Anomaly Detection Through Explanations

Leilani H. Gilpin

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
June 11, 2020

Agenda

Motivate problem: Systems are imperfect

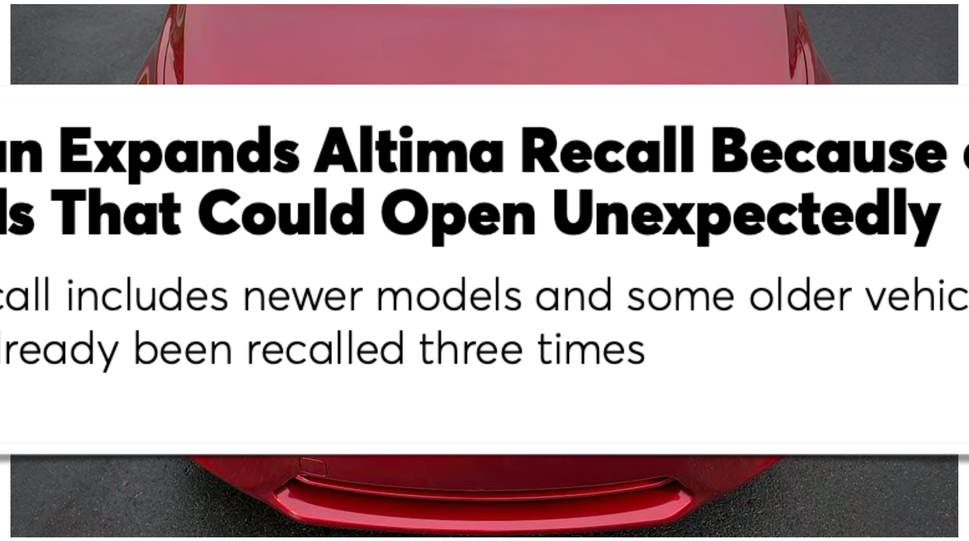
Local sanity checks

System-architecture for failure detection.

Vision: Articulate systems by design.

Question: How to develop self-explaining architectures that more adaptable, more robust, and interpretable?

Complex Systems Fail in Complex Ways



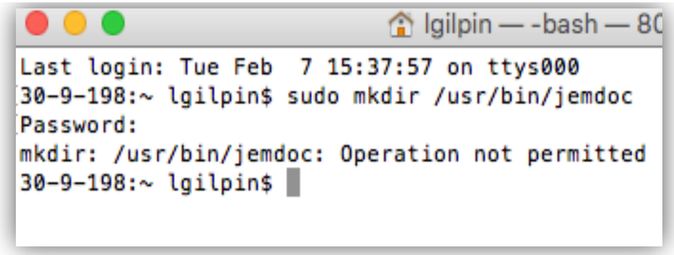
Nissan Expands Altima Recall Because of Hoods That Could Open Unexpectedly

The recall includes newer models and some older vehicles that have already been recalled three times

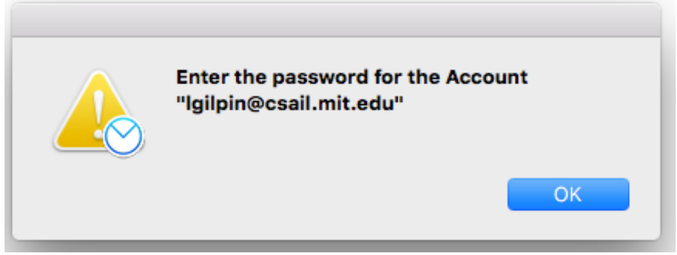
By Keith Barry
June 04, 2020

AI Mistakes Bus-Side Ad for Famous CEO, Charges Her With Jaywalking

By Tang Ziyi / Nov 22, 2018 04:17 PM / Society & Culture



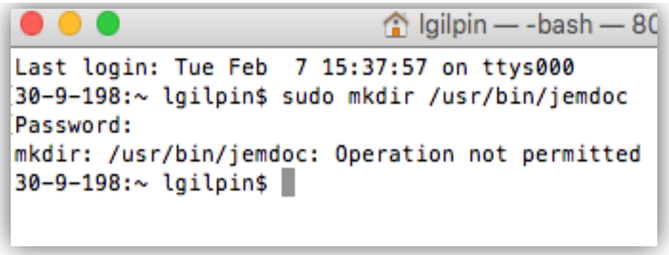
OS Upgrade (Version Skew)



Imprecise (Certificate Missing)

Existing Software Solutions are Rigid

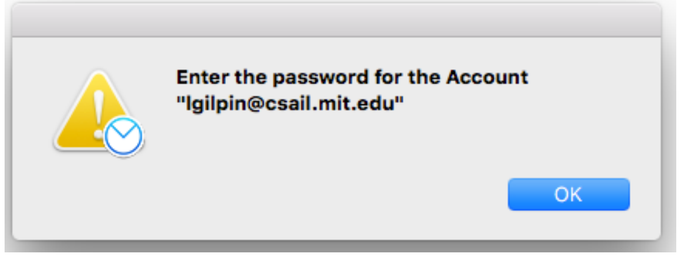
Verification, Unit Testing, Diagnostics



OS Upgrade (Version Skew)

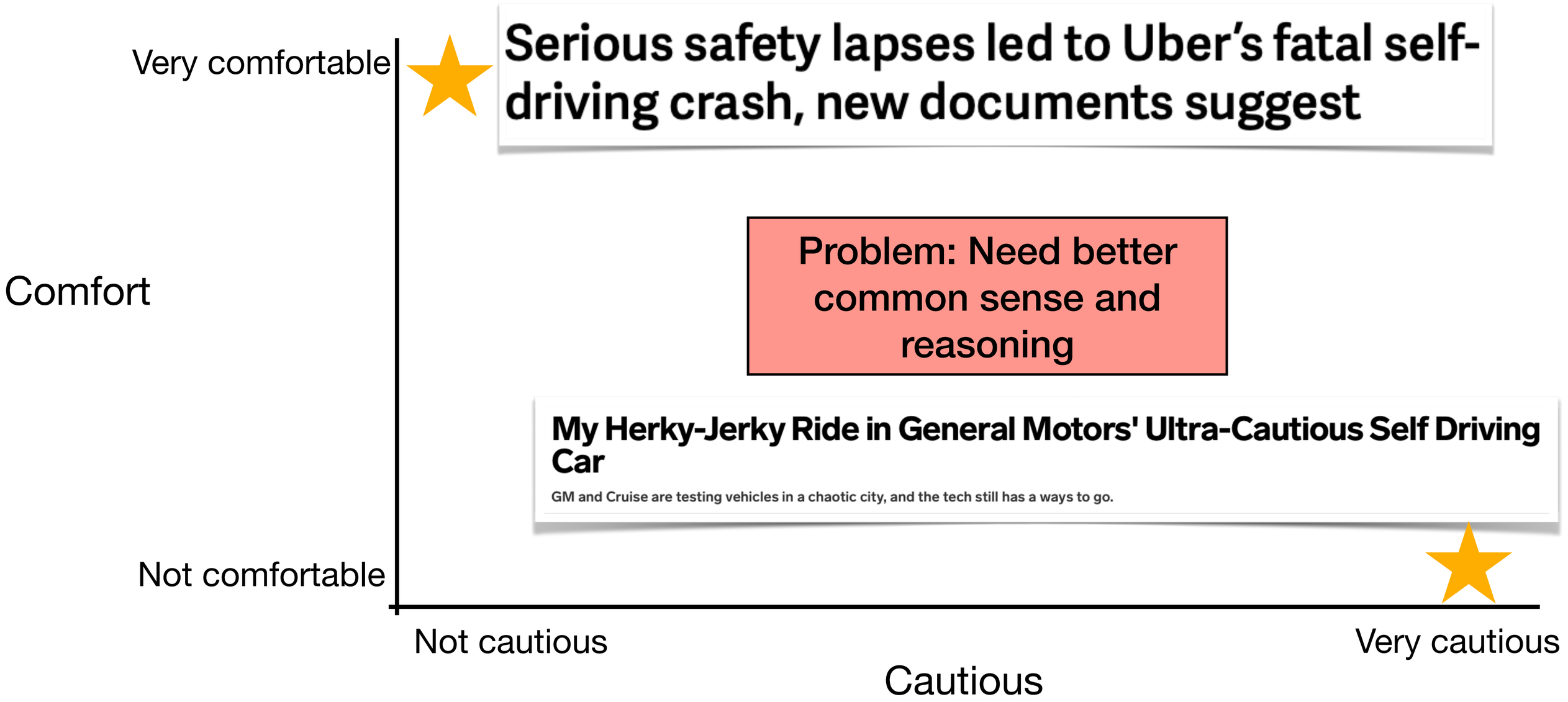
Result: Strong guarantees and provable properties

Problem: Impossible to test all failure modes in open environments



Imprecise (Certificate Missing)

Autonomous Vehicle Solutions are at Two Extremes

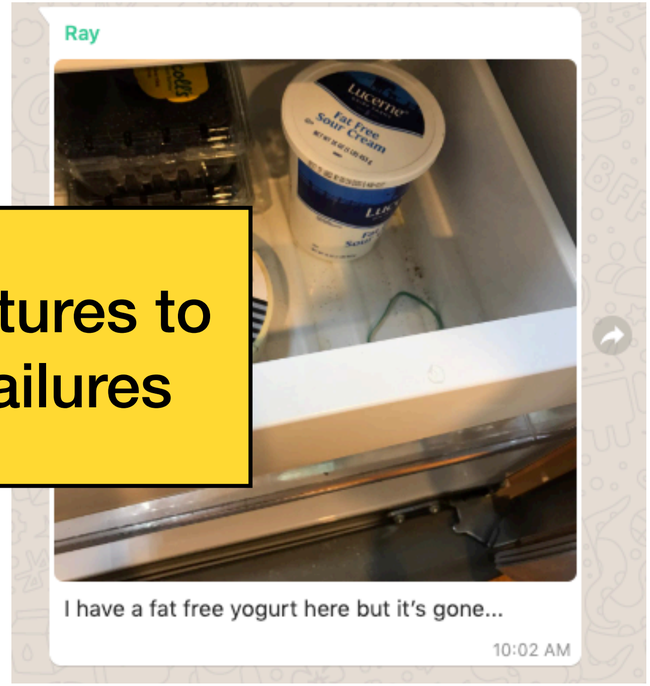


Complex Systems Include People

Misalignment of Expectations

Lack of communication

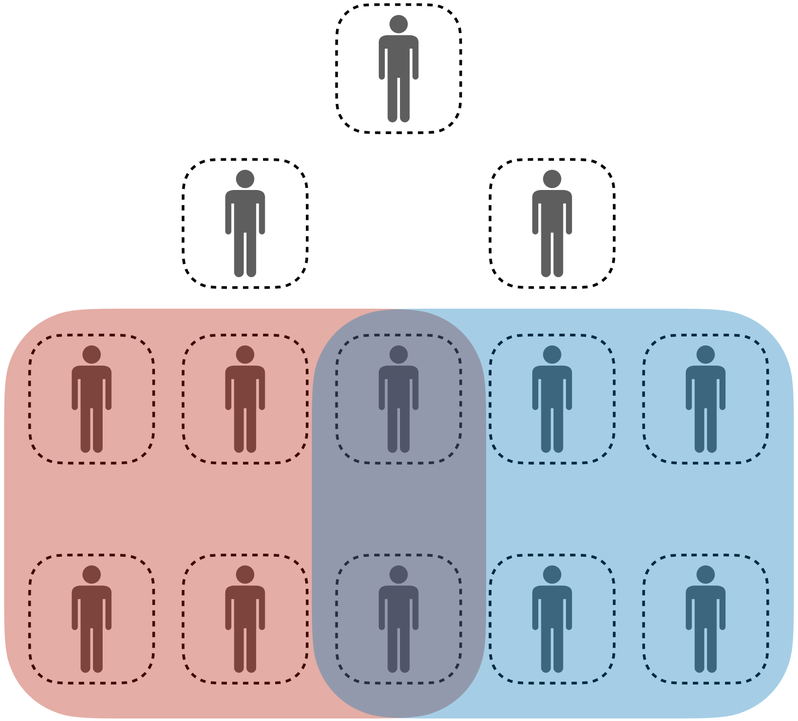
Solution: Built-in structures to deal with flaws and failures



Expectation

Architecture Inspired by Human Organizations

Communication and Sanity Checks



Local Sanity Checks

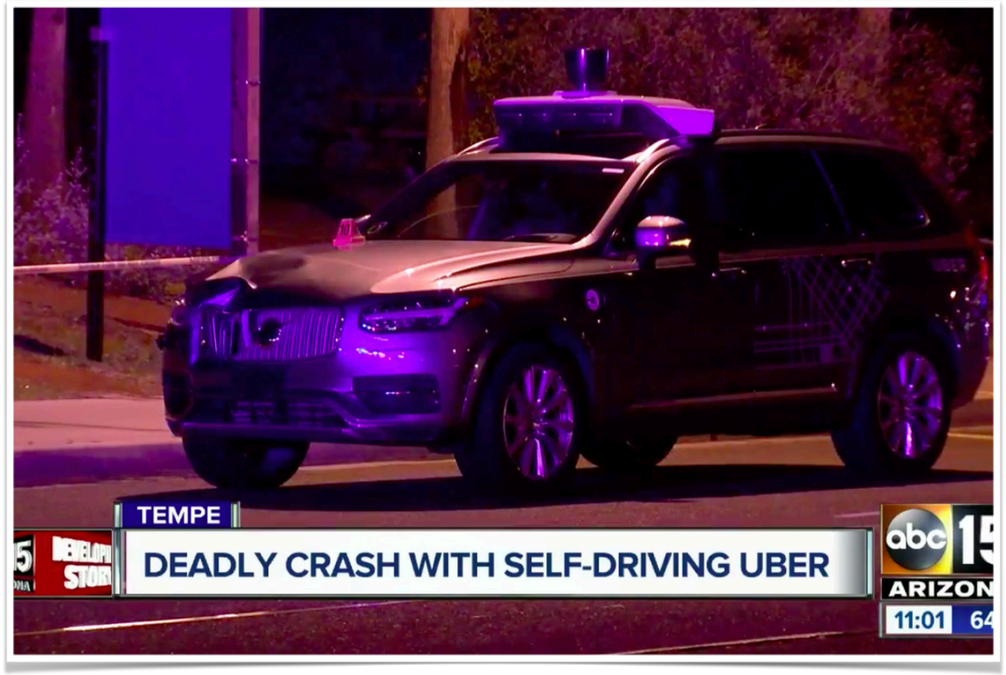
Synthesizer to reconcile inconsistencies between parts.

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



future tense

The Trollable Self-Driving Car

Reconcile conflicting reasons.

Justify new examples.

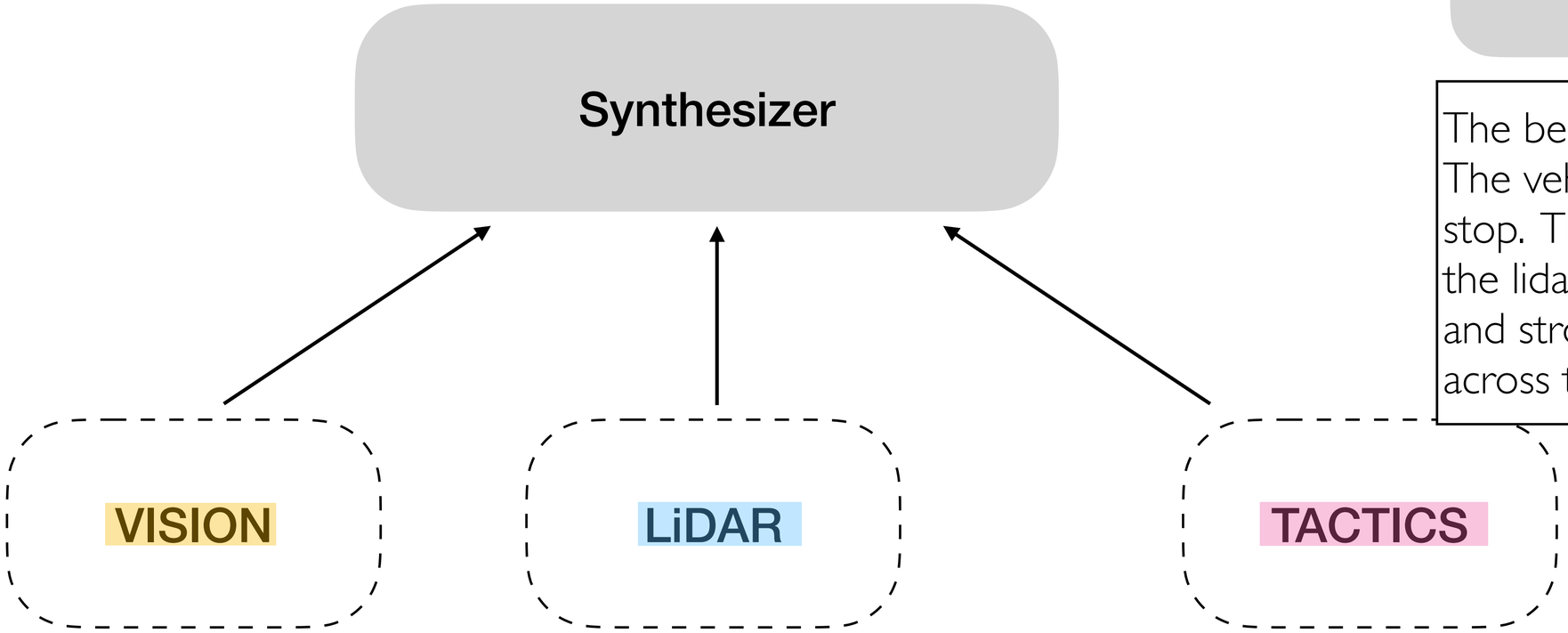
An Existing Problem

The Uber Accident

Solution: Internal Communication

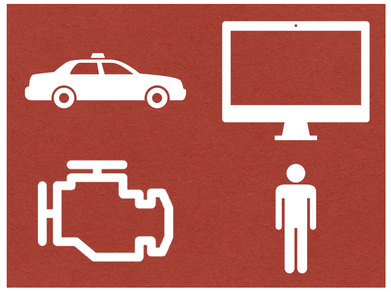
Anomaly Detection through Explanations

Synthesizer to reconcile inconsistencies between monitor outputs.

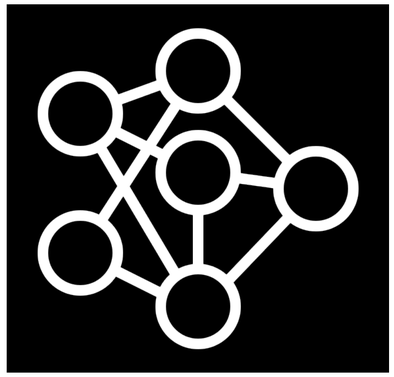


The best option is to veer and slow down. The vehicle is traveling too fast to suddenly stop. The vision system is inconsistent, but the lidar system has provided a reasonable and strong claim to avoid the object moving across the street.

Defense Outline



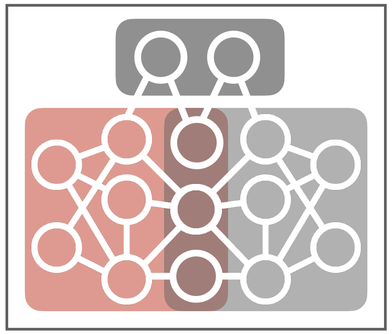
Problem: Complex systems are imperfect.



<u>Error detection for local subsystems.</u>

Opaque subsystems.

Sensor subsystem interpretation.



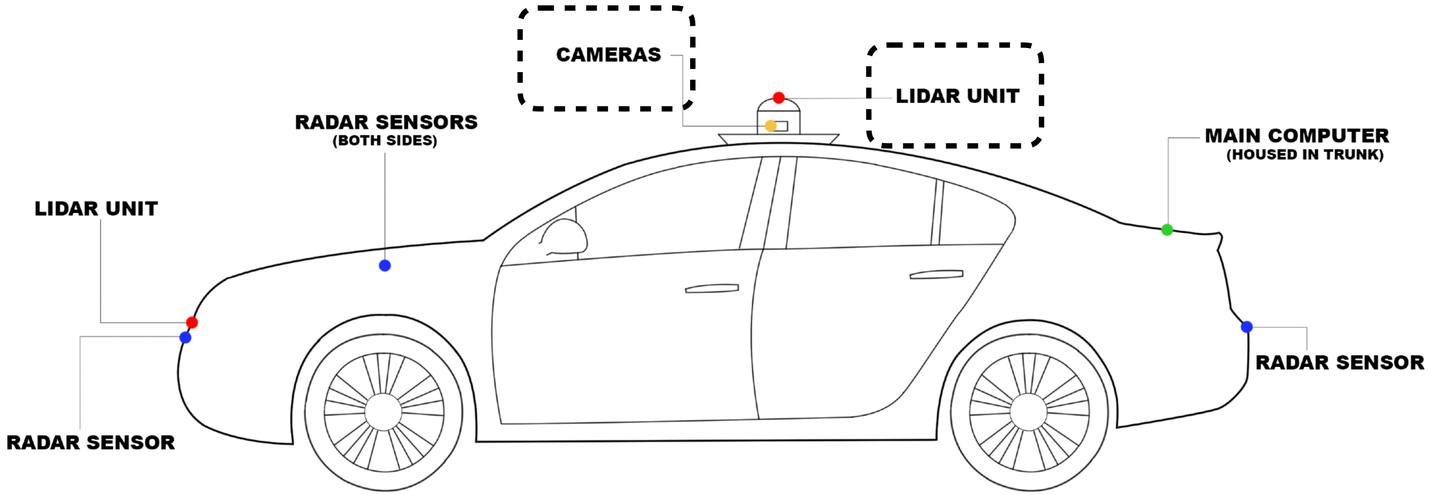
System-wide failure detection.

Vision: Articulate systems by design.

Complex Systems Fail in Two Ways

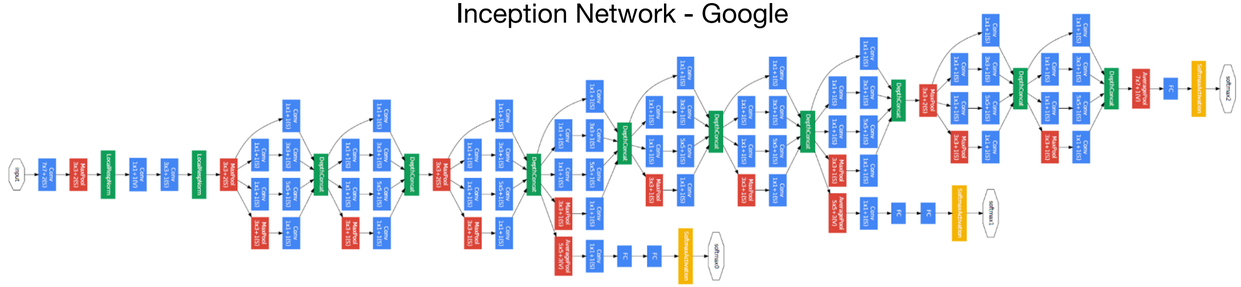
1. Failure *local* to a specific subsystem.

2. A failed *cooperation* amongst subsystems.

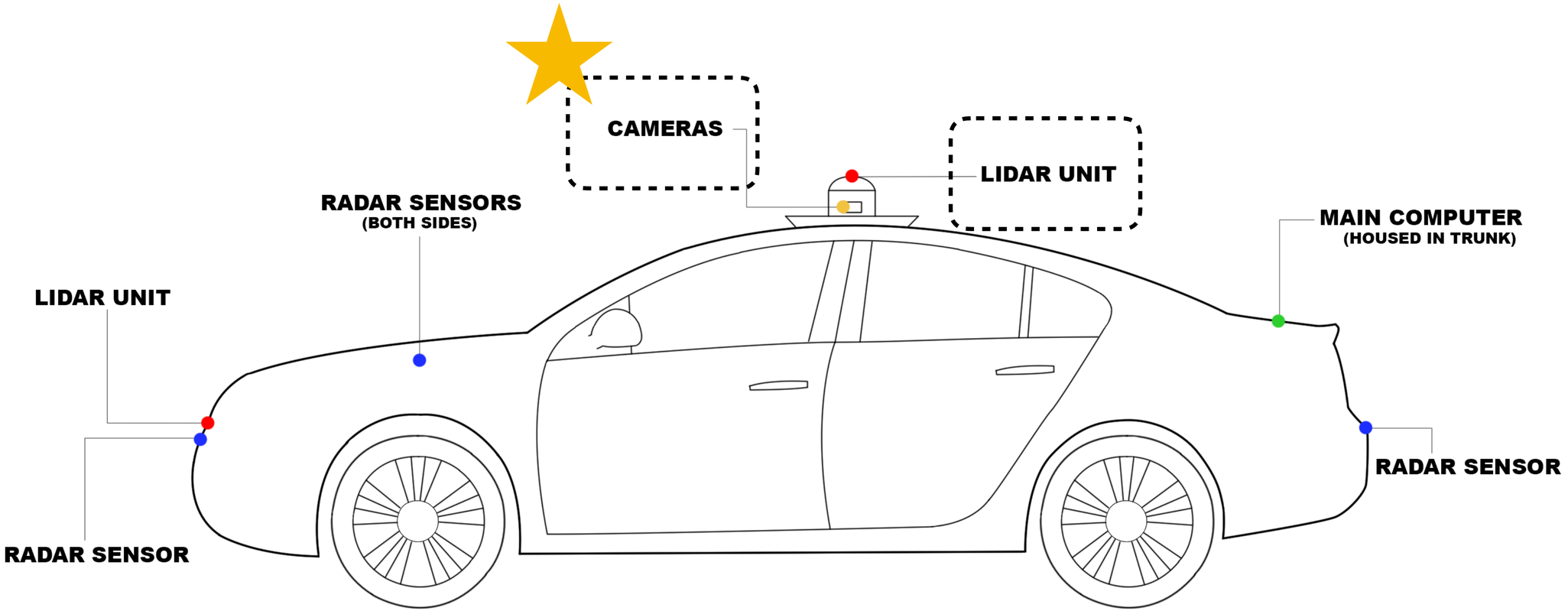


A Neural Network Labels Camera Data

Inception Network - Google

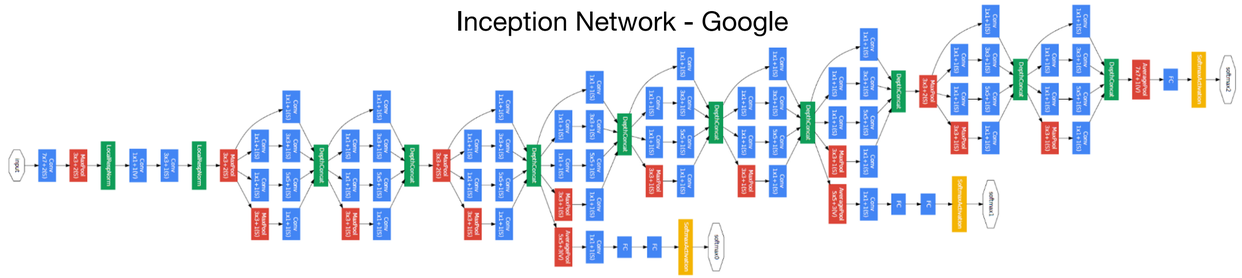


Label
e.g. pedestrian



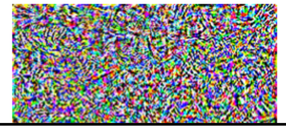
Problem: Neural Networks are Brittle

Inception Network - Google



Label
e.g. pedestrian

“pig”



“airliner”

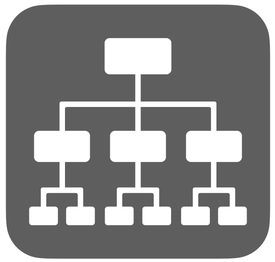
For self-driving, and other mission-critical, safety-critical applications, these mistakes have CONSEQUENCES.

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

Monitor Opaque Subsystems for Reasonableness


Commonsense
Knowledge Base

+


Flexible
Representation

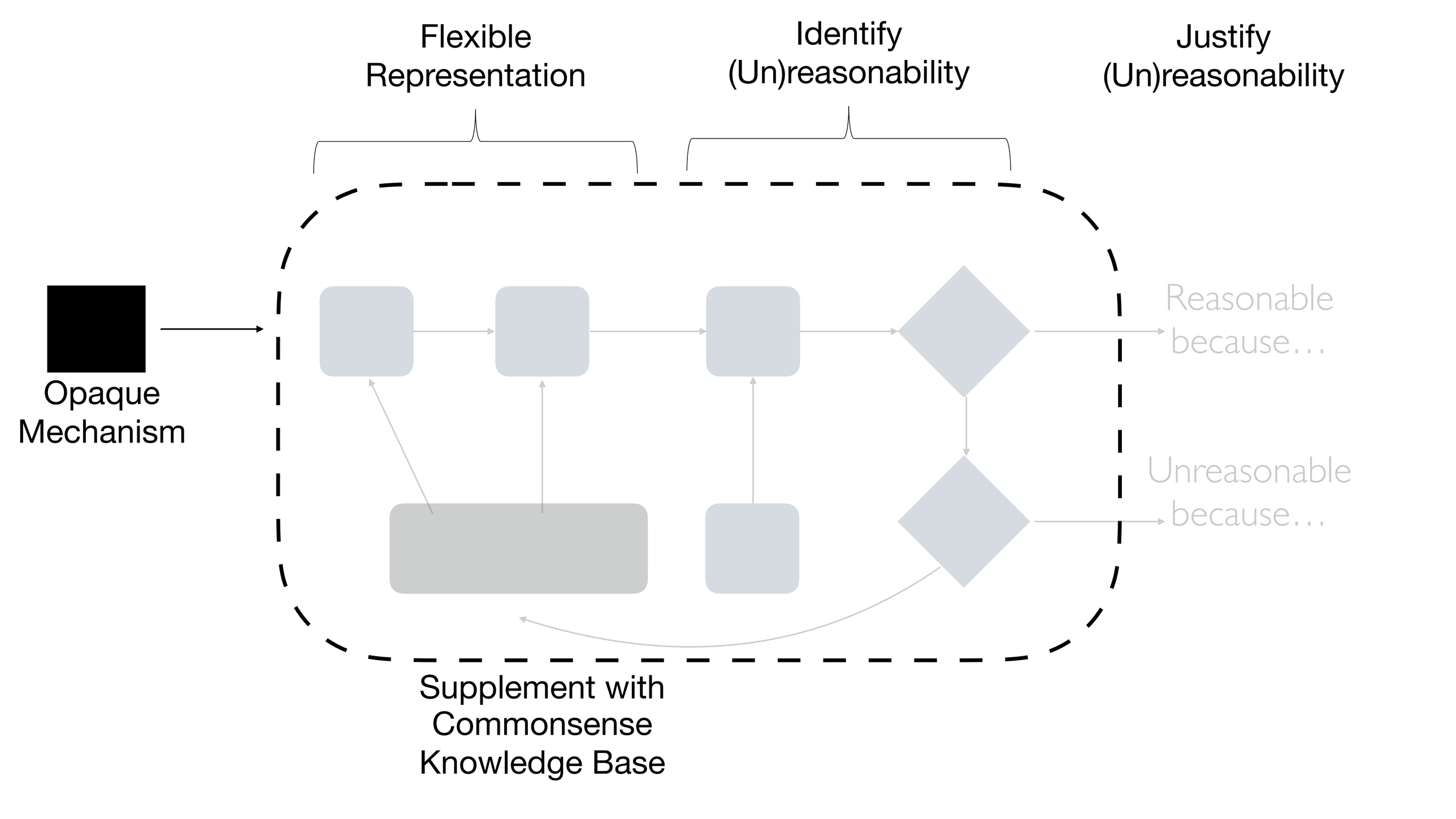
+

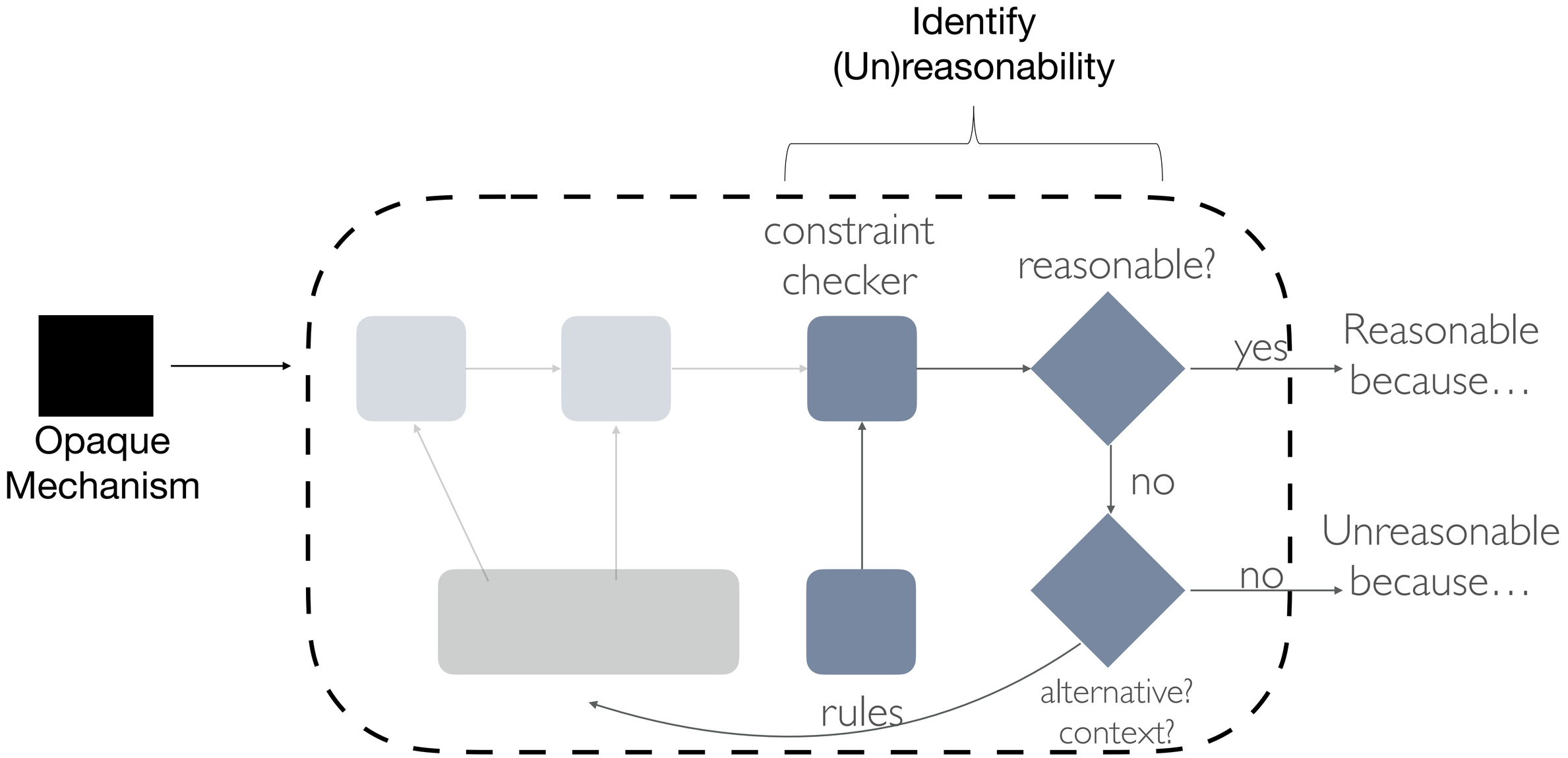
Identify
(Un)reasonability

+

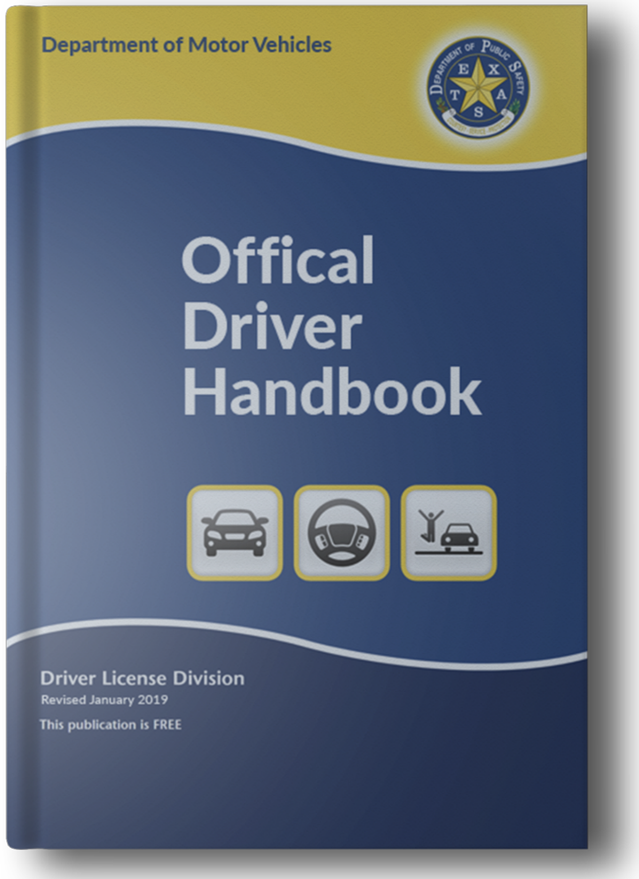

Justify
(Un)reasonability

1. Judgement of reasonableness
2. Justification of reasonableness


Flexible Representation
Identify (Un)reasonability
Justify (Un)reasonability
Opaque Mechanism
Reasonable because…
Unreasonable because…
Supplement with Commonsense Knowledge Base


Identify
(Un)reasonability
constraint
checker
reasonable?
Opaque
Mechanism
yes
Reasonable
because…
no
Unreasonable
because…
no
rules
alternative?
context?

Identify
(Un)reasonability

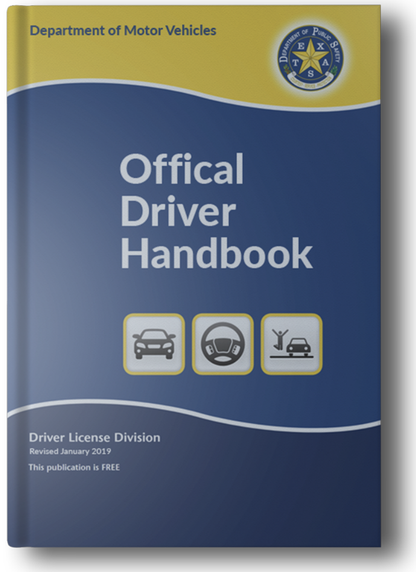


Start with Baseline Rules

1. Automatically parsed pdf text.
    1. Searched for key concepts.
    2. Generated rules.
2. I manually validated the generated rules.

Identify
(Un)reasonability

Start with Baseline Rules




```
:safe_car_policy a air:Policy;
                 air:rule :light-rule;
                 air:rule :pedestrian-rule;
                 air:rile :speed-rule;
                 rdfs:comment "Safe driving tactics";
                 rdfs:label "Safe driving tactics by the state of MA."

:pedestrian-rule a air:Belif-rule;
                 rdfs:comment “Ensure that pedestrians are safe.";
                 air:if {
                        :EVENT a :V;
                               car_ont:InPathOf :V.
                 };
                 air:then [
                         air:description ("There is a pedestrian");
                         air:assert [air:statement{:Event
                                    air:compliant-with :safe_car_policy .}]] .
                 air:else [
                         air:description ("There is not a pedestrian");
                         air:assert [air:statement{:Event
                                    air:non-compliant-with :safe_car_policy .}]] .
```

+ reasoner

http://dig.csail.mit.edu/2009/AIR/

L.H. Gilpin and L. Kagal. “An Adaptable Self-Monitoring Framework for Opaque Machines.” AAMAS 2019.

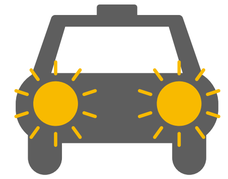
Identify
(Un)reasonability

Baseline rule

Flashing high beams

Turn on lights

New rule



Flashing high beams

Warning signal

Learn Rules

Identify
(Un)reasonability

Learn Rules

constraint
checker

reasonable?

yes

no

no

Reasonable
because…

Unreasonable
because…

Opaque

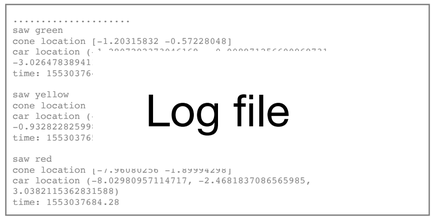
rules

alternative?
context?

reasoning

New rule

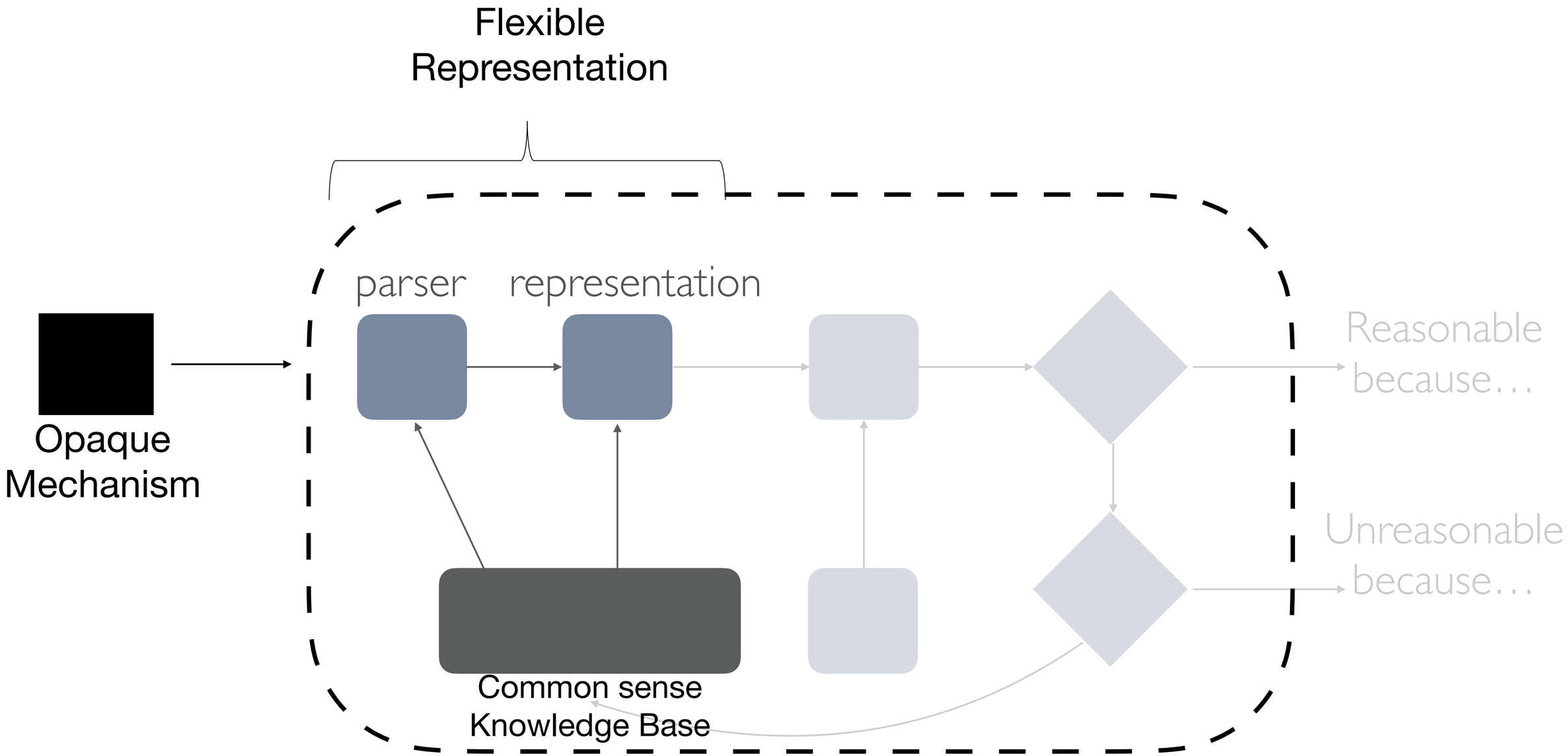
New
environment



Log file

Rule
learning

No rule


Flexible
Representation
parser
representation
Opaque
Mechanism
Reasonable
because…
Unreasonable
because…
Common sense
Knowledge Base

Primitive Representations

Encode Understanding

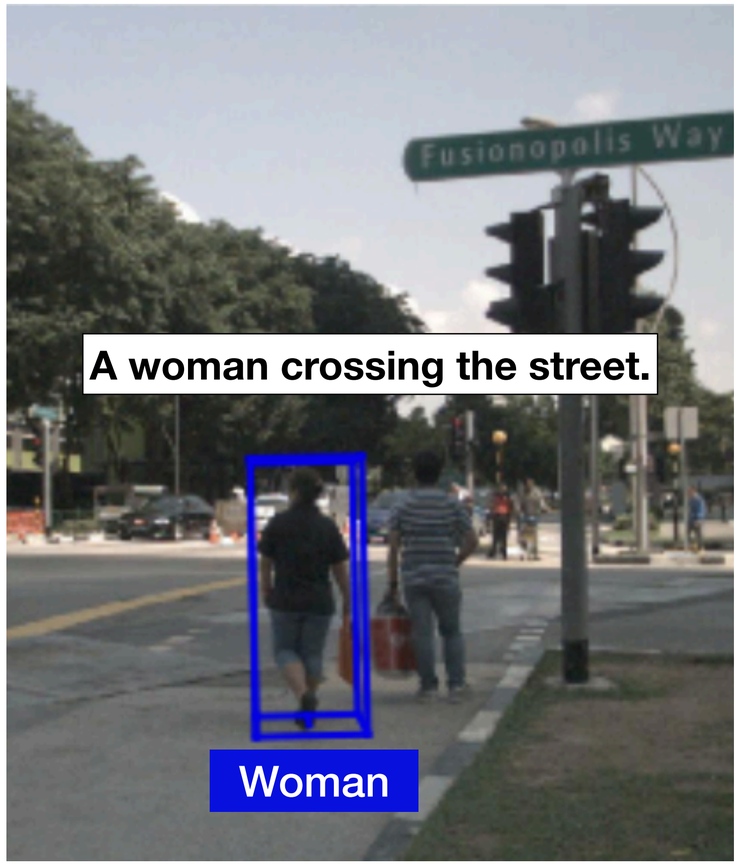
Conceptual Dependency Theory (CD), Schank 1975

11 primitives to account for *most* actions:

ATRANS
ATTEND
INGEST
EXPEL
GRASP
MBUILD
MTRANS
MOVE
PROPEL
PTRANS
SPEAK

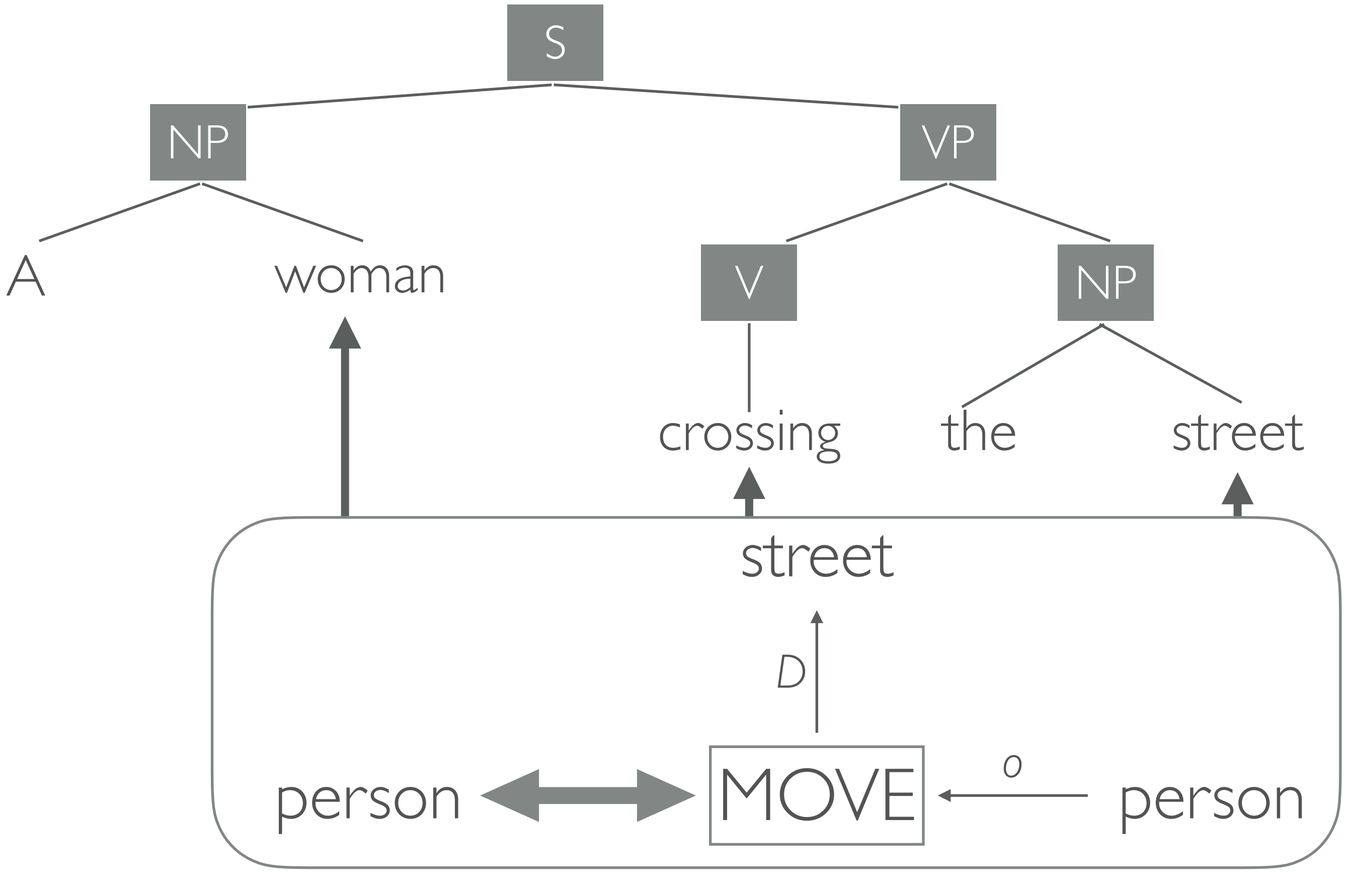
5 for physical actions
Extended to vehicle primitives

Parse Natural Language into Representation

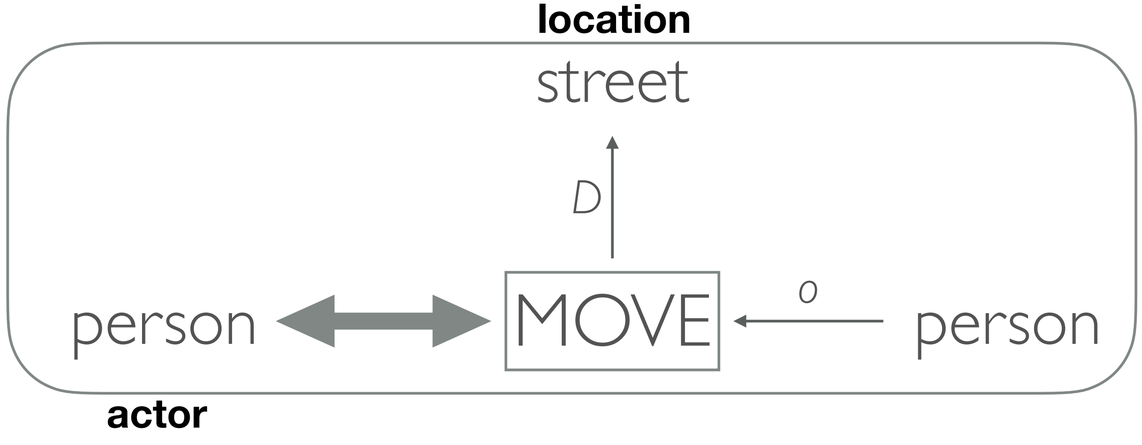


Data from Nuscenes

Parser



Representations with Implicit Rules



A perceived frame is REASONABLE

$$\big((x_1, p_1, y_1), \textbf{isA, REASONABLE}\big) \wedge$$
$$\big((x_2, p_2, y_2), \textbf{isA, REASONABLE}\big) \wedge$$
$$\ldots \wedge$$
$$\big((x_n, p_n, y_n), \textbf{isA, REASONABLE}\big)$$

Move Primitive Reasonability

$$(x, hasProperty, animate) \wedge (x, locatedNear, y) \Rightarrow \big((x, MOVE, y) \textbf{ isA, REASONABLE}\big)$$

actor **location**

Implementing Reasonableness Monitors

For Real-world Error Detection

- End-to-end prototype
  - Machine perception
  - Represented with Schank conceptual dependency primitives.

L.H. Gilpin, J.C. Macbeth and E. Florentine. “Monitoring scene understanders with conceptual primitive decomposition and commonsense knowledge.” ACS 2018.

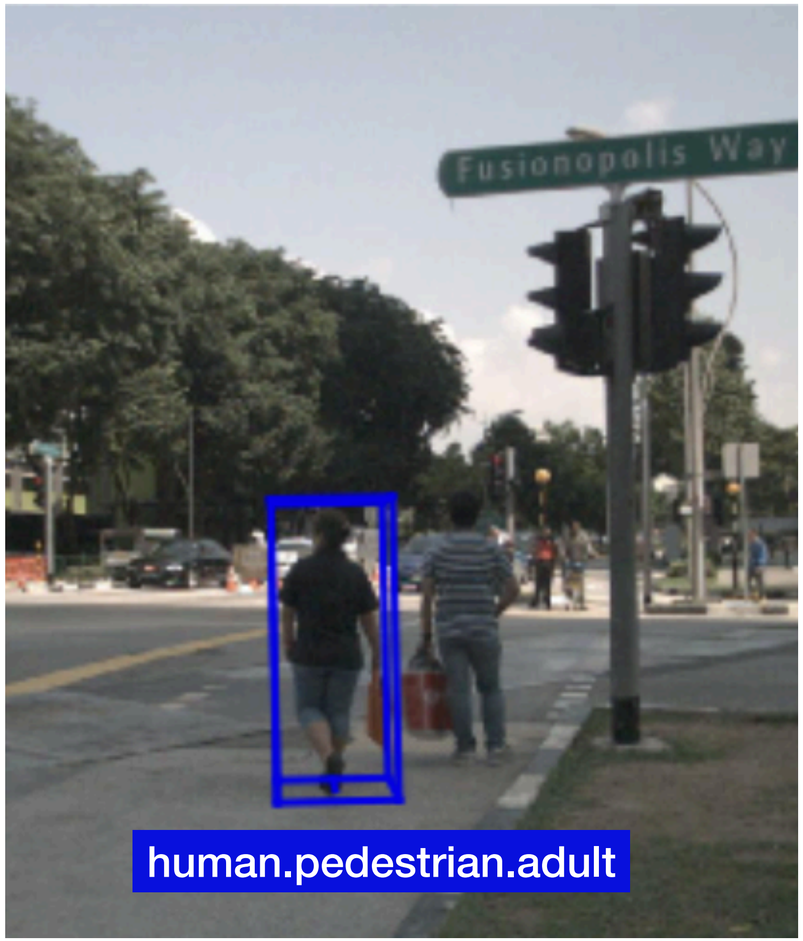
- Generalized framework
  - Reusable web standards
  - Extended Schank representations

L.H. Gilpin and L. Kagal. “An Adaptable Self-Monitoring Framework for Opaque Machines.” AAMAS 2019.

Reasonableness Monitoring on Real Data

NuScenes


```
{'token': '70aecbe9b64f4722ab3c230391a3beb8',
 'sample_token': 'cd21dbfc3bd749c7b10a5c42562e0c42',
 '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': 'a1721876c0944cdd92ebc3c75d55d693',
 'next': '1e8e35d365a441a18dd5503a0ee1c208',
 'num_lidar_pts': 5,
 'num_radar_pts': 0,
 'category_name': 'human.pedestrian.adult'}
```

Data from NuScenes

Commonsense is Unorganized

ConceptNet

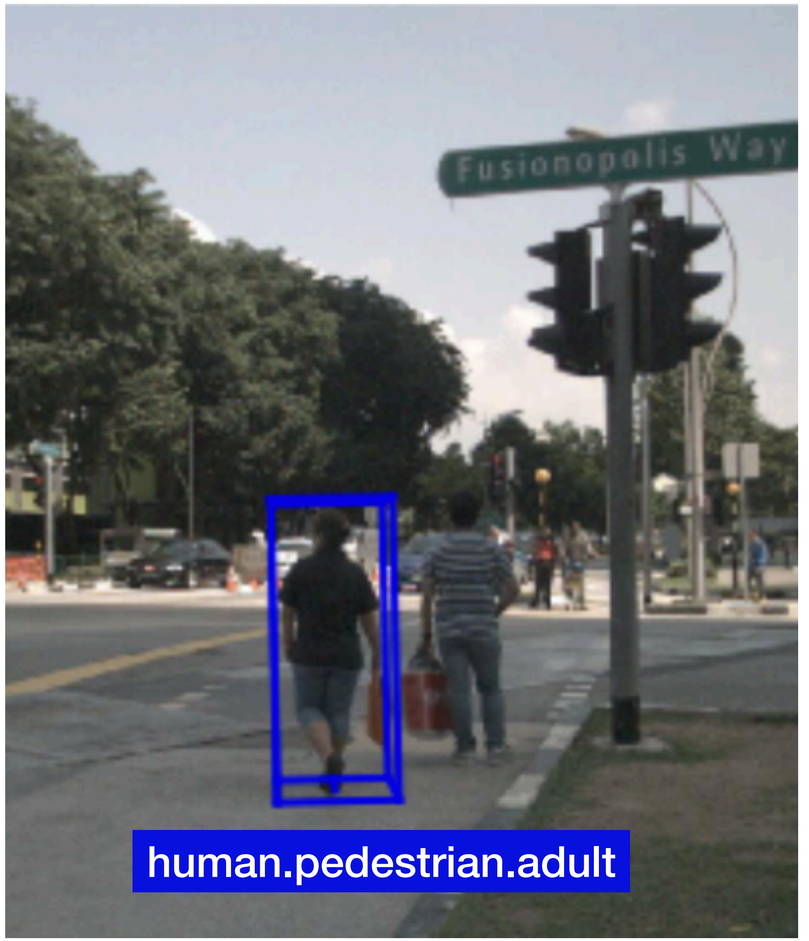
adult is a type of...

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

adult is capable of...

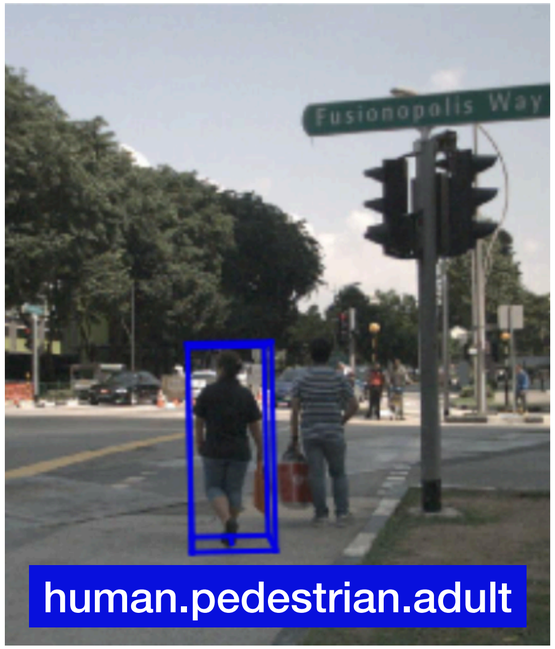
- en help a child
- en dress herself
- en sign a contract
- en drink beer
- en work
- en act like a child
- en dress himself
- en drive a car
- en drive a train
- en explain the rules to a child

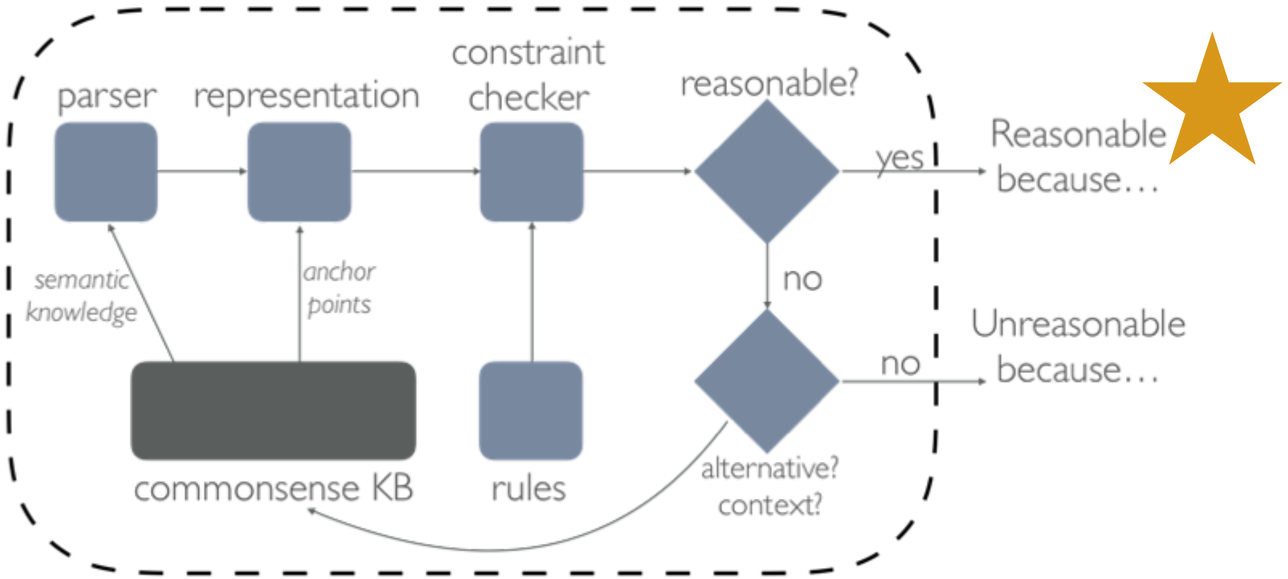
```
(‘adult, ‘typeOf, ‘animal)
(‘adult, ‘isA, ‘bigger than a child’)
…
```



Data from NuScenes

Monitor Outputs a Judgement and Justification





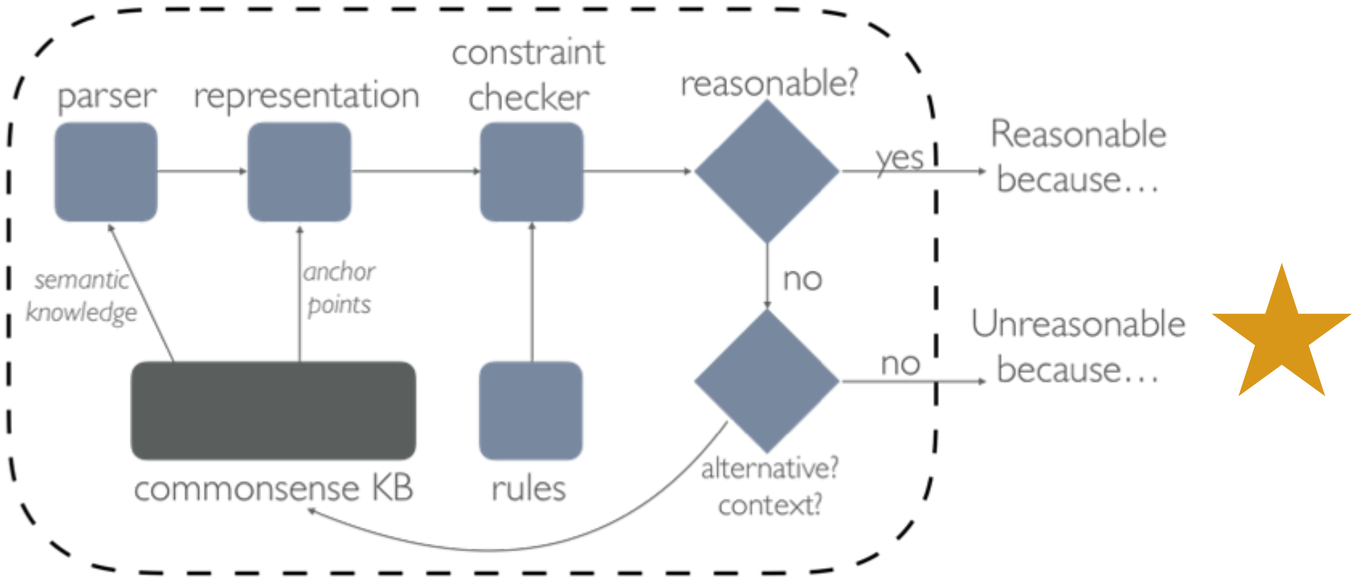
This perception is reasonable. An adult is typically a large person. They are usually located walking on the street. Its approximate dimensions of [0.621, 0.669, 1.642] is approximately the correct size in meters.

Evaluating Reasonableness Monitors

Building Errors

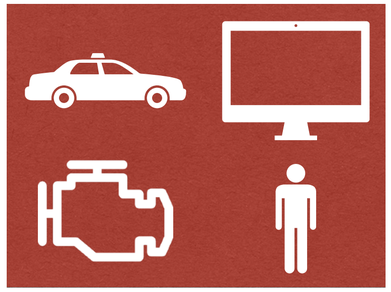
- Built an “unreasonable” image description dataset.
  - 100 descriptions.
  - Average of 4.47 words, with 57 unique words.
  - 14 verbs, 35 nouns, 8 articles/auxiliary verbs, prepositions.
  - 23 of the 100 had prepositional phrases.
- Self-driving image processing errors:
  - Real-time evaluation with Carla.
  - Added errors on existing datasets (NuScenes).
  - Examining errors on the validation dataset of NuScenes leaderboard.
  - Building challenge problems and scenarios.

Adding and Validating Errors

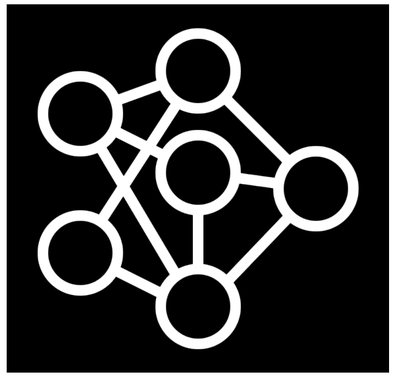


This perception is unreasonable. The movable_object.trafficcone located in the center region is not a reasonable size: it is too tall. There is no common sense supporting this judgement. Discounting objects detected in the same region.

Defense Outline



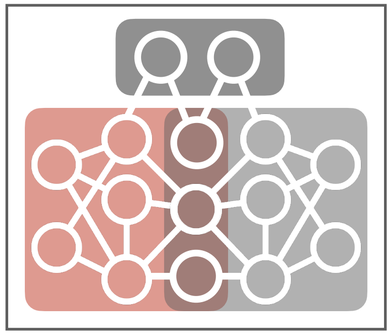
Problem: Complex systems are imperfect.



Error detection for local subsystems.

Opaque subsystems.

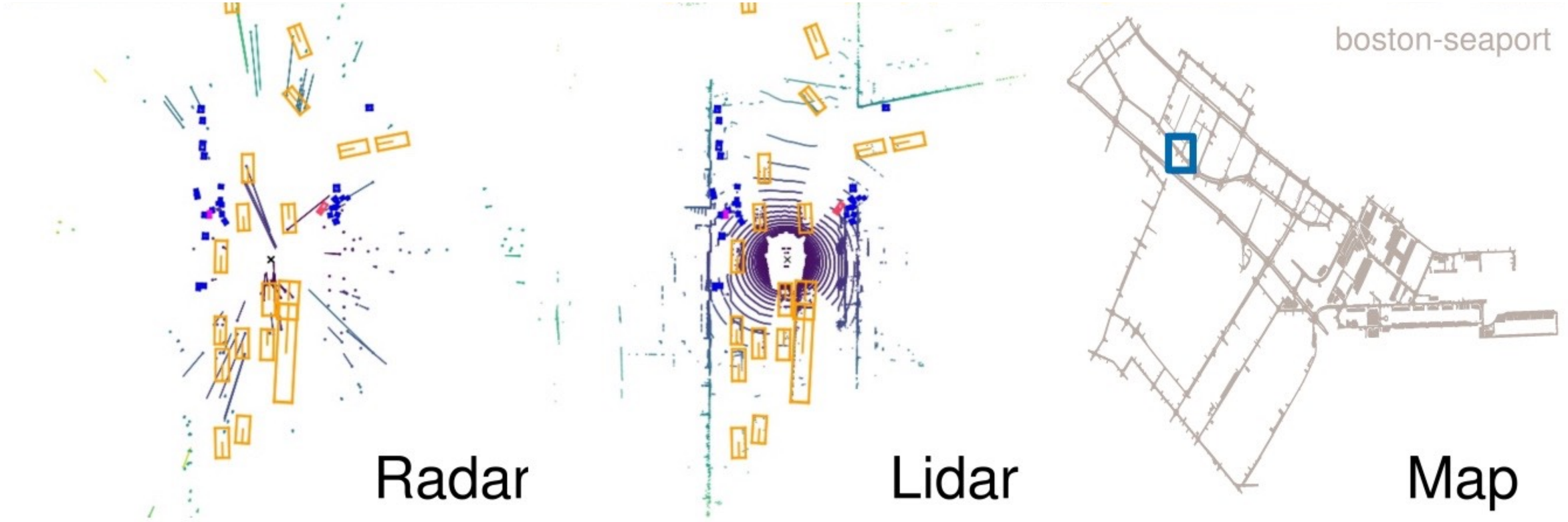
Sensor subsystem interpretation.



System-wide failure detection.

Vision: Articulate systems by design.

Sensor Data is Difficult to Understand

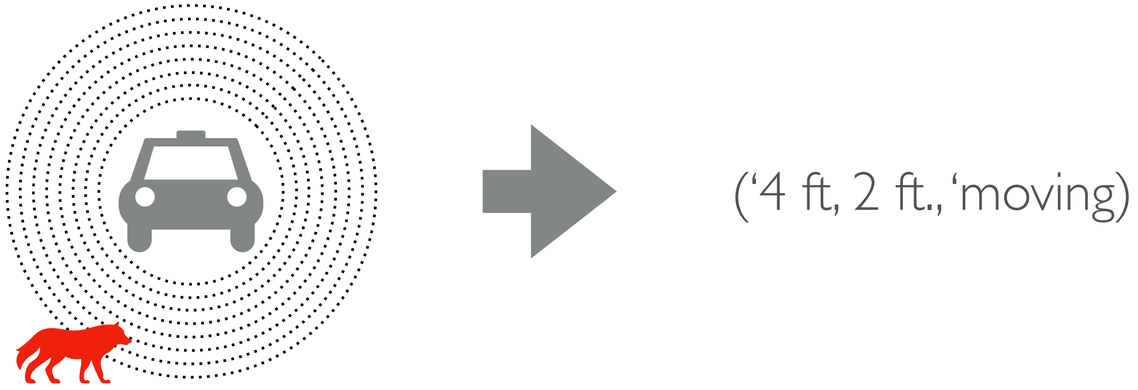


Labeled output: “Pedestrian with a pet, bicycle, car making a u-turn, lane changes, pedestrian crossing in a crosswalk.”

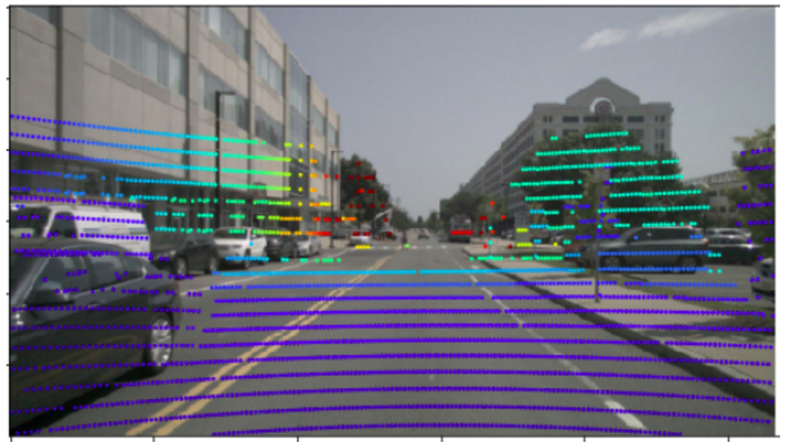
Solution: Sensor Data Interpreter

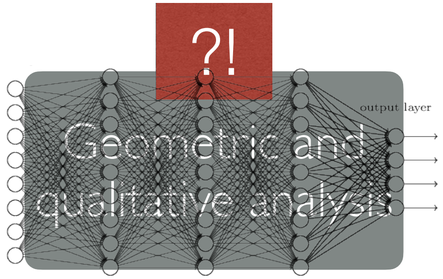
Qualitatively Describe Point Clouds

- Interprets low-level sensor data in qualitative descriptions.
  - Edge detection.
  - Geometric analysis for tracking.
- Qualitative description can be input into a reasonableness monitor for additional reasoning and justifications.

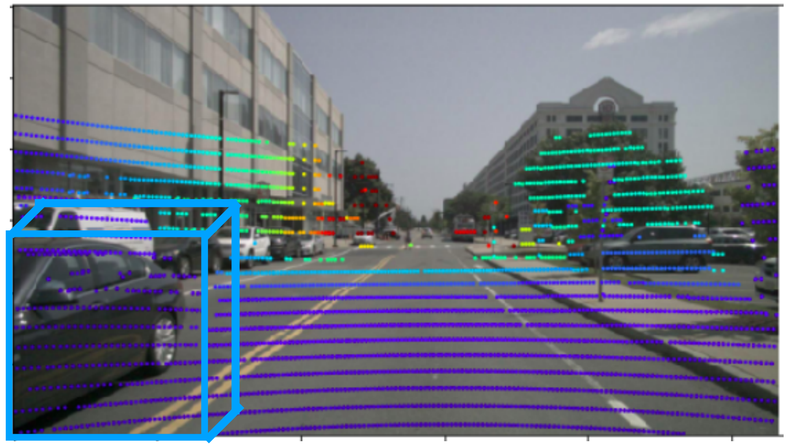


Solution: Process LiDAR Similar to Images



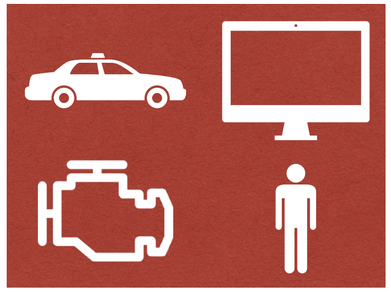


Bounding box dimensions+location

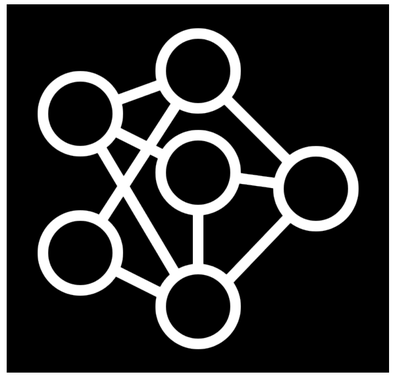


(5 ft wide, 4 ft tall, 8 ft deep, moving, slowly, stable movement, back right, …)

Defense Outline



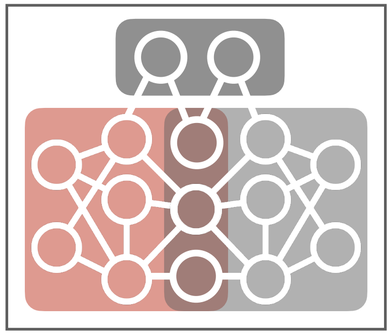
Problem: Complex systems are imperfect.



Error detection for local subsystems.

Opaque subsystems.

Sensor subsystem interpretation.



System-wide failure detection.

Vision: Articulate systems by design.

A Deadly Crash

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

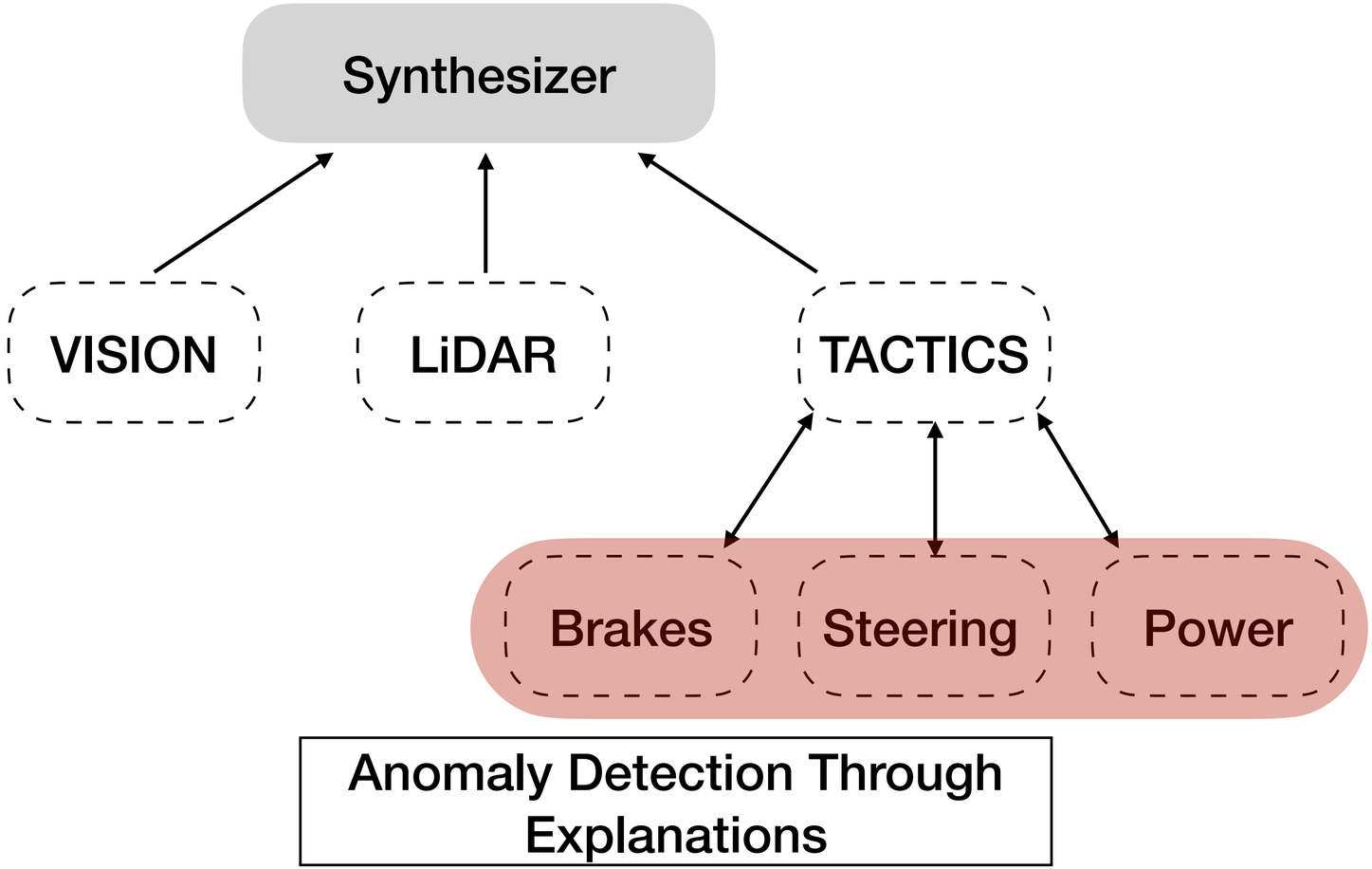
My Herky-Jerky Ride in General Motors' Ultra-Cautious Self Driving Car

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

Reconciling Internal Disagreements

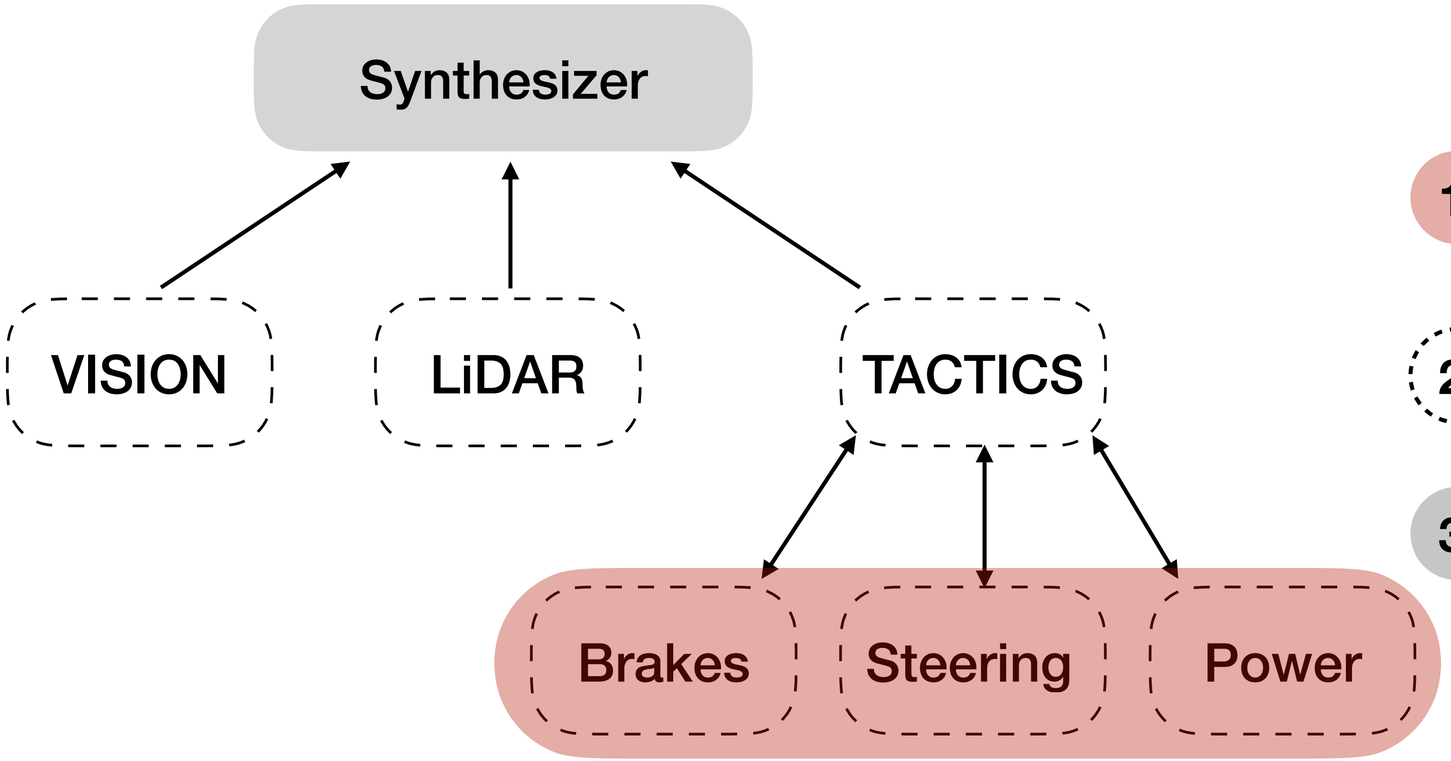
With an Organizational Architecture

- 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

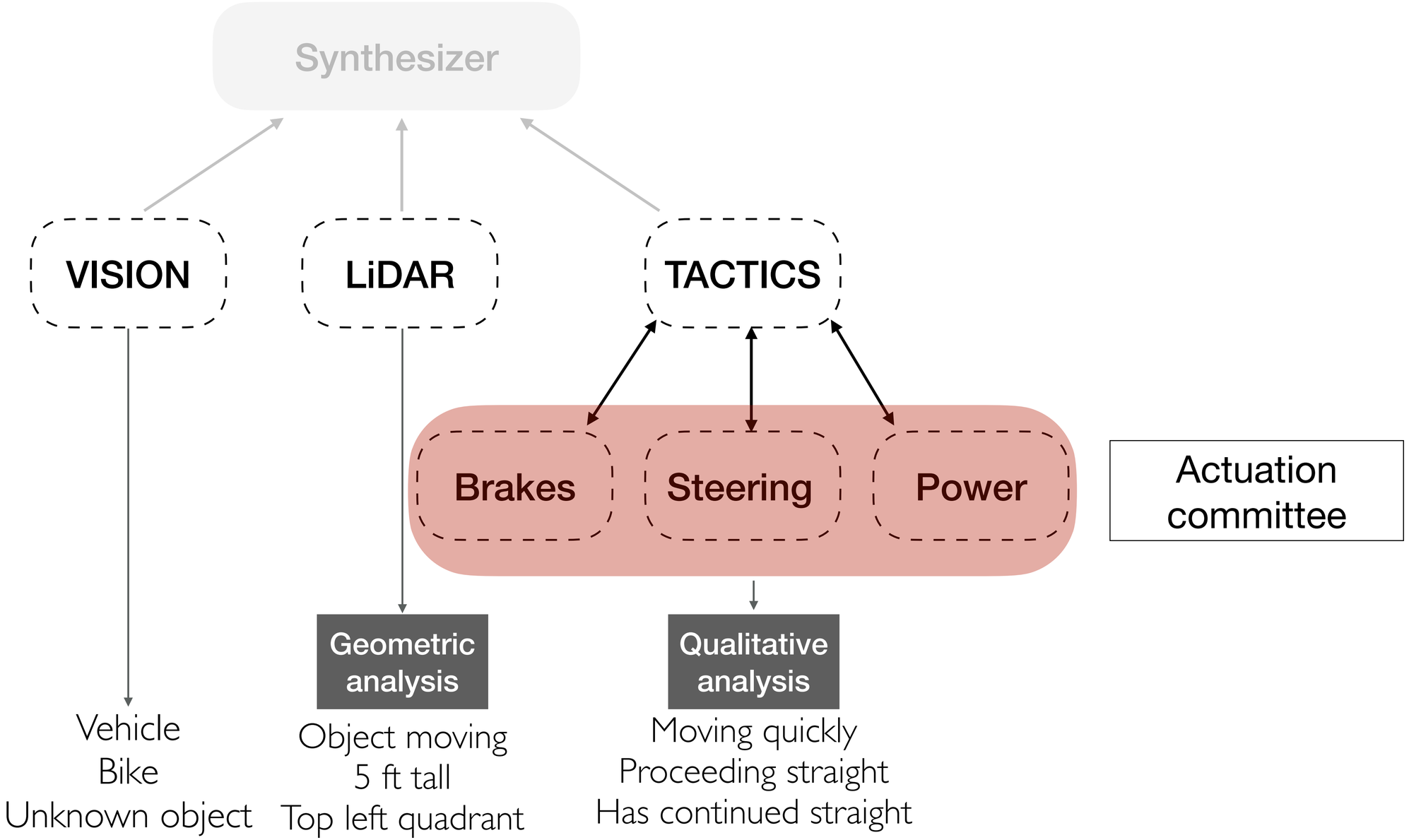


1. Generate Symbolic Qualitative Descriptions for each committee.

2. Input qualitative descriptions into local “reasonableness” monitors.

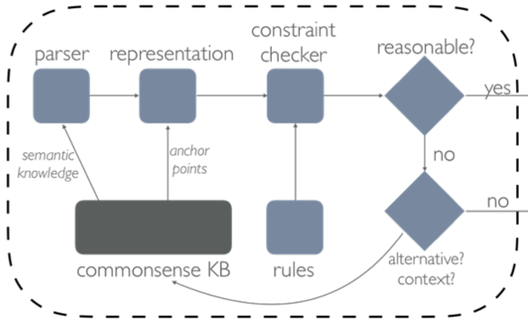
3. Use a synthesizer to reconcile inconsistencies between monitors.

1. Generate Symbolic Qualitative Descriptions for each committee.



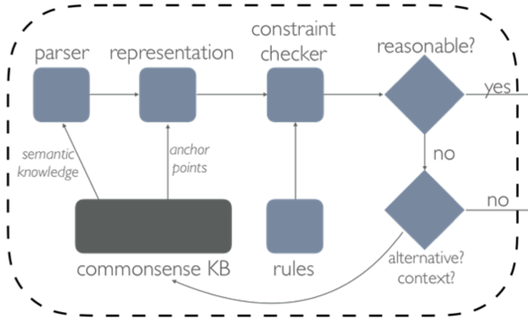
2. Input qualitative descriptions into local "reasonableness" monitors.

Vehicle
Bike
Unknown object



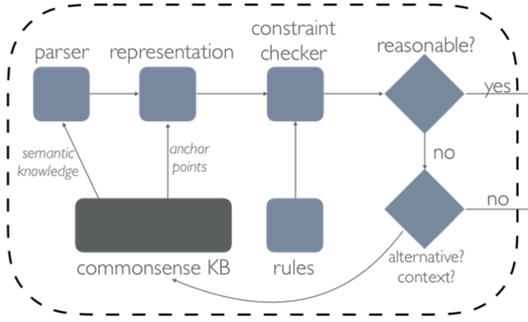
This vision perception is unreasonable. There is no commonsense data supporting the similarity between a vehicle, bike and unknown object except that they can be located at the same location. This component's output should be discounted.

Object moving
5 ft tall
Top left quadrant



This lidar perception is reasonable. An object moving of this size is a large moving object that should be avoided.

Moving quickly
Proceeding straight
Has continued straight



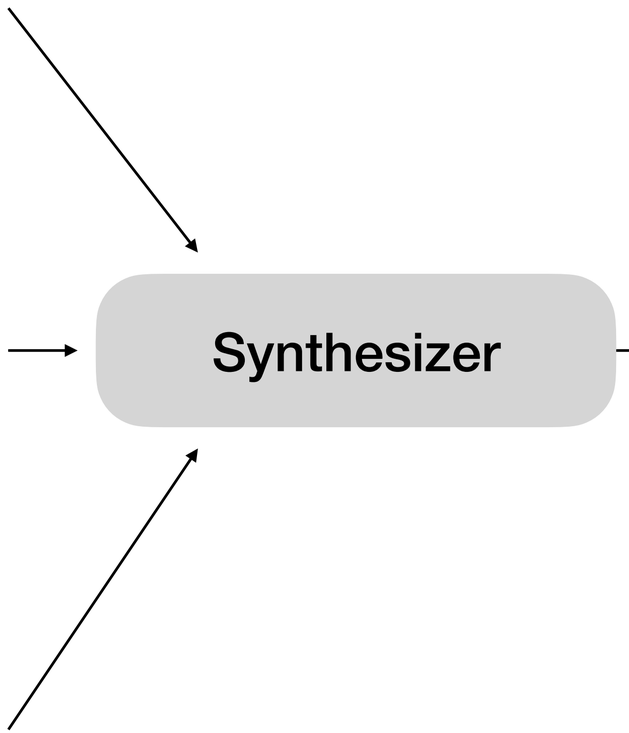
This system state is reasonable given that the vehicle has been moving quickly and proceeding straight for the last 10 second history.

3. Use a synthesizer to reconcile inconsistencies between monitors.

This vision perception is unreasonable. There is no commonsense data supporting the similarity between a vehicle, bike and unknown object except that they can be located at the same location. This component's output should be discounted.

This lidar perception is reasonable. An object moving of this size is a large moving object that should be avoided.

This system state is reasonable given that the vehicle has been moving quickly and proceeding straight for the last 10 second history.



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.

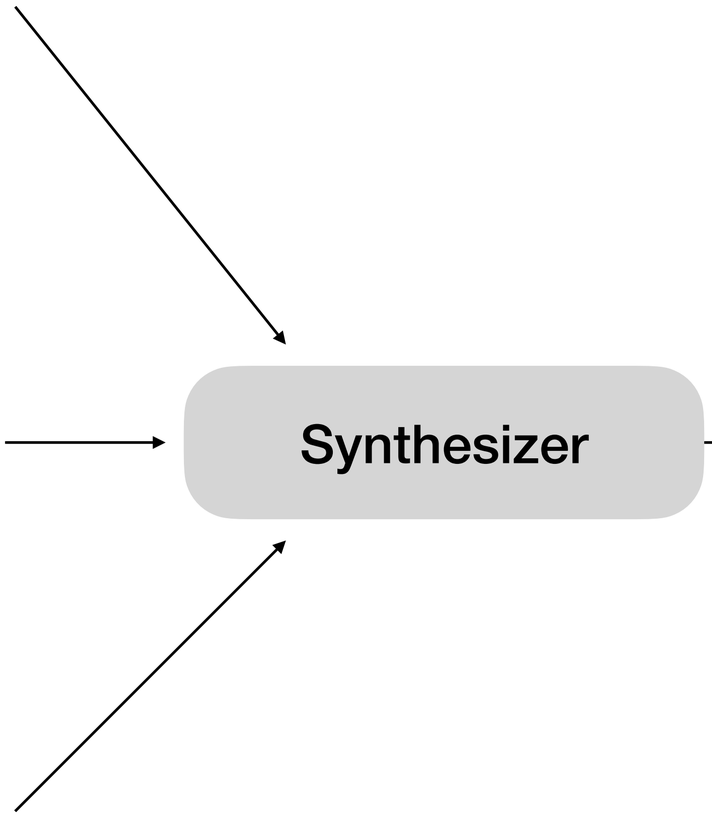
3. Use a synthesizer to reconcile inconsistencies between monitors.

Symbolic reasons

```
(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_data, isType, sensor)
…
(input_data[4], hasSize, large)
(input_data[4], IsA, large_object)
(input_data[4], moving, True)
(input_data[4], hasProperty, 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)
```



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.

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.

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

$$\Rightarrow \big(\textbf{passenger, hasProperty, safe}\big)$$

A passenger is safe if:

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

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

$$\Leftrightarrow \big((x, isA, threat), \textbf{state}, s\big))$$

3. Use a synthesizer to reconcile inconsistencies between monitors.

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

$$\Rightarrow \big(\textbf{passenger, hasProperty, safe}\big)$$

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')
```

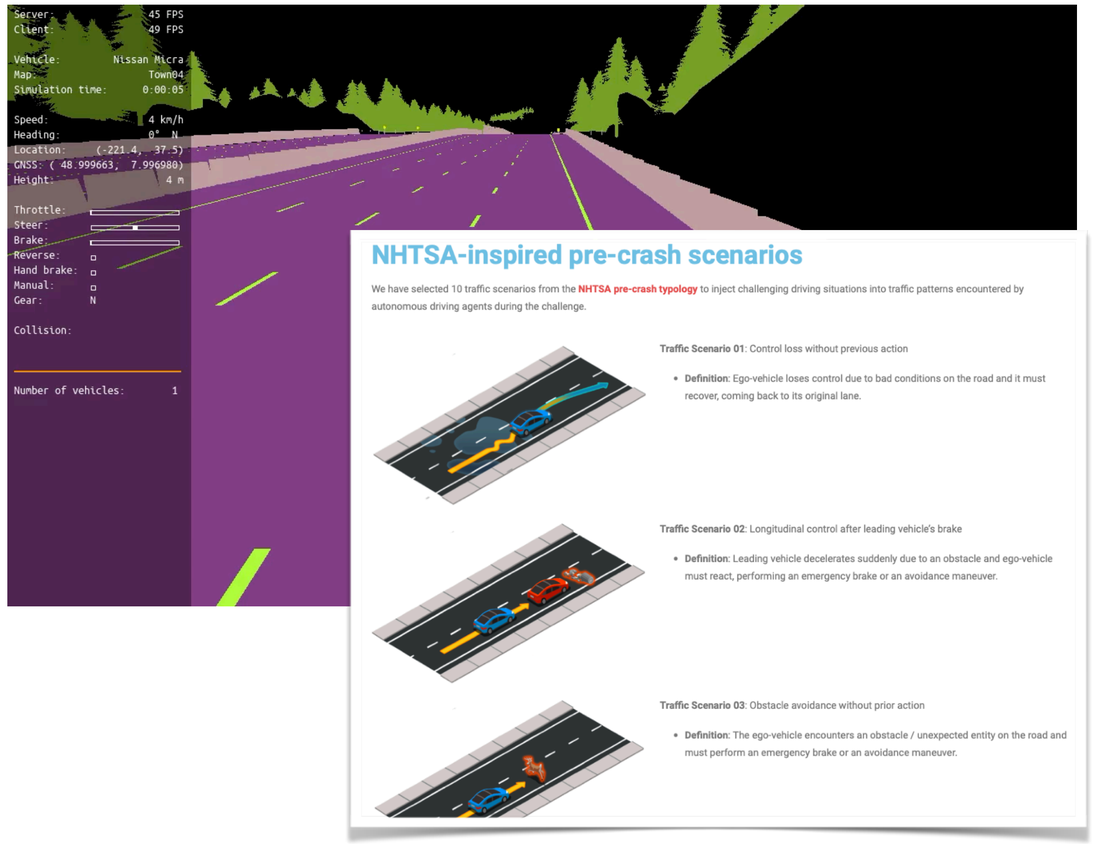
!

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.

Evaluation in Simulation

Evaluation

Real-world Inspired Scenarios

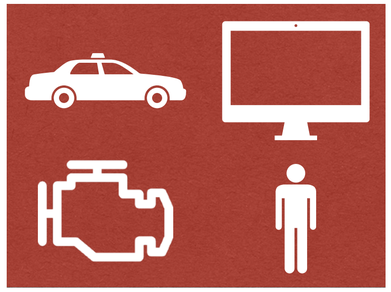


Reconcile Inconsistencies

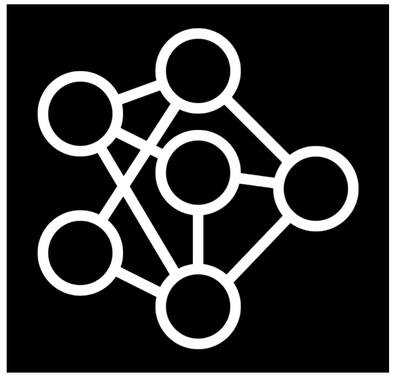
- Detection: Generate logs from scenarios to detect failures.
- Insert errors: Scrambling *multiple* labels on existing datasets.
- Real errors: Examining errors on the validation dataset of NuScenes leaderboard.

Priority	Correctness	False Positives	False Negatives
No synthesizer	85.6%	7.1%	7.3%
Single subsystem	88.9%	7.9%	3.2%
Safety	93.5%	4.8%	1.7%

Defense Outline



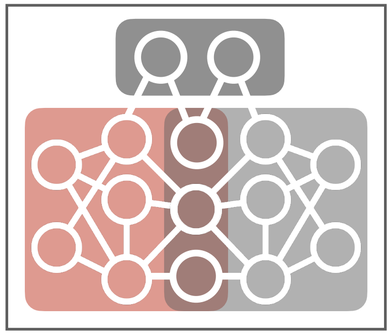
Problem: Complex systems are imperfect.



Error detection for local subsystems.

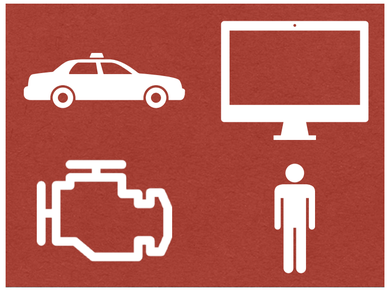
Opaque subsystems.

Sensor subsystem interpretation.



System-wide failure detection.

Vision: Articulate systems by design.



Problem: Complex mechanisms are imperfect.

Explanation

Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter and Lalana Kagal
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{lgilpin, davidbau, bzy, abajwa, specter, lkagal}@ mit.edu

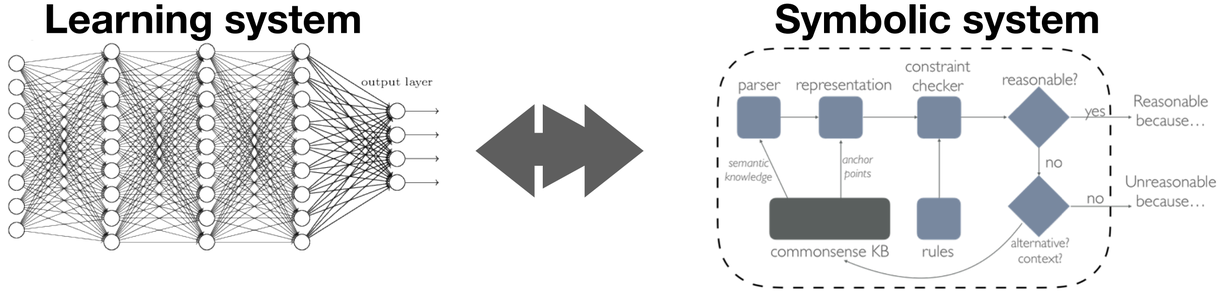
Dynamic explanations, under uncertainty

Self-explaining architectures

Vision: Articulate Machines

Coherent Communication

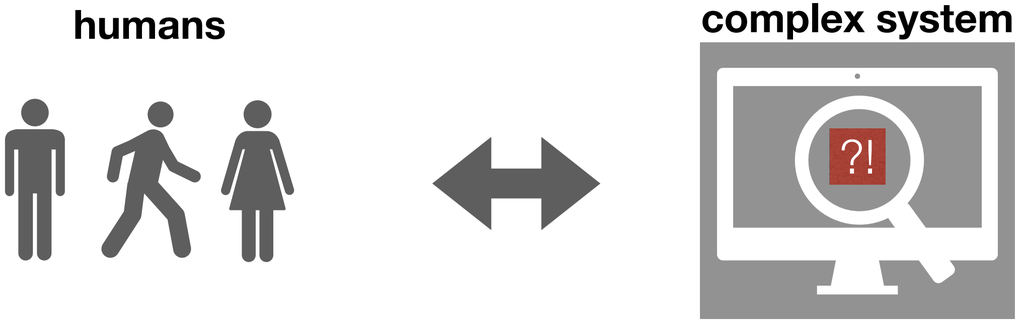
With Other Systems



Common language to complete tasks.

- Redundancy: systems solve problems in multiple ways.
- Hybrid processes: systems that learn from each other.

With Humans



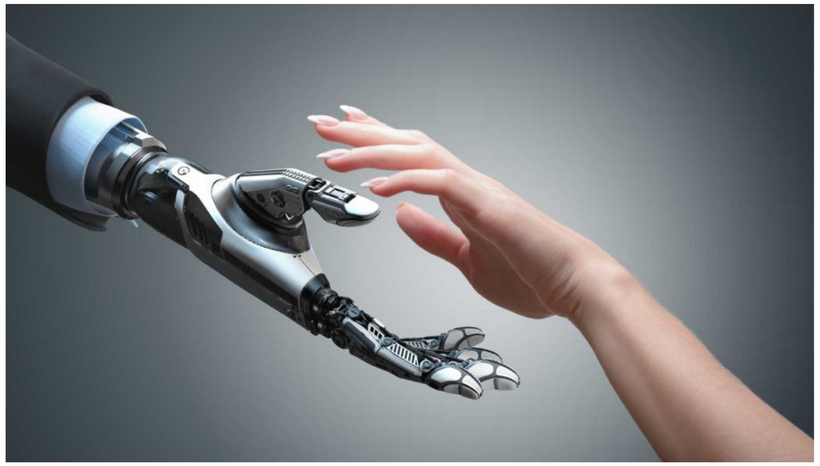
Explanations are a debugging language.

- Debugging: humans can improve complex systems
- Education: complex systems can “improve” or teach humans.

Impact

Confidence and Integrity of Systems

Society



Systems that articulately communicate with humans on shared tasks.

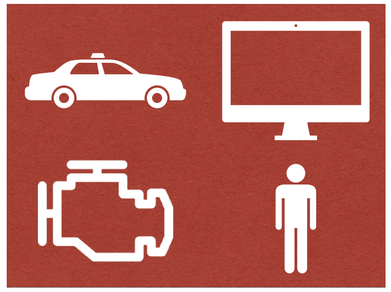
Liability

Systems that can testify, answer questions, and provide insights.

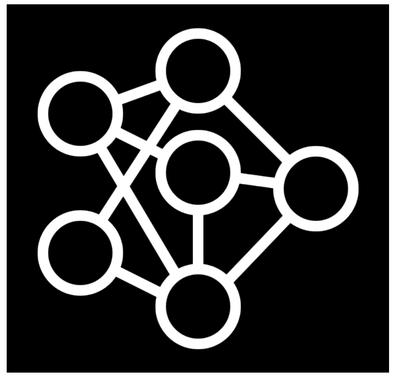
Robustness

Dynamic detection of failure and intrusion with precise mitigation.

Thesis Contributions

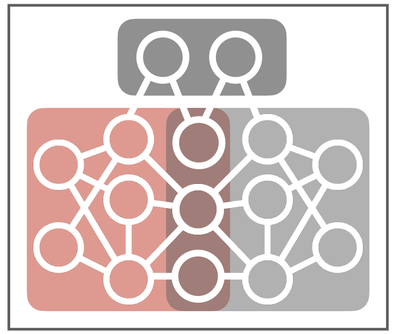


Complex systems need better communication and sanity checks.



Reasonableness monitor for opaque subsystems.

Qualitative representations of sensor data.



An architecture to reason about unreliable parts.

Explanations as a common language.

“You can do it, only you can do it, you can't do it alone.”

Patrick Henry Winston

My committee

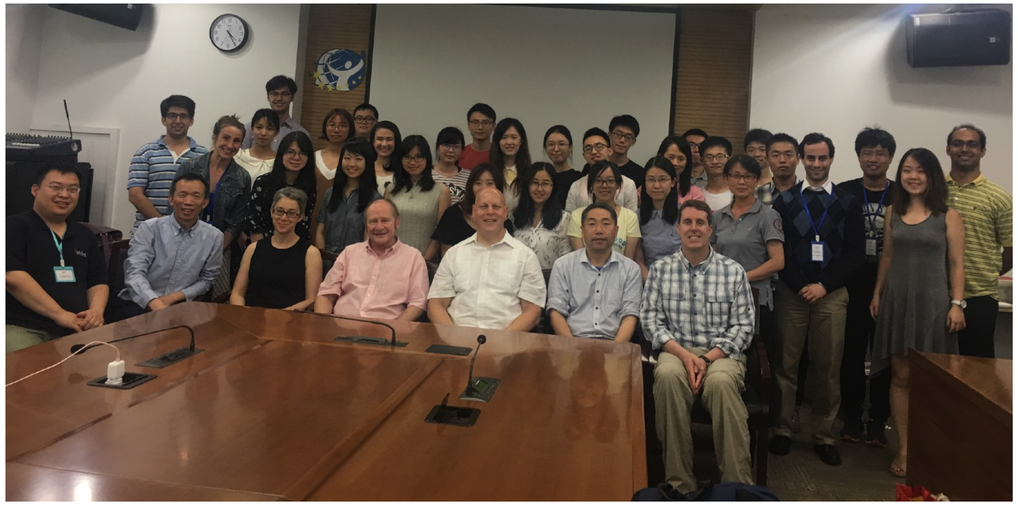
Gerald Jay Sussman, Lalana Kagal, Jacob Andreas, Julie Shah, and Howard Shrobe

Funding

Toyota Research Institute (TRI), Sloan

MIT Academic Community

IPRI, the Genesis Group, EECS / CSAIL



Collaborators

“Fellow Travelers”

- Elizabeth Han
- Evelyn Florentine
- Ishan Pakuwal
- Marla E. Odell
- Matthew Kalinowski
- Michal Reda
- Obada Alkhatib
- Tianye Chen
- Vishnu S. Penubarthi
- Zoe Lu
- Ayesha Bajwa
- Jamie C. Macbeth
- Cagri H. Zaman
- Danielle M. Olson
- Ben Z. Yuan
- Mike Specter
- David Bau
- Tarfah Alrashed
- Cecilia Testart
- Nathanial Frutcher
- Julius Adebayo

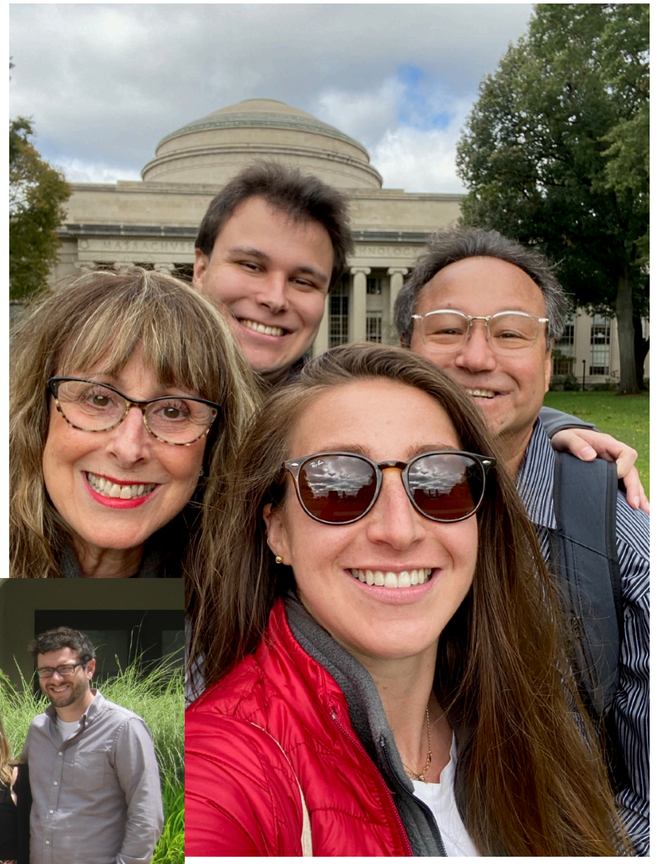
And many more from PARC, INRIA, Stanford, UCSD, DIMACS

Previous Academic Pursuits

PARC Colleagues, Stanford iCME, and UCSD

Family

Brian, Patty (parents) and Cory Gilpin (brother)



Social, Living, and Athletic Communities

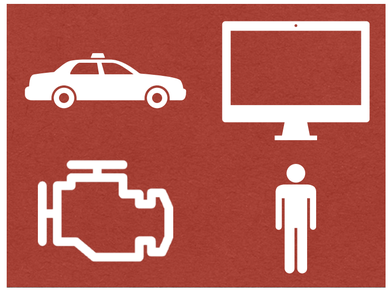
Burton-Conner, Club sports, Roommates

Friends

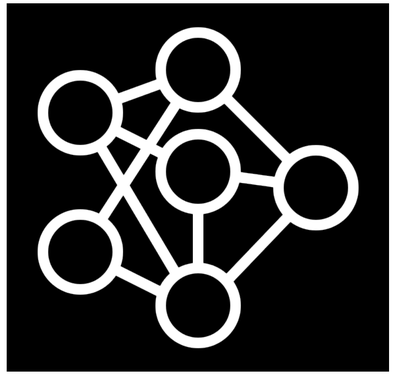
A remembrance

Patrick Henry Winston

Thesis Contributions



Complex systems need better communication and sanity checks.

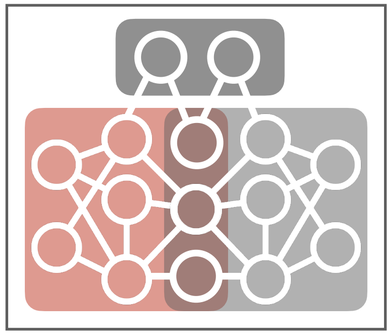


Reasonableness monitor for opaque subsystems.

AAMAS 2019
ACS 2018
AAAI 2018
ICLR Workshop 2019

Qualitative representations of sensor data.

AAAI SS 2016



An architecture to reason about unreliable parts.

AAAI FS 2019

Explanations as a common language.

NeurIps Workshop 2018
DSAA 2018.