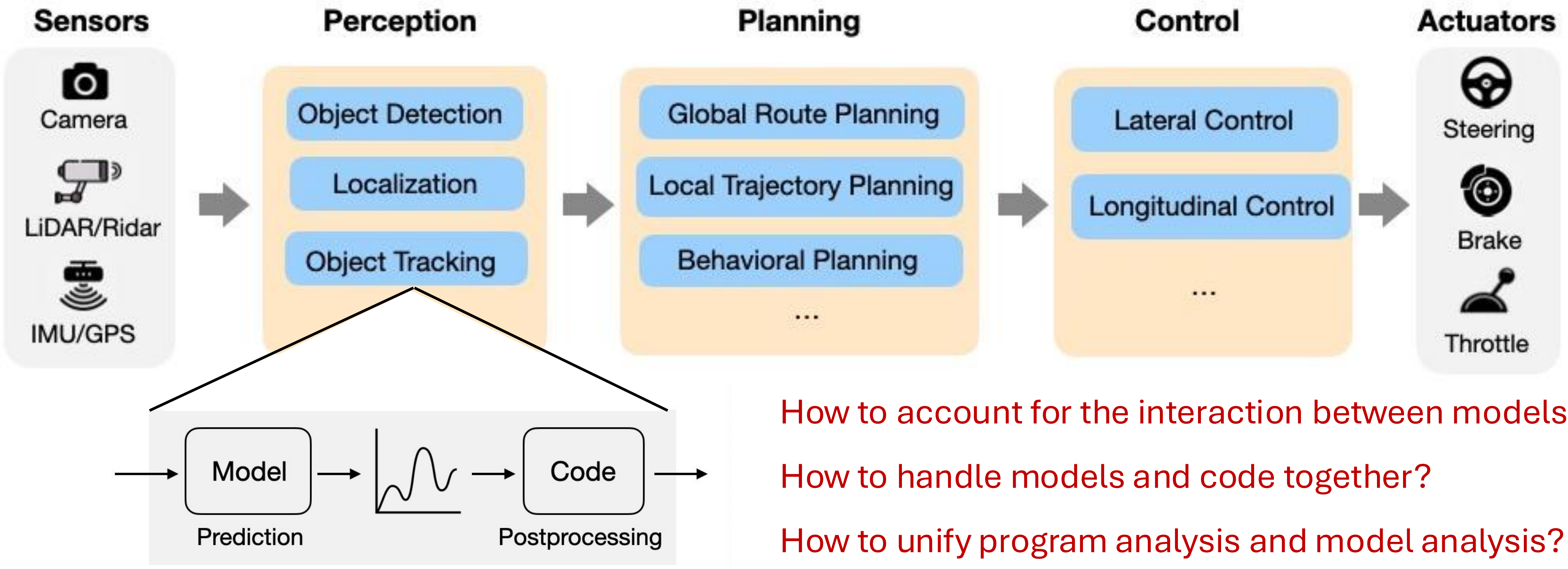
Testing and Debugging Multi-module Autonomous Vehicles in Near-Collision Traffic Scenarios

Tianyi Zhang

Purdue University

Multi-module Autonomous Vehicle Systems



How to account for the interaction between models?

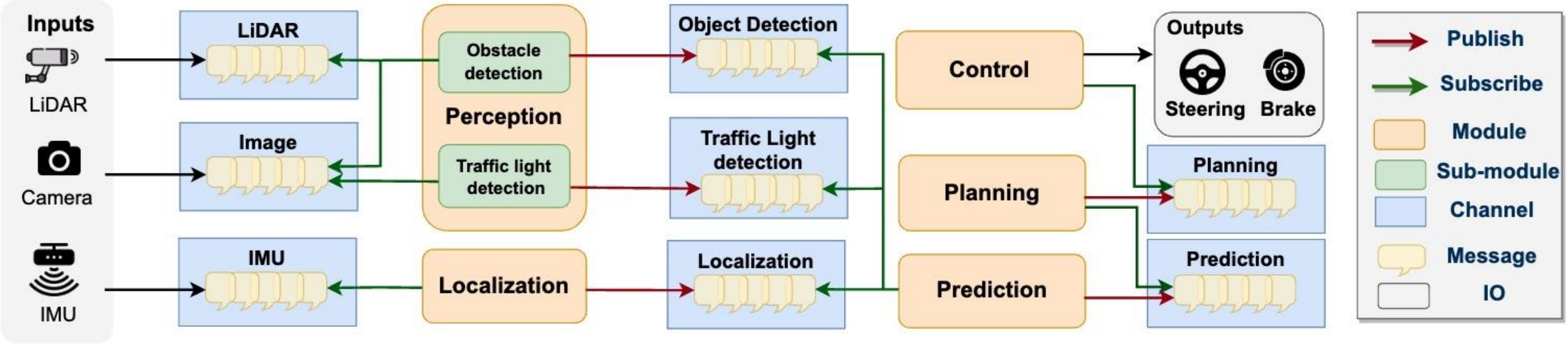
How to handle models and code together?

How to unify program analysis and model analysis?

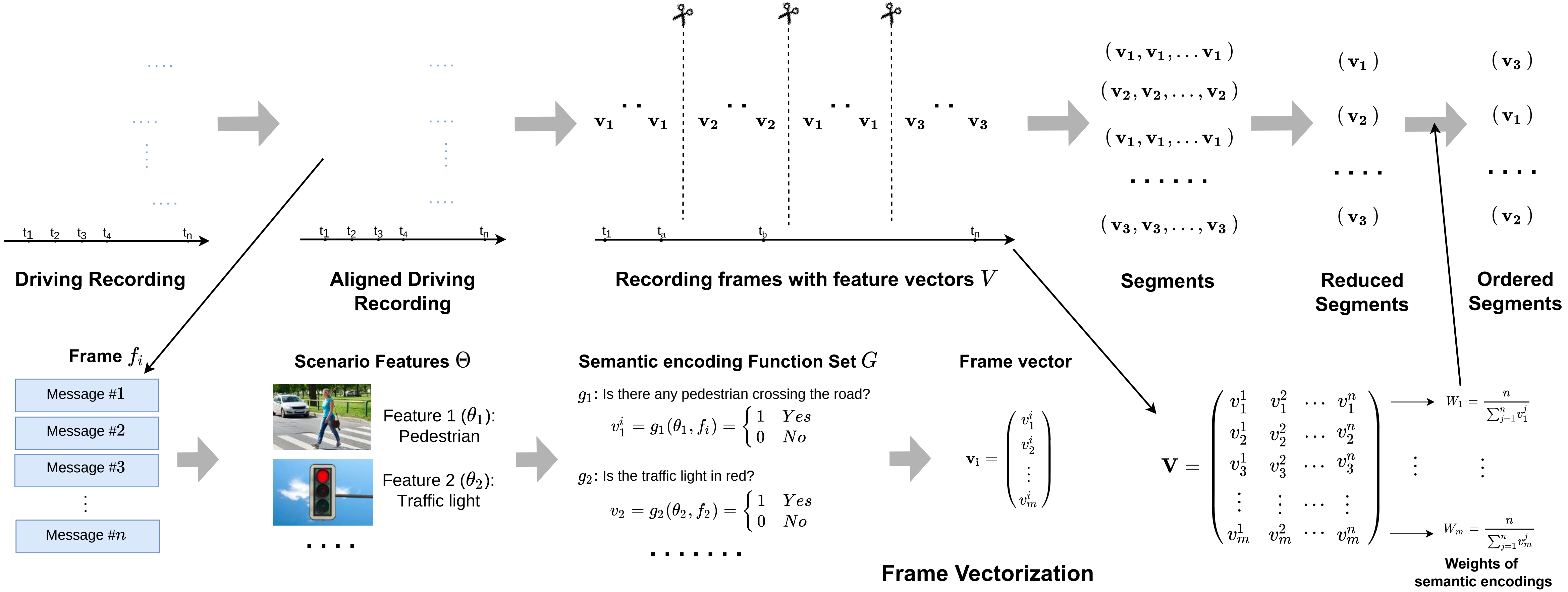
e.g., check the confidence of the lane detection model

System-level Abstraction

- Leverage the publish-subscribe communication mechanism in ROS to derive a system-level abstraction



Scenario-based Test Reduction

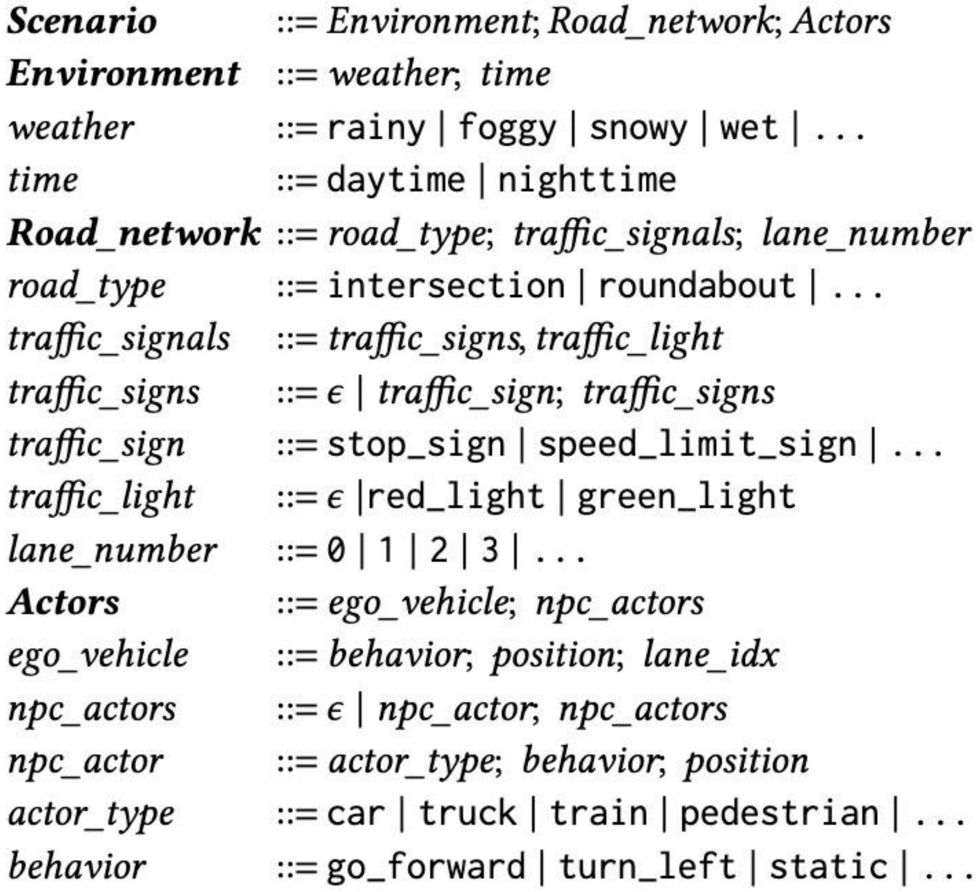


t_1 t_2 t_3 t_4 t_n
Driving Recording
Aligned Driving Recording
$\mathbf{v_1}$ .. $\mathbf{v_1}$ $\mathbf{v_2}$ .. $\mathbf{v_2}$ $\mathbf{v_1}$ .. $\mathbf{v_1}$ $\mathbf{v_3}$ .. $\mathbf{v_3}$
t_1 t_a t_b t_n
Recording frames with feature vectors V
$(\mathbf{v_1}, \mathbf{v_1}, \ldots \mathbf{v_1})$
$(\mathbf{v_2}, \mathbf{v_2}, \ldots, \mathbf{v_2})$
$(\mathbf{v_1}, \mathbf{v_1}, \ldots \mathbf{v_1})$
......
$(\mathbf{v_3}, \mathbf{v_3}, \ldots, \mathbf{v_3})$
Segments
$(\mathbf{v_1})$ $(\mathbf{v_2})$ .... $(\mathbf{v_3})$
Reduced Segments
$(\mathbf{v_3})$ $(\mathbf{v_1})$ .... $(\mathbf{v_2})$
Ordered Segments
Frame f_i
Message #1
Message #2
Message #3
Message #n
Scenario Features Θ
Feature 1 (θ_1): Pedestrian
Feature 2 (θ_2): Traffic light
....
Semantic encoding Function Set G
g_1: Is there any pedestrian crossing the road?
$v_1^i = g_1(\theta_1, f_i) = \begin{cases} 1 & Yes \\ 0 & No \end{cases}$
g_2: Is the traffic light in red?
$v_2 = g_2(\theta_2, f_2) = \begin{cases} 1 & Yes \\ 0 & No \end{cases}$
....
Frame vector
$\mathbf{v_i} = \begin{pmatrix} v_1^i \\ v_2^i \\ \vdots \\ v_m^i \end{pmatrix}$
Frame Vectorization
$\mathbf{V} = \begin{pmatrix} v_1^1 & v_1^2 & \cdots & v_1^n \\ v_2^1 & v_2^2 & \cdots & v_2^n \\ v_3^1 & v_3^2 & \cdots & v_3^n \\ \vdots & \vdots & \cdots & \vdots \\ v_m^1 & v_m^2 & \cdots & v_m^n \end{pmatrix}$
$W_1 = \frac{n}{\sum_{j=1}^n v_1^j}$
$W_m = \frac{n}{\sum_{j=1}^n v_m^j}$
Weights of semantic encodings

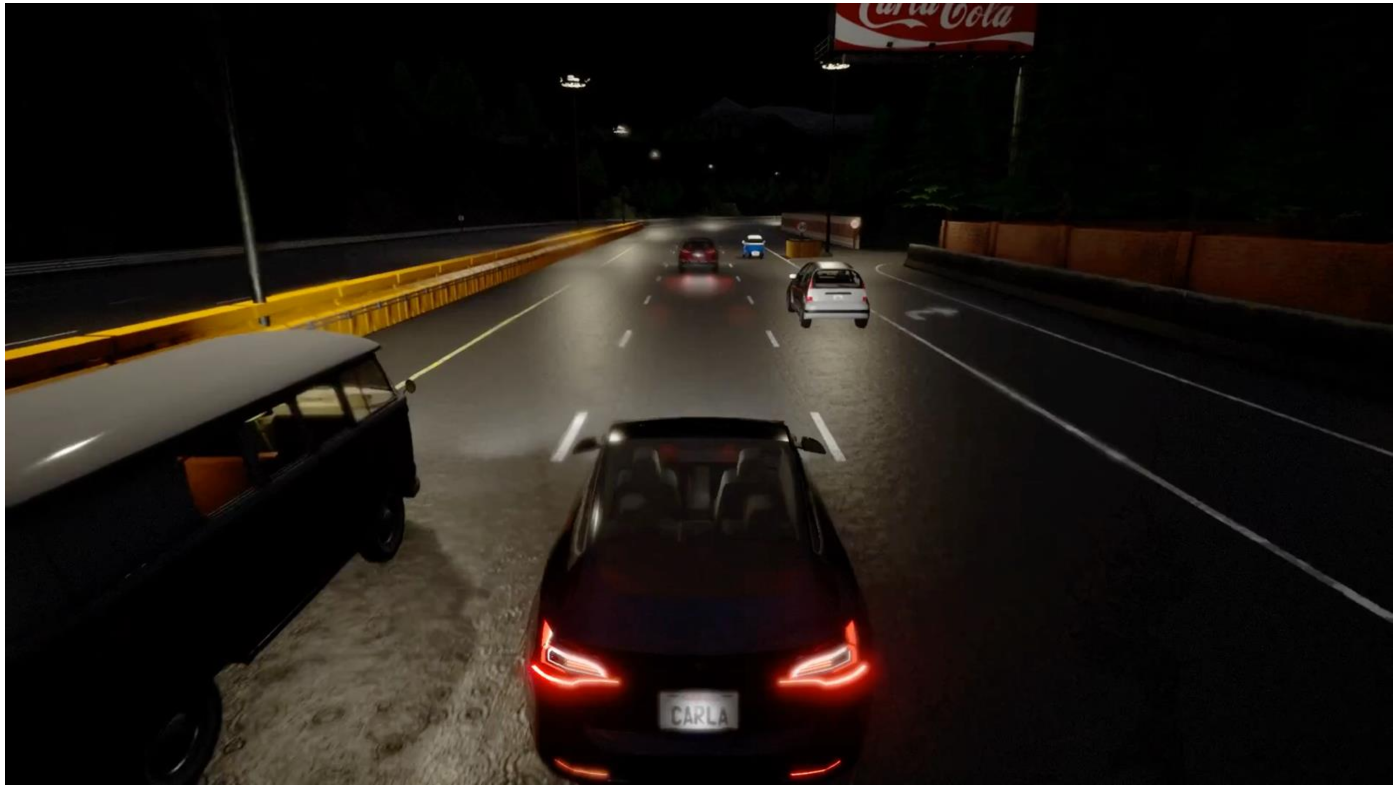
34%~77% reduction!

Deng et al. Scenario-based test reduction and prioritization for multi-module autonomous driving systems. FSE 2022.

Scenario-based Test Generation



```
Scenario       ::= Environment; Road_network; Actors
Environment    ::= weather; time
weather        ::= rainy | foggy | snowy | wet | . . .
time           ::= daytime | nighttime
Road_network   ::= road_type; traffic_signals; lane_number
road_type      ::= intersection | roundabout | . . .
traffic_signals ::= traffic_signs, traffic_light
traffic_signs  ::= ϵ | traffic_sign; traffic_signs
traffic_sign   ::= stop_sign | speed_limit_sign | . . .
traffic_light  ::= ϵ |red_light | green_light
lane_number    ::= 0 | 1 | 2 | 3 | . . .
Actors         ::= ego_vehicle; npc_actors
ego_vehicle    ::= behavior; position; lane_idx
npc_actors     ::= ϵ | npc_actor; npc_actors
npc_actor      ::= actor_type; behavior; position
actor_type     ::= car | truck | train | pedestrian | . . .
behavior       ::= go_forward | turn_left | static | . . .
```

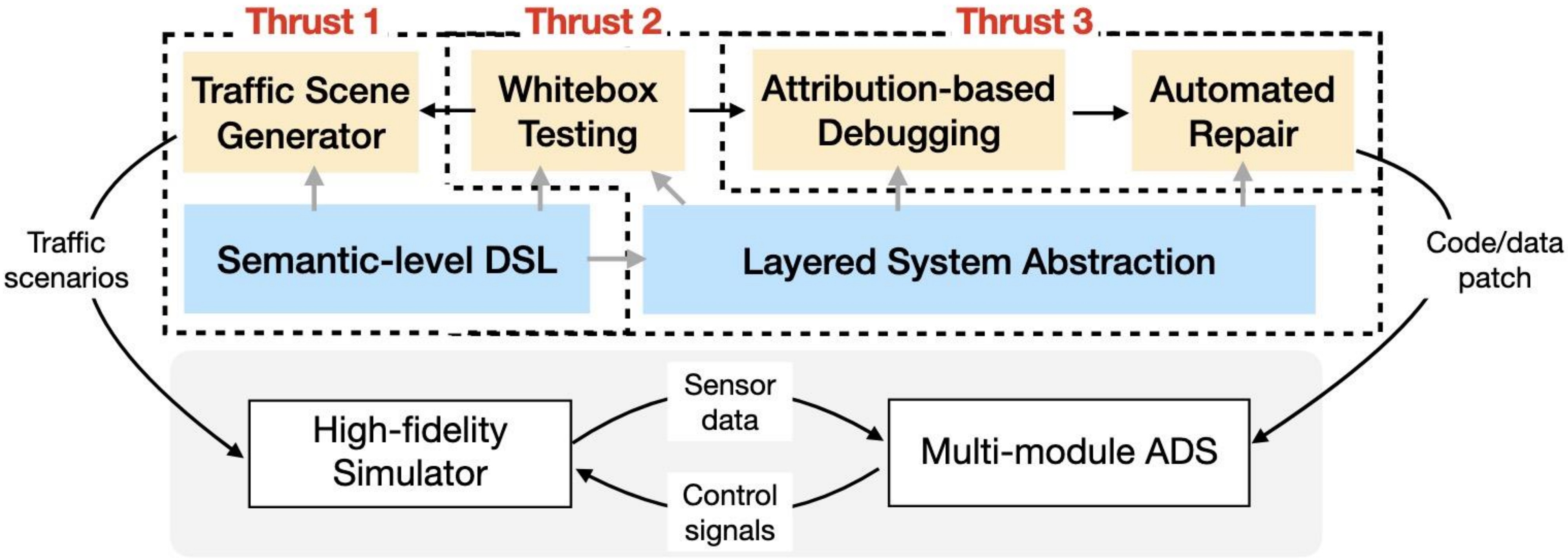


A DSL for traffic scenario specification

A generated traffic of exiting the highway

Tu and Zhang. Multi-Modal Traffic Scenario Generation for Autonomous Driving System Testing. 2025.

Overview of Our SLES Project


Thrust 1
Thrust 2
Thrust 3
Traffic Scene Generator
Whitebox Testing
Attribution-based Debugging
Automated Repair
Semantic-level DSL
Layered System Abstraction
Traffic scenarios
Code/data patch
High-fidelity Simulator
Sensor data
Control signals
Multi-module ADS