Principles for Verified Artificial Intelligence

Sanjit A. Seshia

Professor

EECS, UC Berkeley

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

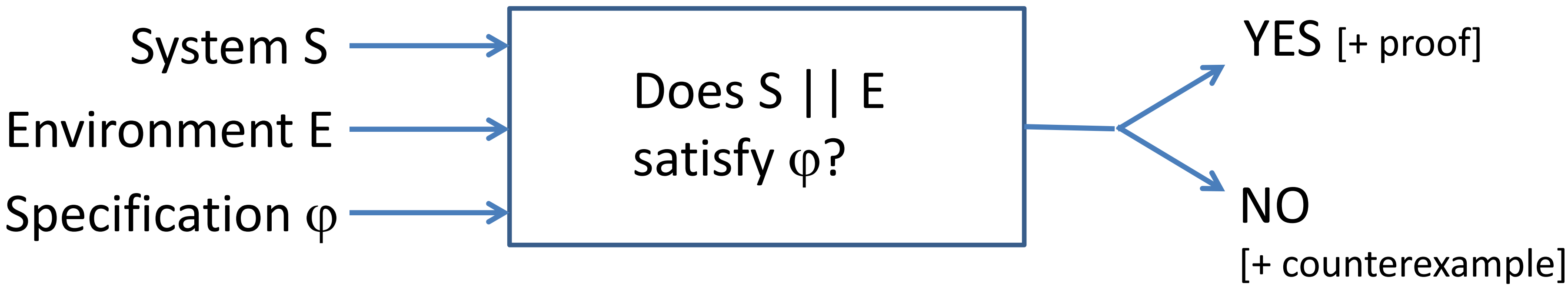
DAC 2017
June 21, 2017

AI / Cognitive Systems / Learning Systems

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

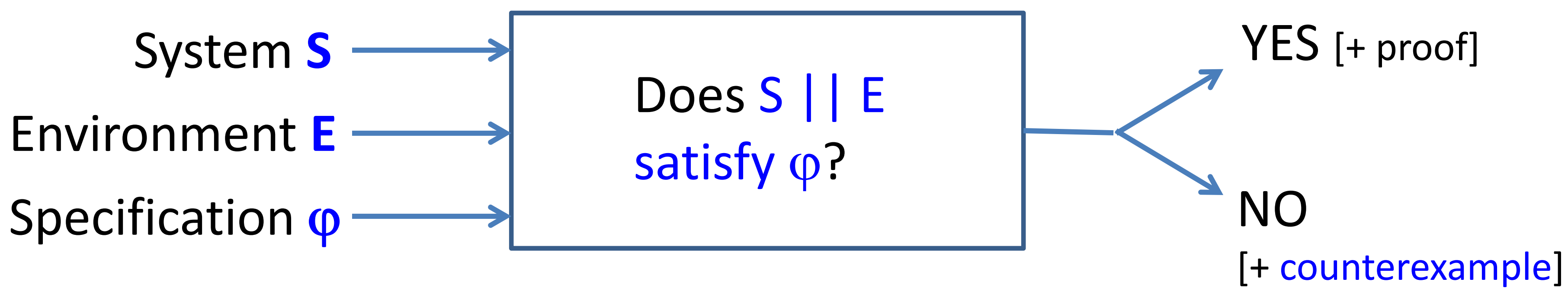
Formal Methods / Verification

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



Five Challenges

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



#1: Environment Modeling – Too Many Unknowns

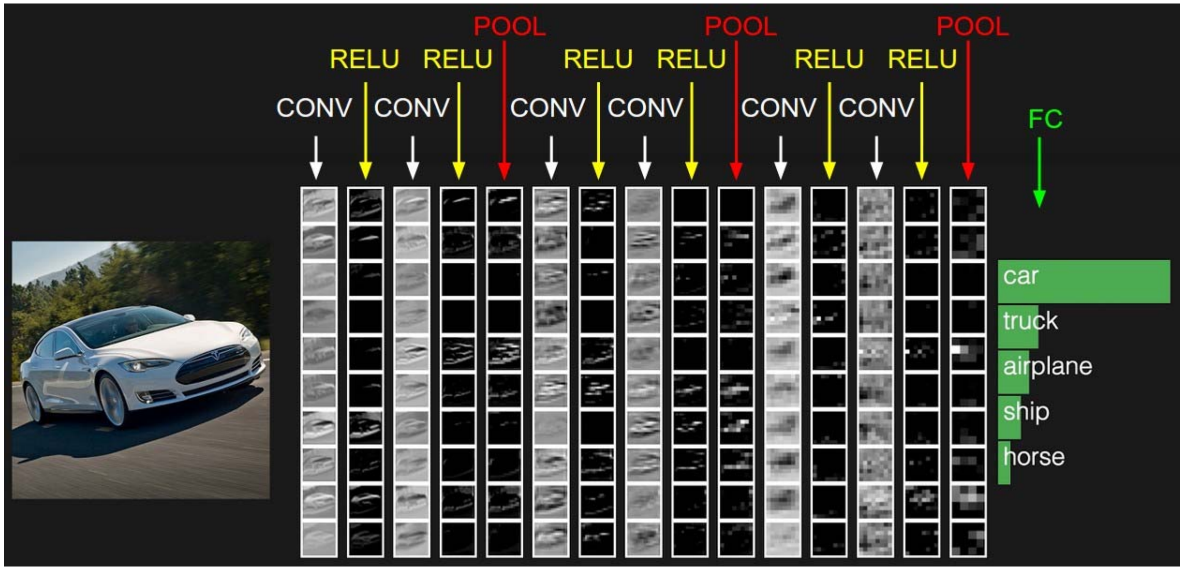
Self-Driving Vehicles: Significant use of machine learning!

Known Unknowns and
Unknown Unknowns!!

“Open World” situation

#2: What's the Specification?

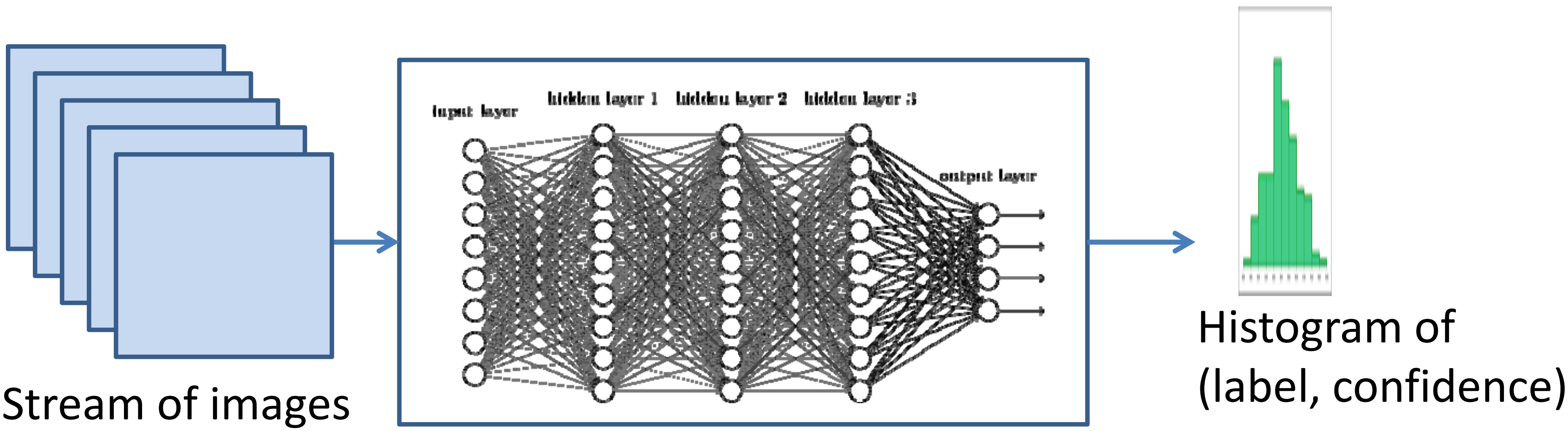
Convolutional Neural Network trained to recognize cars



How do you formally specify "a car"?

#3: Learning Systems Evolve Continuously

How do you model a system that changes over time?



Need a suitable Abstraction -- but how to abstract?

- Over-approximate?
- Under-approximate?
- …?

#4: Intelligent Training / Testing of ML Components

"11 billion miles of (diversified) driving for autonomous vehicles to be just 20% safer than humans"

-- RAND Corporation Report, 2016

Systematic, Algorithmic Simulation / Testing will be necessary!

"Street sign"

Mutation-based

Test data generation

"Bird house"

[Huang et al., 2016]

Fooling classifiers is not hard…

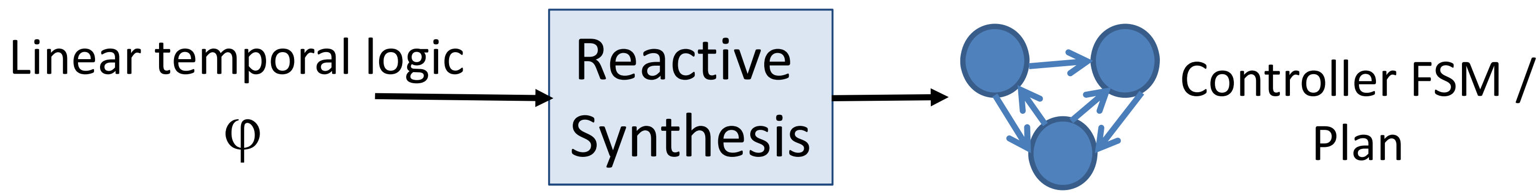
Challenge: Generate not just "Big Data" for training, but the "Right Data"!!!

#5: New Design Methods for Learning Systems

Can we design AI/cognitive systems to be "correct-by-construction"?

<u>Analogies:</u>

1. Synthesis from Temporal Logic (popular in control/robotics)



2. RTL Synthesis

Challenge: Verification is hard enough…
… how are we going to do Synthesis?!!!

Five Principles for Verified AI

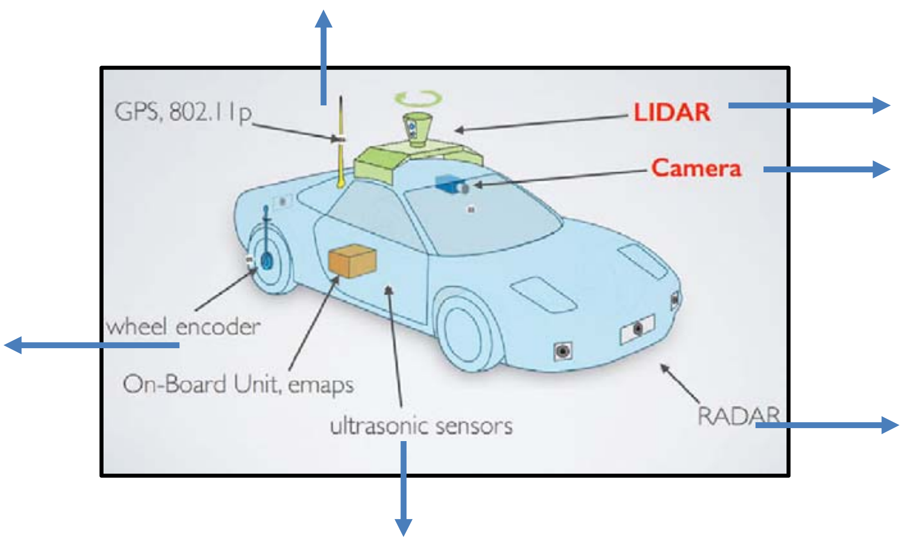
S. A. Seshia, D. Sadigh, S. S. Sastry. *Towards Verified Artificial Intelligence*. July 2016. https://arxiv.org/abs/1606.08514.

#1: Introspective Environment Modeling

Open World modeling problem

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

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



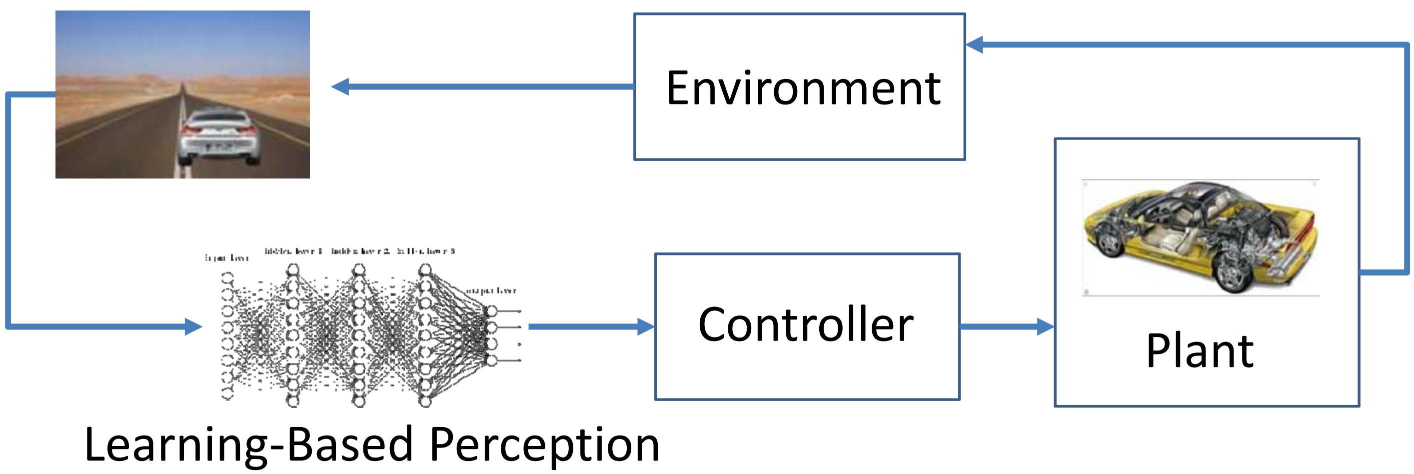
Algorithmic techniques to *generate weakest interface assumptions* and *monitor them at run-time* for potential violation/mitigation

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

#2: Specification: Go System-Level

- ✗ “Verify the Deep Neural Network”
- ✓ “Verify the System containing the Deep Neural Network”

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



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

#3: Learning Systems: Abstract and Explain

1. Function Approximation for ML Component
   – Even simple approximations can be useful for test generation [Dreossi, Donze, Seshia; NFM 2017]
2. Represent Confidence Regions in the System Model
   – E.g. Convex MDP model [Puggelli, Li, et al.; CAV 2013]
3. Learners should accompany output labels with "explanations"
   – "I think it's a car because …"

#4: Train Adversarially and Improvise!

- Counterexample-Guided Training Data Generation [Dreossi et al., NFM 2017]
  - Verification/testing tool acts as an "adversary", generates "corner cases"
- Algorithmic Improvisation [Fremont et al., FSTTCS 2015]
  - Similar to generation of stimuli for constrained random verification
  - Generate random data subject to (hard) constraints, quantitative constraints, and distribution (randomness) requirements
- Many other adversarial training methods in the ML Literature (e.g. GANs)

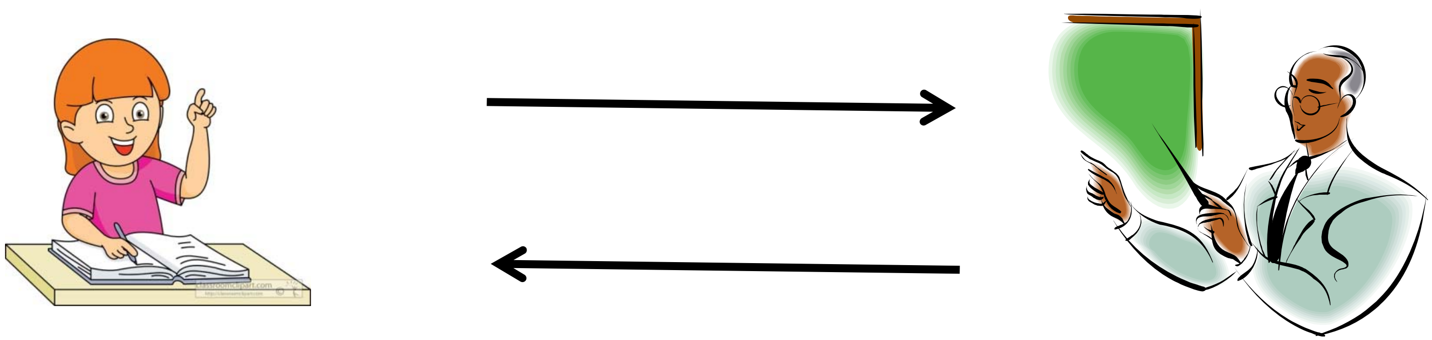
#5: Design with Formal Inductive Synthesis

Inductive Synthesis: Learning from Examples (ML)

Formal Inductive Synthesis: Learn from Examples *while satisfying a Formal Spec.*

Key Idea: **Oracle-Guided Learning**

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



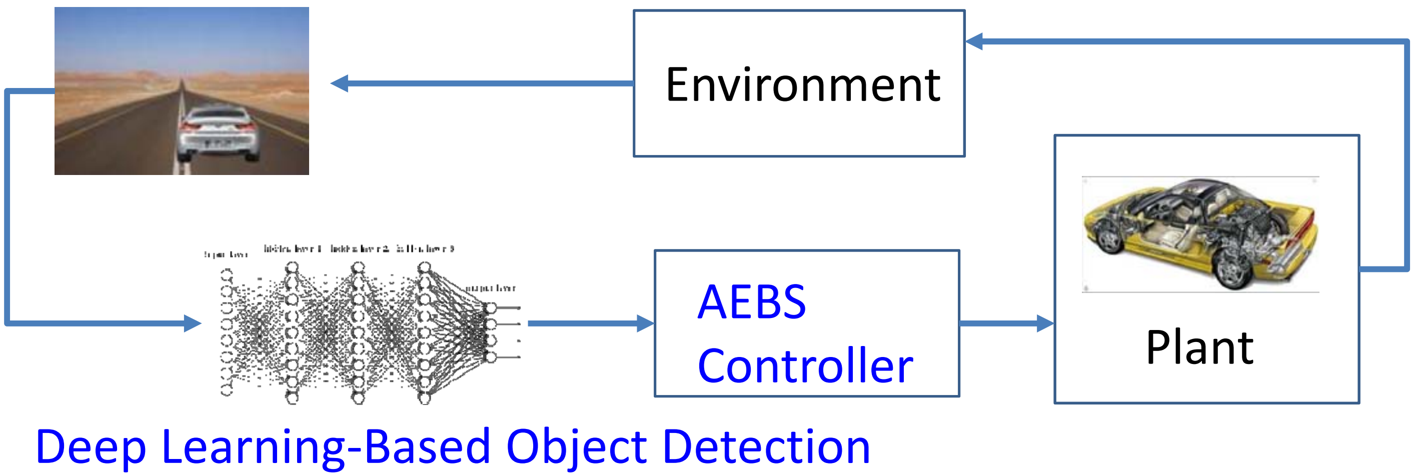
LEARNER ORACLE

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

Recent Results

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.

The Problem: Verify Automatic Emergency Braking System (AEBS)



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

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

Our Approach: Combine Cyber-Physical System (CPS) Falsifier with ML Analyzer

Spec

Model

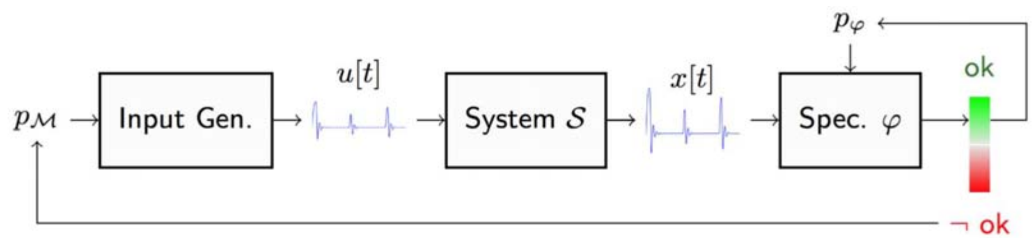
CPS Falsifier

Error?

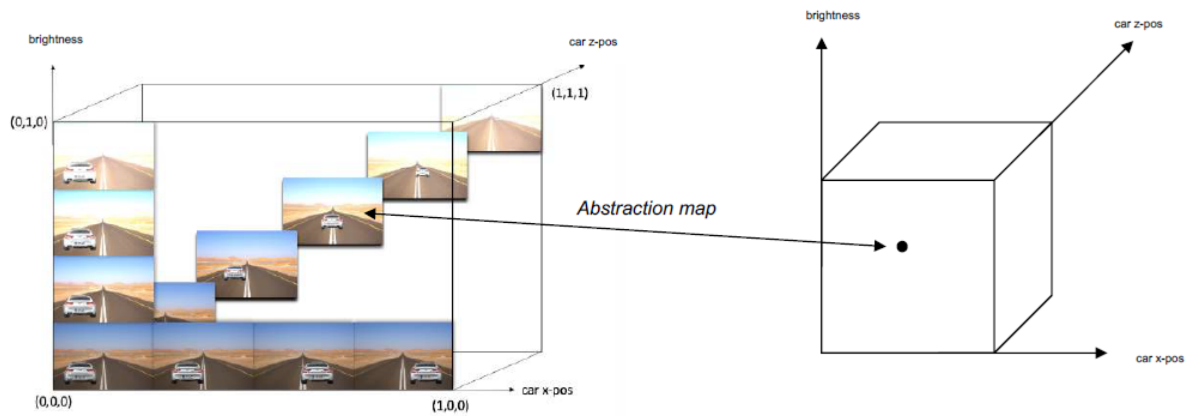
Sensor inputs (images)

Env that violates spec

ML Analyzer



Breach toolkit



Feature space $\tilde{X}$

Abstract space A

Searches for Counterexample Images

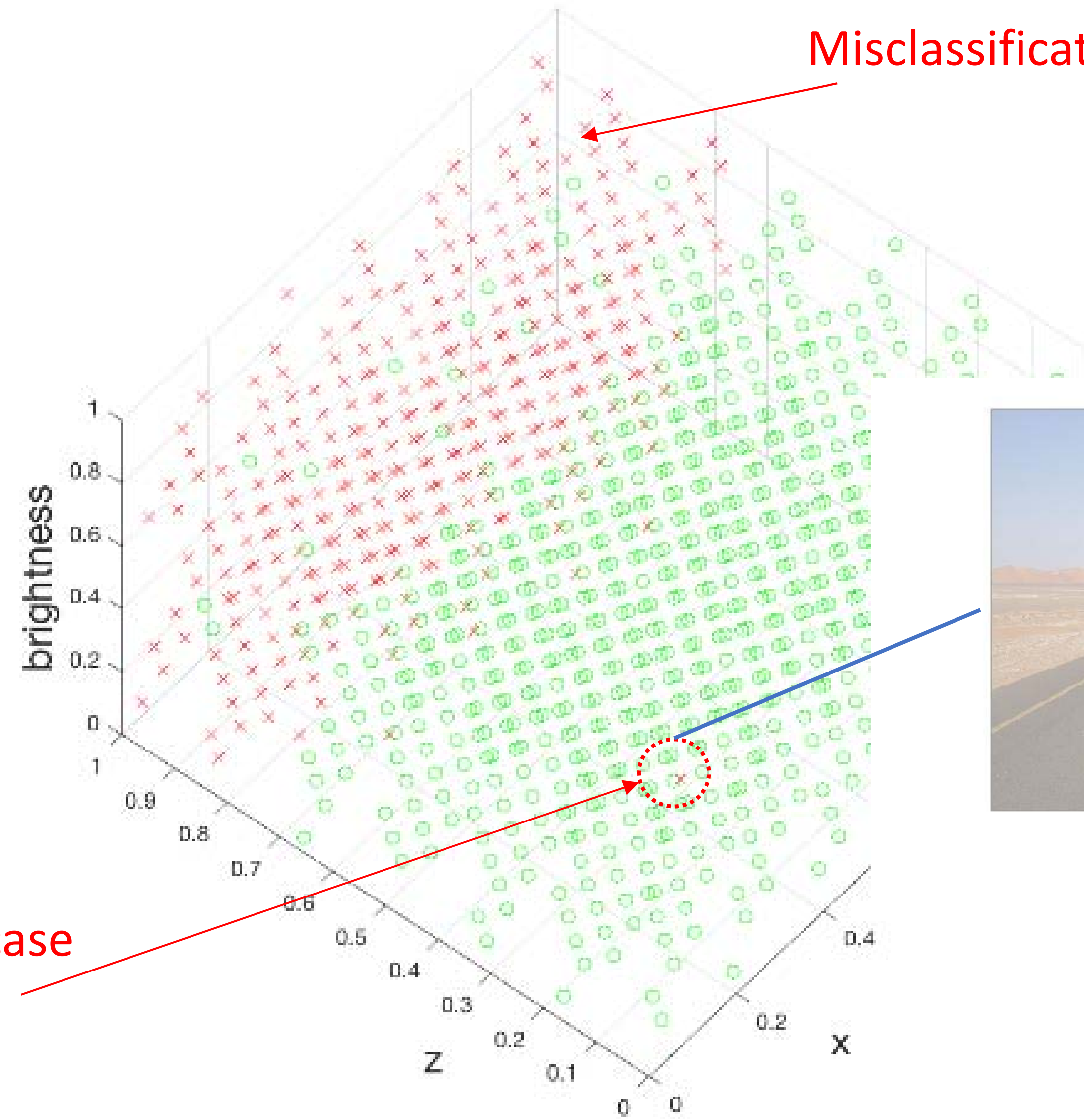
Sample Result

Inception-v3

Neural Network

(pre-trained on ImageNet using TensorFlow)

Misclassifications



Corner case Image

Conclusion: Towards Verified AI / Cognitive Systems

Challenges		**Principles**
1. Environment Modeling	→	Introspective Environment Modeling
2. Specification	→	System-Level Specification
3. Learning Systems Evolve	→	Abstract & Explain
4. Systematic Training / Testing	→	Adversarial Analysis and Improvisation
5. Design for Correctness	→	Formal Inductive Synthesis

Exciting Times Ahead!!! Thank you!