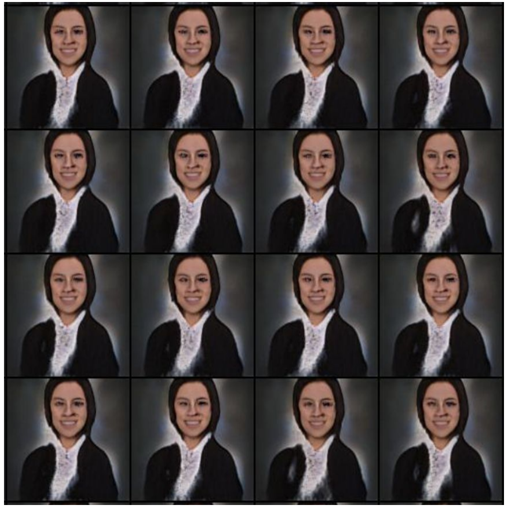
Adversarial Imitation Learning: Concept

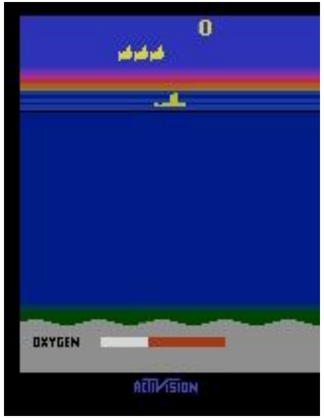
- Collect demonstrations τ^* trajectories from experts;
- Build GAN like architecture where:
    - The generator is now off-the-shelf reinforcement learning (e.g., PPO) algorithm that tries to generate meaningful trajectories
    - Discriminator tries to differentiate real expert trajectories (collected at the beginning) from generated ones
    - In the adversarial two-player game, we try to achieve expert-level policy



Adversarial Imitation Learning: Problems

- Inherited GAN problems:
  - Difficult to optimize
  - Difficult to fine-tune
  - Mode collapse





Inverse Reinforcement Learning

What if we still want to **make underlying intentions** a little bit more **understandable**?

Outline

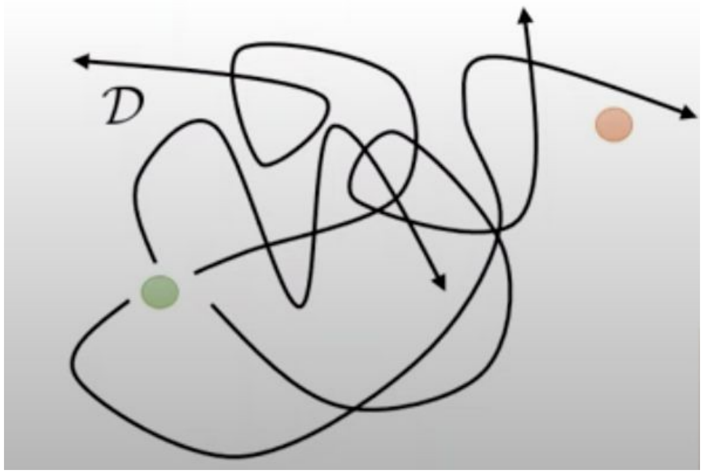
- **Reinforcement learning**
  - Concept
  - Algorithms taxonomy
  - Application domains
  - Notable problems
  - Reward misspecification

- **Imitation learning**
  - Behavioral cloning
  - Direct policy learning
  - Adversarial imitation learning

- **Inverse reinforcement learning**
  - Preference reward learning
  - Adversarial inverse reinforcement learning

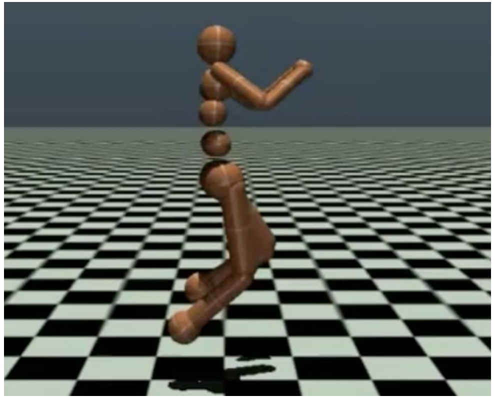
Preference Reward Learning: concept

- Collect trajectories (expert-level included):
  - $\tau_1, \tau_2, \dots, \tau_n$
- Ask expert to give global score (or rank the trajectories with full order):
  - $\tau_4 \prec \tau_{104} \prec \cdots \tau_2$
- Train you reward function $\widehat{r_\theta}(s)$ in supervised manner with ranking loss-funciton:
  - $\mathcal{L}(\theta) = -\sum_{\tau_i \prec \tau_j} log \frac{\exp \sum_{s \in \tau_j} \widehat{r_\theta}(s)}{\exp \sum_{s \in \tau_i} \widehat{r_\theta}(s) + \exp \sum_{s \in \tau_j} \widehat{r_\theta}(s)}$
  - Do not care about reward for individual states,
  - Want that the predicated trajectories rewards align with ground truth rank

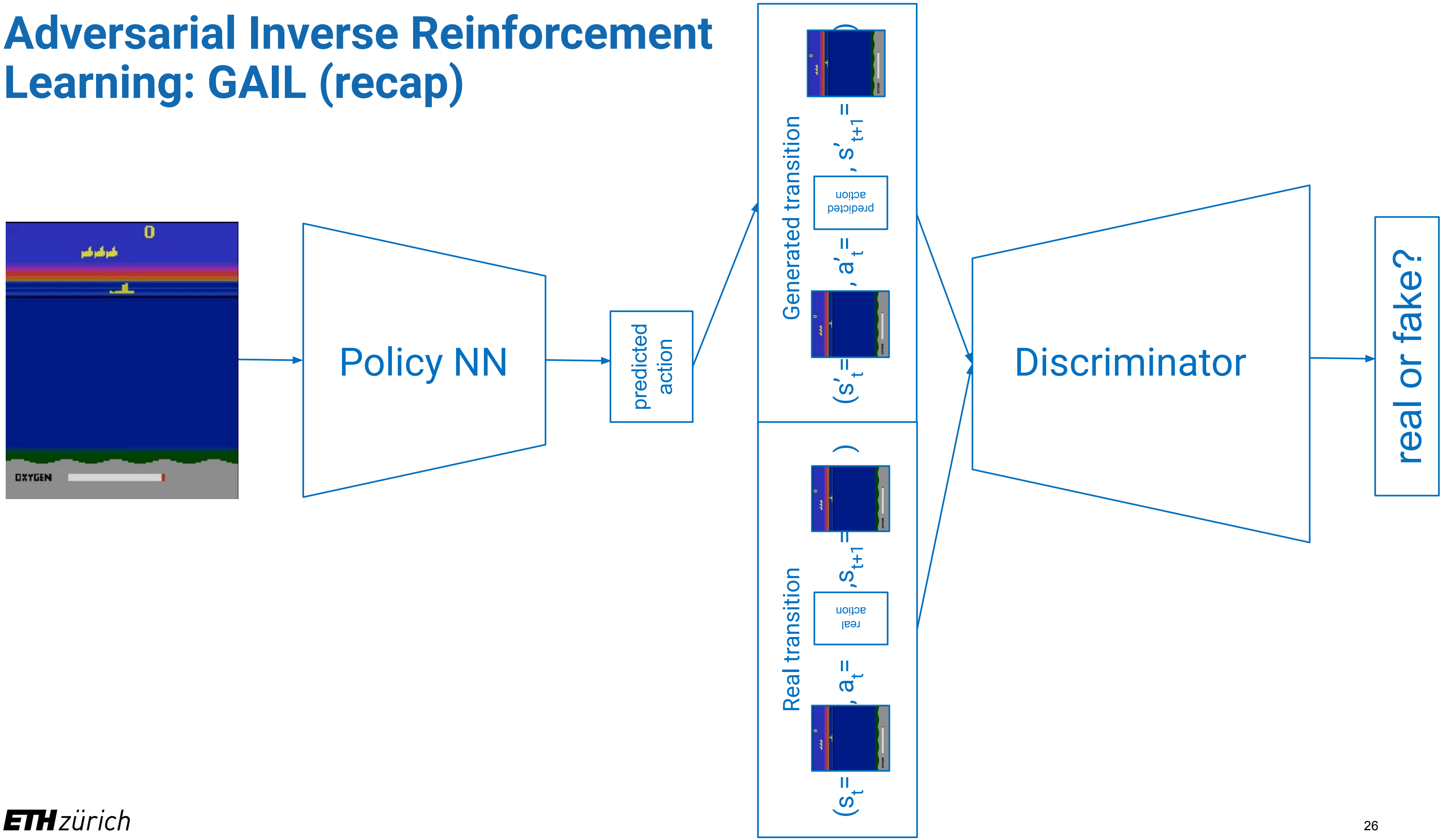


Preference Reward Learning: problems

- Ill-poseness nature of the problem (of IRL)
  - Infinitely many “optimal” reward functions w.r.t to a finite amount of expert trajectories
- Imprecise, works well only for “survival” environments
  - We care about staying alive In the environment longer and do not care about achieving the precise goals, as a result, we are fine with imprecise function

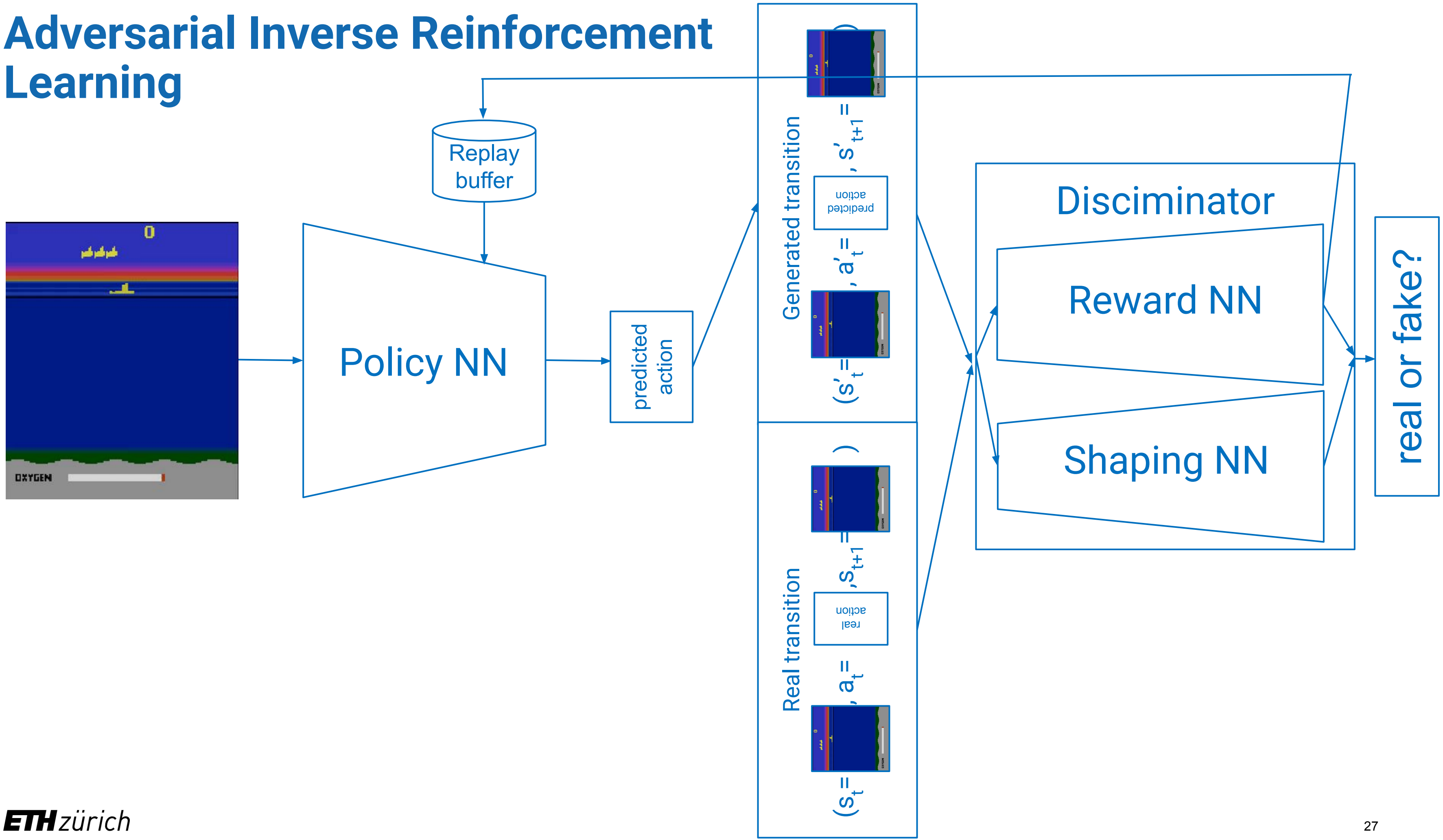


Adversarial Inverse Reinforcement Learning: GAIL (recap)



ETH zürich

Adversarial Inverse Reinforcement Learning



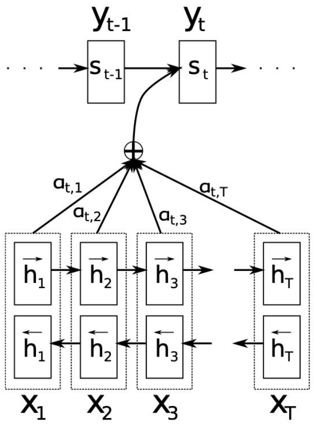
Adversarial Inverse Reinforcement Learning: problems

- Inherited GAN problems:
  - Difficult to optimize
  - Difficult to fine-tune
  - Mode collapse
  - Reward is not the last version of the network, it is curriculum set

Summary

Field	Goal	Example	Advantages	Problems
Reinforcement Learning	Optimize expected discounted reward	PPO, SAC, TRPO etc.	Can be straightforward for simple problems	• Sampe efficency • Training domain shift • Sim2Real • Reward misspecification
Imitation Learning	Imitatie expert behavior	Beahvior cloning	Simple to use	Is not robust
		Direct Policy Learning	Good perfomance	Expensive expert-level oracle needed
		GAIL	Good perfomance	GAN-inhereted problems
Inverse Reinforcement Learning	Learn reward function	Preference Learning	Simple to use	Work only for “survival: problems
		AIRL	Good perfomance	GAN-inhereted problems, need to build curricuum set

Thank you for your attention!



Questions?

Resources

- Fig. 2. Gridworld problem https://towardsdatascience.com/training-an-agent-to-beat-grid-world-fac8a48109a8
- Fig. 3. DeepMind AlphaGo. https://ichef.bbci.co.uk/news/976/cpsprodpb/11B23/production/_88738427_pic1go.jpg
- Fig.4. Reinforcement learning: Distributional Soft Actor-Critic (DSAC) in Gym Mujoco
- Fig. 7. Tokomak machine https://www.rts.ch/rts-
- online/medias/images/2021/thumbnail/fk3wxv-25152632.image?w=640&h=640
- Fig. 9. OpenAI. Reward misspecification https://openai.com/blog/faulty-reward-functions/
- Fig. 11. https://medium.com/startup-grind/even-smart-vcs-invest-in-cross-industry-clones-6a3c63f830e4
- Fig 12. https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c
- Fig 14. Direct Policy Learning https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c