COSC 689: Deep Reinforcement Learning

Spring 2019

Prof. Grace Hui Yang
Department of Computer Science
Georgetown University

Logistics

- Homework 1
    - Short questions and programming
- Midterm — will be after Deep Q-learning

Q&A

- First, I just want to make sure this would be a good generalization - RL and IL seem like they attempt to ultimately accomplish the same goal - development of optimal policies to guide an agent, generally through a Deep Learning network of some kind. RL relies on it's reward function for feedback on it's progress and then pursues an exploitative / exploration strategy to develop a policy/policies for the agent to use. Imitation Learning learns the policy/policies (like in Dagger) based on a given data set / oracle access.

- Question 1) It seems like the two methods could be combined - jump start the training process with IL, and then continue the training with RL. Is this typically done in real life for the development of agents? Or are there often significant complications in combining the two methods?

- Question 2) Combining learning methods would seem to offer several benefits in certain areas, such as helping the agent learn relatively complex actions / sequences that an agent might have trouble working out on it's own. Would this also tend to retard the development of radically new policies that a RL algorithm might learn on it's own if it started from nothing?

- Q1: For learning algorithm SEARN, SMILe and Dagger, all of them start with policy π0, expert's action, which I assume is the optimal action choice. I am thinking a situation that all three algorithms start with a policy which is stochastic (may be average action choice, or even a bad action choice), everything else remains the same, will it change the ultimate efficacy of the learning algorithms?

- Q2: Furthermore, all three learning algorithms are depending on expert's action choices as an input value to some extent throughout their training processes, what if we change all the expert's action choice to a stochastic one, what will happen to the efficacy of the learning algorithms?

Outline

- Imitation Learning
  - What is Imitation Learning? What is the challenge?
  - Behavior Cloning
  - DAgger
- Case Studies: Deep Models for Imitation Learning
- Why imitation learning does not work well?

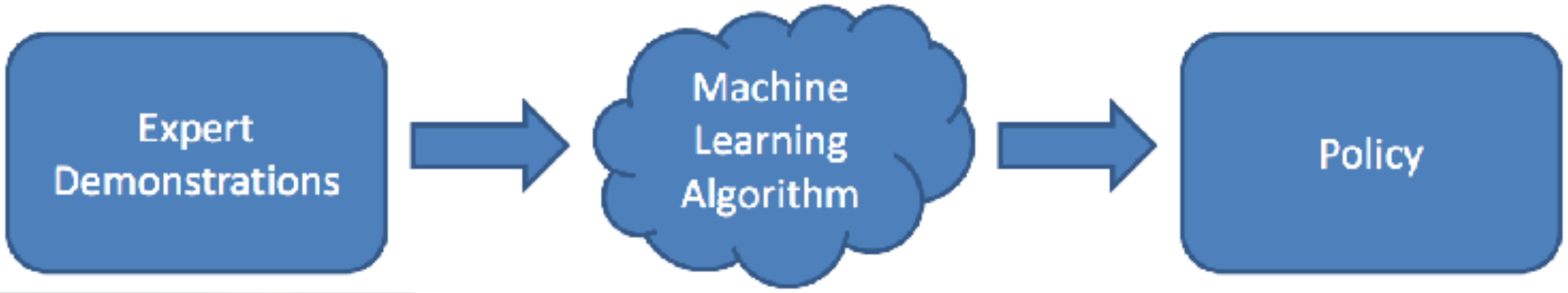
What is Imitation Learning?

- Supervised learning for decision making
- Learning by demonstration
  - Sometimes also known as learning from demonstration (LfD)

IL vs. RL vs. SML

- IL is not RL, but SML
- IL and RL aim to solve the same type of problem — sequential decision making
  - But they take different approaches
- The purpose of mentioning IL in this DRL course
  - Show you an alternative which uses familiar SML
  - Show you it does not work well and why
  - Understand RL better from a different perspective

What is Imitation Learning?

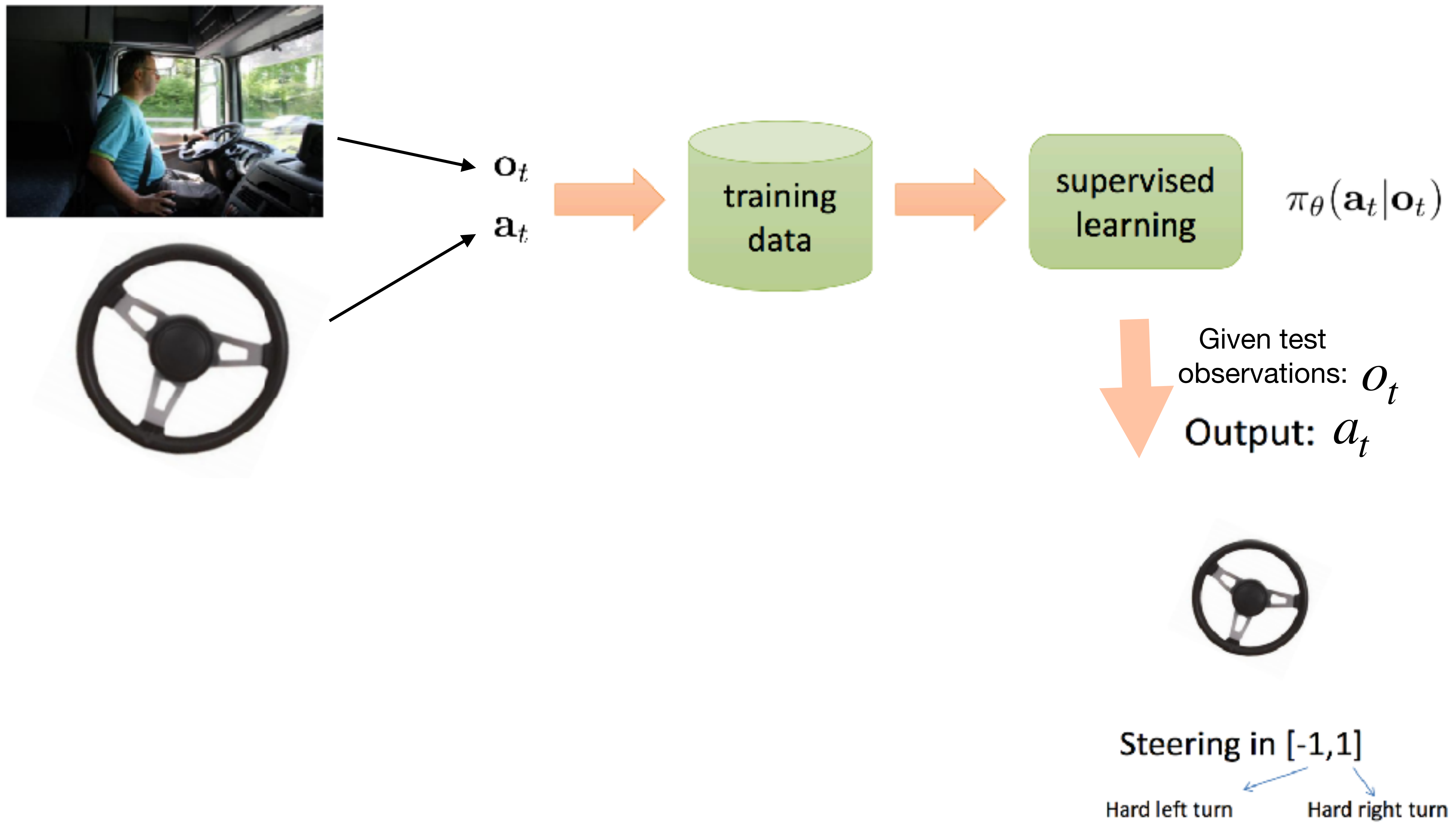


From https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-Slides.pdf

Main Issues

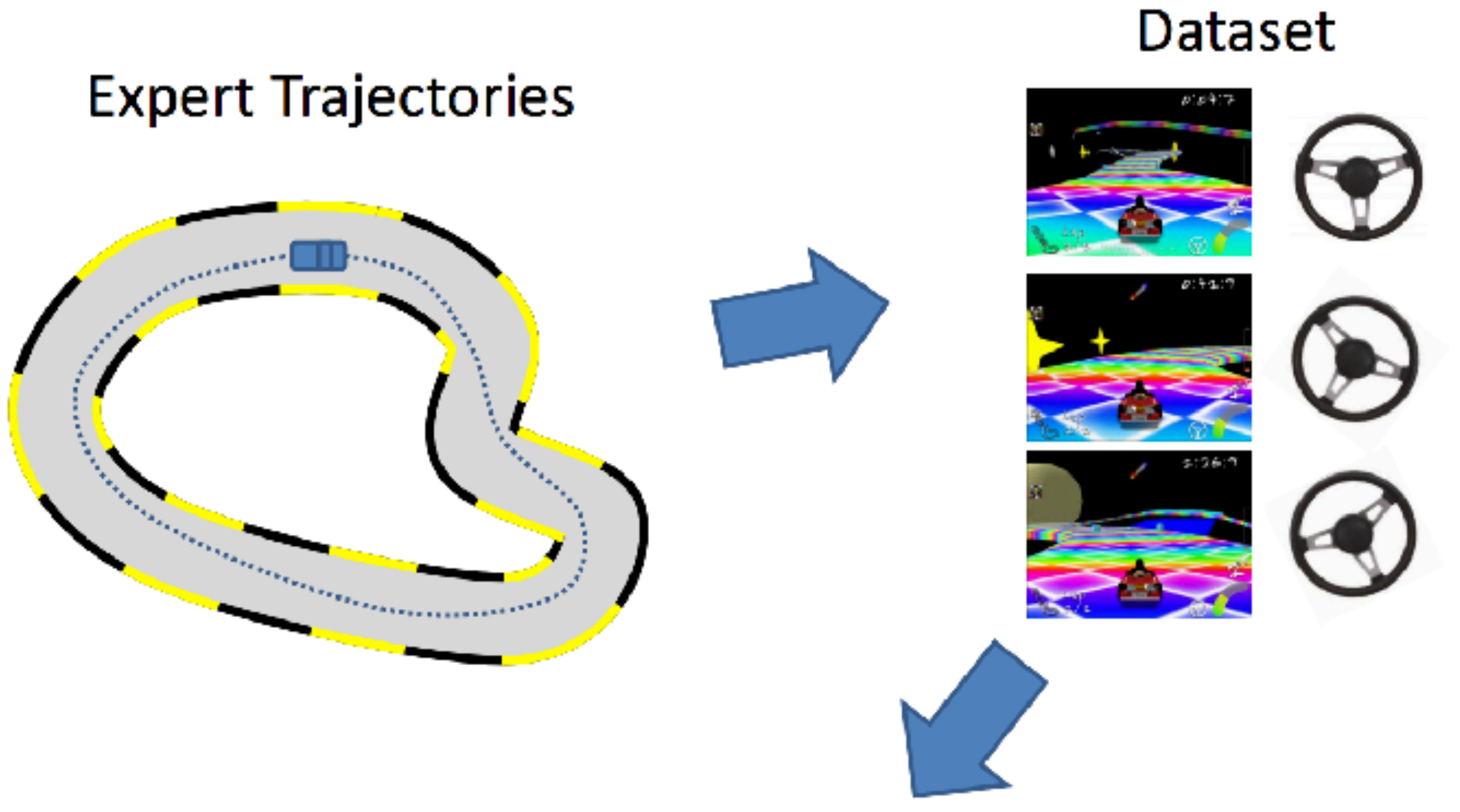
- Severe data sparsity
  - when considering the entire history (sequence of decision making in the demonstration) as training data
  - Hard to find repeated trajectories
- Amplified human inconsistence when dealing with a sequence of actions
  - Unreliable training data

Imitation Learning by SML



Modified from Sergey Levine's CS294-112

IL: Behavior Cloning



Learned Policy: $\hat{\pi}_{sup} = \underset{\pi \in \Pi}{\arg\min} \; \underset{s \sim D(\pi^*)}{\mathbf{E}} [\ell(\pi, s, \pi^*(s))]$

From https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-Slides.pdf

Case Study: Self-Driving Cars

- Input: the training data included video from two cameras coupled with left and right steering wheel angle (1/turning radius) from an expert (a human driver)
- Output: steering wheel angle

CNNs for Self-driving cars

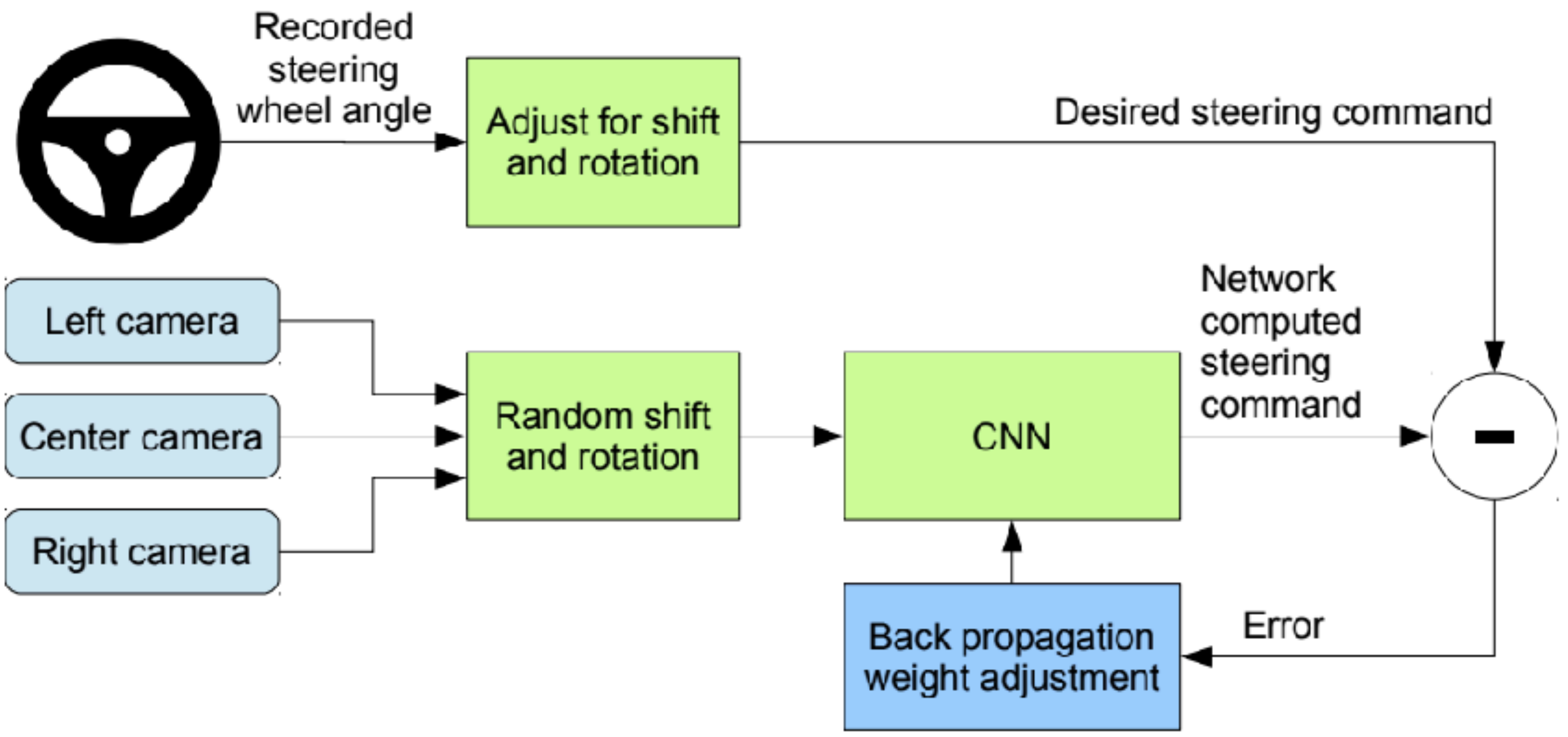

Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Left camera
Center camera
Right camera
Random shift and rotation
CNN
Network computed steering command
–
Back propagation weight adjustment
Error

Image from https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf

CNNs for Self-driving cars

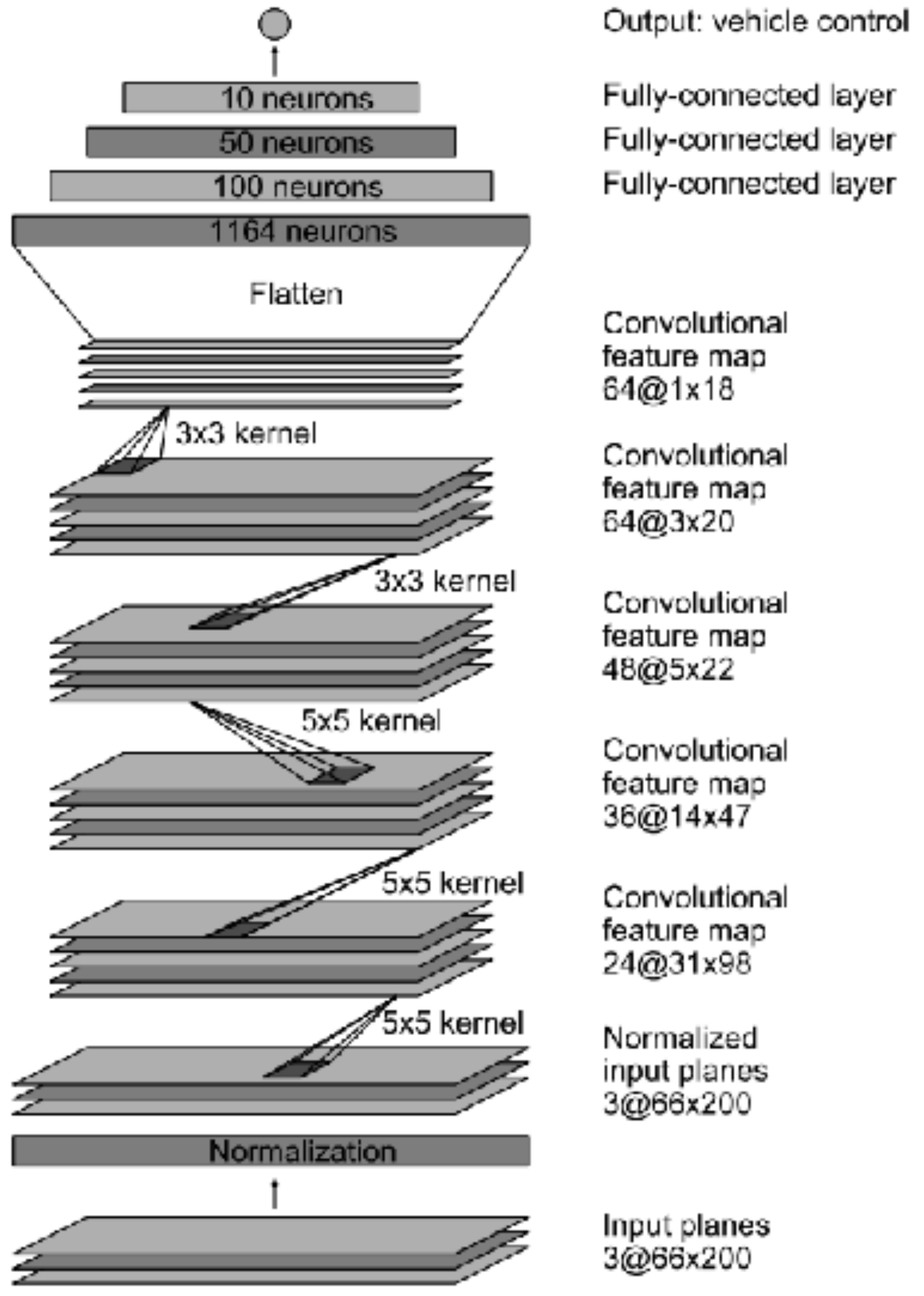
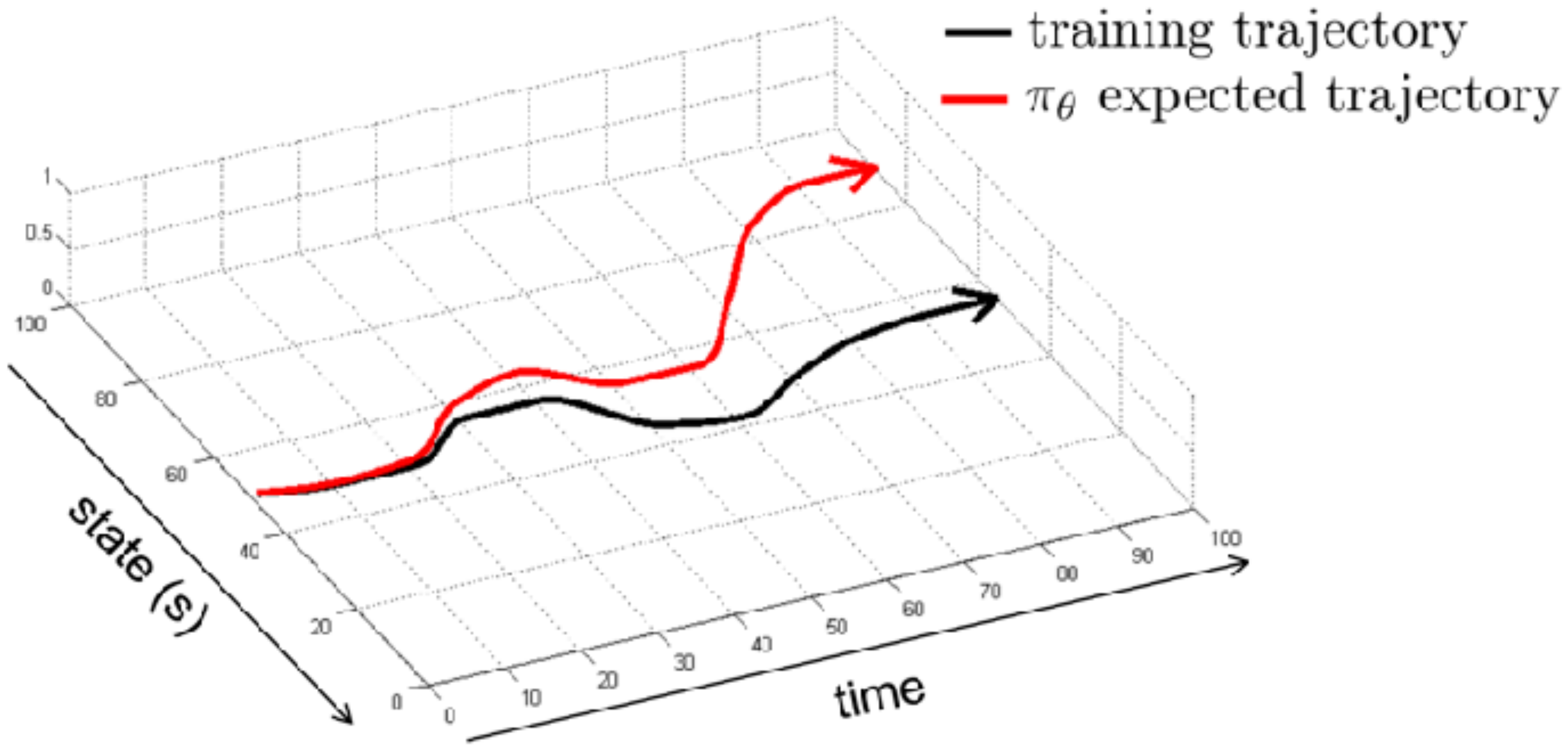


Figure 4: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

Image from https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf

Problems in Behavior Cloning

- Poor performance in practice
  - High training error and test error
- Doesn’t learn how to recover from errors



From Sergey Levine’s CS294-112

Number of errors grow quadratically in time

$$\mathbb{E}_{s\sim d_{\pi^*}}[\ell(s,\pi)] = \epsilon, \textit{ then } J(\pi) \leq J(\pi^*) + T^2\epsilon.$$

Number of time steps in your sequence

How to bridge the gap?

Obtain more data

- Add artificial shifts and rotations to teach the network how to recover from a poor position or orientation
    - DAVE-2
- The magnitude of these perturbations is chosen randomly from a normal distribution.
- The distribution has zero mean, and the standard deviation is twice the standard deviation that we measured with human drivers

Obtain more data

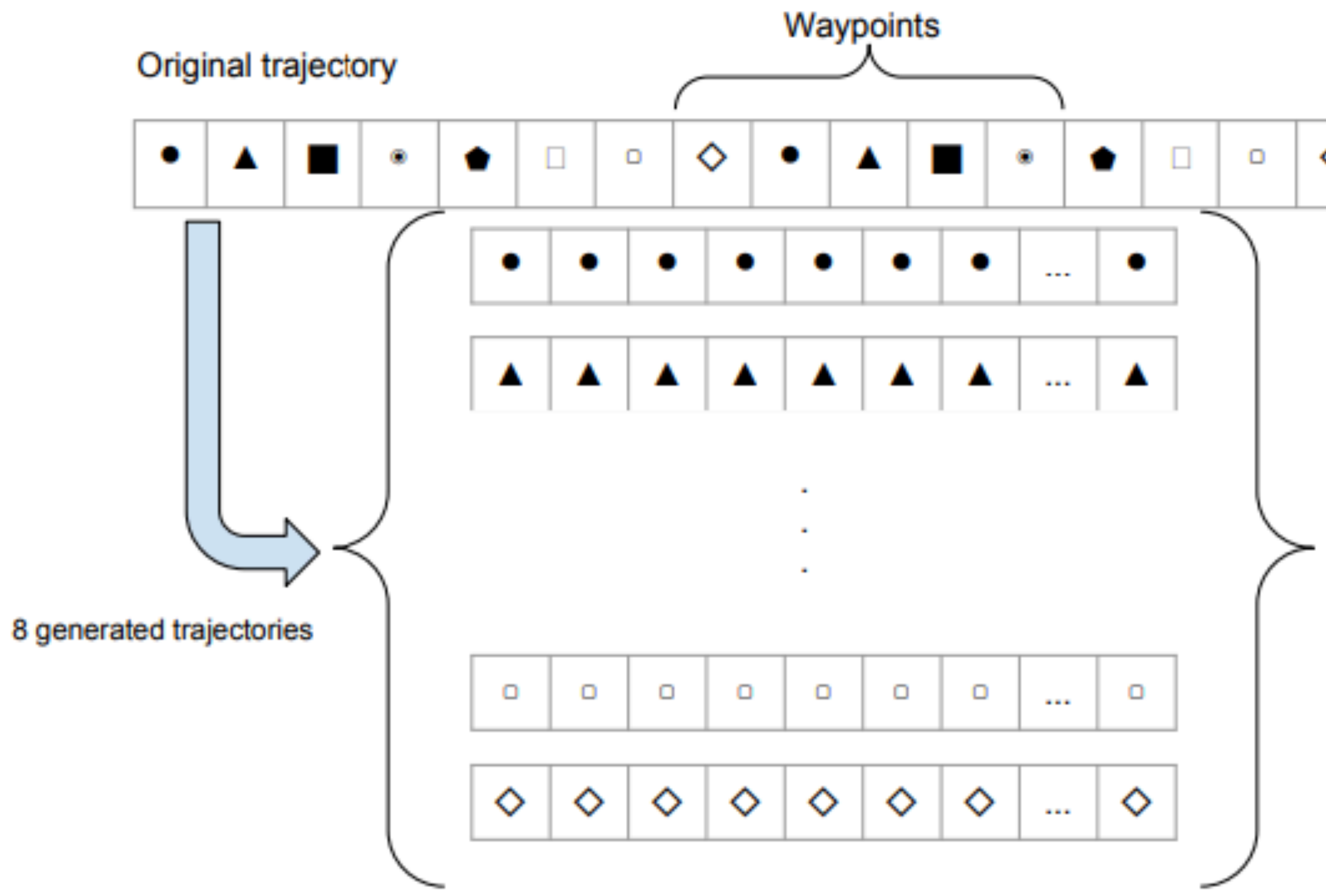


Figure 2: Creating multiple trajectories from a demonstration recorded at a higher frequency.

Image from https://arxiv.org/pdf/1603.03833.pdf

Obtain more data

- Actively ask the human expert to label new data
    - DAgger

DAgger

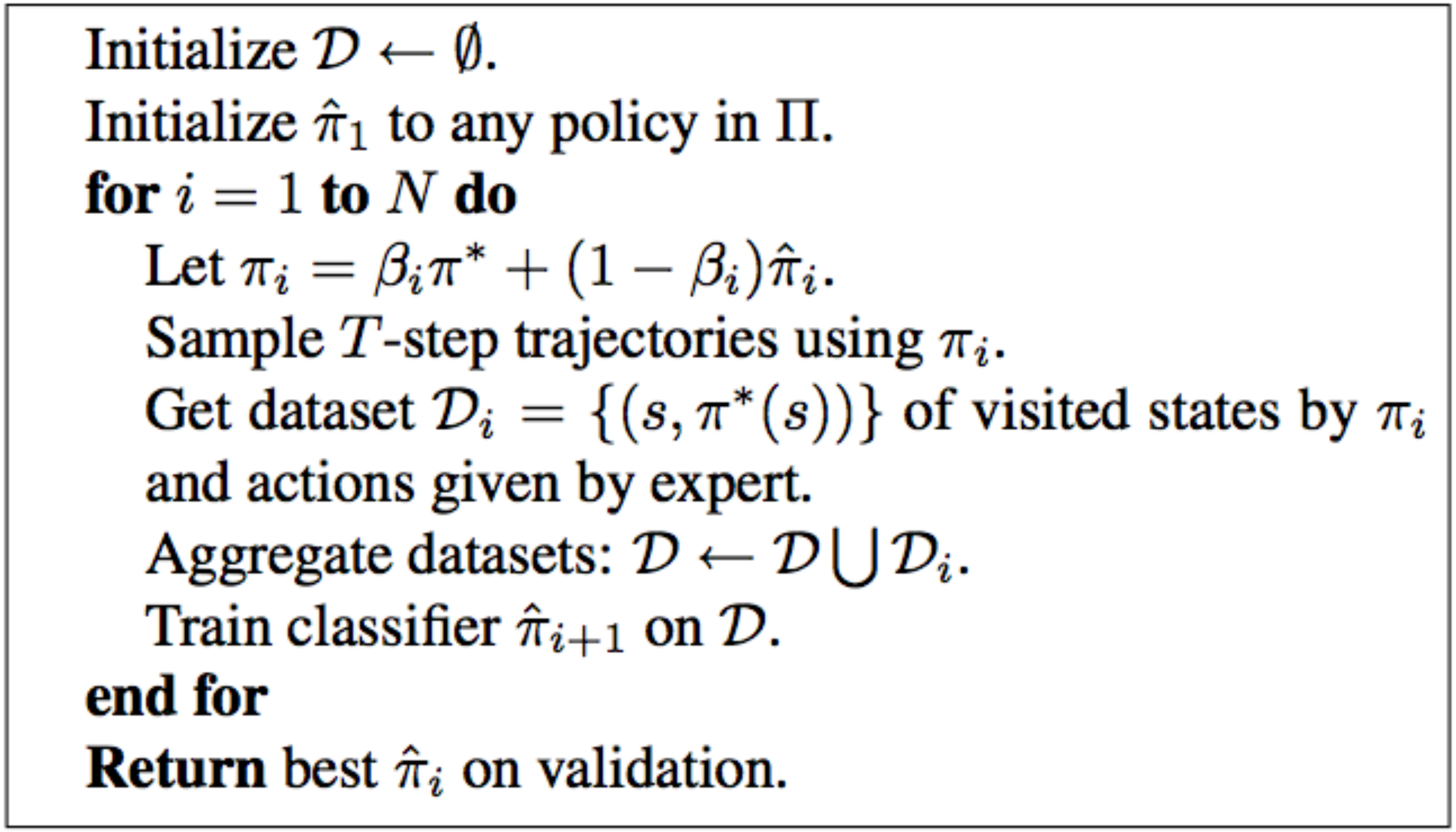
DAgger

- It is a kind of active learning
- Ask feedback from human expert for examples/actions that need new labels

Side note: Supervised ML vs. Active Learning

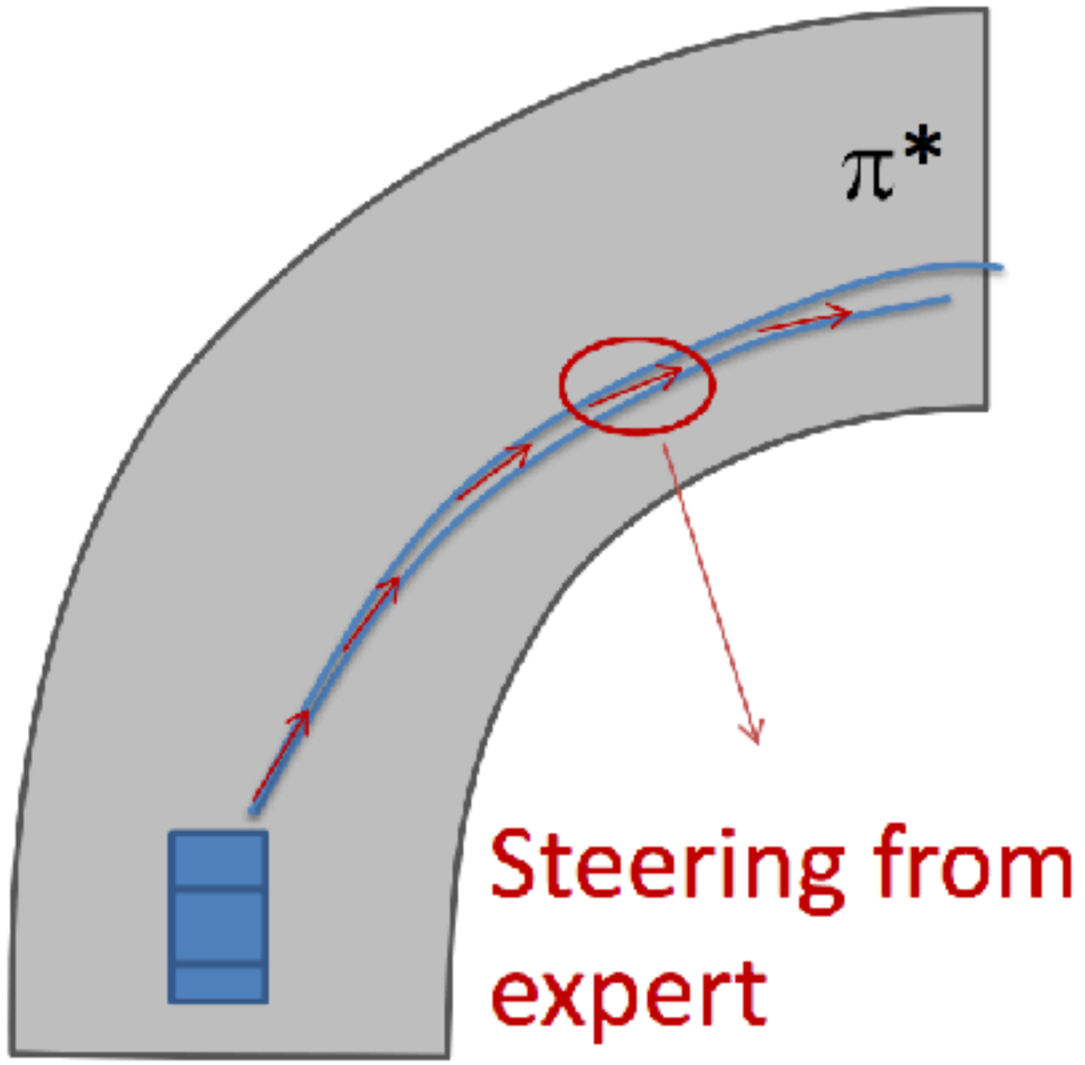
- Active learning: the algorithm asks the human to label some highly uncertain examples, to update the decision boundary
  - Ask smartly, e.g. ask humans to give new labels to data points on a decision boundary
  - It is an iterative process and humans are in the loop

DAgger Algorithm



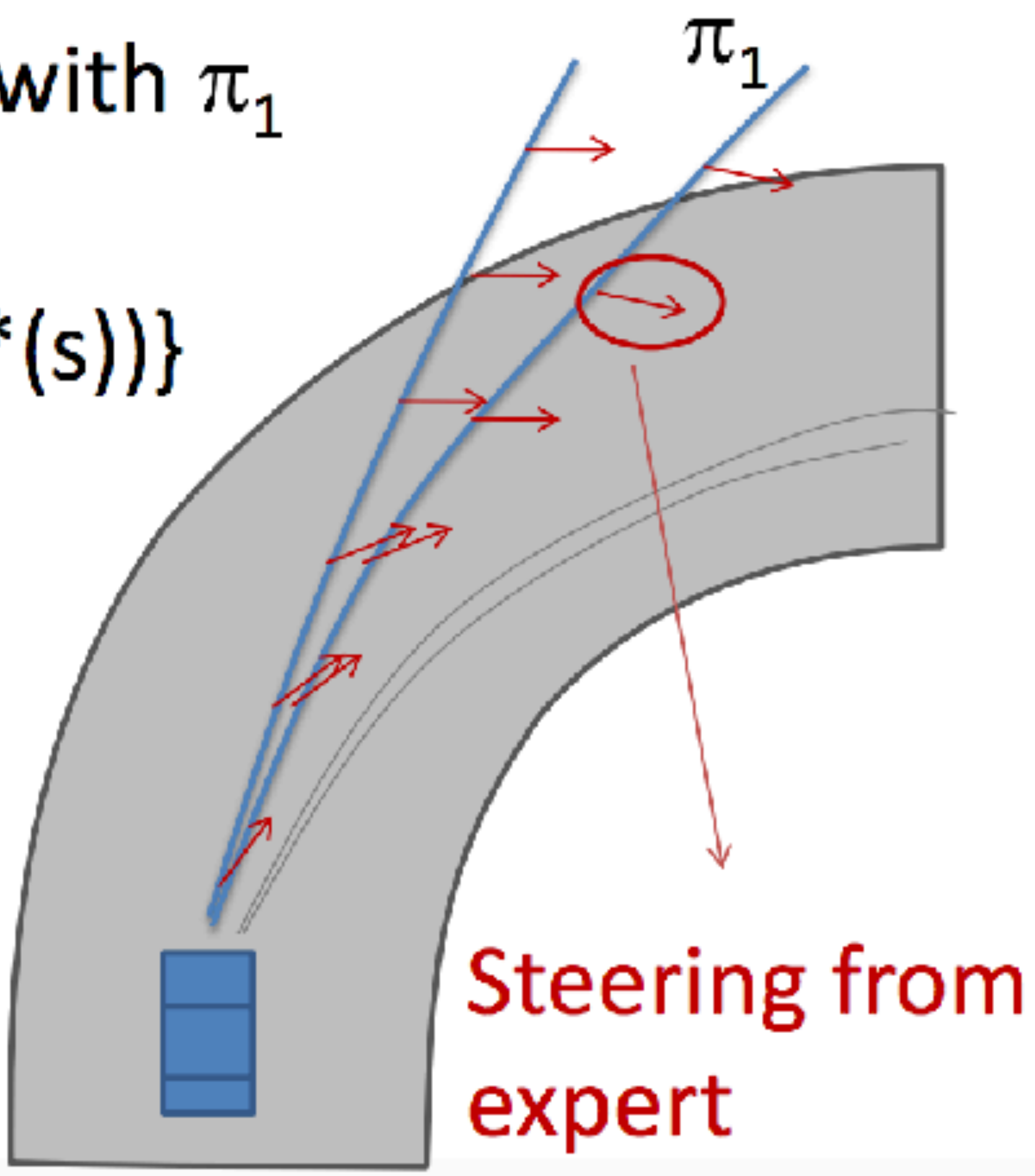
```
Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in $\Pi$.
**for** $i = 1$ **to** $N$ **do**
    Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
    Sample $T$-step trajectories using $\pi_i$.
    Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by $\pi_i$
    and actions given by expert.
    Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
    Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
**end for**
**Return** best $\hat{\pi}_i$ on validation.
```

- Collect trajectories with expert π^*
- Dataset $D_0 = \{(s, \pi^*(s))\}$
- Train π_1 on D_0



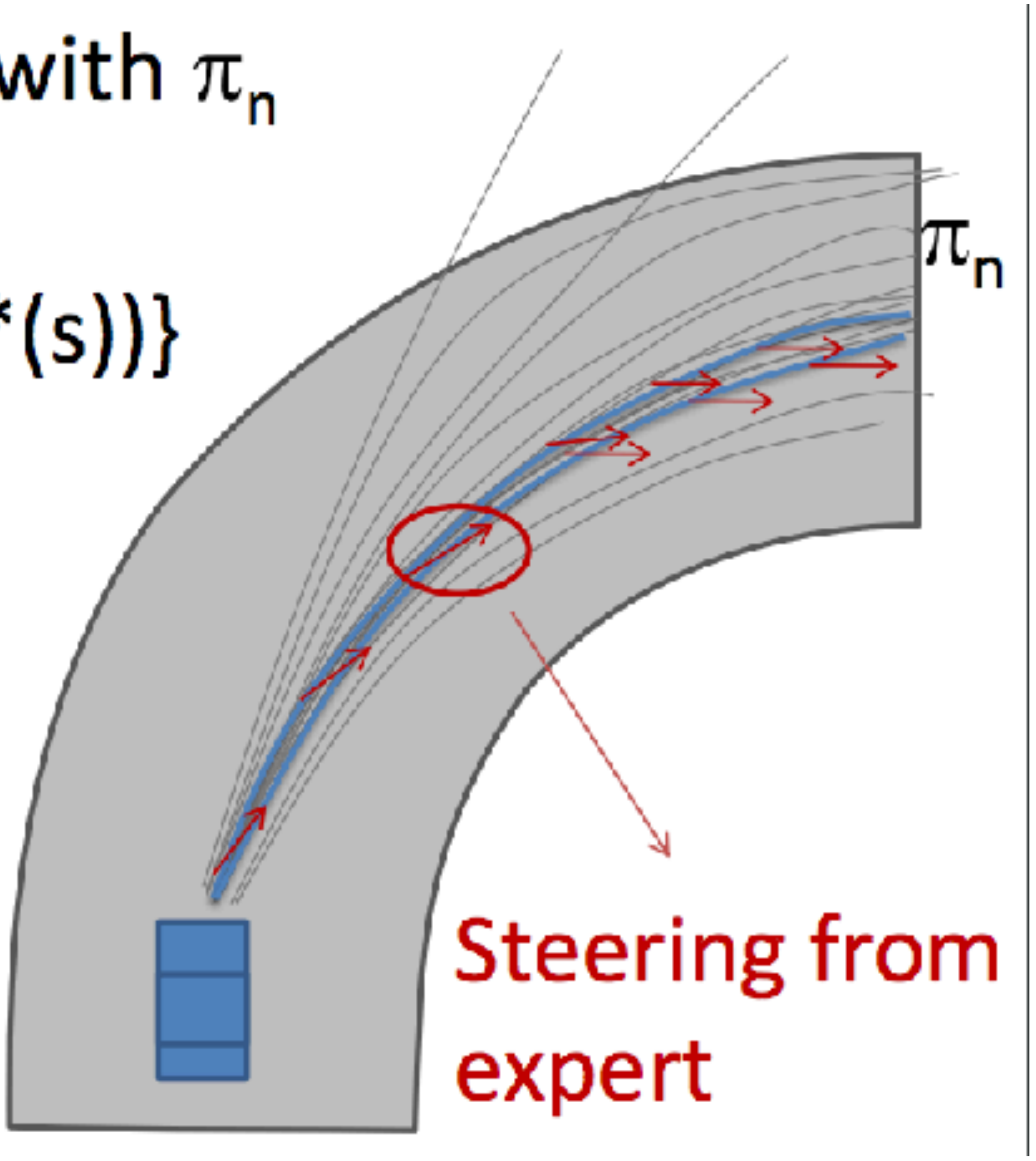
From https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-Slides.pdf

- Collect new trajectories with π_1
- New Dataset $D_1' = \{(s, \pi^*(s))\}$
- Aggregate Datasets:
  $D_1 = D_0 \cup D_1'$
- Train π_2 on D_1



From https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-Slides.pdf

- Collect new trajectories with π_n
- New Dataset $D_n' = \{(s, \pi^*(s))\}$
- Aggregate Datasets:
  $D_n = D_{n-1} \cup D_n'$
- Train π_{n+1} on D_n



From https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-Slides.pdf

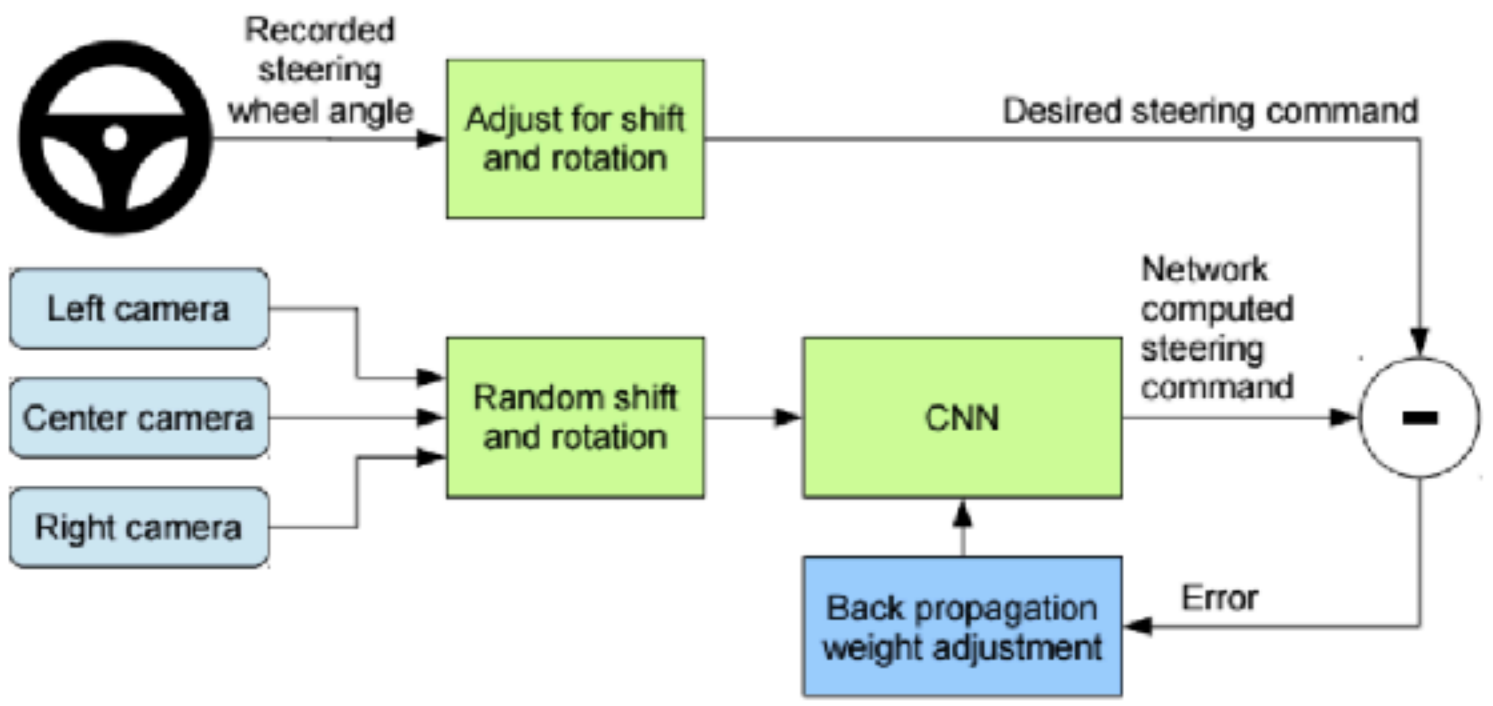
Summary: DAgger

- DAgger addresses the problem of distributional “drift”
- Use a simple iterative procedures to yield better performance
- However, it is still not as effective as RL

Deep Models for IL

- Recently, there are increasing interests in using deep learning models for IL
- Let us see some case studies

Imitation Learning with CNNs


Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Left camera
Center camera
Right camera
Random shift and rotation
CNN
Network computed steering command
-
Back propagation weight adjustment
Error

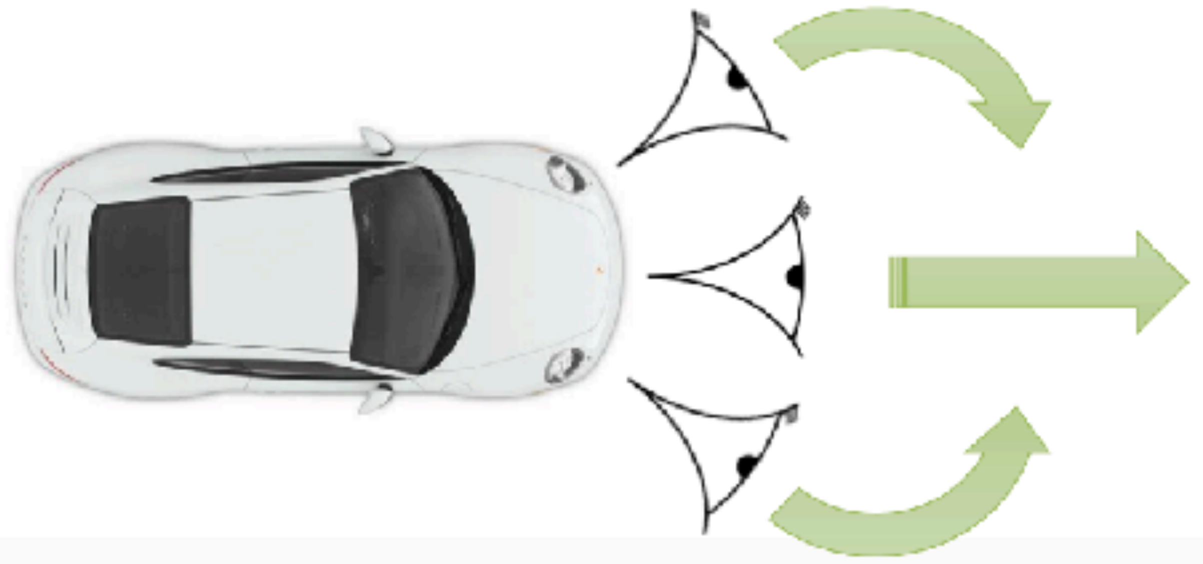


Image from https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf

Imitation Learning with RNNs

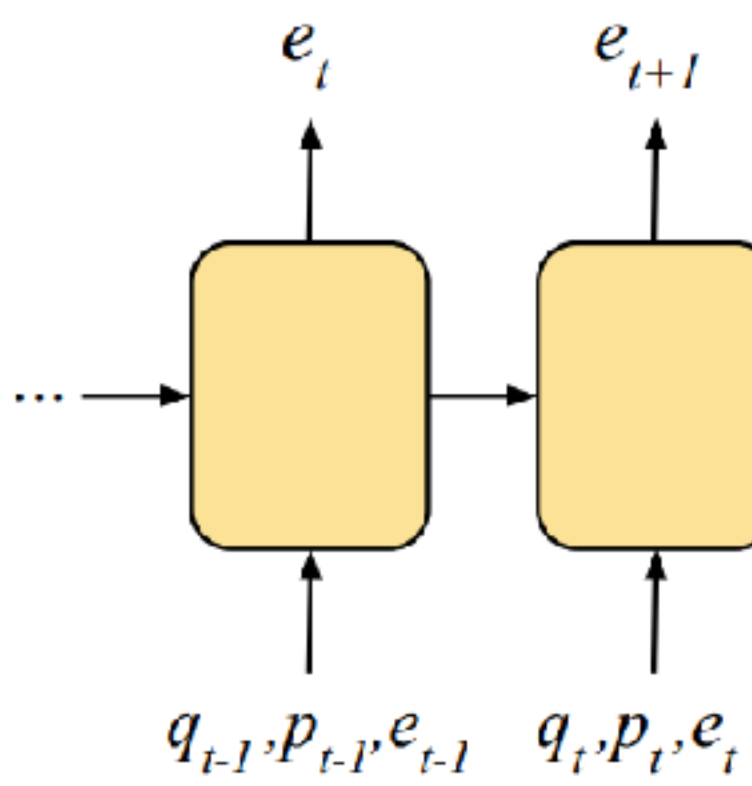


Figure 1: The input and the output of the RNN at each time step. e_t is the end-effector pose augmented with gripper status (open/close) at time t, p_t is the preferences of the human demonstrator, and q_t is the state of the environment containing the pose of all objects involved in the manipulation task.

Image from https://pdfs.semanticscholar.org/885b/5d5a2dee22dc5e7ee43e6340a771fe2192fd.pdf

Imitation Learning with LTSMs

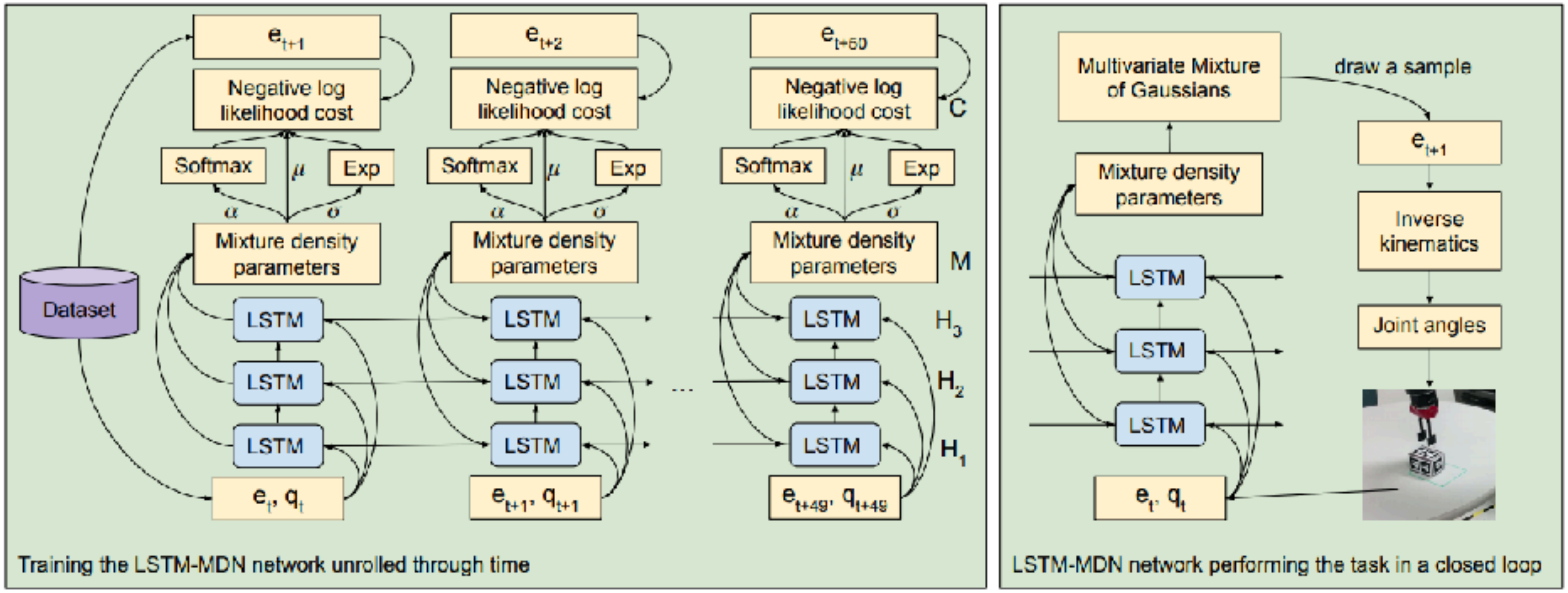


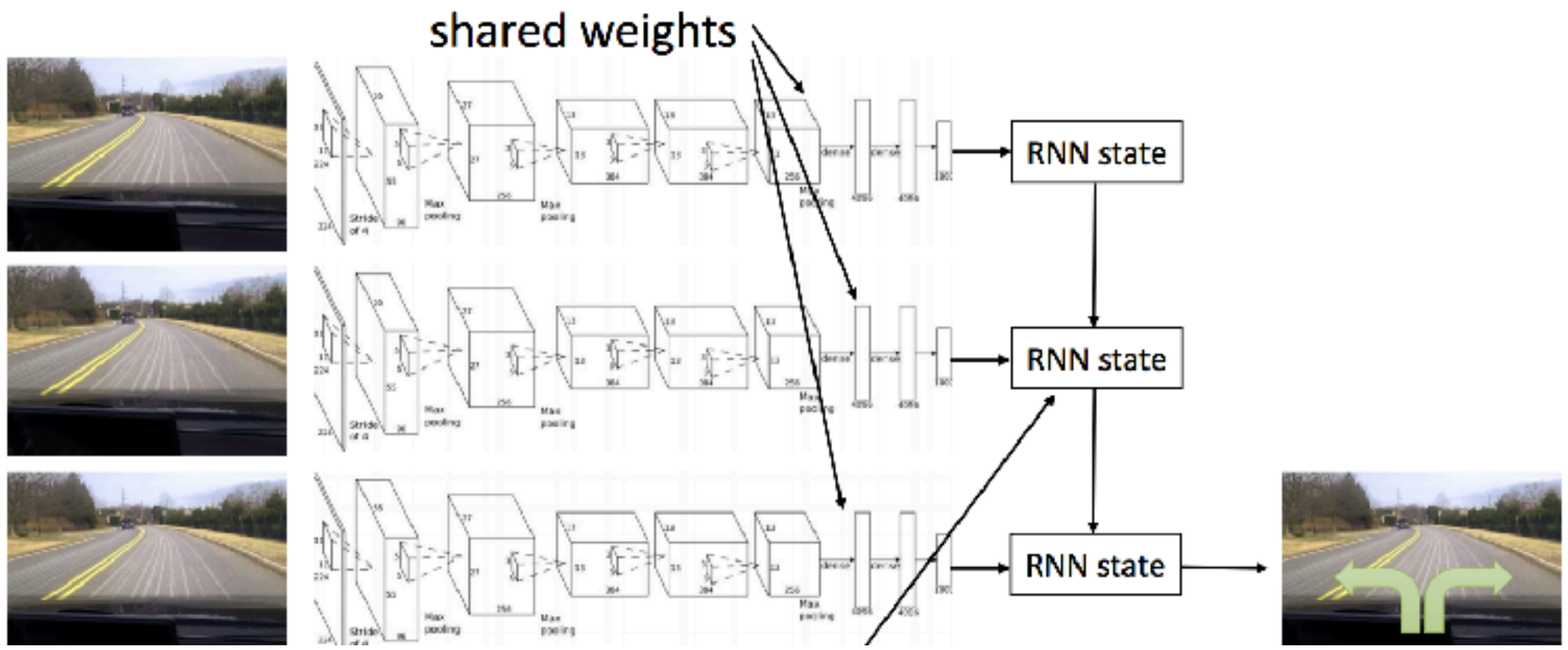
Image from https://arxiv.org/pdf/1603.03833.pdf

Summary: Imitation Learning

- Strengths:
    - It is familiar supervised learning
    - Works when add more on-policy data (DAgger, left/right images in self-driving car)
- Weaknesses:
    - But in general, IL does not work well
    - The active learning approach (DAgger) works relatively better but it must learn from humans
        - Humans are not good at providing some kinds of actions
    - Not work well even with help from Deep leaning models
        - Deep learning works best when data is plentiful, but IL requires human expert to provide training data, which is typically in a limited amount

Amplified Mismatch between Data & Model

- Imitation Learning is non-Markovian
    - Markovian models are more nature to human experts
- Hard to train over the entire history
    - Too many weights to train and too easy to overfit



Modified from Sergey Levine's CS294-112

Amplified Mismatch between Data & Model

- Demonstration from human experts is too complex — multimodal
  - With the added time dimension, the same expert can be inconsistent; different experts show low inter-rater agreement
    - Their behaviors often in the shape of mixture of gaussians
    - Too many hidden variables
- Humans often generate discrete data, because it is easy and nature for humans to do

Reference

- Imitation Learning by CNN: https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf
- Imitation learning by RNN: https://pdfs.semanticscholar.org/885b/5d5a2dee22dc5e7ee43e6340a771fe2192fd.pdf
- Imitation learning by LSTM: https://arxiv.org/pdf/1603.03833.pdf
- DAgger: https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-Slides.pdf
- DAgger: https://www.cs.cmu.edu/~sross1/publications/Ross-AIStats11-NoRegret.pdf
- UC Berkeley CS294-112 Deep Reinforcement Learning