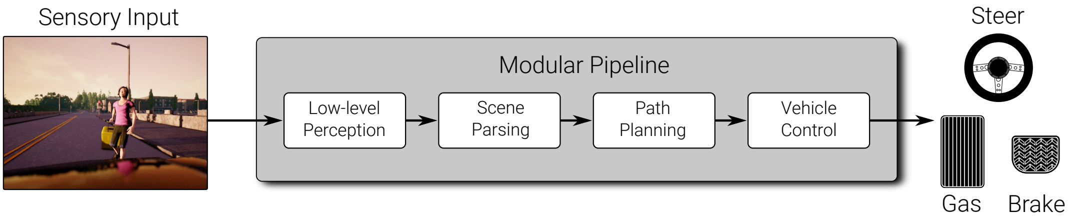
Towards Robust End-to-End Driving

Andreas Geiger

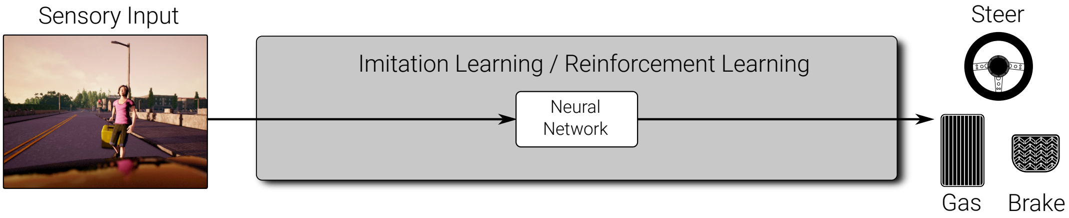
Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems Tübingen

March 23, 2021

Approaches to Self-Driving

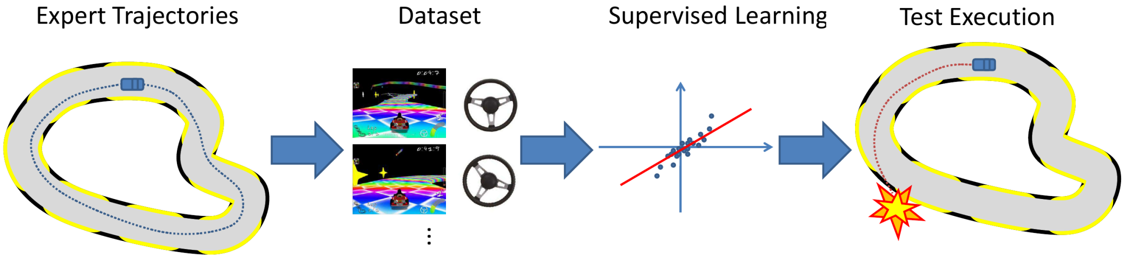


+ Modular + Interpretable - Expert decisions - Piece-wise training



+ End-to-end + Simple - Generalization - Interpretable

Imitation Learning



Motivation: Hard coding policies is difficult $\Rightarrow$ follow data-driven approach!

- **Given:** demonstrations or demonstrator
- **Goal:** train a policy to mimic decision

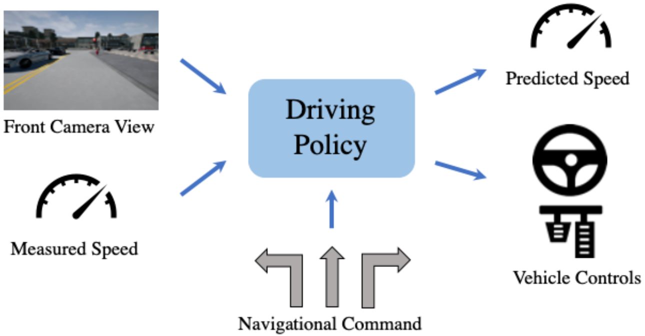
Conditional Imitation Learning

Advantages:

- End-to-End Trainable
- Cheap Annotations

Limitations:

- Generalization
- High Sample Complexity
- Interpretability

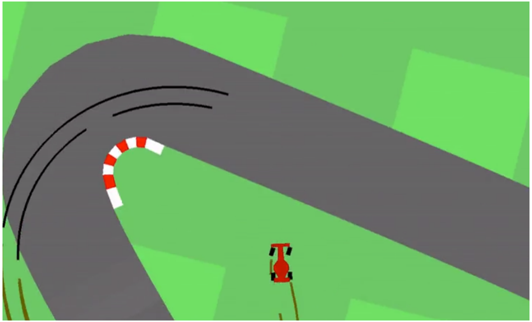


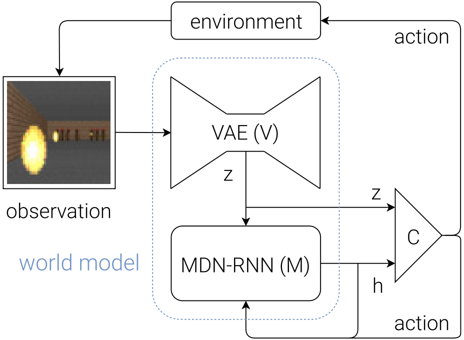
Codevilla, Santana, Lopez and Gaidon: Exploring the Limitations of Behavior Cloning for Autonomous Driving. ICCV, 2019.

How can we learn to drive under the **vast diversity** of all visual, planning and control scenarios?

Situational Driving

Inspiration: World Models

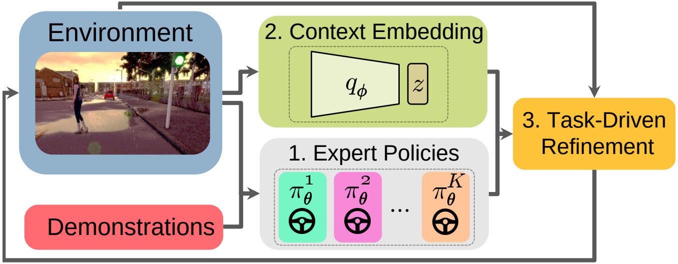




- Step 1: Learn **generative model** of game environments (VAE)
- Step 2: Learn dynamics model and control model in **latent space** (CMA-ES)
- Not sufficient $\Rightarrow$ we combine this idea with **imitation learning**

Ha and Schmidhuber: Recurrent World Models Facilitate Policy Evolution. NeurIPS, 2018.

Learning Situational Driving



- Step 1: Learn a **mixture of expert policies** $\{\alpha_{\boldsymbol{\theta}}^k, \pi_{\boldsymbol{\theta}}^k\}$ via imitation (LSD)
- Step 2: Learn a **general purpose context embedding** q_ϕ as a β-VAE
- Step 3: Perform **task-driven policy refinement** by interacting with the simulation and maximizing a driving task reward (LSD+)

Learning Situational Driving

$$\pi_{\mathbf{\Theta}}(\mathbf{a}|\mathbf{o},c) = \sum_{k=1}^{K} \underbrace{\alpha_{\boldsymbol{\theta}}^{k}(\mathbf{o},c)}_{\text{Mixture Weights}} \underbrace{\pi_{\boldsymbol{\theta}}^{k}(\mathbf{a}|\mathbf{o},c)}_{\text{Expert Models}} + \mathbf{\Psi} \underbrace{\begin{bmatrix} q_{\boldsymbol{\phi}}(\mathbf{I}) \\ v \\ c \end{bmatrix}}_{\text{Context Embedding}}$$

$$\pi_{\boldsymbol{\theta}}^{k}(\mathbf{a}|\mathbf{o},c) = \mathcal{N}\left(\mathbf{a} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{\theta}}^{k}(\mathbf{o},c),\ \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{\theta}}^{k}(\mathbf{o},c))^2\right)$$

Observations: $\mathbf{o} = [\mathbf{I}, v] \in \mathcal{O}$

Command: $c \in \mathcal{C} = \{\text{left, right, straight, follow}\}$

Actions: $\mathbf{a} \in \mathcal{A} = [-1, 1]^2$

Learning Situational Driving

$$\pi_{\boldsymbol{\Theta}}(\mathbf{a}|\mathbf{o},c) = \sum_{k=1}^{K} \underbrace{\alpha_{\boldsymbol{\theta}}^{k}(\mathbf{o},c)}_{\text{Mixture Weights}} \underbrace{\pi_{\boldsymbol{\theta}}^{k}(\mathbf{a}|\mathbf{o},c)}_{\text{Expert Models}} + \boldsymbol{\Psi} \underbrace{\begin{bmatrix} q_{\boldsymbol{\phi}}(\mathbf{I}) \\ v \\ c \end{bmatrix}}_{\text{Context Embedding}}$$

$$\pi_{\boldsymbol{\theta}}^{k}(\mathbf{a}|\mathbf{o},c) = \mathcal{N}\left(\mathbf{a}\,\middle|\,\boldsymbol{\mu}_{\boldsymbol{\theta}}^{k}(\mathbf{o},c),\ \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\theta}}^{k}(\mathbf{o},c))^2\right)$$

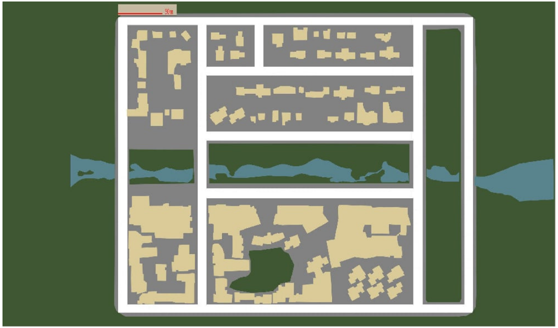
Training:

- Step 1: Learn Mixture of Experts: $\mathcal{L}_{\text{MoE}} = -\log\left[\sum_{k=1}^{K} \alpha_{\boldsymbol{\theta}}^{k}\pi_{\boldsymbol{\theta}}^{k}\right] + \mathcal{L}_{\text{V}} + \mathcal{L}_{\text{R}}$
- Step 2: Learn Context Embedding: $\mathcal{L}_{\text{VAE}} = \beta\,\mathrm{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{I}) \parallel p_0(\mathbf{z})\right) + \|d_{\boldsymbol{\phi}}(\mathbf{z}) - \mathbf{I}\|_2^2$
- Step 3: Task-driven optimization: $\mathcal{J}_{\text{TASK}}(\boldsymbol{\theta}_{\text{readout}}, \boldsymbol{\Psi}) = \mathbb{E}_{\pi_{\boldsymbol{\Theta}}}\left[\sum_{t=0}^{T} r_t\right]$

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

Experiments

CARLA



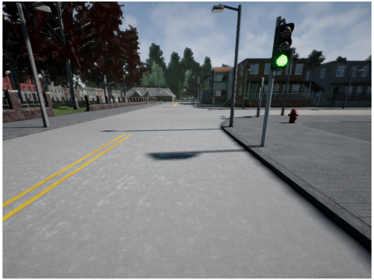
Training Town

Test Town

- Random start and end location, 4 known weathers, 2 unseen weathers
- Metric: Percentage of successfully completed episodes (success rate)
- Collision does not necessarily terminate episode

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

CARLA NoCrash Benchmark



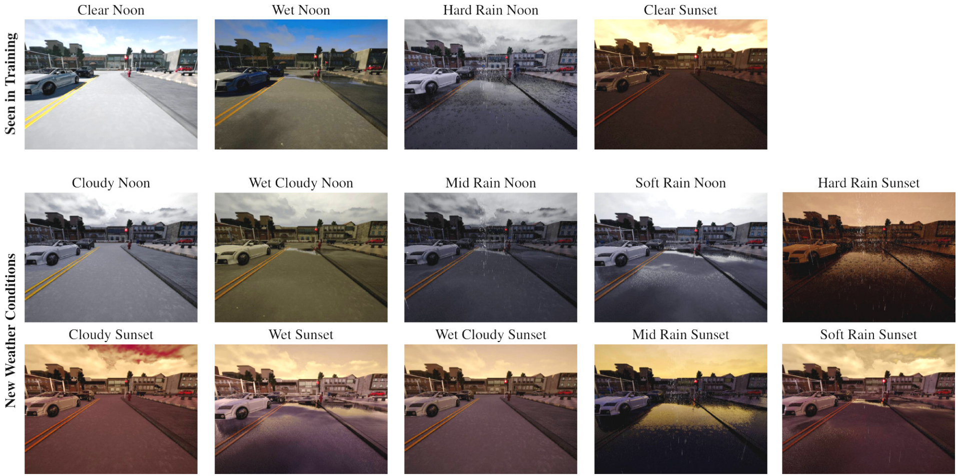
Empty

Regular

Dense

- Difficulty varies with number of dynamic agents in the scene
- Empty: 0 Agents   Regular: 65 Agents   Dense: 220 Agents
- All collisions terminate episode

CARLA AnyWeather Benchmark



- Evaluation on 10 unseen weathers, quantifies generalization performance

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

Importance of Mixture Model

	Training Data and Mixture Components		
Evaluation Task	Navigation (Static, K=1)	Navigation (Dynamic, K=1)	Navigation (Dynamic, K=3)
Straight (Static)	99	64	**100**
One Turn (Static)	98	74	**100**
Navigation (Static)	96	78	**98**
Navigation (Dynamic)	40	78	**92**

Results of Mixture Model on CARLA Benchmark:

- Static model solves static scenes well but cannot handle dynamic objects
- Dynamic model handles dynamic scenes better but degrades on static scenes
- Dynamic mixture model generalizes to all scenarios (without on-policy data)

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

Importance of Mixture Model and Task-based Refinement

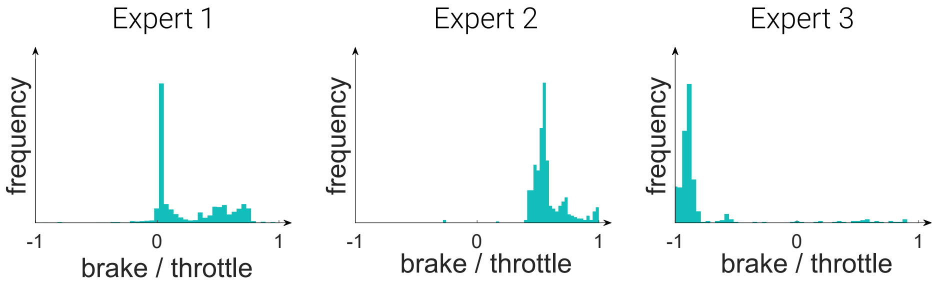
Model	Success Rate (%)
Monolithic (K=1)	75
MoE Shared Backbone (K=3)	89
MoE Shared Backbone (K=5)	90
MoE Shared Backbone (K=8)	87
MoE Separate Backbone (K=3)	94
MoE Separate Backbone (K=5)	93
MoE Separate Backbone (K=8)	93
MoE Separate Backbone + Refinement (K=3)	**98**

Results of Full Model on CARLA Benchmark:

- Performance improves up to 3 or 5 mixture components
- Separate backbones increase diversity and generalization
- Tasked-based refinement improves performance further

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

Emergent Driving Modes



Emergent Driving Modes:

- Acceleration distribution of three different experts during testing

Results on CARLA Benchmark

Driving Task	CIRL	CILRS	CILRS (ours)	**LSD (ours)**	**LSD+R (ours)**
Straight	**100**	96	96	**100**	**100**
One Turn	71	84	86	**99**	**99**
Navigation	53	69	67	**99**	**99**
Navigation Dynamic	41	66	64	94	**98**

- Using reward-based optimization alone (CIRL) is not sufficient
- LSD enables better driving behavior across all driving tasks
- Large improvements in the presence of dynamic objects

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

Results on CARLA NoCrash Benchmark

Driving Task	CILRS	CILRS	**LSD (ours)**	**LSD+R (ours)**	Expert
Empty	66 ± 2	65 ± 2	93 ± 2	$\mathbf{94 \pm 1}$	96 ± 0
Regular	49 ± 5	46 ± 2	66 ± 2	$\mathbf{68 \pm 2}$	91 ± 1
Dense	23 ± 1	20 ± 1	27 ± 2	$\mathbf{30 \pm 4}$	41 ± 2

- All methods perform worse due to challenges (density, collision terminations)
- Expert provided by CARLA often fails in dense environments (e.g., clogging)
- LSD enables better driving behavior across all driving tasks

Results on AnyWeather Benchmark

Task	CILRS	**LSD (ours)**	**LSD+R (ours)**
Straight	83.2	85.2	**85.6**
One Turn	78.4	80.4	**81.6**
Navigation	76.4	78.8	**79.6**
Nav. Dynamic	75.6	77.2	**78.4**

- AnyWeather benchmark test generalization to challenging unseen weathers
- All methods can fail even on simple straight driving tasks
- Some challenging weathers lead to zero success rate for all methods
- More research is required to address these challenges

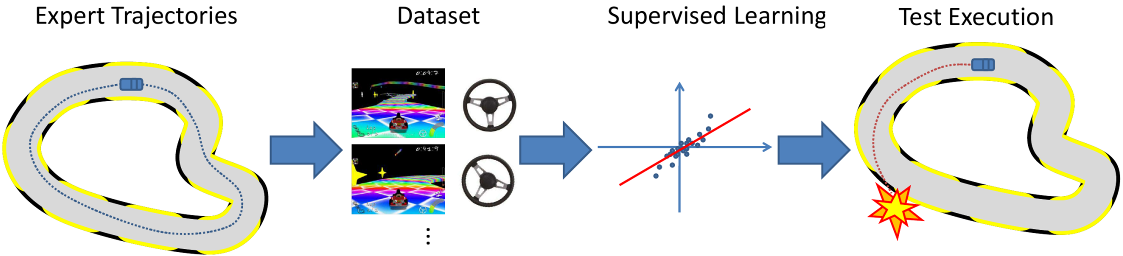
Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

Qualitative Results

Ohn-Bar, Prakash, Behl, Chitta and Geiger: Learning Situational Driving. CVPR, 2020.

How useful is **data aggregation** for self-driving?

Imitation Learning



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

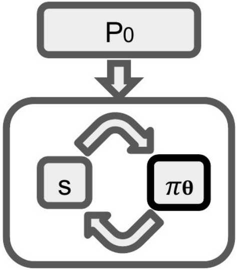
- **Given:** demonstrations or demonstrator
- **Goal:** train a policy to mimic decision

Formal Definition of Imitation Learning

General Imitation Learning:

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

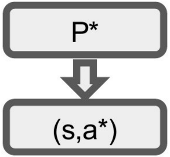
- State distribution $P(s|\pi_\theta)$ depends on rollout determined by current policy π_θ



Behavior Cloning:

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

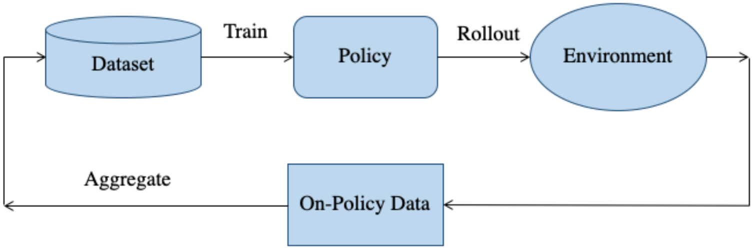
- State distribution P^* provided by expert
- Reduces to supervised learning problem



Challenges of Behavior Cloning

- Behavior cloning makes IID assumption
  - Next state is sampled from states observed during expert demonstration
  - Thus, next state is sampled independently from action predicted by current policy
- What if π_θ makes a mistake?
  - Enters new states that haven't been observed before
  - New states not sampled from same (expert) distribution anymore
  - Cannot recover, can lead to catastrophic failure

DAgger

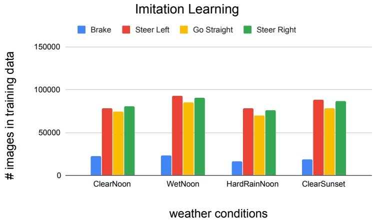


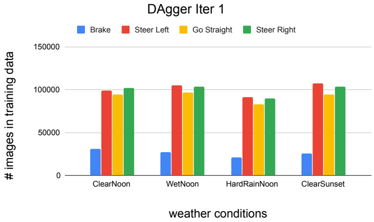
Data Aggregation (DAgger):

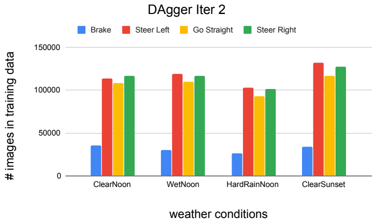
- Iteratively build a set of inputs that the final policy is likely to encounter based on previous experience. Query expert for aggregate dataset.
- But can easily overfit to main mode of demonstrations
- High training variance (random initialization, order of data)

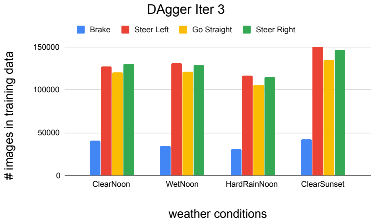
Ross, Gordon and Bagnell: A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. AISTATS, 2011.

Distribution over Driving Actions



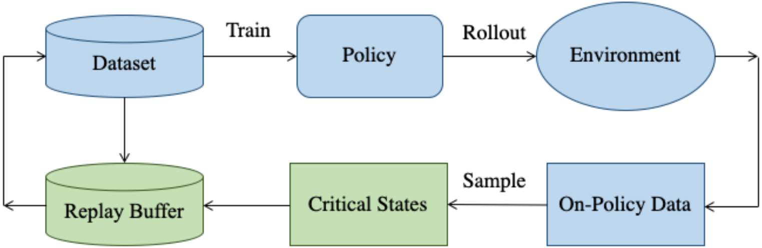






Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

DAgger with Critical States and Replay Buffer

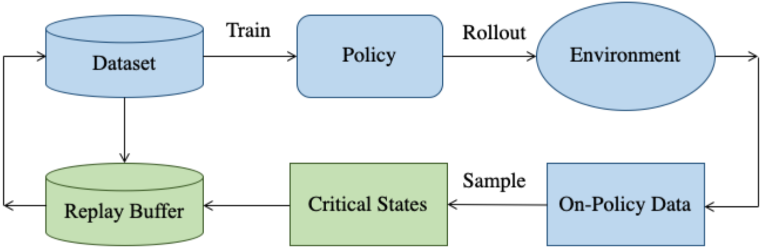


Key Ideas:

1. Sample **critical states** from the collected on-policy data based on the utility they provide to the learned policy in terms of driving behavior
2. Incorporate a **replay buffer** which progressively focuses on the high uncertainty regions of the policy's state distribution

Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

DAgger with Critical States and Replay Buffer

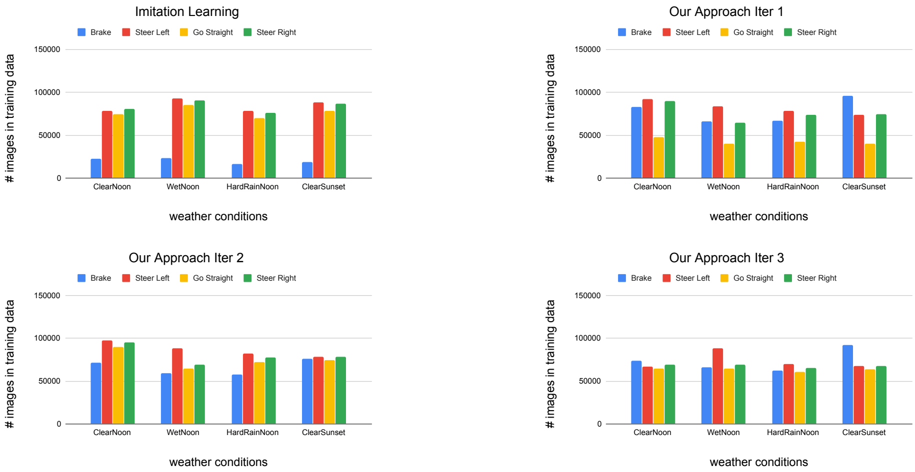


Sampling Strategies:

- Task-based: Sample uniformly from “left”, “right”, “straight”
- Policy-based: Use test-time dropout to estimate epistemic uncertainty
- Expert-based: Highest loss or deviation in brake signal wrt. expert

Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

Distribution over Driving Actions



Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

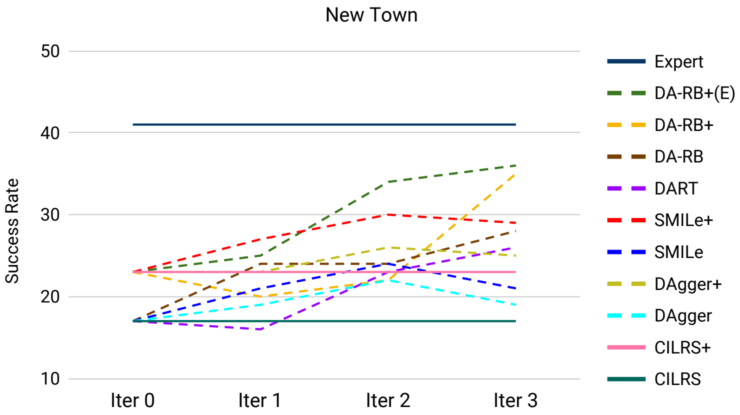
Experiments

Evaluation

Empty | Regular | Dense

- CARLA **NoCrash benchmark**
- **Dense setting** with 220 agents
- Comparison to various baselines with (+) and without data augmentation

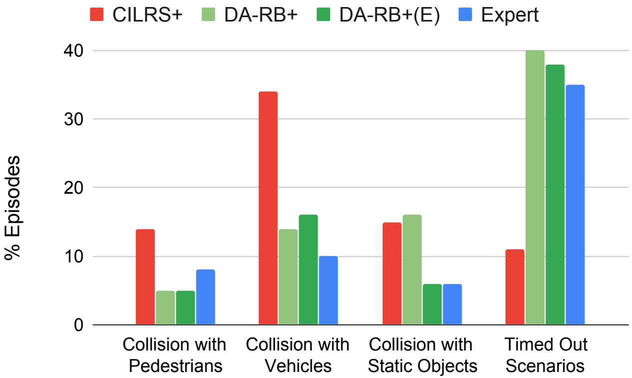
Evaluation



- Data augmentation increases the performance of all methods
- DAgger overfits quickly (!), not better than data augmentation
- Our model consistently improves upon the baselines in all conditions

Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

Infractions Analysis



- Signficiant reduction in collisions with dynamic objects
- More time-outs due to less infractions (e.g., clogged scenes, red lights)

Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

Training Variance

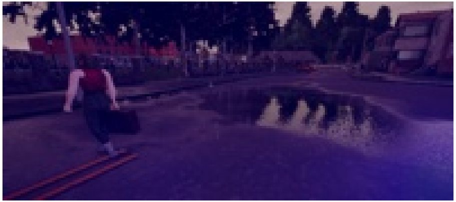
	CILRS^{+}	DAgger^{+}	DA-RB^{+}
Iter 0	$14.6 \pm \mathbf{3.4}$	14.6 ± 3.4	14.6 ± 3.4
Iter 1	-	15.2 ± 5.1	24.8 ± 1.9
Iter 2	-	13.2 ± 1.9	25.4 ± 1.5
Iter 3	-	$17.8 \pm \mathbf{3.6}$	$27.0 \pm \mathbf{0.9}$

Standard deviation wrt. 5 random training seeds (New Town & Weather)

- Significant reduction in variance compared to CILRS and DAgger

Interpretability: GradCAM Attention Maps

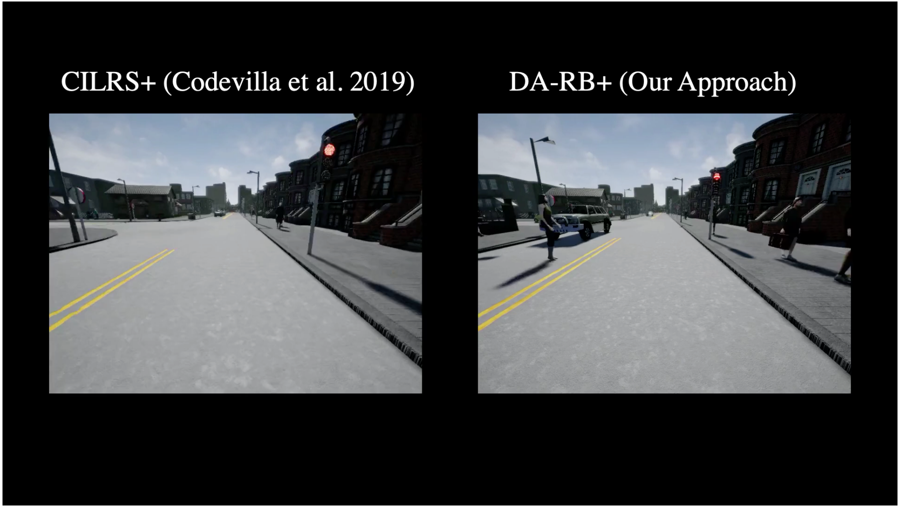
CILRS [Codevilla et al. 2019]



Our Approach

Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

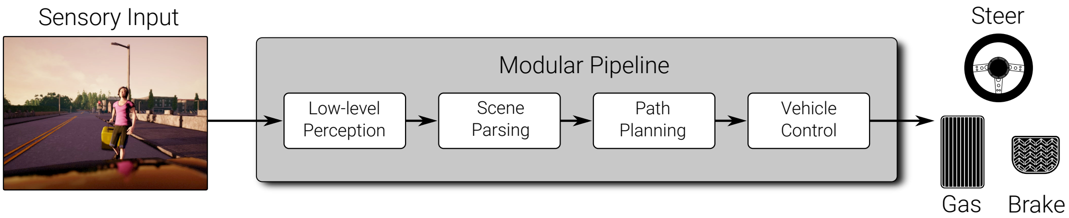
Qualitative Results



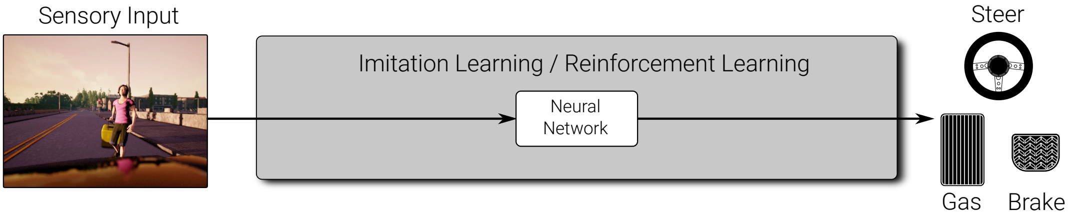
Prakash, Behl, Ohn-bar, Chitta and Geiger: Exploring Data Aggregation in Policy Learning for Vision-based Urban Autonomous Driving. CVPR, 2020.

What is a good **intermediate representation?**

Approaches to Self-Driving



+ Modular + Interpretable - Expert decisions - Piece-wise training

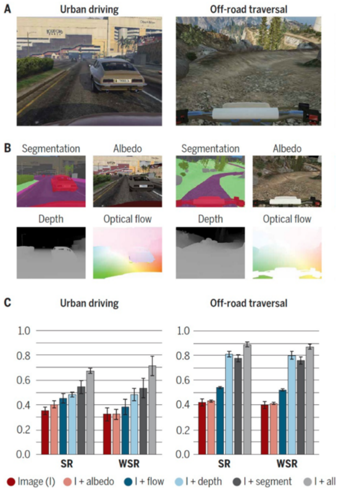


+ End-to-end + Simple - Generalization - Interpretable

Does Computer Vision Matter for Action?

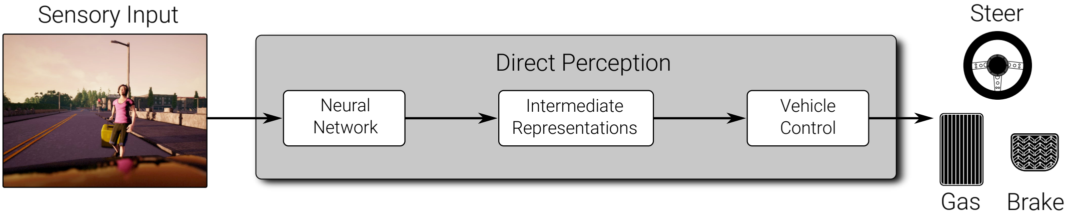
Does Computer Vision Matter for Action?

- Analyze various intermediate representations: segmentation, depth, normals, flow, albedo
- Intermediate representations improve results
- Consistent gains across simulations / tasks
- Depth and semantic provide largest gains

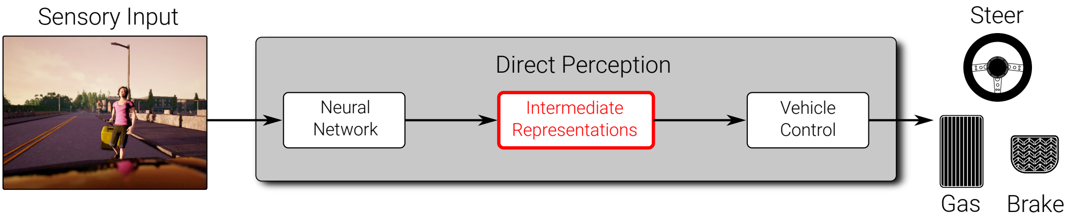


Zhou, Krähenbühl and Koltun: Does computer vision matter for action? Science Robotics, 2019.

Approaches to Self-Driving



Approaches to Self-Driving



Which intermediate modality?

- Semantic segmentation
- Bounding boxes
- Depth
- Optical flow

Visual Abstractions

What is a good visual abstraction?

- Invariant (hide irrelevant variations from policy)
- Universal (applicable to wide range of scenarios)
- Data efficient (in terms of memory/computation)
- Label efficient (require little manual effort)

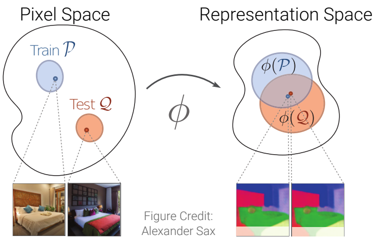


Figure Credit: Alexander Sax

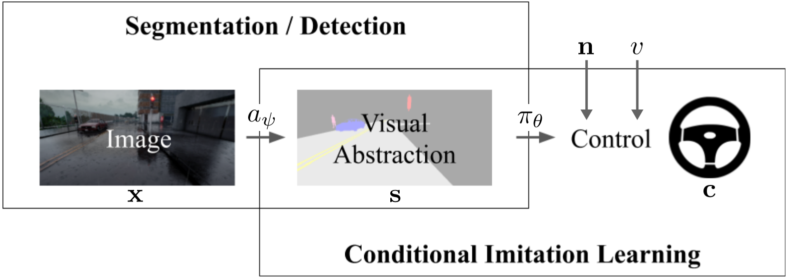
Semantic segmentation:

- Encodes task-relevant knowledge (e.g. road is drivable) and priors (e.g., grouping)
- Can be processed with standard 2D convolutional policy networks

Disadvantage:

- Labelling time: ∼90 min for 1 Cityscapes image

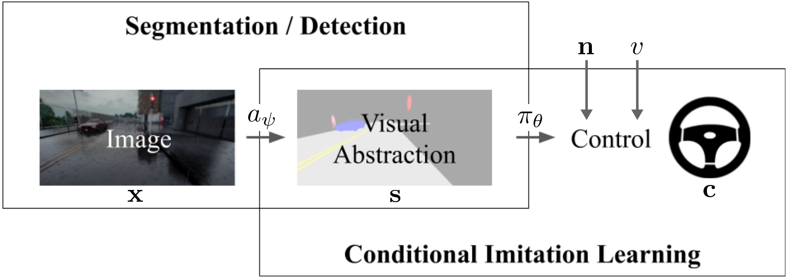
Label Efficient Visual Abstractions



Model:

- Visual abstraction network $a_\psi : \mathbf{x} \mapsto \mathbf{s}$
- Control policy $\pi_\theta : \mathbf{s}, \mathbf{n}, v \mapsto \mathbf{c}$
- Composing both yields $\mathbf{c} = \pi_\theta(a_\phi(\mathbf{x}))$

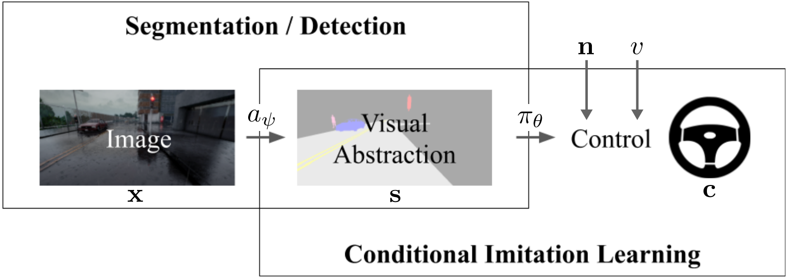
Label Efficient Visual Abstractions



Datasets:

- n_s images annotated with semantic labels $S = \{\mathbf{x}^i, \mathbf{s}^i\}_{i=1}^{n_s}$
- n_c images annotated with expert driving controls $C = \{\mathbf{x}^i, \mathbf{c}^i\}_{i=1}^{n_c}$
- We assume $n_s \ll n_c$

Label Efficient Visual Abstractions



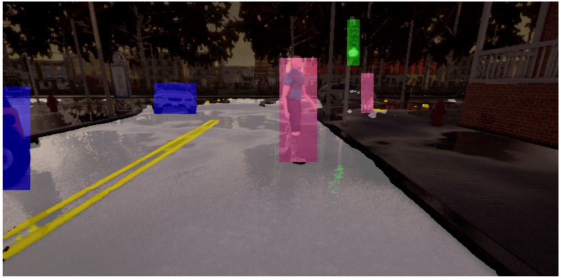
Training:

- Train visual abstraction network $a_\phi(\cdot)$ using semantic dataset S
- Apply this network to obtain control dataset $C_\phi = \{a_\phi(\mathbf{x}^i), \mathbf{c}^i\}_{i=1}^{n_c}$
- Train control policy $\pi_\theta(\cdot)$ using control dataset C_ϕ

Behl, Chitta, Prakash, Ohn-Bar and Geiger: Label Efficient Visual Abstractions for Autonomous Driving. IROS, 2020.

Results

Trained with 6400 finely annotated images and 14 classes
Annotation time ≈ 7500 hours, policy success rate = 50%



Trained with 1600 coarsely annotated images and 6 classes
Annotation time ≈ 50 hours, policy success rate = 58%

Behl, Chitta, Prakash, Ohn-Bar and Geiger: Label Efficient Visual Abstractions for Autonomous Driving. IROS, 2020.

Summary

Summary

- **Mixture models** can significantly improve **generalization**
- **Task-driven optimization** is difficult but important
- **Data augmentation** is important but can easily **overfit** in self-driving
- **Critical states** and **replay buffer** improve performance and reduce variance
- Exploiting visual abstractions leads to **more robust driving** models
- Higher **segmentation accuracy** does not necessarily imply better driving
- Hybrid representations **reduce annotation costs**
- Visual abstractions can significantly **lower training variance**
- **Attention** is helpful for self-driving, but hasn't been explored much yet

Thank you!

http://autonomousvision.github.io

erc
DFG
Federal Ministry of Education and Research
Federal Ministry for Economic Affairs and Energy
intel
Microsoft Research
NVIDIA