Explaining Multimodal Errors in Autonomous Vehicles

Leilani H. Gilpin*, Vishnu Penubarthi, and Lalana Kagal

DSAA 2021

Problem: AVs Have 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

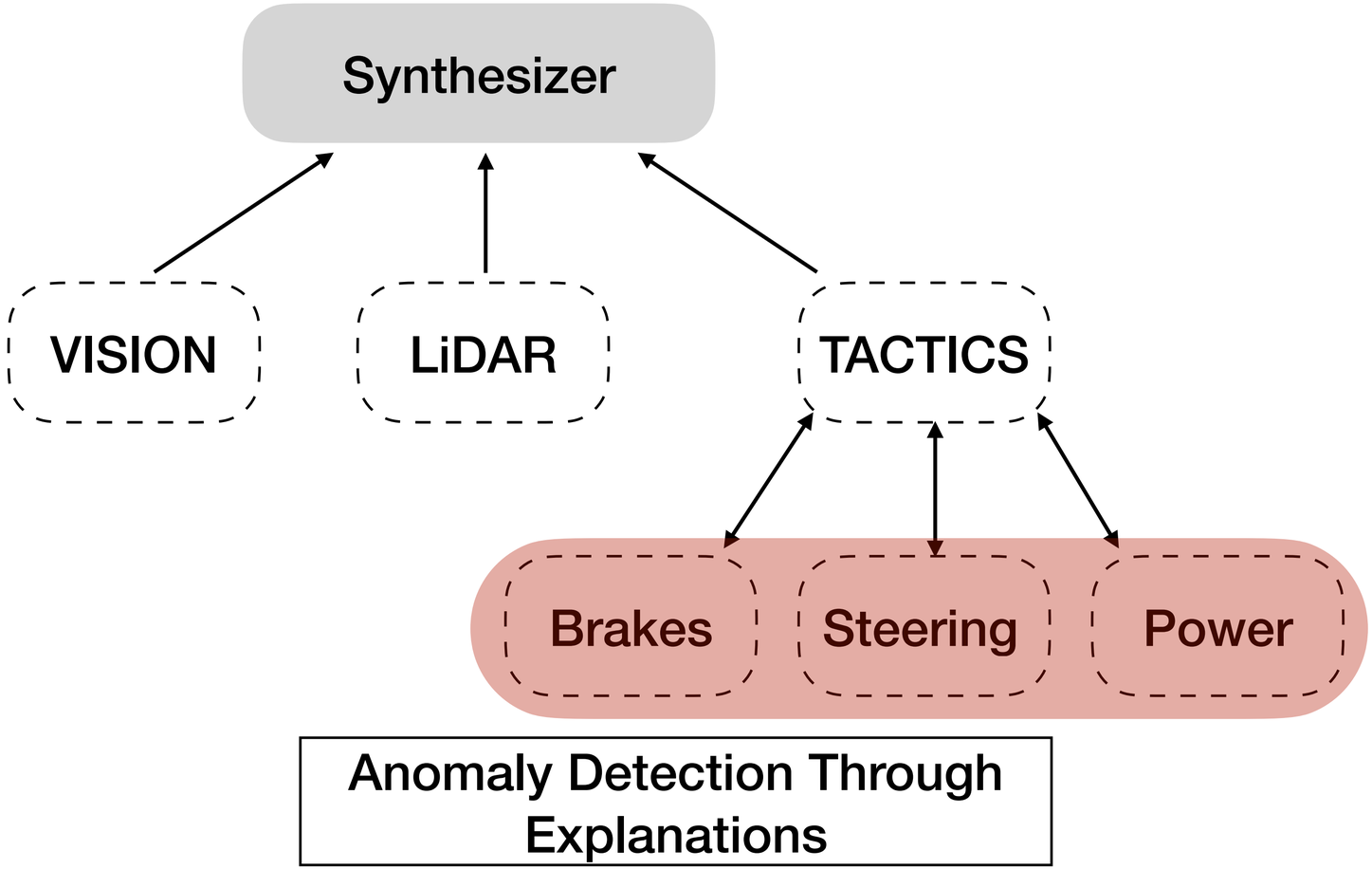
GM and Cruise are testing vehicles in a chaotic city, and the tech still has a ways to go.

A Deadly Crash

Reconciling Internal Disagreements

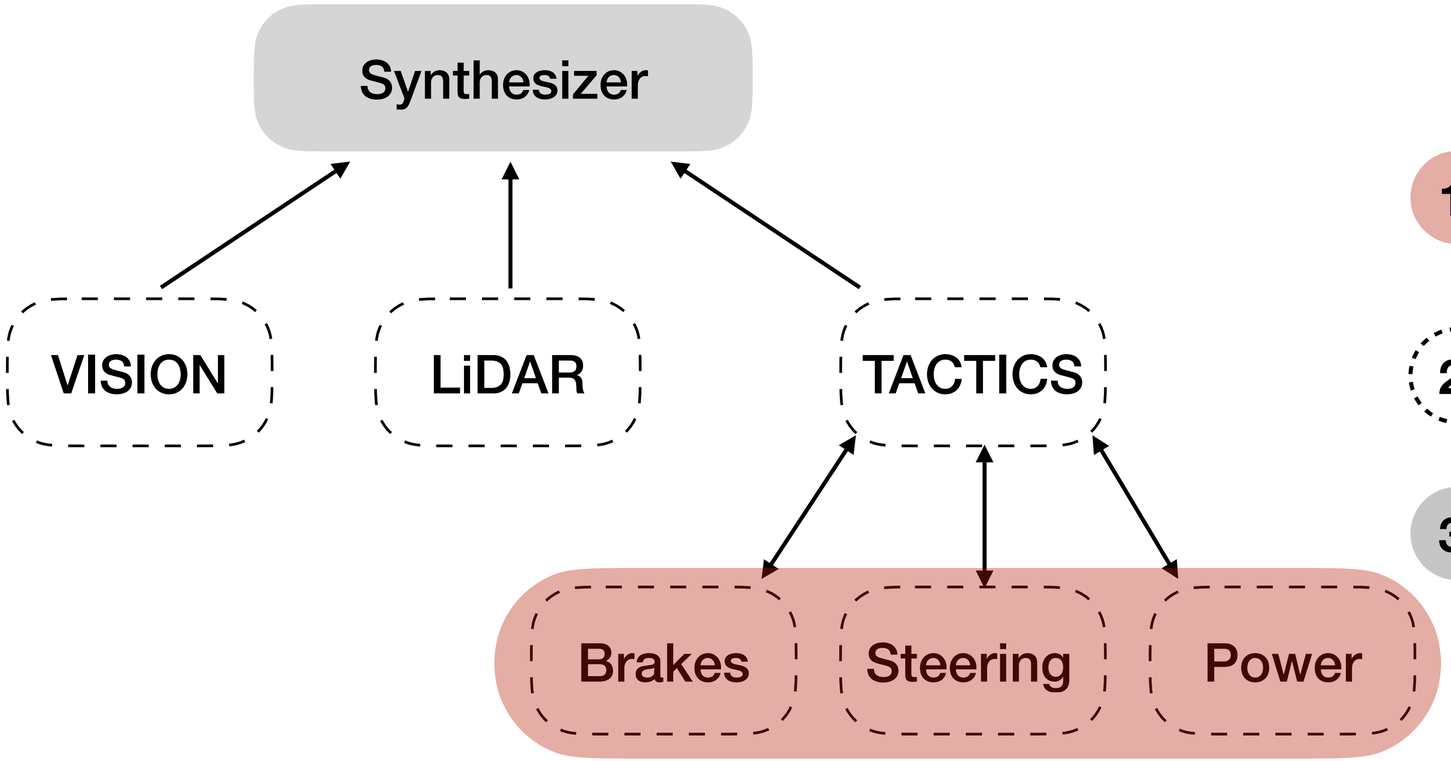
With an Organizational Architecture

- 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

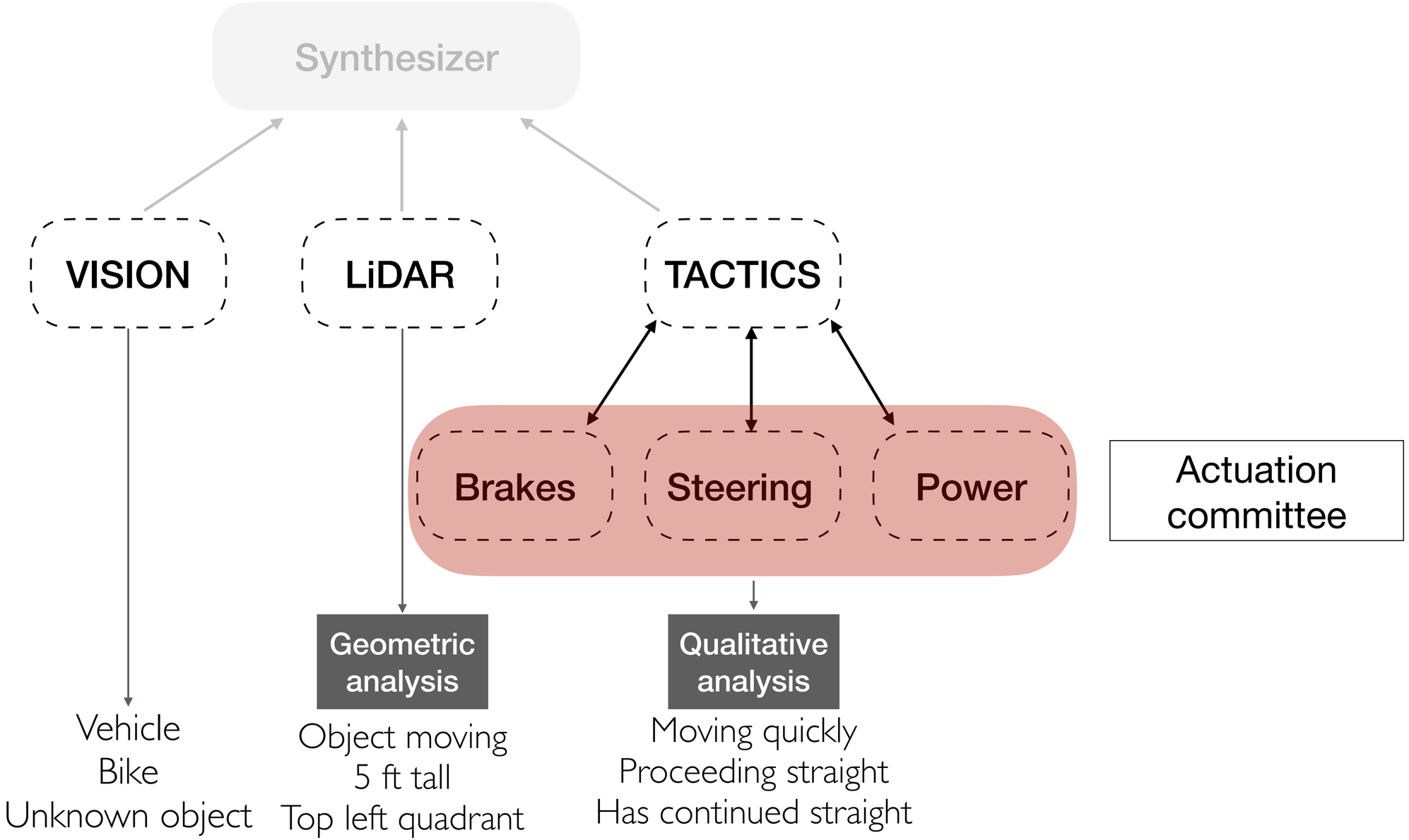


1. Generate Symbolic Qualitative Descriptions for each committee.

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

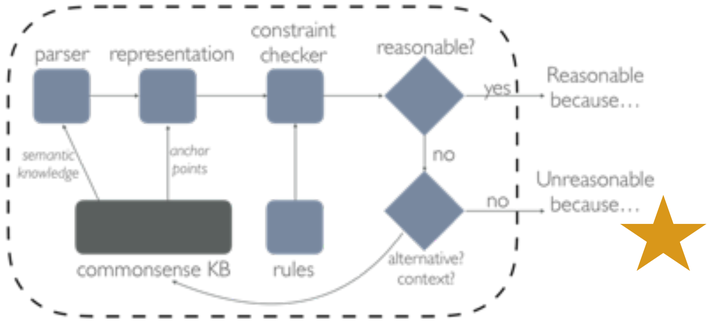
3. Use a synthesizer to reconcile inconsistencies between monitors.

1. Generate Symbolic Qualitative Descriptions for each committee.



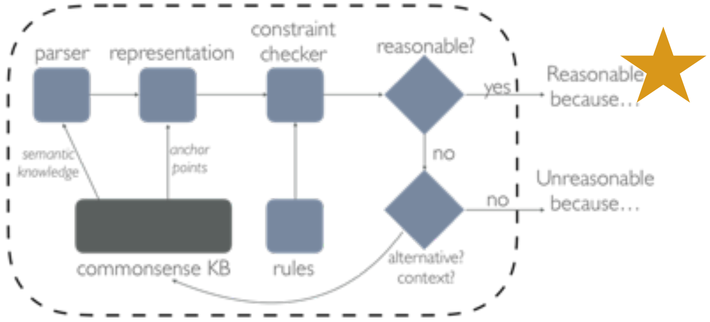
2. Input qualitative descriptions into local "reasonableness" monitors.

Vehicle
Bike
Unknown object



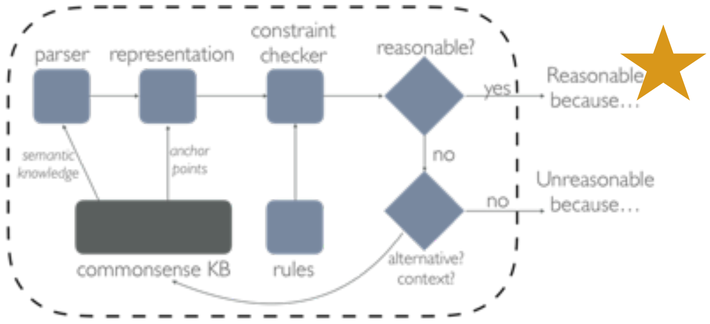
This vision perception is unreasonable. There is no commonsense data supporting the similarity between a vehicle, bike and unknown object except that they can be located at the same location. This component's output should be discounted.

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



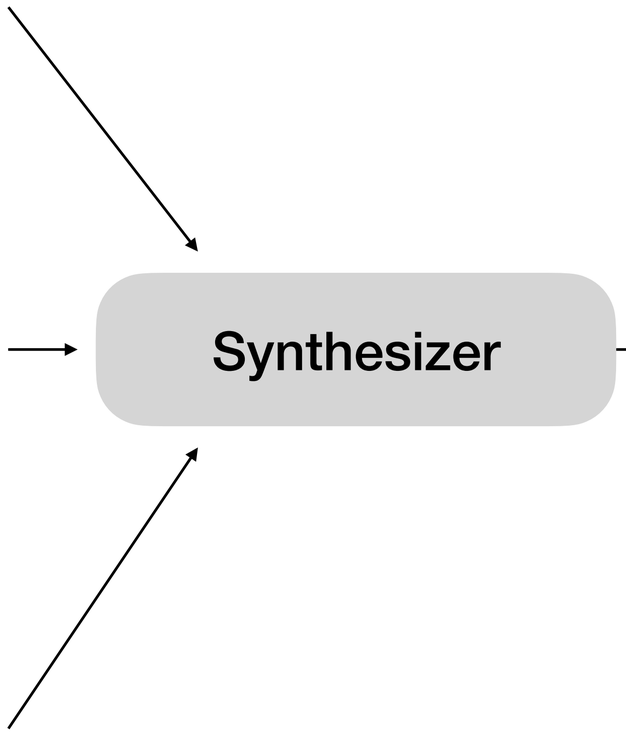
This system state is reasonable given that the vehicle has been moving quickly and proceeding straight for the last 10 second history.

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.

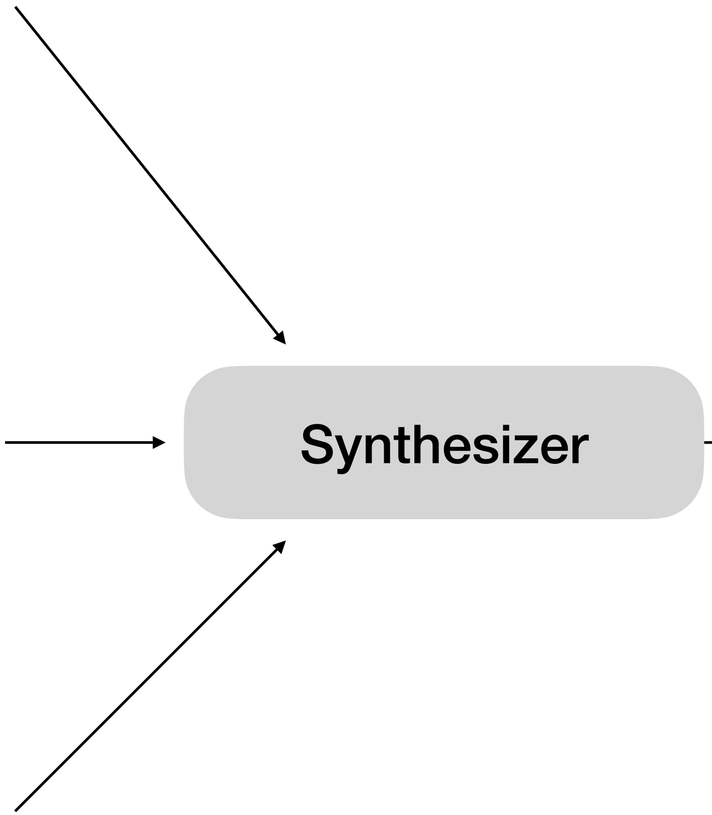
3. Use a synthesizer to reconcile inconsistencies between monitors.

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.

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

A passenger is safe if:

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

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

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

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

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 (?y)’,
    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)
```

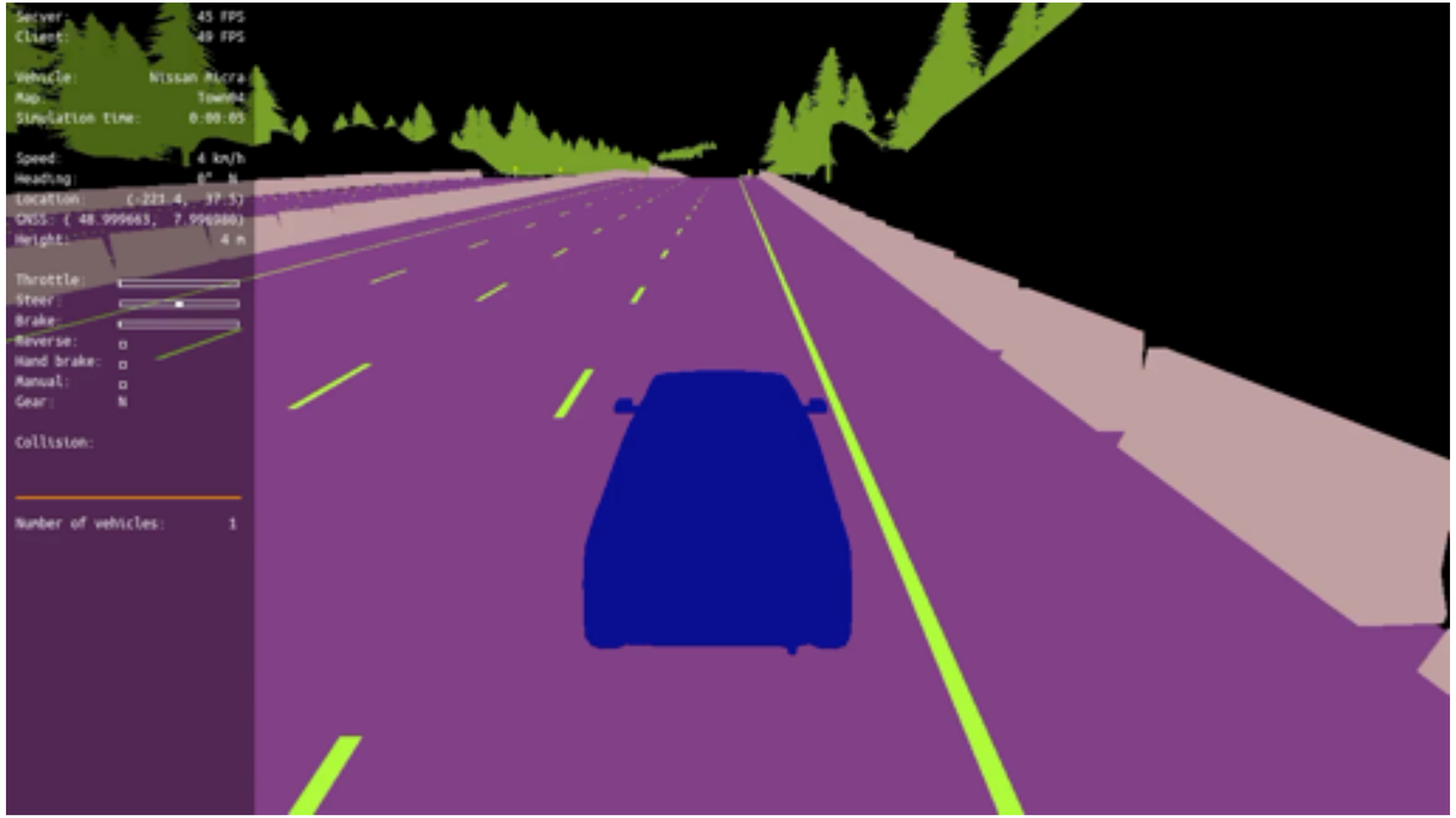
Abstract Goal Tree

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

!

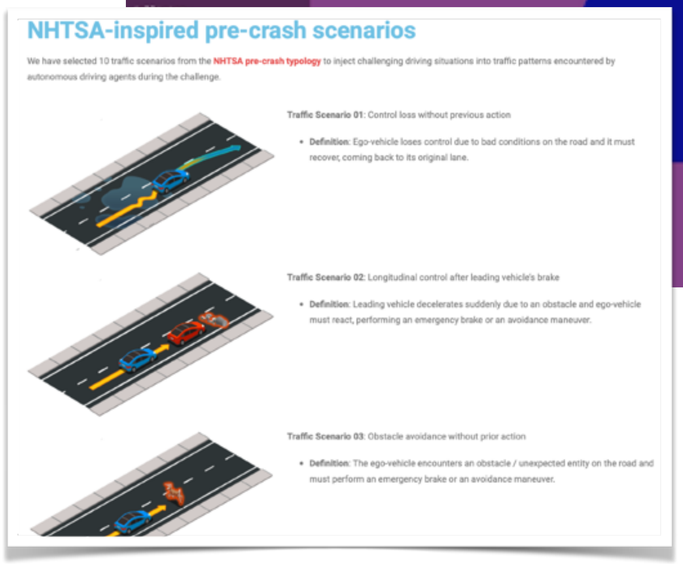
The best option is to veer and slow down. The vehicle is traveling too fast to suddenly stop. The vision system is inconsistent, but the lidar system has provided a reasonable and strong claim to avoid the object moving across the street.

Evaluation in Simulation



Evaluation

Real-world Inspired Scenarios



NHTSA-inspired pre-crash scenarios

We have selected 10 traffic scenarios from the **NHTSA pre-crash typology** to inject challenging driving situations into traffic patterns encountered by autonomous driving agents during the challenge.

Traffic Scenario 01: Control loss without previous action

- **Definition**: Ego-vehicle loses control due to bad conditions on the road and it must recover, coming back to its original lane.

Traffic Scenario 02: Longitudinal control after leading vehicle's brake

- **Definition**: Leading vehicle decelerates suddenly due to an obstacle and ego-vehicle must react, performing an emergency brake or an avoidance maneuver.

Traffic Scenario 03: Obstacle avoidance without prior action

- **Definition**: The ego-vehicle encounters an obstacle / unexpected entity on the road and must perform an emergency brake or an avoidance maneuver.

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%

Contributions

- An organizational architecture, ADE, to mitigate inconsistencies between parts.
- A reasoning system: an explanation synthesizer that uses a priority hierarchy to determine which parts to trust.
- An implementation of ADE for an autonomous vehicle, evaluated in Carla.