

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

就是监督学习

Behavior Cloning

Yes, this is supervised learning.

- Self-driving cars as example

observation



Expert (Human driver): 向前

Machine: 向前

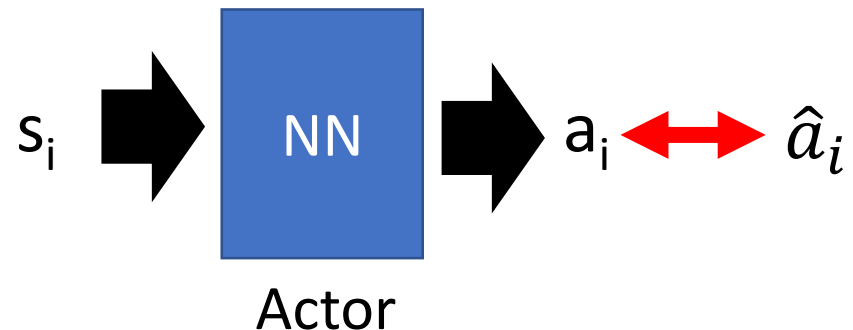
Training
data:

(s_1, \hat{a}_1)

(s_2, \hat{a}_2)

(s_3, \hat{a}_3)

.....



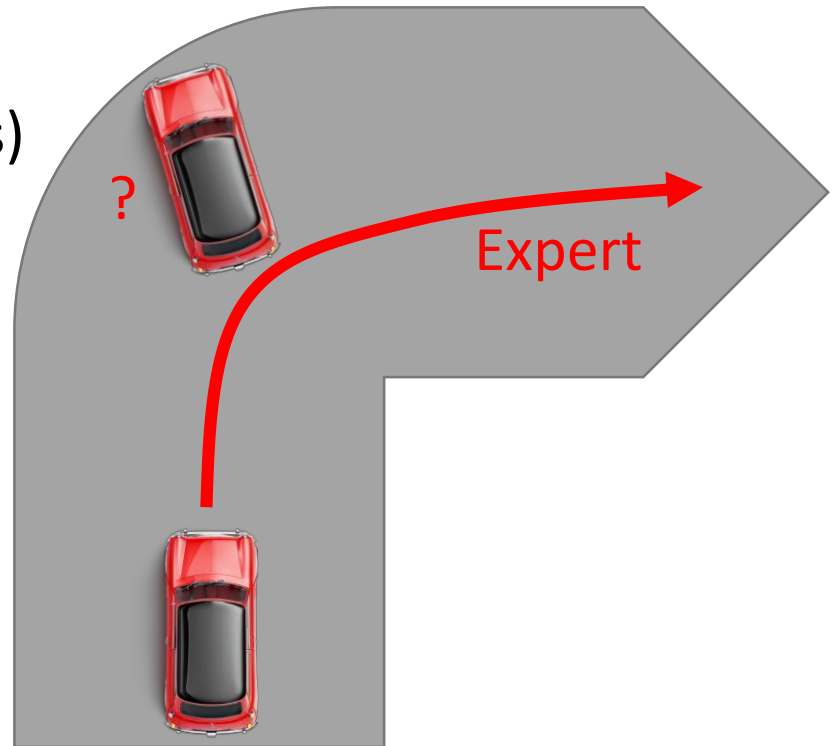
Behavior Cloning

- Problem

Expert only samples
limited observation (states)

Let the expert in the
states seen by
machine

Dataset Aggregation



Behavior Cloning

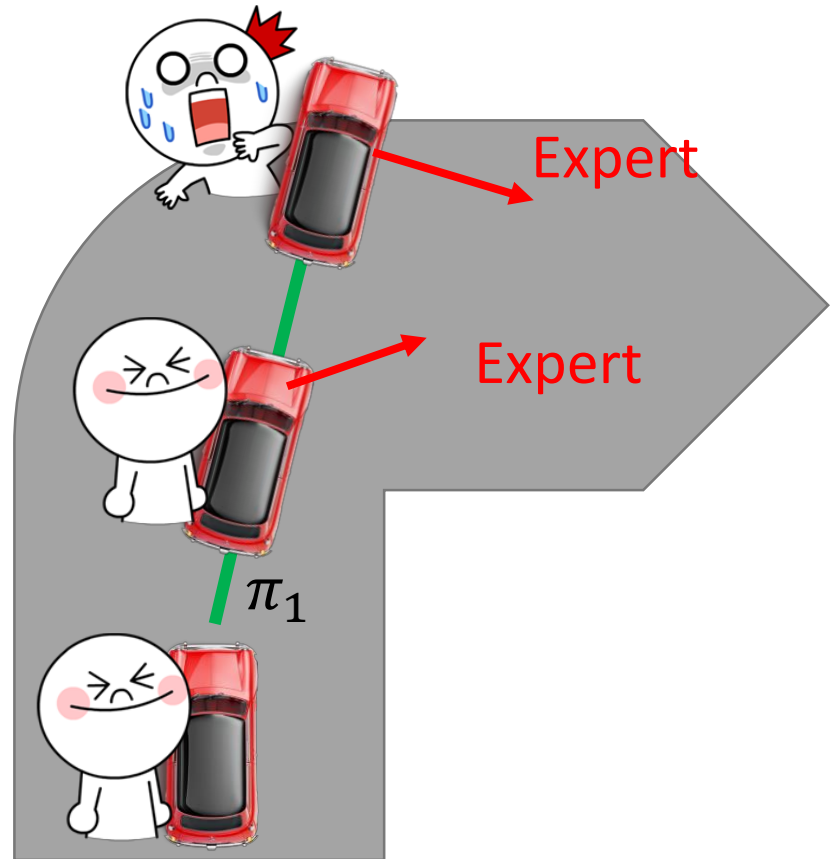
- Dataset Aggregation

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.

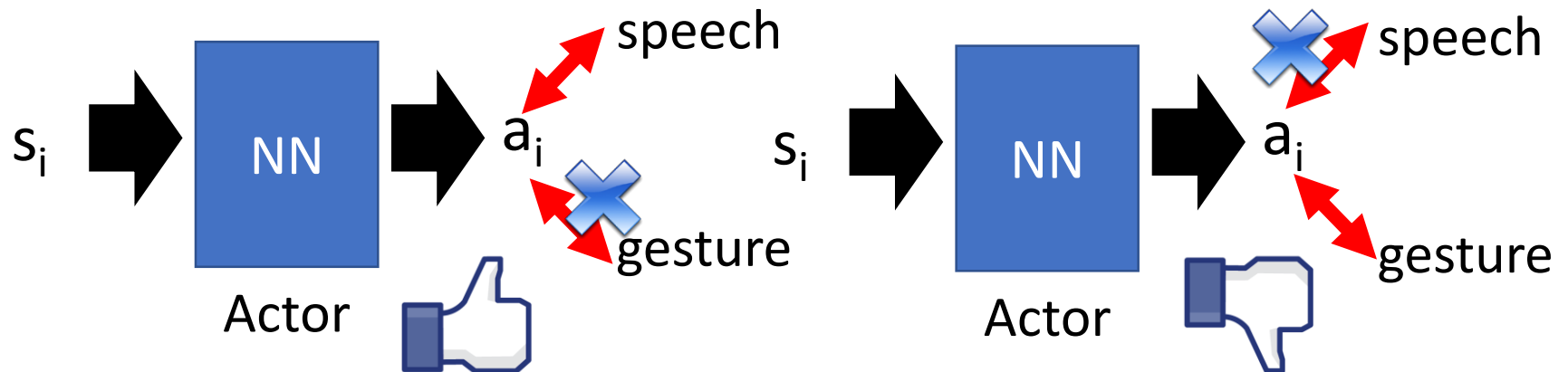


BANDICUT
Easy Video Cutter & Joiner
www.bandicam.com/bandicut

<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

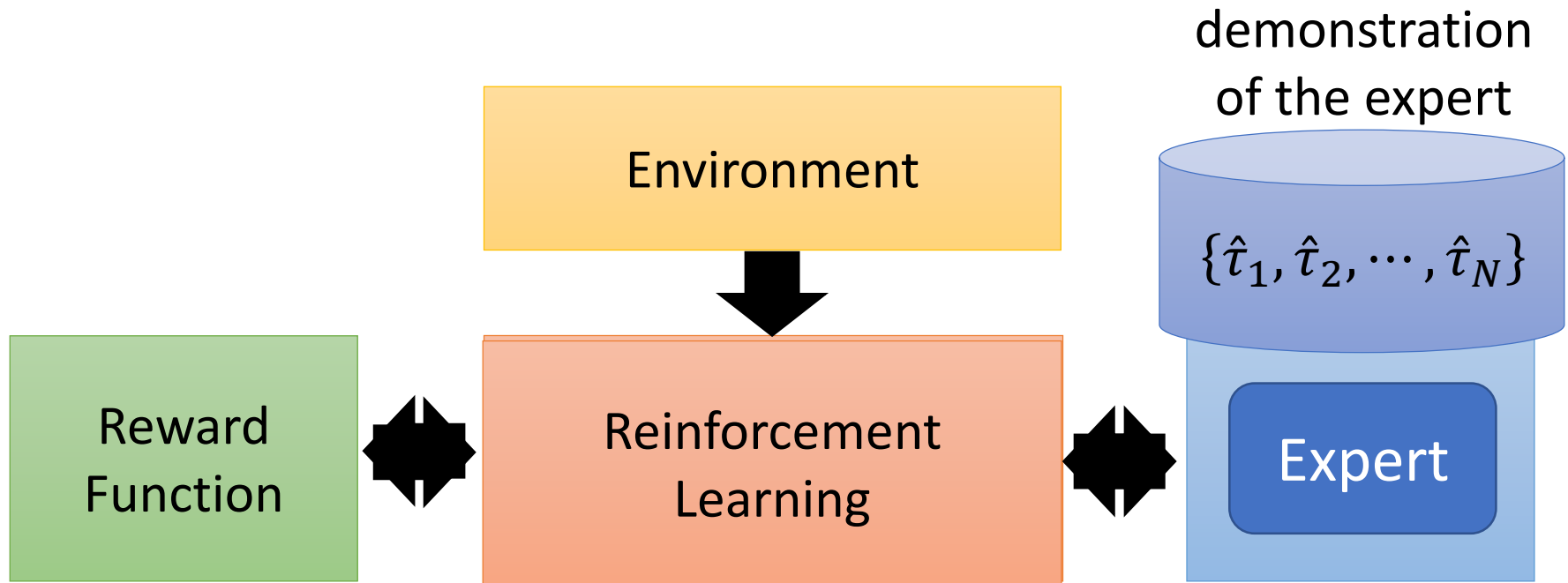
Mismatch



- 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)

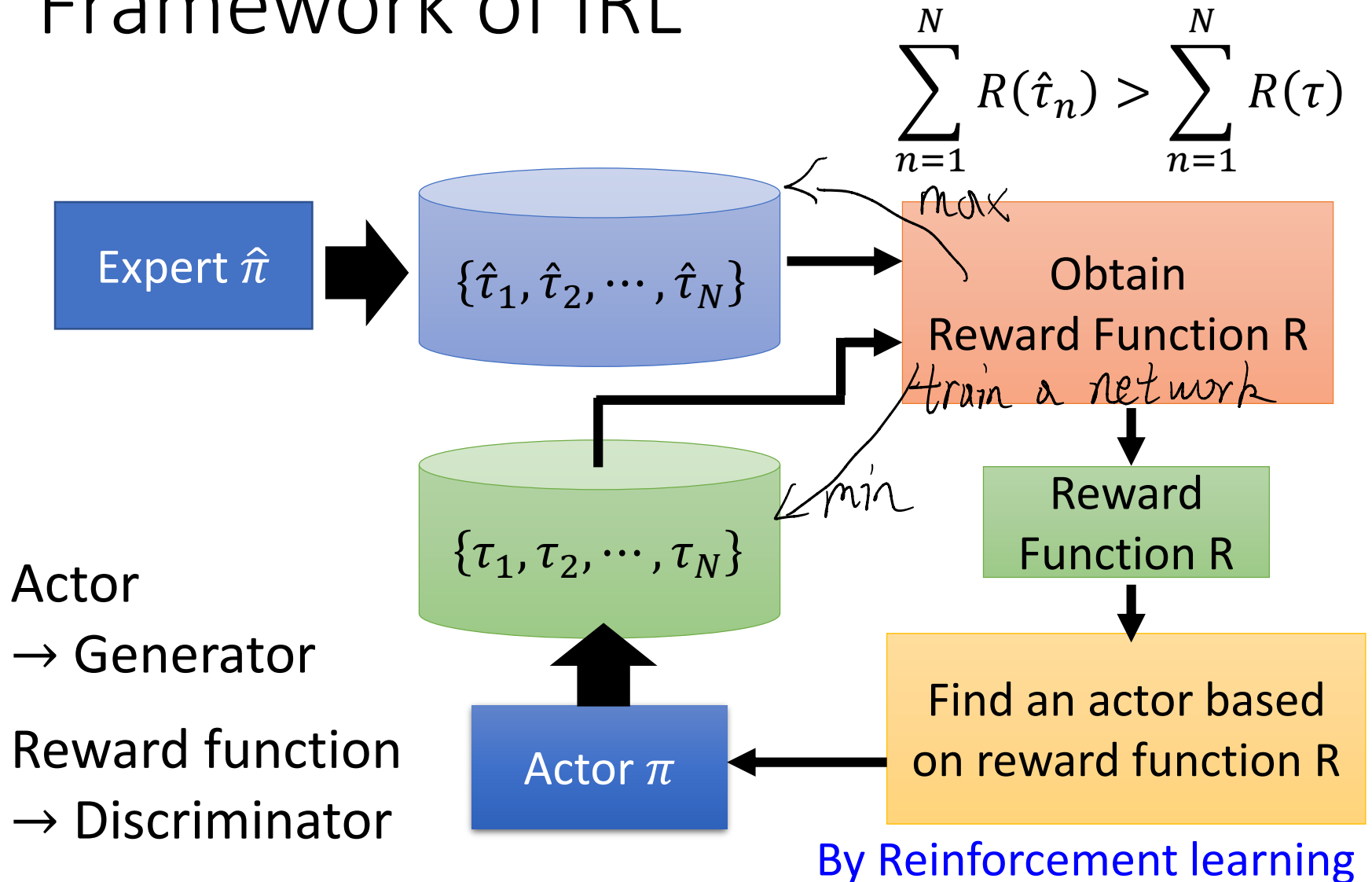
Inverse Reinforcement Learning



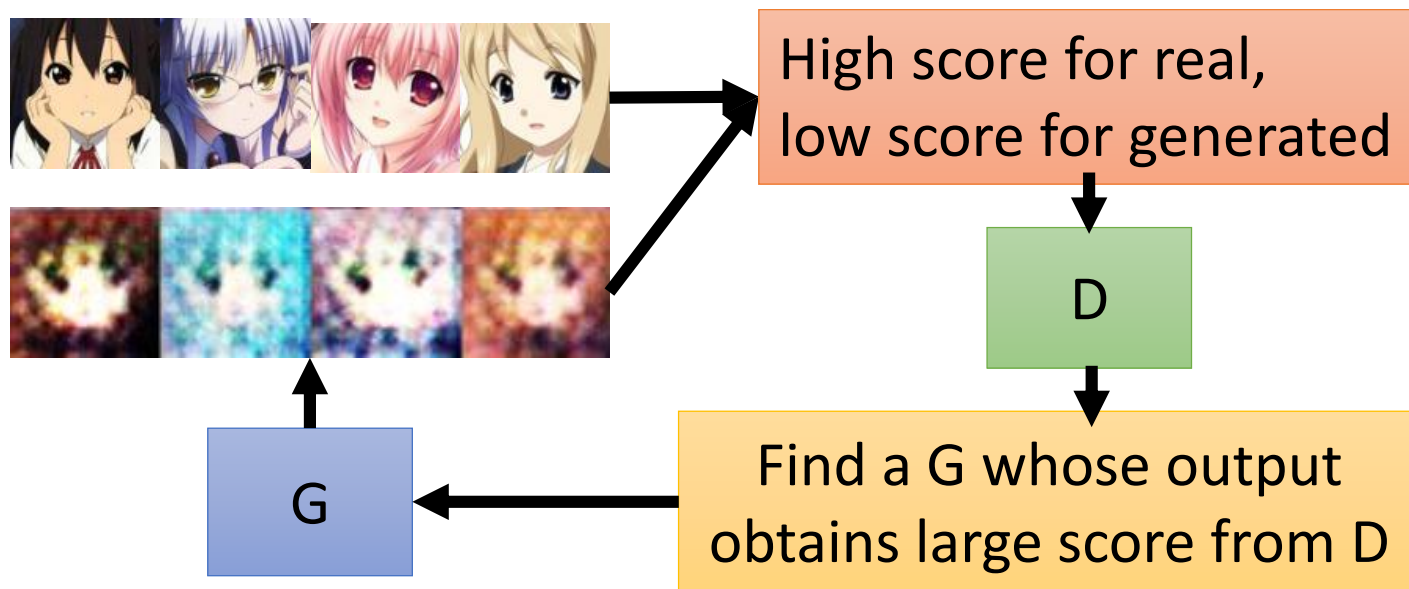
- Using the reward function to find the **optimal actor**.
- Modeling reward can be easier. Simple reward function can lead to complex policy.

The expert is always the best.

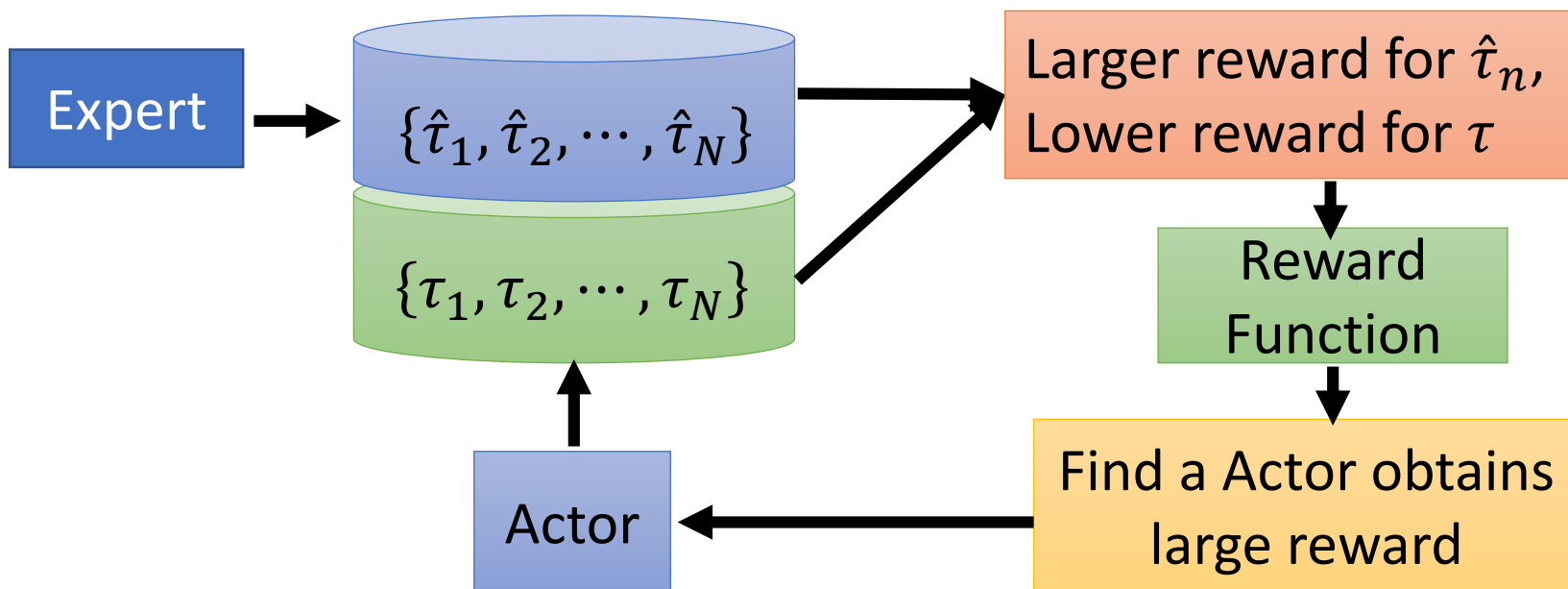
Framework of IRL



GAN



IRL



Parking Lot Navigation

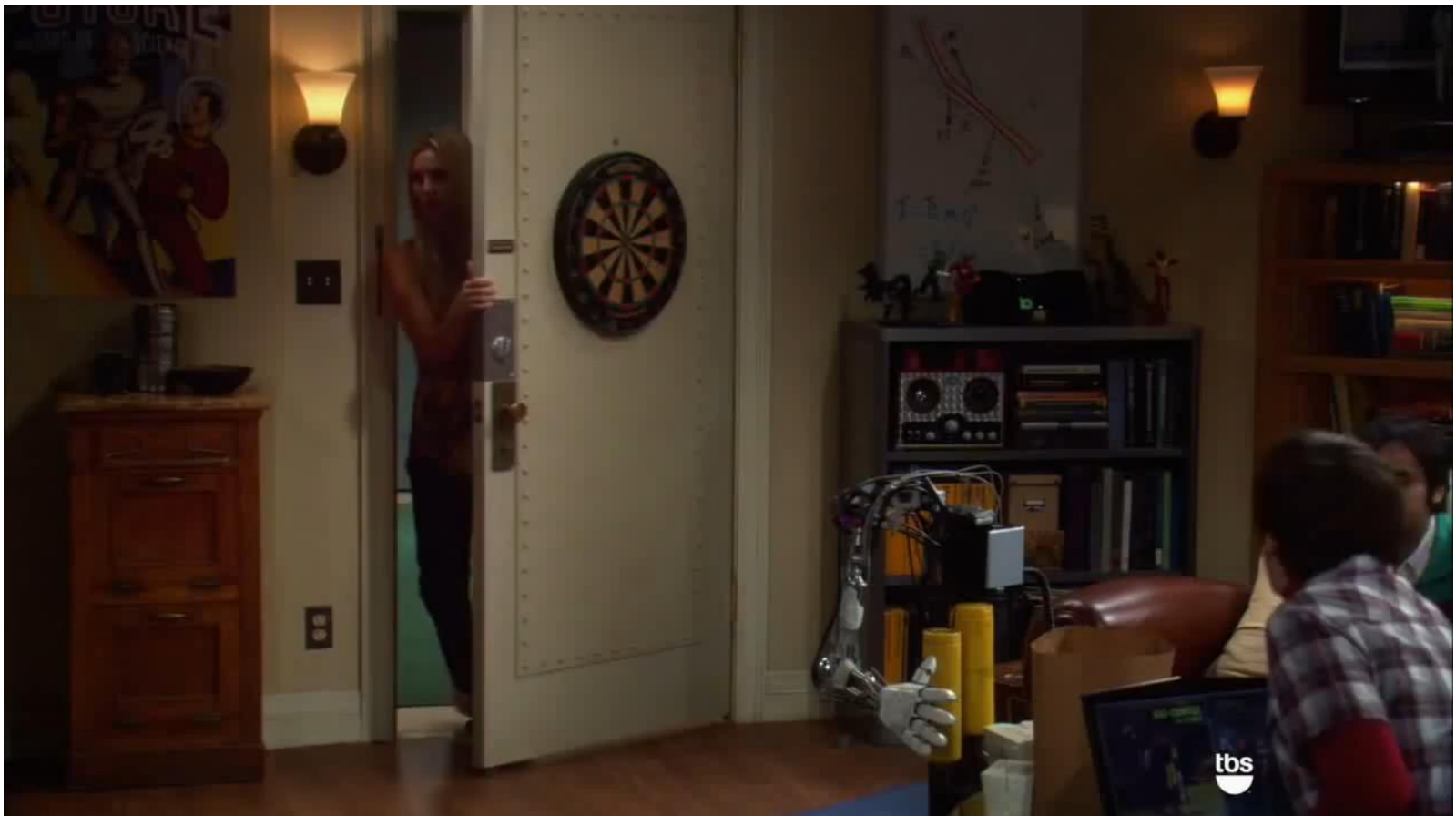


- Reward function:
 - Forward vs. reverse driving
 - Amount of switching between forward and reverse
 - Lane keeping
 - On-road vs. off-road
 - Curvature of paths



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

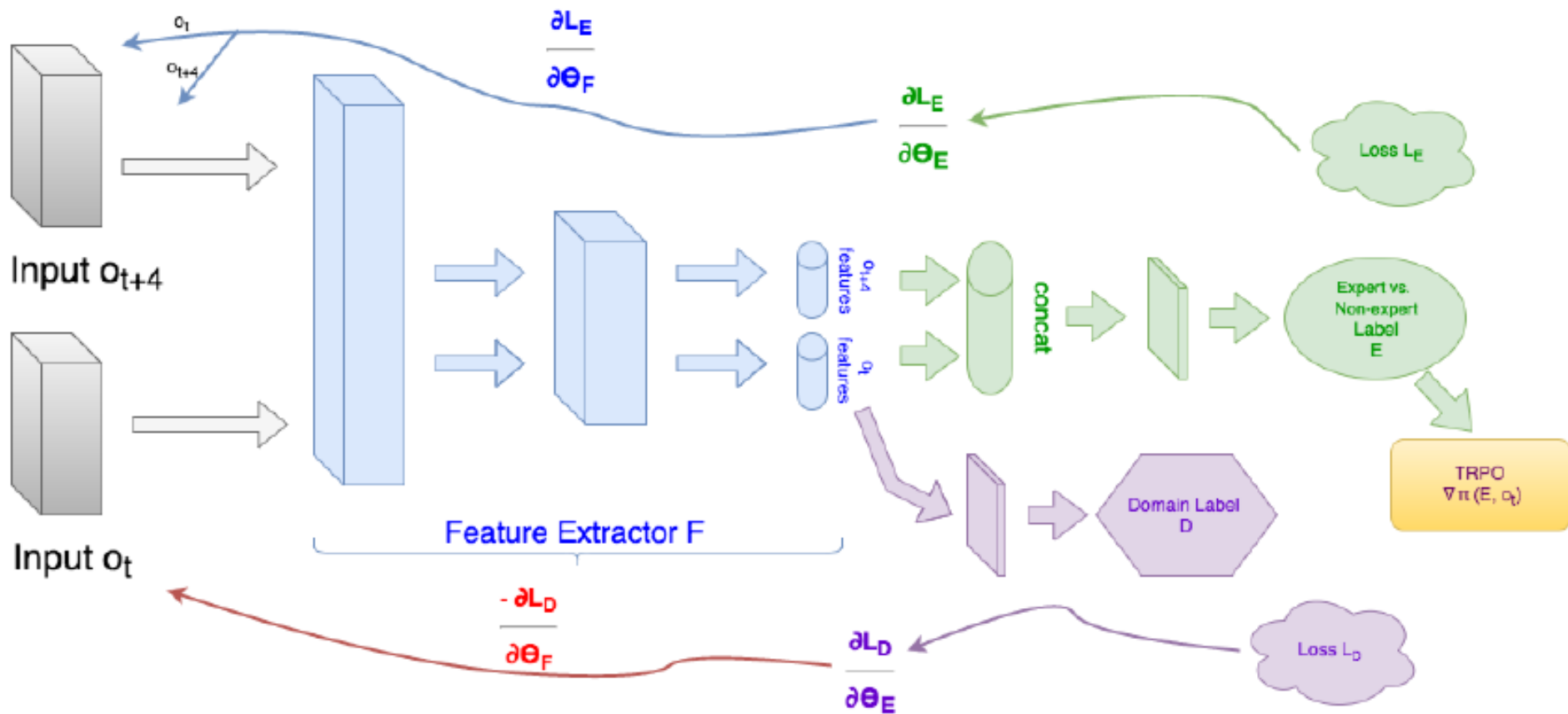
Third Person



<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



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.