Explaining Errors in Complex Systems

A Diagnosis Tool and Testing Framework for Robust Decision Making

Leilani H. Gilpin, Assistant Professor
Dept. of CSE, UC Santa Cruz
CSE 200
November 18, 2021

Agenda

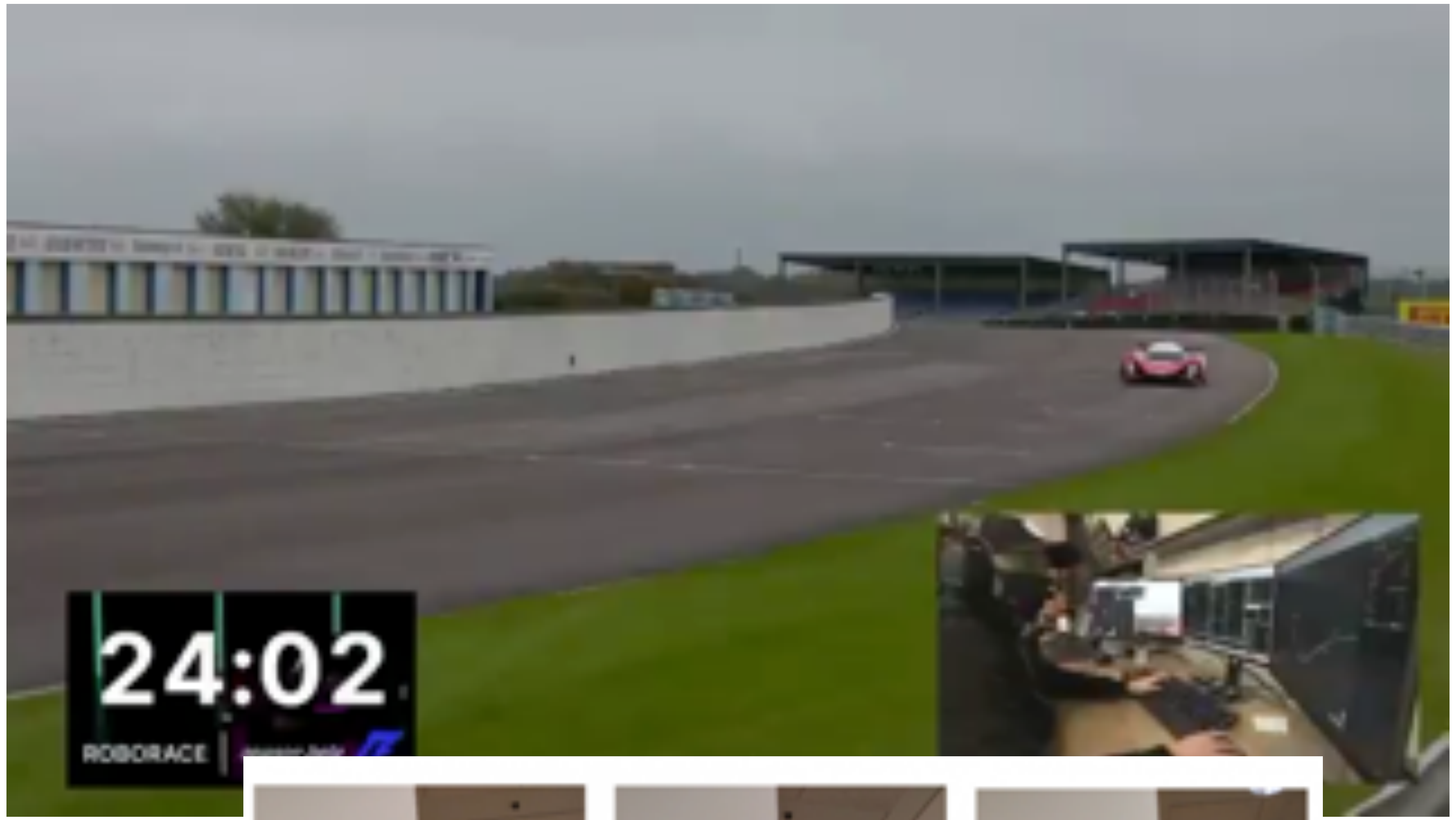
<u>Motivate problem: Autonomous Vehicles are Prone to Failure</u>

Anomaly Detection through Explanations (ADE): a Diagnosis Tool for AVs.

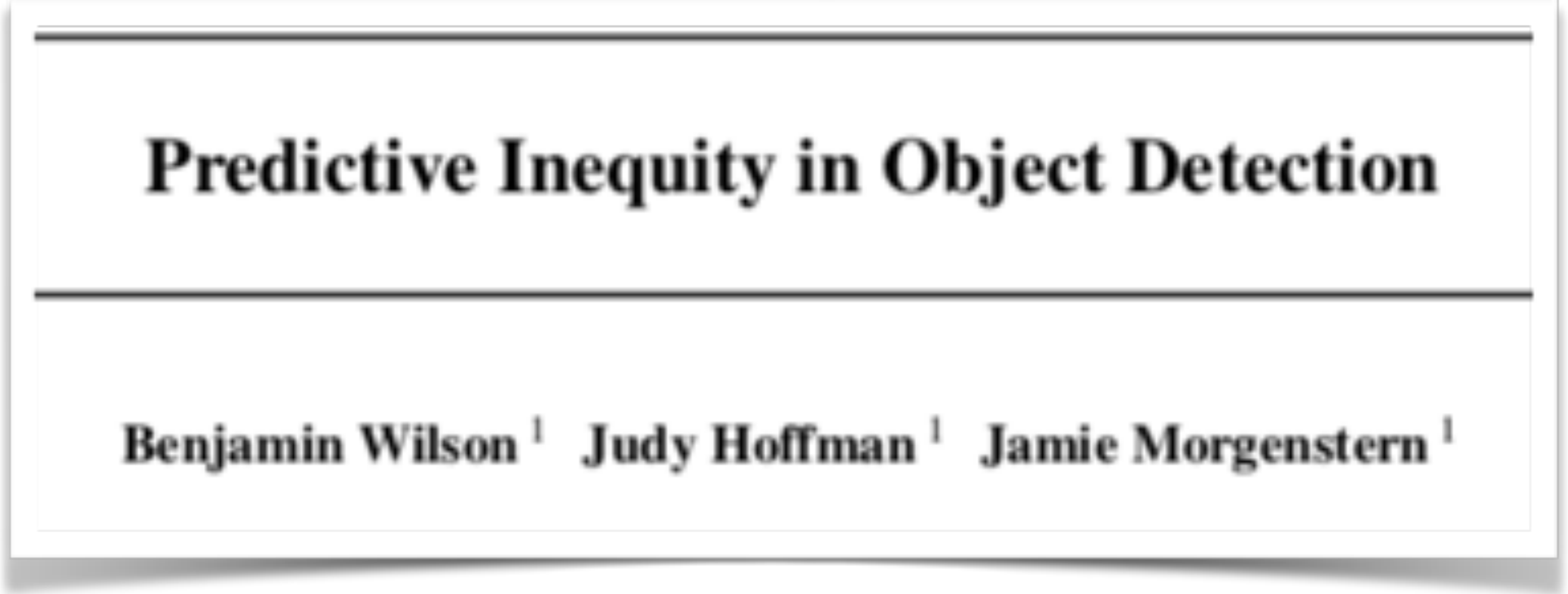
Future work: Explainable Tasks for Robust and Secure Hybrid Systems.

Question: How to develop self-explaining architectures that can help anticipate failures instead of after-the-fact?

Complex Systems Fail in Complex Ways



K. Eykholt et al. “Robust Physical-World Attacks on Deep Learning Visual Classification.”

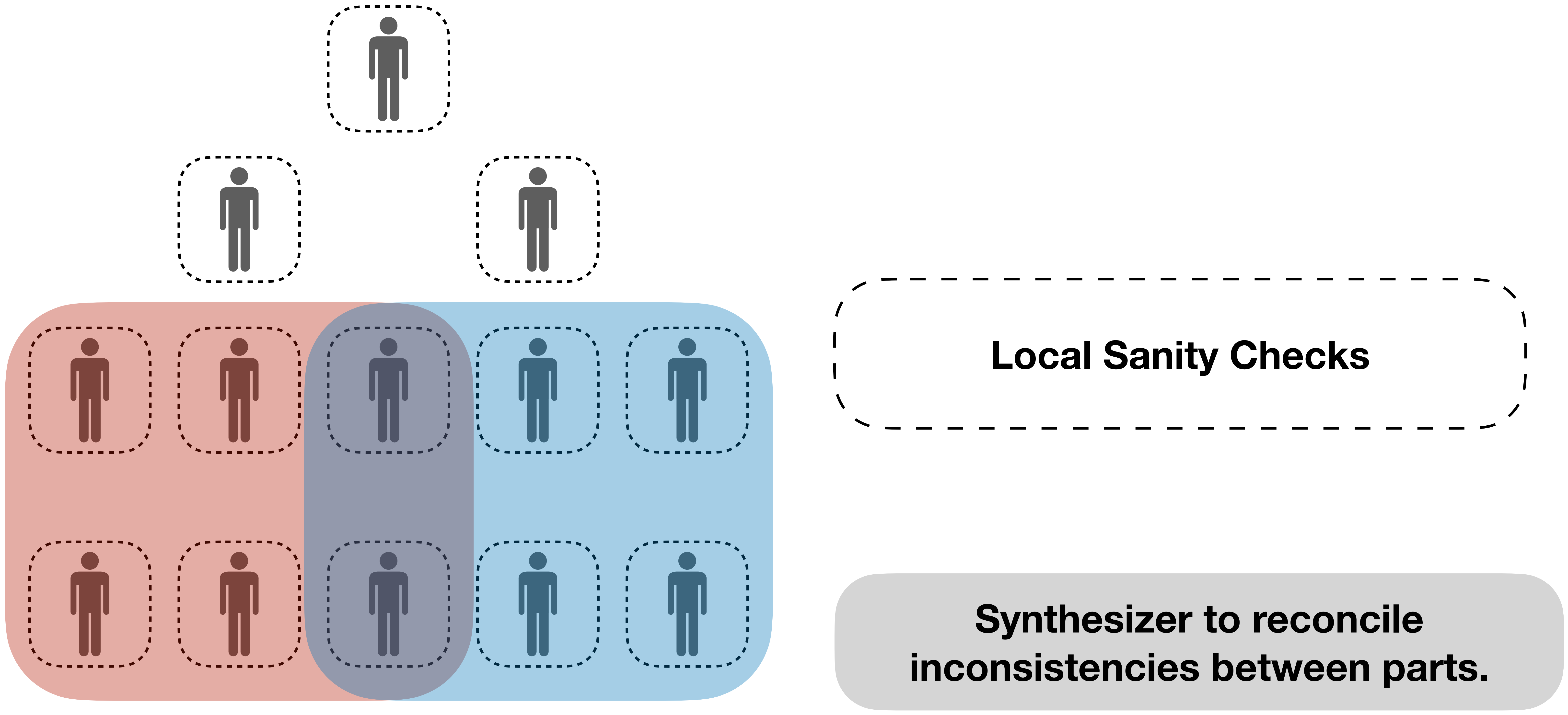

Predictive Inequity in Object Detection

Benjamin Wilson [1] Judy Hoffman [1] Jamie Morgenstern [1]

Autonomous Vehicle Solutions are at Two Extremes

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

Synthesizer to reconcile inconsistencies between parts.

Local Sanity Checks

Reconcile conflicting reasons.


future tense

The Trollable Self-Driving Car

Justify new examples.

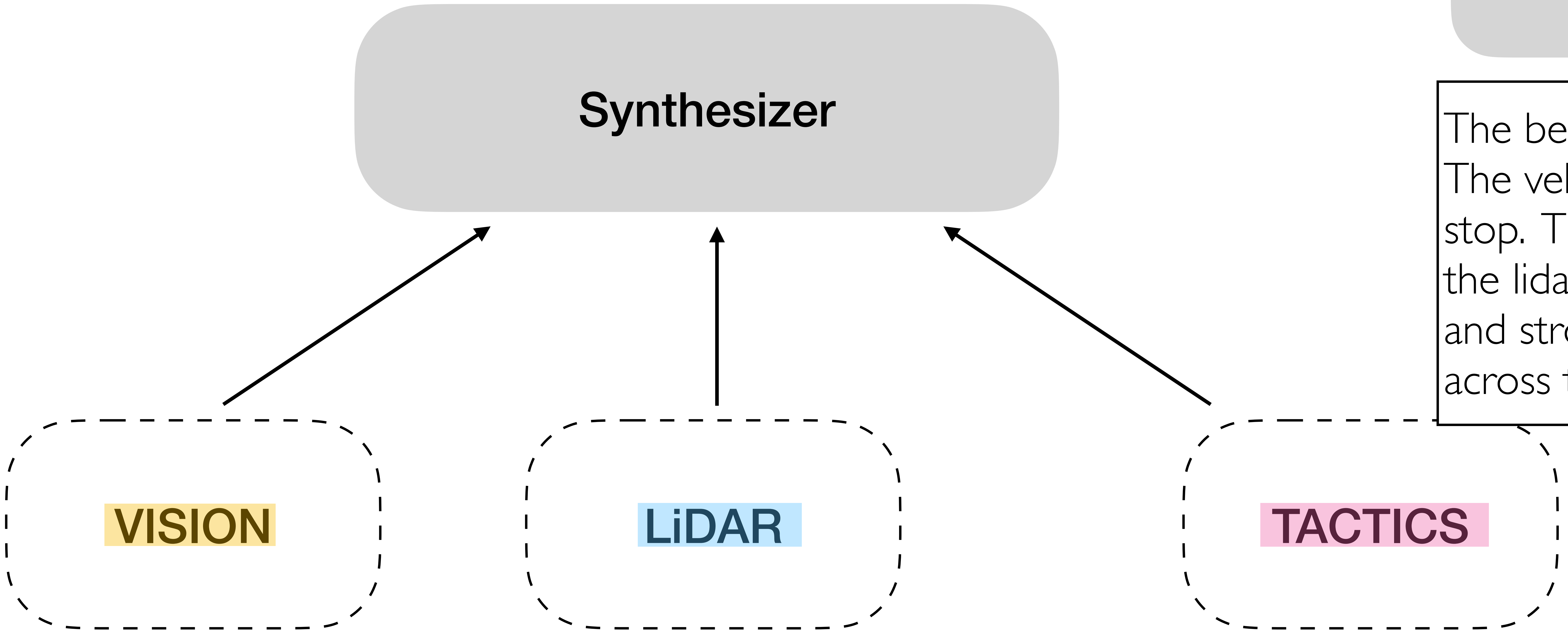
An Existing Problem

The Uber Accident

Solution: Internal Communication

Anomaly Detection through Explanations

Synthesizer to reconcile inconsistencies between monitor outputs.



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: Autonomous Vehicles are Prone to Failure

Anomaly Detection through Explanations (ADE): a Diagnosis Tool for AVs.

Future work: Explainable Tasks for Robust and Secure Hybrid Systems.

Limited Internal Reasoning

A Google self-driving car caused a crash for the first time

A bad assumption led to a minor fender-bender

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

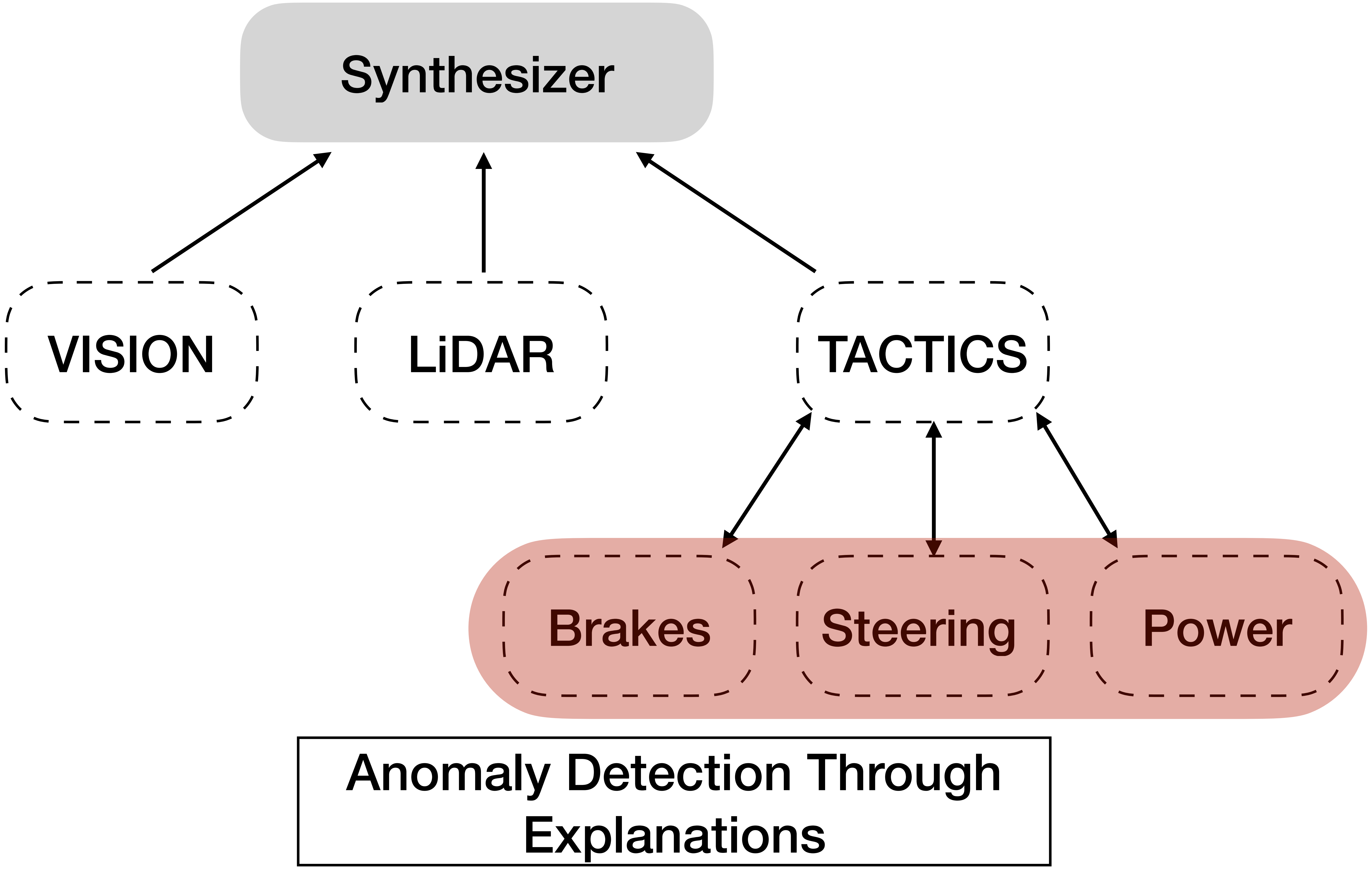
My Herky-Jerky Ride in General Motors' Ultra-Cautious Self Driving Car

GM and Cruise are testing vehicles in a chaotic city, and the tech still has a ways to go.

Reconciling Internal Disagreements

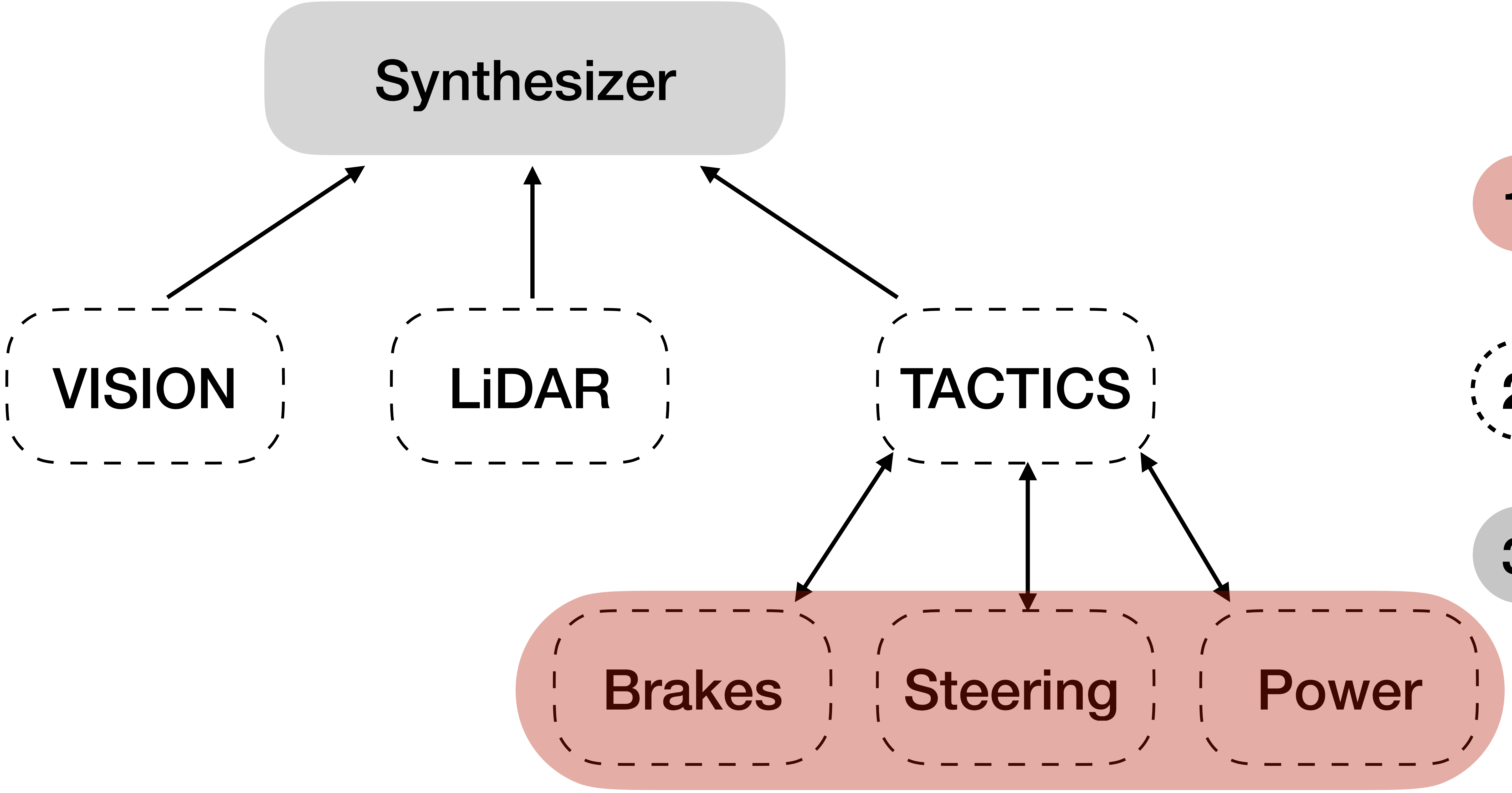
With an Organizational Architecture

- 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.

3. Use a synthesizer to reconcile inconsistencies between monitors.

Synthesizer + Priority Hierarchy ⟶ 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.

$$
\begin{aligned}
(\forall s, t \in \mathit{STATE}, v \in \mathit{VELOCITY}\\
\big((\mathit{self}, \mathit{moving}, v), \textbf{state}, s\big) \wedge\\
(t, \textbf{isSuccesorState}, s) \wedge\\
\big((\mathit{self}, \mathit{moving}, v), \textbf{state}, t\big) \wedge\\
(\nexists x \in \mathit{OBJECTS}\ \textbf{s.t.}\\
\big((x, \mathit{isA}, \mathit{threat}), \textbf{state}, s\big) \vee\\
\big((x, \mathit{isA}, \mathit{threat}), \textbf{state}, t\big)))\\
\Rightarrow \big(\textbf{passenger, hasProperty, safe}\big)
\end{aligned}
$$

A passenger is safe if:

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

$$
\begin{aligned}
(\forall s \in \mathit{STATE}, x \in \mathit{OBJECT}, v \in \mathit{VELOCITY}\\
\big((x, \mathit{moving}, v), \textbf{state}, s\big) \wedge\\
\big((x, \mathit{locatedNear}, \mathit{self}), \textbf{state}, s\big) \wedge\\
\big((x, \mathit{isA}, \mathit{large\_object}), \textbf{state}, s\big)\\
\Leftrightarrow \big((x, \mathit{isA}, \mathit{threat}), \textbf{state}, s\big))
\end{aligned}
$$

3. Use a synthesizer to reconcile inconsistencies between monitors.

$$
\begin{aligned}
&(\forall s, t \in STATE, v \in VELOCITY \\
&\quad \big((self, moving, v), \textbf{state}, s\big) \wedge \\
&\quad (t, \textbf{isSuccesorState}, s) \wedge \\
&\quad \big((self, moving, v), \textbf{state}, t\big) \wedge \\
&\quad (\nexists x \in OBJECTS \textbf{ s.t.} \\
&\quad \big((x, isA, threat), \textbf{state}, s\big) \vee \\
&\quad \big((x, isA, threat), \textbf{state}, t\big))) \\
&\qquad \Rightarrow \big(\textbf{passenger, hasProperty, safe}\big)
\end{aligned}
$$

Abstract Goal Tree

```
'passenger is safe',
AND(
  'safe transitions',
  NOT('threatening objects')
```

3. Use a synthesizer to reconcile inconsistencies between monitors.

Abstract Goal Tree

```
'passenger is safe',
AND(
  ‘safe transitions’,
  NOT(‘threatening objects’)
```

List of Rules → Backwards Chain → AND/OR TREE

```
IF ( AND('moving (?v) at state (?y)',
             '(?z) succeeds (?y)',
             'moving (?v) at state (?z)'),
     THEN('safe driving at (?v) during (?y) and (?z)'))

IF (OR('obj is not moving',
       'obj is not located near',
       'obj is not a large object')),
    THEN('obj not a threat at (?x)'))

IF (AND('obj not a threat at (?y)',
        'obj not a threat at (?z)',
        '(?z) succeeds (?z)',
    THEN('obj is not a threat between (?y) and (?z)'))
```

```
passenger is safe at V between s and t
  AND (AND (moving V at state s
            t succeeds s
            moving V at state t )
        AND (
            OR ( obj is not moving at s
                 obj is not locatedNear at s
                 obj is not a large object at s )
            OR ( obj is not moving at t
                 obj is not locatedNear at t
                 obj is not a large object at t ) ) )
```

3. Use a synthesizer to reconcile inconsistencies between monitors.

```
(monitor, judgement, unreasonable)
(input, isType, labels)
(all_labels, inconsistent, negRel)
(isA, hasProperty, negRel)
…
(all_labels, notProperty, nearMiss)
(all_labels, locatedAt, consistent)
(monitor, recommend, discount)
```

```
(monitor, judgement, reasonable)
(input, isType, sensor)
…
(input_data[4], hasSize, large)
(input_data[4], IsA, large_object)
(input_data[4], moving, True)
(input_data[4], hasProperty, avoid)
…
(monitor, recommend, avoid)
```

!

```
(monitor, judgement, reasonable)
(input, isType, history)
(input_data, moving, True)
(input_data, direction, forward)
(input_data, speed, fast)
(input_data, consistent, True)
(monitor, recommend, proceed)
```

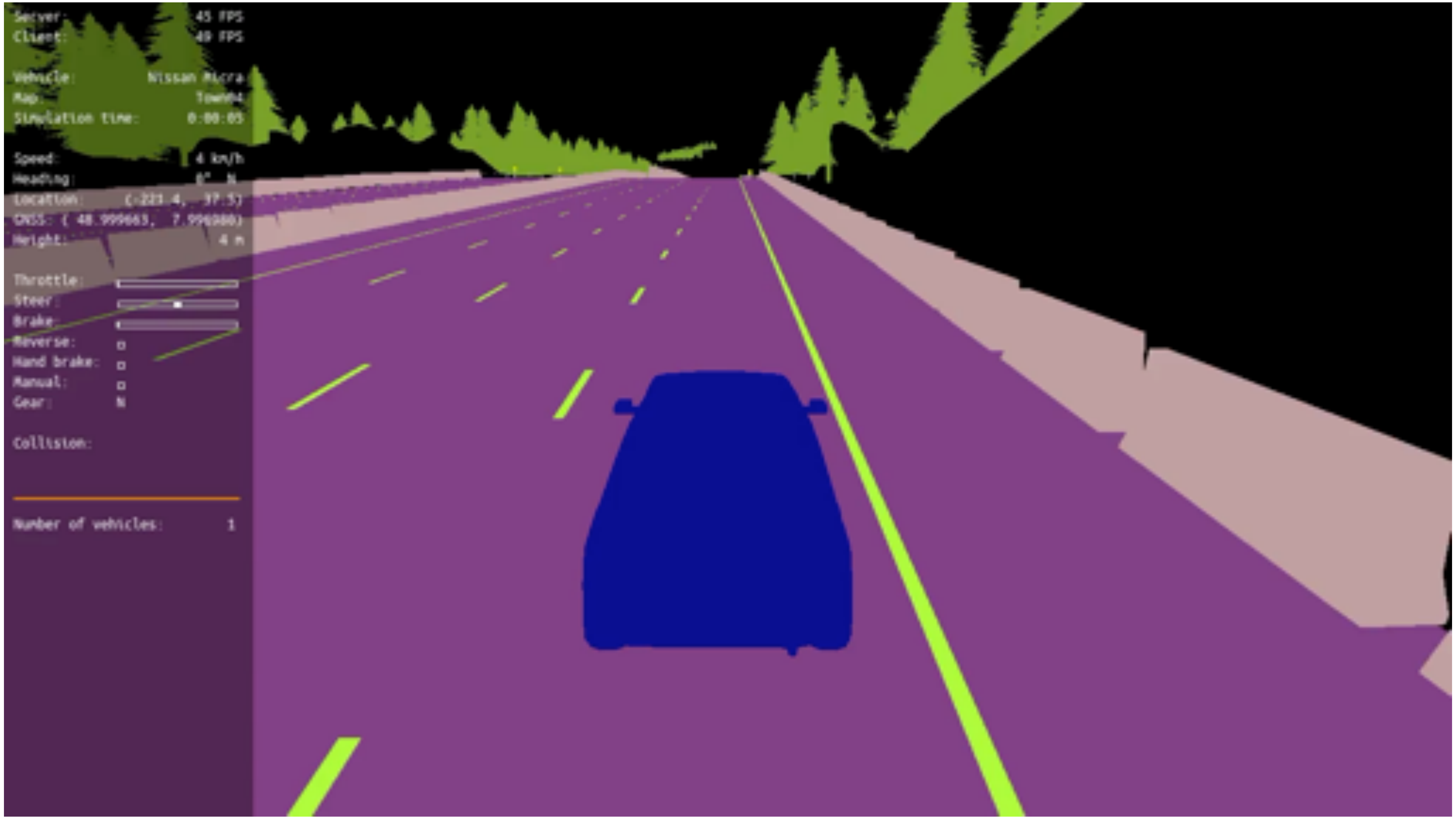
Abstract Goal Tree

```
'passenger is safe',
AND(
  ‘safe transitions’,
  NOT(‘threatening objects’)
```

!

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.

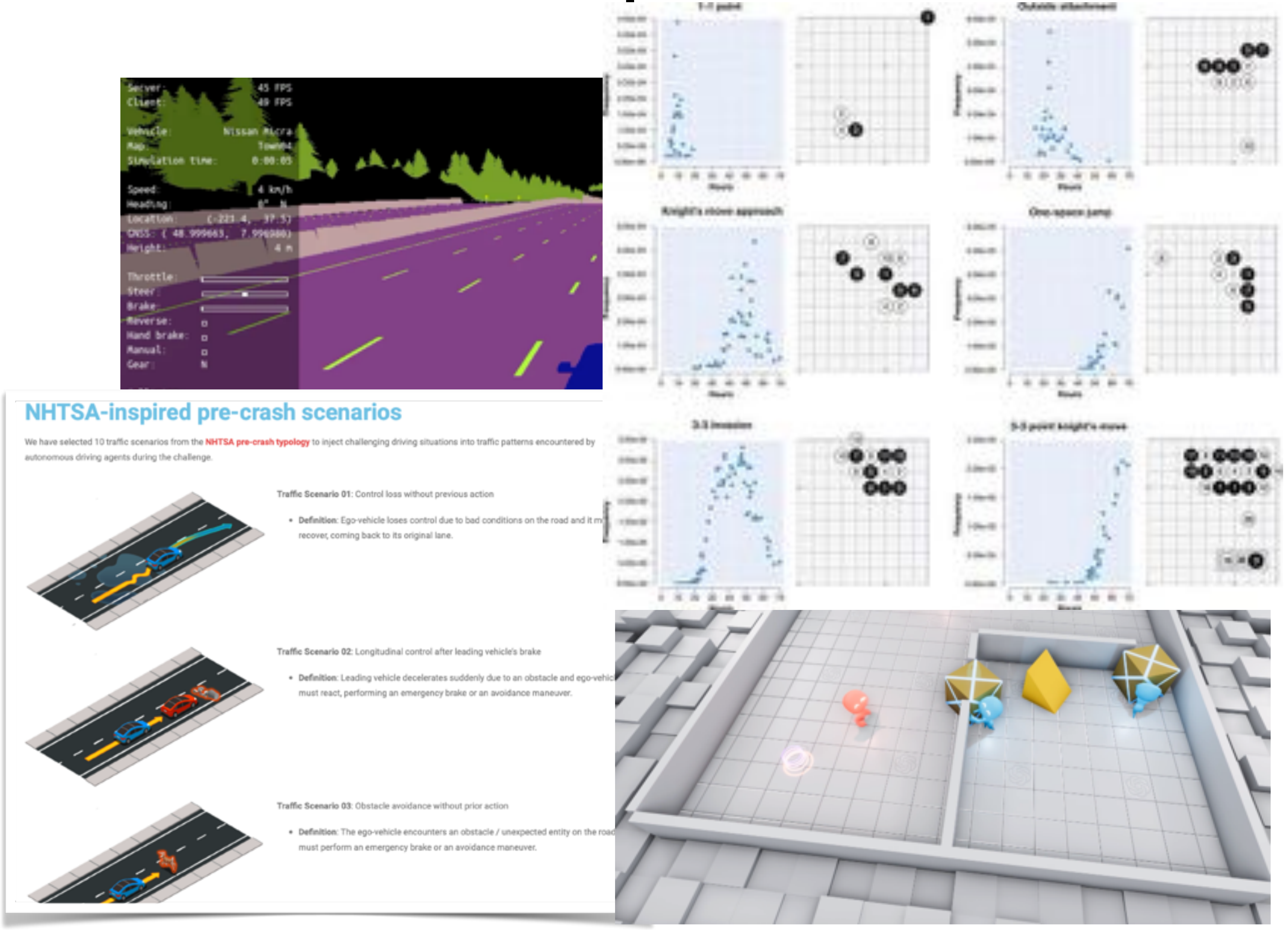
Uber Example in Simulation



L. H. Gilpin, V. Penubarthi and L. Kagal, "Explaining Multimodal Errors in Autonomous Vehicles," *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 2021, pp. 1-10, doi: 10.1109/DSAA53316.2021.9564178.

Evaluation of Error Detection is Difficult

Real-world Inspired Scenarios

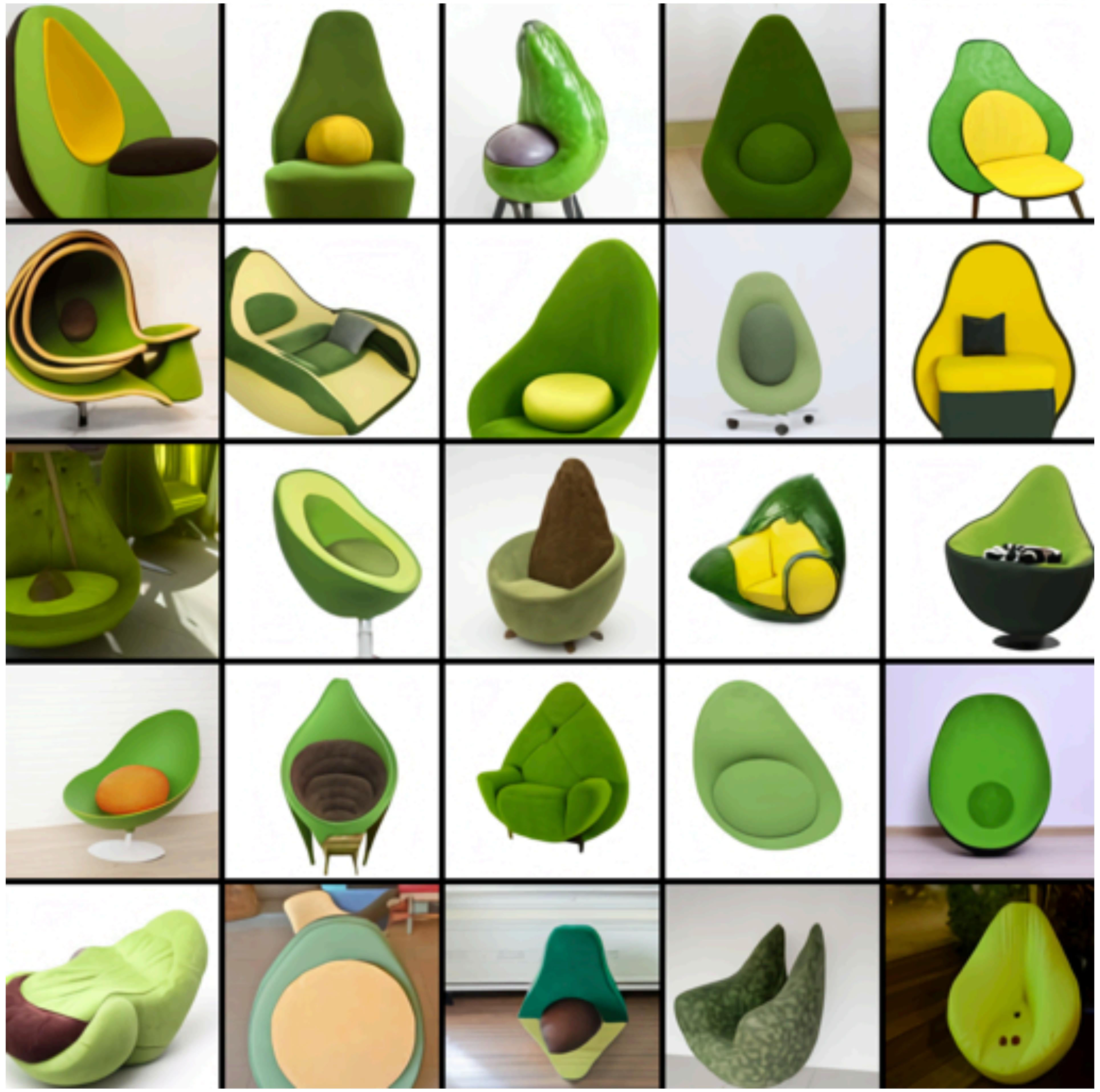


Reconcile Inconsistencies

- Detection: Generate logs from scenarios to detect failures.
- Insert errors: Scrambling *multiple* labels on existing datasets.
- Real errors: Examining errors on the validation dataset of NuScenes leaderboard.

Approach: Content Generation

Anticipatory Thinking Layer for Error Detection



DALL-E Generates “A chair in the shape of an avocado”

Synthetic images produced by StyleGAN, a GAN created by Nvidia researchers.

Approach: Content Generation

Anticipatory Thinking Layer for Error Detection

+

Generate images with shadows before tunnels.

Generate images with fallen signs.

Generate images with trucks carrying traffic lights.

Agenda

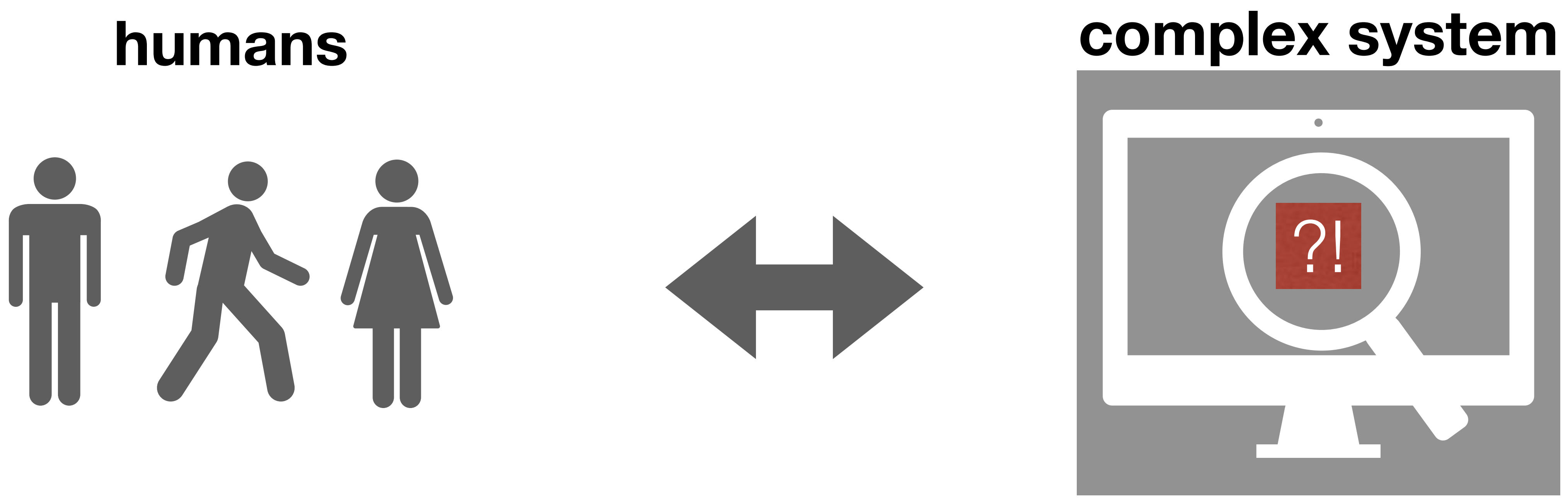
Motivate problem: Autonomous Vehicles are Prone to Failure

Anomaly Detection through Explanations (ADE): a Diagnosis Tool for AVs.

Future work: Explainable Tasks for Robust and Secure Hybrid Systems.

Hybrid Systems with Humans and Machines

Working Together on Shared Tasks

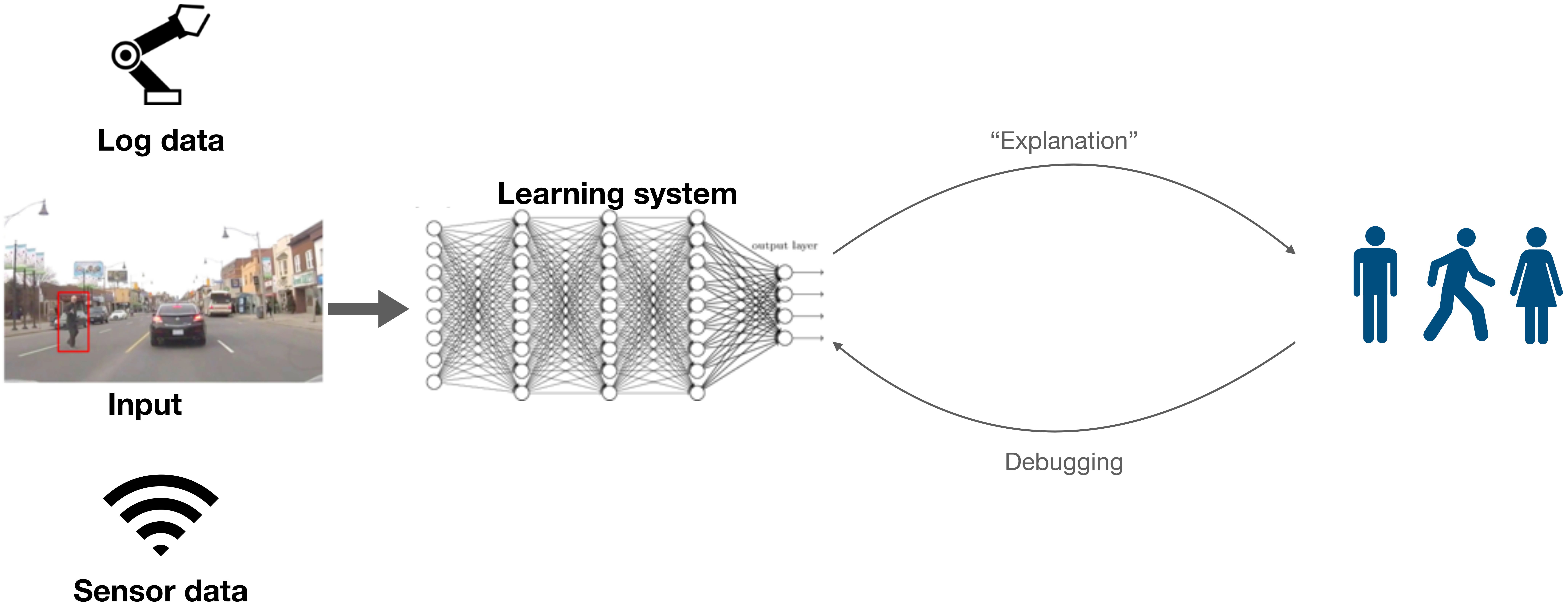


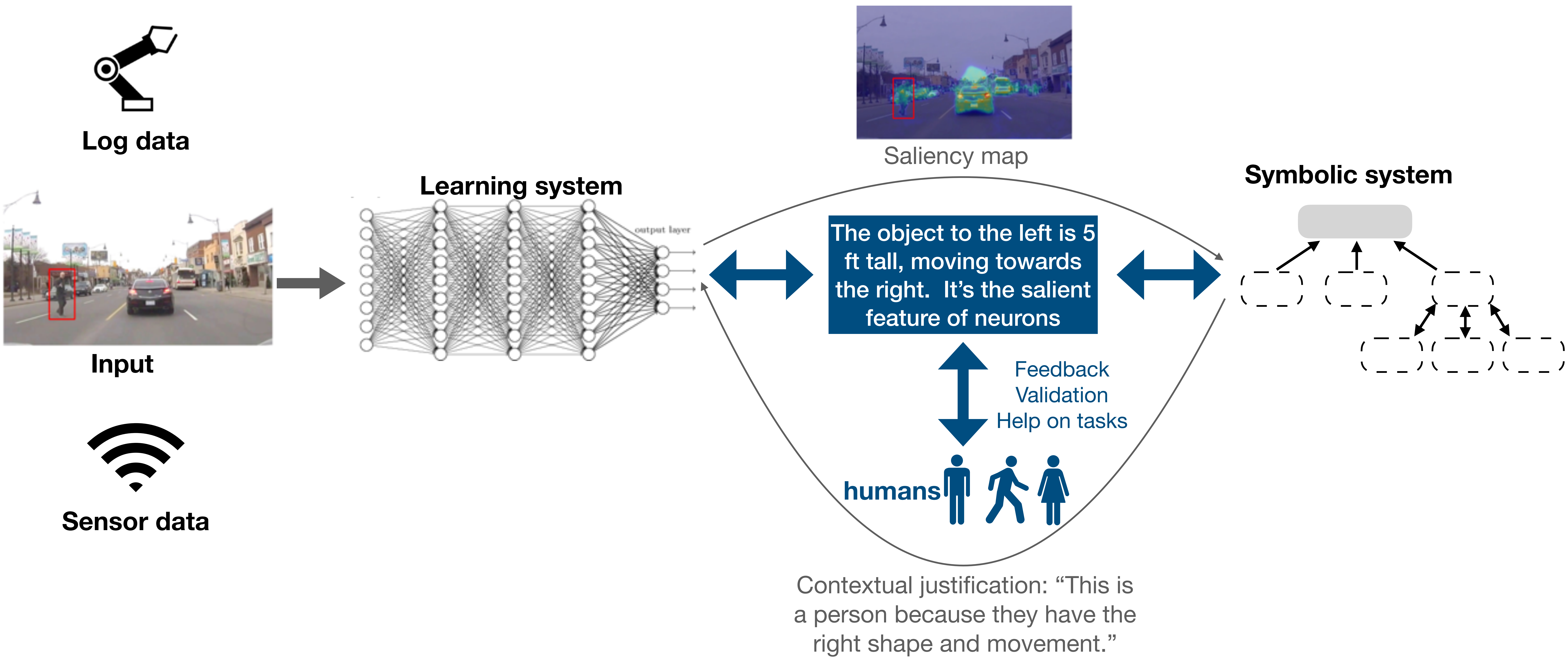
Explanations are a debugging language.

- Debugging: humans can improve complex systems.

- Education: complex systems can "improve" or teach humans.

Ex post facto explanations



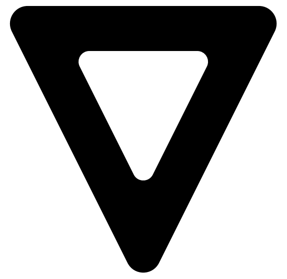


Dev testing　　Game adversaries　　Security

Explaining Errors in Complex Systems

Opaque Systems

Autonomous Systems



Error Detection

Explanations and Reasons that Society can **Trust**

- Systems that can **testify**, answer questions, and **provide insights.**
- Systems that use **commonsense**, similar to the ways that humans do.

A Common Language for Debugging and **Diagnosis**

- Interactive tools using explanations as a common **debugging language**.
- Systems that **articulately communicate with humans** on shared tasks.

Articulate Mechanisms that are **Robust**

- Hybrid, symbolic, learning systems that solve problems in **multiple ways.**
- **Dynamic explanations**, under uncertainty for safety or mission-critical tasks.

lgilpin@ucsc.edu
Leilani H. Gilpin

http://lgilpin.com
UC Santa Cruz