How We **Test** Self-Driving Cars

And How We **Explain Their Failures**

Leilani H. Gilpin
Assistant Professor
Dept. of Computer Science & Engineering, UC Santa Cruz
Introduction and Disclaimer: My Car Failures
Motivate problem: Complex systems are prone to failure

Local sanity checks for vehicle perception

Explanations as an Internal Debugging Language for Complex Systems

Ongoing Work: Testing Autonomous Vehicles by Augmenting Datasets

Question: What are the eXplanatory AI (XAI) methods for testing autonomous vehicles in safety-critical scenarios?
Complex Systems Fail in Complex Ways

Autonomous Vehicle Solutions are at Two Extremes

Serious safety lapses led to Uber’s fatal self-driving crash, new documents suggest

Problem: Need better common sense and reasoning

My Herky-Jerky Ride in General Motors' Ultra-Cautious Self Driving Car

GM and Cruise are testing vehicles in a chaotic city, and the tech still has a ways to go.
Architecture Inspired by Human Organizations

Communication and Sanity Checks

1. Hierarchy of overlapping committees.
2. Continuous interaction and communication.
3. When failure occurs, a story can be made, combining the members’ observations.
An Architecture to Mitigate Common Problems

Reconcile conflicting reasons.

Local Sanity Checks

Justify new examples.
An Existing Problem

The Uber Accident
The best option is to veer and slow down. The vehicle is traveling too fast to suddenly stop. The vision system is inconsistent, but the lidar system has provided a reasonable and strong claim to avoid the object moving across the street.
Agenda

Motivate problem: Complex systems are prone to failure

Local sanity checks for vehicle perception

Explanations as an internal debugging language

Ongoing Work: Testing Autonomous Vehicles by Augmenting Datasets
A Neural Network Labels Camera Data

Inception Network - Google

Label

e.g. pedestrian
Problem: Neural Networks are Brittle

For self-driving, and other mission-critical, safety-critical applications, these mistakes have CONSEQUENCES.

Monitor Opaque Subsystems for Reasonableness

1. Judgement of reasonableness
2. Justification of reasonableness

Opaque Mechanism

Commonsense Knowledge Base + Flexible Representation + Identify (Un)reasonability + Justify (Un)reasonability

Label e.g. pedestrian
Reasonable because…

Unreasonable because…

Flexible Representation

Identify (Un)reasonability

Justify (Un)reasonability

Opaque Mechanism

Supplement with Commonsense Knowledge Base

Reasonable because…

Unreasonable because…
Flexible Representation

Opaque Mechanism

Reasonable because…

Unreasonable because…

Common sense Knowledge Base

parser representation
Primitive Representations
Encode Understanding

11 primitives to account for most actions:
- ATRANS
- ATTEND
- INGEST
- EXPEL
- GRASP
- MBUILD
- MTRANS
- MOVE
- PROPEL
- PTRANS
- SPEAK

5 for physical actions
Extended to vehicle primitives

Conceptual Dependency Theory
(CD), Schank 1975
A woman crossing the street.

**Data from Nuscenes**

**Parser**

- S
  - NP: woman
  - VP: crossing the street

- D: person
  - MOVE: person
  - street
  - person
Representations with Implicit Rules

A perceived frame is REASONABLE

\[ ((x_1, p_1, y_1), \text{isA, REASONABLE}) \land ((x_2, p_2, y_2), \text{isA, REASONABLE}) \land \ldots \land ((x_n, p_n, y_n), \text{isA, REASONABLE}) \]

Move Primitive Reasonability

\( (x, \text{hasProperty, animate}) \land (x, \text{locatedNear, y}) \Rightarrow ((x, \text{MOVE, y}) \text{isA, REASONABLE}) \)
Reasonableness Monitoring on Real Data
NuScenes

{"token": '70aecbe9b64f4722ab3c230391a3bebb8',
'sample_token': 'cd21dbf3bd749c7b10a5c42562e0c42',
'instance_token': '6dd2cbf4c24b4caeb625035869bca7b5',
'visibility_token': '4',
'attribute_tokens': ['4d8821270b4a47e3a8a300cbec48188e'],
'translation': [373.214, 1130.48, 1.25],
'size': [0.621, 0.669, 1.642],
'rotation': [0.9831098797903927, 0.0, 0.0, -0.18301629506281616],
'prev': 'a1721876c0944cdd92ebc3c755d55d693',
'next': '1e8e35d365a441a18dd5503a0ee1c208',
'num_lidar_pts': 5,
'num_radar_pts': 0,
'category_name': 'human.pedestrian.adult'}

Data from NuScenes
Commonsense is Unorganized
ConceptNet

adult is a type of...

- animal (n, wn) ➔
- person (n, wn) ➔
- animal (n) ➔

adult is capable of...

- help a child ➔
- dress herself ➔
- sign a contract ➔
- drink beer ➔
- work ➔
- act like a child ➔
- dress himself ➔
- drive a car ➔
- drive a train ➔
- explain the rules to a child ➔

(‘adult, ‘typeOf, ‘animal)
(‘adult, ‘isA, ‘bigger than a child’)
...

Data from NuScenes
This perception is reasonable. An adult is typically a large person. They are usually located walking on the street. Its approximate dimensions of [0.621, 0.669, 1.642] is approximately the correct size in meters.
Agenda

Motivate problem: Complex systems are prone to failure

Local sanity checks for vehicle perception

Explanations as an internal debugging language

Ongoing Work: Testing Autonomous Vehicles by Augmenting Datasets
Reconciling Internal Disagreements
With an Organizational Architecture

- Monitored subsystems combine into a system architecture.
- Explanation synthesizer to deal with inconsistencies.
  - Argument tree.
  - Queried for support or counterfactuals.
Anomaly Detection through Explanations
Reasoning in Three Steps

1. Generate Symbolic Qualitative Descriptions for each committee.
2. Input qualitative descriptions into local “reasonableness” monitors.
3. Use a synthesizer to reconcile inconsistencies between monitors.

- VISION
- LiDAR
- TACTICS
- Brakes
- Steering
- Power
3. Use a synthesizer to reconcile inconsistencies between monitors.

Synthesizer + Priority Hierarchy \rightarrow Abstract Goals

- Explanation synthesizer to deal with inconsistencies.
- Argument tree.
- Queried for support or counterfactuals.

1. Passenger Safety
2. Passenger Perceived Safety
3. Passenger Comfort
4. Efficiency (e.g. Route efficiency)

A passenger is safe if:

- The vehicle proceeds at the same speed and direction.
- The vehicle avoids threatening objects.
3. Use a synthesizer to reconcile inconsistencies between monitors.

\[
(\forall s, t \in \text{STATE}, v \in \text{VELOCITY} \\
\quad ((\text{self}, \text{moving}, v), \text{state}, s) \land \\
\quad (t, \text{isSuccessorState}, s) \land \\
\quad ((\text{self}, \text{moving}, v), \text{state}, t) \land \\
\quad (\forall x \in \text{OBJECTS} \text{ s.t.} \\
\quad ((x, \text{isA}, \text{threat}), \text{state}, s) \lor \\
\quad ((x, \text{isA}, \text{threat}), \text{state}, t)))
\]

\[\Rightarrow (\text{passenger}, \text{hasProperty}, \text{safe})\]

A passenger is safe if:

- The vehicle proceeds at the same speed and direction.
- The vehicle avoids threatening objects.

\[
(\forall s \in \text{STATE}, x \in \text{OBJECT}, v \in \text{VELOCITY} \\
\quad ((x, \text{moving}, v), \text{state}, s) \land \\
\quad ((x, \text{locatedNear}, \text{self}), \text{state}, s) \land \\
\quad ((x, \text{isA}, \text{large_object}), \text{state}, s) \\
\quad \Leftrightarrow ((x, \text{isA}, \text{threat}), \text{state}, s))
\]
The best option is to veer and slow down. The vehicle is traveling too fast to suddenly stop. The vision system is inconsistent, but the lidar system has provided a reasonable and strong claim to avoid the object moving across the street.

Abstract Goal Tree

- 'passenger is safe',
  AND
  'safe transitions',
  NOT('threatening objects')
Uber Example in Simulation

Agenda

Motivate problem: Complex systems are prone to failure

Local sanity checks for vehicle perception

Explanations as an internal debugging language

**Ongoing Work: Testing Autonomous Vehicles by Augmenting Datasets**
Vision: Real World Adversarial Examples

“Realistic” Adversarial examples

Vision: Real World Adversarial Examples
Anticipatory Thinking Layer for Error Detection

The traffic lights are on top of the truck. The lights are not illuminated. The lights are moving at the same rate as the truck, therefore this is not a “regular” traffic light for slowing down and stopping at.

“Realistic” Adversarial examples
Lack of Data and Challenges for AVs

- Existing Challenges
  - Targeted as optimizing a mission or trajectory and not safety.
  - Data is hand-curated.
  - Failure data is not available
    - Unethical to get it (cannot just drive into bad situations).
  - Want the data to be realistic (usually difficult in simulation).

Data from NuScenes
Approach: Content Generation
Anticipatory Thinking Layer for Error Detection

Approach: Content Generation
Anticipatory Thinking Layer for Error Detection

Approach: Content Generation
Anticipatory Thinking Layer for Error Detection

Behaviors that are Inherently Explainable

Contributions

Motivate problem: Complex systems are prone to failure

Local sanity checks for vehicle perception

Explanations as an internal debugging language

Ongoing Work: Testing Autonomous Vehicles by Augmenting Datasets