

Verified Artificial Intelligence and Autonomy

Sanjit A. Seshia

Professor

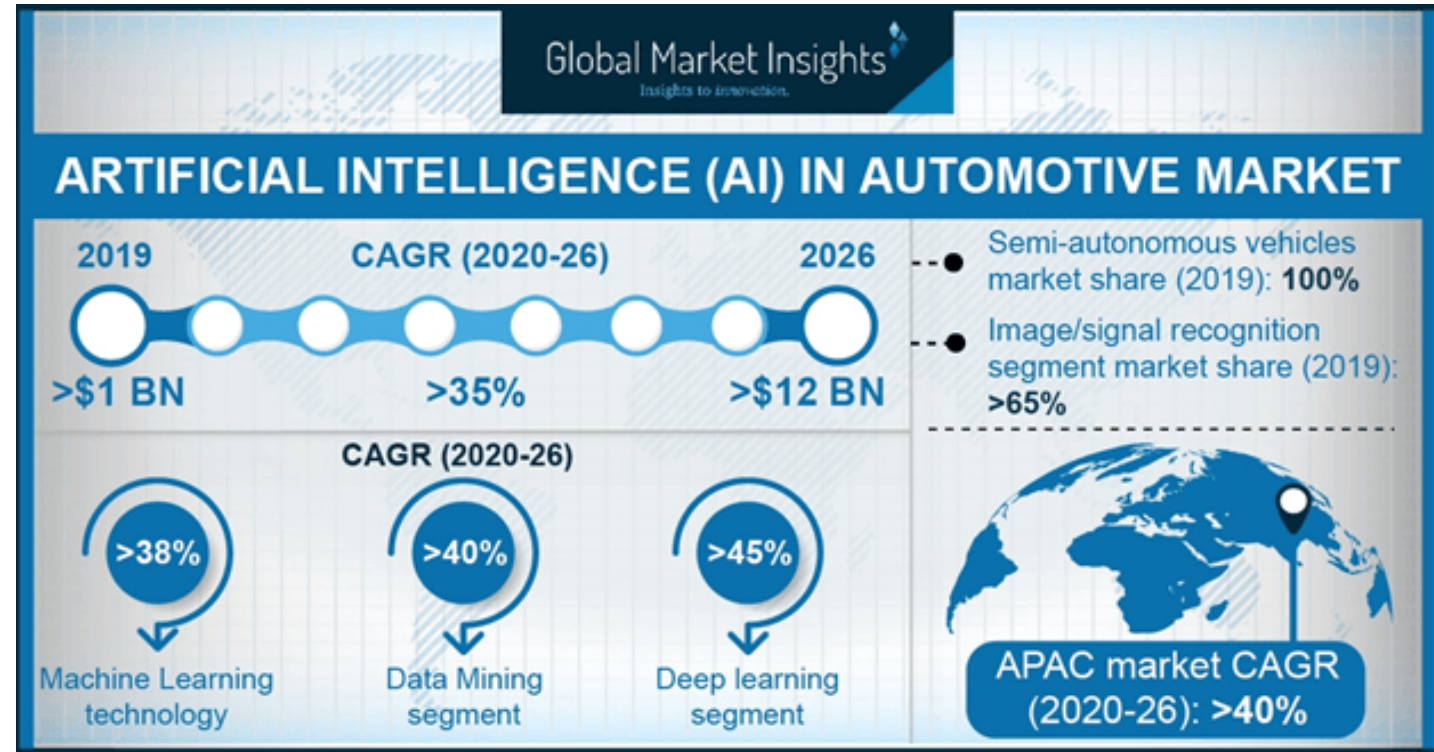
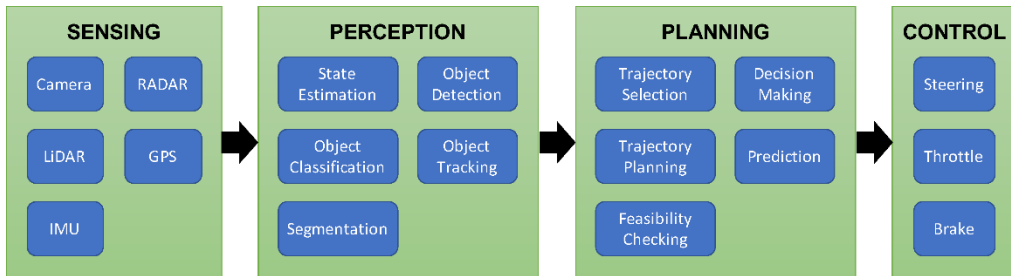
EECS Department, UC Berkeley



IFIP WG10.4 IVDS Workshop
January 31, 2021

VeriCal
<http://vehical.org>

Growing Use of Machine Learning/Artificial Intelligence in Safety-Critical Autonomous Systems



Source: gminsights.com

Growing Concerns about Safety:

- Numerous papers showing that *Deep Neural Networks can be easily fooled*
- *Accidents*, including some *fatal*, involving potential failure of AI/ML-based perception systems in self-driving cars

Can Formal Methods Help?

Formal methods = Mathematical, Algorithmic techniques for modeling, design, analysis

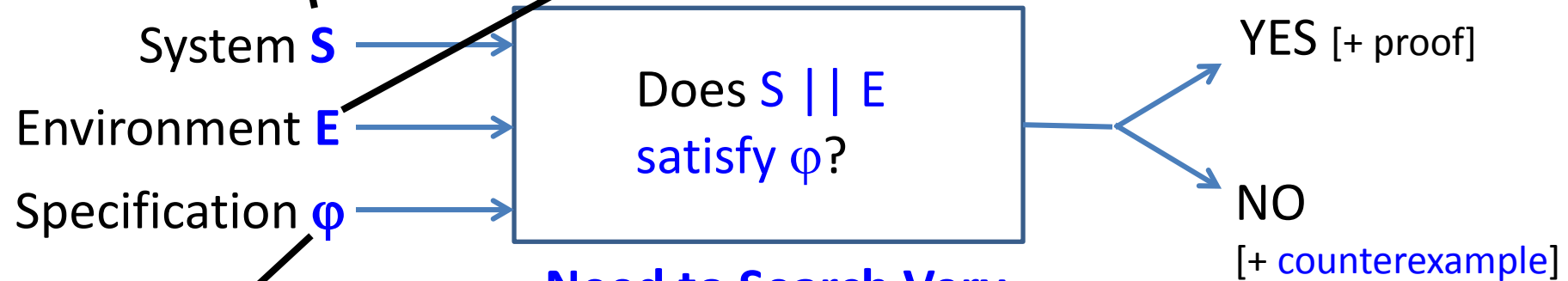
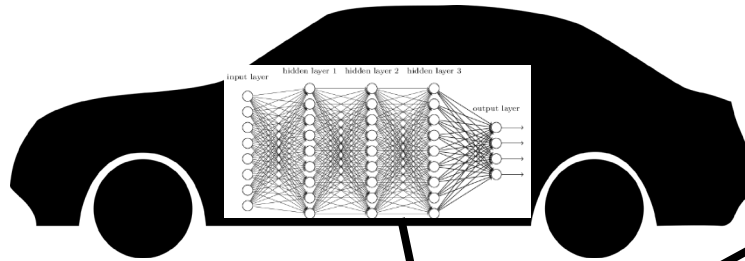
- Specification: WHAT the system must/must not do
- Verification: WHY it meets the spec. (or not)
- Synthesis: HOW it meets the spec. (correct-by-construction design)

***Can we address the Design & Verification challenges of AI/ML-based
Autonomy with Formal Methods?***

Challenges for Verified AI

S. A. Seshia, D. Sadigh, S. S. Sastry.

Towards Verified Artificial Intelligence. July 2016. <https://arxiv.org/abs/1606.08514>.








Need to Search Very High-Dimensional Input and State Spaces



Design Correct-by-Construction?

Need Principles for Verified AI

Challenges

1. Environment (incl. Human) Modeling 
2. Formal Specification 
3. Learning Systems Representation 
4. Scalable Training, Testing, Verification 
5. Design for Correctness 

Principles



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

<http://learnverify.org/VerifiedAI>

Talk Outline

- Environment Modeling
- Simulation-Based Verification
- Simulation → Road Testing
- Principles for Verified AI

Environment Modeling: Know Your Assumptions!

What's Unknown/
Uncertain

Approach

Parameters



Probabilistic Programming and Reasoning
[D. Fremont et al., PLDI 2019]

Behaviors /
Dynamics



Learning Models from Data/Interaction
[D. Sadigh et al., RSS & IROS 2016;
M. Vazquez-Chanlatte et al., NeurIPS 2018]

Agents /
Objects



Introspective Environment Modeling
[S. A. Seshia, RV 2019]

More Challenging



SCENIC: Environment Modeling and Data Generation

- *Scenic* is a **probabilistic programming language** defining *distributions over scenes/scenarios*
- *Use cases*: data generation, test generation, verification, debugging, design exploration, etc.

```
model scenic.domains.driving.model
ego = Car
spot = OrientedPoint on visible curb
badAngle = Uniform(1.0, -1.0) * Range(10, 20) deg
parkedCar = Car left of spot by 0.5,
             facing badAngle relative to roadDirection
```

Example: Badly-parked car



Image
created
with
GTA-V

```
model scenic.domains.driving.model
behavior PullIntoRoad():
    while (distance from self to ego) > 15:
        wait
        FollowLaneBehavior(lane=ego.lane)
ego = Car with behavior DriveAvoidingCollisions
spot = OrientedPoint on visible curb
badAngle = Uniform(1.0, -1.0) * Range(10, 20) deg
parkedCar = Car left of spot by 0.5,
             facing badAngle relative to roadDirection,
             with behavior PullIntoRoad
```



Video
created
with
CARLA

[D. Fremont et al., “Scenic: A Language for Scenario Specification and Scene Generation”, TR 2018, PLDI 2019.]

Some Applications of Scenic

[details in PLDI 2019 paper]

- Data Generation, (Re)-Training
 - More controllable, interpretable
 - Improves performance significantly
 - Rare scenarios, controlled distributions, etc.



Car detection with occlusions

- Debugging Failures
 - Vary scenarios systematically
 - Explain failures of ML



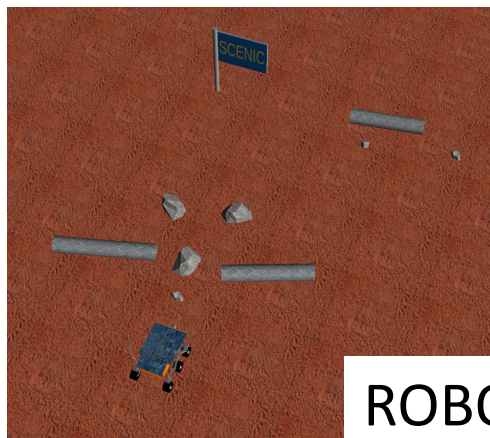
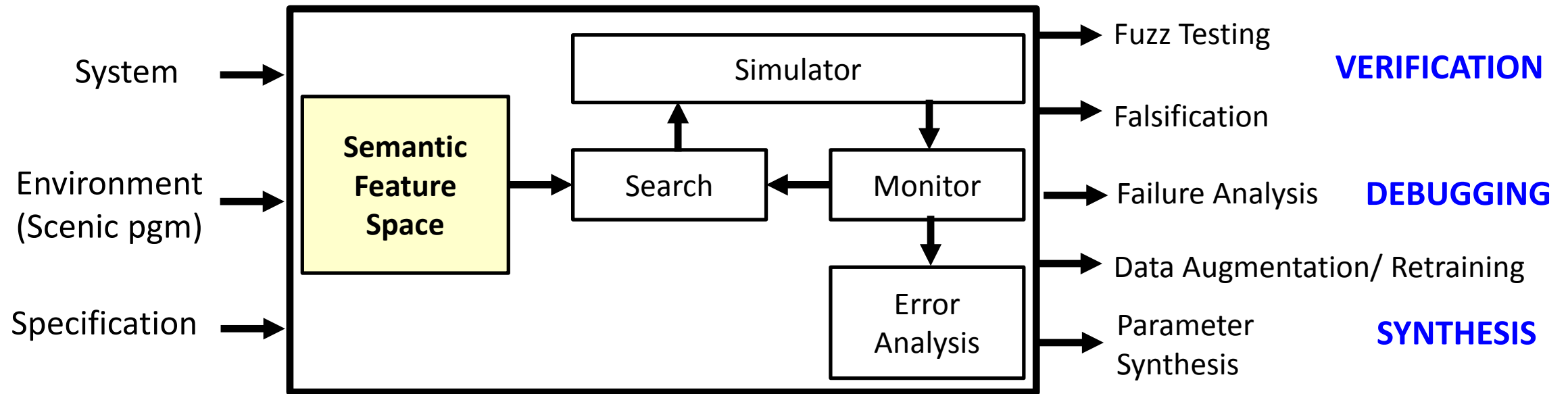
- Design Space Exploration

Test Hypothesis: does the car model lead to a mis-detection?

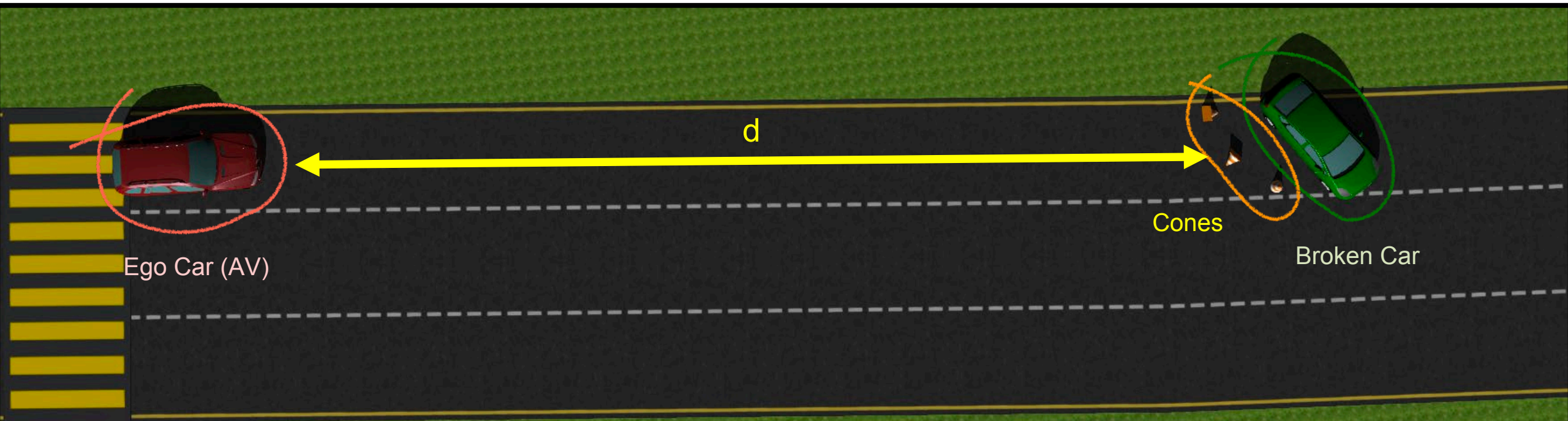
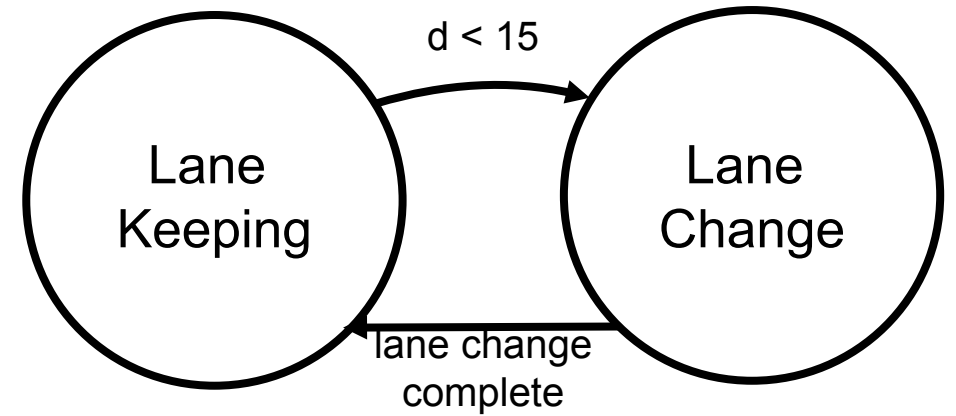
Simulation-Based Verification

- Start with **System-Level Specification**
 - Temporal Logic/Cost Function
 - Transform Logical Spec into Cost Function
 - $G_{[0,\tau]}(\text{dist}(\text{vehicle}, \text{obstacle}) > \delta) \rightarrow \inf_{[0,\tau]} [\text{dist}(\text{vehicle}, \text{obstacle}) - \delta]$
- Falsification: Verification as Optimization
 - Directed search for property violations in simulation
- Scalability requires *Compositional* Falsification
 - Abstract high-dimensional ML (DNN) models
 - Model semantic feature space (e.g. with Scenic program)
 - Semantic adversarial analysis of ML models
 - see [Dreossi, Donze, Seshia, NASA Formal Methods 2017; Dreossi, Jha, Seshia, CAV 2018]

VERIFAI: A Toolkit for the Design and Analysis of AI-Based Systems [CAV 2019] <https://github.com/BerkeleyLearnVerify/VerifAI>



Case Study for Temporal Logic Falsification with VerifAI: Navigation around an accident scenario



Modeling Case Study in the SCENIC Language

```
# Pick location for blockage randomly along curb
blockageSite = OrientedPoint on curb

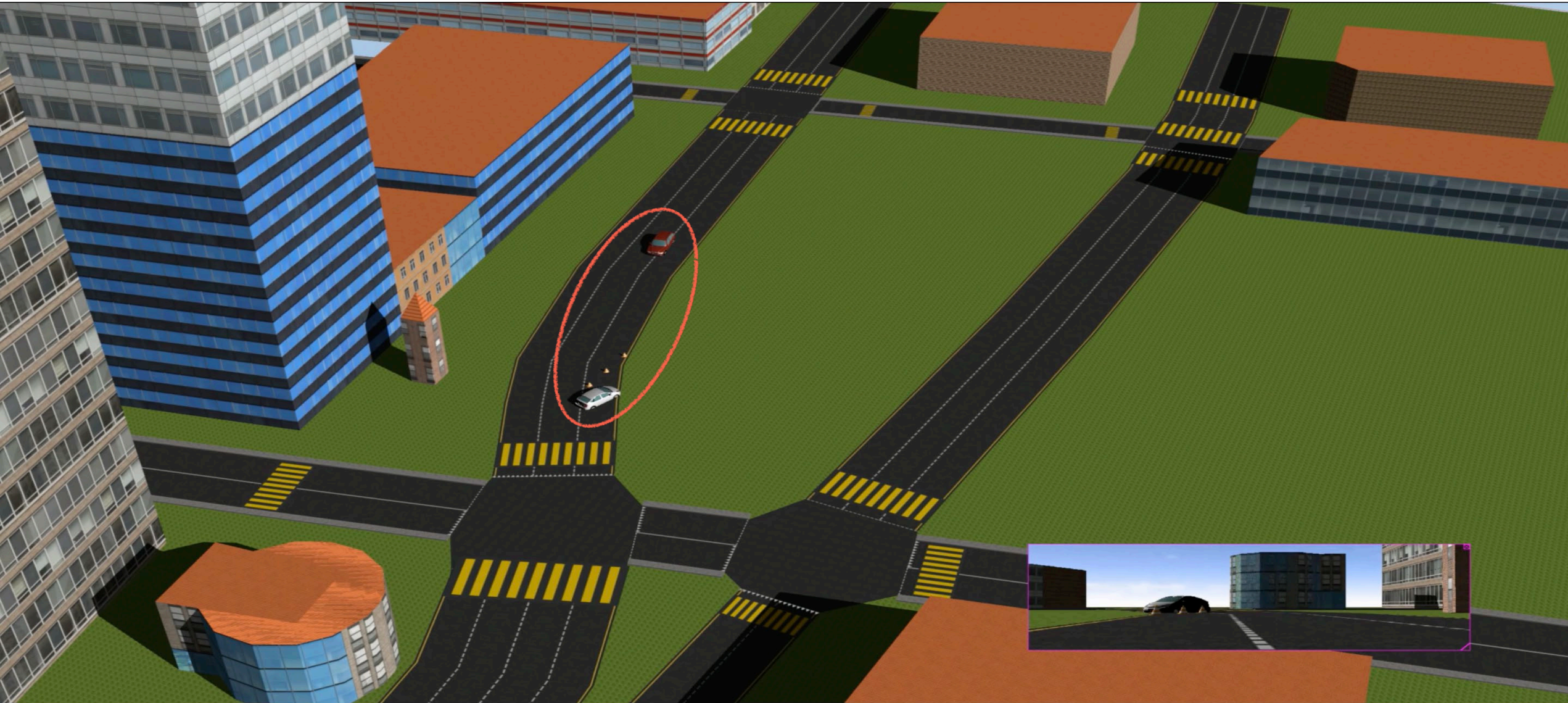
# Place traffic cones
spot1 = OrientedPoint left of blockageSite by (0.3, 1)
cone1 = TrafficCone at spot1,
        facing (0, 360) deg

...

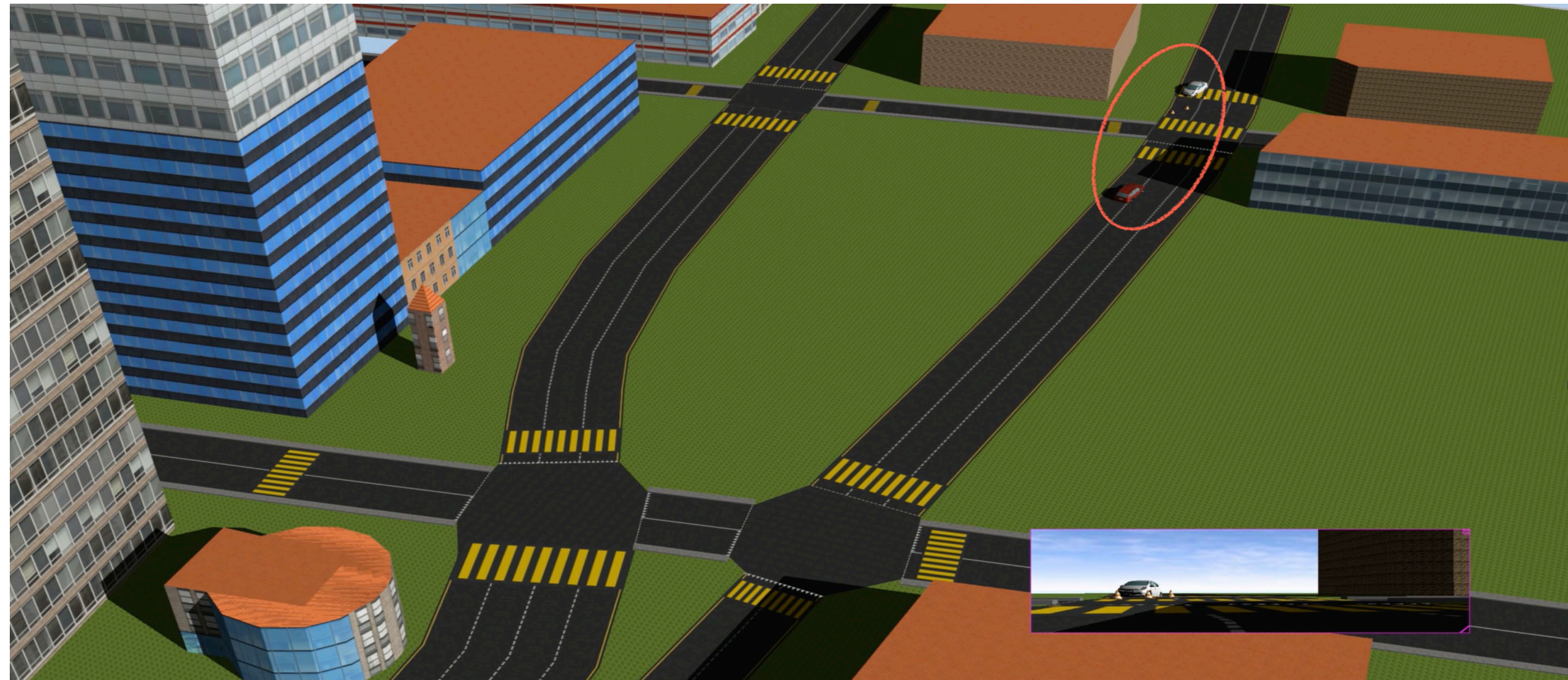
# Place disabled car ahead of cones
SmallCar ahead of spot2 by (-1, 0.5) @ (4, 10),
        facing (0, 360) deg
```



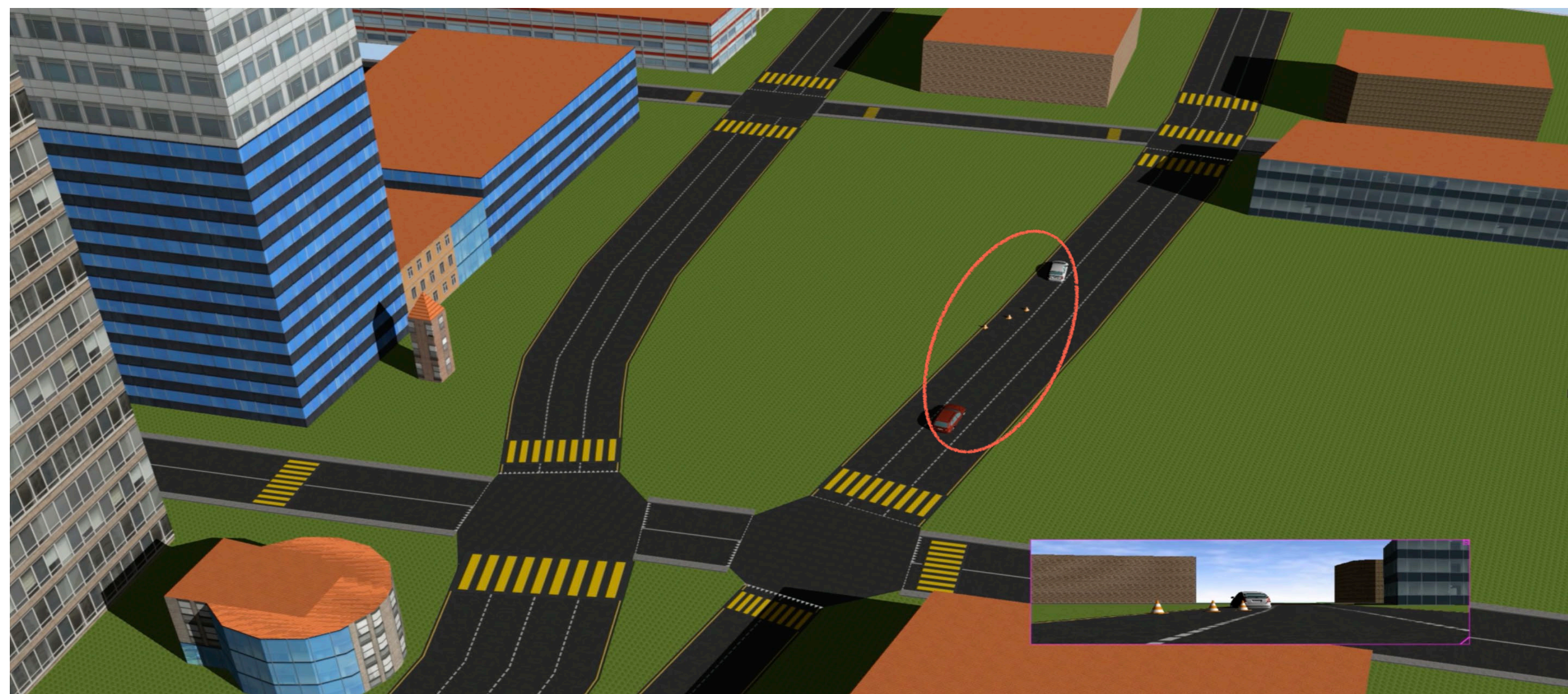
Using Scenic to Generate Initial Scenes



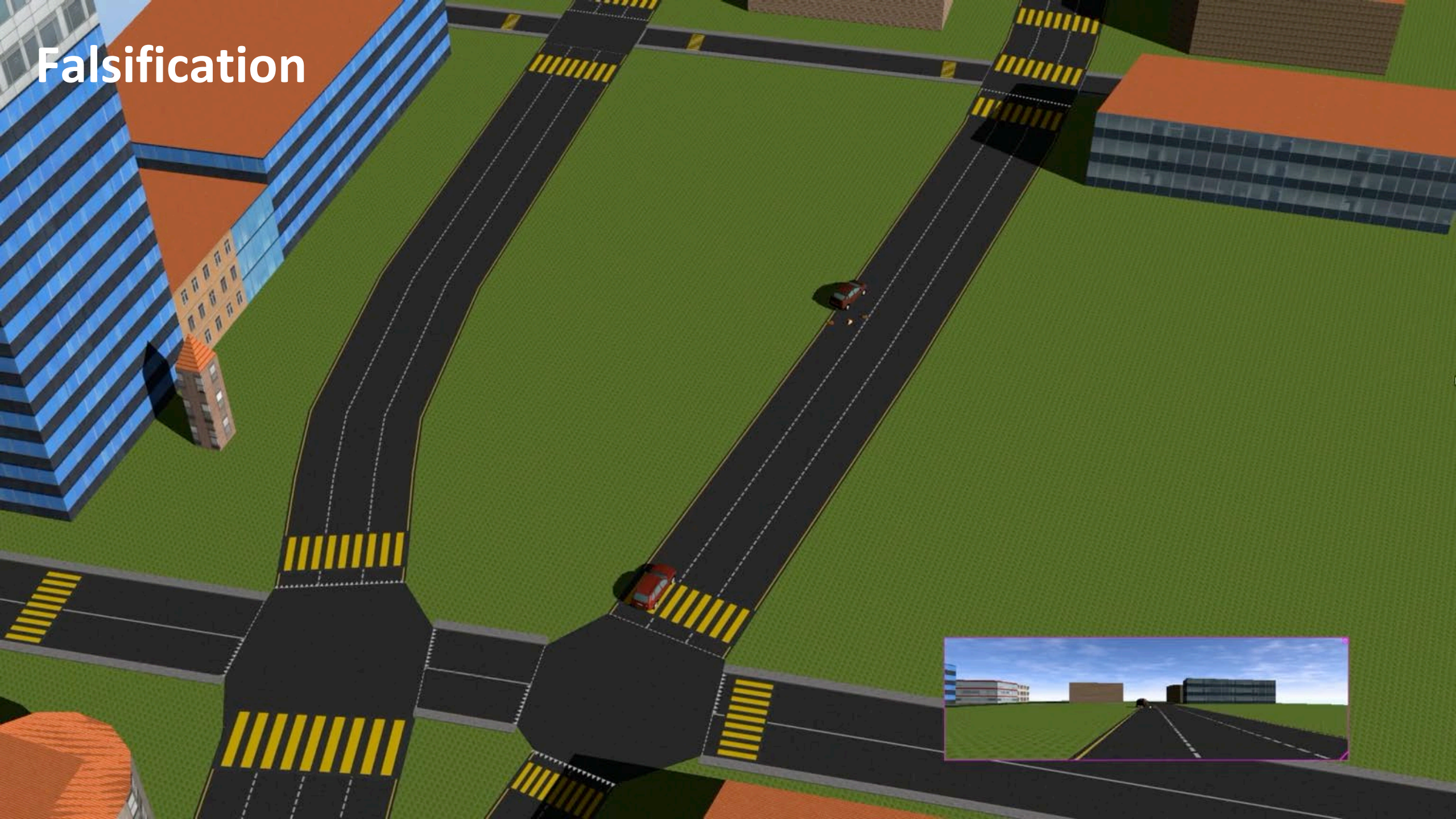
Using Scenic to Generate Initial Scenes



Using Scenic to Generate Initial Scenes



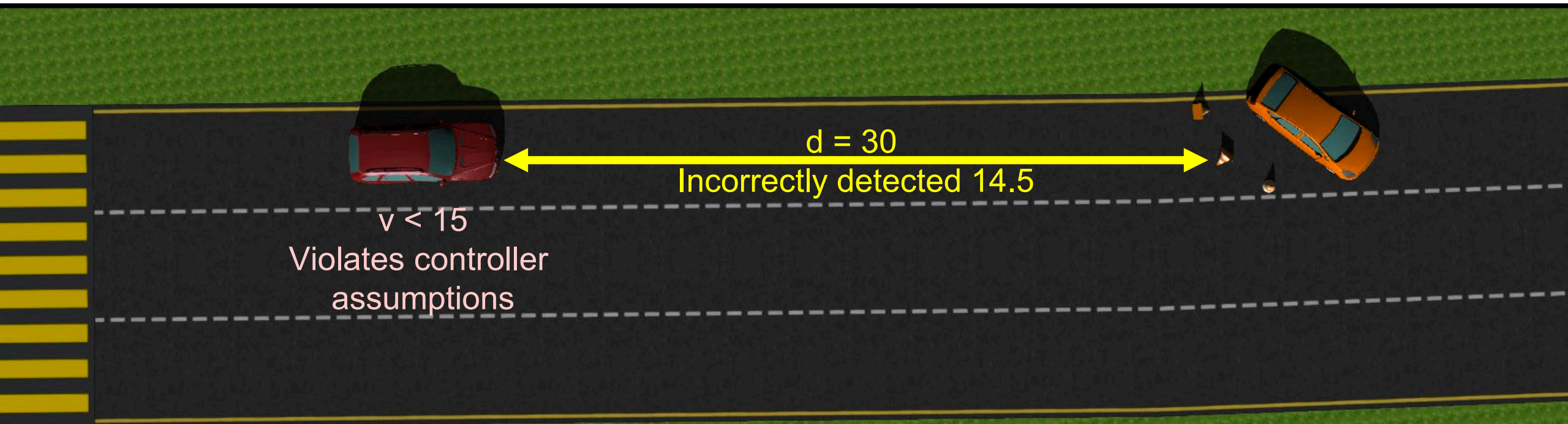
Falsification



Analyzing the failure

Fix the controller:
Update model assumptions
and re-design controller

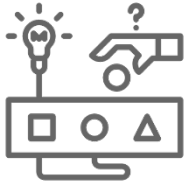
Retrain the perception module:
Collect the counter-example images and
retrain the network [IJCAI'18]



From Simulation to Real-World Testing: Key Questions



#1 Safety violations in simulation: Do they transfer to the real world? How well?



#2 Scenario testability: Can we use formally guided simulation to effectively design real-world tests?

First use of formal methods for scenario-based testing of AI-based autonomy in both simulation and real world

Fremont, Kim, Pant, Seshia, Acharya, Brusio, Wells, Lemke, Lu, Mehta, **“Formal Scenario-Based Testing of Autonomous Vehicles: From Simulation to the Real World”**, Arxiv e-prints, <https://arxiv.org/abs/2003.07739> [ITSC 2020]

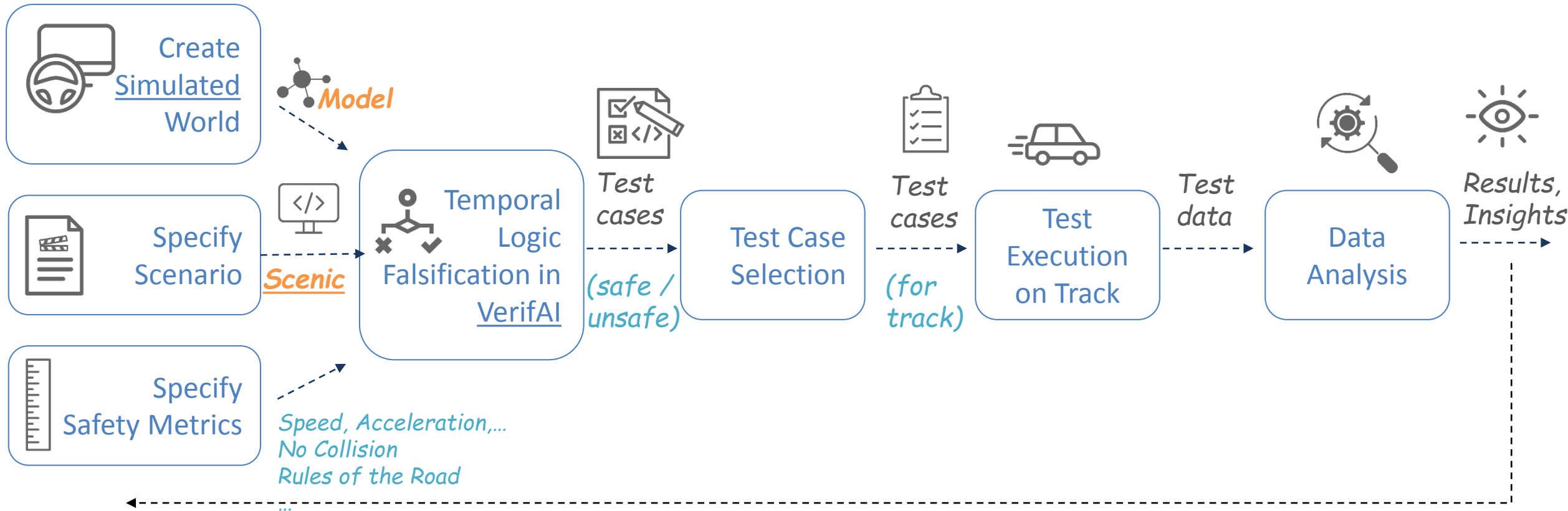


Owned and Operated by AAA NCNU



Simulation software to accelerate safe autonomous vehicle development

Formal Scenario-Based Testing (with Scenic and VerifAI)



Source: Fremont et al., "Formal Scenario-Based Testing of Autonomous Vehicles: From Simulation to the Real World", Intelligent Transportation Systems Conference (ITSC), September 2020. <https://arxiv.org/abs/2003.07739>

Scenario Overview: Focus on Vulnerable Road Users (VRUs)

+53%



Pedestrian fatalities: 53% increase in the last decade (2009-2019)
2019: ~6500 (estimated)

17%



Of all traffic fatalities, 17% are **Pedestrians**

67%



Fatalities at **night** (low-light, limited vision environment)

Source:

GHSA: https://www.thecarconnection.com/news/1127308_pedestrian-deaths-reach-30-year-high-in-2019

IIHS: <https://www.iihs.org/topics/pedestrians-and-bicyclists>

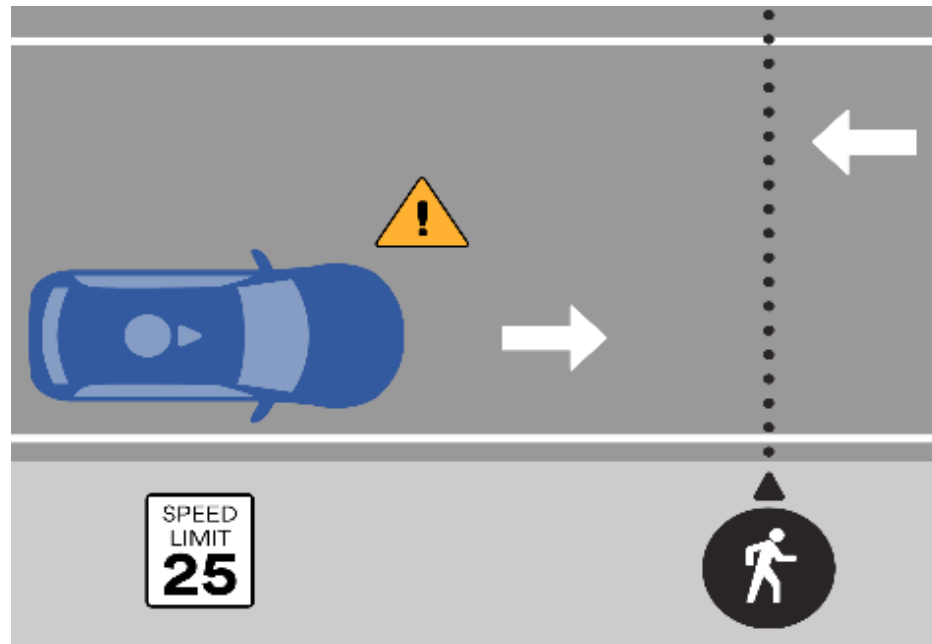
Test Equipment and Use at AAA GoMentum Testing Grounds

Robotic platform for Test Targets



Scenario Execution

[Shows EuroNCAP VRU AEB]



Scenario Evaluation

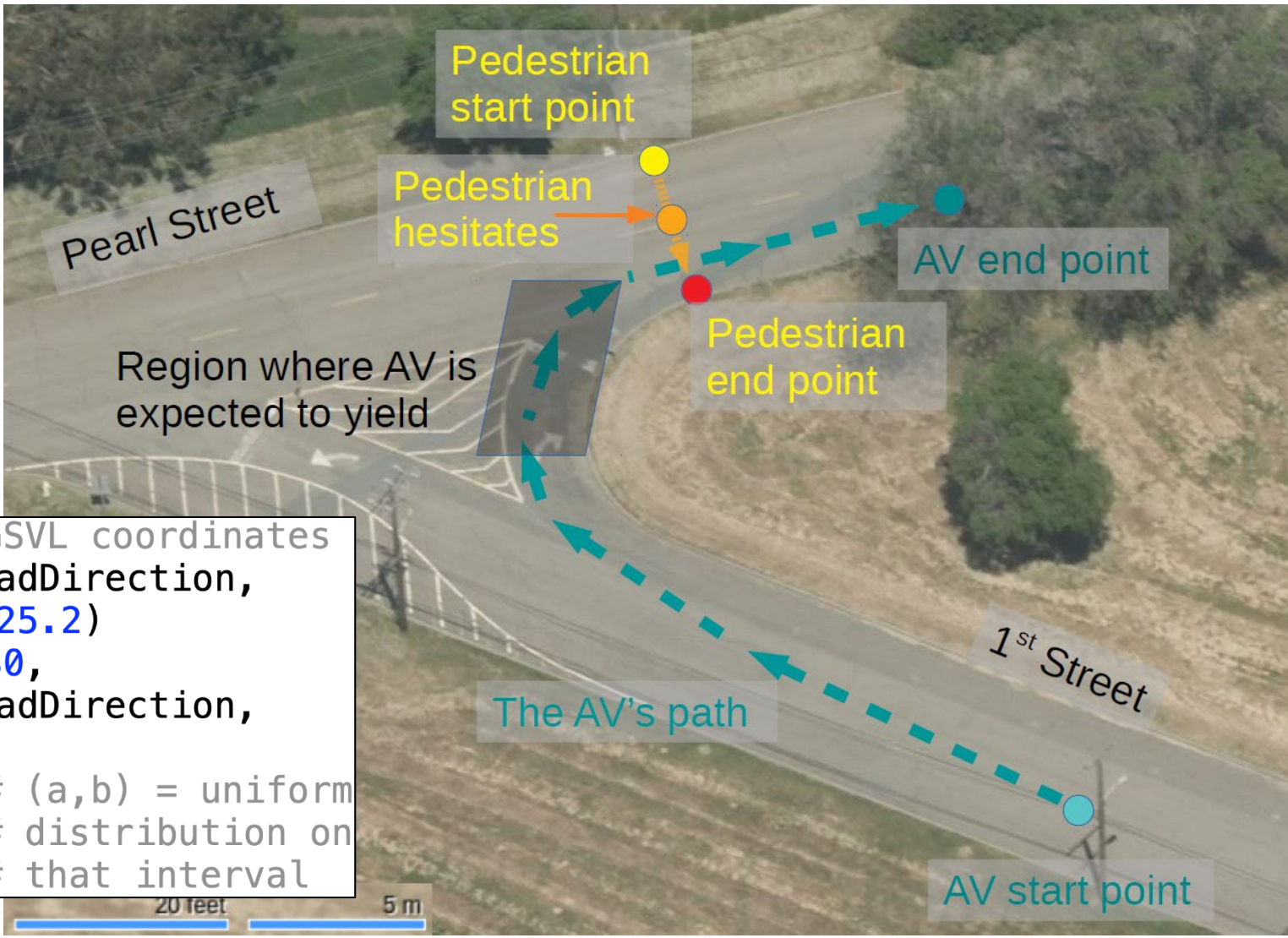
Object & Event Detection/Response: Metrics & Evaluation

- Object detection
- Time to collision
- Separation distance
- Deceleration profile
- Autonomy
- Disengagement

Example Scenario: AV making right turn, pedestrian crossing



Lincoln MKZ running Apollo 3.5



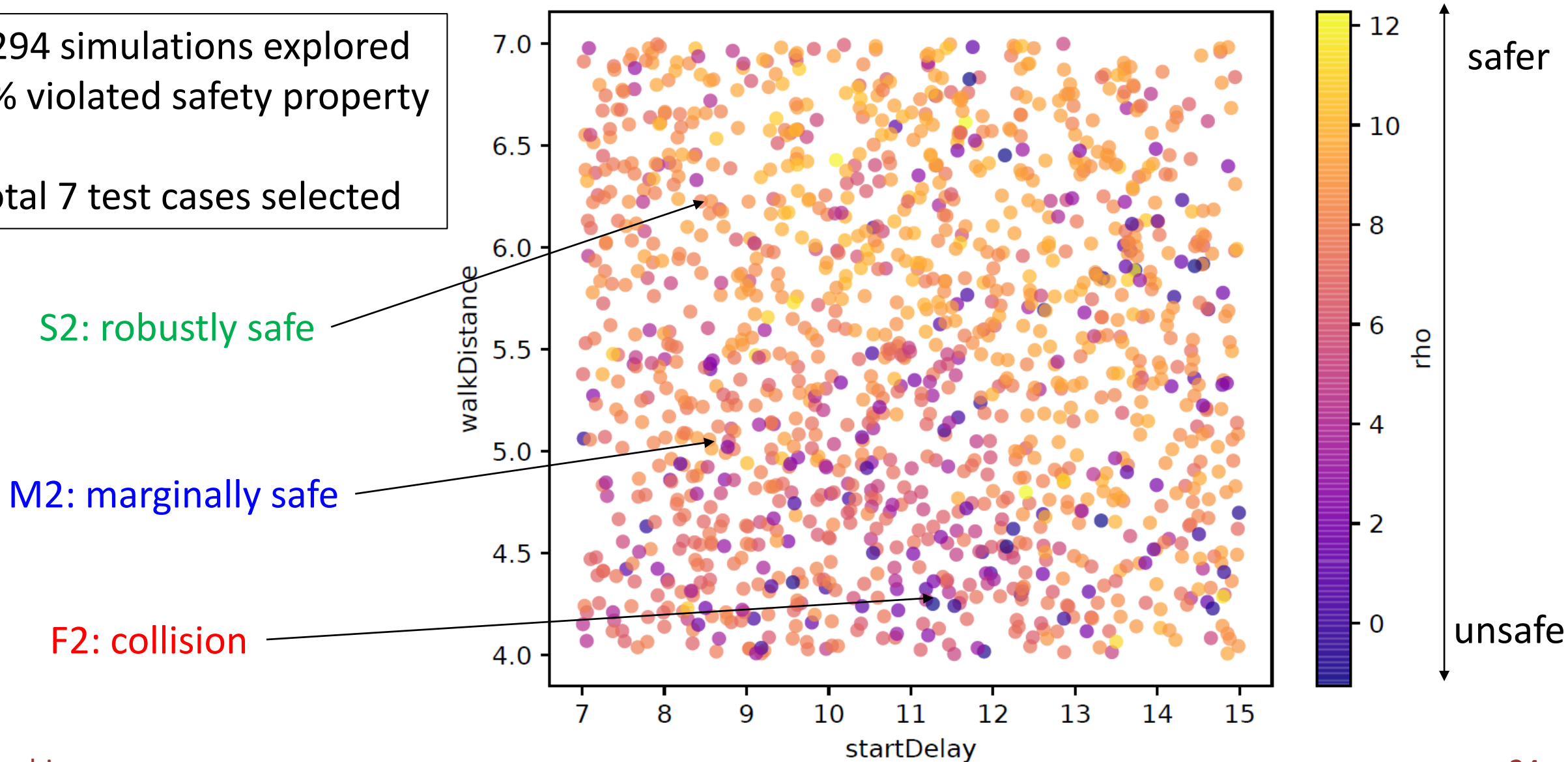
```

ego = EgoCar at 38.6 @ 183.9, # LGSVL coordinates
    facing 10 deg relative to roadDirection,
    with behavior DriveTo(40 @ 225.2)
ped = Pedestrian at 19.782 @ 225.680,
    facing 90 deg relative to roadDirection,
    with behavior Hesitate,
    with startDelay (7, 15), # (a,b) = uniform
    with walkDistance (4, 7), # distribution on
    with hesitateTime (1, 3) # that interval
    
```

Snippet of Scenic program

Results: Falsification and Test Selection

1294 simulations explored
2% violated safety property
Total 7 test cases selected



Results: Does Safety in Simulation → Safety on the Road?

Unsafe in simulation → unsafe on the road: **62.5% (incl. collision)**

Safe in simulation → safe on the road: **93.5% (no collision)**



Results: Why did the AV Fail?

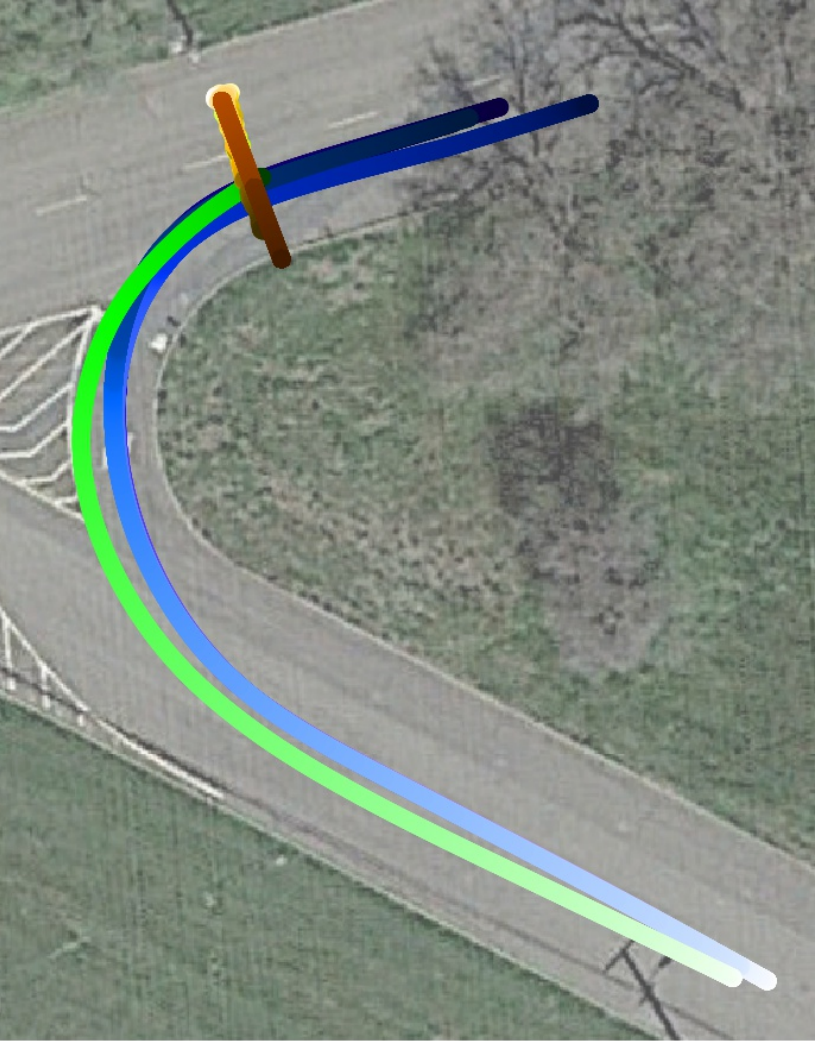
Perception Failure: Apollo 3.5 lost track of the pedestrian several times

The screenshot displays the Apollo 3.5 simulation environment. The top navigation bar includes the 'apollo' logo and several control buttons: 'Docked Version', 'Co-Driver', 'Mute', 'Mkz Standard Debug', '-- vehicle --', and 'Gomentum'. The main simulation area shows a 3D view of a vehicle on a road with various sensors and data overlays. On the right side, there are gauges for 'Brake' and 'Accelerator' at 0%, and a 'NO SIGNAL UNKNOWN' indicator. The bottom of the screen features a 'Console' panel with error messages: 'You haven't selected a vehicle yet!'.

Results: How well do the trajectories match?



S1 Run 2



F1 Run 1

Green – AV real

Blue – AV sim

Orange – Ped real

Yellow – Ped sim

Other Contributions



Open-Source Verified AI Toolkit
(VerifAI & Scenic, on github)

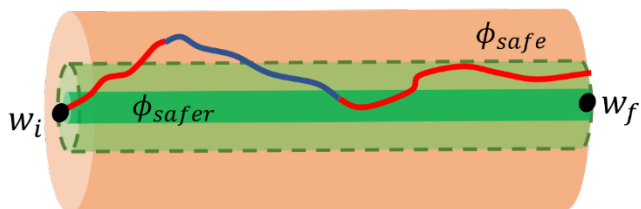


Verified Human-Robot Collaboration

Learning Specifications from Demonstrations, Interaction-Aware Control, etc. [IROS 2016, NeurIPS 2018, CAV 2020]

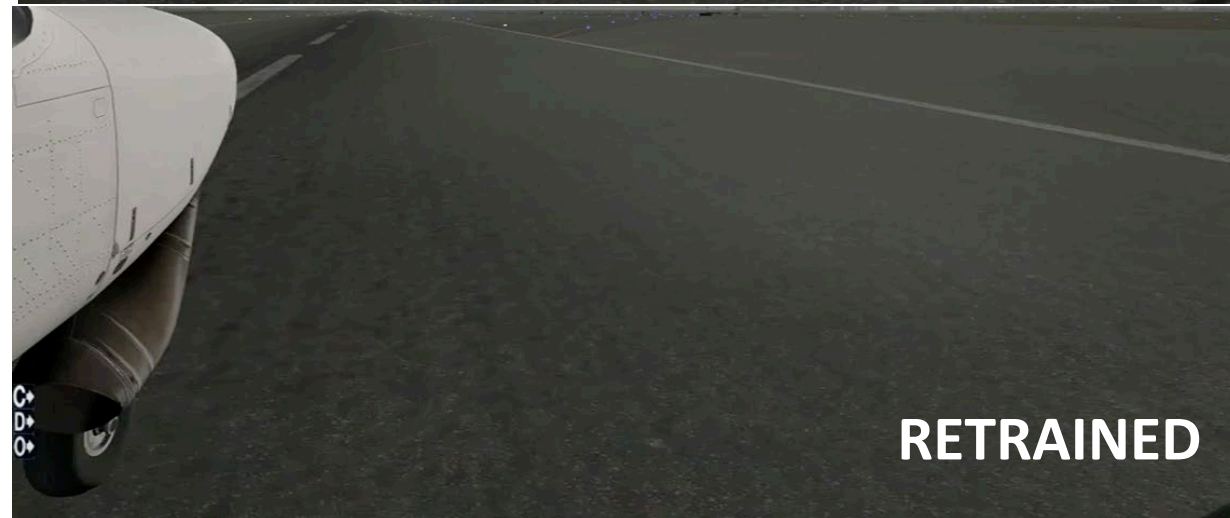
Run-Time Assurance

SOTER framework based on Simplex architecture [DSN 2019]



Counterexample-Guided Retraining

Boeing research automated taxiing system
Used Scenic and VerifAI to find failures and retrain to eliminate [CAV 2020]



Conclusion: Towards Verified AI/ML based Autonomy

Challenges

Core Principles

1. Environment (incl. Human) Modeling	→	Data-Driven, Introspective, Probabilistic Modeling
2. Specification	→	Start with System-Level Specification, then Component Spec (robustness, ...)
3. Learning Systems Complexity	→	Abstraction, Semantic Representation, and Explanations
4. Efficient Training, Testing, Verification	→	Compositional Analysis and Semantics-directed Search/Training
5. Design for Correctness	→	Oracle-Guided Inductive Synthesis; Run-Time Assurance

Thank you!

S. A. Seshia, D. Sadigh, S. S. Sastry. *Towards Verified Artificial Intelligence*.

July 2016. <https://arxiv.org/abs/1606.08514>.