Imitation Learning

Introduction

- Imitation Learning
    - Also known as learning by demonstration, apprenticeship learning
- An expert demonstrates how to solve the task
    - Machine can also interact with the environment, but cannot explicitly obtain reward.
    - It is hard to define reward in some tasks.
    - Hand-crafted rewards can lead to uncontrolled behavior
- Two approaches:
    - Behavior Cloning
    - Inverse Reinforcement Learning (inverse optimal control)

Behavior Cloning

Behavior Cloning

Yes, this is supervised learning.

- Self-driving cars as example

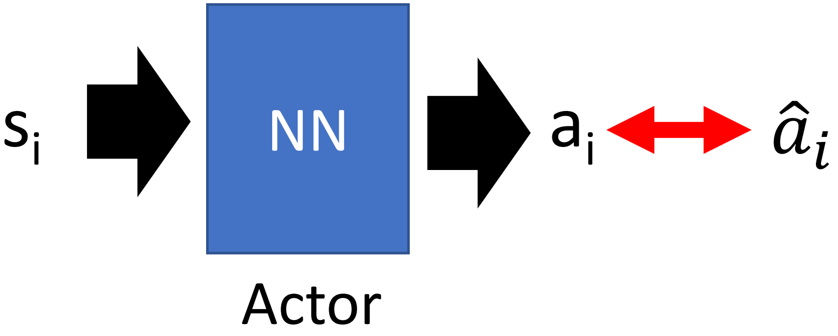
observation

Expert (Human driver): 向前

Machine: 向前

Training data:

$(s_1, \hat{a}_1)$
$(s_2, \hat{a}_2)$
$(s_3, \hat{a}_3)$
……



s_i NN a_i $\hat{a}_i$

Actor

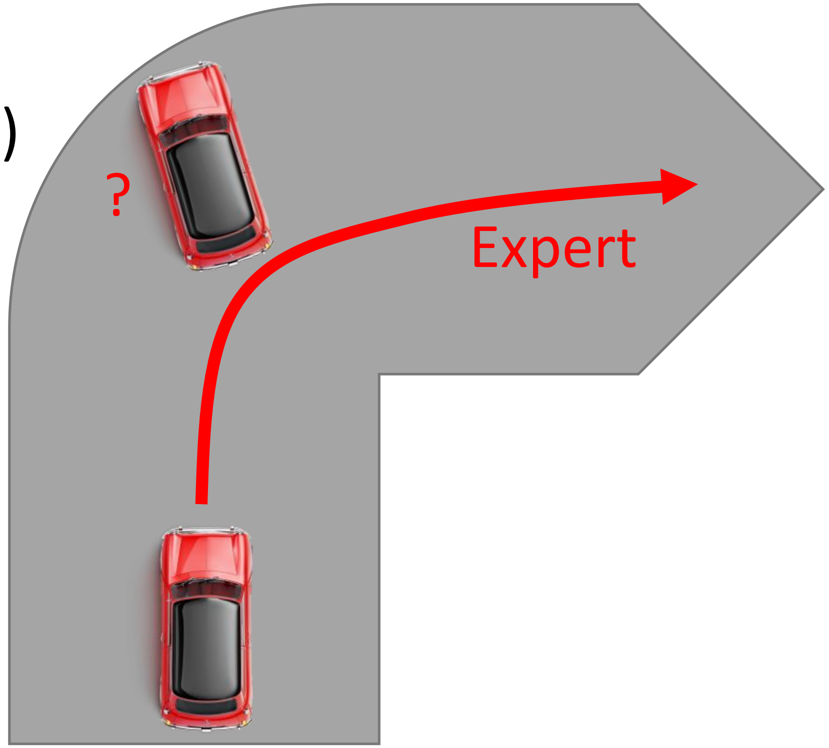
Behavior Cloning

- Problem

Expert only samples limited observation (states)

Let the expert in the states seem by machine

Dataset Aggregation



Behavior Cloning

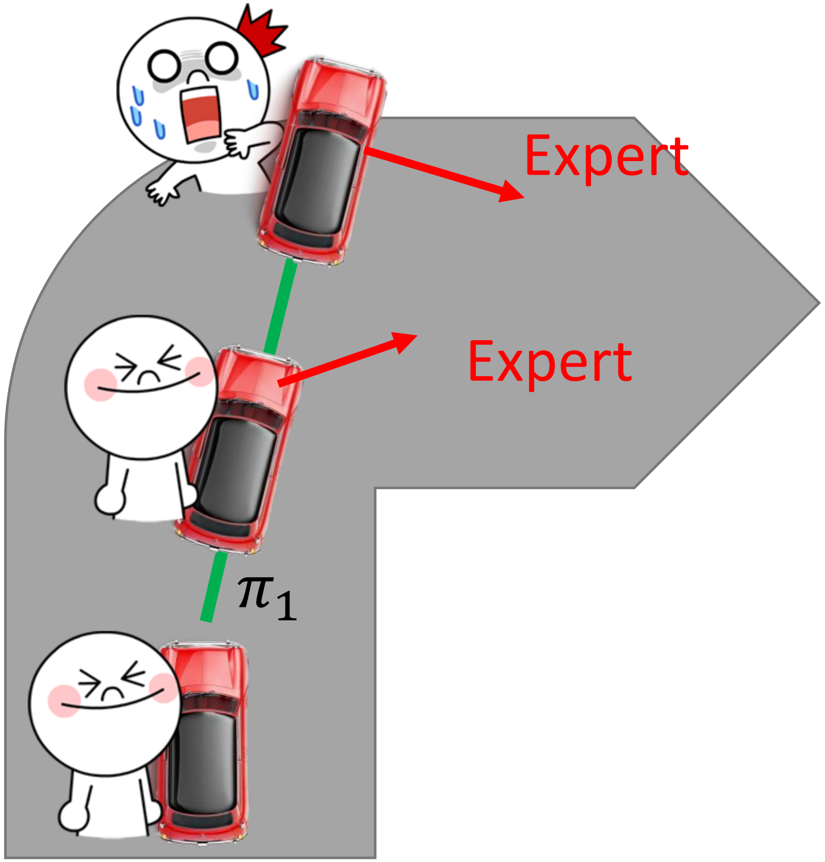
- Dataset Aggregation

Get actor π_1 by behavior cloning

Using π_1 to interact with the environment

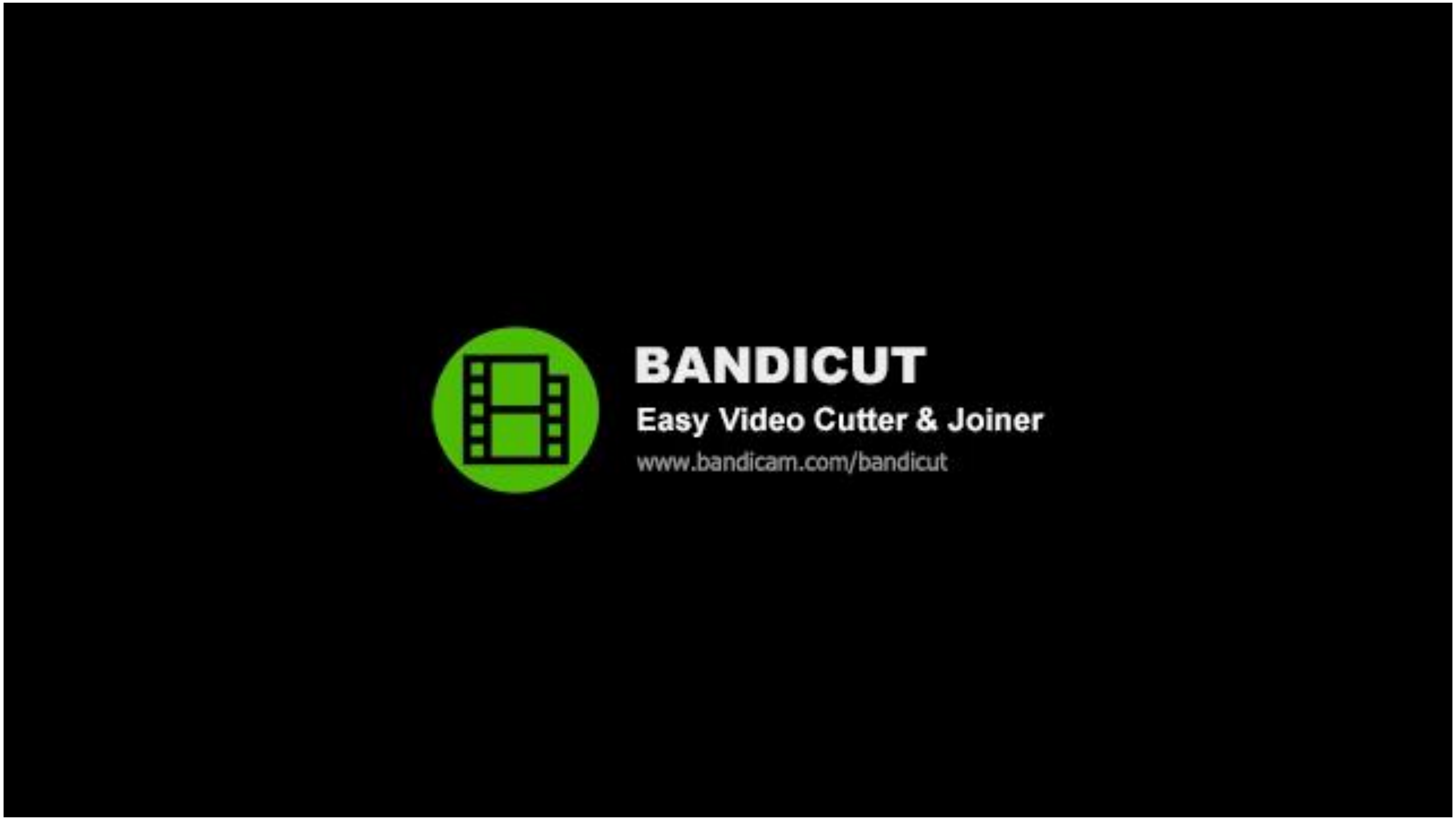
Ask the expert to label the observation of π_1

Using new data to train π_2



Behavior Cloning

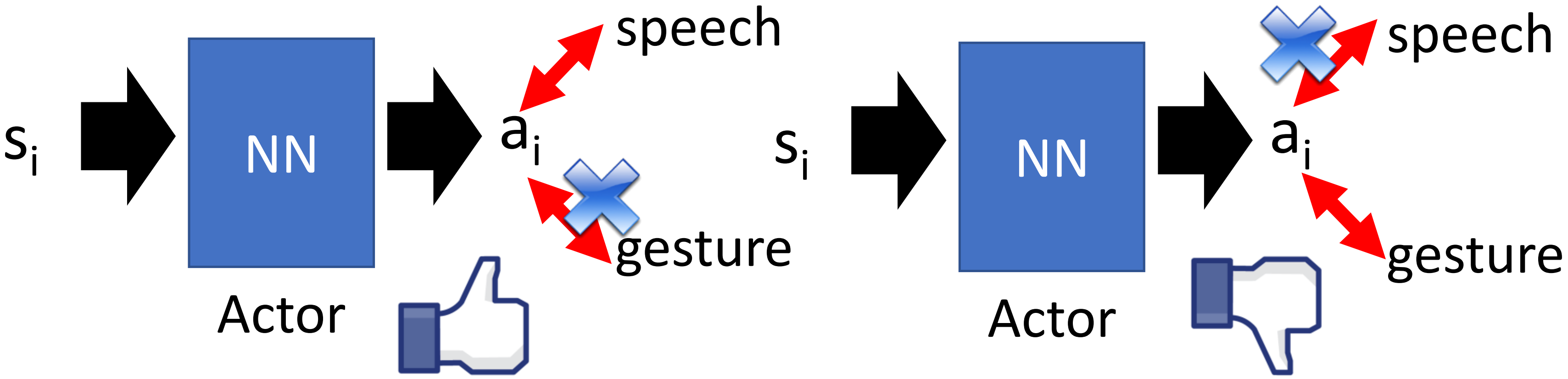
The agent will copy every behavior, even irrelevant actions.



https://www.youtube.com/watch?v=j2FSB3bseek

Behavior Cloning

- Major problem: if machine has limited capacity, it may choose the wrong behavior to copy.



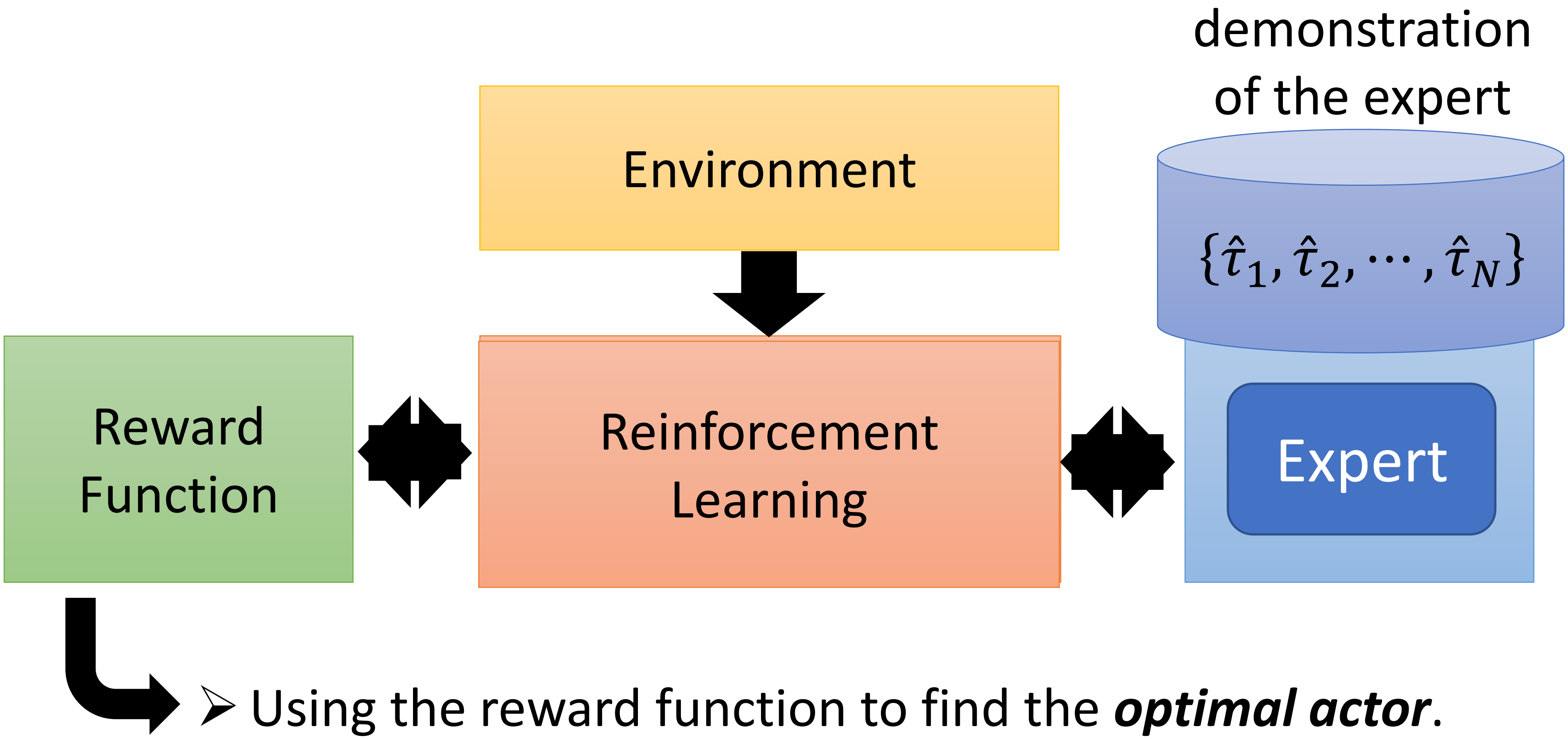
- Some behavior must copy, but some can be ignored.
  - Supervised learning takes all errors equally

Mismatch

- In supervised learning, we expect training and testing data have the same distribution.
- In behavior cloning:
  - Training: $(s, a) \sim \hat{\pi}$ (expert)
    - ***Action a taken by actor influences the distribution of s***
  - Testing: $(s', a') \sim \pi^*$ (actor cloning expert)
    - If $\hat{\pi} = \pi^*$, (s, a) and (s', a') from the same distribution
    - If $\hat{\pi}$ and π^* have difference, the distribution of s and s′ can be very different.

Inverse Reinforcement Learning (IRL)

Inverse Reinforcement Learning

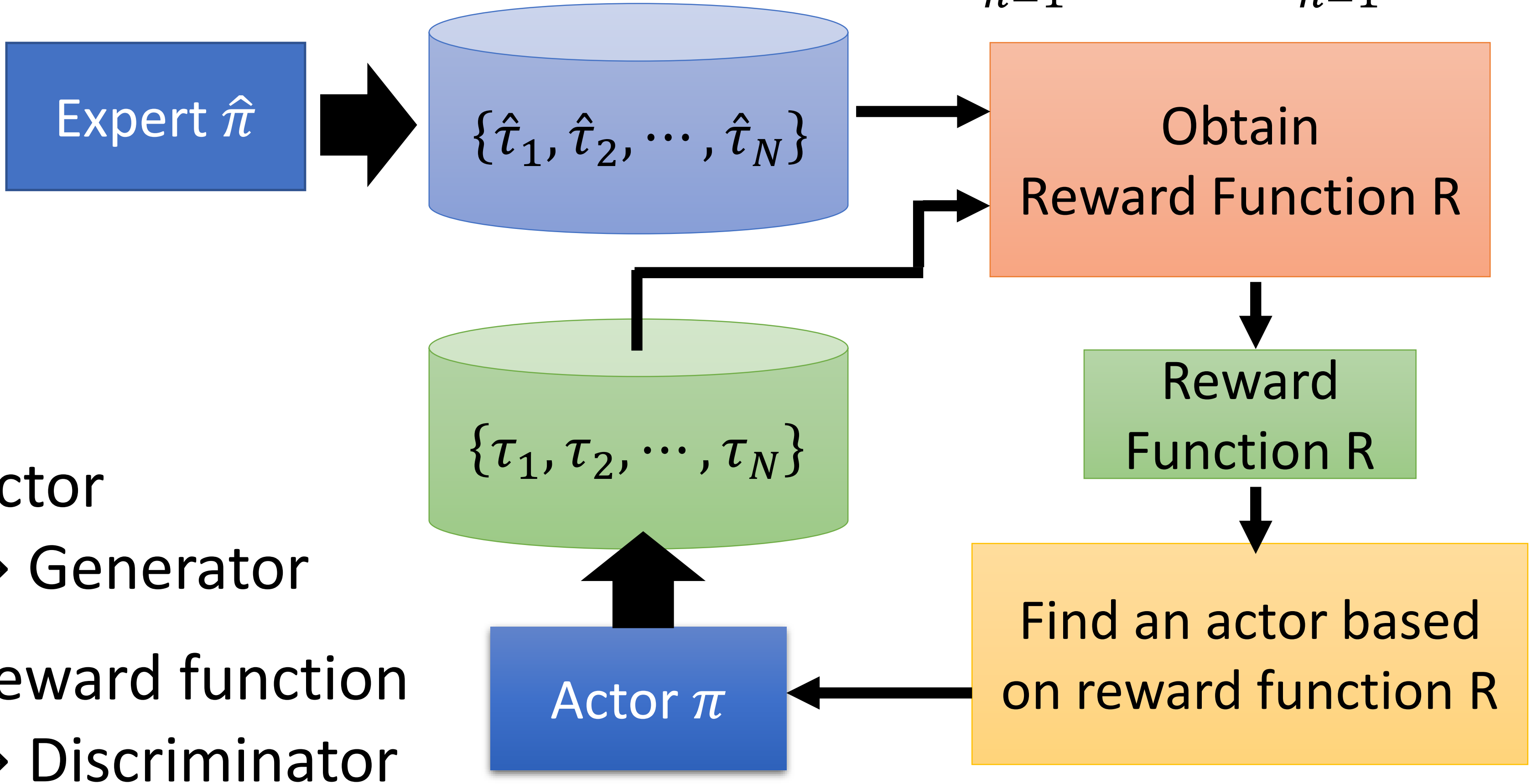

demonstration of the expert
Environment
$\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_N\}$
Reward Function
Reinforcement Learning
Expert

- Using the reward function to find the ***optimal actor***.
- Modeling reward can be easier. Simple reward function can lead to complex policy.

Framework of IRL

The expert is always the best.

$$\sum_{n=1}^{N} R(\hat{\tau}_n) > \sum_{n=1}^{N} R(\tau)$$

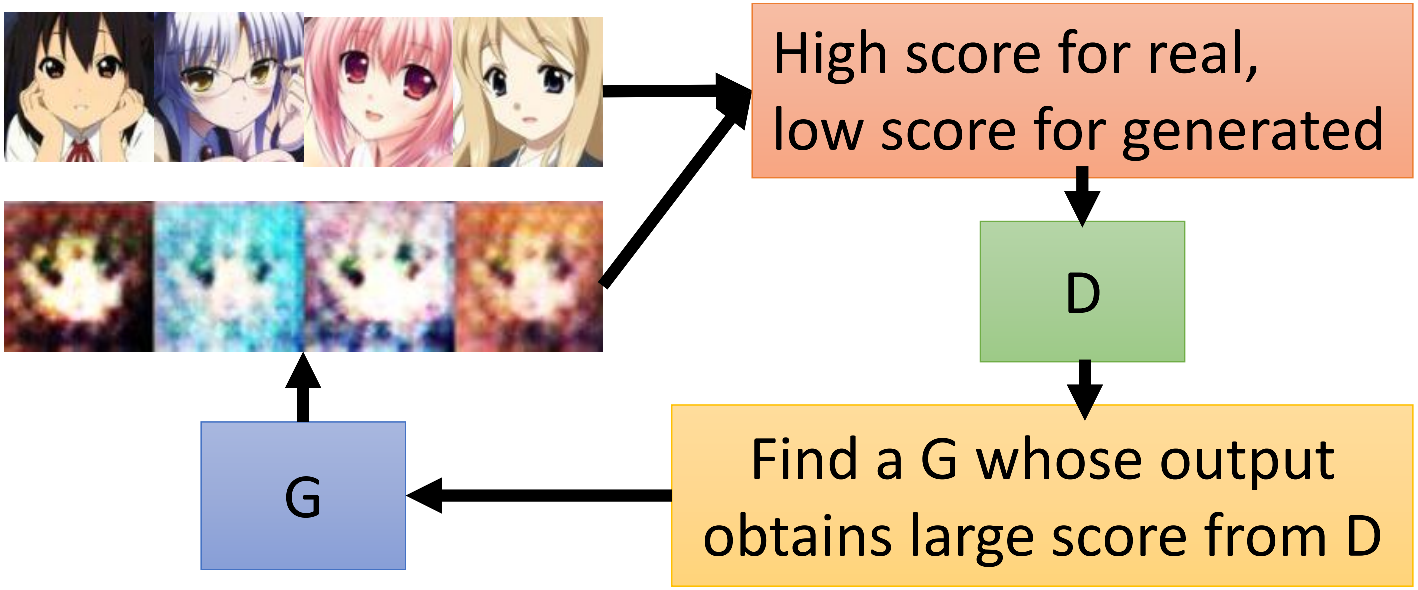

Expert $\hat{\pi}$
$\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_N\}$
Obtain Reward Function R
$\{\tau_1, \tau_2, \cdots, \tau_N\}$
Reward Function R
Find an actor based on reward function R
Actor π

By Reinforcement learning

Actor
→ Generator

Reward function
→ Discriminator

GAN



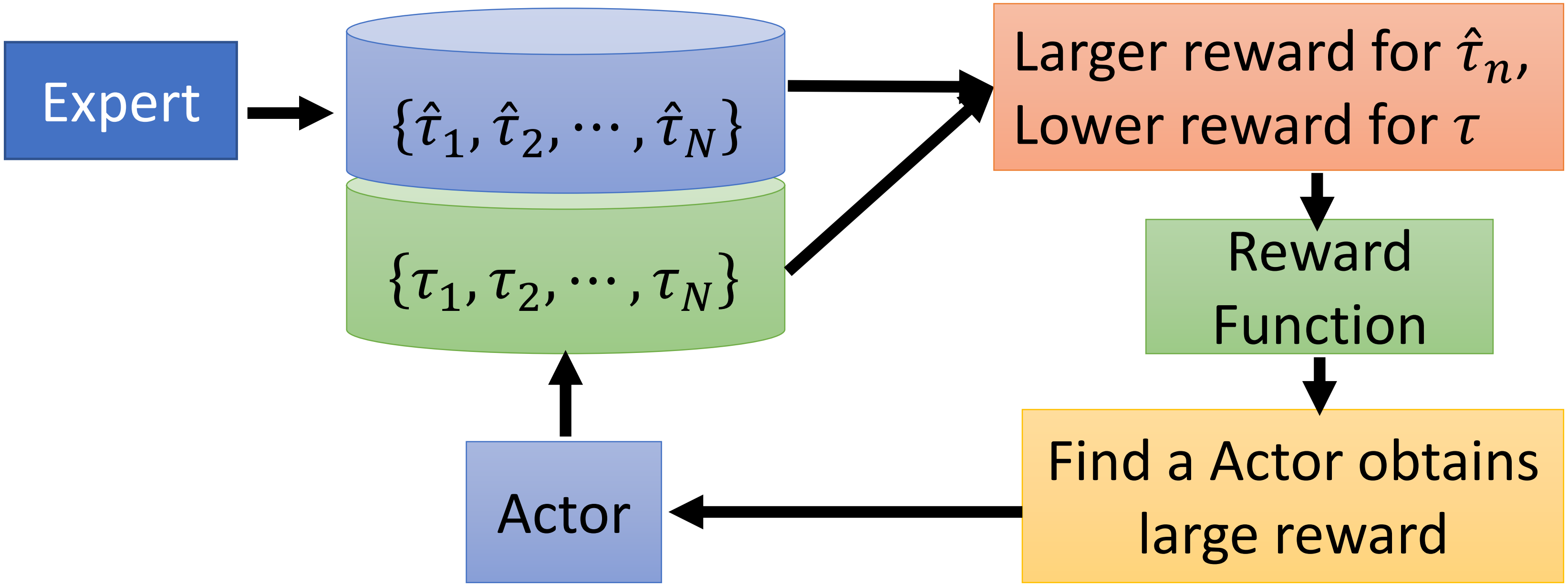
High score for real, low score for generated

D

Find a G whose output obtains large score from D

G

IRL



Expert

$\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_N\}$

$\{\tau_1, \tau_2, \cdots, \tau_N\}$

Larger reward for $\hat{\tau}_n$, Lower reward for τ

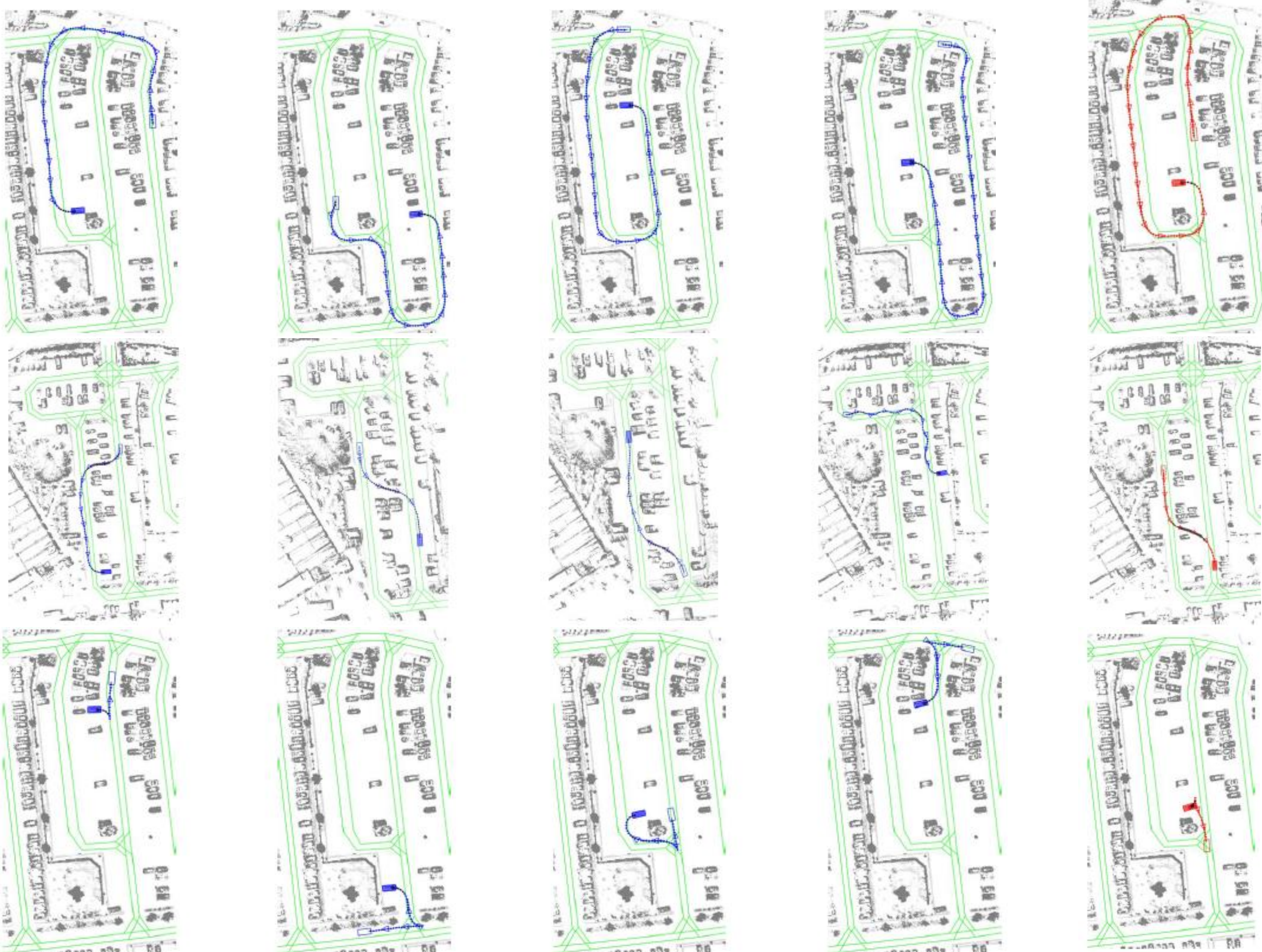
Reward Function

Find a Actor obtains large reward

Actor

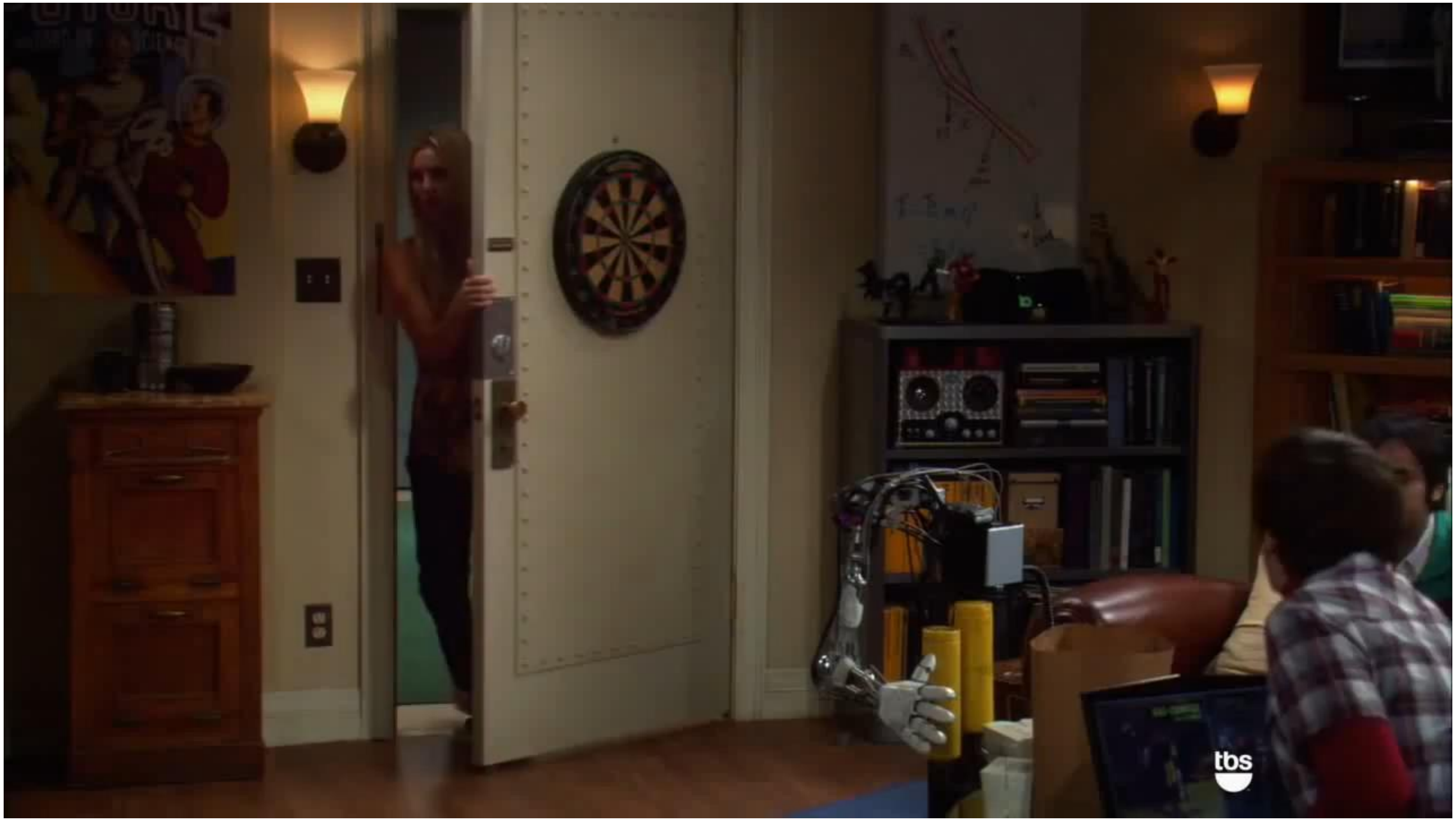
Parking Lot Navigation

- Reward function:
    - Forward vs. reverse driving
    - Amount of switching between forward and reverse
    - Lane keeping
    - On-road vs. off-road
    - Curvature of paths



Robot

- How to teach robots? https://www.youtube.com/watch?v=DEGbtjTOIB0



Robot

Chelsea Finn, Sergey Levine, Pieter Abbeel, " Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization", ICML, 2016

http://rll.berkeley.edu/gcl/

Guided Cost Learning:
Deep Inverse Optimal Control via Policy Optimization

Chelsea Finn, Sergey Levine, Pieter Abbeel
UC Berkeley

Third Person Imitation Learning

- Ref: Bradly C. Stadie, Pieter Abbeel, Ilya Sutskever, “Third-Person Imitation Learning”, arXiv preprint, 2017

First Person

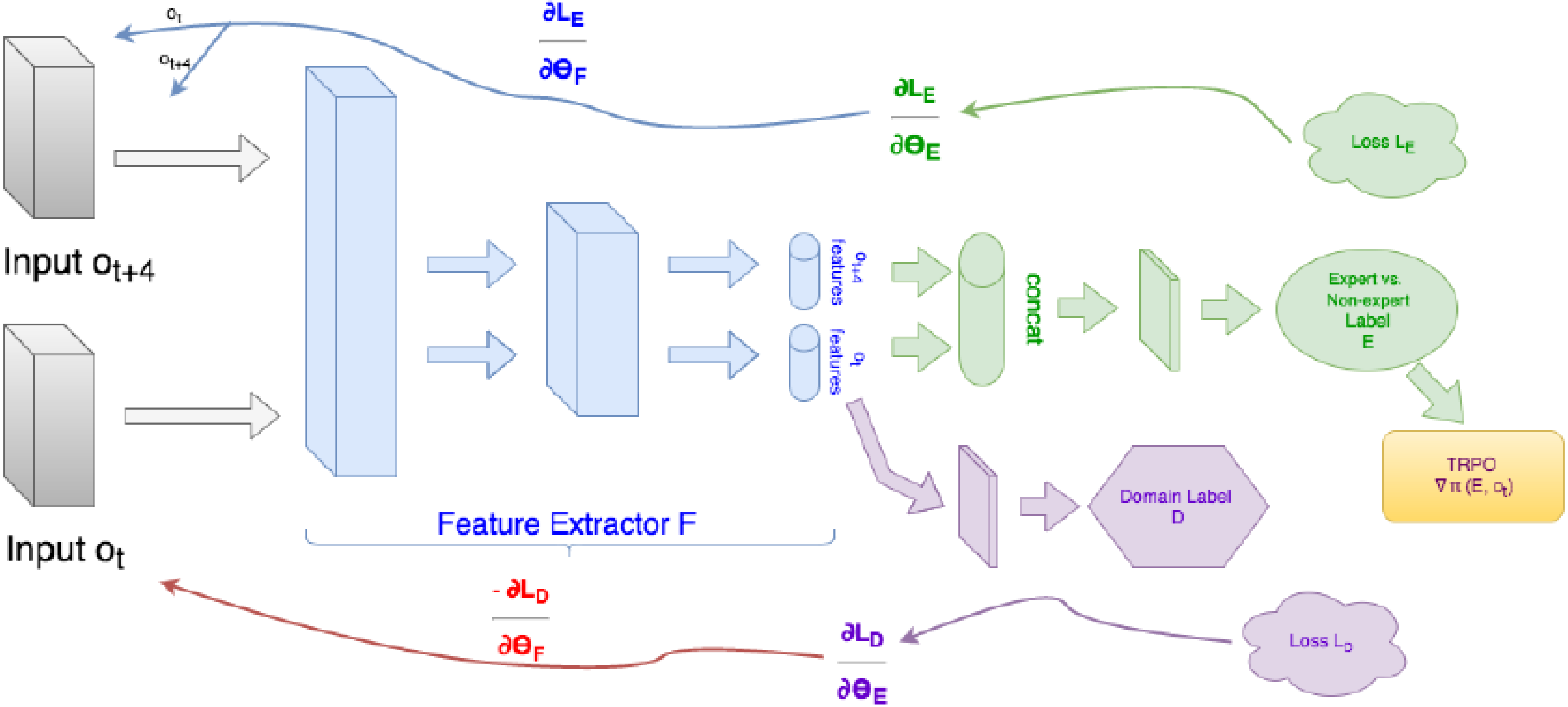
http://lasa.epfl.ch/research_new/ML/index.php

Third Person

https://kknews.cc/sports/q5kbb8.html

http://sc.chinaz.com/Files/pic/icons/1913/%E6%9C%BA%E5%99%A8%E4%BA%BA%E5%9B%BE%E6%A0%87%E4%B8%8B%E8%BD%BD34.png

Third Person Imitation Learning


o_t
o_{t+4}
$\frac{\partial L_E}{\partial \theta_F}$
$\frac{\partial L_E}{\partial \theta_E}$
Loss L_E
Input o_{t+4}
o_{t+4} features
o_t features
concat
Expert vs. Non-expert Label E
TRPO $\nabla \pi (E, o_t)$
Domain Label D
Feature Extractor F
Input o_t
$-\frac{\partial L_D}{\partial \theta_F}$
$\frac{\partial L_D}{\partial \theta_E}$
Loss L_D

Recap: Sentence Generation & Chat-bot

Sentence Generation

Expert trajectory:
床 前 明 月 光

(s_1, a_1): (“<BOS>”,”床”)

(s_2, a_2): (“床”,”前”)

(s_3, a_3): (“床前”,”明”)

⋮ ⋮

Chat-bot

Expert trajectory:
input: how are you
Output: I am fine

(s_1, a_1): (“input, <BOS>”,”I”)

(s_2, a_2): (“input, I”, ”am”)

(s_3, a_3): (“input, I am”, ”fine”)

⋮ ⋮

Maximum likelihood is behavior cloning. Now we have better approach like SeqGAN.