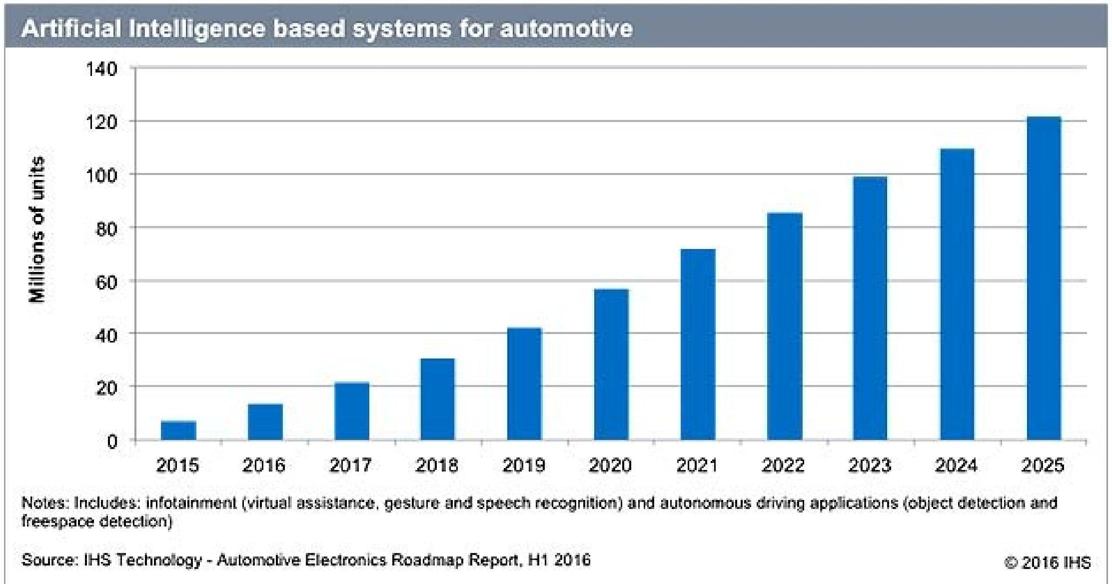
Challenges and Principles for Verified Learning-Based Systems

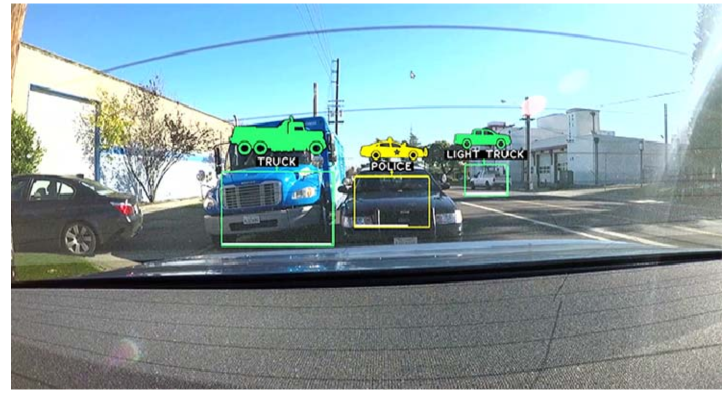
Sanjit A. Seshia
EECS, UC Berkeley

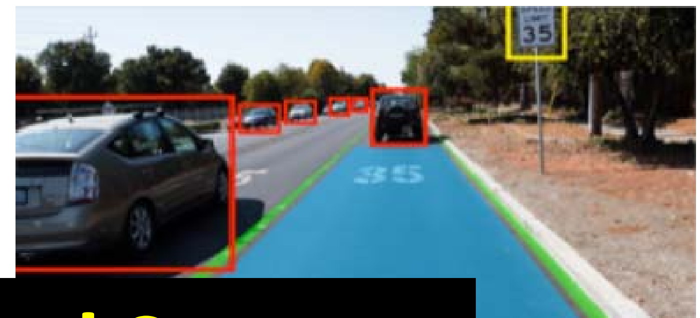
EECS 219C: Formal Methods

Based on talk given at NSF CPS PI Meeting 2017

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







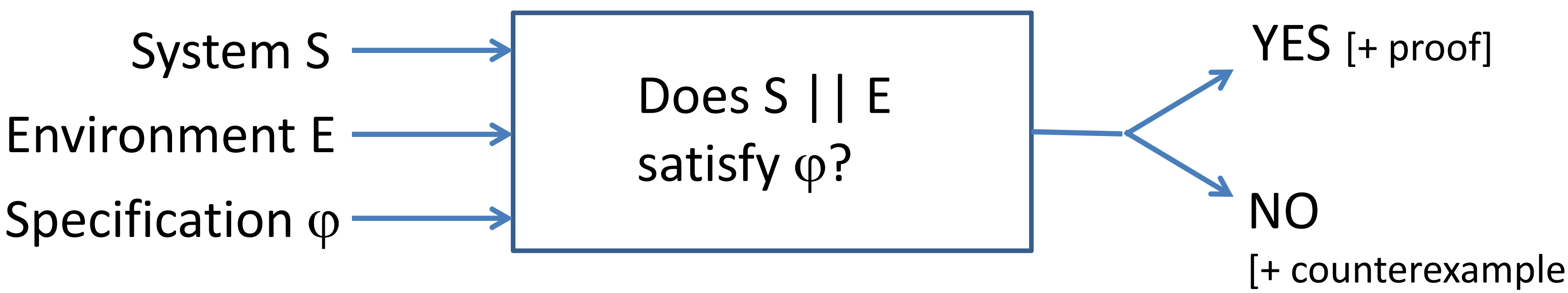
Many Safety-Critical Systems

AI / Cognitive Systems / Learning Systems

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

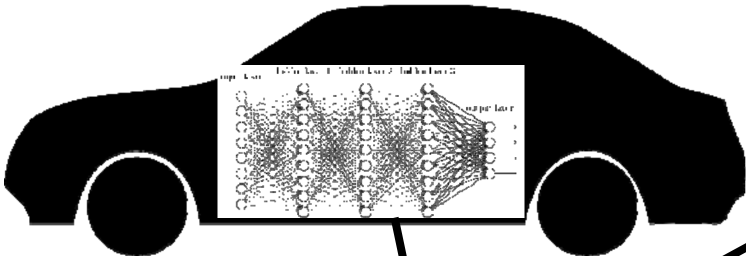
Formal Methods / Verification

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?

Challenge 1: Environment Modeling -- Principle: **Introspection and Action**

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!!

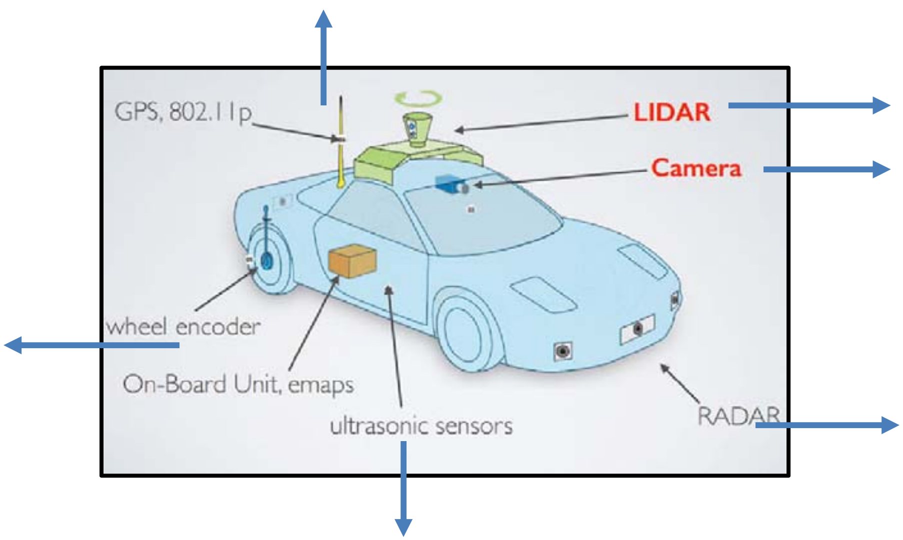
Cannot represent all possible environment scenarios

#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

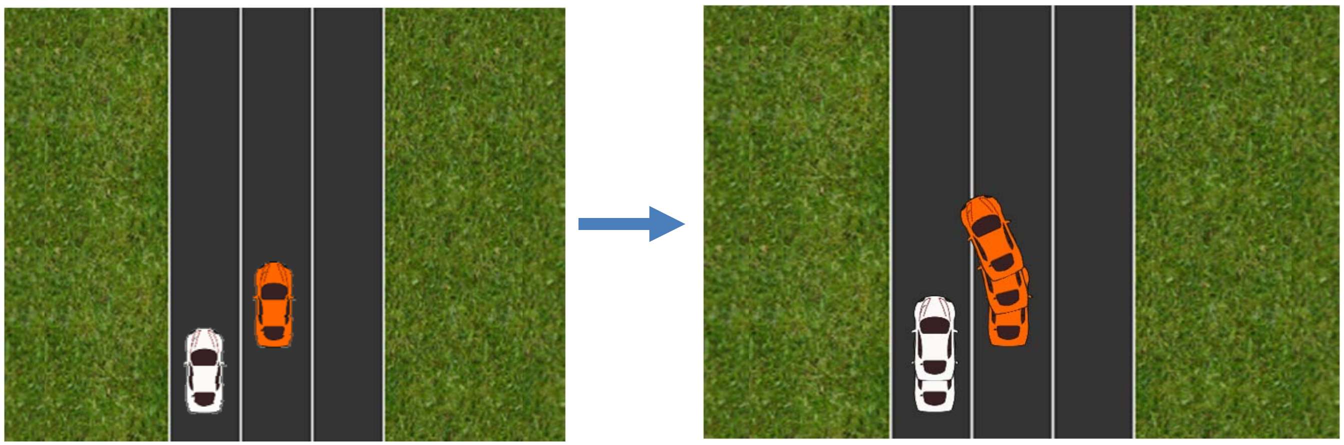


Algorithmic techniques to *generate weakest interface assumptions* and *monitor them at run-time* for potential violation/mitigation

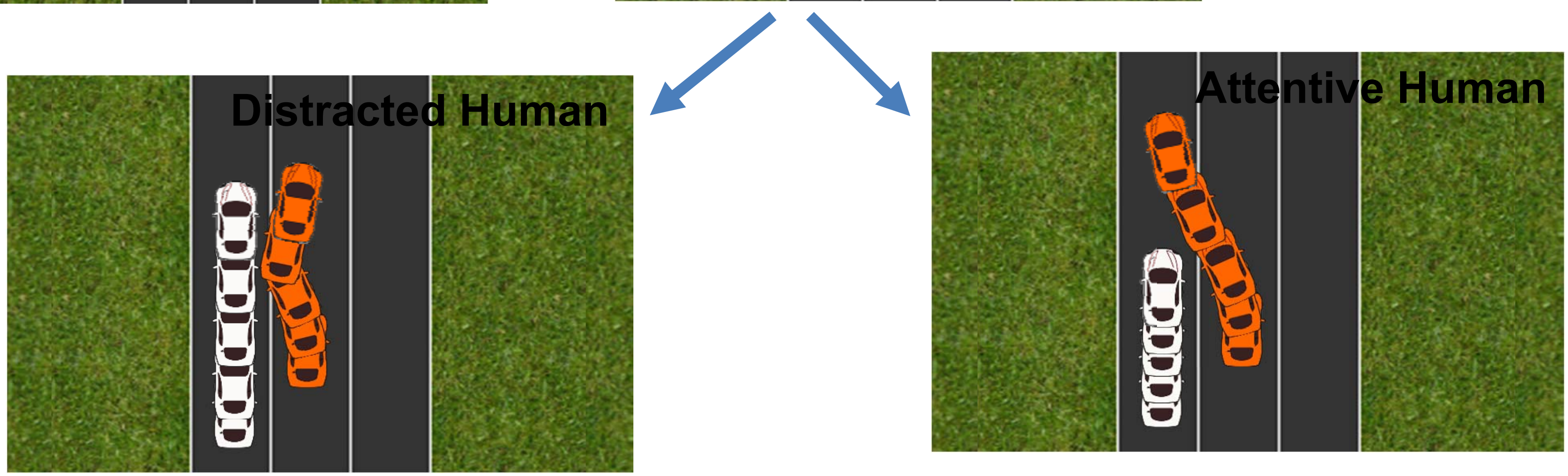
[Li, Sadigh, Sastry, Seshia; TACAS'14]

#2: Active Data Gathering and Learning

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



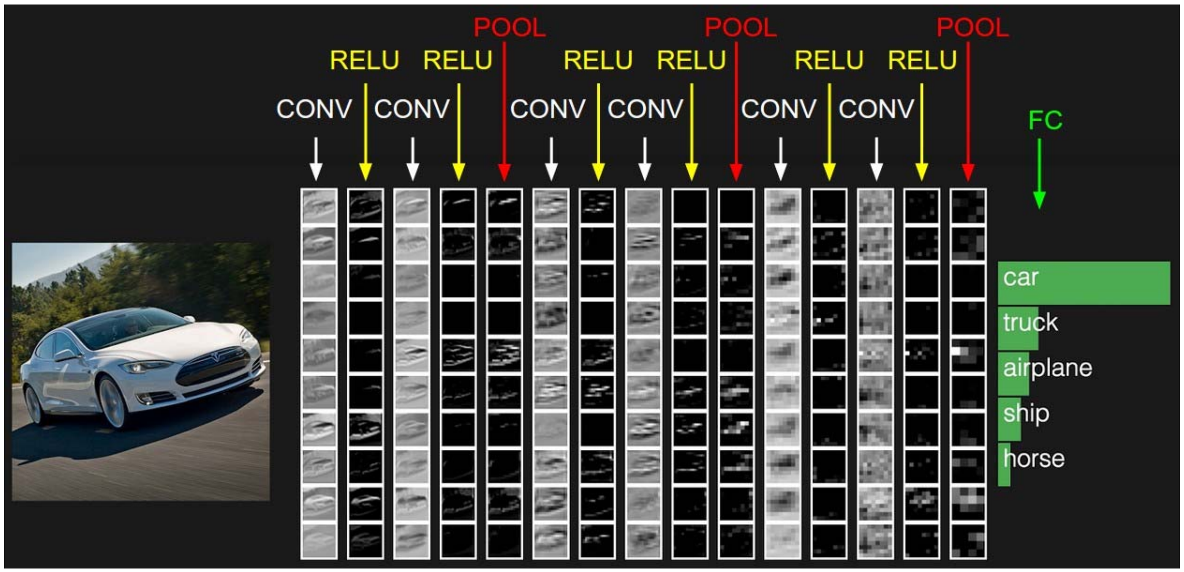
[Sadigh et al., IROS’16]



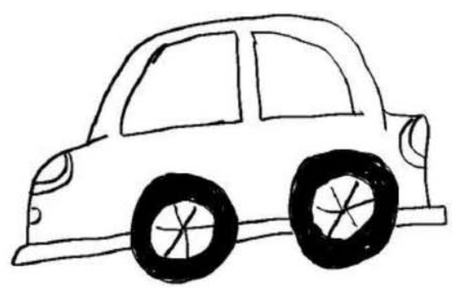
Challenge 2: Formal Specification -- Principle: Go System Level (i.e. Specify Semantic Behavior of the Overall System)

What's the Specification for Perception Tasks?

Convolutional Neural Network trained to recognize cars



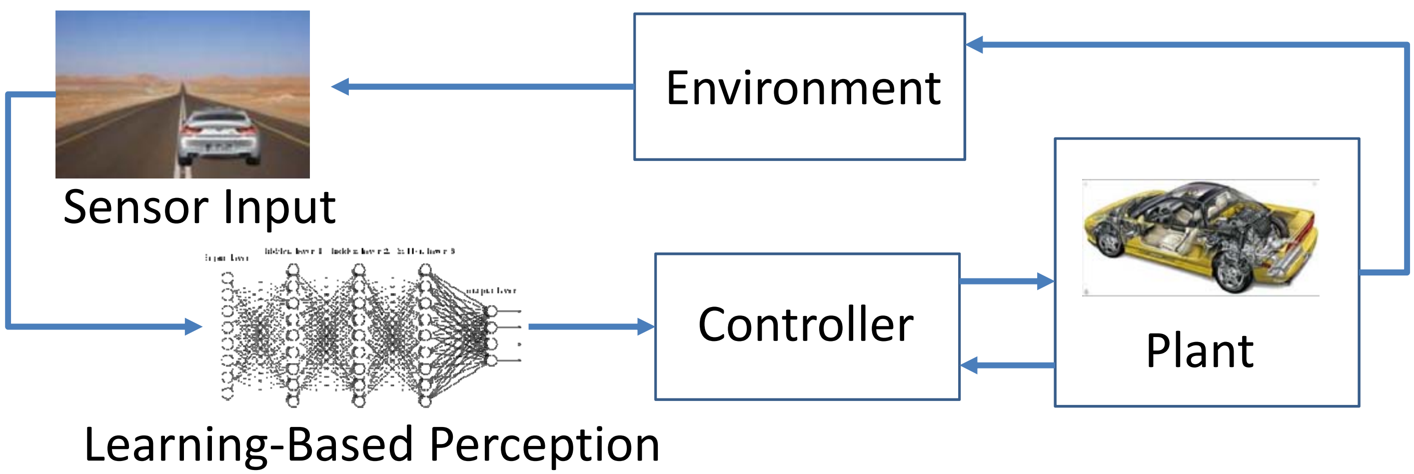
How do you formally specify "a car"?



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)

Bridging Boolean and Quantitative Specs.

- Boolean specification: Traces → {true,false}

- Quantitative specification: Traces → $\mathbf{R}$
  - (or some numerical domain)
  - E.g. a cost/reward function

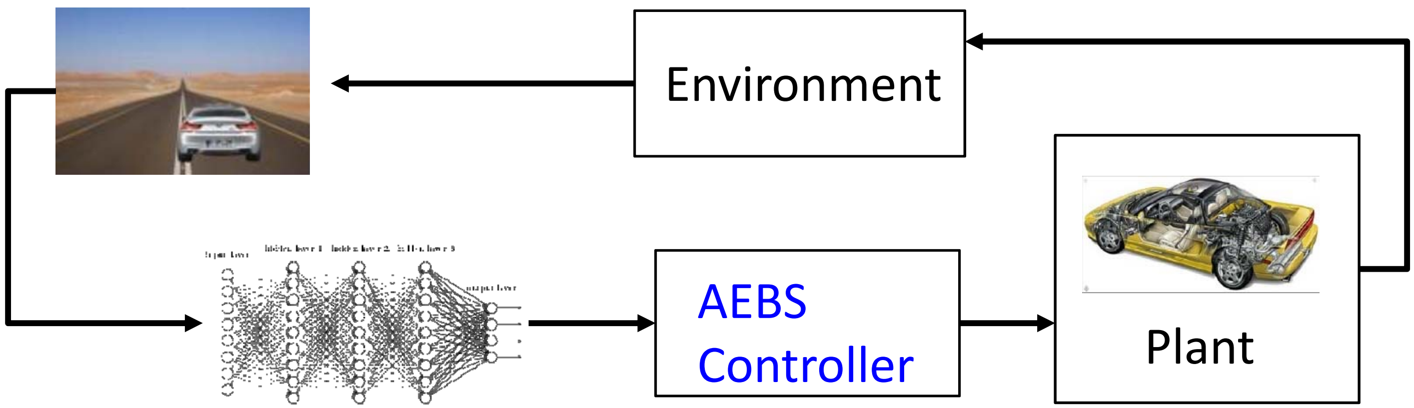
Quantitative specs. more common in AI/ML

- How to bridge the gap?

Challenge 3: Learning Systems Representation/Modeling -- Principle: Abstract and Explain

Challenge 4: Efficient Training, Testing, and Verification -- Principle: Semantic Adversarial Analysis and Compositional Methods

The Problem: Verify Automatic Emergency Braking System (AEBS)

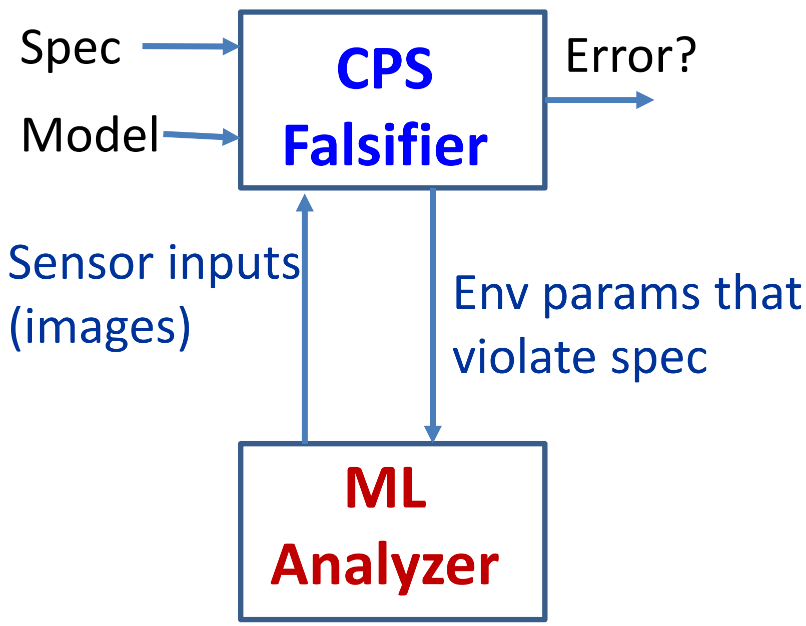


Deep Learning-Based Object Detection

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

- 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
  - more recent: squeezeDet, Yolo, … trained on KITTI

Our Approach: Combine Temporal Logic CPS Falsifier with ML Analyzer

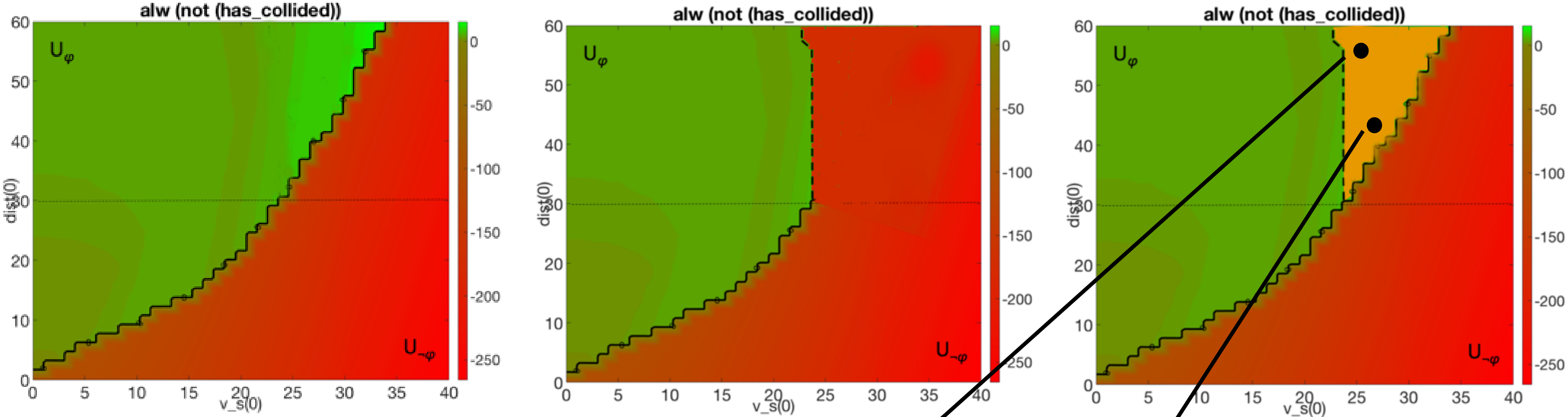


- CPS Falsifier uses abstraction of ML component
  - Optimistic analysis: assume ML classifier is always correct
  - Pessimistic analysis: assume classifier is always wrong
- Difference is the region of interest where output of the ML component "matters"

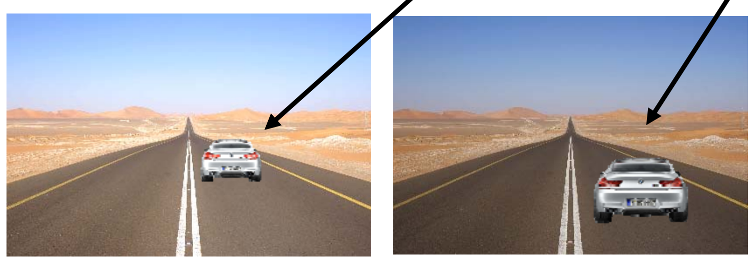
Compositional:

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

Identifying Region of Uncertainty (*ROU*) for Automatic Emergency Braking System

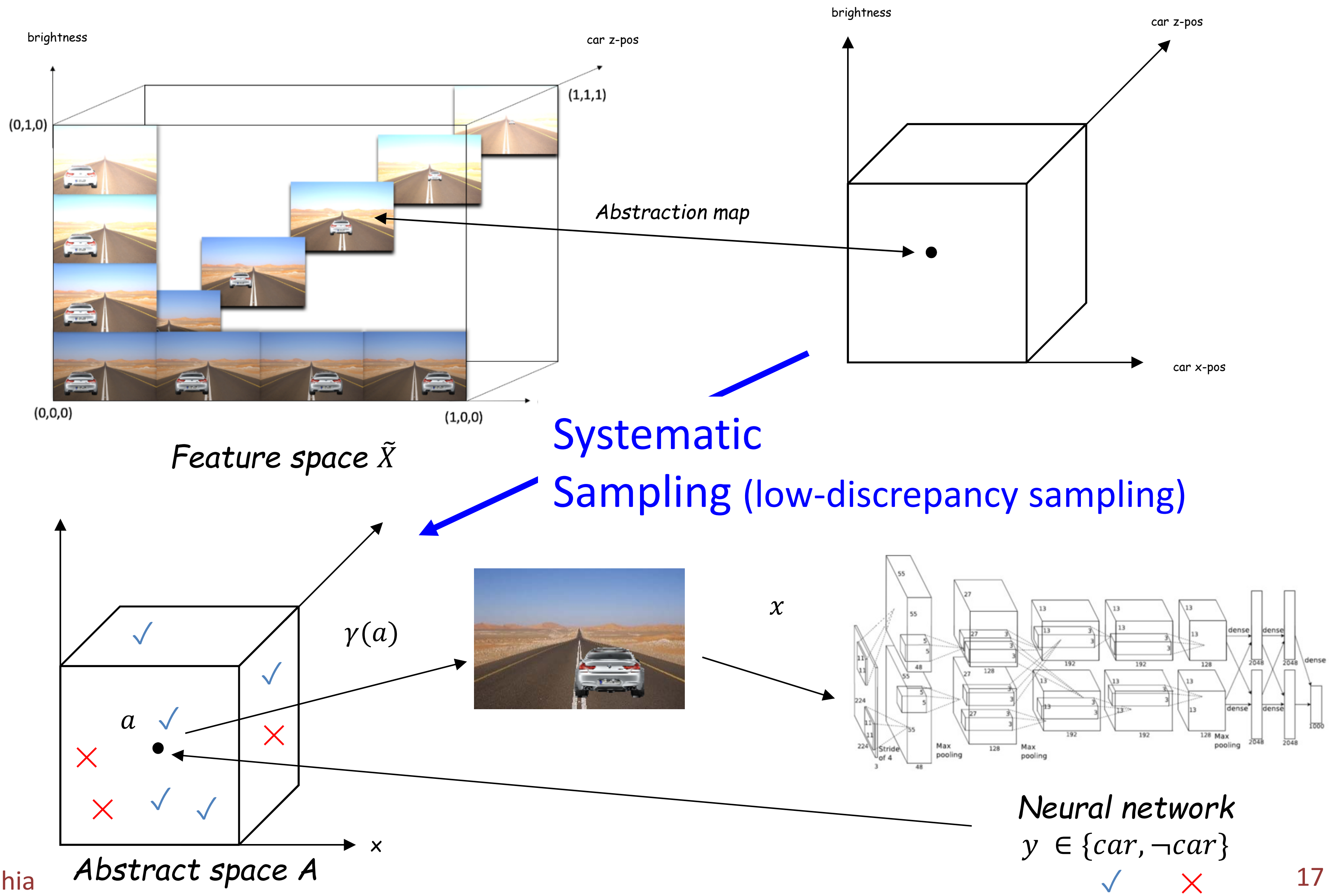


ML always correct *ML always wrong* *Potentially unsafe region depending on ML component (yellow)*



Machine Learning Analyzer

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



Sample Result

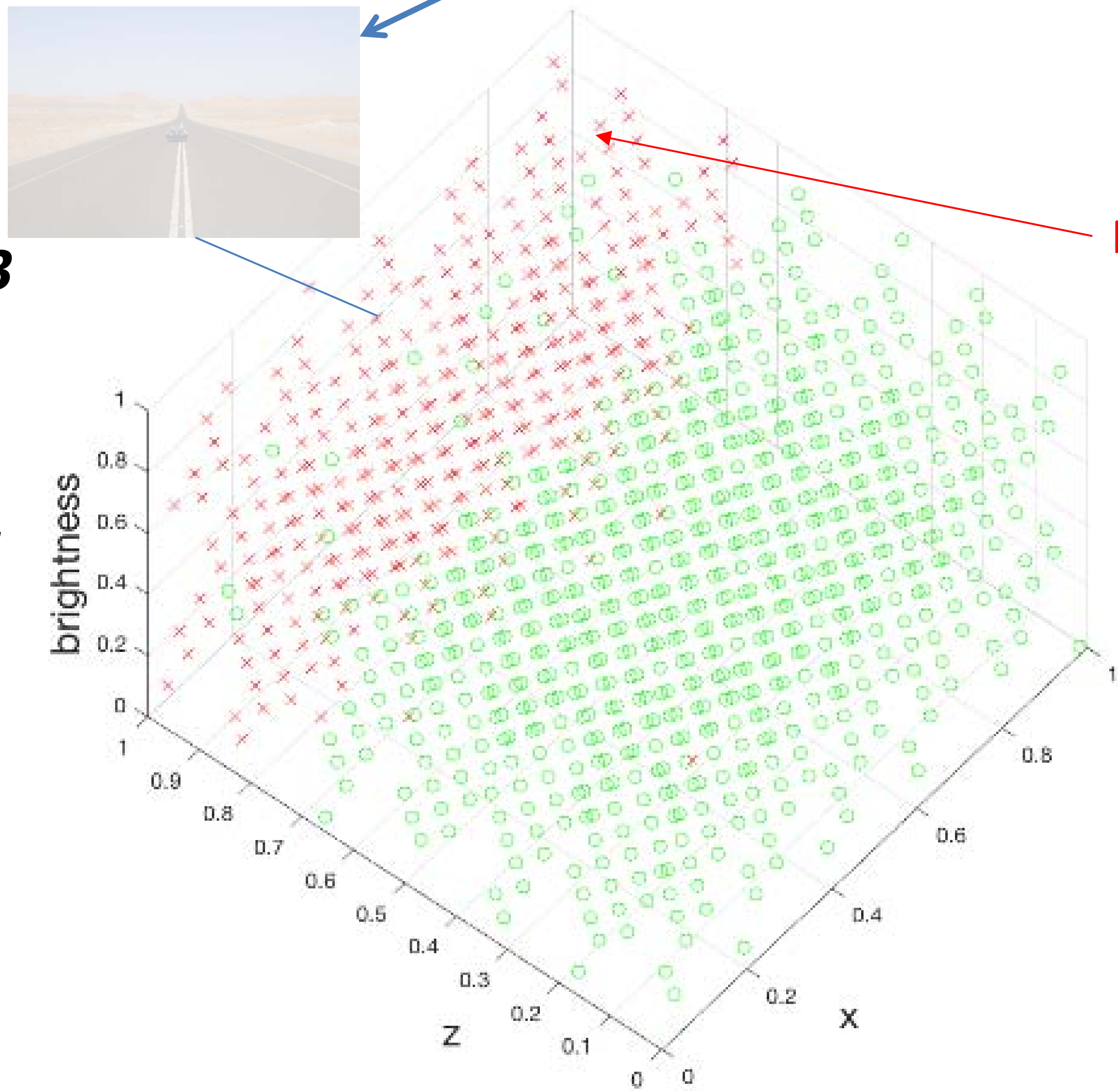
This misclassification may not be of concern

Misclassifications

Inception-v3

Neural Network

(pre-trained on ImageNet using TensorFlow)



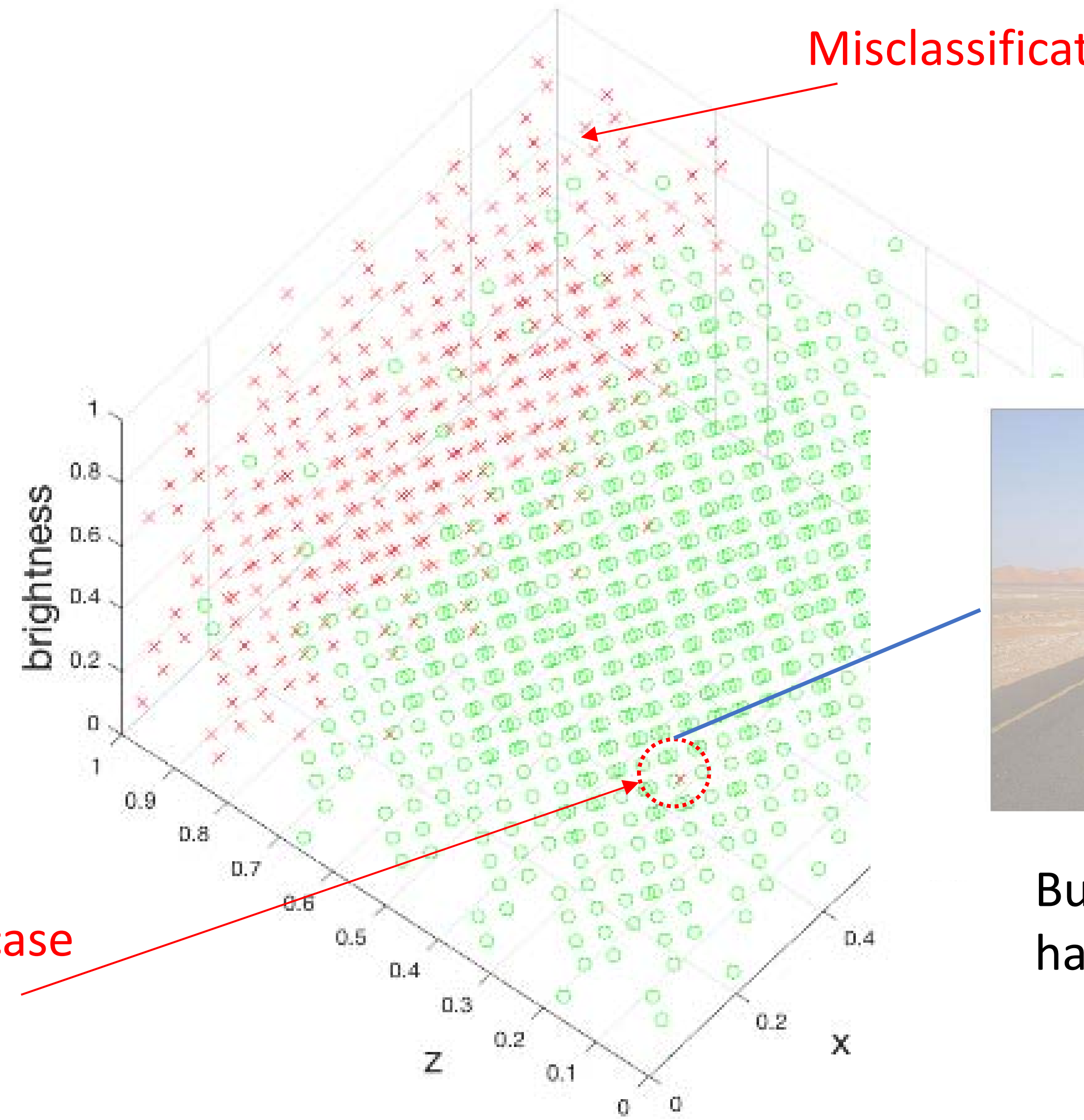
Sample Result

Inception-v3

Neural Network

(pre-trained on ImageNet using TensorFlow)

Misclassifications



Corner case Image

But this one is a real hazard!

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

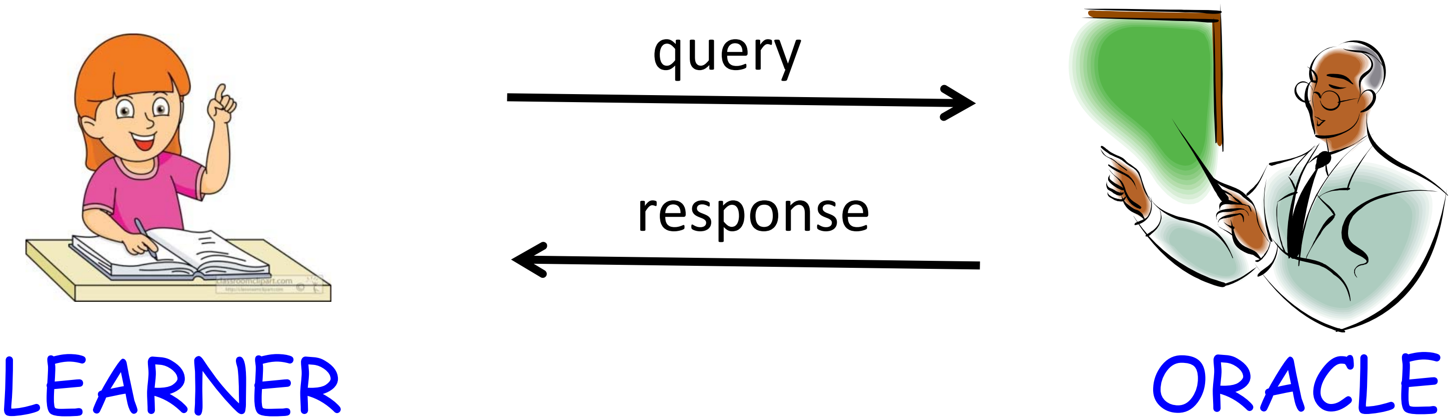
Correct-by-Construction Design with Formal Inductive Synthesis

Inductive Synthesis: Learning from Examples (ML)

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

Key Idea: **Oracle-Guided Learning**

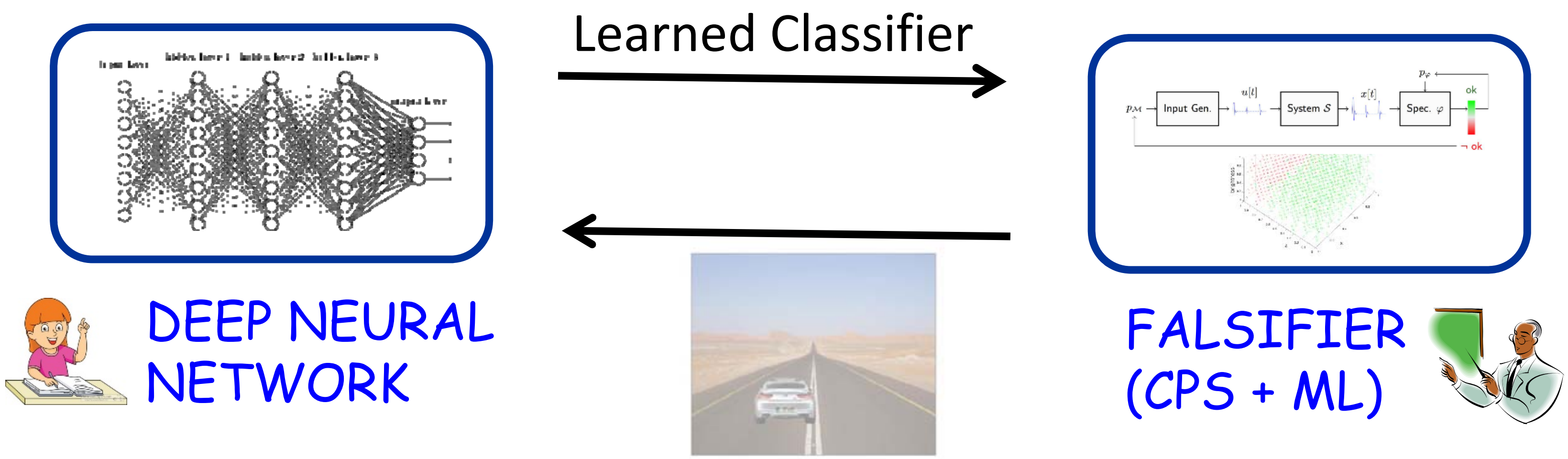
Combine Learner with Oracle (e.g., Verifier) that answers Learner's Queries



[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



"Counterexample-Guided Data Augmentation", T. Dreossi, S. Ghosh, X. Yue, K. Keutzer, A. Sangiovanni-Vincentelli, S. A. Seshia, IJCAI 2018.

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 Representation	→	Abstract & Explain
4. Efficient Training, Testing, Verification	→	Semantic Adversarial Analysis and Compositional Methods
5. Design for Correctness	→	Formal Inductive Synthesis

Exciting Times Ahead!!!

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