Self-Driving Cars

Lecture 3 - Direct Perception

Prof. Dr.-Ing. Andreas Geiger

Autonomous Vision Group
MPI-IS / University of Tübingen

November 8, 2018

University of Tübingen
MPI for Intelligent Systems

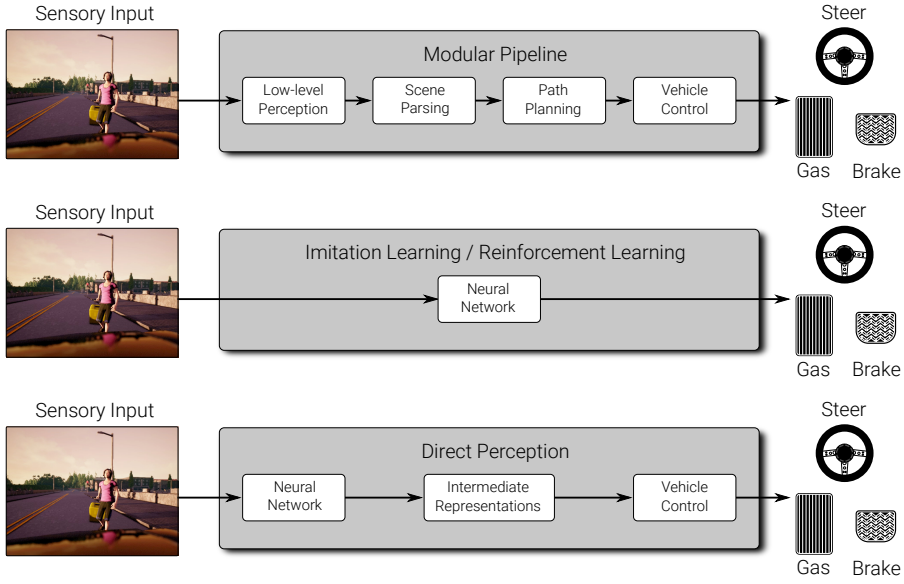
Autonomous Vision Group

Agenda

Date	Lecture (Thursday)	Date	Exercise (Friday)
18.10.	01 - Introduction to Self-Driving Cars	19.10.	00 - Introduction Pytorch & OpenAI Gym
25.10.	02 - DNNs, ConvNets, Imitation Learning	26.10.	01 - Intro: Imitation Learning
1.11.	none (Allerheiligen)	2.11.	
8.11.	03 - Direct Perception	9.11.	01 - Q&A
15.11.	none (CVPR Deadline)	16.11.	
22.11.	04 - Reinforcement Learning	23.11.	01 - Discussion & 02 - Intro: Reinforcement Learning
29.11.	05 - Vehicle Dynamics & Control	30.11.	
6.12.	06 - Localization & Visual Odometry	7.12.	02 - Q&A
13.12.	07 - Simultaneous Localization and Mapping (J. Stückler)	14.12.	
20.12.	08 - Road and Lane Detection	21.12.	02 - Discussion & 03 - Intro: Modular Pipeline
10.1.	09 - Reconstruction and Motion Estimation	11.1.	
17.1.	10 - Object Detection & Tracking	18.1.	
24.1.	11 - Scene Understanding	25.1.	03 - Q&A
31.1.	12 - Planning	1.2.	03 - Discussion & Announcement of Winners
7.2.	13 - Winner's Presentations and Exam Q&A	8.2.	

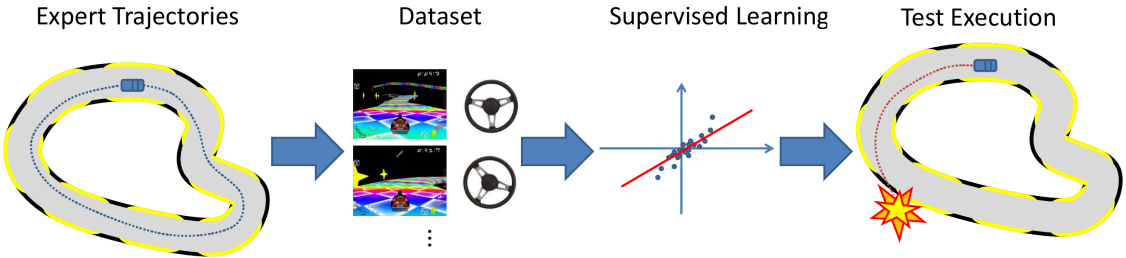
Recap

Approaches to Self-Driving



Imitation Learning

Imitation Learning in a Nutshell



Hard coding policies is often difficult ⇒ Rather use a data-driven approach!

- **Given:** demonstrations or demonstrator
- **Goal:** train a policy to mimic decision
- **Variants:** behavior cloning, inverse optimal control, …

Formal Definition of Imitation Learning

General Imitation Learning:

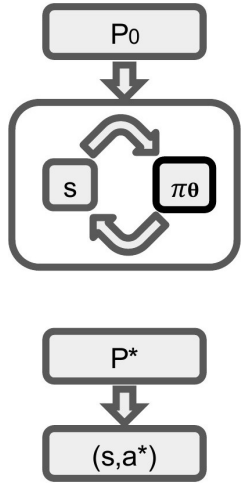
$$\underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{s\sim P(s|\pi_\theta)}\left[\mathcal{L}\left(\pi^*(s), \pi_\theta(s)\right)\right]$$

- State distribution $P(s|\pi_\theta)$ depends on rollout determined by current policy π_θ

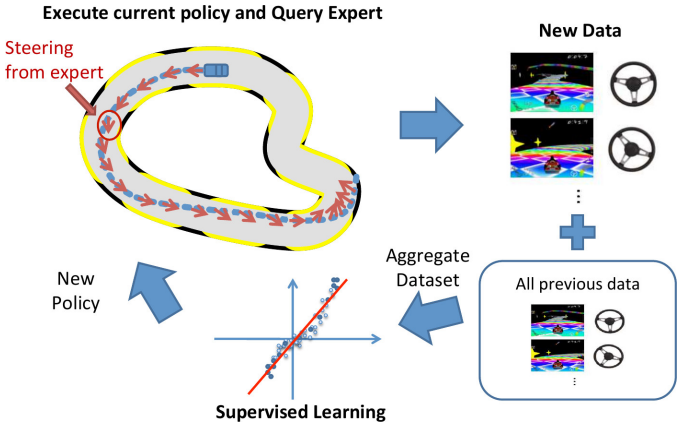
Behavior Cloning:

$$\underset{\theta}{\operatorname{argmin}}\ \underbrace{\mathbb{E}_{(s^*,a^*)\sim P^*}\left[\mathcal{L}\left(a^*, \pi_\theta(s^*)\right)\right]}_{=\sum_{i=1}^{N}\mathcal{L}\left(a_i^*, \pi_\theta(s_i^*)\right)}$$

- State distribution P^* provided by expert
- Reduces to supervised learning problem



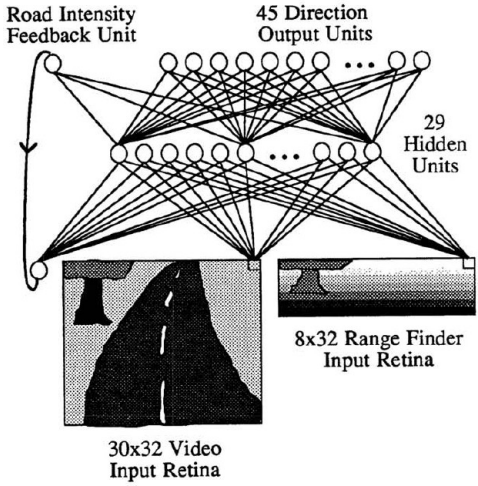
DAGGER: Dataset Aggregation



- **Idea:** Iteratively build a set of inputs that the final policy is likely to encounter based on previous experience. Query expert to obtain policy at new states.

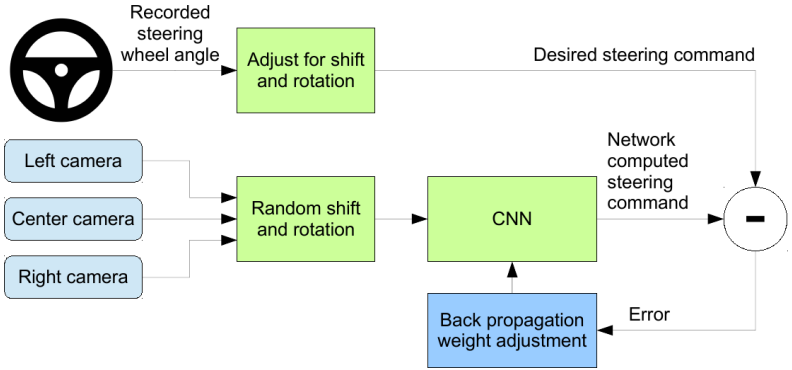
ALVINN: An Autonomous Land Vehicle in a Neural Network

- Fully connected 3 layer neural net
- 36k parameters
- Maps road images to turn radius
- Directions discretized (45 bins)
- Trained on simulated road images!
- Tested on unlined paths, lined city streets and interstate highways
- 90 consecutive miles at up to 70 mph



Pomerleau: ALVINN: An Autonomous Land Vehicle in a Neural Network. NIPS, 1988.

PilotNet: System Overview

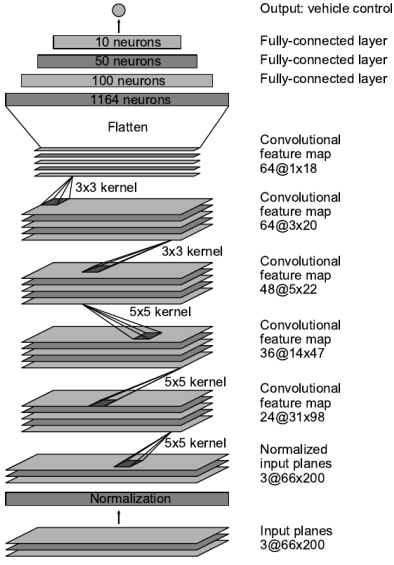


- Steering command represented as $1/r$ where r = turning radius in meters
- Data augmentation by 3 cameras and virtually shifted / rotated images assuming the world is flat (homography), adjusting the steering angle appropriately

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

PilotNet: Architecture

- Convolutional network (250k param)
- Input: YUV image representation
- 1 Normalization layer
  - Not learned
- 5 Convolutional Layers
  - 3 strided 5x5
  - 2 non-strided 3x3
- 3 Fully connected Layers
- Output: turning radius
- No lateral control!
- Trained on 72h of driving



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

PilotNet: Feature Visualization

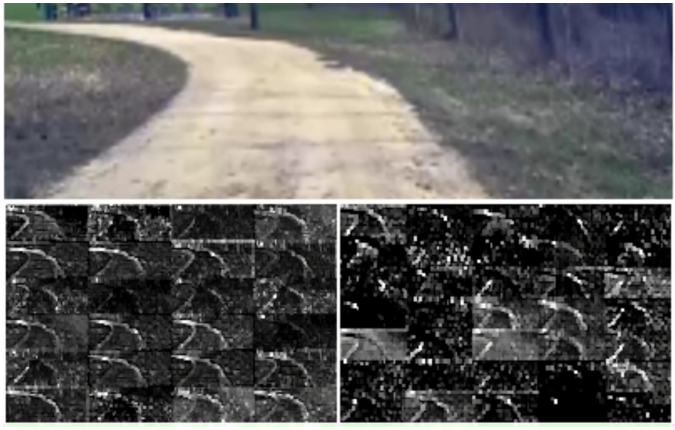


Figure 7: How the CNN “sees” an unpaved road. Top: subset of the camera image sent to the CNN. Bottom left: Activation of the first layer feature maps. Bottom right: Activation of the second layer feature maps. This demonstrates that the CNN learned to detect useful road features on its own, i. e., with only the human steering angle as training signal. We never explicitly trained it to detect the outlines of roads.

PilotNet: Feature Visualization

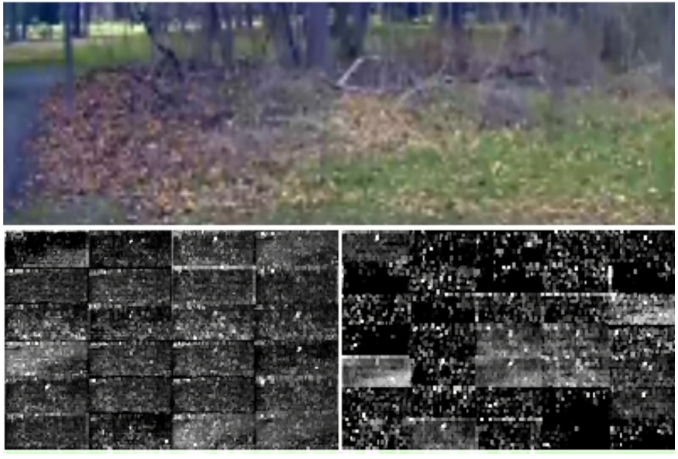
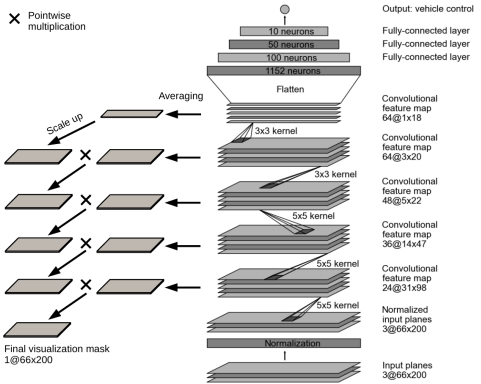
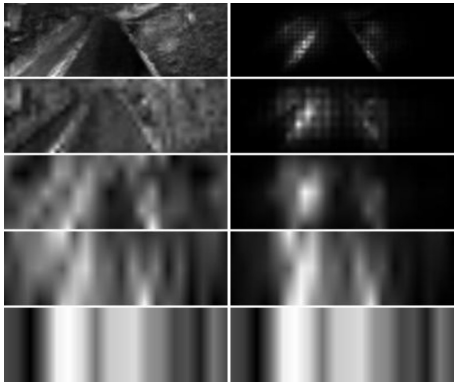


Figure 8: Example image with no road. The activations of the first two feature maps appear to contain mostly noise, i. e., the CNN doesn't recognize any useful features in this image.

Is this surprising?

VisualBackProp





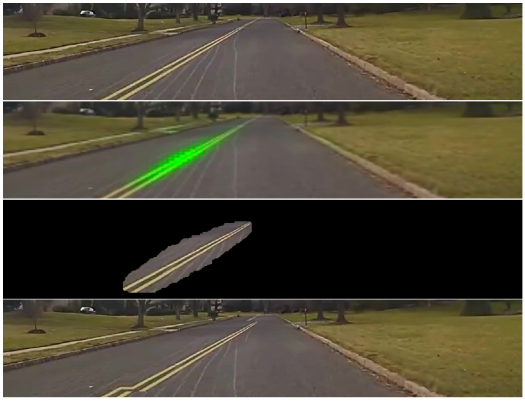
- Central idea: find **salient image regions** that lead to high activations
- Forward pass, then iteratively scale-up activations (faster than gradient methods)

VisualBackProp

Bojarski et al.: VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving. ICRA, 2018.

How to verify that the salient regions are relevant?

VisualBackProp



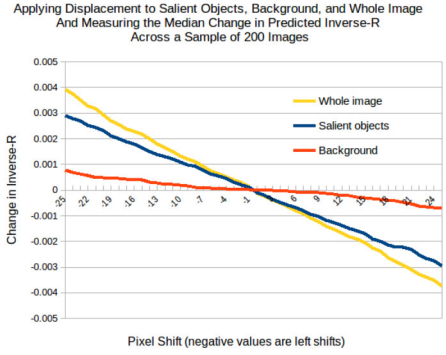


Figure 7: Images used in experiments to show the effect of image-shifts on steer angle.

Verification Criterion:

- Shift in salient objects affects predicted turn radius more than non-salient objects

When does behavior cloning fail?

[Blackboard]

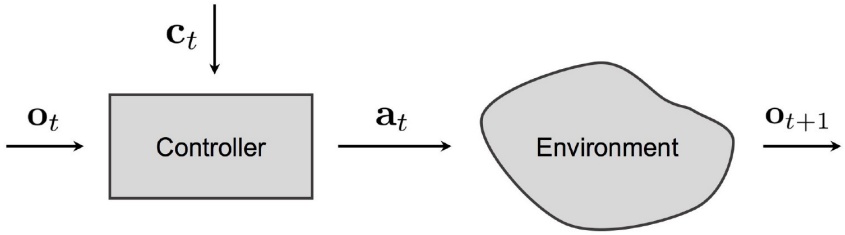
Conditional Imitation Learning

Conditional Imitation Learning

(a) Aerial view of test environment (b) Vision-based driving, view from onboard camera (c) Side view of vehicle

Codevilla, Müller, López, Koltun and Dosovitskiy: End-to-End Driving Via Conditional Imitation Learning. ICRA, 2018.

Conditional Imitation Learning



Idea:

- Condition controller on **navigation command** $c \in \{left, right, straight\}$
- High-level navigation command can be provided by consumer GPS, i.e., telling the vehicle to **turn left/right** or go **straight** at the next intersection
- This removes the task ambiguity induced by the environment (not: noise)
- Observation o_t: current image Action a_t: steering angle & acceleration

Codevilla, Müller, López, Koltun and Dosovitskiy: End-to-End Driving Via Conditional Imitation Learning. ICRA, 2018.

Comparison to Behavior Cloning

Behavior Cloning:

- Training Set:

$$\mathcal{D} = \{(a_i^*, s_i^*)_{i=1}^N\}$$

- Objective:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}\left(a_i^*, \pi_\theta(s_i^*)\right)$$

- Assumption:

$$\exists f(\cdot) : a_i = f(s_i)$$

Often violated in practice! Why?

Conditional Imitation Learning:

- Training Set:

$$\mathcal{D} = \{(a_i^*, s_i^*, c_i^*)_{i=1}^N\}$$

- Objective:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}\left(a_i^*, \pi_\theta(s_i^*, c_i^*)\right)$$

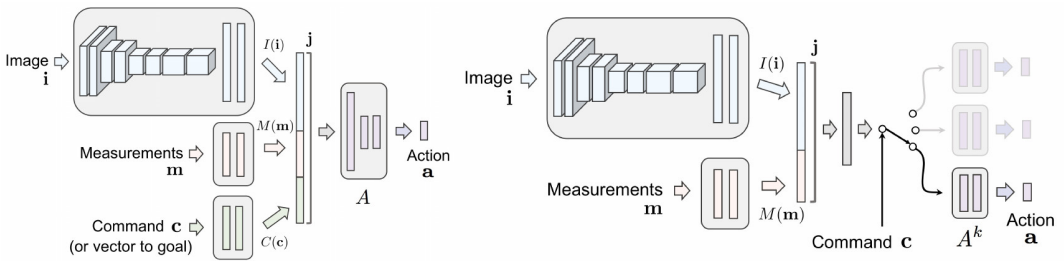
- Assumption:

$$\exists f(\cdot, \cdot) : a_i = f(s_i, c_i)$$

Better approximation!

Codevilla, Müller, López, Koltun and Dosovitskiy: End-to-End Driving Via Conditional Imitation Learning. ICRA, 2018.

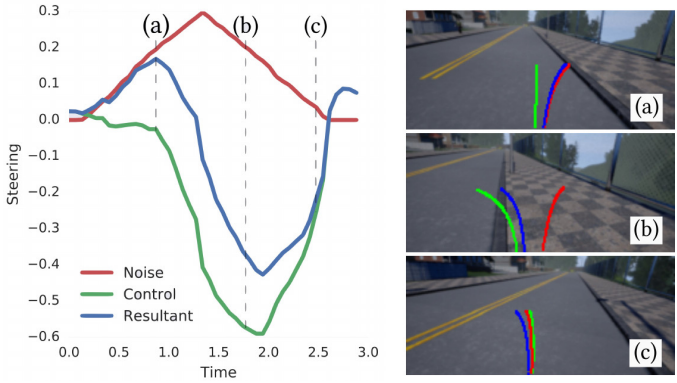
Conditional Imitation Learning: Network Architecture



- This paper proposes two network architectures:
  - (a) Extract features $C(c)$ and concatenate with image features $I(\mathbf{i})$
  - (b) Command c acts as switch between specialized submodules
- Measurements m capture additional information (here: speed of vehicle)

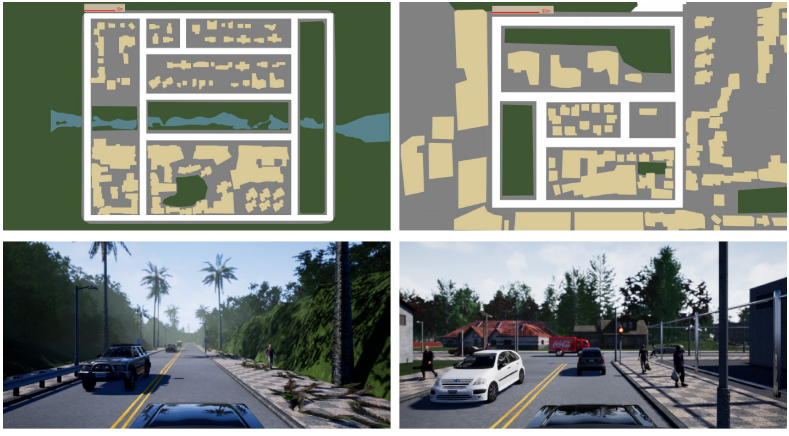
Does CIL also suffer from catastrophic drift?

Conditional Imitation Learning: Noise Injection



- Temporally correlated noise injected into trajectories ⇒ drift (only 12 minutes)
- Record driver's (=expert's) corrective response ⇒ recover from drift

CARLA Simulator



Town 1 (training) Town 2 (testing)

http://www.carla.org

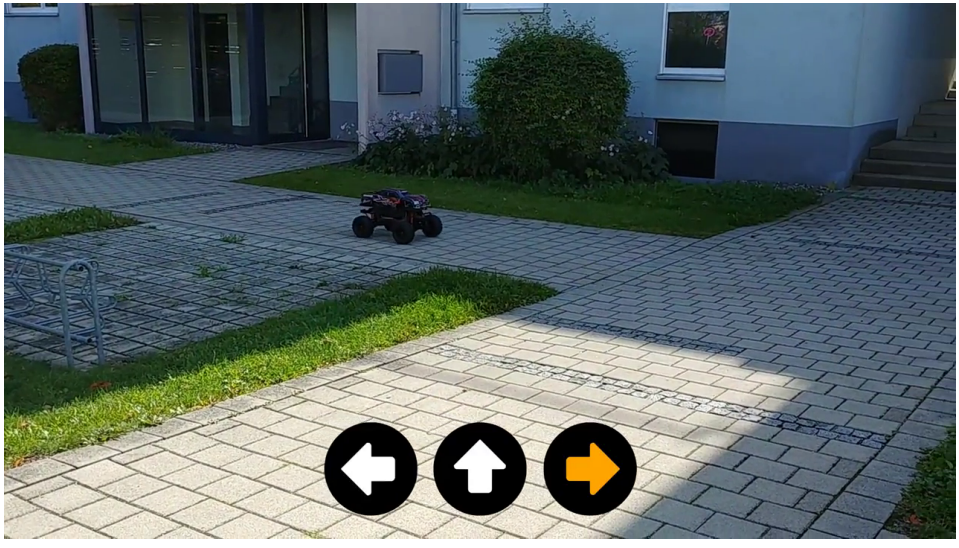
Conditional Imitation Learning: Results

Model	Success rate		Km per infraction	
	Town 1	Town 2	Town 1	Town 2
Non-conditional	20%	26%	5.76	0.89
Goal-conditional	24%	30%	1.87	1.22
Ours branched	**88%**	**64%**	2.34	1.18
Ours cmd. input	78%	52%	3.97	1.30
Ours no noise	56%	22%	1.31	0.54
Ours no aug.	80%	0%	4.03	0.36
Ours shallow net	46%	14%	0.96	0.42

Methods:

- Goal-conditional: provided with vector to goal (i.e., compass)
- Ours branched: Network architecture (b)
- Ours cmd. input: Network architecture (a)
- Ours no noise: no noise injected into trajectories (see previous slide)
- Ours no aug.: no augmentation (contrast, brightness, blur, noise, region dropout)

Conditional Imitation Learning: Results



Codevilla, Müller, López, Koltun and Dosovitskiy: End-to-End Driving Via Conditional Imitation Learning. ICRA, 2018.

Imitation Learning: Summary

Advantages of Imitation Learning:

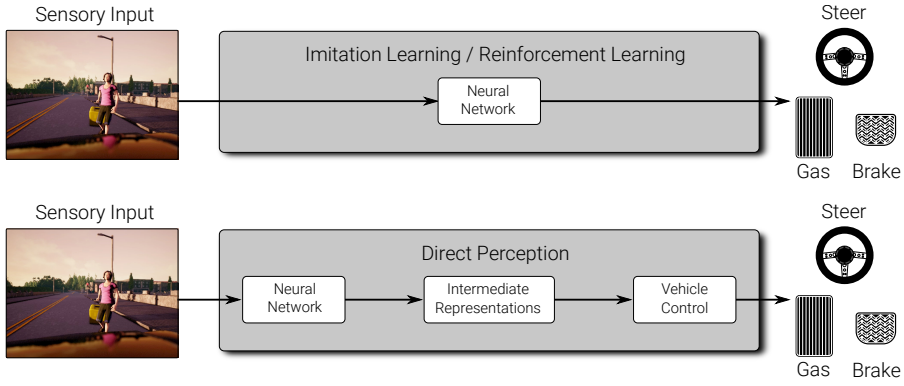
- Easy to implement
- Cheap annotations (just driving while recording images and actions)
- Entire model trained end-to-end
- Conditioning removes ambiguity at intersections

Challenges for Imitation Learning?

- Behavior cloning uses IID assumption which is violated in practice
- Direct mapping from images to control $\Rightarrow$ No long term planning
- No memory (can't remember speed signs, etc.)
- Mapping is difficult to interpret ("black box"), despite visualization techniques
- Not trained on the true task loss ("driving")


Direct Perception

Direct Perception

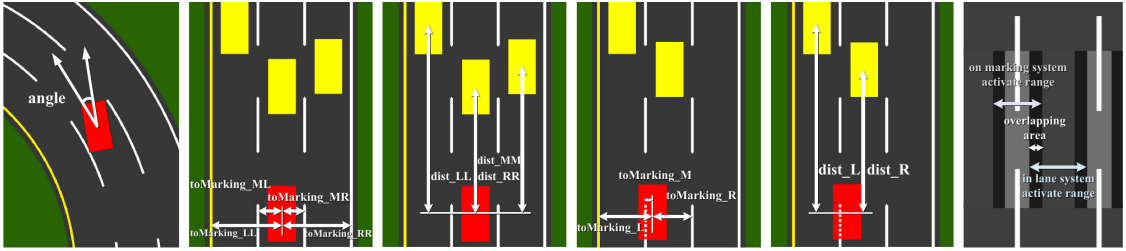


Idea of Direct Perception:

- Learn to predict interpretable low-dimensional intermediate representation
- Exploit classical controllers / finite state machines
- Hybrid model between imitation learning and modular pipelines

Direct Perception for Autonomous Driving

Direct Perception for Autonomous Driving



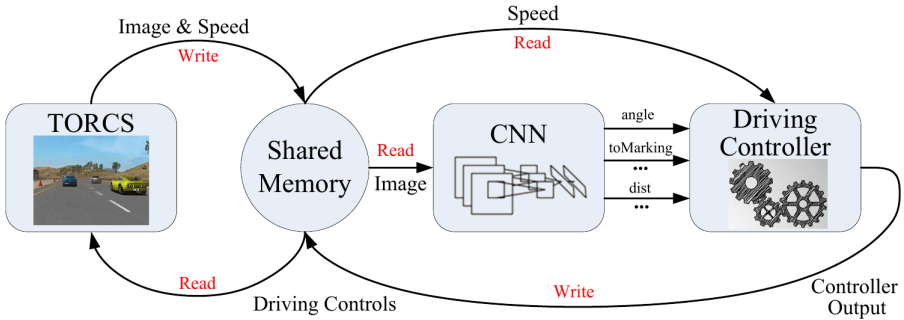
(a) angle (b) in lane: toMarking (c) in lane: dist (d) on mark.: toMarking (e) on marking: dist (f) overlapping area

Affordances:

- Attributes of the environment which limit space of actions [Gibson, 1966]
- Here: 13 affordance indicators
  - 12 of them conditioned on lateral position wrt. road
  - 2 categories: Lane perception and car perception

Chen, Seff, Kornhauser and Xiao: Learning Affordance for Direct Perception in Autonomous Driving. ICCV, 2015.

Direct Perception for Autonomous Driving: Overview



- Simulator: TORCS: Open source car racing game simulator
- Network: AlexNet (5 conv layers, 4 fully conn. layers), 13 output neurons
- Training: Affordance indicators trained with ℓ_2 loss

Chen, Seff, Kornhauser and Xiao: Learning Affordance for Direct Perception in Autonomous Driving. ICCV, 2015.

Direct Perception for Autonomous Driving: State Machine

always:

1) angle: angle between the car's heading and the tangent of the road

"in lane system", when driving in the lane:

2) toMarking_LL: distance to the left lane marking of the left lane
3) toMarking_ML: distance to the left lane marking of the current lane
4) toMarking_MR: distance to the right lane marking of the current lane
5) toMarking_RR: distance to the right lane marking of the right lane
6) dist_LL: distance to the preceding car in the left lane
7) dist_MM: distance to the preceding car in the current lane
8) dist_RR: distance to the preceding car in the right lane

"on marking system", when driving on the lane marking:

9) toMarking_L: distance to the left lane marking
10) toMarking_M: distance to the central lane marking
11) toMarking_R: distance to the right lane marking
12) dist_L: distance to the preceding car in the left lane
13) dist_R: distance to the preceding car in the right lane

Figure 4: **Complete list of affordance indicators in our direct perception representation.**

while (in autonomous driving mode)
  ConvNet outputs affordance indicators
  check availability of both the left and right lanes
  if (approaching the preceding car in the same lane)
    if (left lane exists **and** available **and** lane changing allowable)
      left lane changing decision made
    else if (right lane exists **and** available **and** lane changing allowable)
      right lane changing decision made
    else
      slow down decision made
  if (normal driving)
    $center\_line$= center line of current lane
  else if (left/right lane changing)
    $center\_line$= center line of objective lane
  compute steering command
  compute $desired\_speed$
  compute acceleration/brake command based on $desired\_speed$

Figure 5: **Controller logic.**

Direct Perception for Autonomous Driving: Controller

Steering controller:

$$s = \theta_1(\alpha - d_c/w)$$

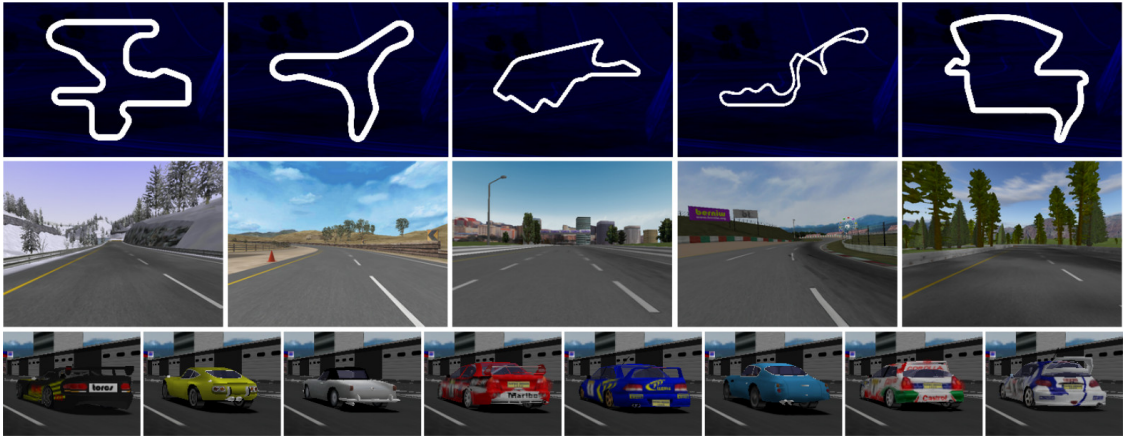
- s: steering command θ_1: parameter
- α: relative orientation d_c: distance to centerline w: road width

Speed controller:

$$v = v_{max}\left(1 - \exp\left(-\theta_2\, d_p - \theta_3\right)\right)$$

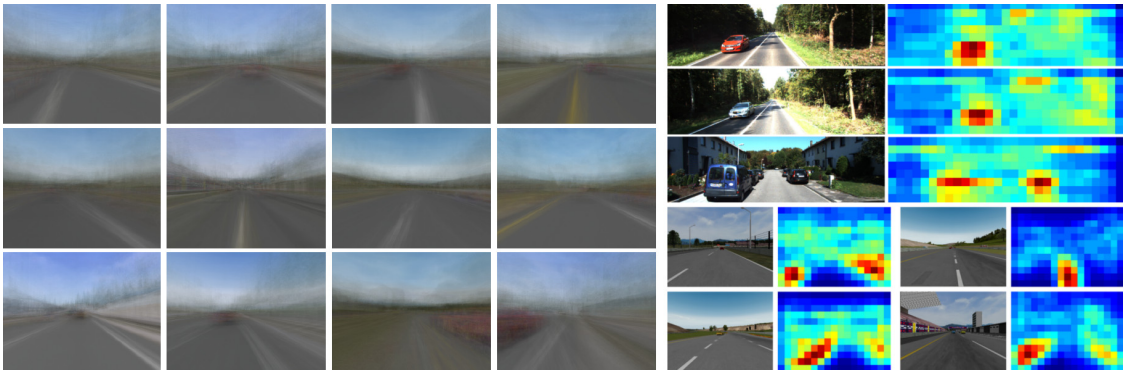
- v: target velocity v_{max} maximal velocity
- d_p: distance to preceding car $\theta_{2,3}$: parameters

Direct Perception for Autonomous Driving: TORCS Simulator



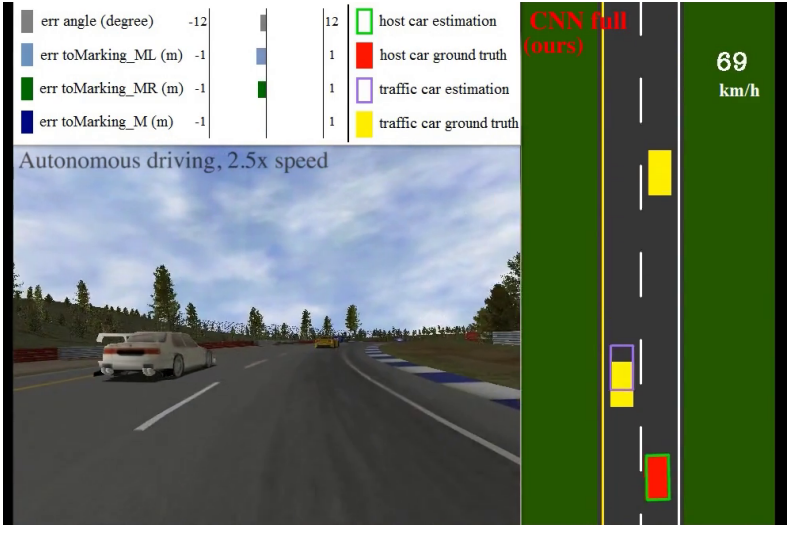
- TORCS: Open source car racing game http://torcs.sourceforge.net/

Direct Perception for Autonomous Driving: Visualization



- Left: Averaged top 100 images activating a neuron in first fully connected layer
- Right: Maximal response of 4th conv. layer (note: focus on cars and markings)

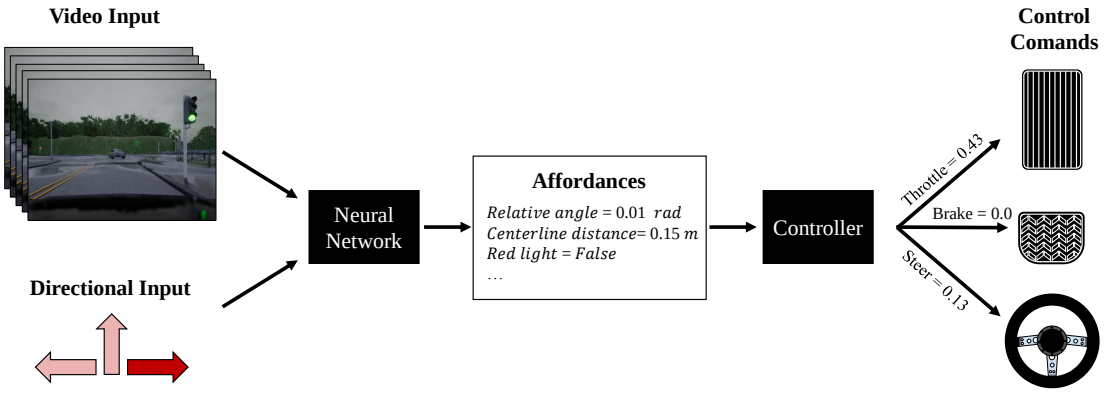
Direct Perception for Autonomous Driving: Results



Chen, Seff, Kornhauser and Xiao: Learning Affordance for Direct Perception in Autonomous Driving. ICCV, 2015.

Conditional Affordance Learning

Conditional Affordance Learning



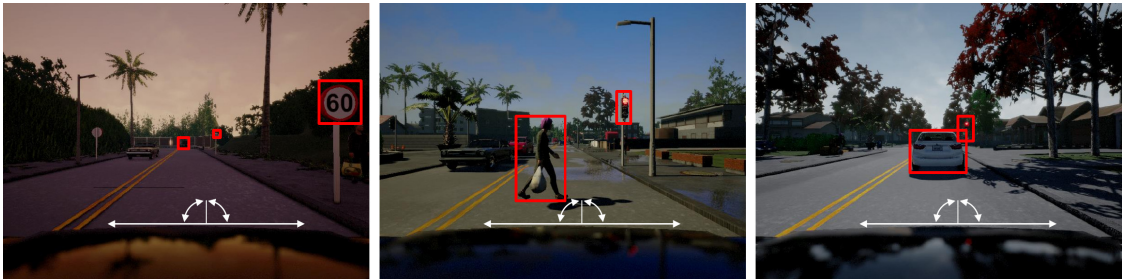
Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

Conditional Affordance Learning: Goals

- Goal: drive from A to B as fast, safely and comfortably as possible
- Infractions:
  - Driving on wrong lane
  - Driving on sidewalk
  - Running a red light
  - Violating speed limit
  - Colliding with vehicles
  - Hitting other objects

Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

Conditional Affordance Learning: Affordances



Affordances:

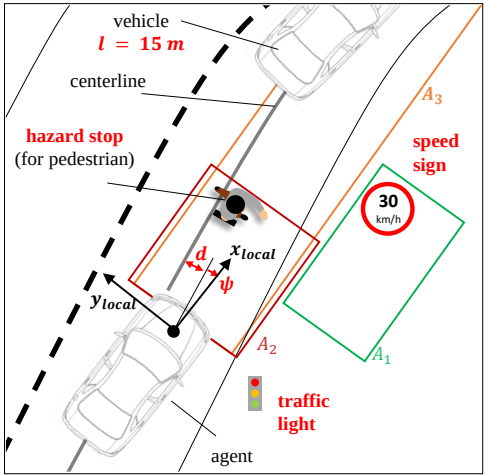
- Distance to centerline
- Relative angle to road
- Distance to lead vehicle
- Speed signs
- Traffic lights
- Hazard stop

Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

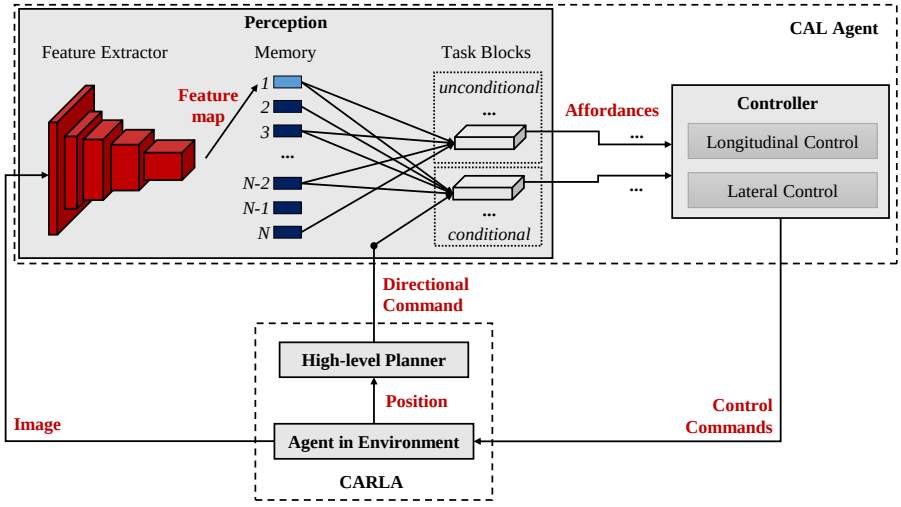
Conditional Affordance Learning: Affordances

Affordances:

- Distance to centerline
- Relative angle to road
- Distance to lead vehicle
- Speed signs
- Traffic lights
- Hazard stop

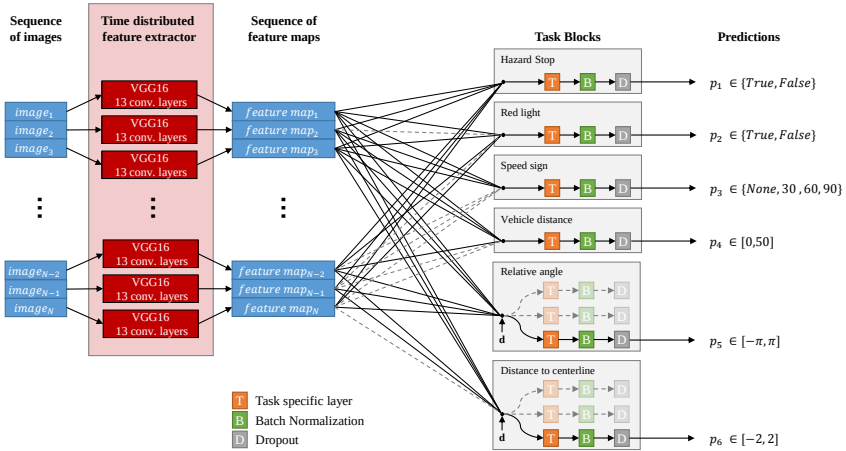


Conditional Affordance Learning: System Overview



Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

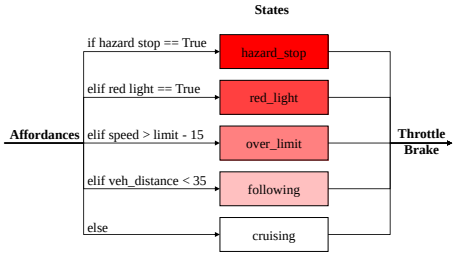
Conditional Affordance Learning: Perception Stack



- Feature network: VGG16
- Task network: FC / TCN

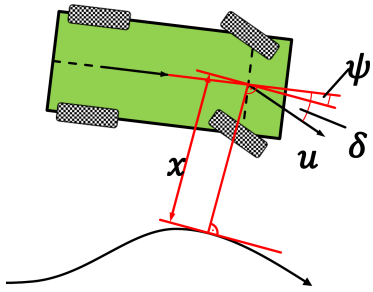
Conditional Affordance Learning: Controller

Longitudinal Control



- Finite-state machine
- PID controller for cruising
- Car following model

Lateral Control



- Stanley controller
- $\delta(t) = \psi(t) + \arctan\left(\frac{kx(t)}{u(t)}\right)$
- Damping term

Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

Conditional Affordance Learning: Parameter Estimation

Perception Stack:

- Multi-task learning: single forward pass
- Dataset: random driving using controller operating on GT affordances
  $\Rightarrow$ 240k images with GT affordances
- Loss functions:
  - Discrete affordances: Class-weighted cross-entropy (CWCE)
  - Continuous affordances: Mean average error (MAE)
- Hyperparameter search: type, nodes, temp. window & dilation factor
- Optimized with ADAM (batch size 32)

Controller:

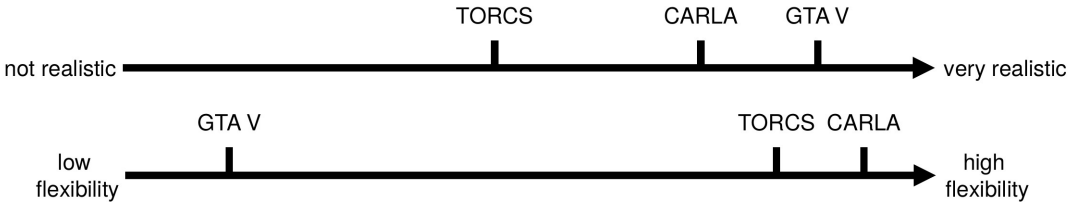
- Ziegler-Nichols

Conditional Affordance Learning: Simulators

TORCS

GTA V

CARLA



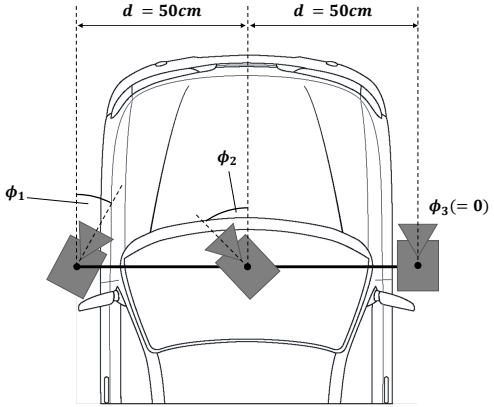
Conditional Affordance Learning: Data Collection

Data Collection:

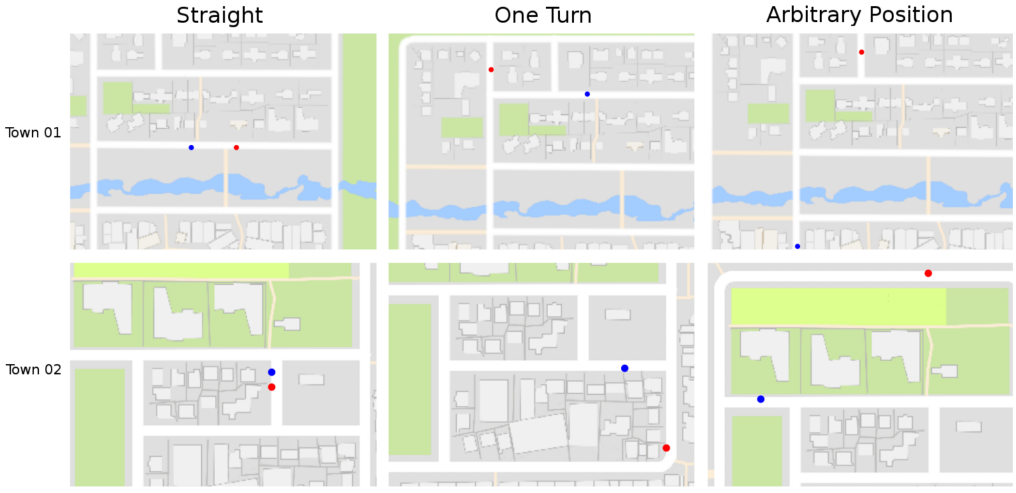
- Navigation based on true affordances & random inputs
- Image-level annotations

Data Augmentation:

- No image flipping
- Color, contrast, brightness
- Gaussian blur & noise
- Provoke rear-end collisions
- Camera pose randomization



Conditional Affordance Learning: Tasks



- Goal-oriented navigation

Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

Conditional Affordance Learning: Results

	Training conditions				New weather				New town				New town and new weather			
Task	MP	CIL	RL	CAL	MP	CIL	RL	CAL	MP	CIL	RL	CAL	MP	CIL	RL	CAL
Straight	98	95	89	**100**	**100**	98	86	**100**	92	**97**	74	93	50	80	68	**94**
One turn	82	89	34	**97**	95	90	16	**96**	61	59	12	**82**	50	48	20	**72**
Navigation	80	86	14	**92**	**94**	84	2	90	24	40	3	**70**	47	44	6	**68**
Nav. dynamic	77	**83**	7	**83**	**89**	82	2	82	24	38	2	**64**	44	42	4	**64**

Baselines:

- MP = Modular Pipeline [Dosovitskiy et al., CoRL 2017]
- CIL = Conditional Imitation Learning [Codevilla et al., ICRA 2018]
- RL = Reinforcement Learning A3C [Mnih et al., ICML 2016]

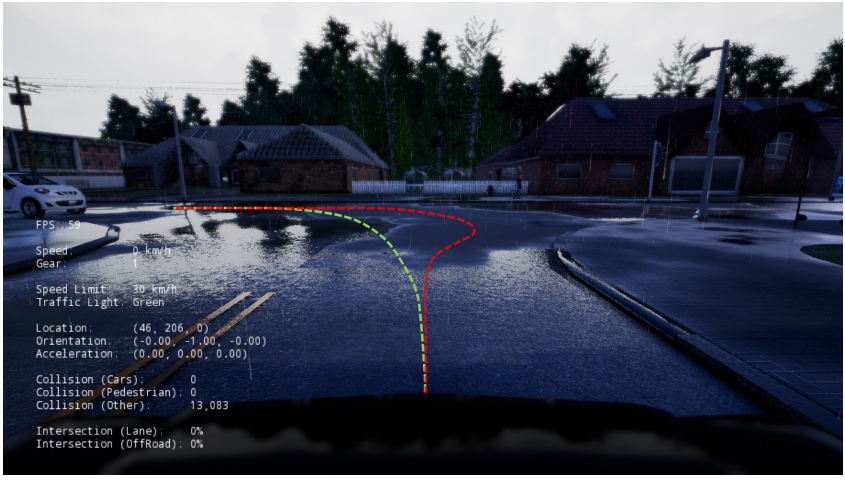
Conditional Affordance Learning: Results

Hazard Stop

Conditional Affordance Learning: Attention

Attention to Hazard Stop

Conditional Affordance Learning: Path Planning



Optimal Path (green) vs. Traveled Path (red)

Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.

Conditional Affordance Learning: Failure Cases

Hazard Stop: False Positive

Conditional Affordance Learning: Failure Cases

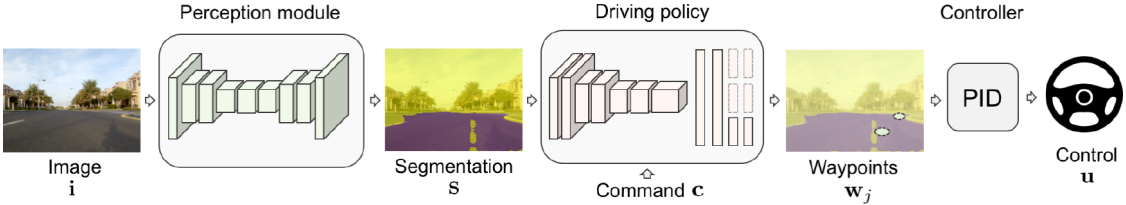
Hazard Stop: False Negative

Conditional Affordance Learning: Failure Cases

Red Light: False Positive

Driving Policy Transfer

Driving Policy Transfer: Overview



Problem:

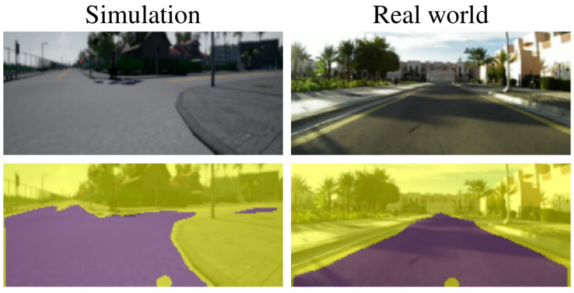
- Driving policies learned in simulation often do not transfer well to the real world

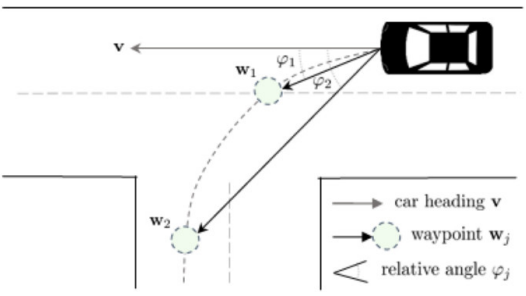
Idea:

- Encapsulate driving policy such that it is not directly exposed to raw perceptual input or low-level control (input: semantic segmentation, output: waypoints)
- Allows for transferring driving policy without retraining or finetuning

Müller, Dosovitskiy, Ghanem and Koltun: Driving Policy Transfer via Modularity and Abstraction. CoRL, 2018.

Driving Policy Transfer: Waypoints

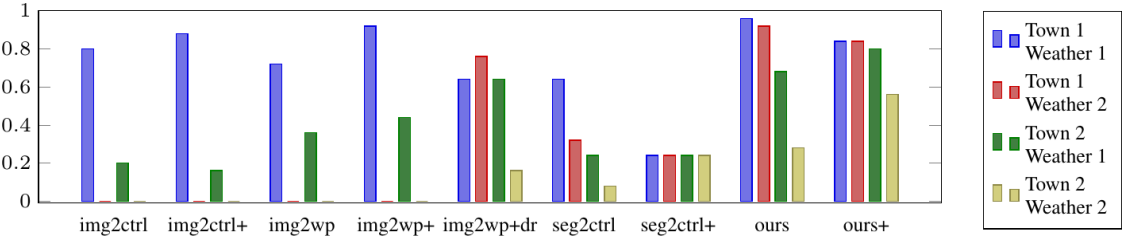




Representation:

- Input: Semantic segmentation (per pixel “road” vs. “non-road”)
  - ERFNet [Romera et al., IV 2017] trained on Cityscapes [Cordts et al., CVPR 2016]
- Output: 2 waypoints (distance to vehicle, relative angle wrt. vehicle heading)
  - One sufficient for steering, second one for braking before turns

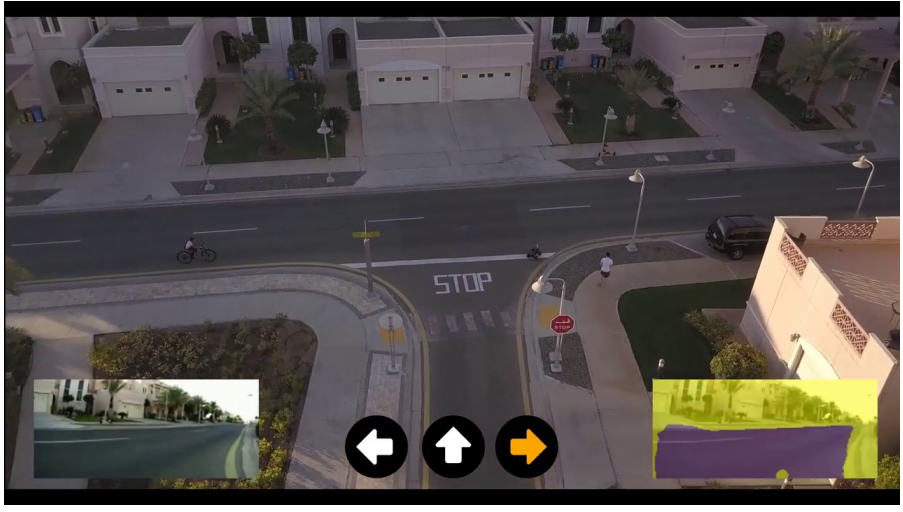
Driving Policy Transfer: Results



Success Rate over 25 Navigation Trials

- Driving policy: Conditional Imitation Learning (branched)
- Training: Expert agent, random initialization, noise injection
- Control: PID for lateral and longitudinal control
- Results: Full method generalizes best (“+” = with data augmentation)

Driving Policy Transfer: Results



Müller, Dosovitskiy, Ghanem and Koltun: Driving Policy Transfer via Modularity and Abstraction. CoRL, 2018.

Further Readings

- Bojarski et al.: VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving. ICRA, 2018.
- Codevilla, Müller, López, Koltun and Dosovitskiy: End-to-End Driving Via Conditional Imitation Learning. ICRA, 2018.
- Chen, Seff, Kornhauser and Xiao: Learning Affordance for Direct Perception in Autonomous Driving. ICCV, 2015.
- Sauer, Savinov and Geiger: Conditional Affordance Learning for Driving in Urban Environments. CoRL, 2018.
- Müller, Dosovitskiy, Ghanem and Koltun: Driving Policy Transfer via Modularity and Abstraction. CoRL, 2018.

Questions?