Principles for Verified Learning-Based CPS

Sanjit A. Seshia

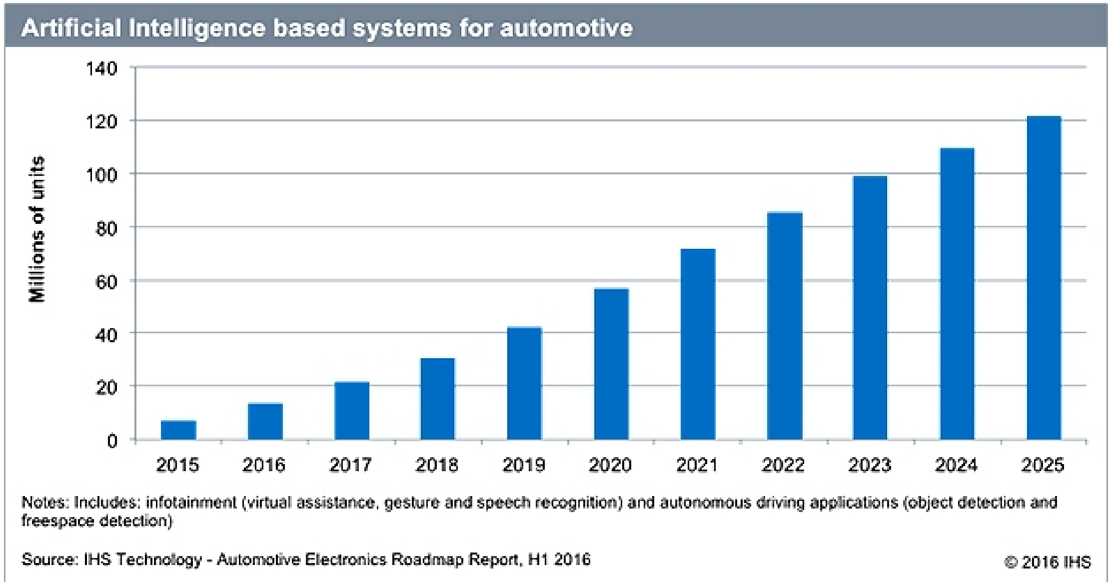
Professor

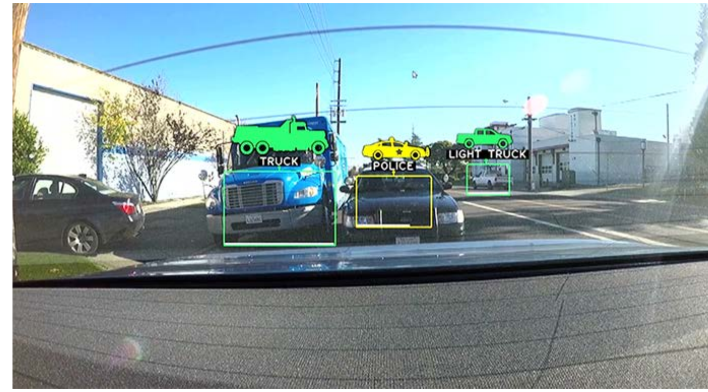
EECS, UC Berkeley

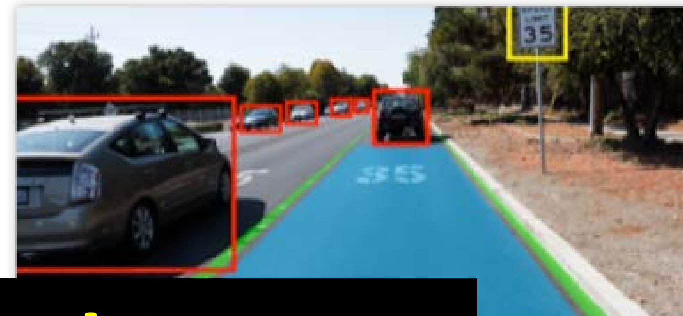
Joint work with
Dorsa Sadigh, Tommaso Dreossi, Alexander Donze,
S. Shankar Sastry

NSF CPS PI Meeting 2017
November 14, 2017

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







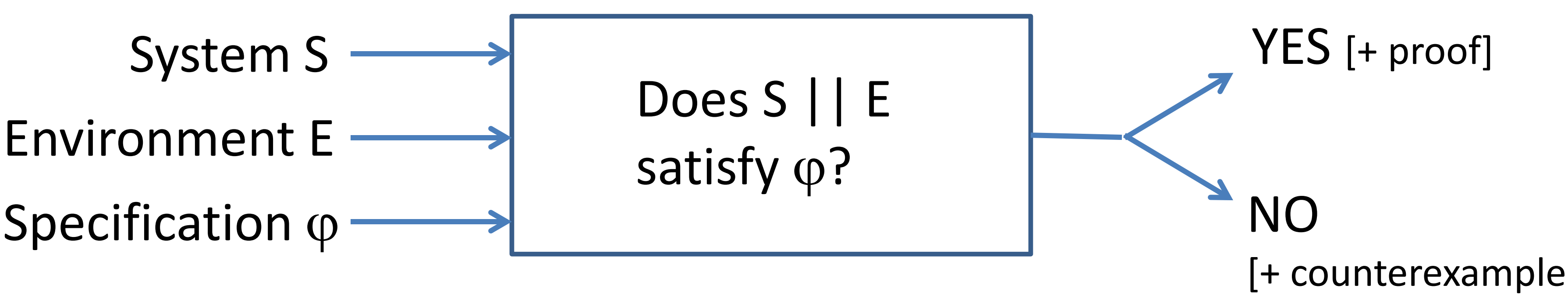
Many Safety-Critical Systems

AI / Cognitive Systems / Learning Systems

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

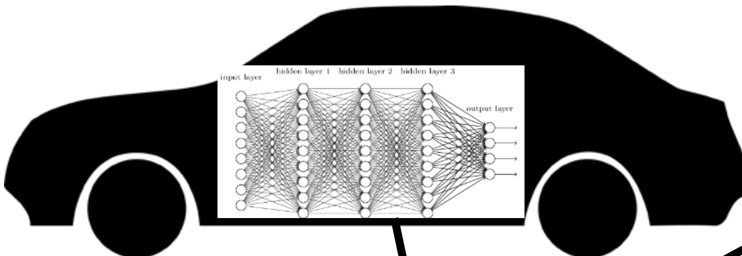
Formal Methods / Verification

Computational Proof Techniques: SAT Solving, SMT Solving, Directed simulation, Model checking, Theorem proving, …



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?

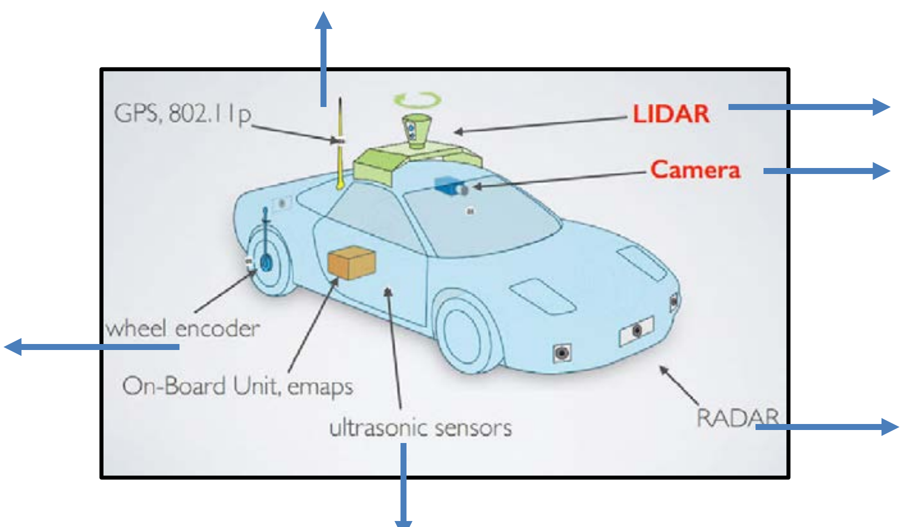
Principle 1: Environment Modeling -- Introspection and Action

#1: Introspective Environment Modeling

Impossible to model all possible scenarios

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

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

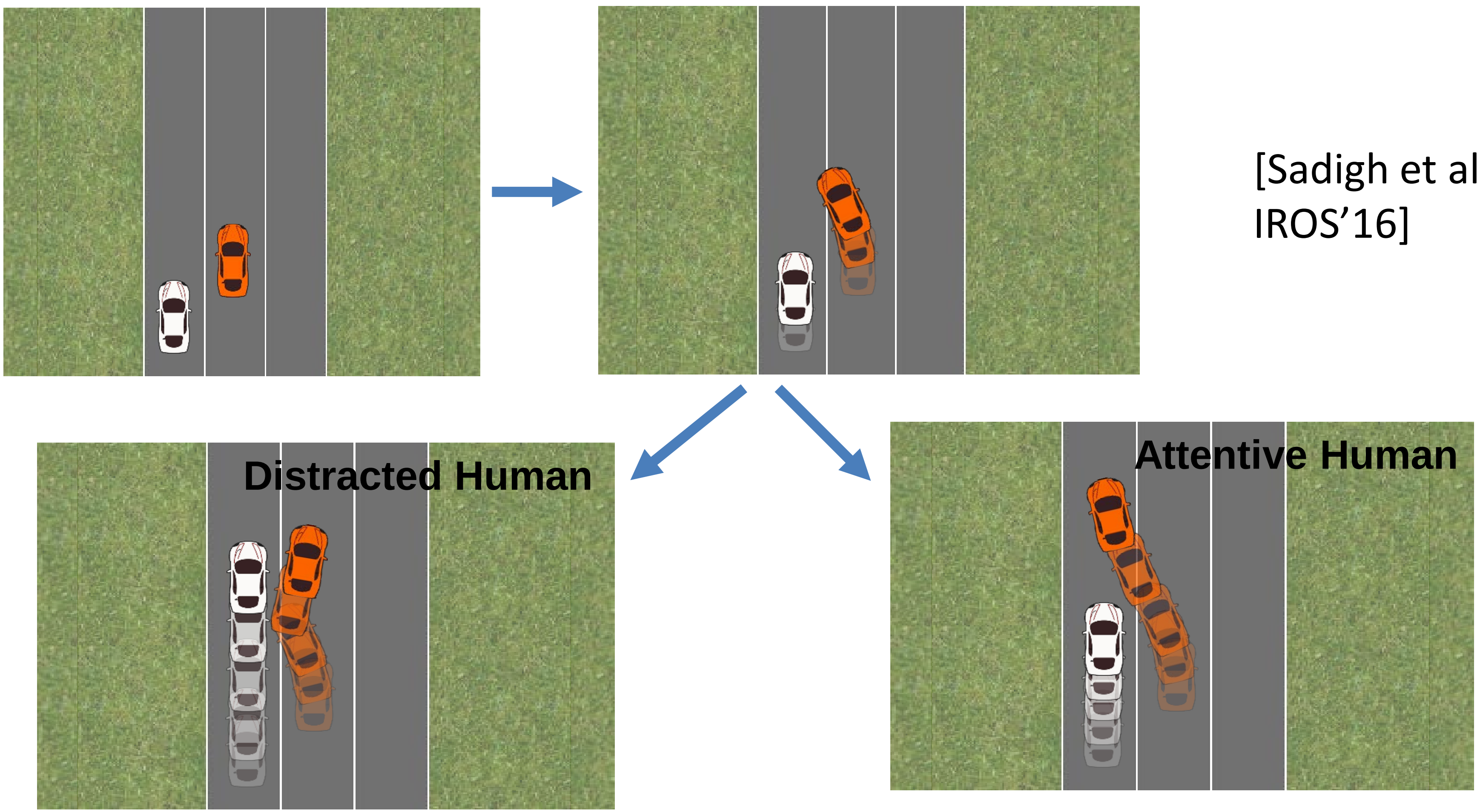


Algorithmic techniques to *generate weakest interface assumptions* and *monitor them at run-time* for potential violation/mitigation

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

#2: Active Data Gathering and Learning

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



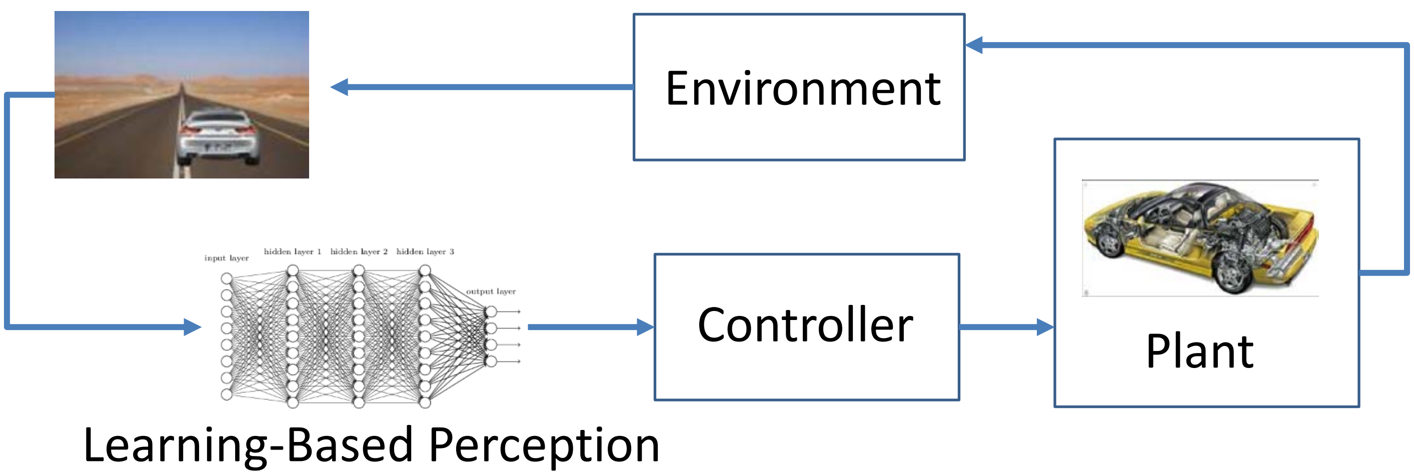
[Sadigh et al., IROS'16]

Principle 2: Formal Specification -- Go System Level

Use a **System-Level** Specification

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

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

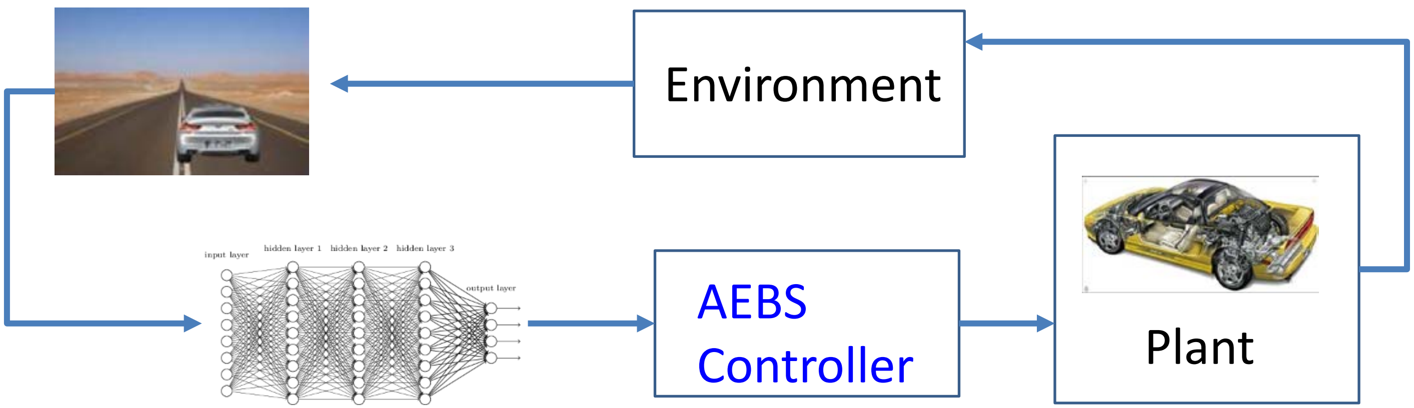


Spec: **G** (*dist*(ego vehicle, env object) > Δ)

Principle 3: Learning Systems Complexity -- Abstract and Explain

Principle 4: Efficient Training, Testing, and Verification -- Verification-Guided Analysis and Improvisation

The Problem: Verify Automatic Emergency Braking System (AEBS)

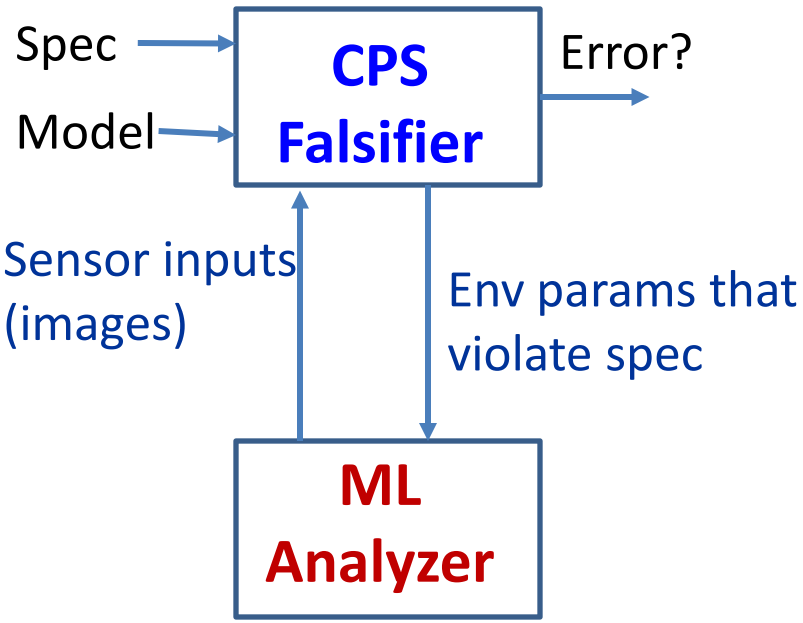


Deep Learning-Based Object Detection

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

- Controller, Plant, Env models in Matlab/Simulink
- Multiple Deep Neural Networks: Inception-v3, AlexNet, …

Our Approach: Combine Temporal Logic CPS Falsifier with ML Analyzer



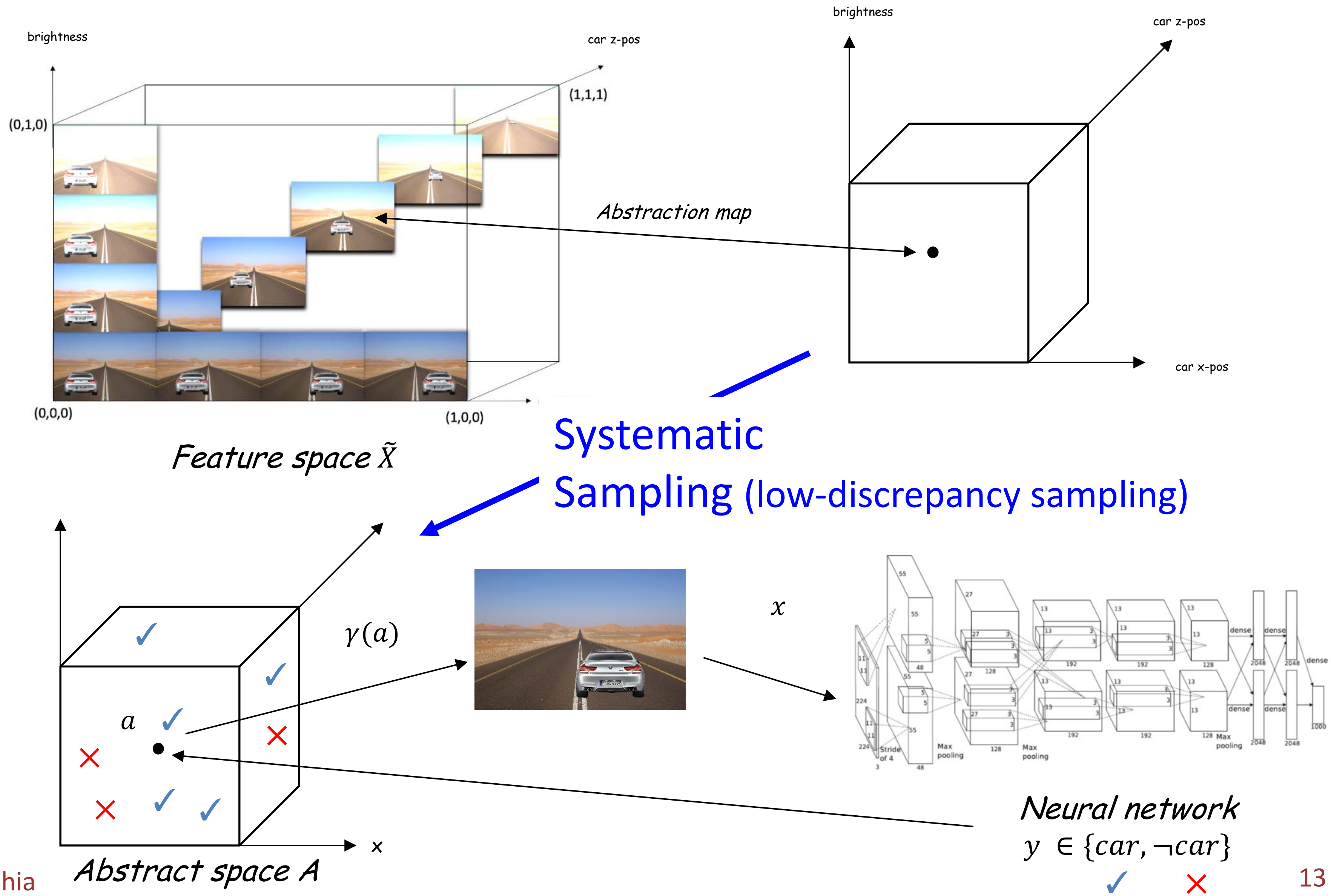
- CPS Falsifier uses abstraction of ML component
  - Optimistic analysis: assume ML classifier is always correct
  - Pessimistic analysis: assume classifier is always wrong
- Difference is the region of interest where output of the ML component “matters”

Compositional:

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

Machine Learning Analyzer

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



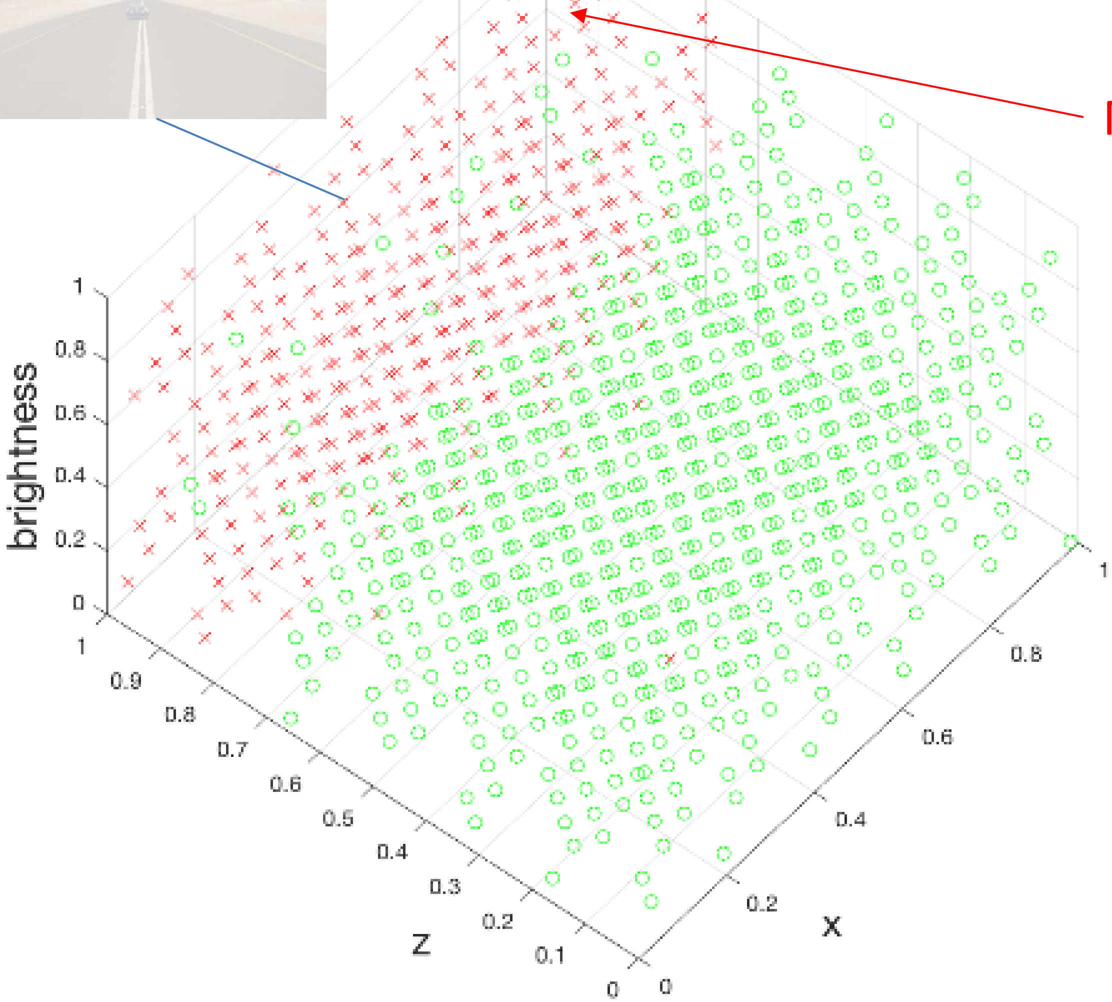
Sample Result

This misclassification may not be of concern

Inception-v3

Neural Network

(pre-trained on ImageNet using TensorFlow)

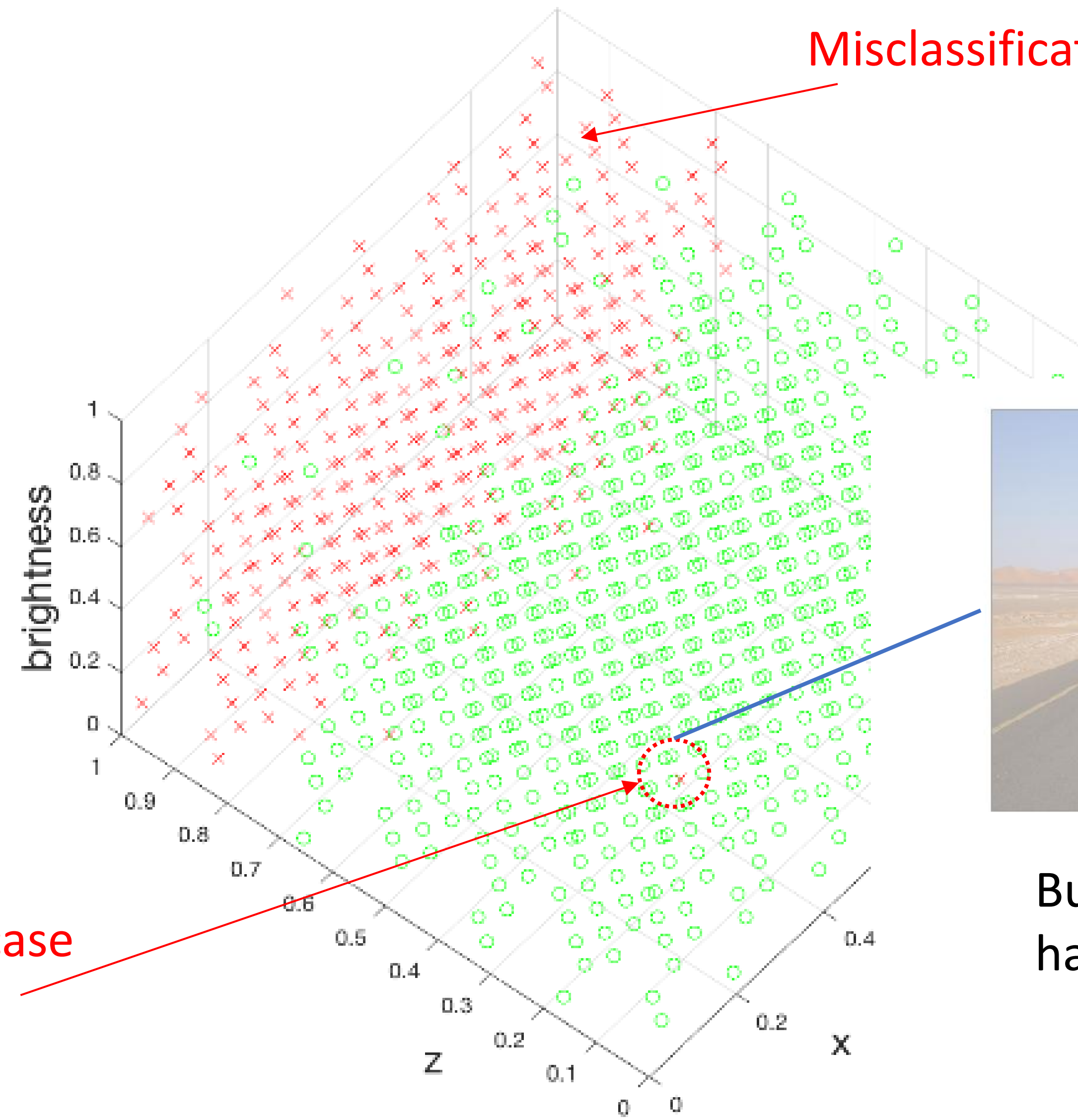


Misclassifications

Sample Result

Misclassifications

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



Corner case Image

But this one is a real hazard!

Principle 5: Correct-by-Construction -- Formal Inductive Synthesis

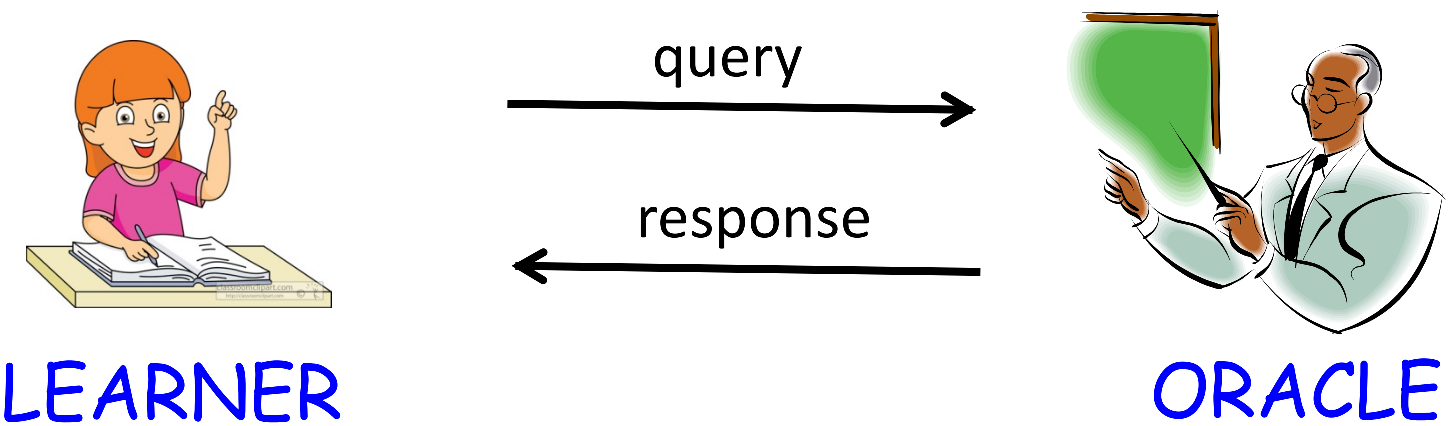
Correct-by-Construction Design with Formal Inductive Synthesis

Inductive Synthesis: Learning from Examples (ML)

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

Key Idea: **Oracle-Guided Learning**

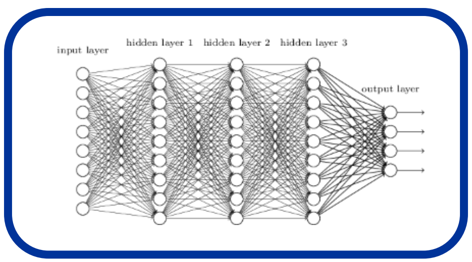
Combine Learner with Oracle (e.g., Verifier) that answers Learner's Queries



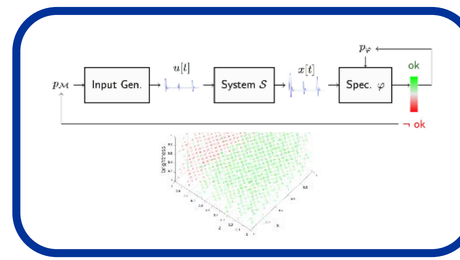
[Jha & Seshia, "A Theory of Formal Synthesis via Inductive Learning", 2015, Acta Informatica 2017.]

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 Learning-based CPS

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.