APS360
Fundamentals of AI

Florian Shkurti

Behavioral Cloning vs. Imitation

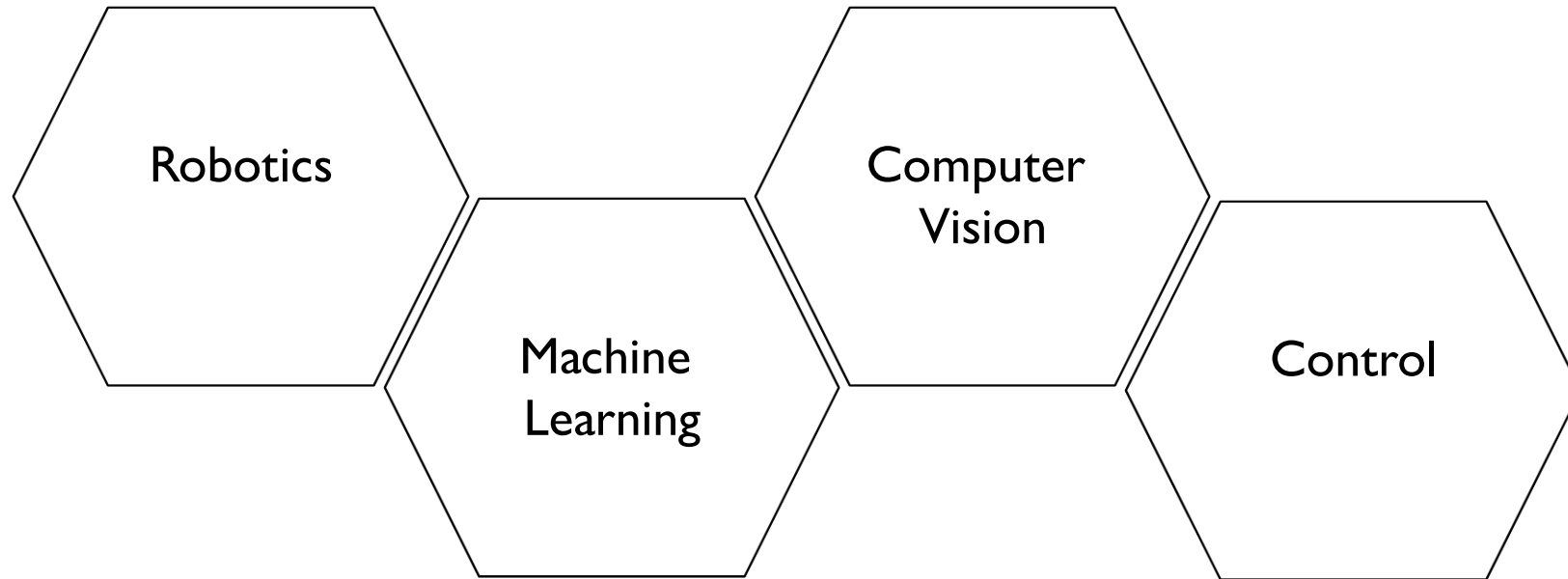# New robotics faculty in CS

Jessica Burgner-Kahrs

Animesh Garg

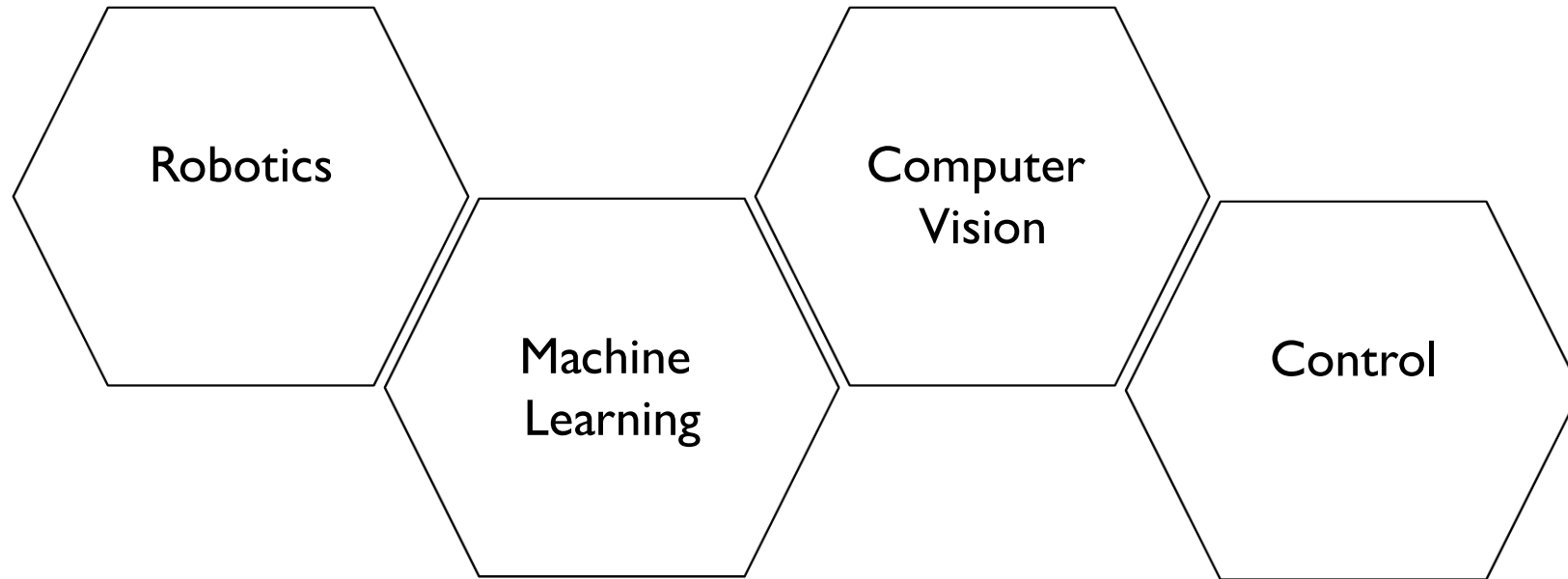Myself

My Research

How can robots learn to perform well alongside humans with only minimal explicit supervision from them?

My Research

Robotics

Machine Learning

Computer Vision

Control

Long term research question

How can robots learn to perform well alongside humans with ~~only minimal~~ explicit supervision from them?
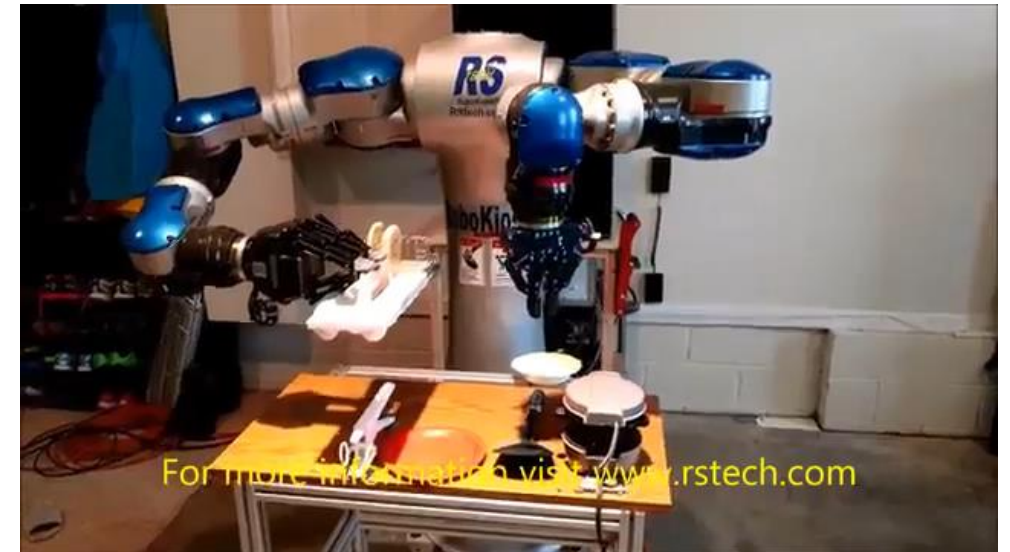
NO

Today's agenda

- Behavioral cloning (supervised learning)
- Imitation learning

Applications

Any control problem where:

- **writing down a dense cost function is difficult**

- there is a hierarchy of interacting decision-making processes

- our engineered solutions might not cover all cases

- unrestricted exploration during learning is slow or dangerous

https://www.youtube.com/watch?v=M8r0gmQXm1Y

Applications

Any control problem where:

- **writing down a dense cost function is difficult**

- there is a hierarchy of interacting decision-making processes

- our engineered solutions might not cover all cases

- unrestricted exploration during learning is slow or dangerous

https://www.youtube.com/watch?v=Q3LXJGha7Ws

Applications

Any control problem where:

- **writing down a dense cost function is difficult**

- there is a hierarchy of interacting decision-making processes

- our engineered solutions might not cover all cases

- unrestricted exploration during learning is slow or dangerous

2017-07-26 22:22:57   --- MPH

PIERDELIER®
DR650GW-2CH/FHD-HD

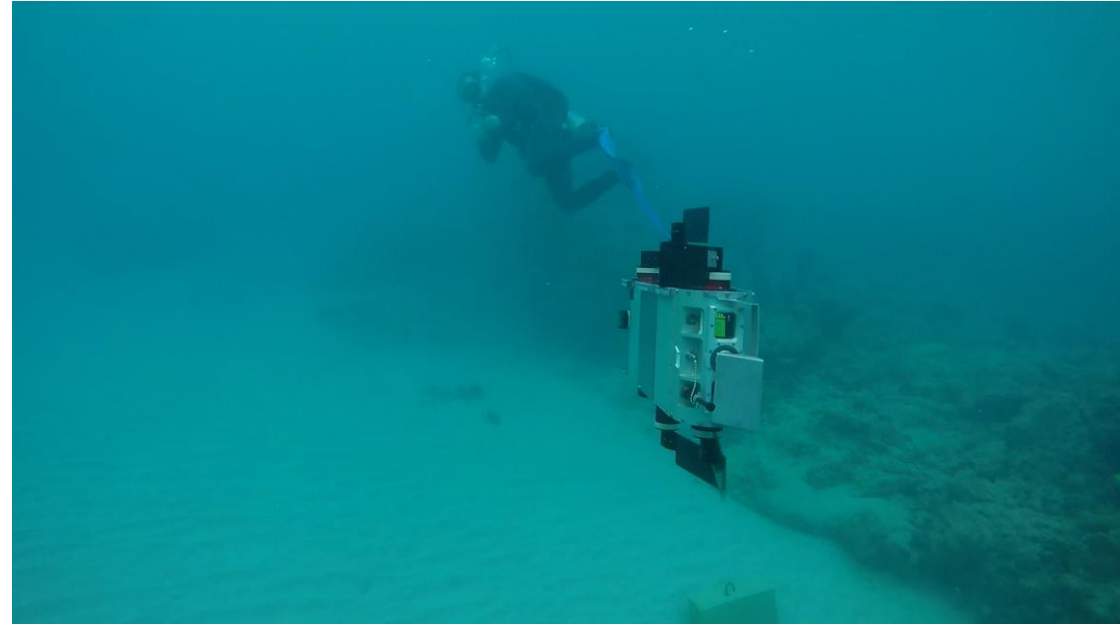https://www.youtube.com/watch?v=RjGe0GiiFzw

Applications

Any control problem where:

- **writing down a dense cost function is difficult**

- there is a hierarchy of interacting decision-making processes

- our engineered solutions might not cover all cases

- unrestricted exploration during learning is slow or dangerous

Robot videographer / documentarian

Applications

Any control problem where:

- **writing down a dense cost function is difficult**

- there is a hierarchy of interacting decision-making processes

- our engineered solutions might not cover all cases

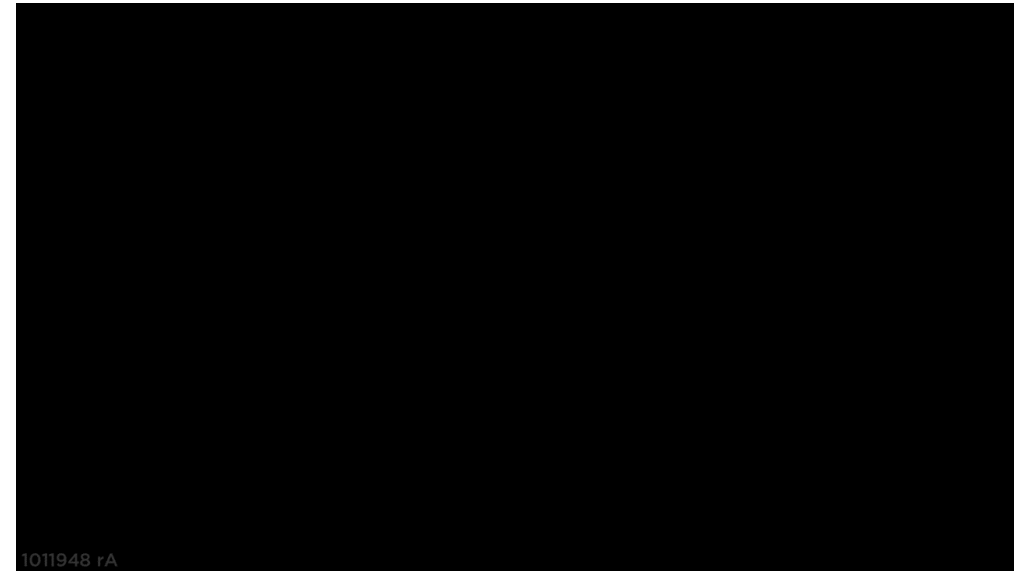- unrestricted exploration during learning is slow or dangerous

Robot explorer

Applications

Any control problem where:

- **writing down a dense cost function is difficult**

- there is a hierarchy of interacting decision-making processes

- our engineered solutions might not cover all cases

- unrestricted exploration during learning is slow or dangerous

https://www.youtube.com/watch?v=0XdC1HUp-rU

Back to the future

https://www.youtube.com/watch?v=2KMAAmkz9go

Navlab 1 (1986-1989)

[Courtesy of Dean Pomerleau]

https://www.youtube.com/watch?v=ilP4aPDTBPE

Navlab 2 + ALVINN (Dean Pomerlau's PhD thesis, 1989-1993)

30 x 32 pixels, 3 layer network, outputs steering command
~5 minutes of training per road type

ALVINN: architecture

30 Output Units

Sharp Left   Straight Ahead   Sharp Right

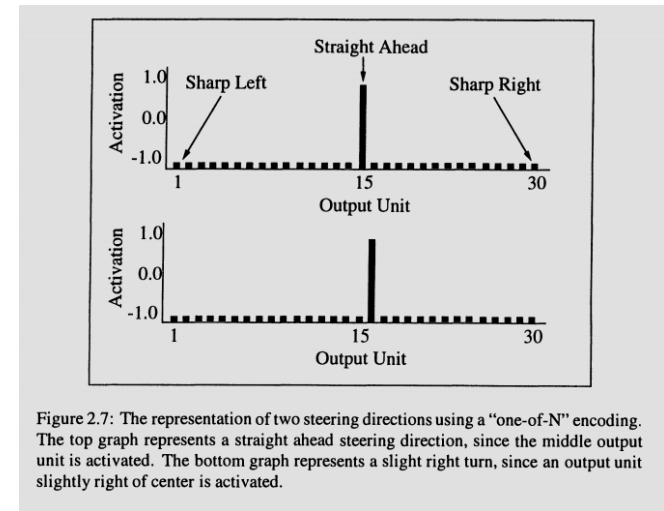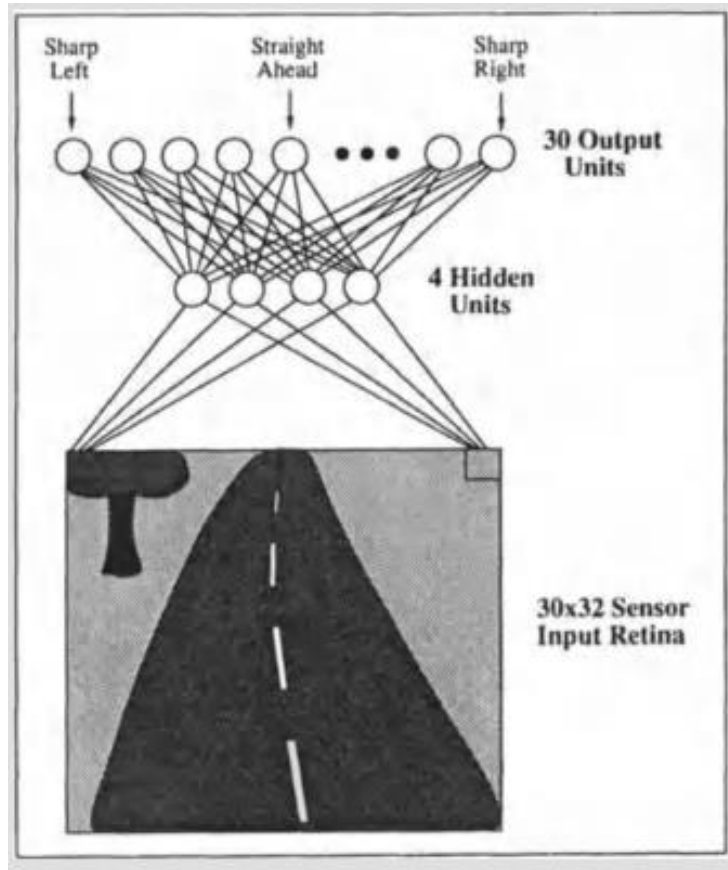4 Hidden Units

30x32 Sensor Input Retina

Figure 2.7: The representation of two steering directions using a "one-of-N" encoding. The top graph represents a straight ahead steering direction, since the middle output unit is activated. The bottom graph represents a slight right turn, since an output unit slightly right of center is activated.
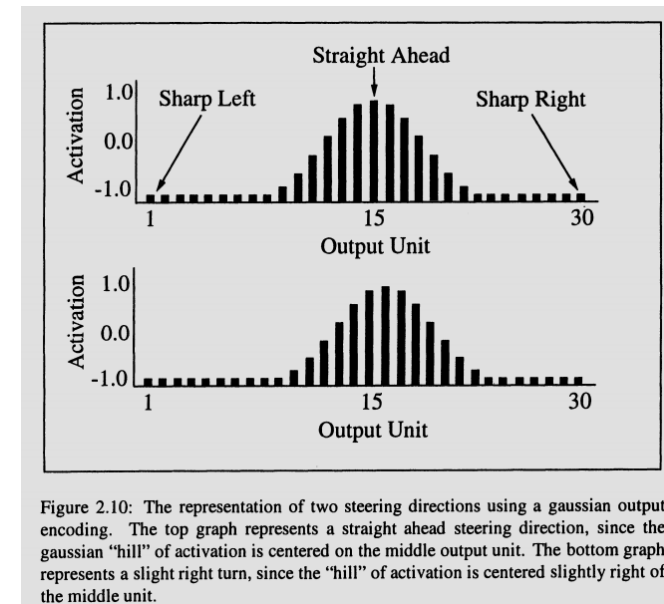

Figure 2.10: The representation of two steering directions using a gaussian output encoding. The top graph represents a straight ahead steering direction, since the gaussian "hill" of activation is centered on the middle output unit. The bottom graph represents a slight right turn, since the "hill" of activation is centered slightly right of the middle unit.

https://drive.google.com/file/d/0Bz9namoRIUKMa0pJYzRGSFVwbm8/view

Dean Pomerlau's PhD thesis

ALVINN: training set

To generate synthetic training data for the task of autonomous road following, I developed a program that generated aerial views of simulated stretches of roads and then used a model of the camera to back-project the aerial map into a 2D image of the road ahead. The simulated road image generator used nearly 200 parameters in order to generate a variety of realistic road images. Some of the most important parameters are listed in Figure 3.1.
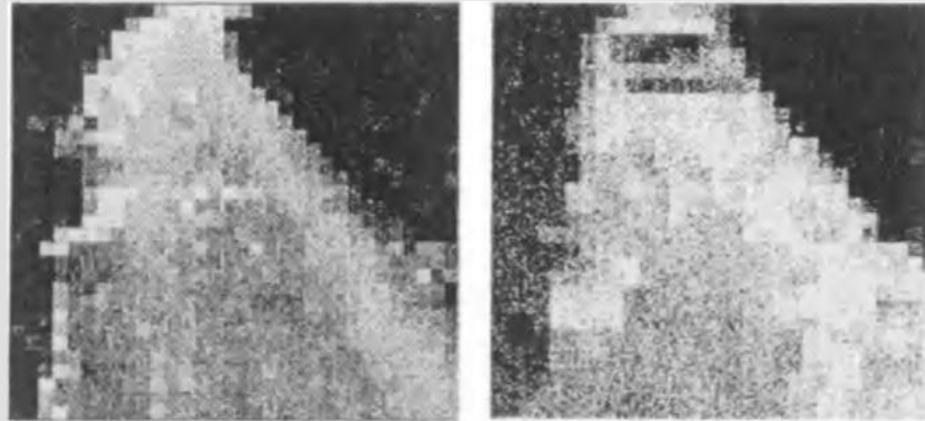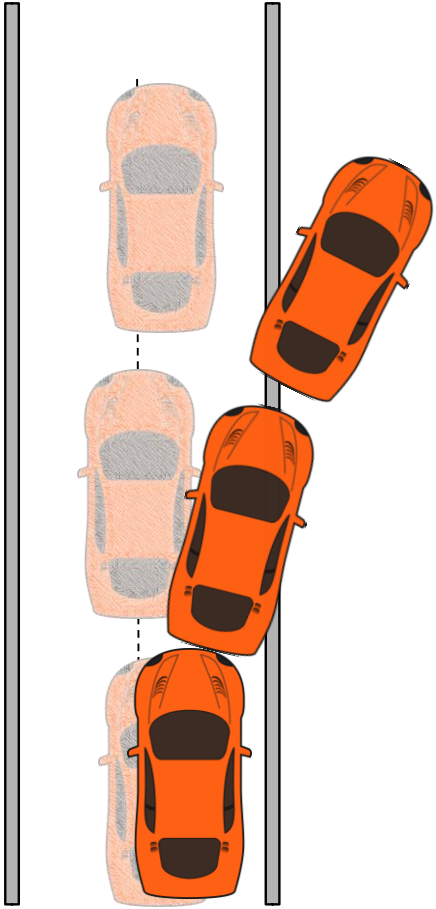

Figure 3.2: A low resolution video image of a single lane road (left) and an artificial single lane road image created by the road image generator (right).

3.2 Training "on-the-fly" with Real Data

Online updates via backpropagation

Problems Identified by Pomerlau

Test distribution is different from training distribution (covariate shift)

the vehicle back to the middle of the road. The second problem is that naively training the network with only the current video image and steering direction may cause it to overlearn recent inputs. If the person drives the Navlab down a stretch of straight road at the end of training, the network will be presented with a long sequence of similar images. This sustained lack of diversity in the training set will cause the network to "forget" what it had learned about driving on curved roads and instead learn to always steer straight ahead.

Catastrophic forgetting
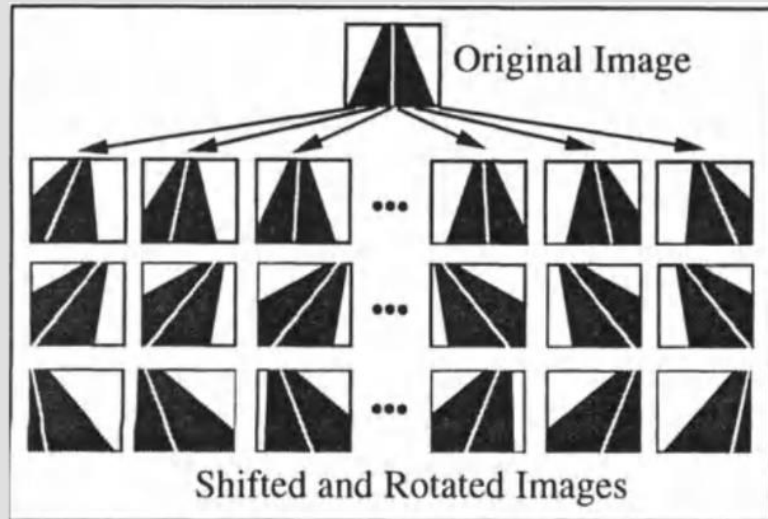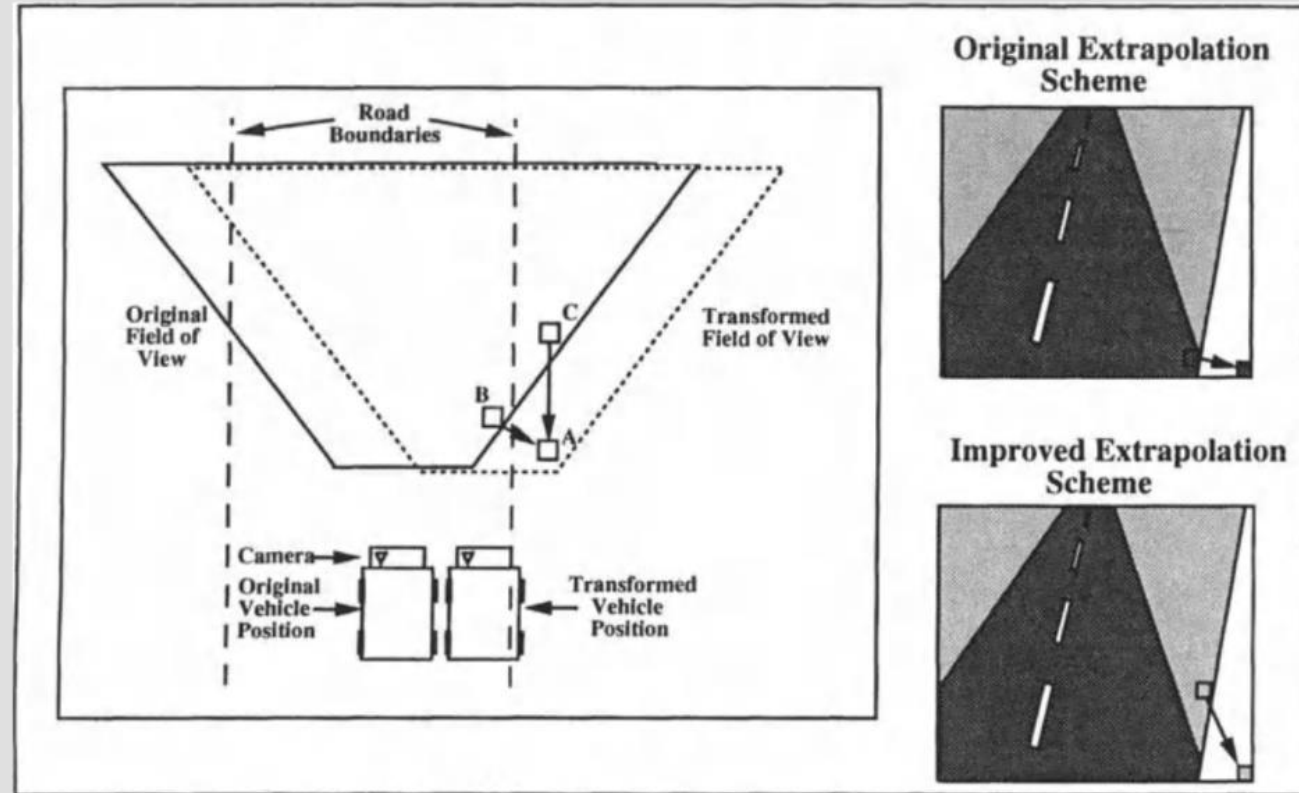
(Partially) Addressing Covariate Shift

Figure 3.4: The single original video image is shifted and rotated to create multiple training exemplars in which the vehicle appears to be at different locations relative to the road.

(Partially) Addressing Catastrophic Forgetting

1. Maintains a buffer of old (image, action) pairs

2. Experiments with different techniques to ensure diversity and avoid outliers

Behavioral Cloning = Supervised Learning

25 years later

March 2016

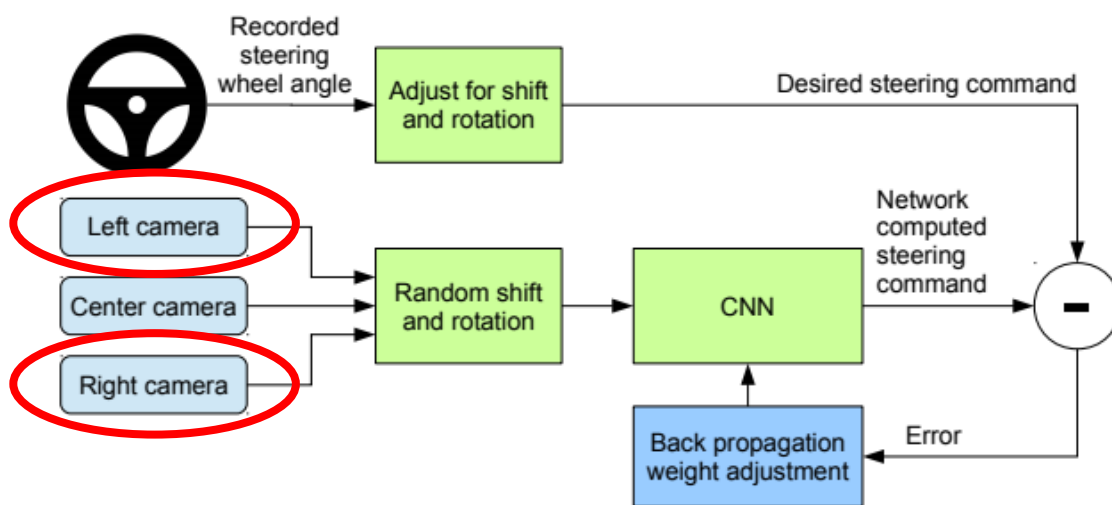https://www.youtube.com/watch?v=qhUvQiKec2U

How much has changed?

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

Figure 2: Training the neural network.

offline

Output: vehicle control
Fully-connected layer — 10 neurons
Fully-connected layer — 50 neurons
Fully-connected layer — 100 neurons
1164 neurons
Flatten
Convolutional feature map 64@1x18
3x3 kernel — Convolutional feature map 64@3x20
3x3 kernel — Convolutional feature map 48@5x22
5x5 kernel — Convolutional feature map 36@14x47
5x5 kernel — Convolutional feature map 24@31x98
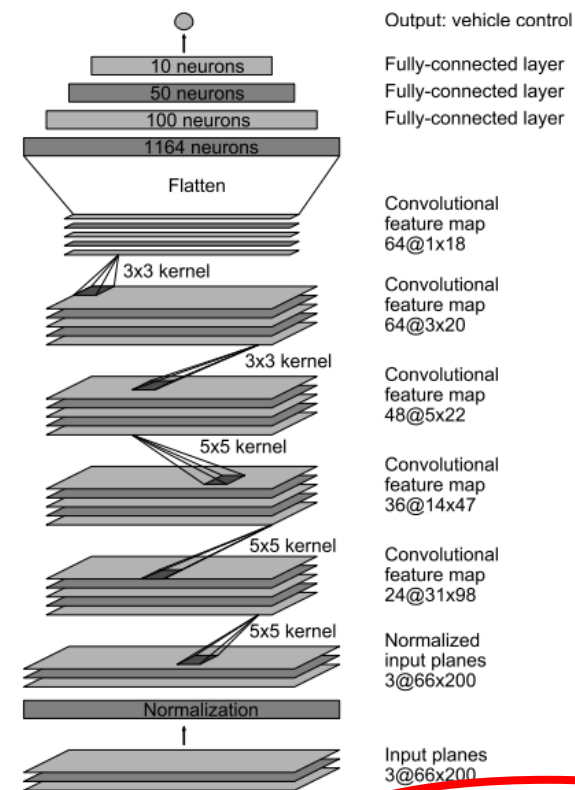5x5 kernel — Normalized input planes 3@66x200
Normalization
Input planes 3@66x200

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

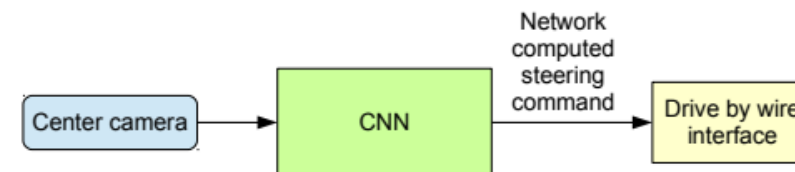Center camera → CNN → Network computed steering command → Drive by wire interface

Figure 3: The trained network is used to generate steering commands from a single front-facing center camera.

End to End Learning for Self-Driving Cars, Bojarski et al, 2016

How much has changed?

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

Figure 2: Training the neural network.

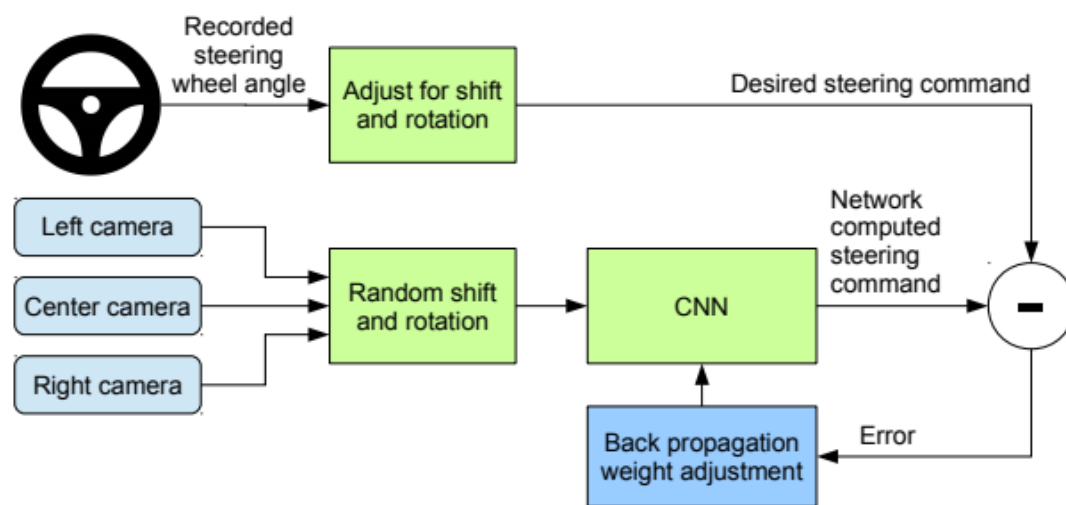Figure 3: The trained network is used to generate steering commands from a single front-facing center camera.

"Our collected data is labeled with road type, weather condition, and the driver's activity (staying in a lane, switching lanes, turning, and so forth)."
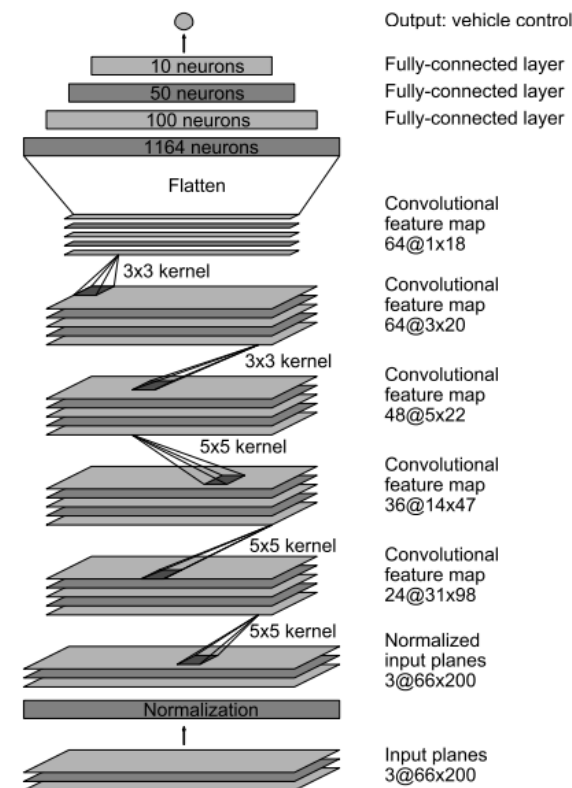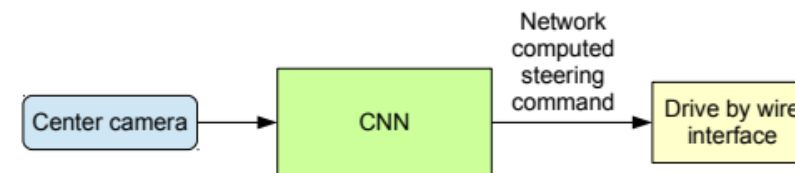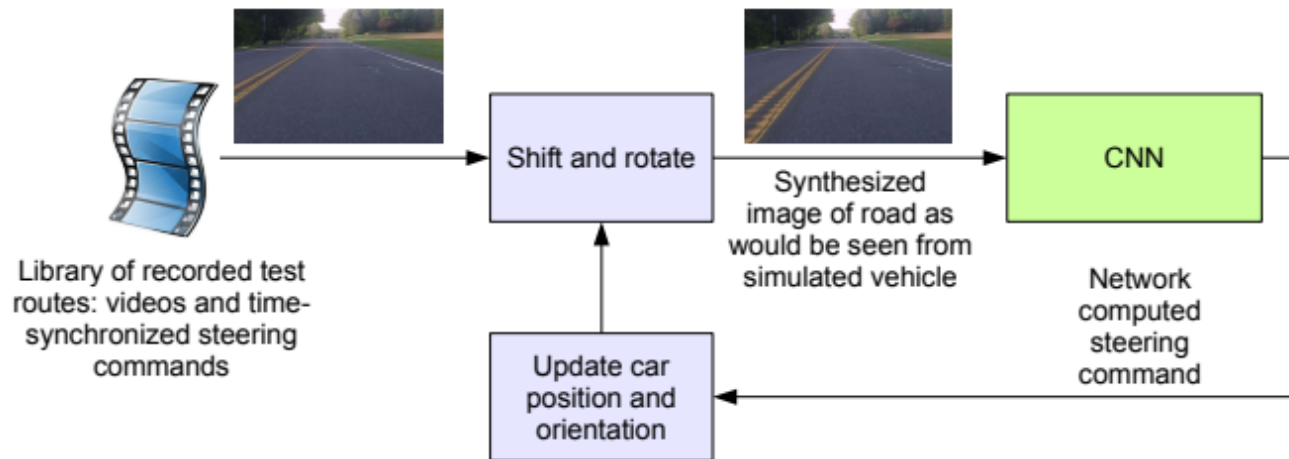
End to End Learning for Self-Driving Cars, Bojarski et al, 2016

How much has changed?

Figure 5: Block-diagram of the drive simulator.

How much has changed?

A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots, Giusti et al., 2016

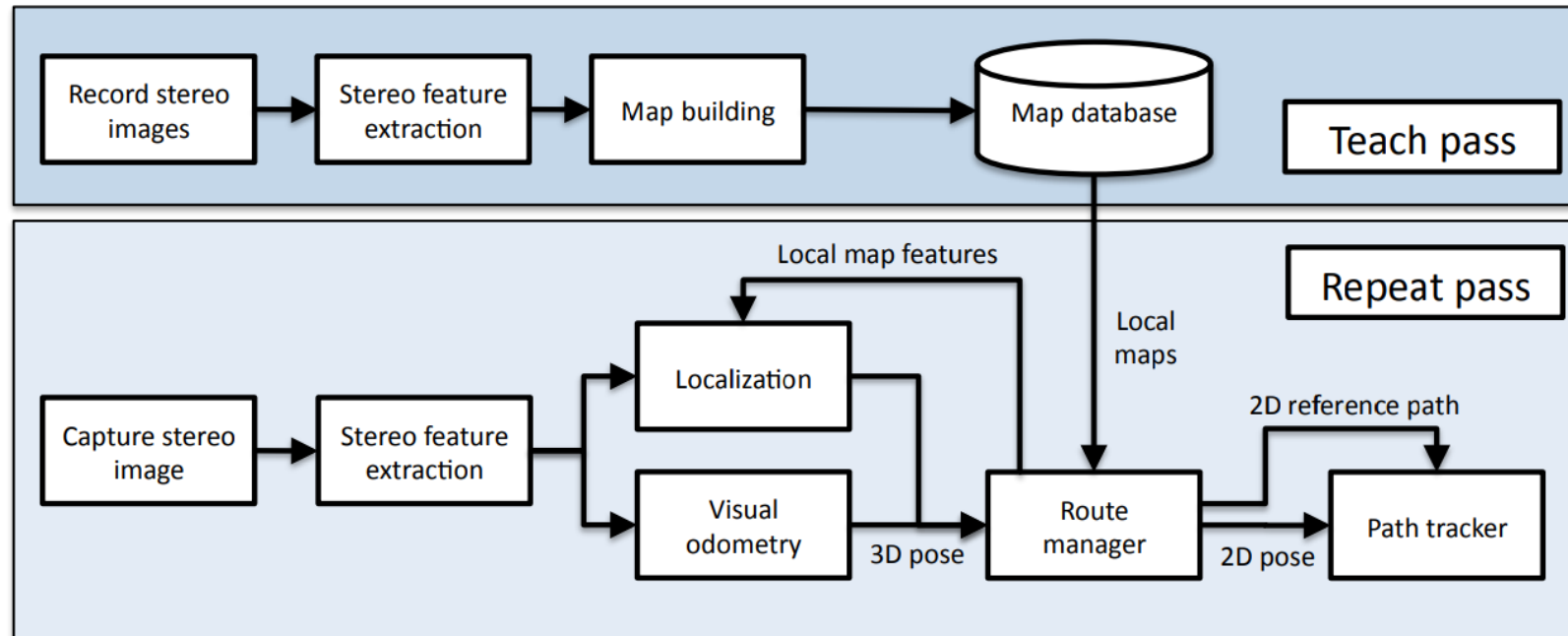https://www.youtube.com/watch?v=umRdt3zGgpU

How much has changed?

Not a lot for learning to do lane following with neural networks.

But, there are a few other beautiful ideas that do not involve end-to-end learning.

Visual Teach & Repeat

Human Operator or
Planning Algorithm

Visual Path Following on a Manifold in Unstructured Three-Dimensional Terrain, Furgale & Barfoot, 2010

Visual Teach & Repeat

Key Idea #1: Manifold Map

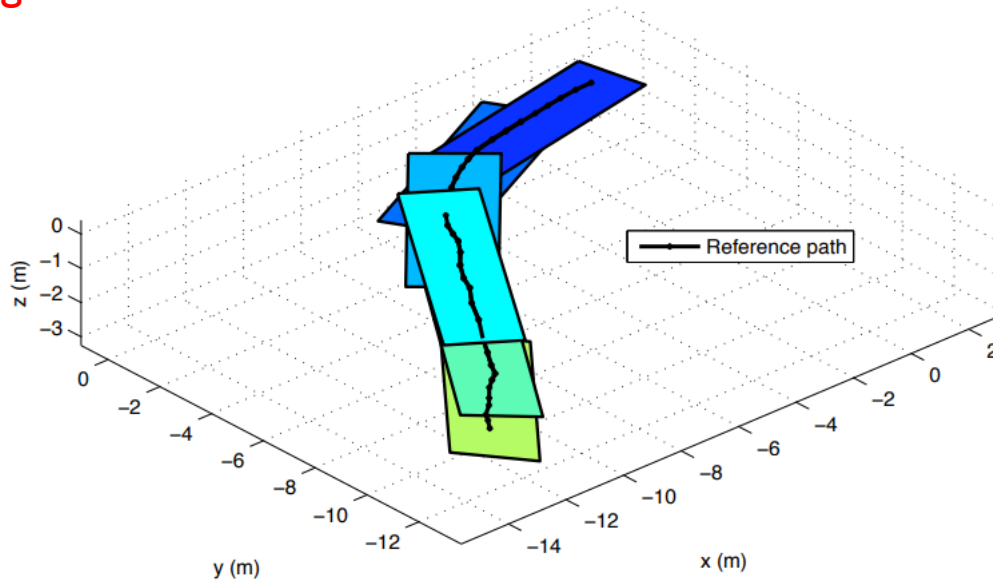Build local maps relative to the
path. No global coordinate frame.

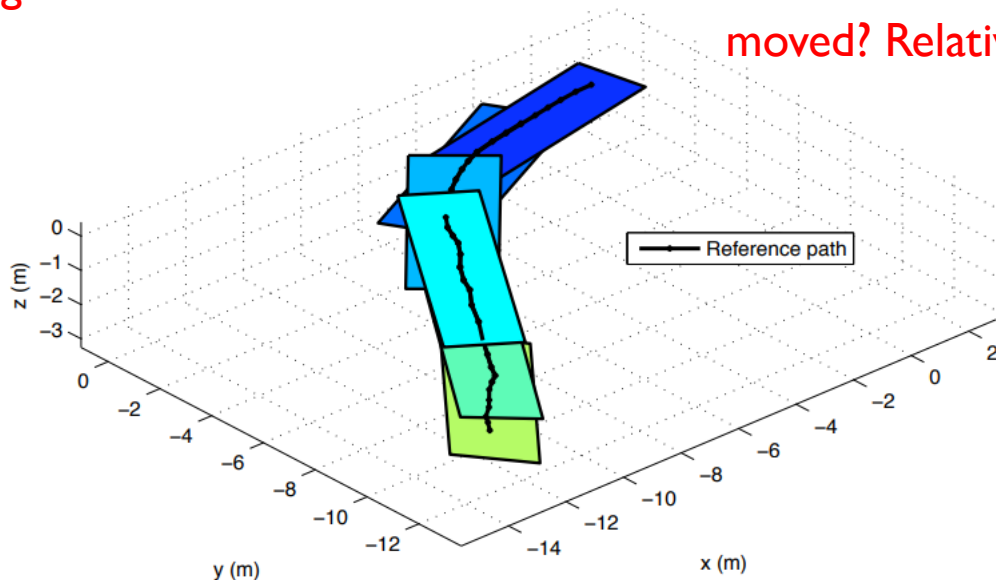Fig. 5. A view of six overlapping submaps with the reference path plotted above.

Visual Path Following on a Manifold in Unstructured Three-Dimensional Terrain, Furgale & Barfoot, 2010

Visual Teach & Repeat

Key Idea #1: Manifold Map

Build local maps relative to the path. No global coordinate frame.

Key Idea #2: Visual Odometry

Given two consecutive images, how much has the camera moved? Relative motion.

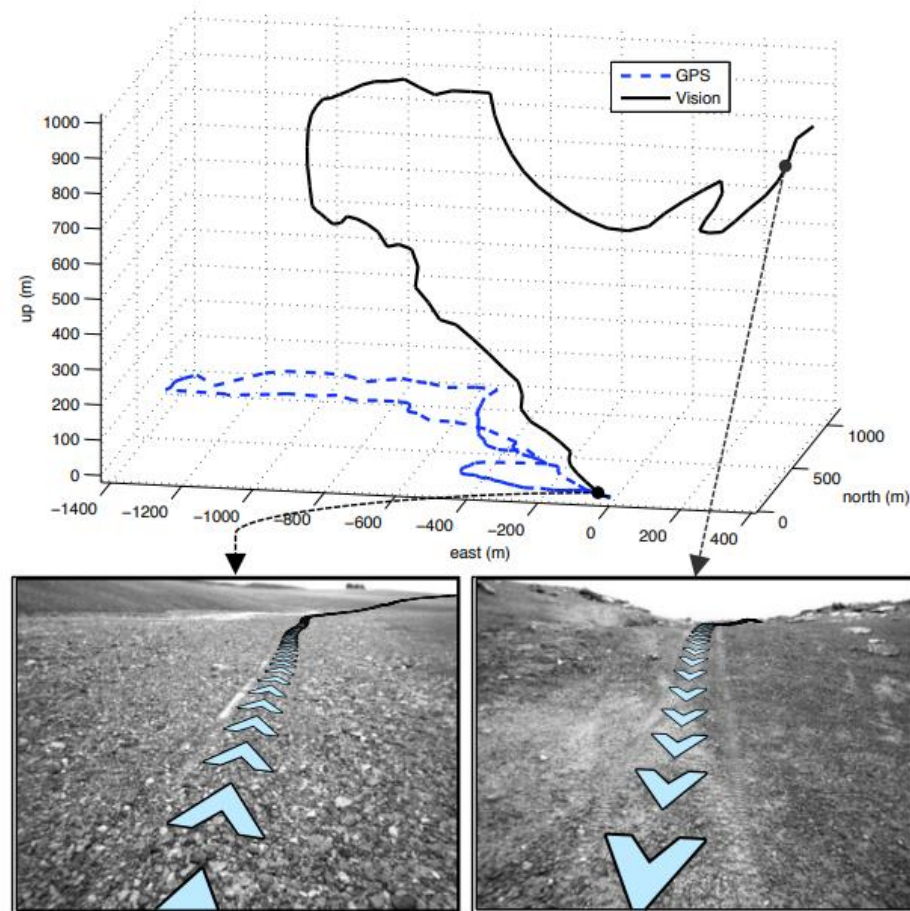Fig. 5. A view of six overlapping submaps with the reference path plotted above.

Fig. 6. The visual reconstruction of a five kilometer rover traverse plotted against GPS (Top). Although the reconstruction is wildly inaccurate at this scale, locally it is good enough to enable retracing of the route. The bottom images show views from either end of the path, with the reference path plotted as a series of chevrons. To the rover, the map is locally Euclidean.

Visual Path Following on a Manifold in Unstructured Three-Dimensional Terrain, Furgale & Barfoot, 2010

Visual Teach & Repeat

https://www.youtube.com/watch?v=_ZdBfU4xJnQ

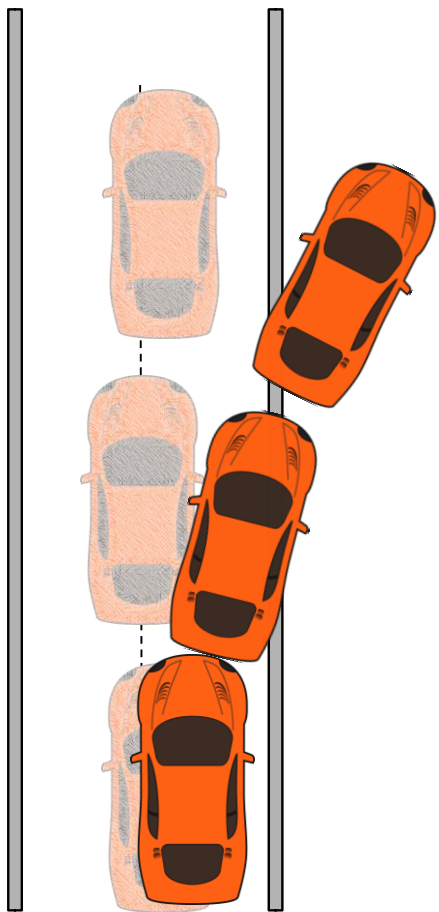https://www.youtube.com/watch?v=9dN0wwXDuqo

Centimeter-level precision in tracking the demonstrated path over kilometers-long trails.

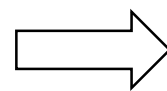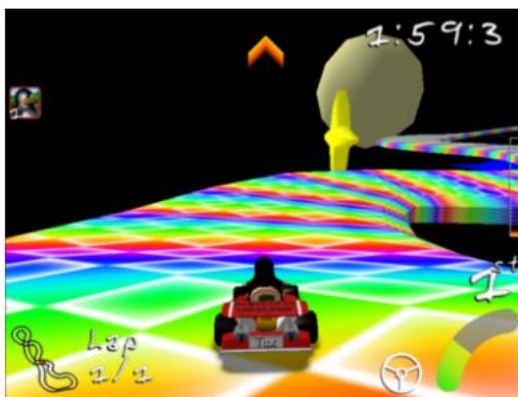# Today's agenda

- Behavioral cloning
- Imitation learning

Back to Pomerlau

(Ross & Bagnell, 2010): How are we sure these errors are not due to overfitting or underfitting?

1. Maybe the network was too small (underfitting)
2. Maybe the dataset was too small and the network overfit it

Steering commands $\pi_\theta(s) = \theta^\top s$ where s are image features

Test distribution is different from training distribution (covariate shift)

Back to Pomerlau

Test distribution is different from training distribution (covariate shift)

(Ross & Bagnell, 2010): How are we sure these errors are not due to overfitting or underfitting?
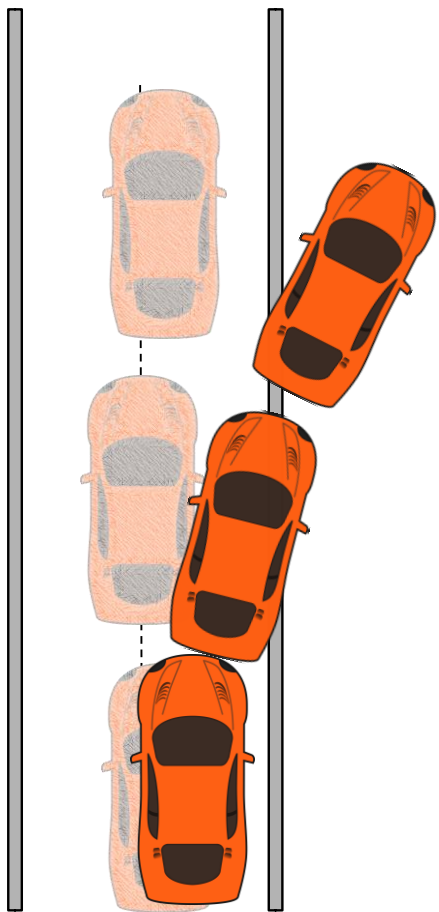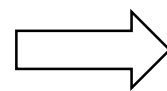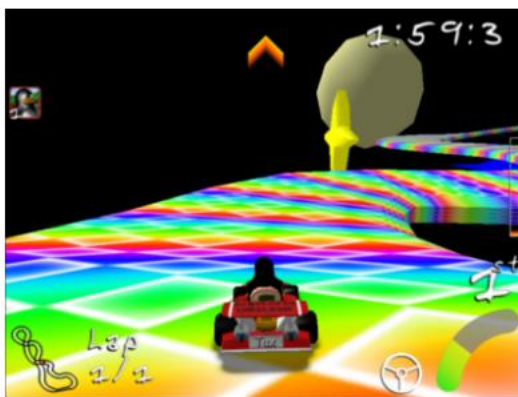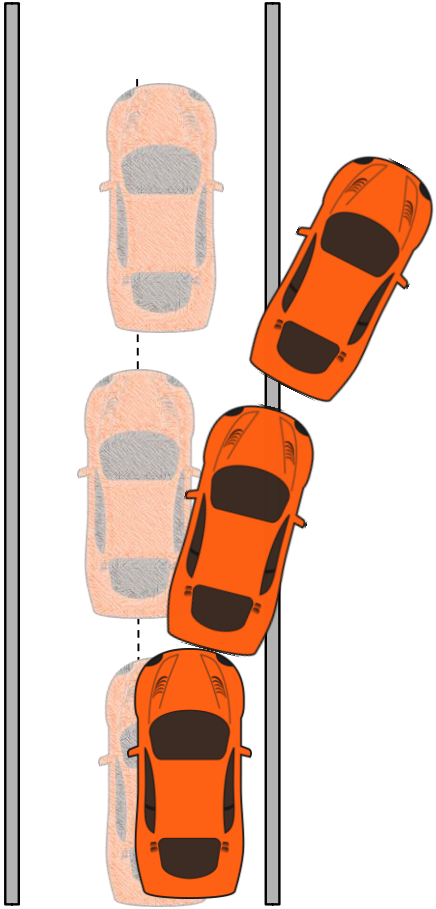
1. Maybe the network was too small (underfitting)
2. Maybe the dataset was too small and the network overfit it

Steering commands $\pi_\theta(s) = \theta^\top s$ where s are image features

It was not 1: they showed that even a linear policy can work well.
It was not 2: their error on held-out data was close to training error.

Imitation learning $\neq$ Supervised learning

(Ross & Bagnell, 2010): IL is a sequential decision-making problem.

- Your actions affect future observations/data.
- This is not the case in supervised learning

Supervised Learning
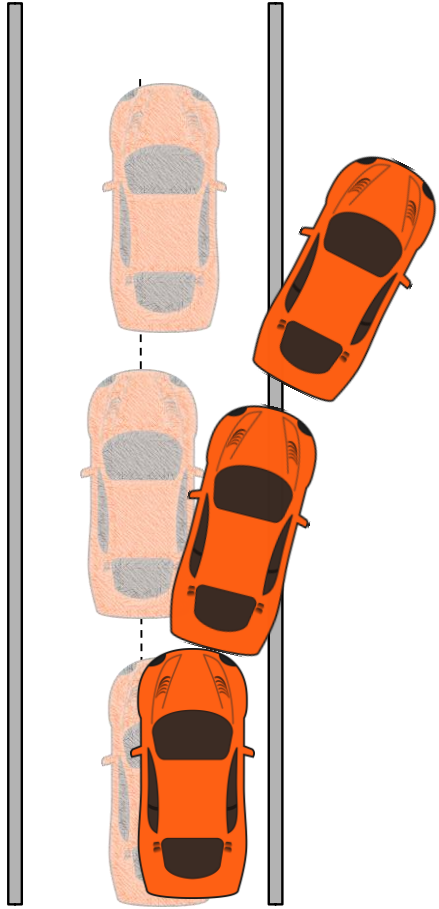
Assumes train/test data are i.i.d.

If expected training error is ϵ
Expected test error after T decisions

$$T\epsilon$$

Errors are independent

Test distribution is different from training distribution (covariate shift)

Imitation learning $\neq$ Supervised learning

(Ross & Bagnell, 2010): IL is a sequential decision-making problem.

- Your actions affect future observations/data.
- This is not the case in supervised learning

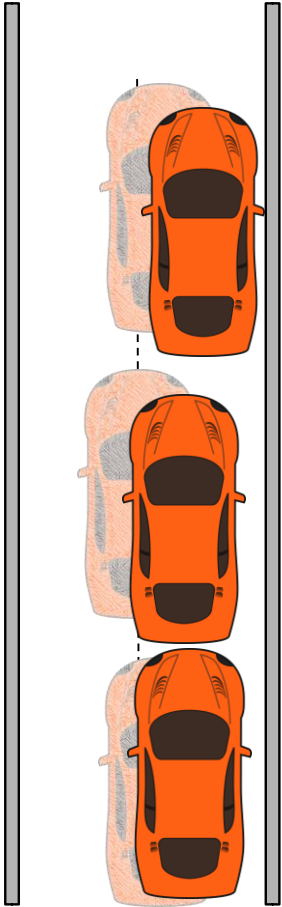| **Imitation Learning** $\longleftarrow$	**Supervised Learning**
Train/test data are not i.i.d.	Assumes train/test data are i.i.d.
If expected training error is ϵ Expected test error after T decisions is up to $$T^2\epsilon$$	If expected training error is ϵ Expected test error after T decisions $$T\epsilon$$
Errors compound	Errors are independent

Test distribution is different from training distribution (covariate shift)

DAgger

(Ross & Gordon & Bagnell, 2011): DAgger, or Dataset Aggregation

- Imitation learning as interactive supervision
- Aggregate training data from expert with test data from execution

---

Algorithm 1 DAgger

1: $D = \{(s, a)\}$ initial expert demonstrations
2: $\theta_1 \leftarrow$ train learner's policy parameters on D
3: **for** $i = 1...N$ **do**
4:      Execute learner's policy π_{θ_i}, get visited states $S_{\theta_i} = \{s_0, ..., s_T\}$
5:      Query the expert at those states to get actions $A = \{a_0, ..., a_T\}$
6:      Aggregate dataset $D = D \cup \{(s, a) \mid s \in S_{\theta_i}, \ a \in A\}$
7:      Train learner's policy $\pi_{\theta_{i+1}}$ on dataset D
8: Return one of the policies π_{θ_i} that performs best on validation set

---

DAgger

(Ross & Gordon & Bagnell, 2011): DAgger, or Dataset Aggregation

- Imitation learning as interactive supervision
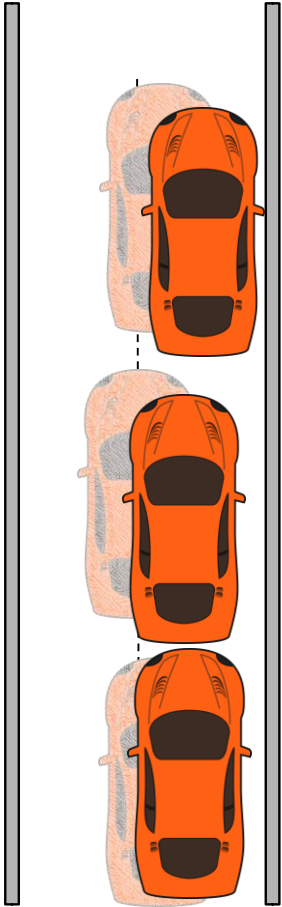- Aggregate training data from expert with test data from execution

Imitation Learning via DAgger

Train/test data are not i.i.d.

If expected training error on aggr. dataset is ϵ
Expected test error after T decisions is

$$O(T\epsilon)$$

Errors do not compound

Supervised Learning

Assumes train/test data are i.i.d.

If expected training error is ϵ
Expected test error after T decisions

$$T\epsilon$$

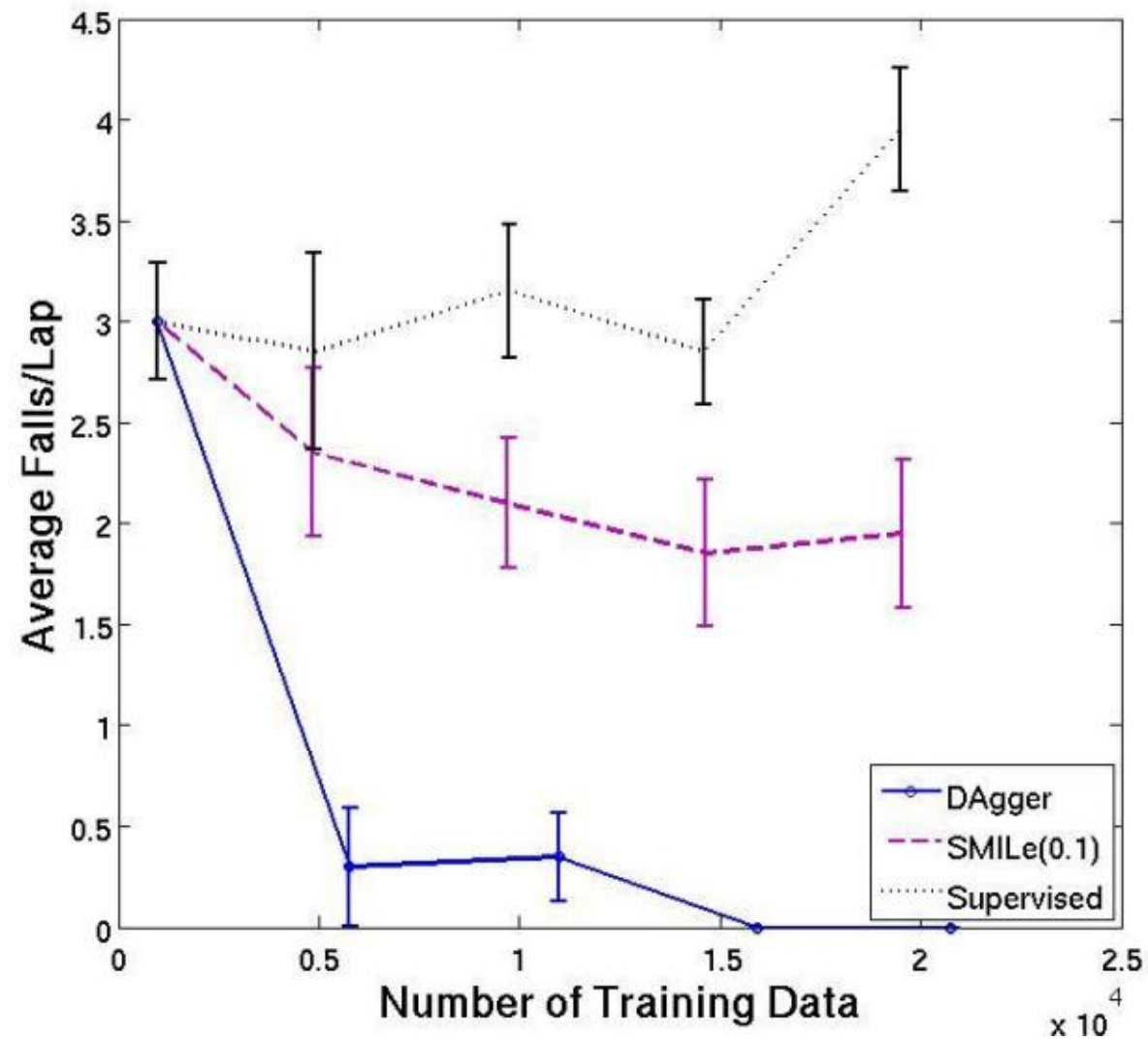Errors are independent

DAgger

Initial expert trajectories

Supervised learning

DAgger

https://www.youtube.com/watch?v=V00npNnWzSU

DAgger

DAgger

Q: Any drawbacks of using it in a robotics setting?

DAgger

The system observes a human expert pilot the drone

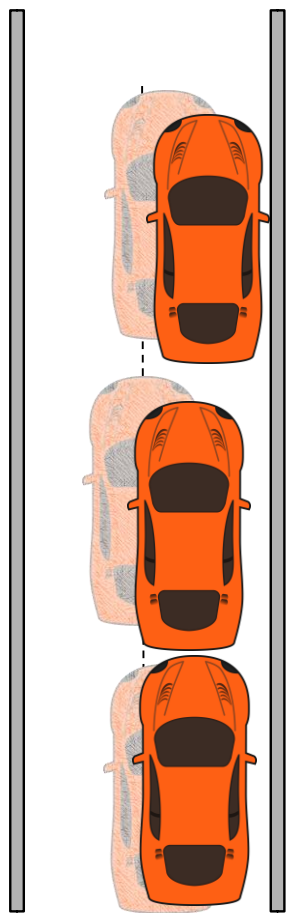https://www.youtube.com/watch?v=hNsP6-K3Hn4

Learning Monocular Reactive UAV Control in Cluttered Natural Environments, Ross et al, 2013

Imitation learning $\neq$ Supervised learning

Any questions?

DAgger: Assumptions for theoretical guarantees

(Ross & Gordon & Bagnell, 2011): DAgger, or Dataset Aggregation

- Imitation learning as interactive supervision
- Aggregate training data from expert with test data from execution

Imitation Learning via DAgger

Train/test data are not i.i.d.

If expected training error on aggr. dataset is ϵ
Expected test error after T decisions is

$$O(T\epsilon)$$

Errors do not compound

Supervised Learning

Assumes train/test data are i.i.d.

If expected training error is ϵ
Expected test error after T decisions

$$T\epsilon$$

Errors are independent

Strongly convex loss
No-regret online learner

Appendix: No-Regret Online Learners

Intuition: No matter what the distribution of input data, your online policy/classifier will do asymptotically as well as the best-in-hindsight policy/classifier.

$$r_N = \frac{1}{N} \sum_{i=1}^{N} L_i(\theta_i) - \min_{\theta \in \Theta} \left[\frac{1}{N} \sum_{i=1}^{N} L_i(\theta) \right]$$

Policy has access to data up to round i

Policy has access to data up to round N

No-regret: $\lim_{N \to \infty} r_N = 0$