Towards Verified Artificial Intelligence

Sanjit A. Seshia

Professor

EECS, UC Berkeley

Joint work with

Dorsa Sadigh, **Tommaso Dreossi**, Alexander Donze, Anca Dragan, S. Shankar Sastry

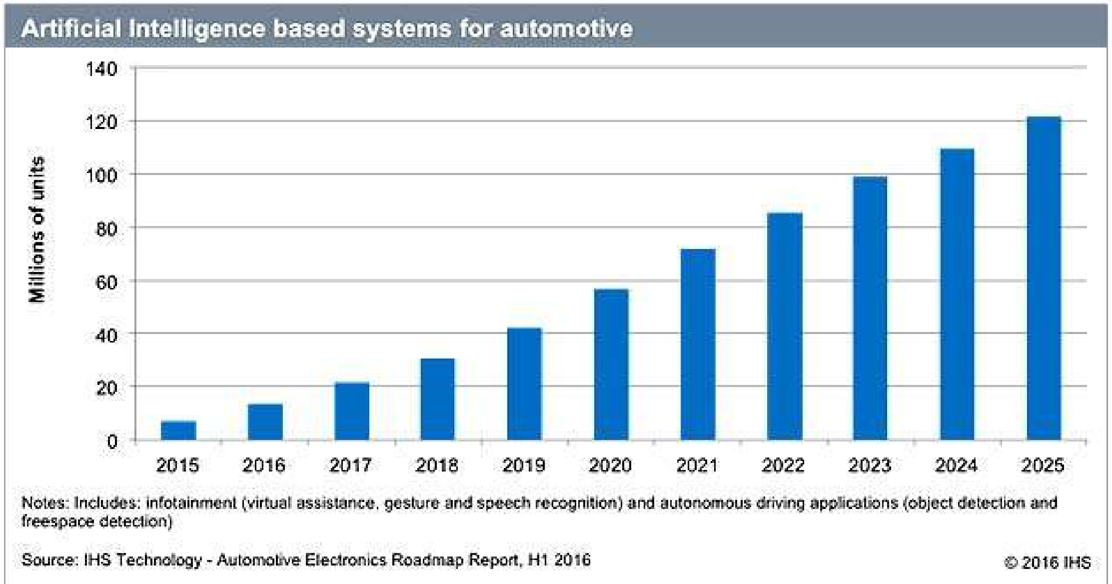
SETTA 2017

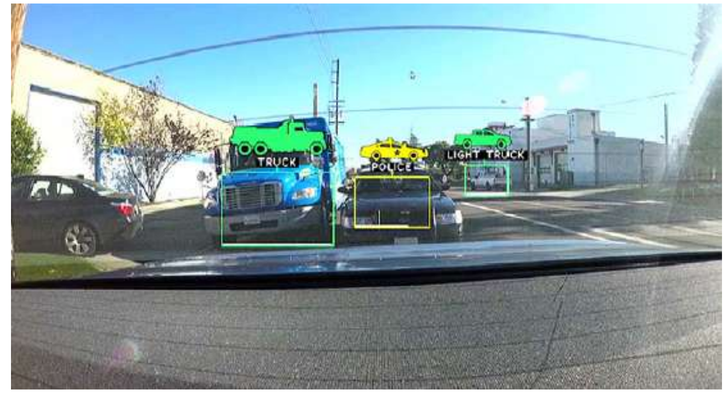
October 24, 2017

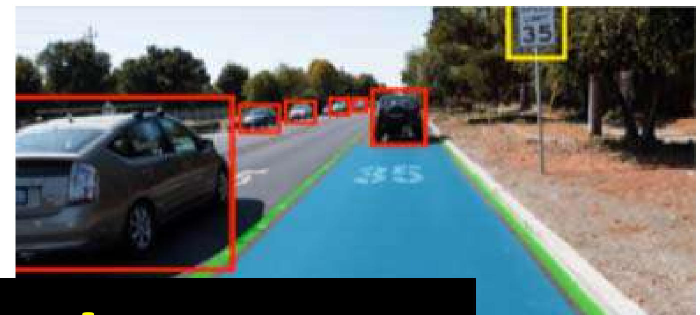
Human Cyber-Physical Systems (h-CPS)

"***Safety critical***" systems ***interacting*** with humans, often in a complex environment

Growing Use of Machine Learning/AI in Cyber-Physical Systems







Many Safety-Critical Systems

Artificial Intelligence (AI)

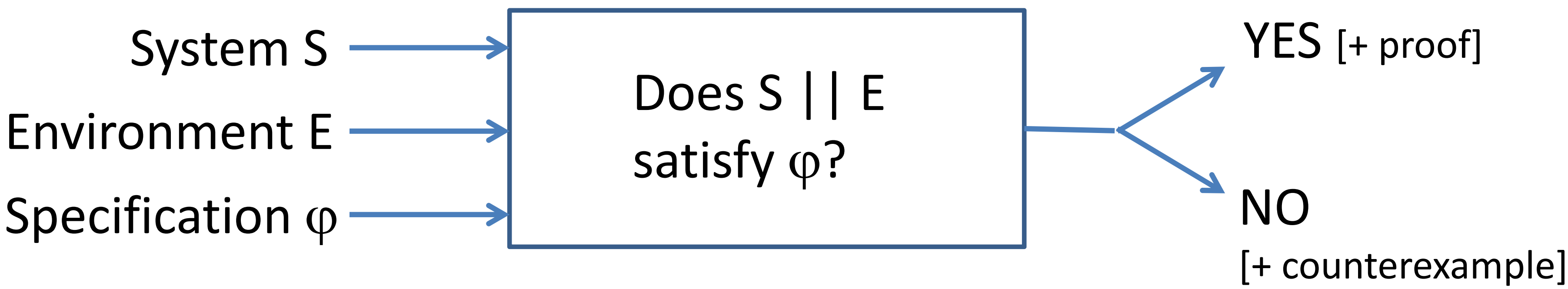
Computational Systems that attempt to mimic aspects of human intelligence, including especially the ability to learn from experience.

How do we ensure that AI-based systems are Dependable?

The Formal Methods Lens

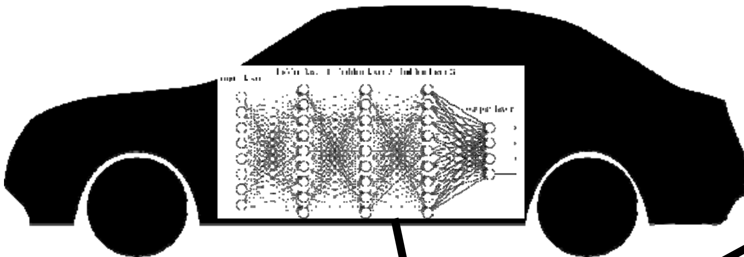
- Formal Methods ≈ Computational Proof methods
  – Specification/Modeling ≈ Statement of Conjecture/Theorem
  – Verification ≈ Proving/Disproving the Conjecture
  – Synthesis ≈ Generating (parts of) Conjecture/Proof
  ➢ Tools/techniques: SAT / SMT solvers, model checkers, theorem provers, simulation-based falsification, …

Verification:



Challenges for Verified AI

S. A. Seshia, D. Sadigh, S. S. Sastry. *Towards Verified Artificial Intelligence*. July 2016. https://arxiv.org/abs/1606.08514.



System **S**

Environment **E**

Specification φ

Does S || E satisfy φ?

YES [+ proof]

NO [+ counterexample]

Design Correct-by-Construction instead?

Counterexamples, etc. from Rich Signal Spaces?

Talk Outline

- Environment Modeling Challenge
  - ➢ Interaction-Aware Control for Human-CPS

- Specification (& Verification) Challenge
  - ➢ Verifying Robustness (of Interaction-Aware Controller)
  - ➢ Falsification for Deep Learning based CPS

- Conclusions and Future Directions
  - – Towards a New Design Methodology for AI-based Systems

Environment Modeling Challenge – Uncertainty and Unknowns

Self-Driving Vehicles: Interact with Humans in Complex Environments; Significant use of machine learning!

Known Unknowns and Unknown Unknowns!!

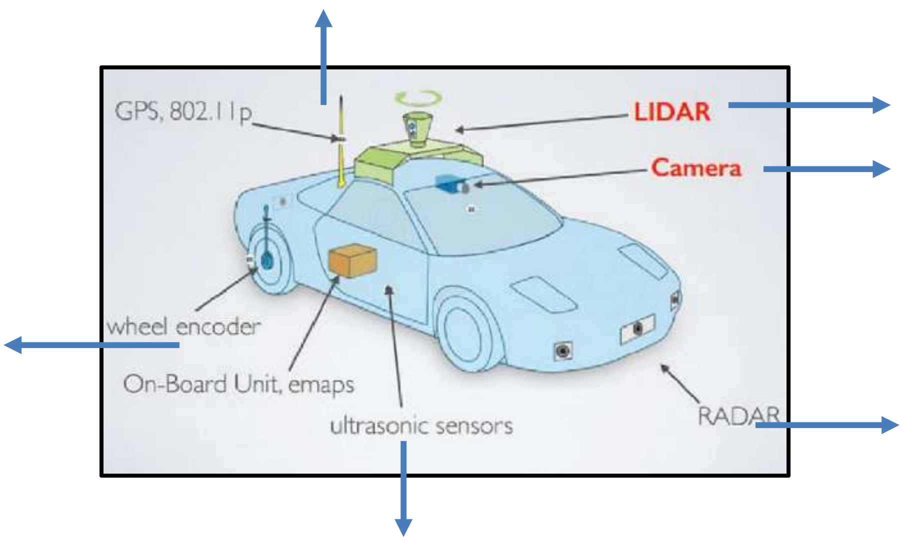
Cannot represent all possible environment scenarios

Idea 1: Introspective Environment Modeling

Impossible to model all possible scenarios

Approach: ***Introspect on System to Model the Environment***

Identify: (i) **Interface** between System & Environment,
(ii) (Weakest) **Assumptions** needed to Guarantee Safety/Correctness



Algorithmic techniques to *generate weakest interface assumptions* and *monitor them at run-time* for potential violation/mitigation

[Li, Sadigh, Sastry, Seshia; TACAS'14]

Idea 2: Active Data Gathering and Learning

Monitor and Interact with the Environment,
Offline and Online, to Model It.

Google's Driverless Cars Run Into Problem: Cars With Drivers

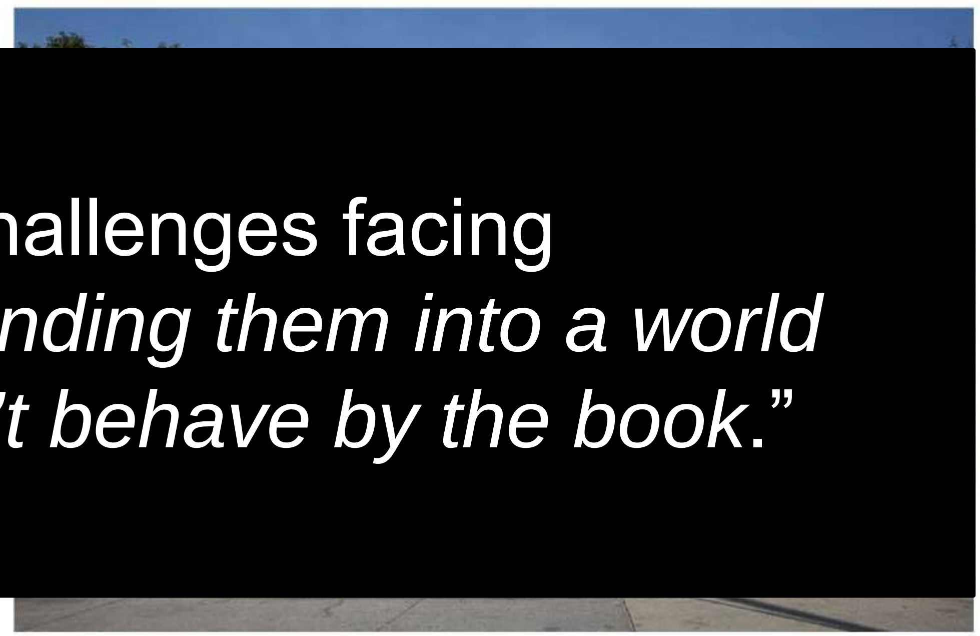
By MATT RICHTEL and CONOR DOUGHERTY SEPT. 1, 2015

Email

MOUNTAIN VIEW, Calif. — Google, a leader in

"One of the biggest challenges facing automated cars is *blending them into a world in which humans don't behave by the book.*"

it can be tough to get around if you are a stickler for the rules. One Google car, in a test in 2009, couldn't get through a four-way stop because its sensors kept waiting for other (human) drivers

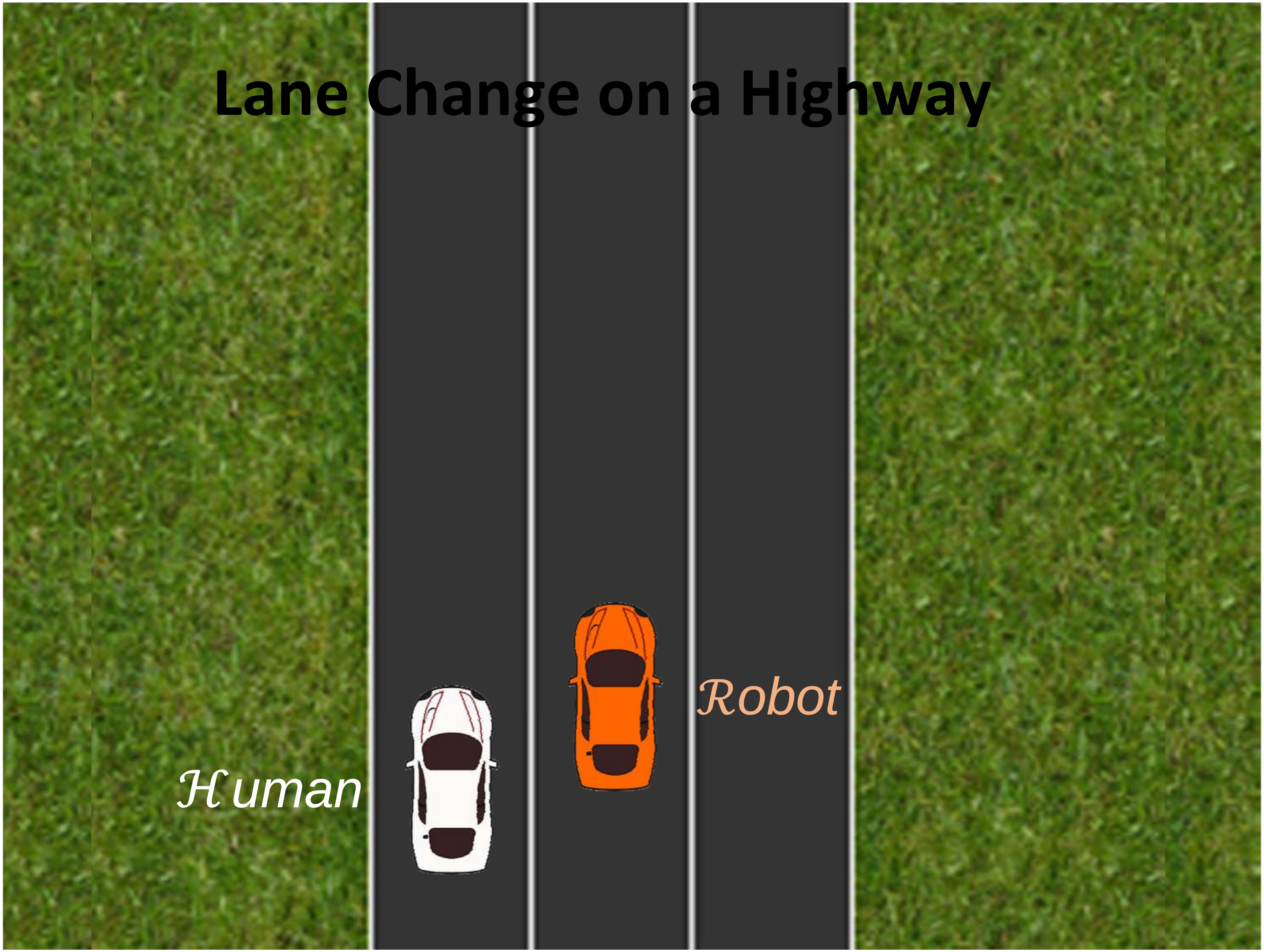


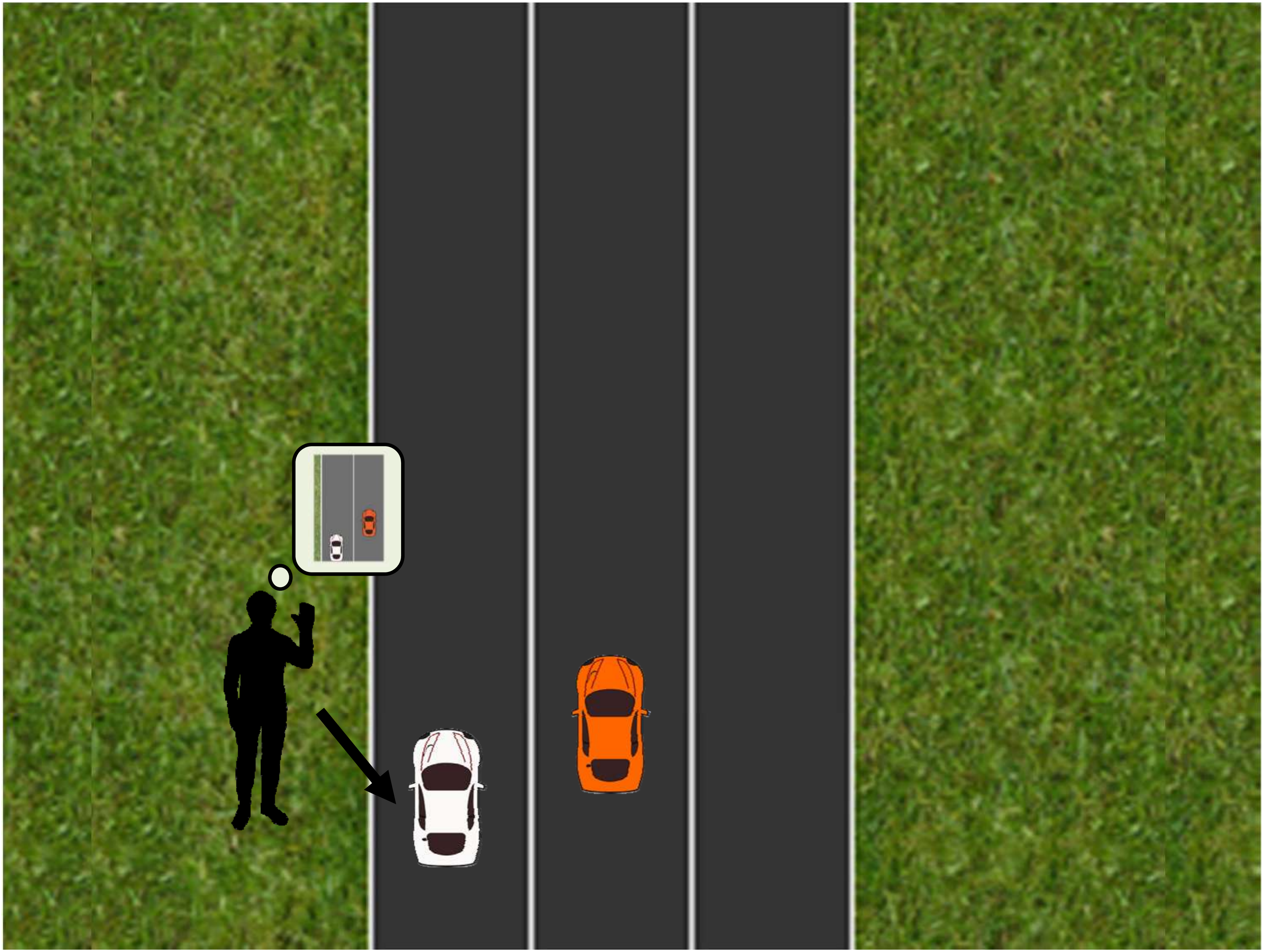
The Google self-driving car, with Eric Schmidt, left, the company's executive chairman, and Transportation Secretary Anthony Foxx. Justin Sullivan/Getty Images

Interaction-Aware Control

- D. Sadigh, S. Sastry, S. A. Seshia, A. Dragan. *Planning for Autonomous Cars that Leverages Effects on Human Actions*. In RSS, 2016.
- D. Sadigh, S. Sastry, S. A. Seshia, A. Dragan. *Information Gathering Actions over Internal Human State*. In IROS, 2016.
- D. Sadigh, A. Dragan, S. Sastry, S. A. Seshia. *Active Preference-Based Learning of Reward Functions.* In RSS, 2017.

Lane Change on a Highway



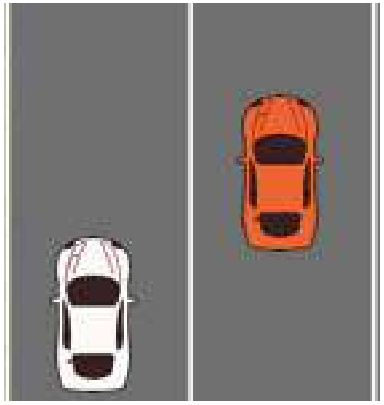


Interaction as a Dynamical System

$$x^{t+1} = f_{\mathcal{H}}(f_{\mathcal{R}}(x^t, u^t_{\mathcal{R}}), u^t_{\mathcal{H}})$$

Robot actions

$\boldsymbol{u}_R$



Human actions $\boldsymbol{u}_H$

Model the problem as a *Stackelberg (turn-based) Game*. Robot moves first.

Assumptions/Simplifications

Model Predictive (Receding Horizon) Control:

Optimize over short time horizon N, replan at every step t.

$$R_{\mathcal{R}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{R}}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t) \qquad R_{\mathcal{H}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{H}}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t)$$

Assume *deterministic "rational"* human model, human optimizes reward function which is a linear combination of "features".

Human has full access to $\boldsymbol{u}_{\mathcal{R}}$ for the short time horizon.

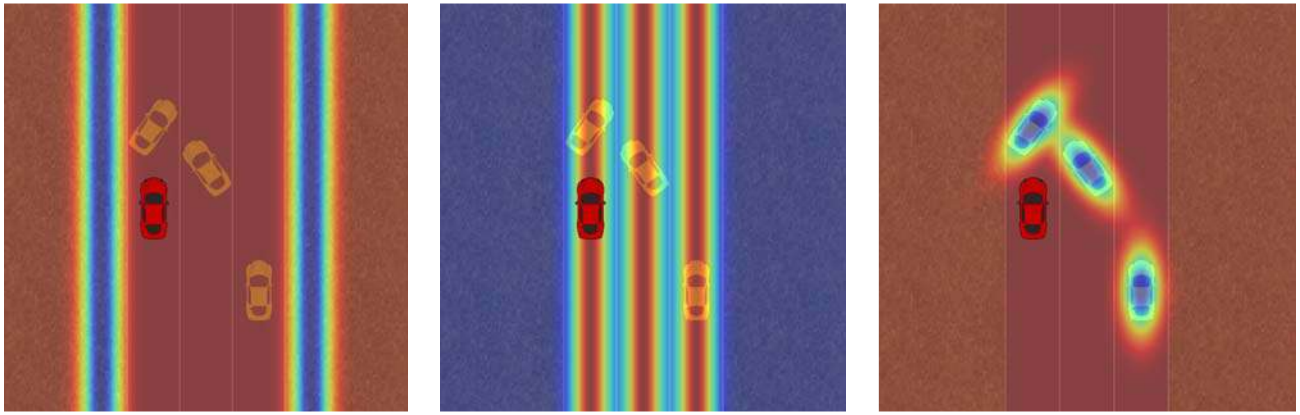
$$\boldsymbol{u}_H^*(x_0, \boldsymbol{u}_R) = \operatorname*{argmax}_{\boldsymbol{u}_H} R_H(x_0, \boldsymbol{u}_R, \boldsymbol{u}_H)$$

Learning (Human) Driver Models

Learn Human's reward function based on Inverse Reinforcement Learning [Ziebart et al, AAAI'08; Levine & Koltun, 2012].

Assume structure of human reward function:

$$r_H(x^t, u_R^t, u_H^t) = w^\top \phi(x^t, u_R^t, u_H^t)$$



(a) Features for the boundaries of the road

(b) Feature for staying inside the lanes.

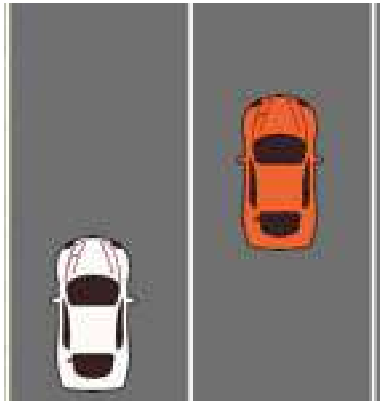
(c) Features for avoiding other vehicles.

B. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In AAAI, 2008.
S. Levine, V. Koltun. Continuous inverse optimal control with locally optimal examples. arXiv , 2012.

Interaction as a Dynamical System

$$\boldsymbol{u}_R^* = \underset{\boldsymbol{u}_R}{\operatorname{argmax}} R_R(x_0, \boldsymbol{u}_R, \boldsymbol{u}_H^*(x_0, \boldsymbol{u}_R))$$

Model $\boldsymbol{u}_H^*$ as optimizing the human reward function R_H.



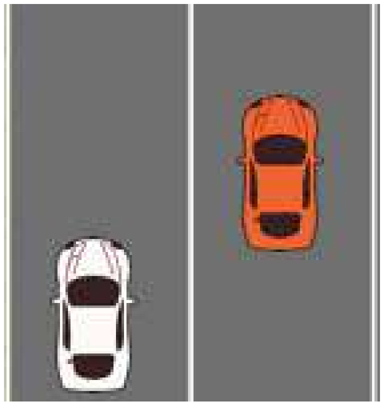
Find optimal actions for the autonomous vehicle while accounting for the human response $\boldsymbol{u}_H^*$.

$$\boldsymbol{u}_H^*(x_0, \boldsymbol{u}_R) = \underset{\boldsymbol{u}_H}{\operatorname{argmax}} R_H(x_0, \boldsymbol{u}_R, \boldsymbol{u}_H)$$

Solution of Nested Optimization

$$u_{\mathcal{R}}^* = \underset{u_{\mathcal{R}}}{\operatorname{argmax}} R_{\mathcal{R}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}^*(x, u_{\mathcal{R}}))$$

$$R_{\mathcal{R}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{R}}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t)$$



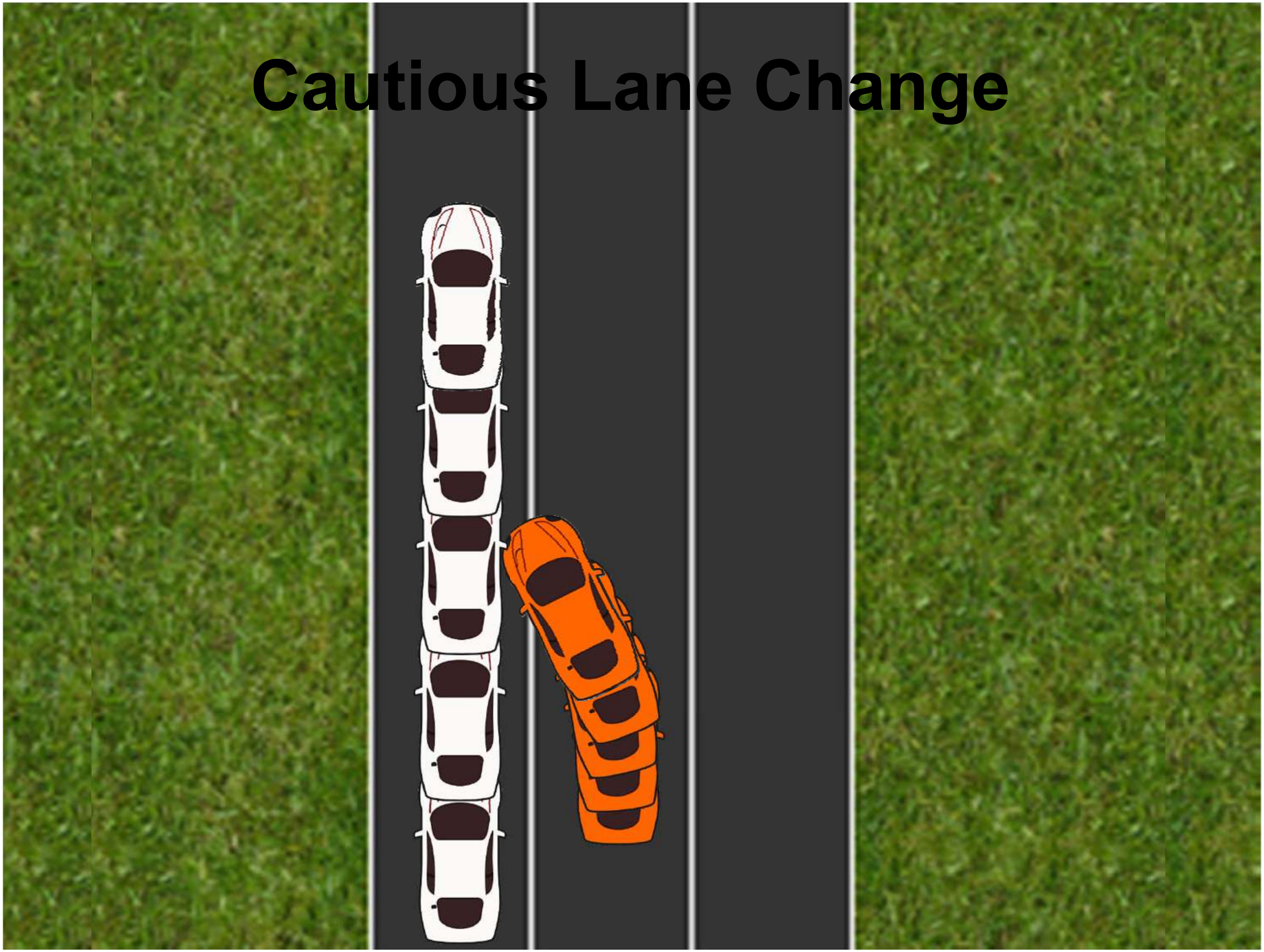
Gradient-Based Method (Quasi-Newton): (solve using L-BFGS technique)

$$\begin{cases} R_{\mathcal{R}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}^*) \\ \dfrac{\partial R_{\mathcal{R}}}{\partial u_{\mathcal{R}}} = \dfrac{\partial R_{\mathcal{R}}}{\partial u_{\mathcal{H}}} \dfrac{\partial u_{\mathcal{H}}^*}{\partial u_{\mathcal{R}}} + \dfrac{\partial R_{\mathcal{R}}}{\partial u_{\mathcal{R}}} \end{cases}$$

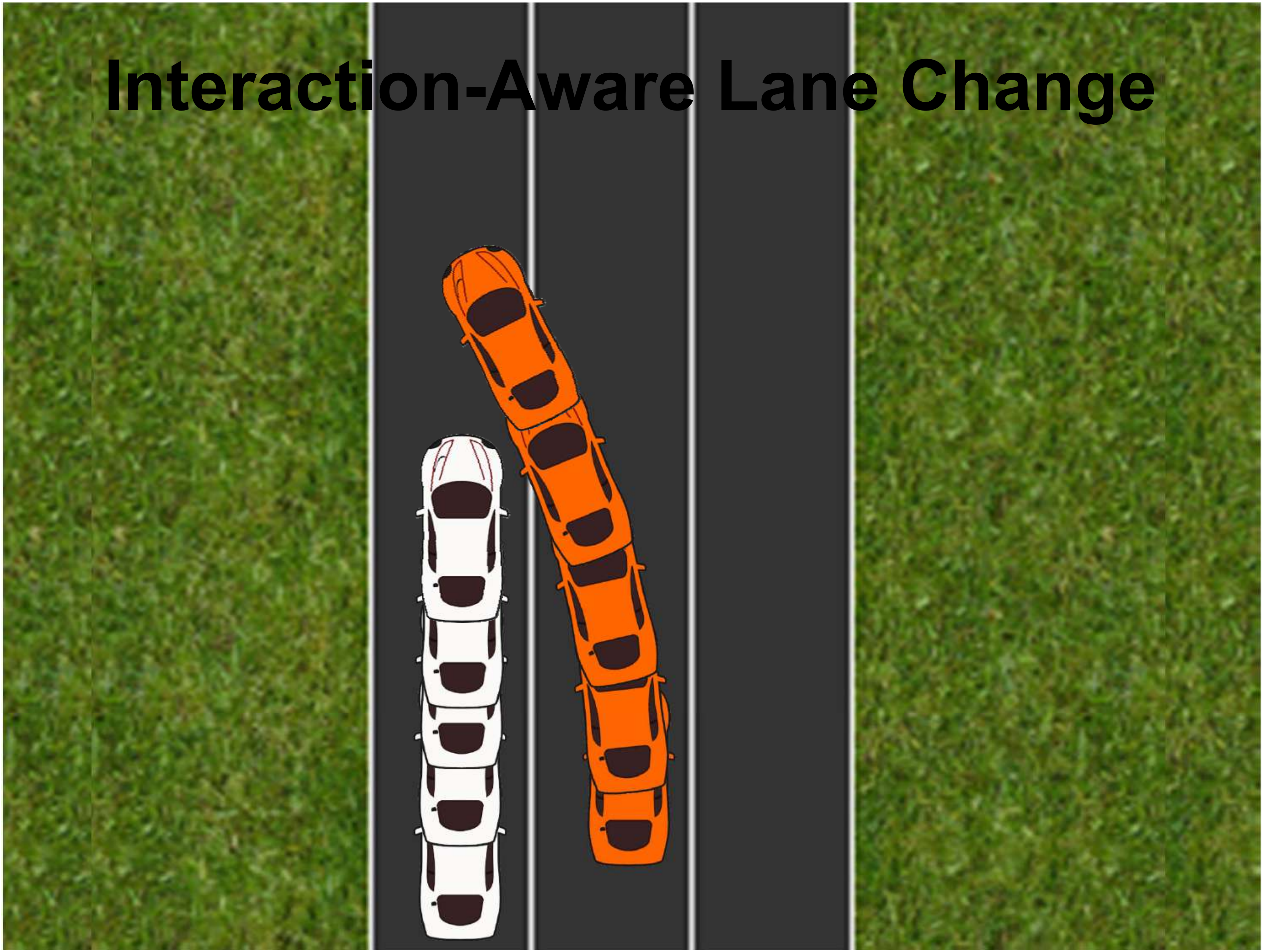
$$u_{\mathcal{H}}^*(x, u_{\mathcal{R}}) \approx \underset{u_{\mathcal{H}}}{\operatorname{argmax}} R_{\mathcal{H}}(x, u_{\mathcal{R}}, u_{\mathcal{H}})$$

$$R_{\mathcal{H}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}) = \sum_{t=1}^{N} r_{\mathcal{H}}(x^t, u_{\mathcal{R}}^t, u_{\mathcal{H}}^t)$$

Cautious Lane Change



Interaction-Aware Lane Change



Aggressive Driver

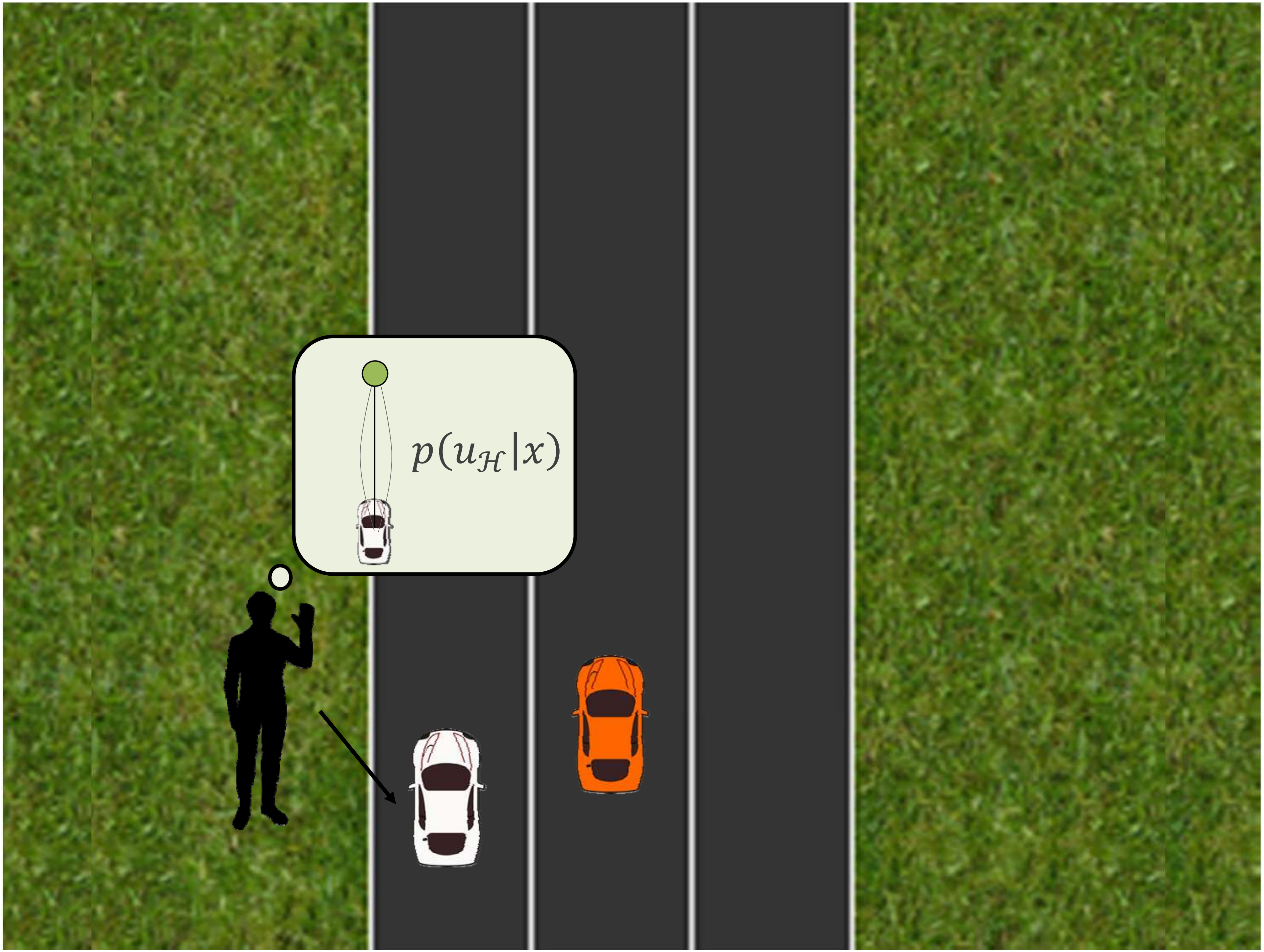
Distracted Driver

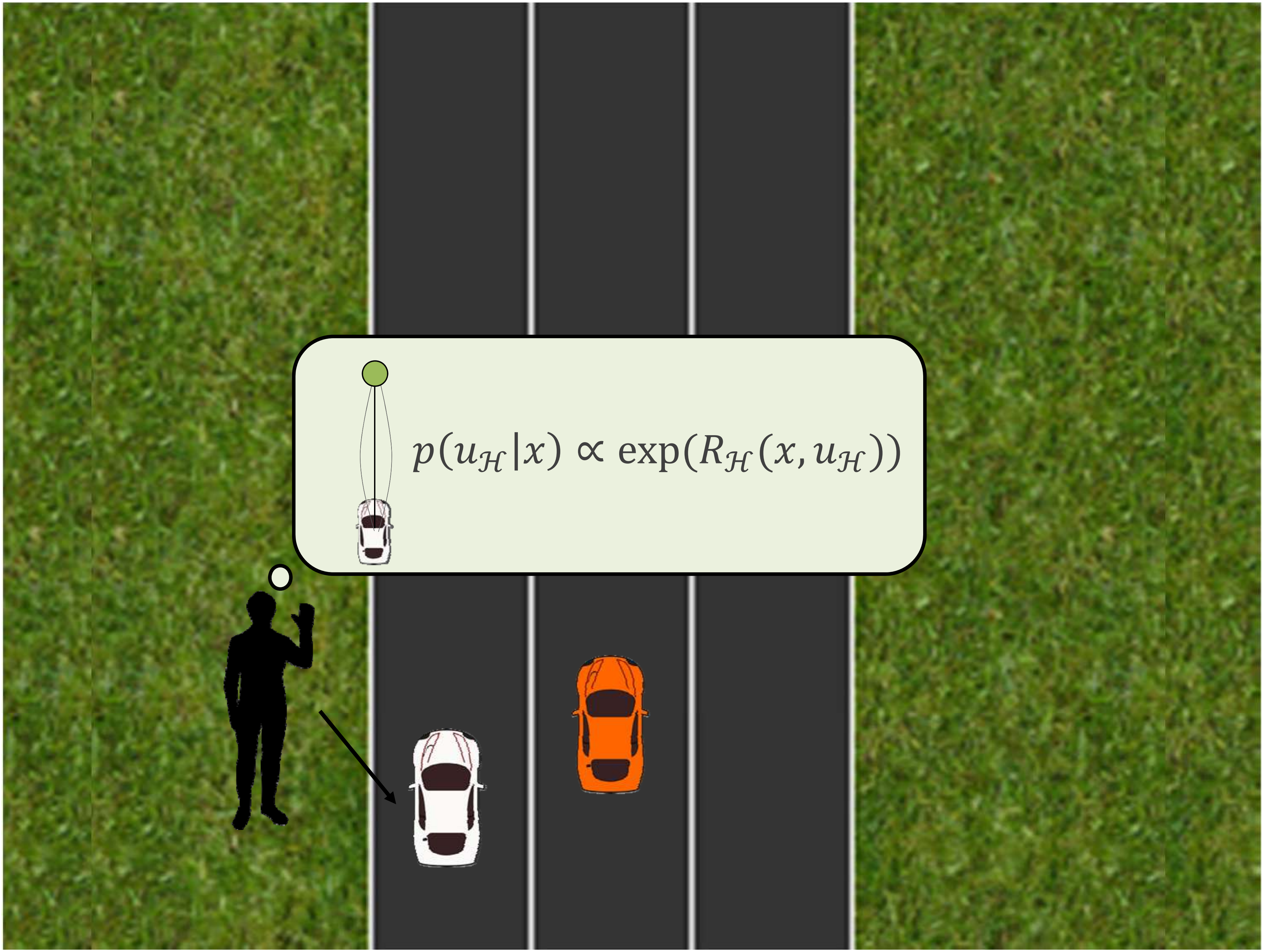
Cautious Driver

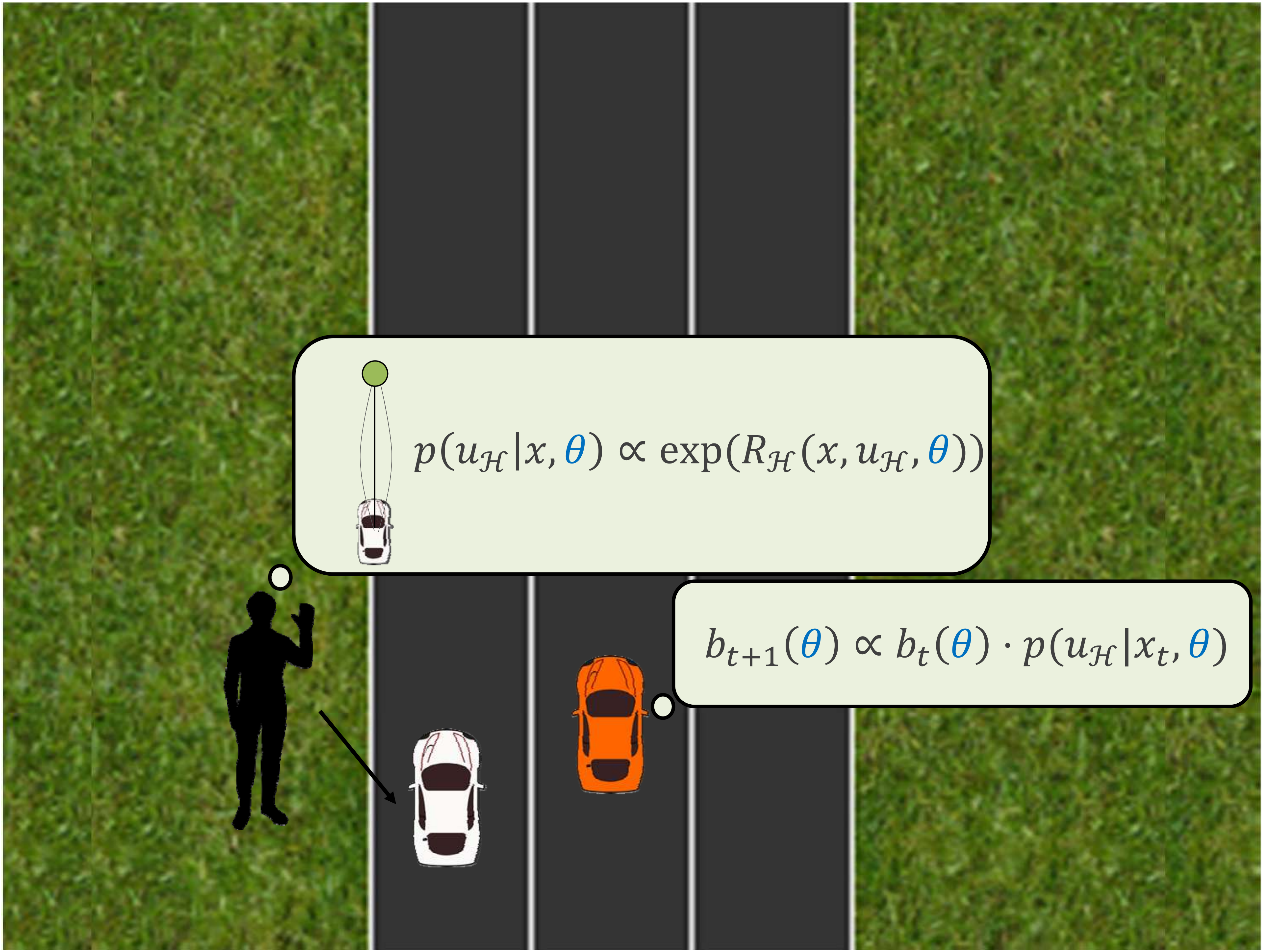
Attentive Driver

We can’t rely on a
single driver model.

We need to differentiate
between different drivers.

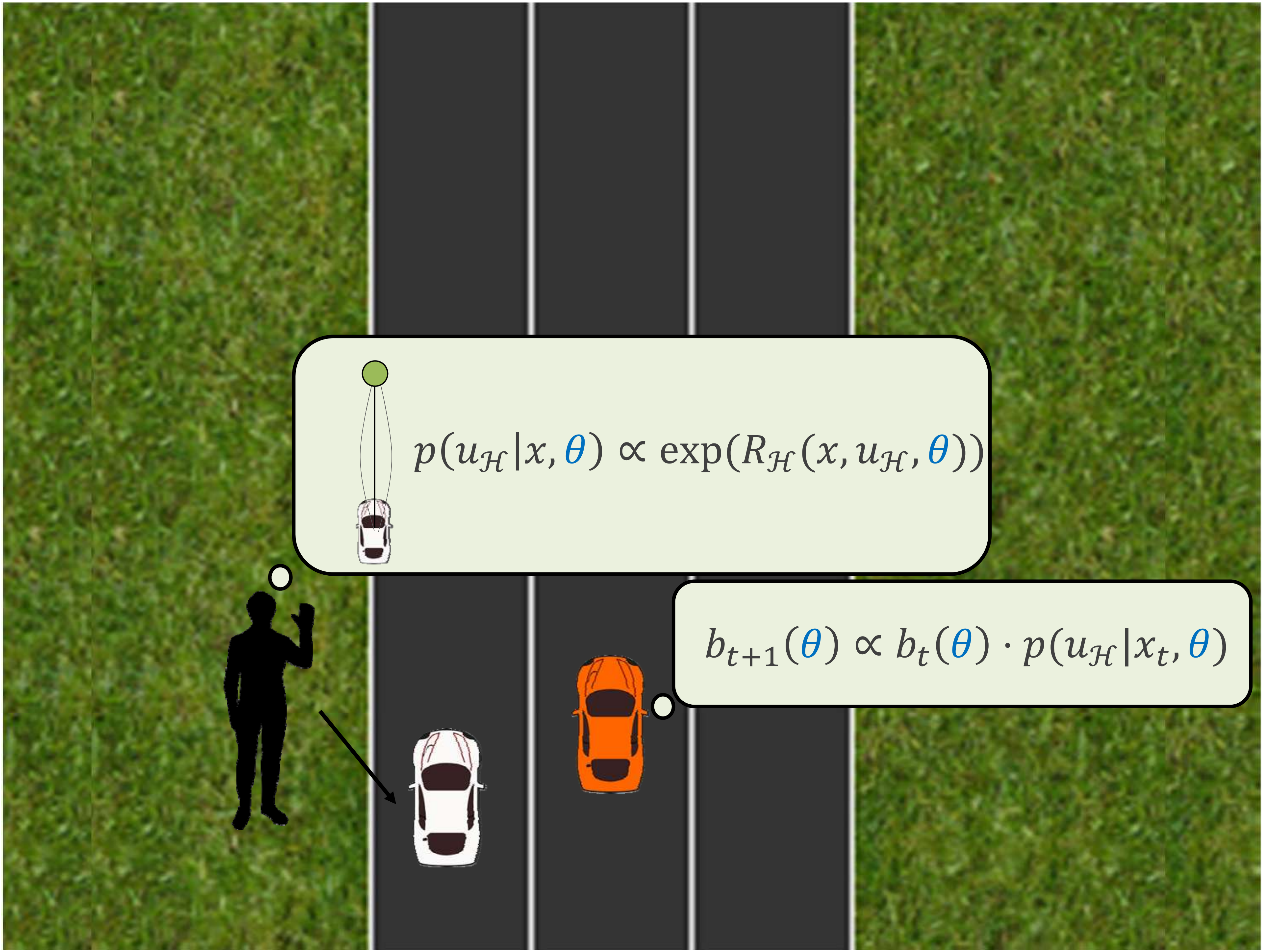

$p(u_{\mathcal{H}}|x)$


$p(u_{\mathcal{H}}|x) \propto \exp(R_{\mathcal{H}}(x, u_{\mathcal{H}}))$

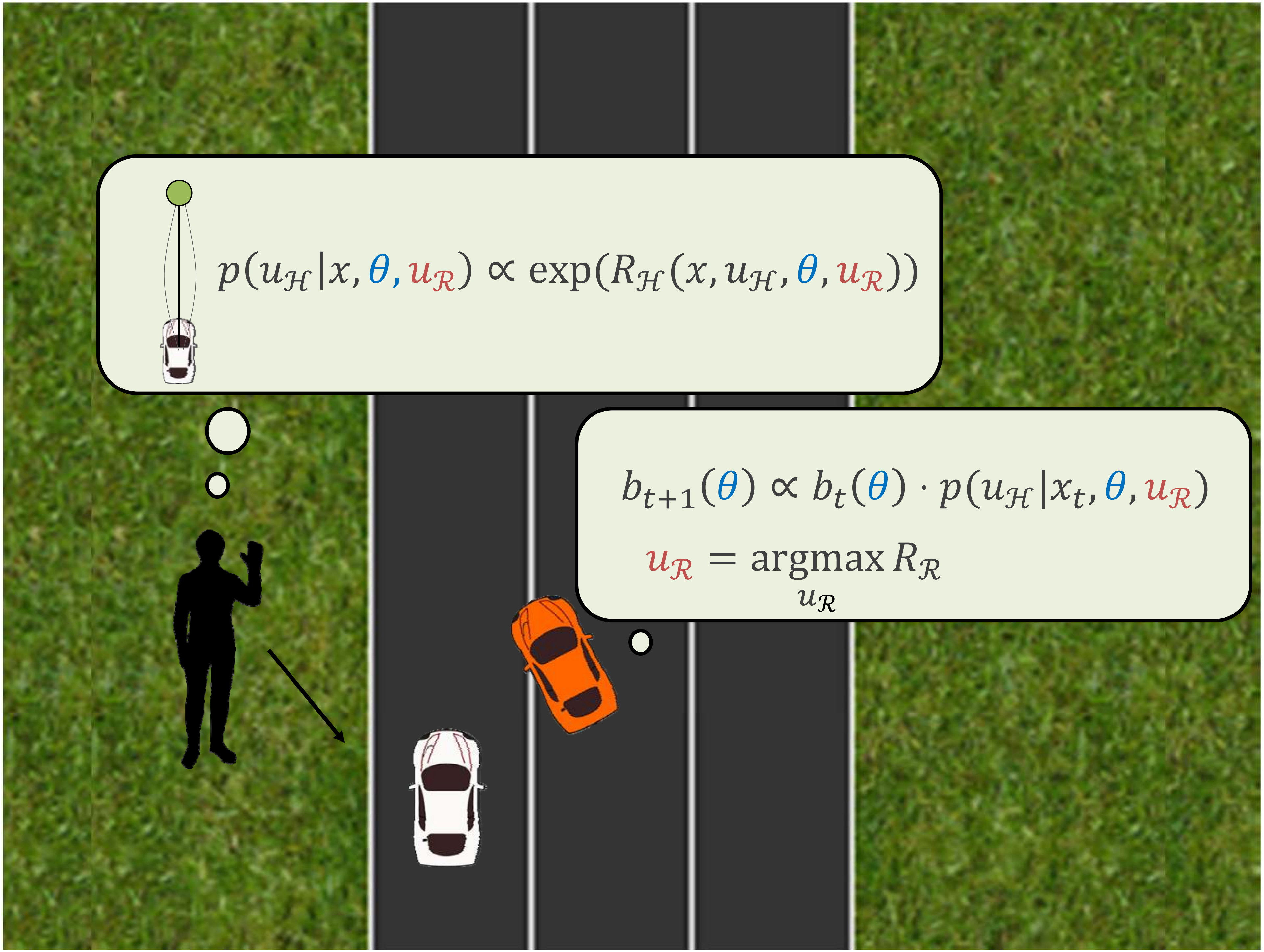


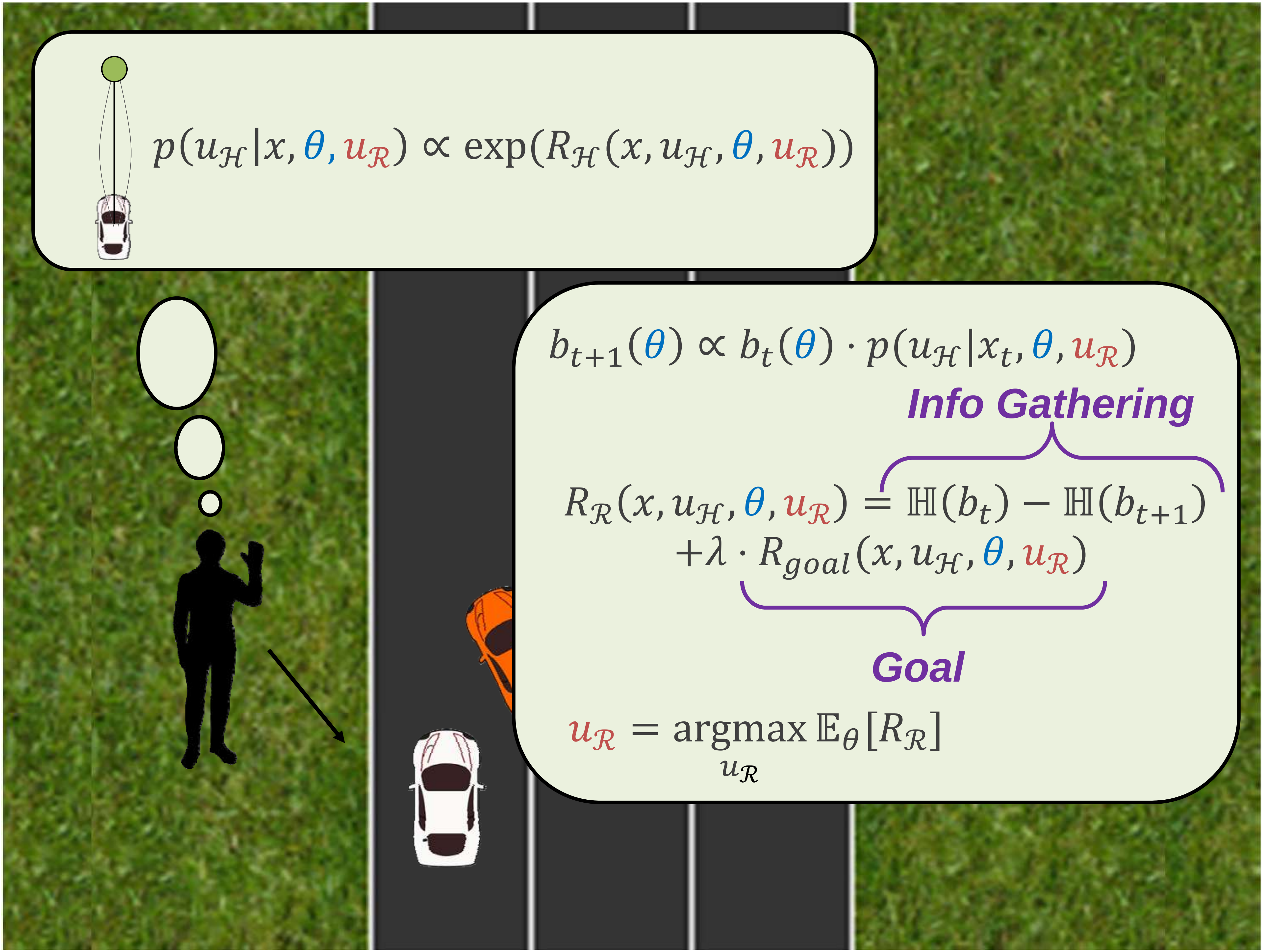
$p(u_{\mathcal{H}}|x,\theta) \propto \exp(R_{\mathcal{H}}(x,u_{\mathcal{H}},\theta))$
$b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_{\mathcal{H}}|x_t,\theta)$


gifsboom
.net



$p(u_{\mathcal{H}} | x, \theta) \propto \exp(R_{\mathcal{H}}(x, u_{\mathcal{H}}, \theta))$
$b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_{\mathcal{H}} | x_t, \theta)$

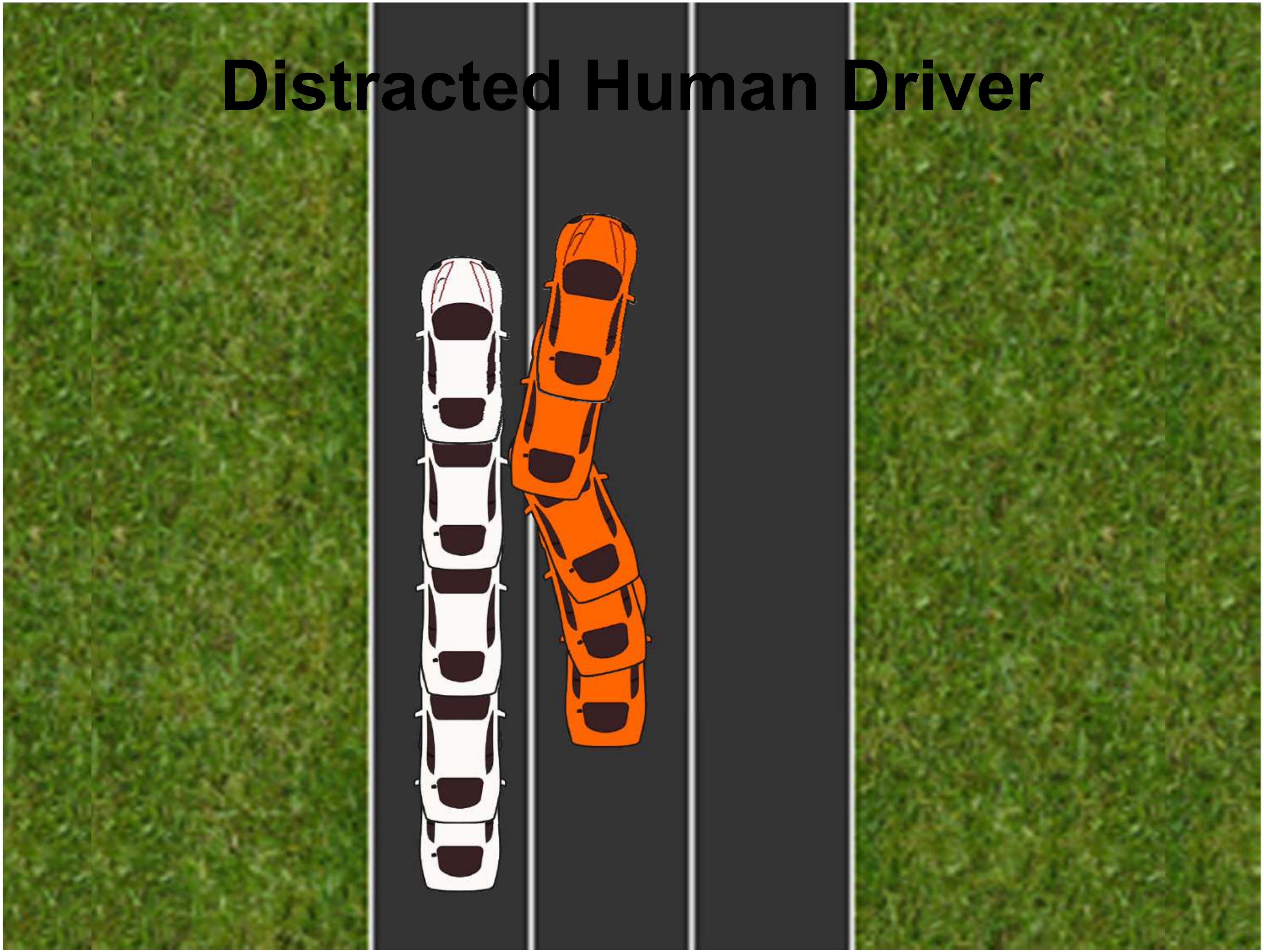

$p(u_{\mathcal{H}}|x,\theta,u_{\mathcal{R}}) \propto \exp(R_{\mathcal{H}}(x,u_{\mathcal{H}},\theta,u_{\mathcal{R}}))$
$b_{t+1}(\theta) \propto b_t(\theta)\cdot p(u_{\mathcal{H}}|x_t,\theta,u_{\mathcal{R}})$
$u_{\mathcal{R}} = \operatorname{argmax}_{u_{\mathcal{R}}} R_{\mathcal{R}}$



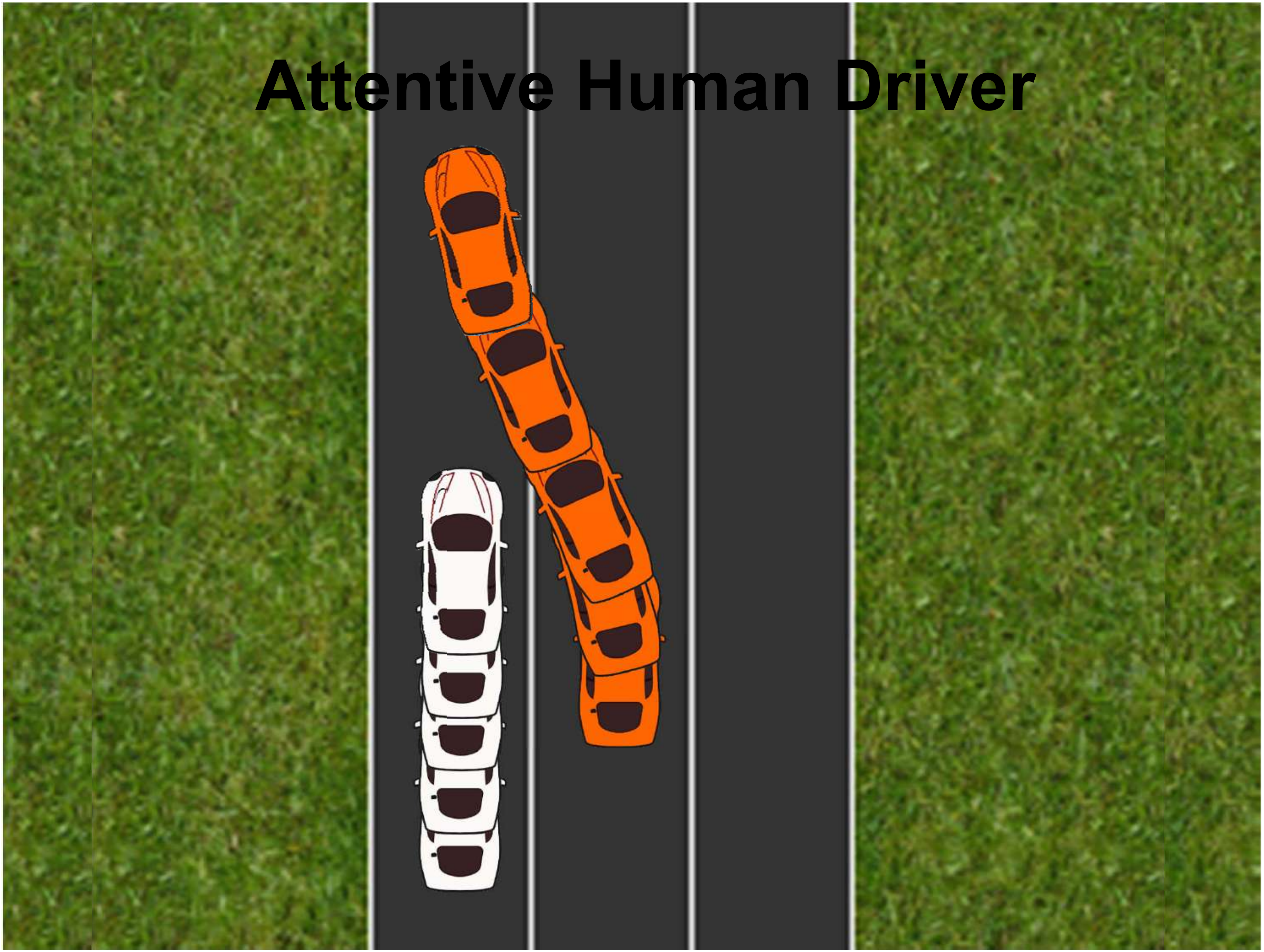
$p(u_{\mathcal{H}} | x, \theta, u_{\mathcal{R}}) \propto \exp(R_{\mathcal{H}}(x, u_{\mathcal{H}}, \theta, u_{\mathcal{R}}))$
$b_{t+1}(\theta) \propto b_t(\theta) \cdot p(u_{\mathcal{H}} | x_t, \theta, u_{\mathcal{R}})$
Info Gathering
$R_{\mathcal{R}}(x, u_{\mathcal{H}}, \theta, u_{\mathcal{R}}) = \overbrace{\mathbb{H}(b_t) - \mathbb{H}(b_{t+1})}^{\text{Info Gathering}} + \lambda \cdot \underbrace{R_{goal}(x, u_{\mathcal{H}}, \theta, u_{\mathcal{R}})}_{\text{Goal}}$
Goal
$u_{\mathcal{R}} = \underset{u_{\mathcal{R}}}{\operatorname{argmax}} \mathbb{E}_{\theta}[R_{\mathcal{R}}]$

Nudging in for Active Info Gathering

Nudging in for Active Info Gathering


Distracted Human Driver

Attentive Human Driver



Key Ideas:

Actively gather data about the environment (human) by *affecting* the environment's behavior

Learn environment (human) model from data, update online

Questions:

- How to verify such human-robot systems?
- What are more realistic human models? (e.g. "bounded rationality")

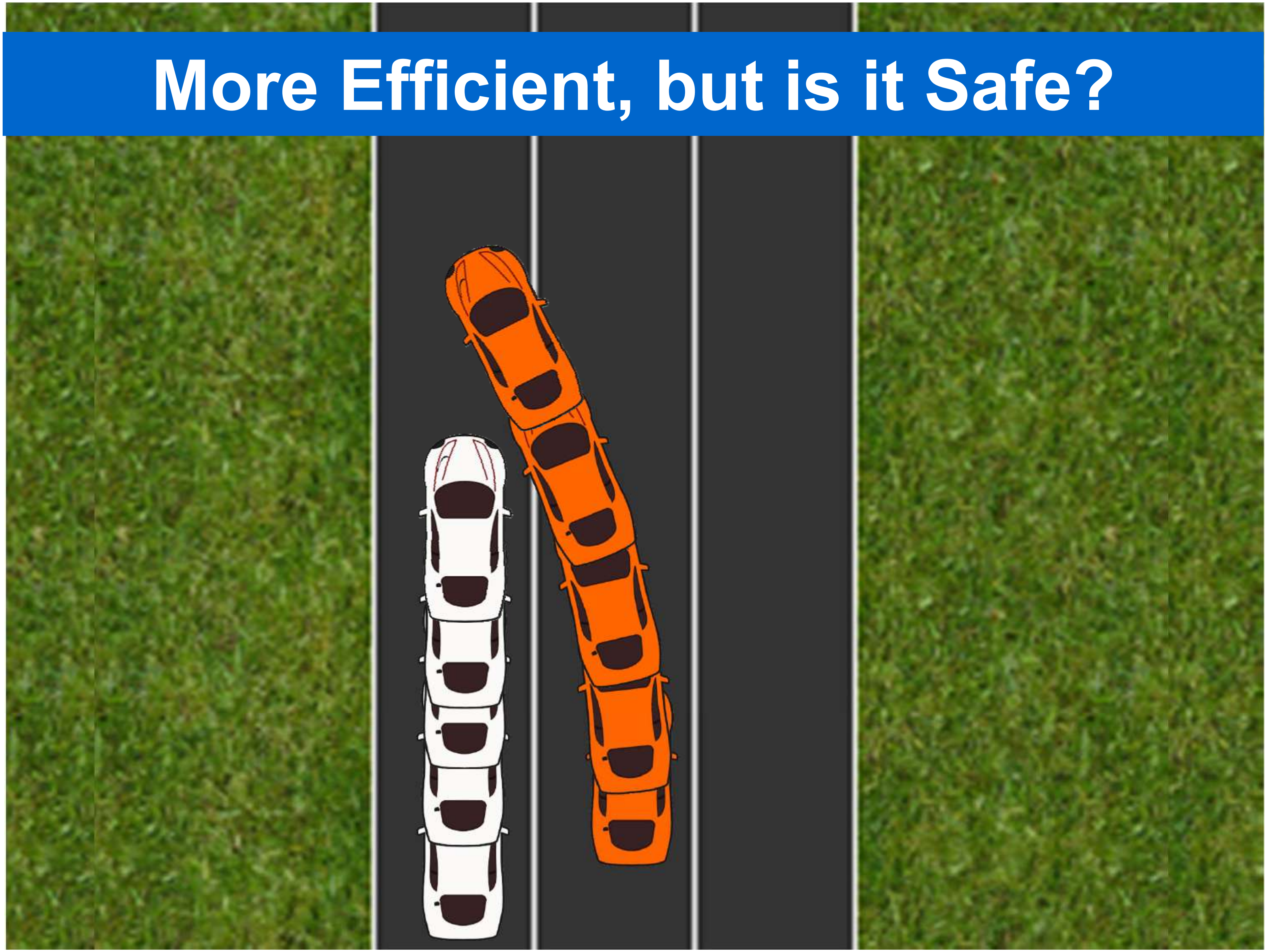
Talk Outline

- Environment Modeling Challenge
  - ➢ Interaction-Aware Control for Human-CPS

- Specification (& Verification) Challenge
  - ➢ Verifying Robustness (of Interaction-Aware Controller)
  - ➢ Falsification for Deep Learning based CPS

- Conclusions and Future Directions
  - – Towards a New Design Methodology for AI-based Systems

More Efficient, but is it Safe?



Verifying Temporal Logic Requirements

Signal Temporal Logic (STL) [Maler & Nickovic, '04]

Predicates over continuous signals, Propositional Formulas φ ($\wedge, \vee, \neg$ of the predicates), Temporal Operators ($\boldsymbol{G}, \boldsymbol{F}, \boldsymbol{X}, \boldsymbol{U}$), real-time interval τ.

$\boldsymbol{G}_\tau\, \varphi$	φ is true at all future moments in τ.
$\boldsymbol{F}_\tau\, \varphi$	φ is true in some future moment in τ.
$\varphi_1 \boldsymbol{U}_\tau\, \varphi_2$	φ_1 is true until φ_2 becomes true in τ.

Safety (invariance): Vehicle maintains specified distance from obstacles.

$$\boldsymbol{G}_{[0,\tau]}\,[\ \text{dist(vehicle, obstacle)} > \Delta]$$

From Logical Formulas to Objective Functions

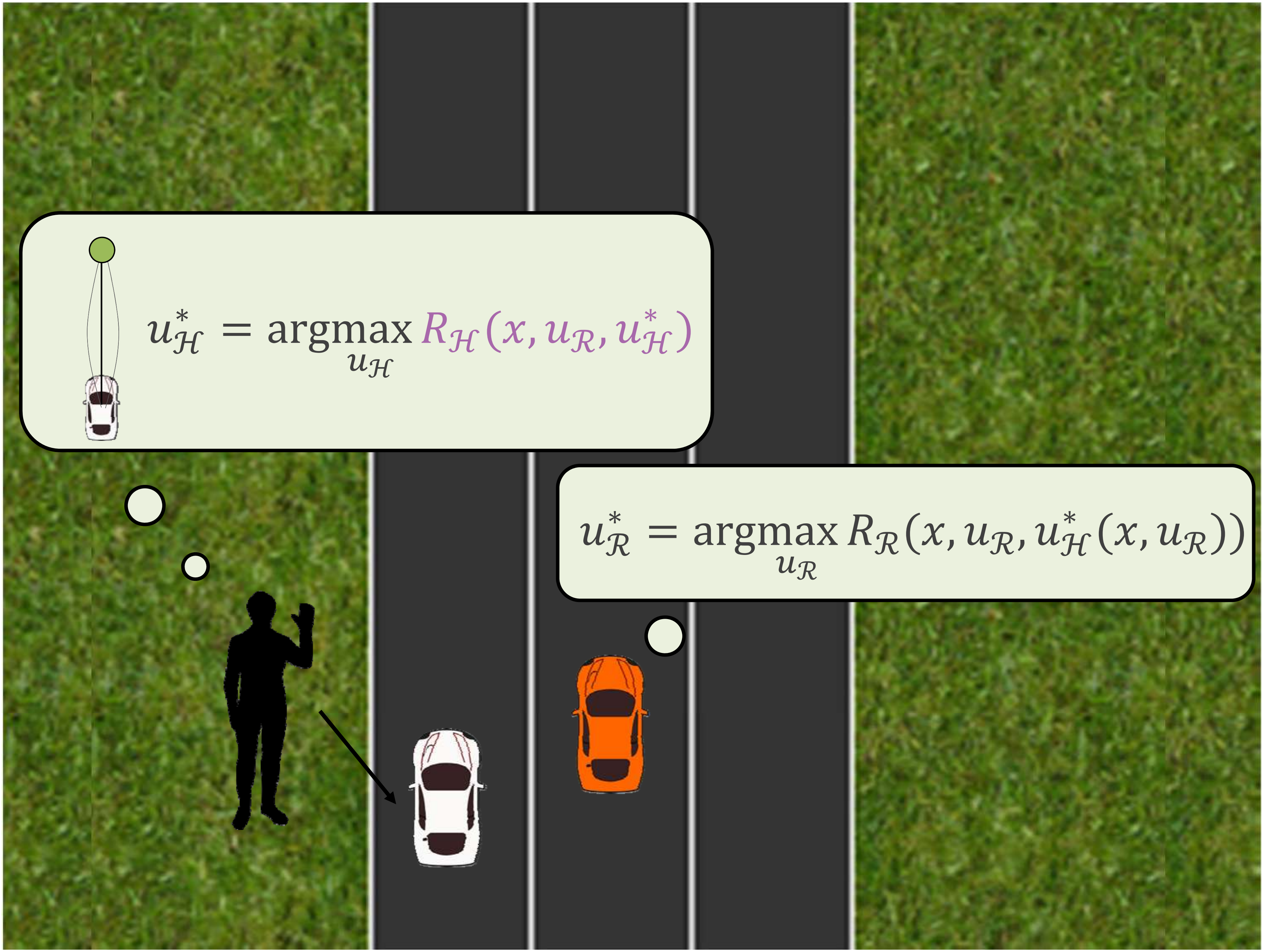
- STL formula has both
  - Boolean semantics: true/false
  - Quantitative semantics: value in $\mathbb{R}$

- Example:

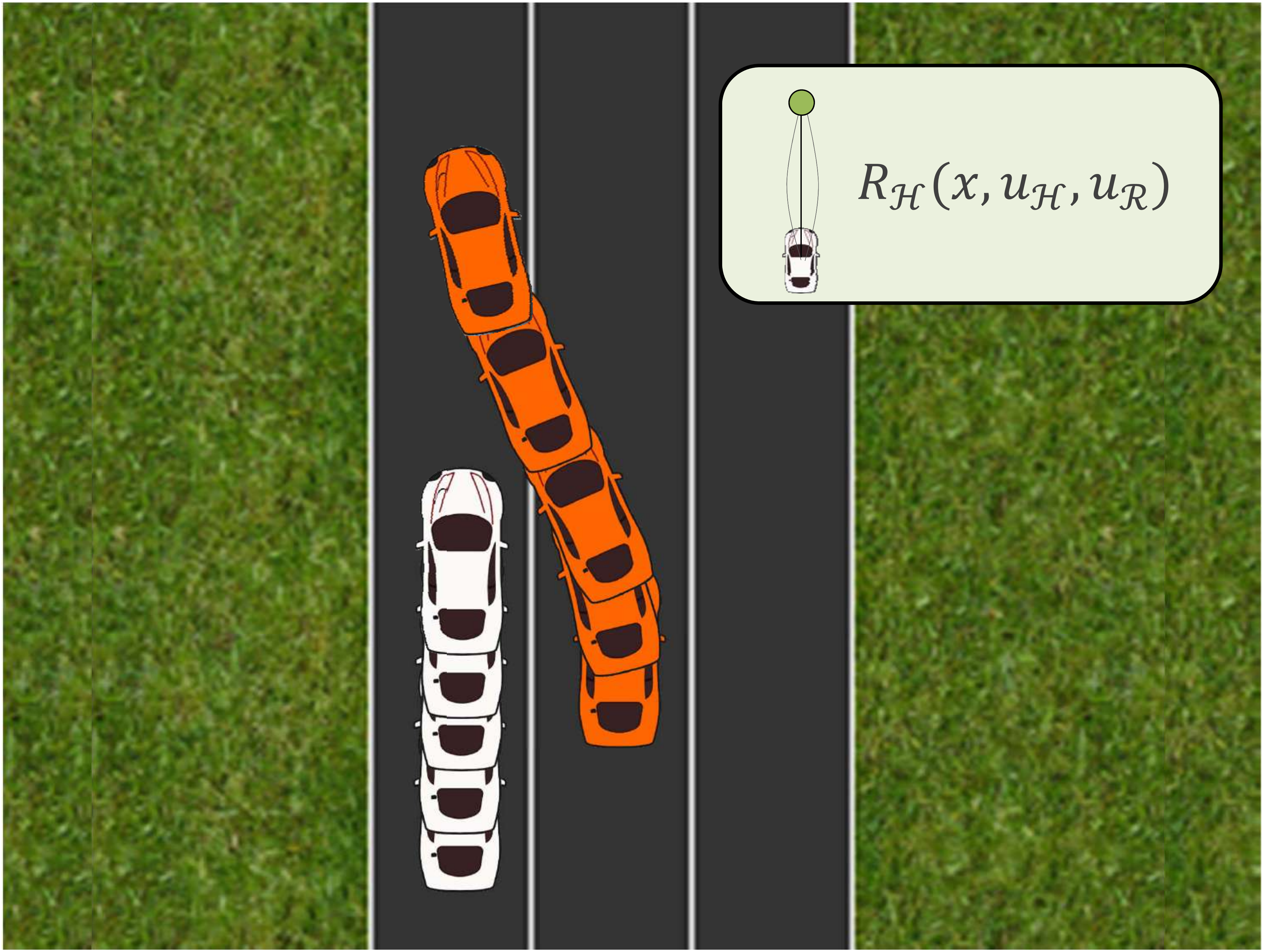
$$\boldsymbol{G}_{[0,\tau]}(\text{dist(vehicle, obstacle)} > \Delta)$$

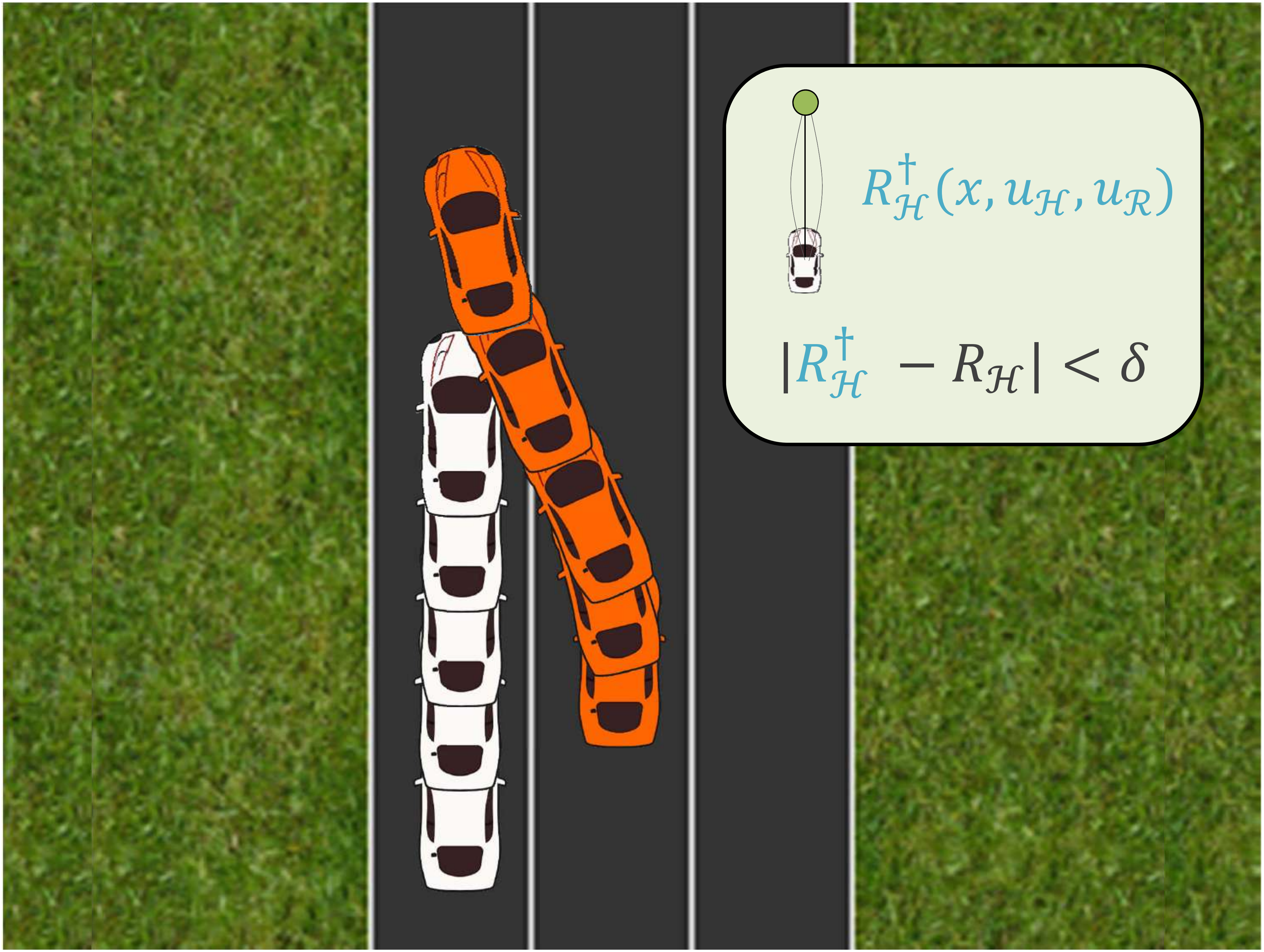
$$\Downarrow$$

$$\inf_{[0,\tau]} [\ \text{dist(vehicle, obstacle)} - \Delta]$$



$u_{\mathcal{H}}^* = \underset{u_{\mathcal{H}}}{\operatorname{argmax}} R_{\mathcal{H}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}^*)$
$u_{\mathcal{R}}^* = \underset{u_{\mathcal{R}}}{\operatorname{argmax}} R_{\mathcal{R}}(x, u_{\mathcal{R}}, u_{\mathcal{H}}^*(x, u_{\mathcal{R}}))$


$R_{\mathcal{H}}(x, u_{\mathcal{H}}, u_{\mathcal{R}})$



$R^{\dagger}_{\mathcal{H}}(x, u_{\mathcal{H}}, u_{\mathcal{R}})$
$|R^{\dagger}_{\mathcal{H}} - R_{\mathcal{H}}| < \delta$

How robust is the learning-based controller?

How to algorithmically find falsifying actions by the human?

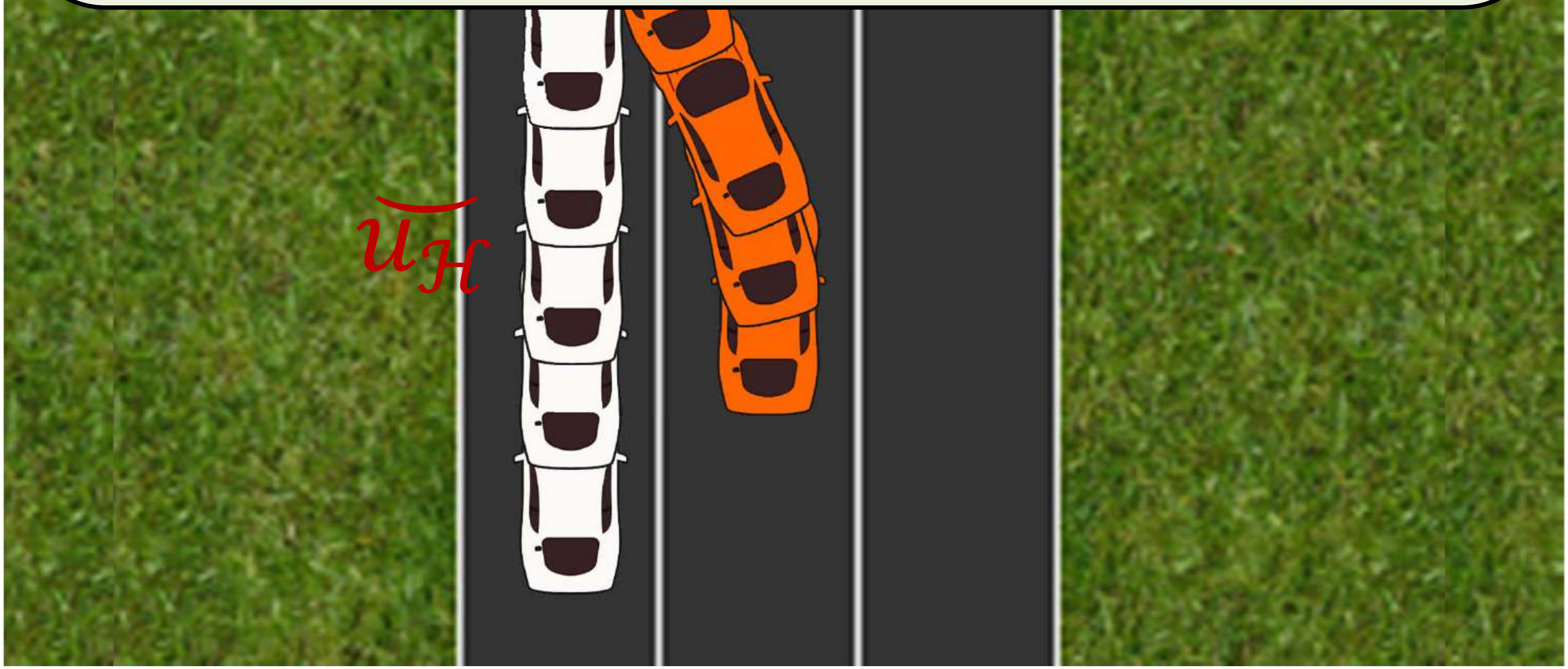
$$\widetilde{u_{\mathcal{H}}} = \arg\min_{u_{\mathcal{H}}} R_{\mathcal{R}}(x, u^*_{\mathcal{R}}, u_{\mathcal{H}})$$

Falsifying actions

$$\text{s.t. } \exists\, R^\dagger_{\mathcal{H}} : \; u_{\mathcal{H}} = \arg\max_{\widehat{u_{\mathcal{H}}}} R^\dagger_{\mathcal{H}}(x, u^*_{\mathcal{R}}, \widehat{u_{\mathcal{H}}})$$

Optimizing a perturbed version of the learned reward function.

$$|R^\dagger_{\mathcal{H}} - R_{\mathcal{H}}| < \delta$$



Theorem:

$$\widetilde{u_{\mathcal{H}}} = \arg\min_{u_{\mathcal{H}}} R_{\mathcal{R}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}})$$

$$|R_{\mathcal{H}}^{\dagger} - R_{\mathcal{H}}| < \delta$$

$$\text{s.t. } \exists\, R_{\mathcal{H}}^{\dagger} : \; u_{\mathcal{H}} = \arg\max_{\widehat{u_{\mathcal{H}}}} R_{\mathcal{H}}^{\dagger}(x, u_{\mathcal{R}}^*, \widehat{u_{\mathcal{H}}})$$

Reduction

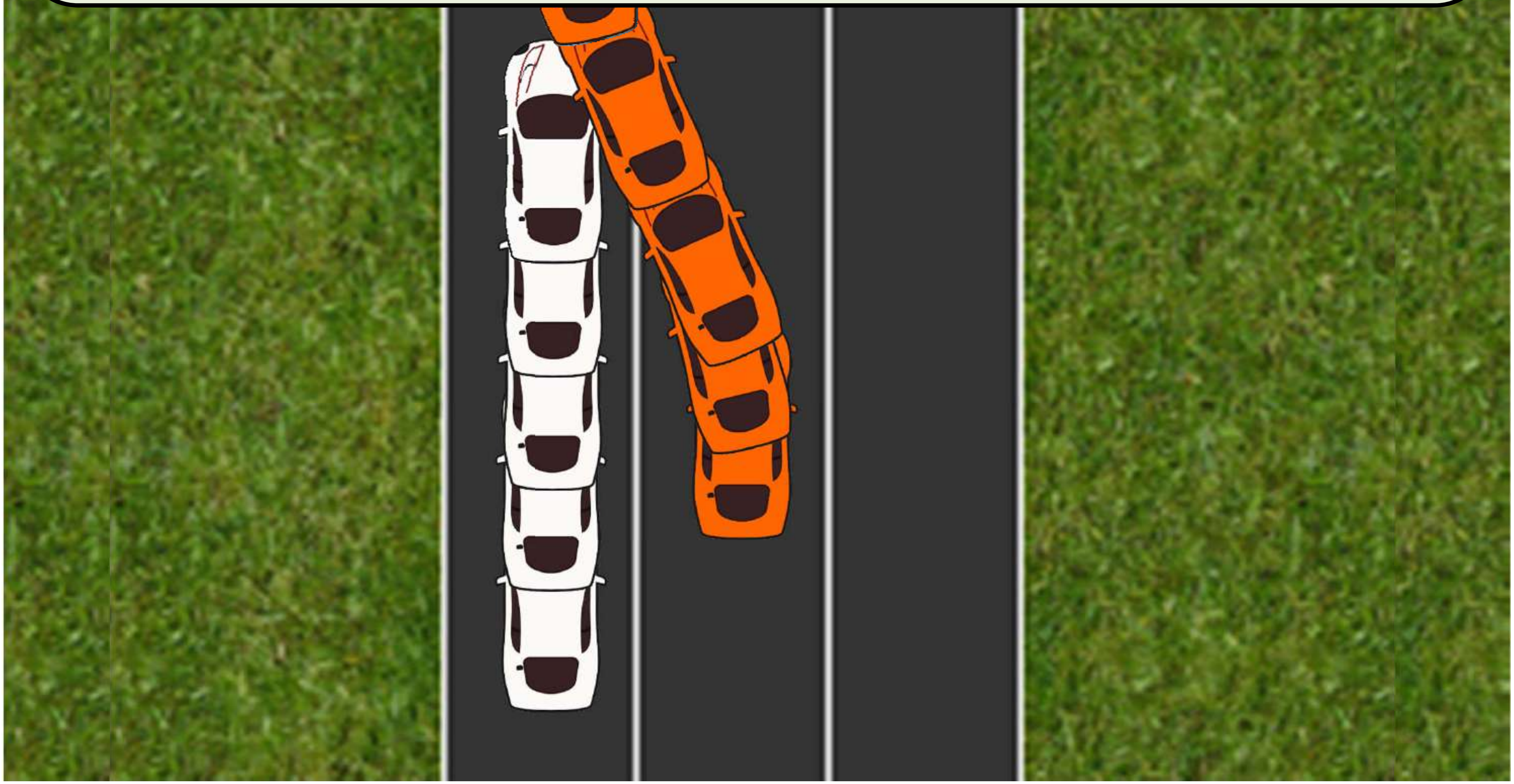
$$\widetilde{u_{\mathcal{H}}} = \arg\min_{u_{\mathcal{H}}} R_{\mathcal{R}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}})$$

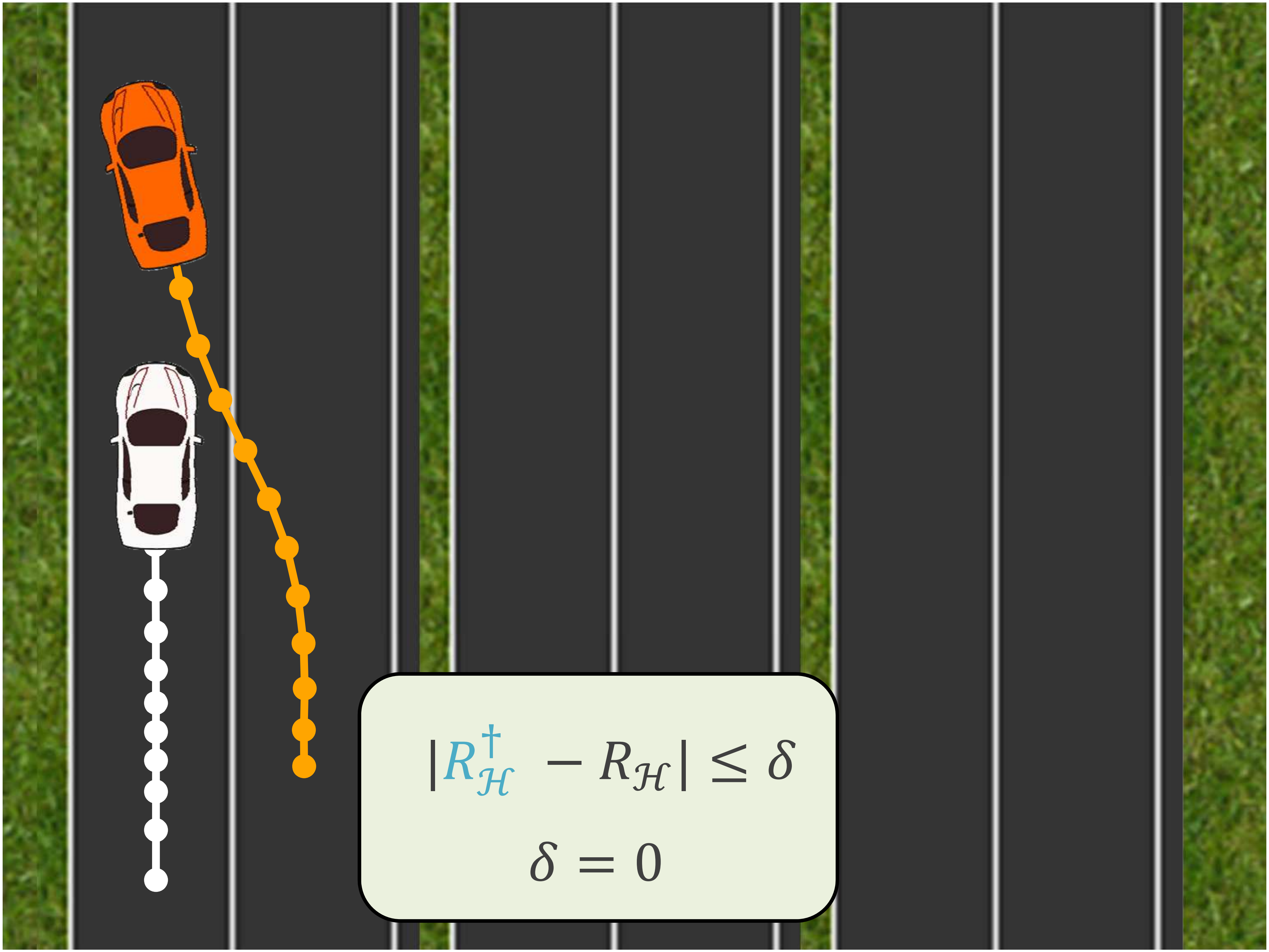
$$\text{s.t. } R_{\mathcal{H}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}}) > R_{\mathcal{H}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}}^*) - 2\delta$$

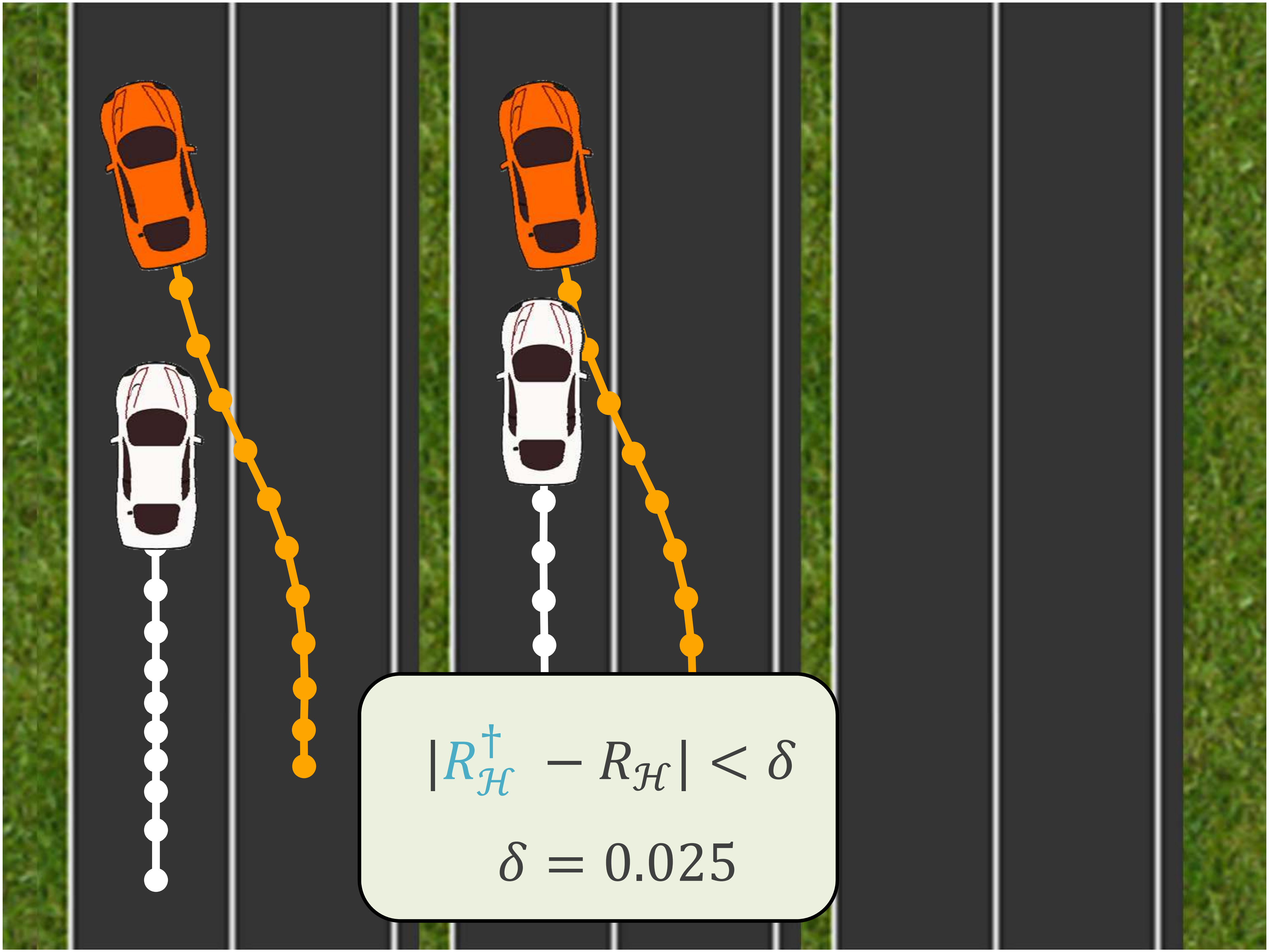
D. Sadigh, S. Sastry, S. Seshia. In Review 2017.

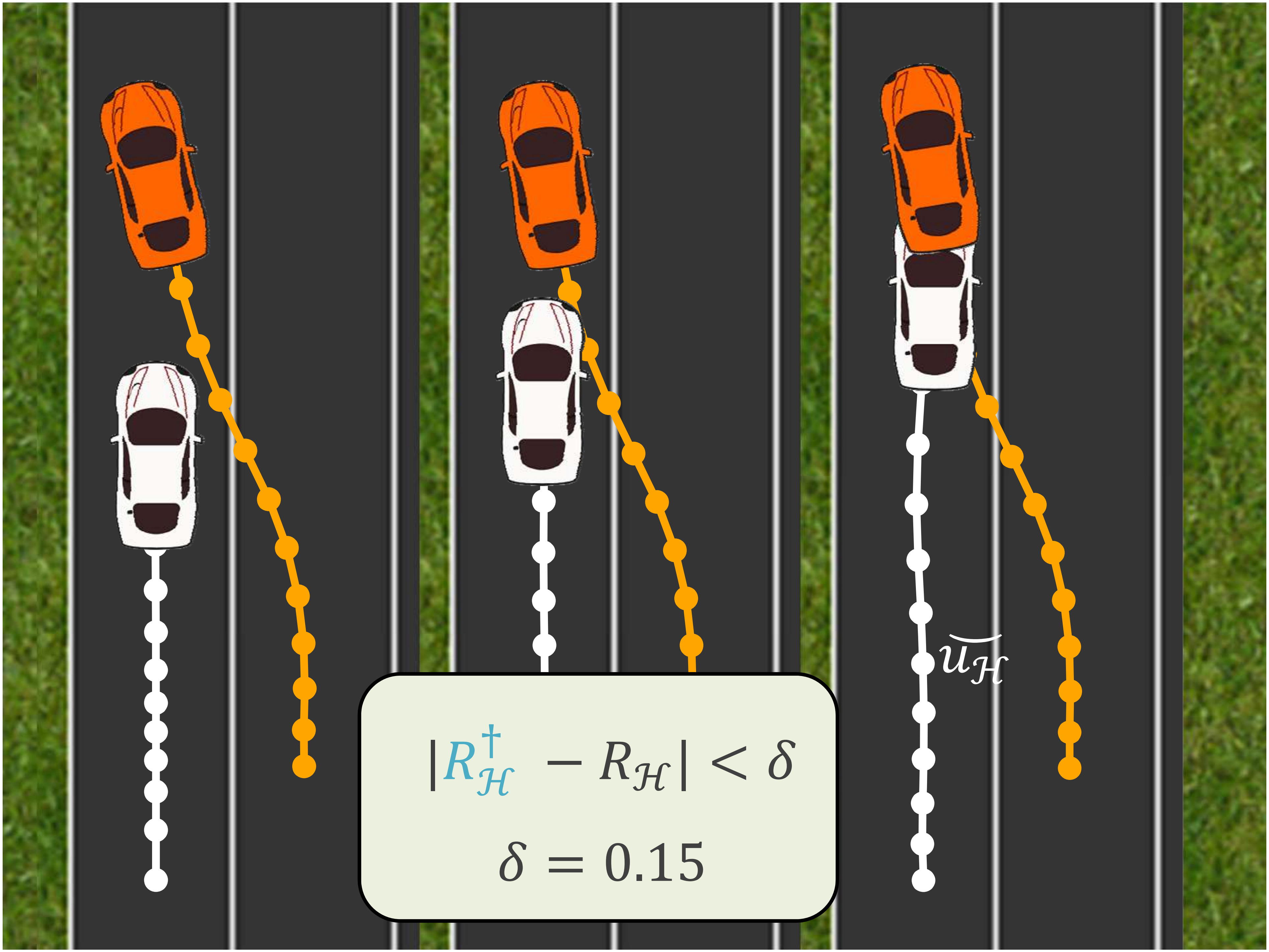
$$\widetilde{u_{\mathcal{H}}} = \arg\min_{u_{\mathcal{H}}} R_{\mathcal{R}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}})$$

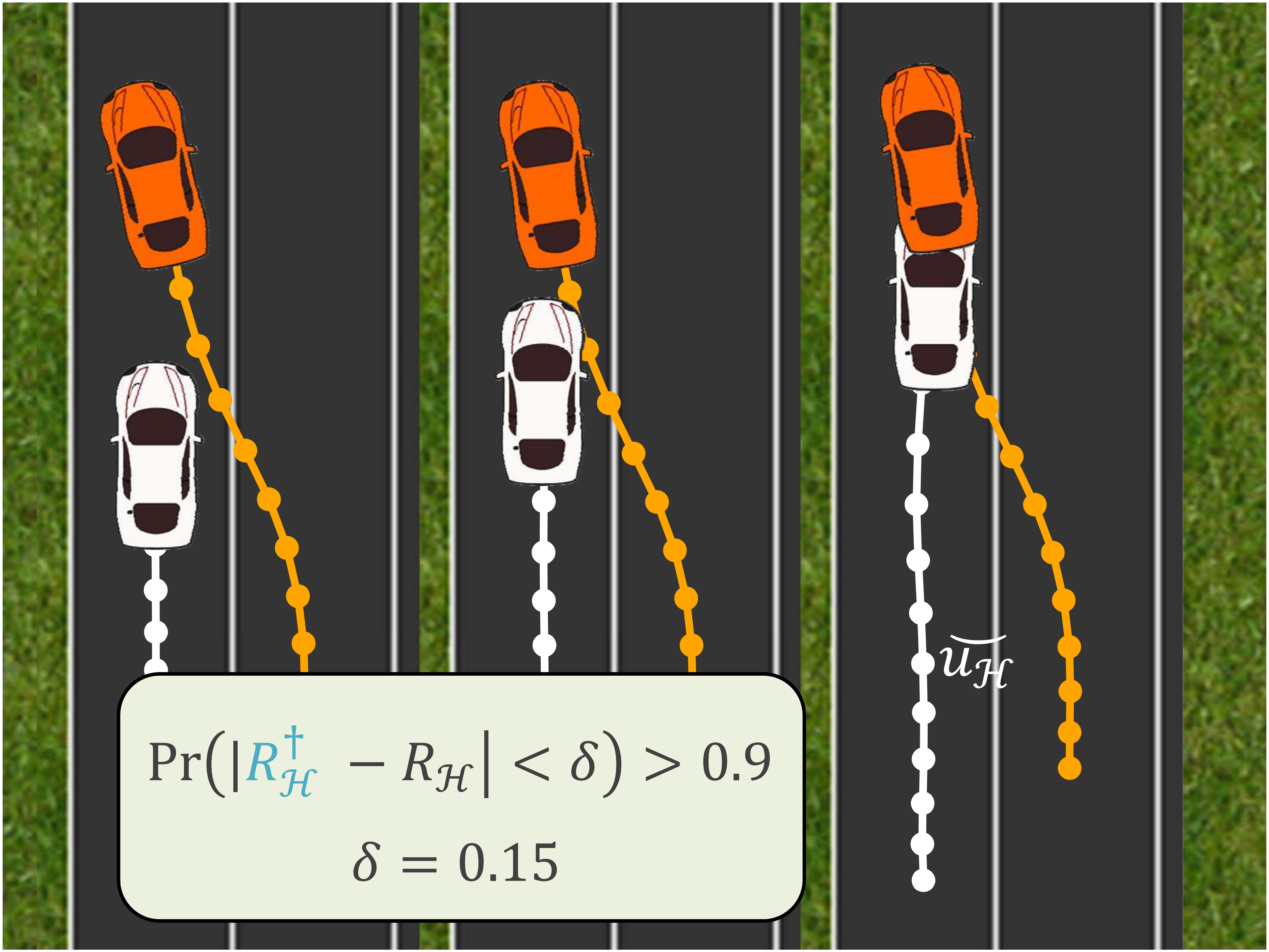
$$\text{s.t. } R_{\mathcal{H}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}}) > R_{\mathcal{H}}(x, u_{\mathcal{R}}^*, u_{\mathcal{H}}^*) - 2\delta$$




$|R^{\dagger}_{\mathcal{H}} - R_{\mathcal{H}}| \leq \delta$
$\delta = 0$


$|R_{\mathcal{H}}^{\dagger} - R_{\mathcal{H}}| < \delta$
$\delta = 0.025$


$\widetilde{u_{\mathcal{H}}}$
$|R_{\mathcal{H}}^{\dagger} - R_{\mathcal{H}}| < \delta$
$\delta = 0.15$



$\widetilde{u_{\mathcal{H}}}$
$\Pr\left(\left|R_{\mathcal{H}}^{\dagger} - R_{\mathcal{H}}\right| < \delta\right) > 0.9$
$\delta = 0.15$

Key Ideas:

Turn Verification (falsification) into **Optimization**

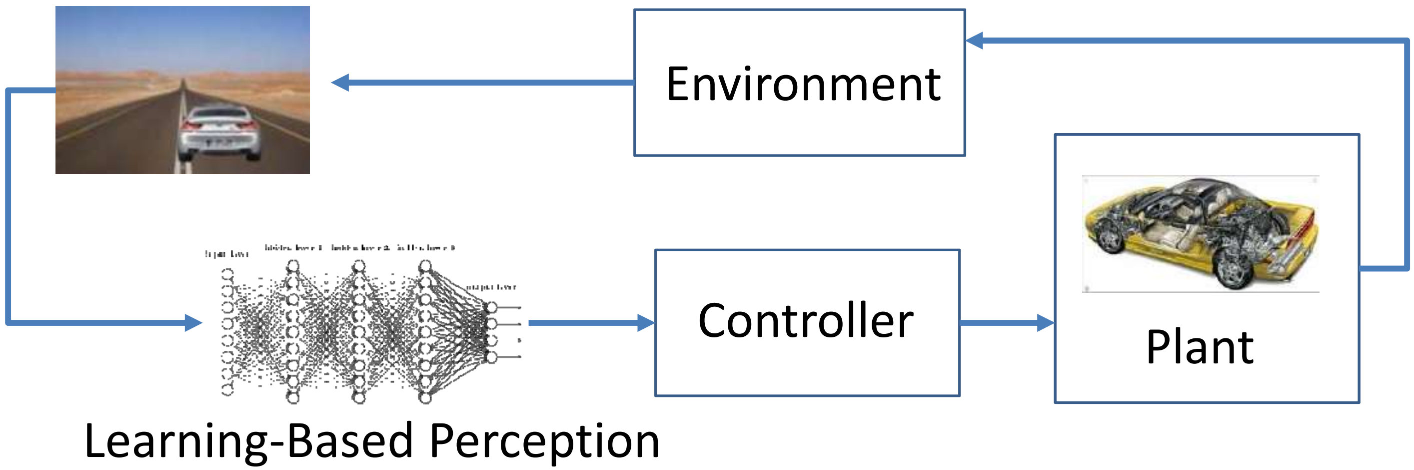
Important Property: **Robustness** of AI/Learning-based system to small perturbations in data/learned function

Challenge: Specification, Verification, Training/Testing for Learning Systems

Falsification of Cyber-Physical Systems with Machine Learning Components

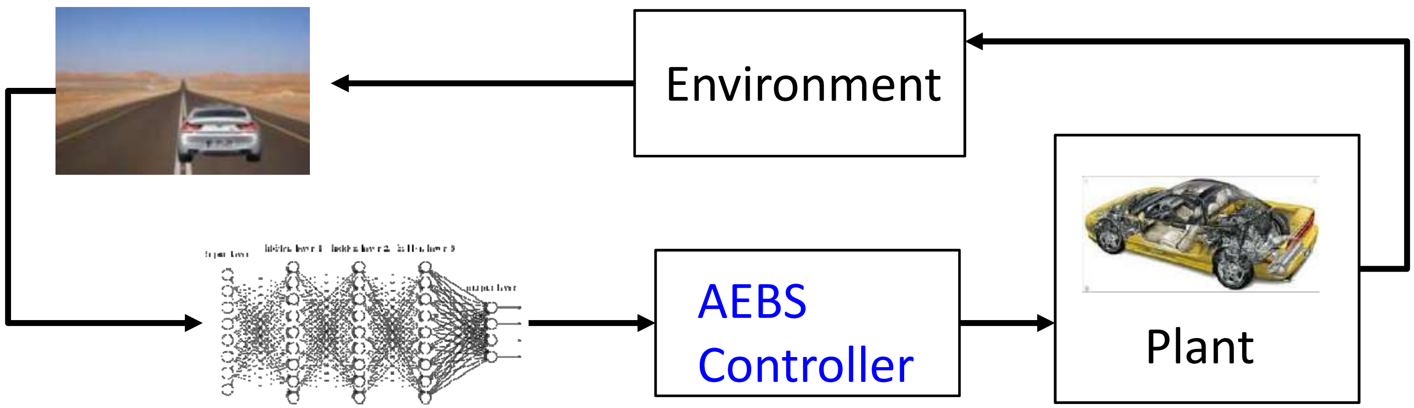
T. Dreossi, A. Donze, and S. A. Seshia. *Compositional Falsification of Cyber-Physical Systems with Machine Learning Components*, In NASA Formal Methods Symposium, May 2017.

Problem: Verify Automotive System (CPS) that uses ML-based Perception



- Focus:
  - Falsification: finding scenarios that violate safety properties
  - Test (Data) Generation: generate “interesting” data for training / testing → improve accuracy
  - Deep Neural Networks, given the increasing interest and use in the automotive context.

Automatic Emergency Braking System (AEBS)

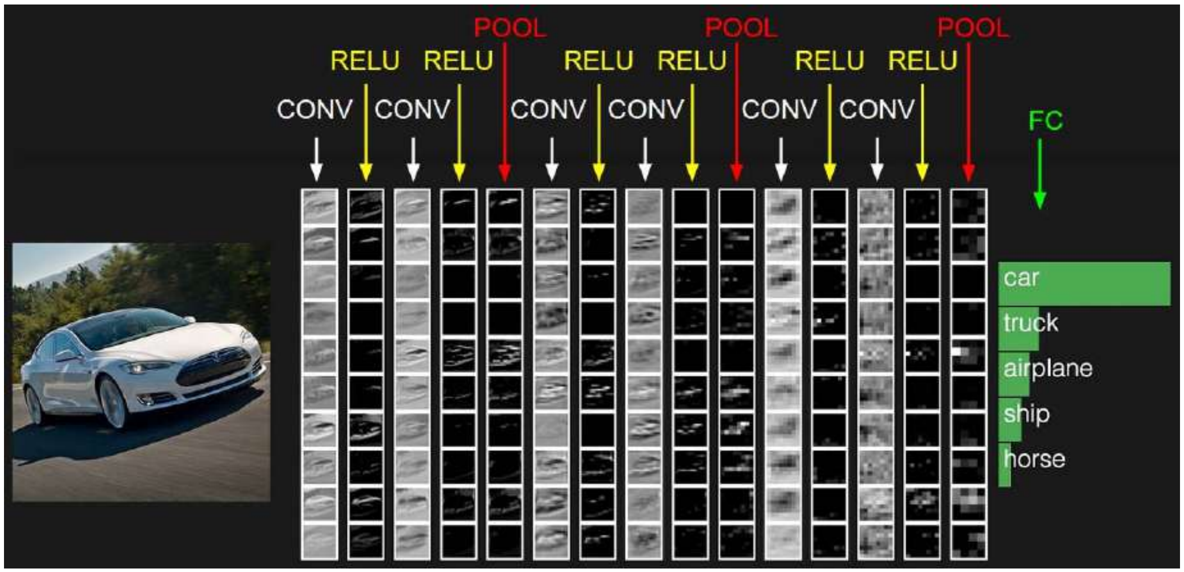


Deep Learning-Based Object Detection

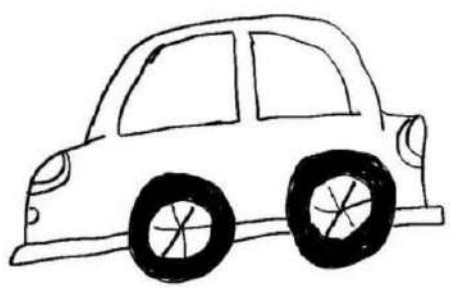
- Goal: Brake when an obstacle is near, to maintain a minimum safety distance
  - Controller, Plant, Env models in Matlab/Simulink
- Object detection/classification system based on deep neural networks
  - Inception-v3, AlexNet, … trained on ImageNet

What's the Specification for Perception Tasks?

Convolutional Neural Network trained to recognize cars



How do you formally specify "a car"?

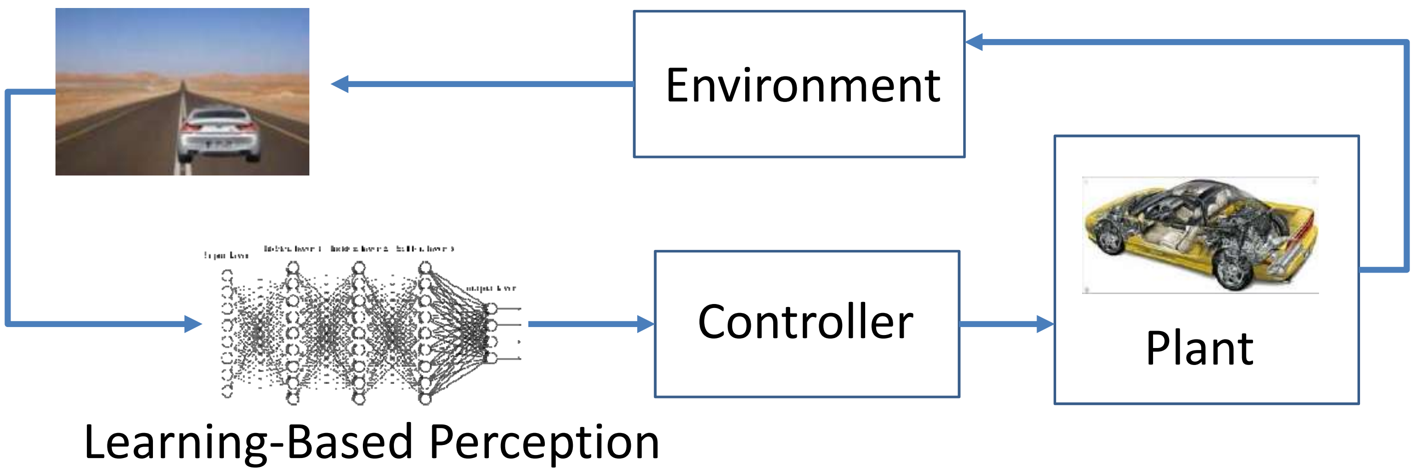


Idea: Use a **System-Level** Specification

- ✗ "Verify the Deep Neural Network"
- ✓ "Verify the System containing the Deep Neural Network"

Formally Specify the *End-to-End Behavior* of the System

STL Formula: **G** (*dist*(ego vehicle, env object) > Δ)

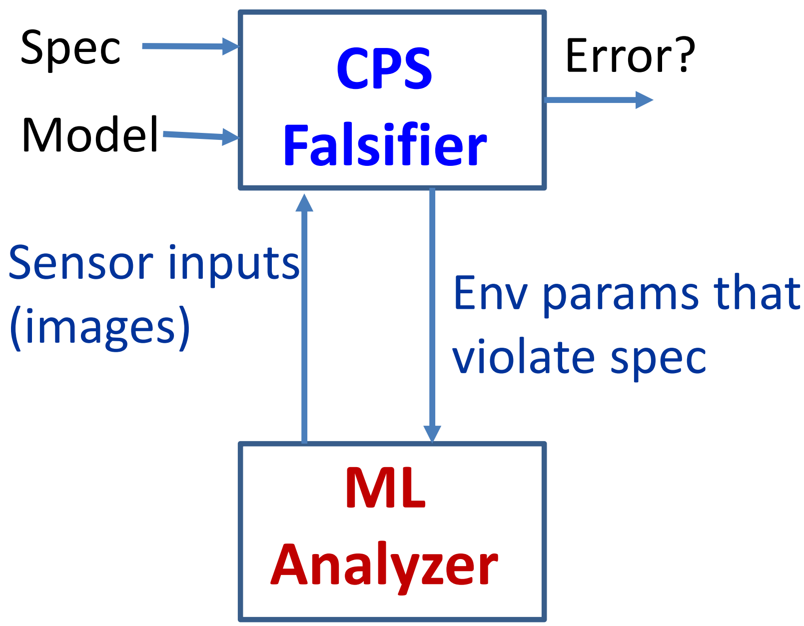


Tool: Simulation-Based Falsification of Signal Temporal Logic for CPS

- STL has quantitative semantics
  - Logical formula → Cost Function ρ
  - Quantifies "how much" a trace satisfies a property

- *Advantage:* Finding a bug (property violation) corresponds to minimizing the function ρ and checking if the value falls below 0.
  - This view of "verification as optimization" underlies the Breach toolkit and similar tools

Our Approach: Combine Temporal Logic CPS Falsifier with ML Analyzer

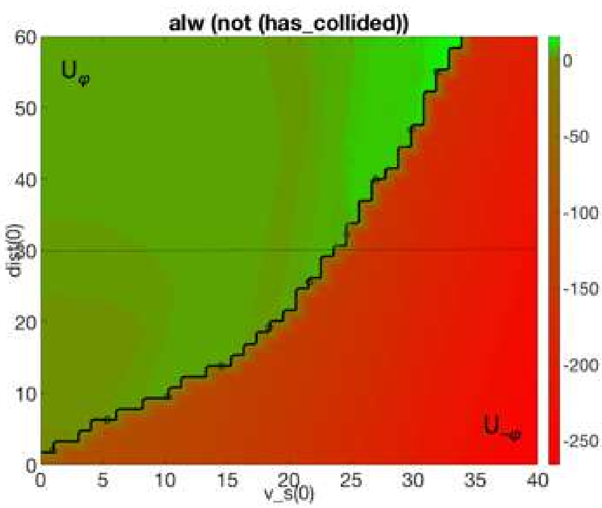


- CPS Falsifier uses abstraction of ML component
  - Optimistic analysis: assume ML classifier is always correct
  - Pessimistic analysis: assume classifier is always wrong
- Difference is the region of interest where output of the ML component "matters"

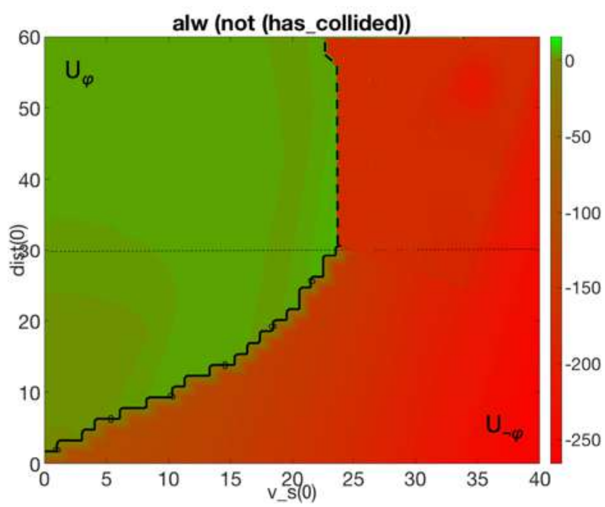
Compositional:

CPS Falsifier and ML Analyzer can be designed and run independently (& communicate)!

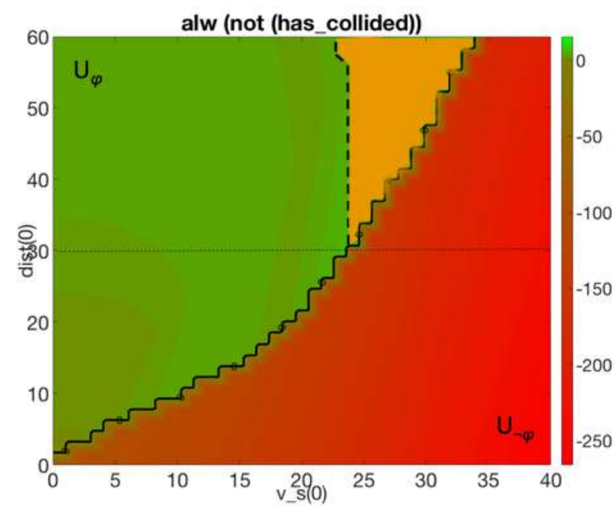
Identifying Region of Interest for Automatic Emergency Braking System



ML always correct



ML always wrong

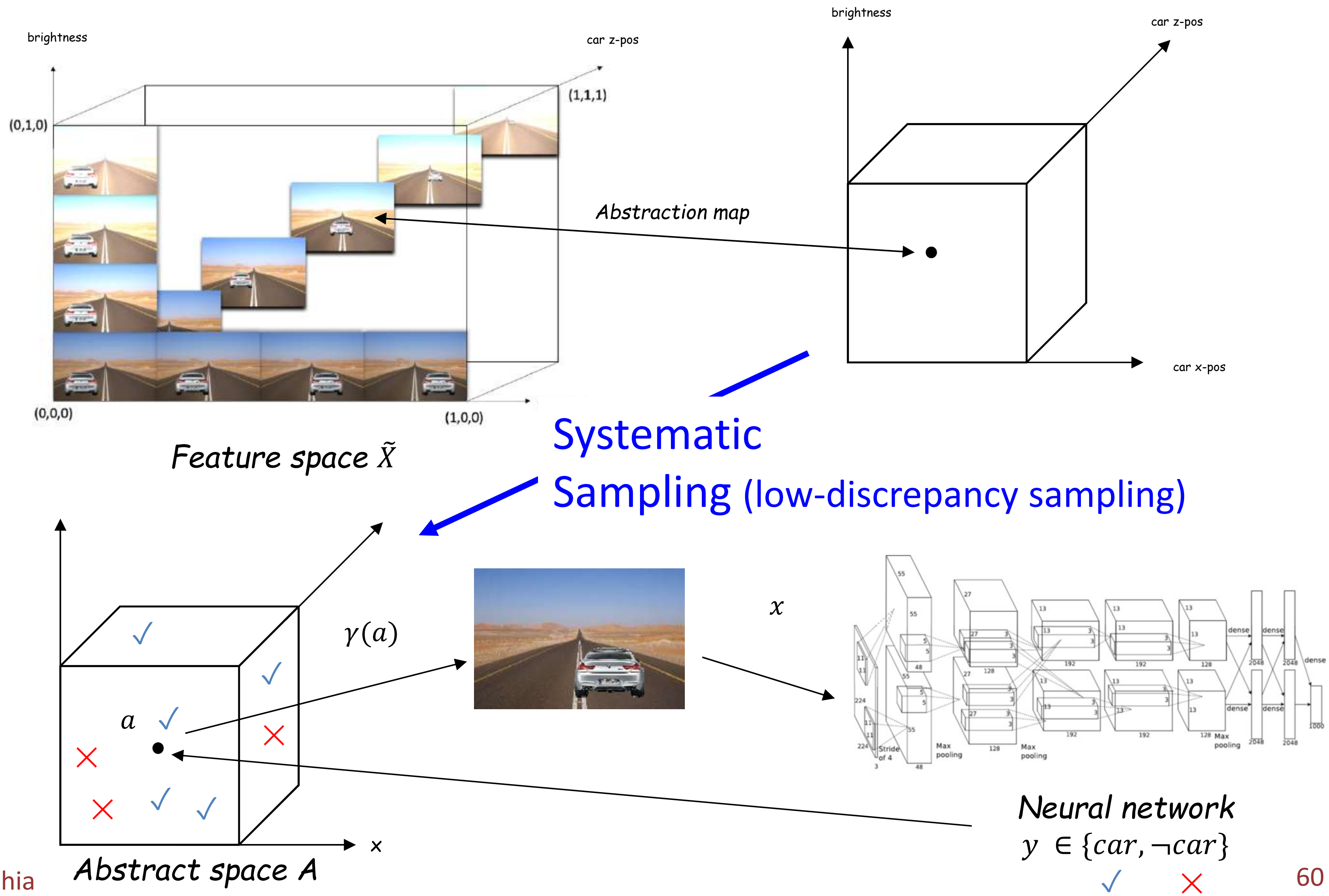


Potentially unsafe region depending on ML component (yellow)

Perform Optimistic and Pessimistic Analyses on the Deep Neural Network

Machine Learning Analyzer

Systematically Explore Region of Interest in the Image (Sensor) Space

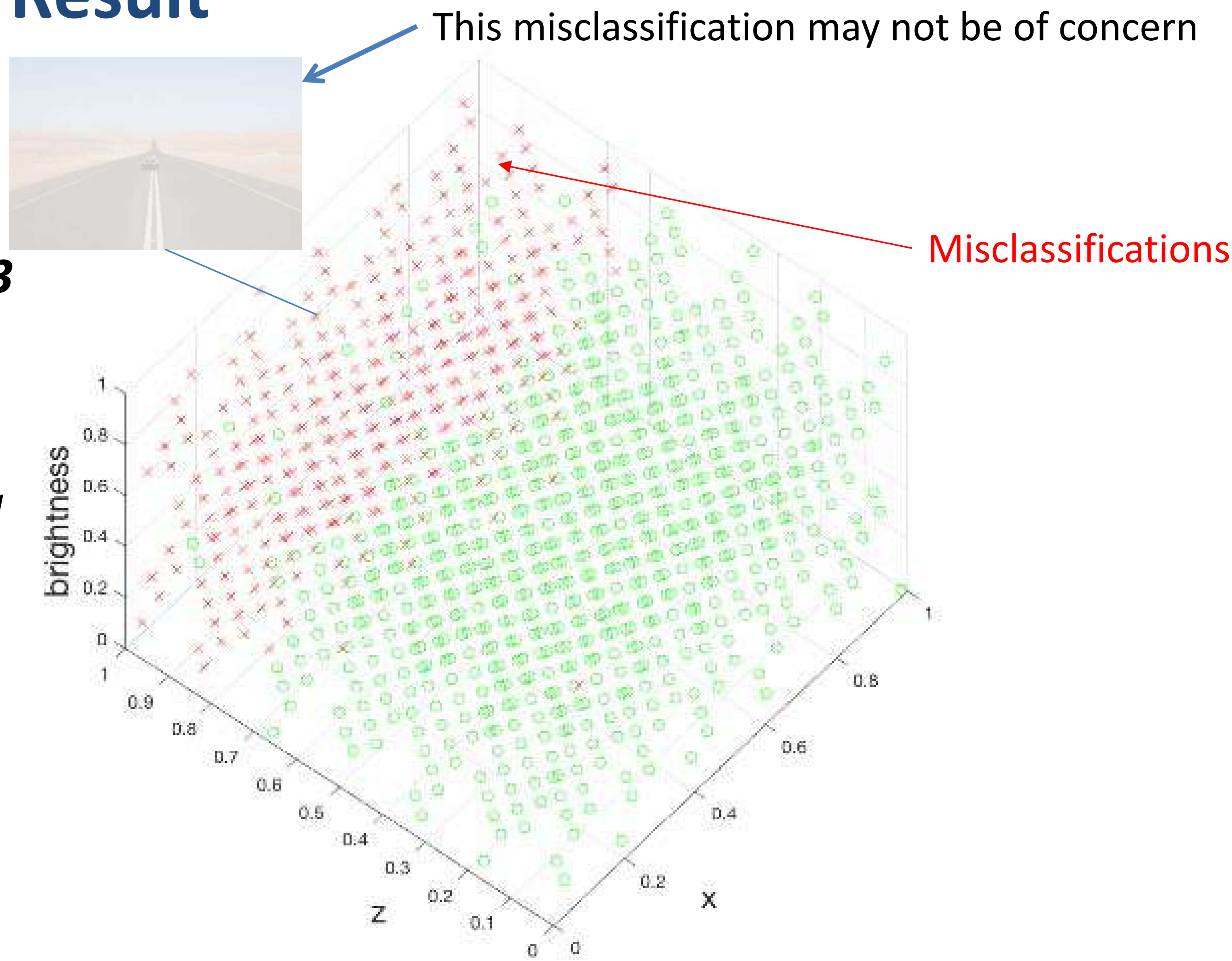


Sample Result

Inception-v3

Neural Network

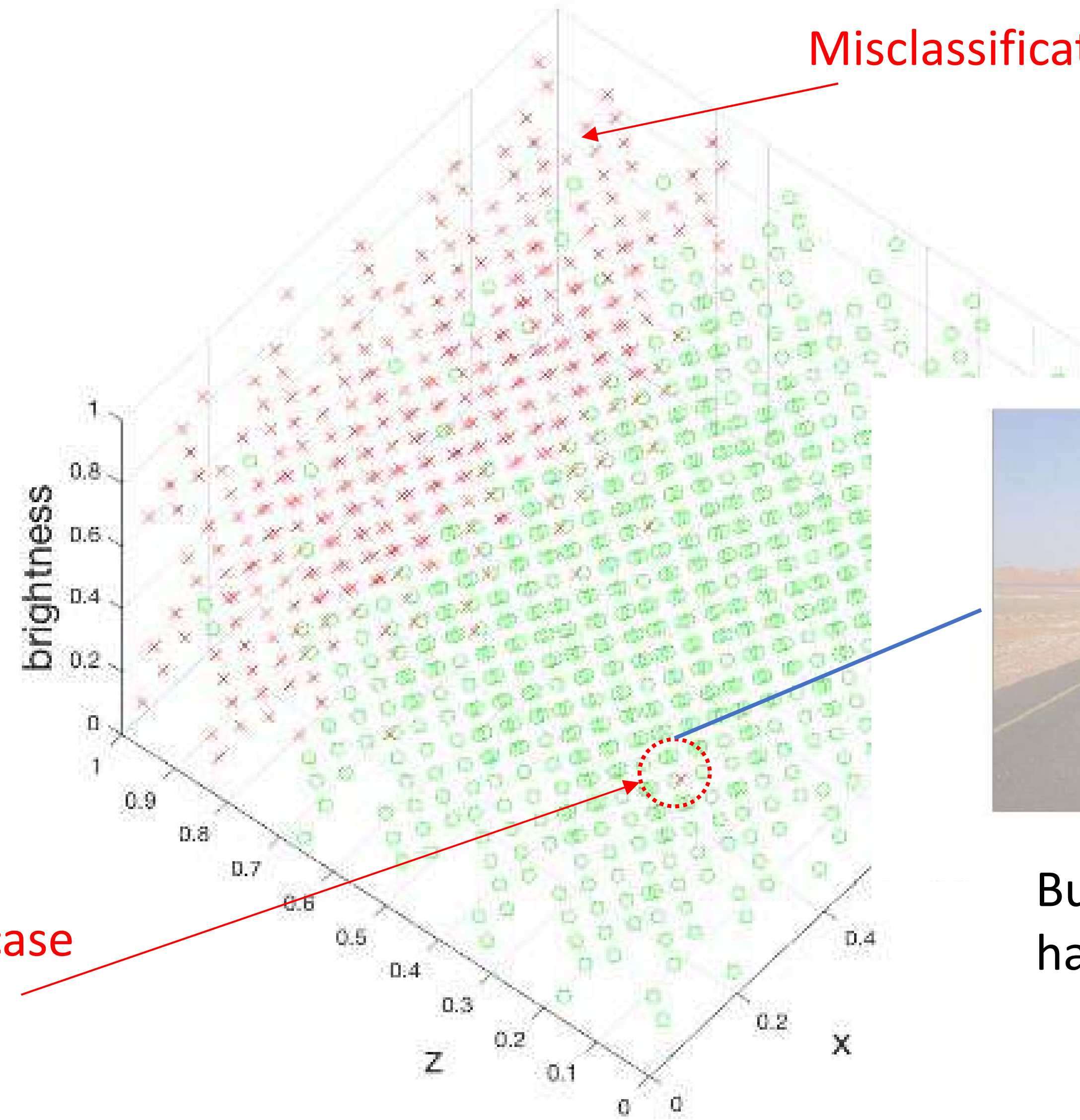
(pre-trained on ImageNet using TensorFlow)



Sample Result

Inception-v3
Neural Network
(pre-trained on ImageNet using TensorFlow)

Misclassifications

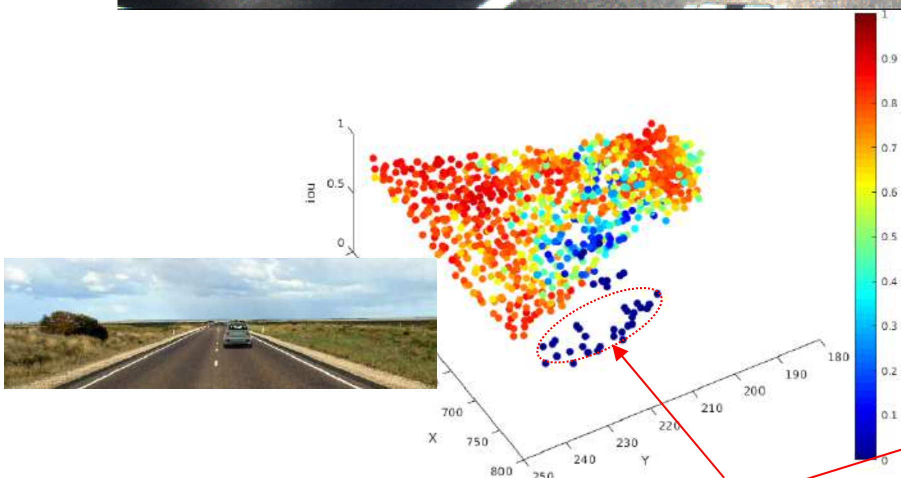


Corner case Image

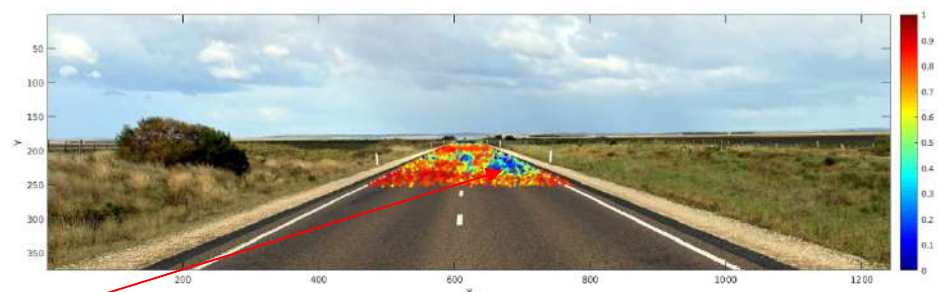
But this one is a real hazard!

Newer Results

[Dreossi, Ghosh, et al., ICML 2017 workshop]



Blind spots



Superimposition of tests on background

Results on squeezeDet NN and KITTI dataset for autonomous driving

Summary of Key ideas

- Generate adversarial examples that violate *system-level specification*
- *Compositional* Approach blends the strengths of the CPS Falsifier with a Machine Learning Analyzer
- Counterexample images can be added to the training set to improve ML accuracy ("right" data vs. "big" data)
- Ongoing/Future Work:
  - Improving ML analyzer
  - New benchmarks (datasets and networks)
  - Evaluating training/test accuracy improvements

Concluding Thoughts

Towards Verified Artificial Intelligence

Challenges	Principles
1. Environment (incl. Human) Modeling	→ Data-Driven, Introspective Environment Modeling
2. Specification	→ System-Level Specification; Robustness/Quantitative Spec.
3. Learning Systems Complexity	→ Abstract & Explain
4. Efficient Training, Testing, Verification	→ Verification-Guided, Adversarial Analysis and Improvisation
5. Design for Correctness	→ Formal Inductive Synthesis

S. A. Seshia, D. Sadigh, S. S. Sastry. *Towards Verified Artificial Intelligence*. July 2016. https://arxiv.org/abs/1606.08514.

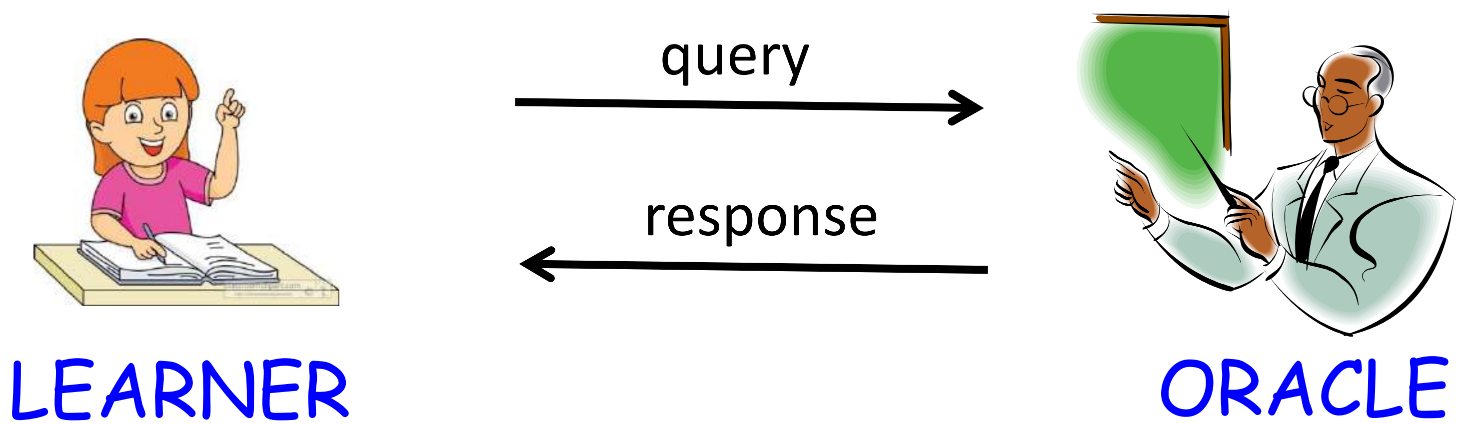
Correct-by-Construction Design with Formal Inductive Synthesis

Inductive Synthesis: Learning from Examples (ML)

Formal Inductive Synthesis: Learn from Examples *while satisfying a Formal Specification*

Key Idea: **Oracle-Guided Learning**

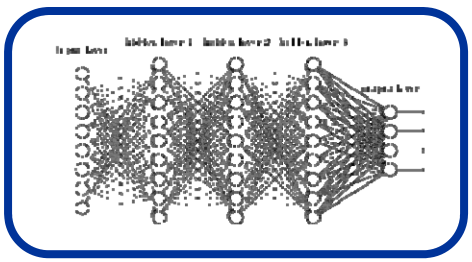
Combine Learner with Oracle (e.g., Verifier) that answers Learner's Queries



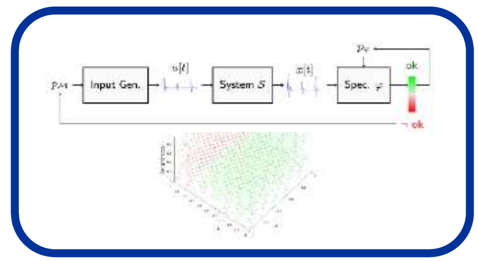
[Jha & Seshia, "A Theory of Formal Synthesis via Inductive Learning", 2015, Acta Informatica 2017.]

Verifier-Guided Training of Deep Neural Networks

- Instance of Oracle-Guided Inductive Synthesis
- Oracle is Verifier (CPSML Falsifier) used to perform counterexample-guided training of DNNs
- Substantially increase accuracy with only few additional examples



Learned Classifier



DEEP NEURAL NETWORK

FALSIFIER (CPS + ML)

Towards Verified Artificial Intelligence

Challenges		Principles
1. Environment (incl. Human) Modeling	→	Data-Driven, Introspective Environment Modeling
2. Specification	→	System-Level Specification; Robustness/Quantitative Spec.
3. Learning Systems Complexity	→	Abstract & Explain
4. Efficient Training, Testing, Verification	→	Verification-Guided, Adversarial Analysis and Improvisation
5. Design for Correctness	→	Formal Inductive Synthesis

Exciting Times Ahead!!! Thank you!

S. A. Seshia, D. Sadigh, S. S. Sastry. *Towards Verified Artificial Intelligence*. July 2016. https://arxiv.org/abs/1606.08514.