Query-Efficient Imitation Learning for End-to-End Simulated Driving

Jiakai Zhang, Kyunghyun Cho

New York University

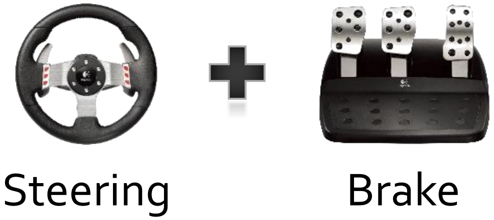
Overview

- Introduction
  - End-to-end learning for self-driving
  - Related work
- Learning method
  - Convolutional neural network
  - Imitation learning using SafeDAgger
- Experiment
  - Setup
  - Results
- Conclusion and future work

Introduction

- End-to-end learning for self-driving
  - Sensory input from front-facing camera

  - Control signal



Introduction

➢ Related work

- Supervised learning
  - ALVINN net [Pomerleau 1989]
  - DeepDriving [Chen et al. 2015]
  - End-to-end learning for self-driving cars [Bojarski et al. 2016]
- Imitation learning
  - DAgger [Ross, Gordon, and Bagnell 2010]
  - SafeDAgger [Zhang and Cho 2017]

DAgger algorithm

Initialize: Dataset D_0 + Policy $\hat{\pi}_1$

Iteration:

Policy $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$ → Dataset D'

↓

Dataset $D_i = D' \cup D_{i-1}$ → Policy $\hat{\pi}_i$

Return: Best policy $\hat{\pi}_i$

Disadvantage:

- Query a reference policy constantly
- Safe issue to environment

SafeDAgger algorithm

Initialize: Dataset D_0 + Policy $\hat{\pi}_1$ + Safety classifier c_1

Iteration:

- Policy $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$
- Safety classifier c_i

→ Dataset D' not safe

↓

Dataset $D_i = D' \cup D_{i-1}$

←

- Policy $\hat{\pi}_i$
- Safety classifier c_1

Return: Best policy $\hat{\pi}_i$ + Safety classifier c_i

Advantage:

- Query-efficient
- Safety feature

Safety classifier

- Deviation of a primary policy from a reference policy defined

$$\epsilon(\pi, \pi^*, \phi(s)) = \|\pi(\phi(s)) - \pi^*(\phi(s))\|^2$$

- Optimal safety classifier defined as

$$c^*_{\text{safe}}(\pi, \phi(s)) = \begin{cases} 0, & \text{if } \epsilon(\pi, \pi^*, \phi(s)) > \tau \\ 1, & \text{otherwise} \end{cases}$$

Learning safety classifier

- Minimize a binary cross-entropy loss

$$l_{\text{safe}}(c_{\text{safe}}, \pi, \pi^*, D') = -\frac{1}{N}\sum_{n=1}^{N} c^*_{\text{safe}}(\phi(s)'_n) \log c_{\text{safe}}(\phi(s)'_n, \pi) + (1 - c^*_{\text{safe}}(\phi(s)'_n)) \log(1 - c_{\text{safe}}(\phi(s)'_n, \pi))$$

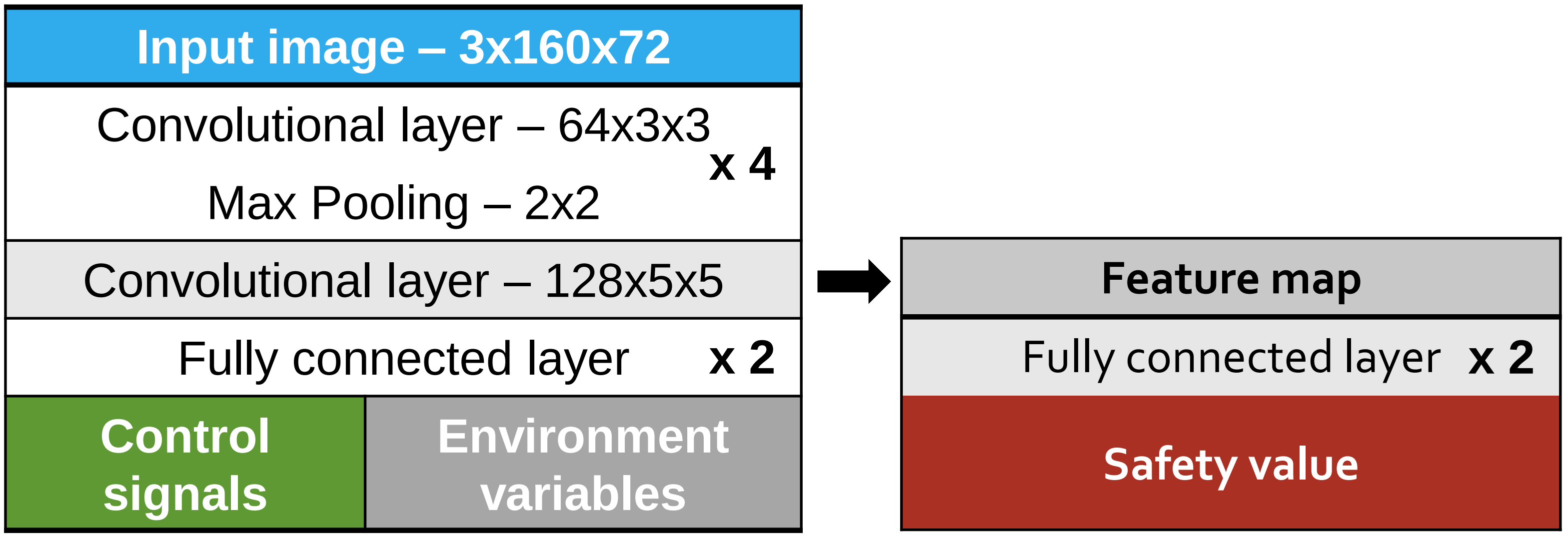
Experiment – Setup

- TORCS – Open source racing game

Training tracks

Test tracks

Experiment – Model

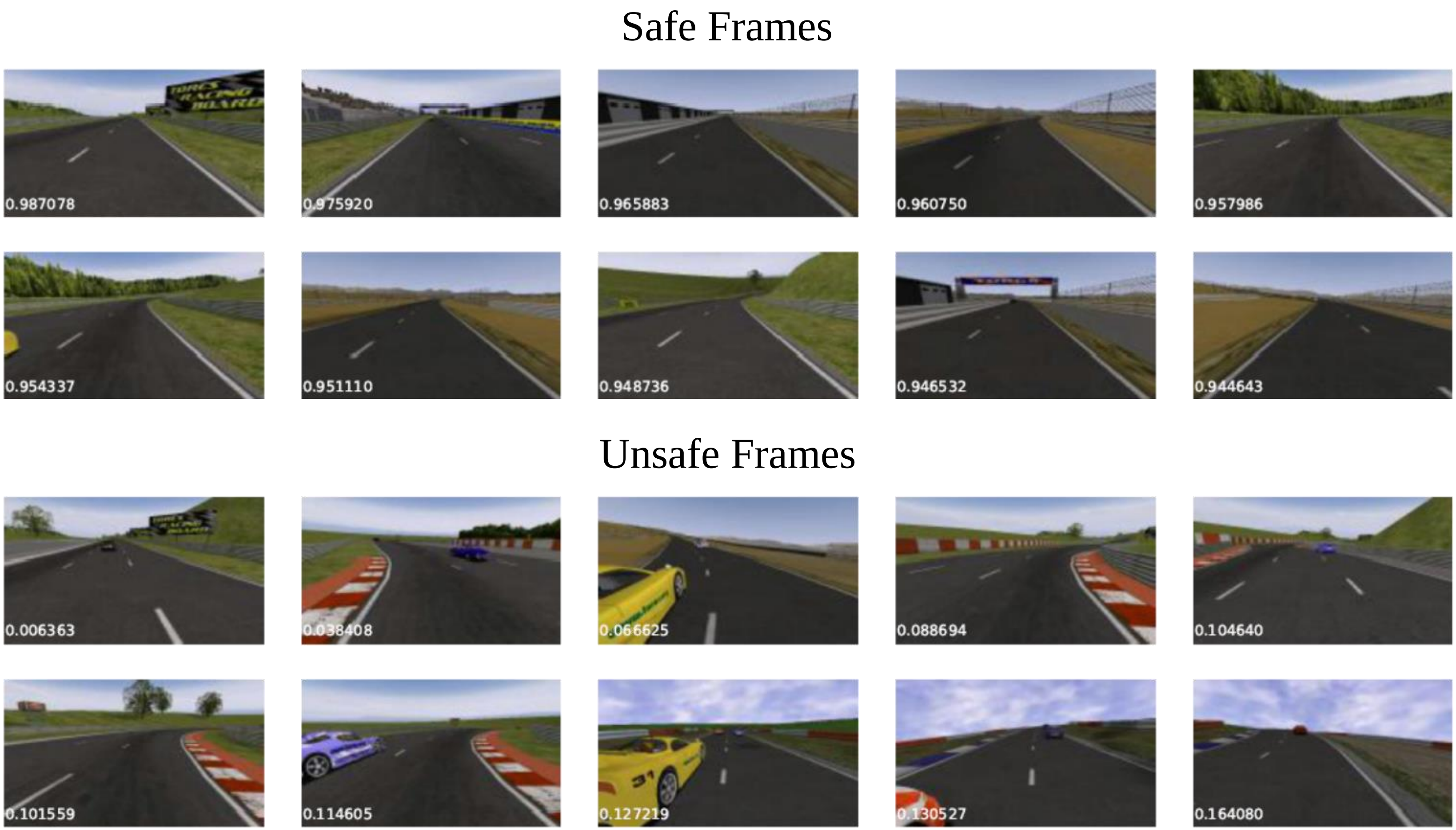


Primary policy

Safety classifier

Optimization algorithm: stochastic gradient descent

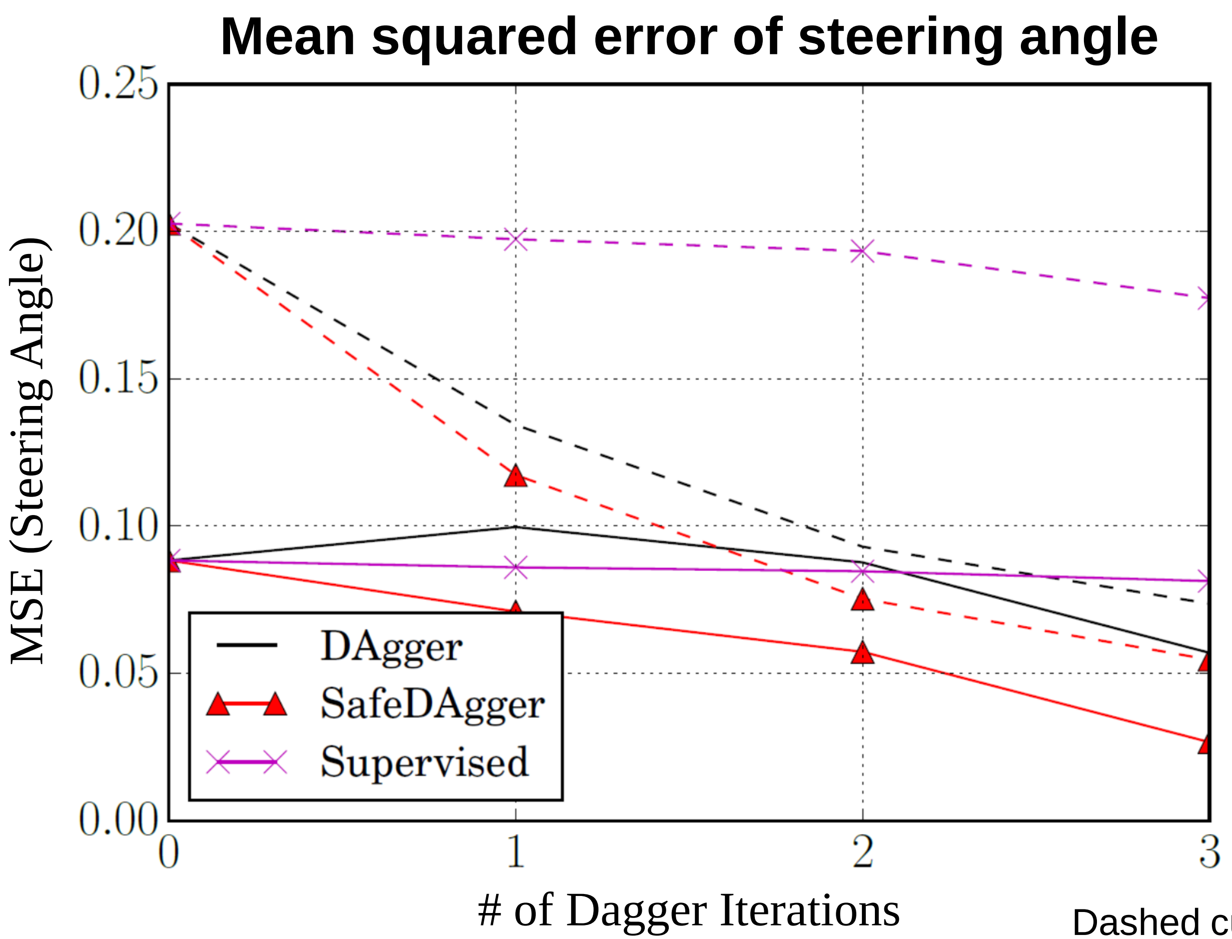
Results


Safe Frames
0.987078
0.975920
0.965883
0.960750
0.957986
0.954337
0.951110
0.948736
0.946532
0.944643
Unsafe Frames
0.006363
0.038408
0.066625
0.088694
0.104640
0.101559
0.114605
0.127219
0.130527
0.164080

Results

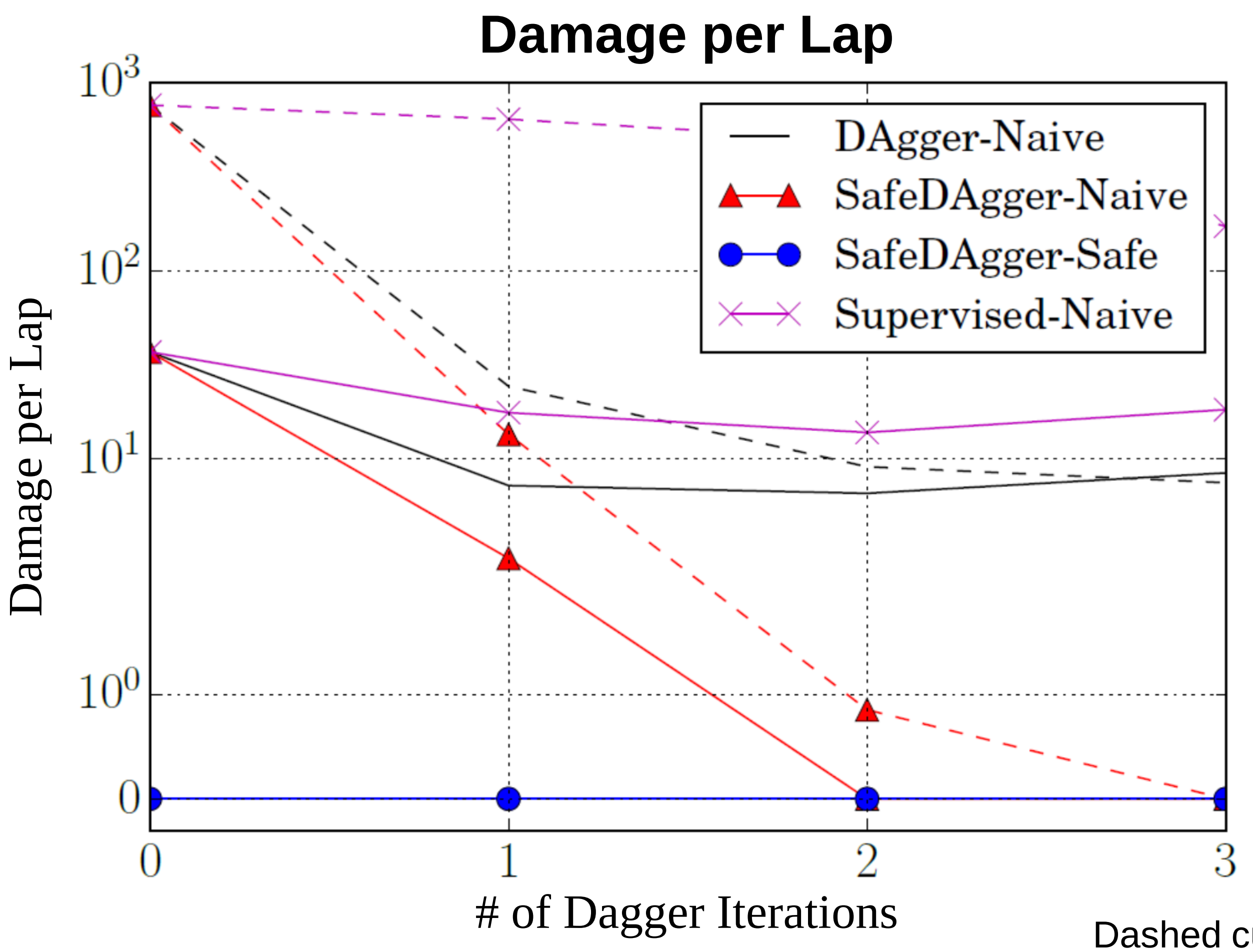
- Evaluation on test tracks
  1. Mean squared error of steering angle
  2. Damage per lap
  3. Number of laps
  4. Portion of time driven by a reference policy

Results



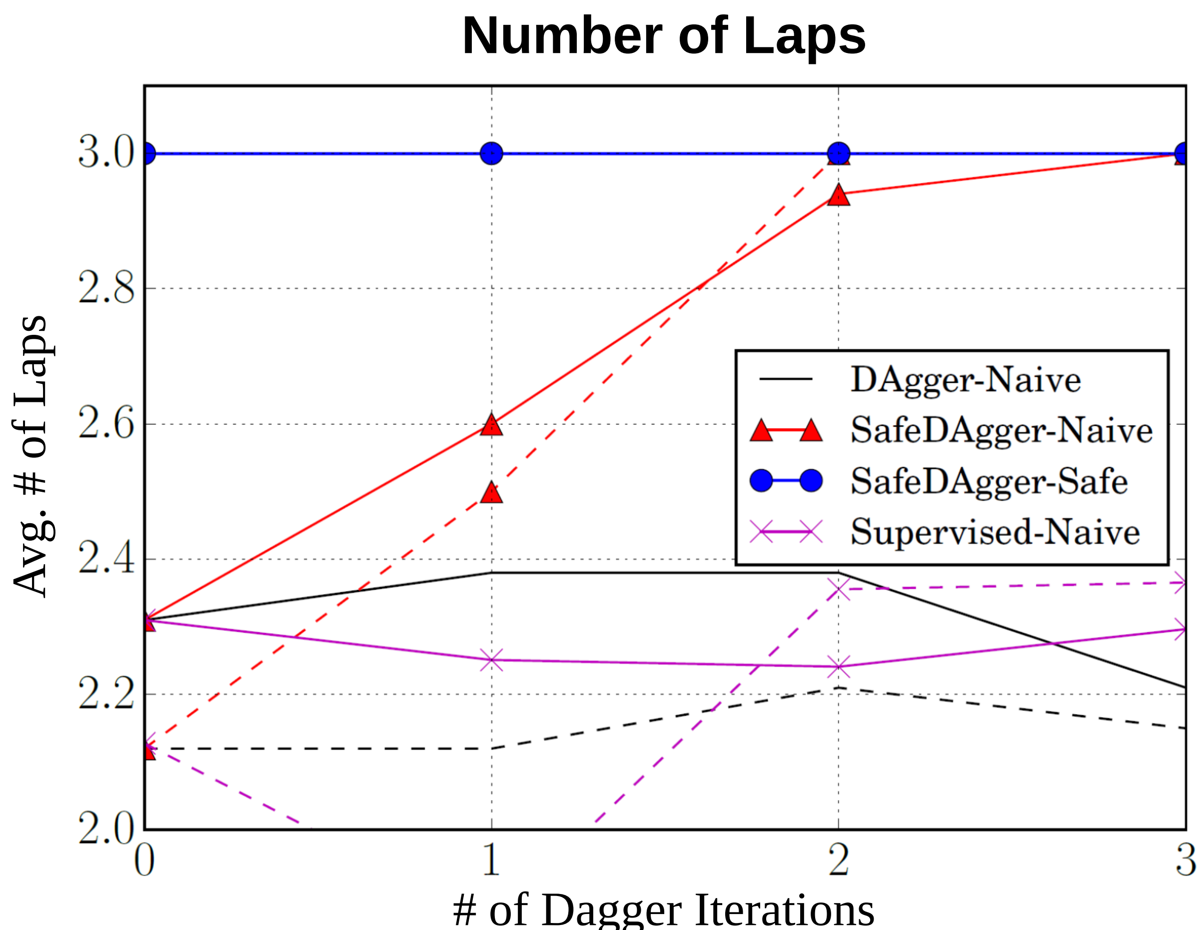
Dashed curve – with traffic
Solid curve – without traffic

Results



Dashed curve – with traffic
Solid curve – without traffic

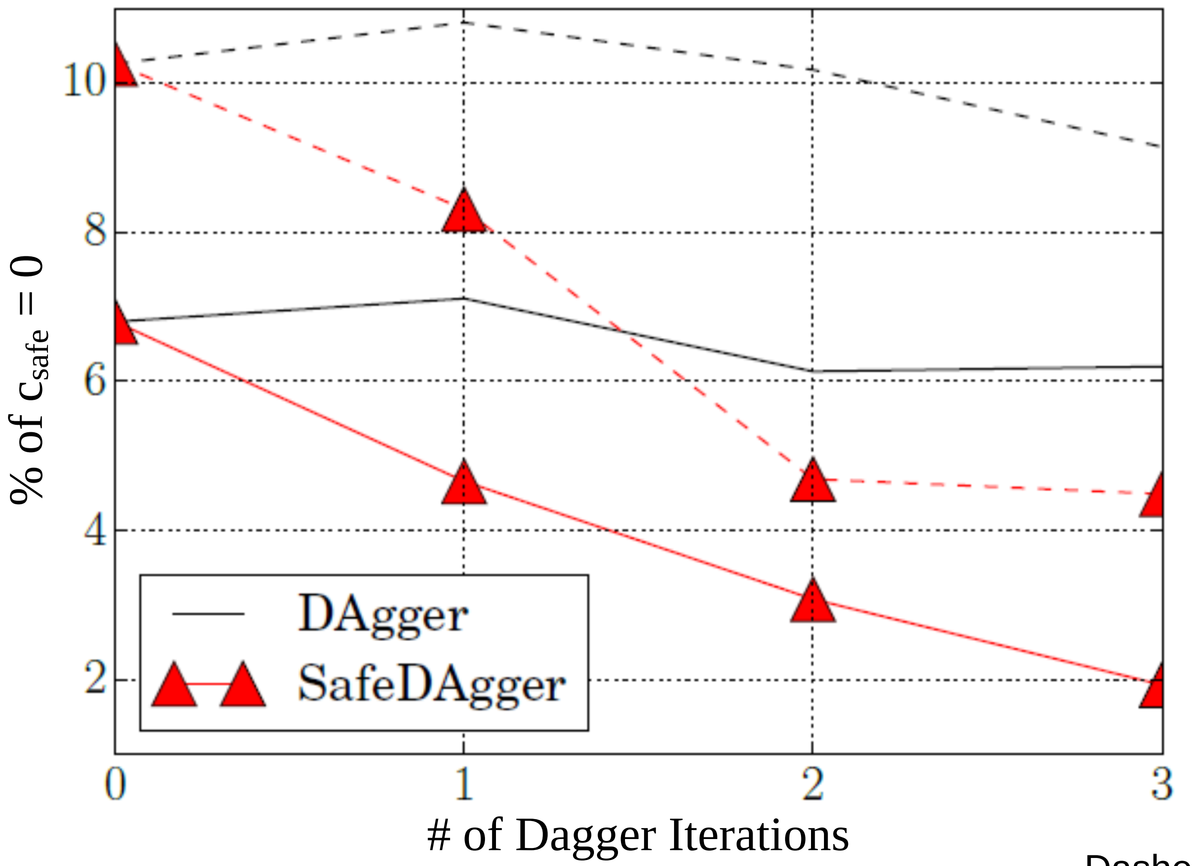
Results



Dashed curve – with traffic
Solid curve – without traffic

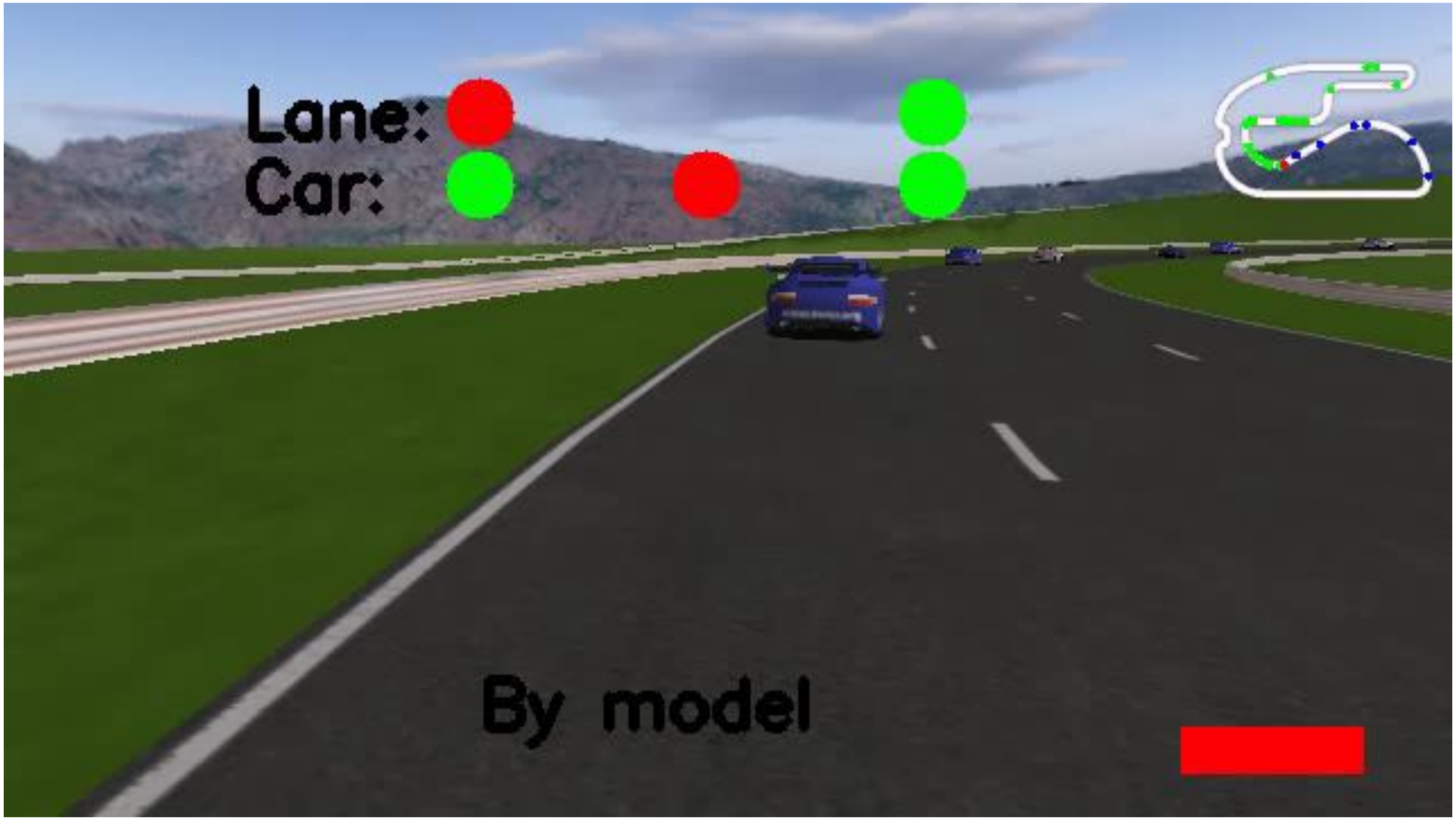
Results

Portion of time driven by a reference policy



Dashed curve – with traffic
Solid curve – without traffic

Demo



Conclusion

- Proposed SafeDAgger algorithm
  - Query efficient
  - Safety feature
- End-to-end simulated driving
  - Trained a convolutional neural network to drive in TORCS with traffic

Future work

- Evaluate SafeDAgger in the real world
- Learn to use temporal information