DriveFuzz:
Discovering Autonomous Driving Bugs through Driving Quality-Guided Fuzzing

Seulbae Kim[1]   Major Liu[2]   Junghwan Rhee[3]

Yuseok Jeon[4]   Yonghwi Kwon[5]   Chung Hwan Kim[2]

[1] Georgia Institute of Technology, [2] University of Texas at Dallas,
[3] University of Central Oklahoma, [4] UNIST, [5] University of Virginia
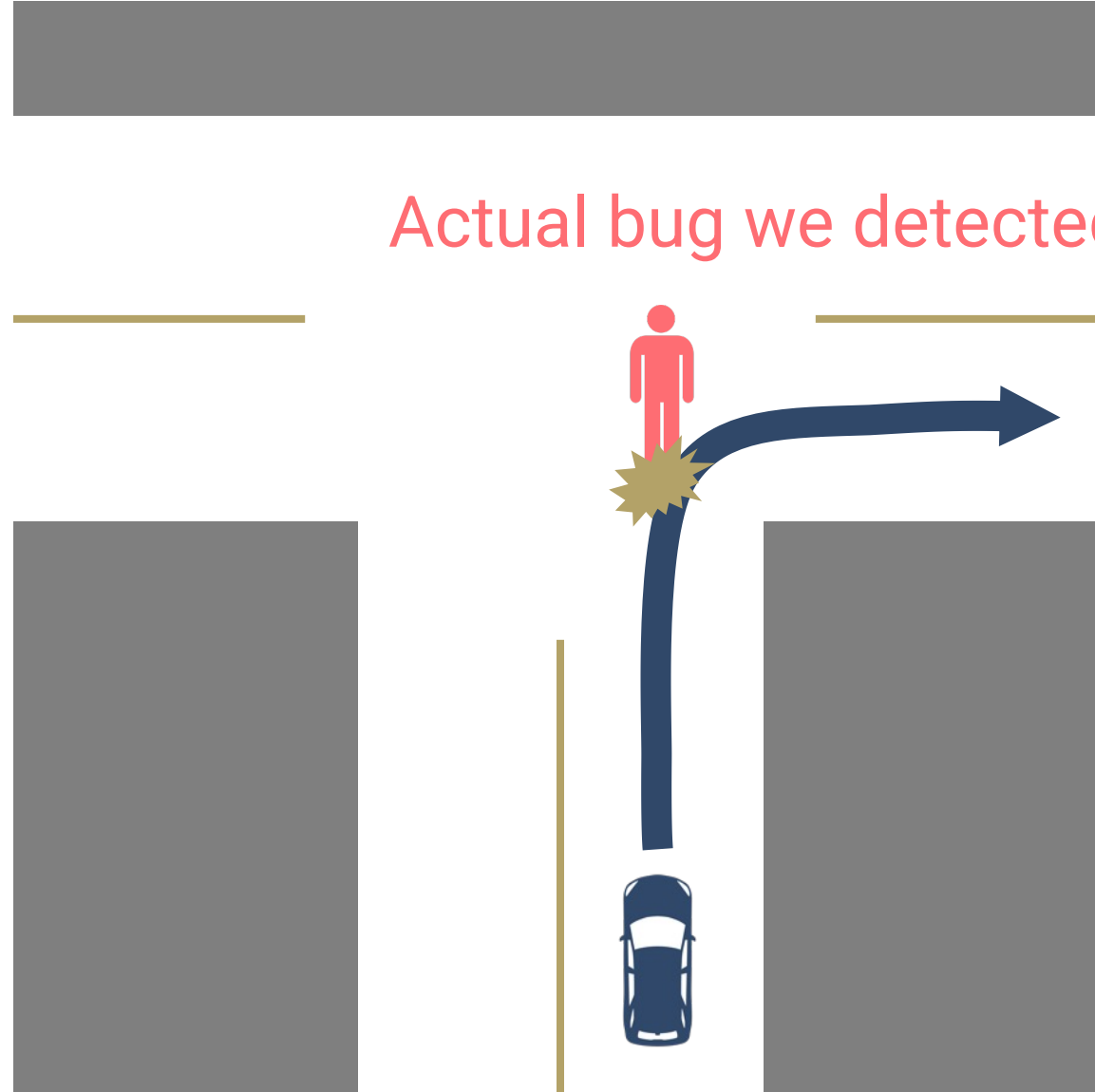
Can we trust autonomous driving systems?

- Expectation

  vs

  Reality

Actual bug we detected!

Can we trust autonomous driving systems?

- Fatal autopilot accidents continue

EDITORS' PICK | 17,238 views | Jun 2, 2020, 06:20pm EDT

Tesla In Taiwan Crashes Directly Into Overturned Truck, Ignores Pedestrian, With Autopilot On

Business > Technology • News

Tesla on 'Autopilot' hits police vehicle which hits ambulance, driver possibly drunk: police

Tesla has said its automated system can be safely used by attentive drivers

A crash, allegedly involving Tesla's controversial 'Autopilot' driver-assistance system, between a Tesla and a police vehicle on July 14, 2020 in Arizona (Arizona Department of Public Safety)

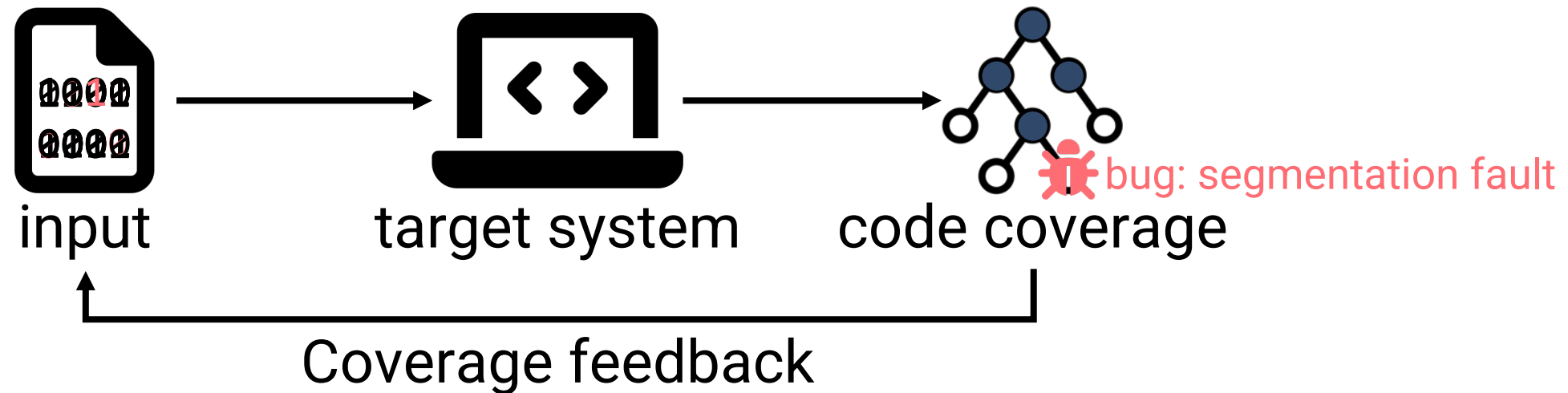# Finding bugs via manual testing

Source: "Will Tesla Autopilot hit a dog, human, or traffic cone?"
– Youtube Lowlifemike

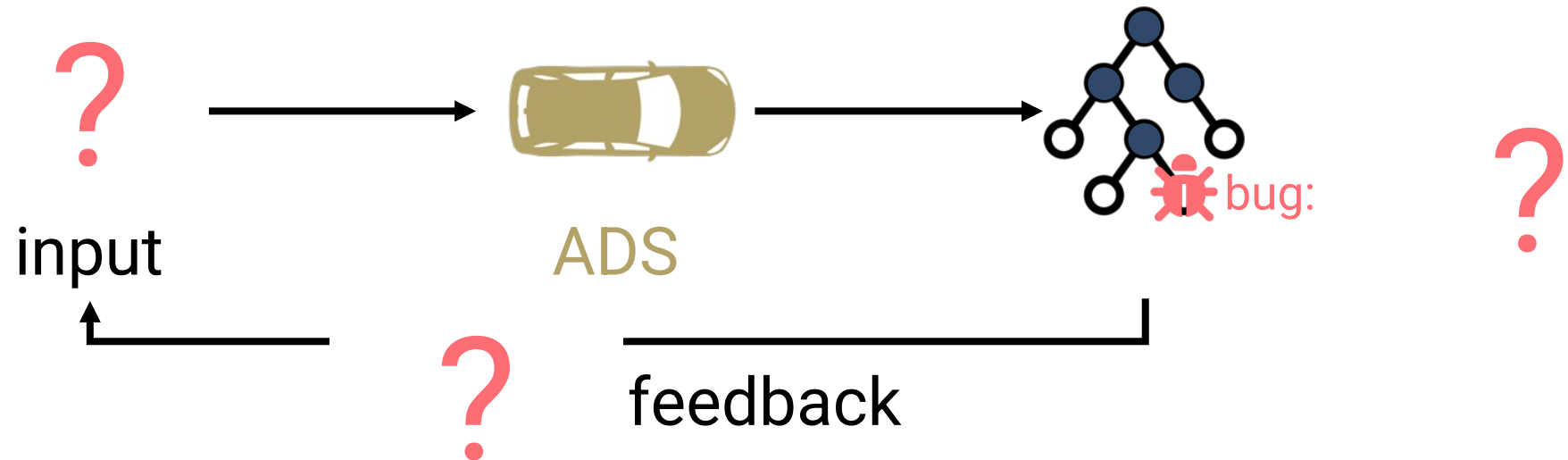Source: "Will a Tesla KILL a cat?"
– Youtube Carwow

4

Finding bugs via automated testing

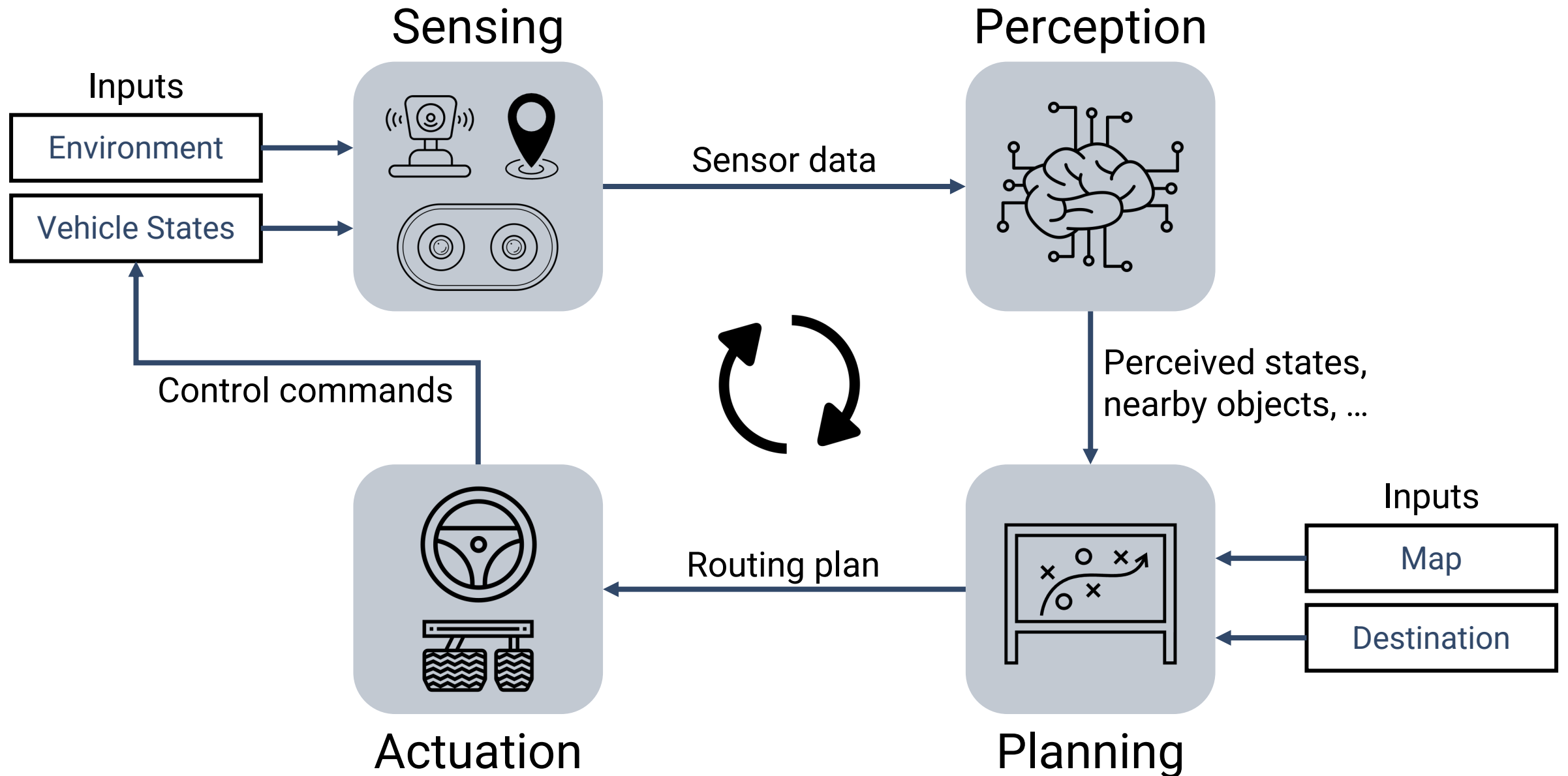- Feedback-driven fuzzing for traditional software

input   target system   code coverage

bug: segmentation fault

Coverage feedback

Finding bugs via automated testing

- Feedback-driven fuzzing for autonomous driving systems?

input   ADS   bug:

feedback

Layers and workflow of Autonomous Driving System (ADS)

Sensing

Perception

Inputs

Environment

Vehicle States

Sensor data

Perceived states, nearby objects, ...

Control commands

Actuation

Routing plan

Planning

Inputs

Map

Destination

Considerations in designing test inputs

Sensing

Perception

Inputs

Environment

Sensor data

The test input should not be a snapshot

Control commands

Perceived states,
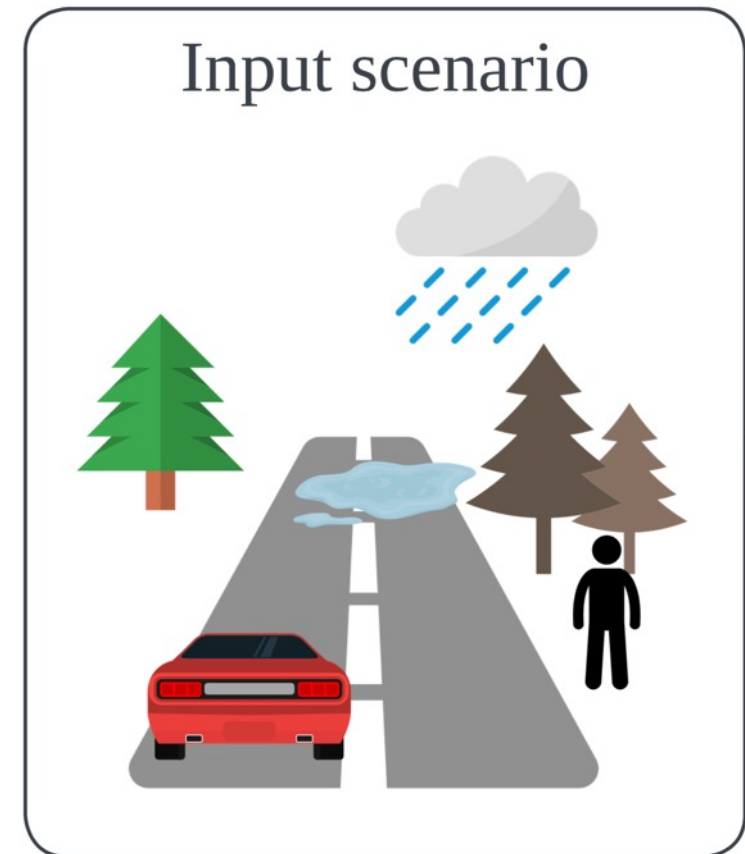nearby objects, …

The test input should be able to stress all layers
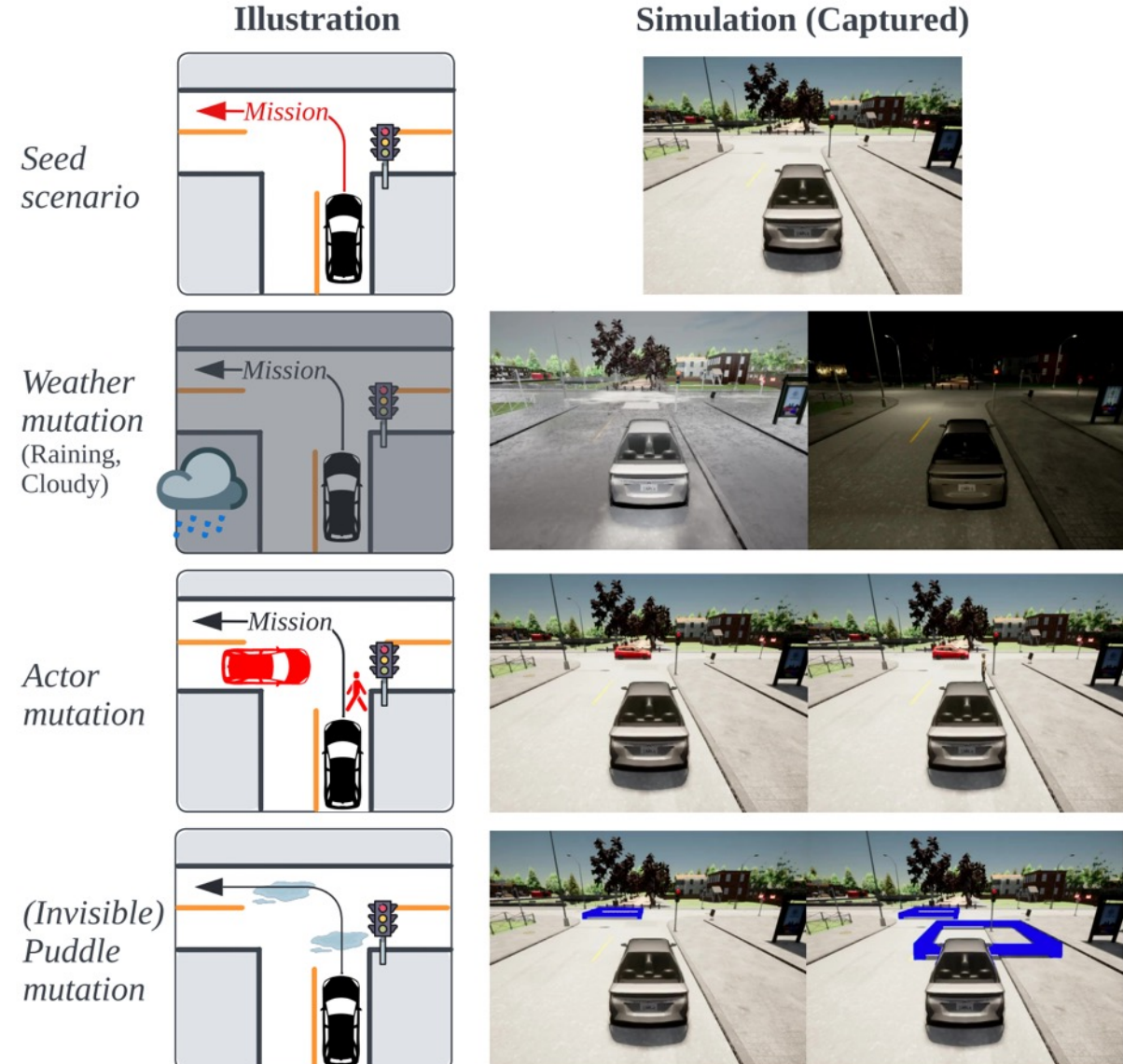
Destination

Actuation

Planning
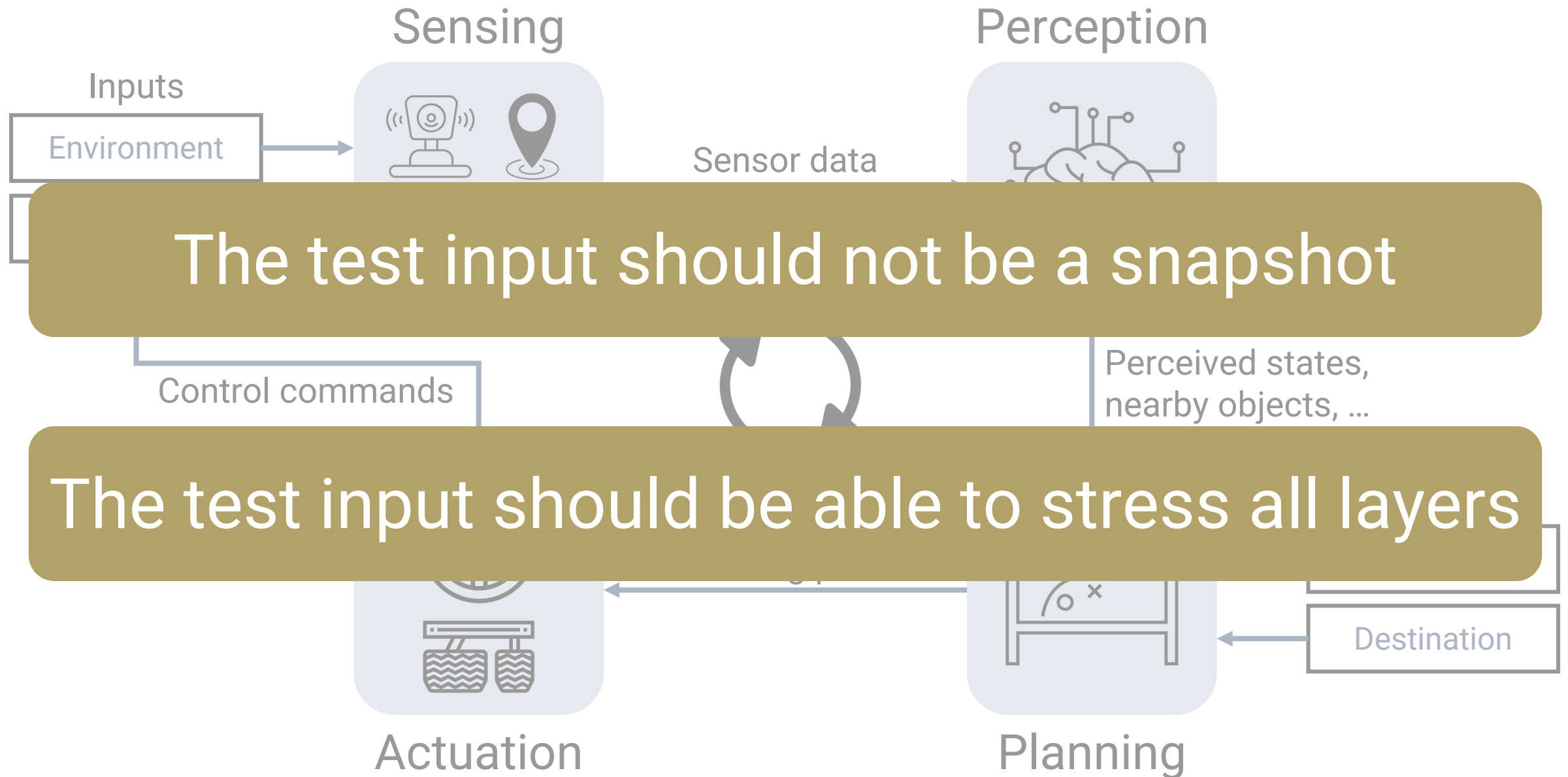
Our input space: Driving scenarios

- Representing temporal and spatial domains of real world
- Consists of
  - 3D map
  - Mission (initial and goal positions)
  - Actors (vehicles or pedestrians)
  - Puddles (e.g., black ice)
  - Weather conditions

Input scenario

Mutation of driving scenarios

- Map and mission selection
  - stress ADS with diverse environments
- Actor generation & mutation
  - render diverse interactive situations
- Puddle generation & mutation
  - stress planning & actuation layers with frictional diversity
- Weather mutation
  - affect sensing and perception

Illustration   Simulation (Captured)

Seed scenario

Weather mutation (Raining, Cloudy)

Actor mutation

(Invisible) Puddle mutation

# Confining mutation to feasible scenarios

- Ensuring physically valid mutation
  - Spatial constraint
    - Initial positions of all actors and objects are spread away (e.g., 5 m)
    - Preventing unrealistic jams (e.g., vehicles overlapping)
  - Temporal constraint
    - Maximum speed of actor vehicles and pedestrians are conservatively set
    - Preventing reckless behaviors (e.g., a person running into a vehicle too quickly)

- Both constraints are configurable
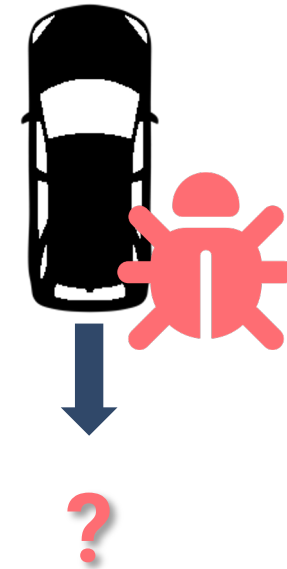
Feedback-driven fuzzing for ADS
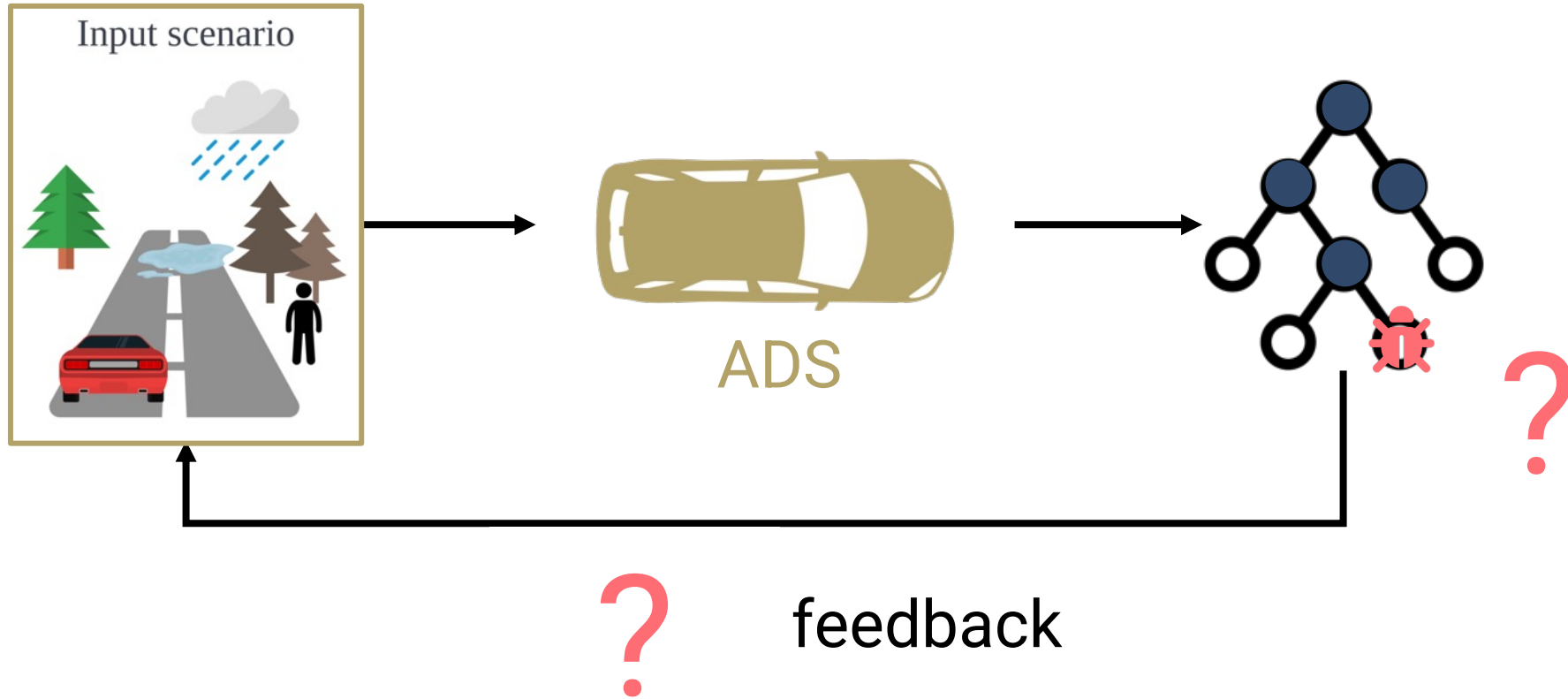
Defining bugs

- What happens to a buggy ADS?
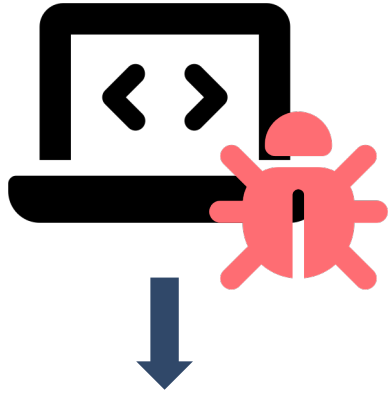
```
$ ./buggy_program
[1]    3541023 segmentation fault  ./buggy_program
```

Classic software bugs
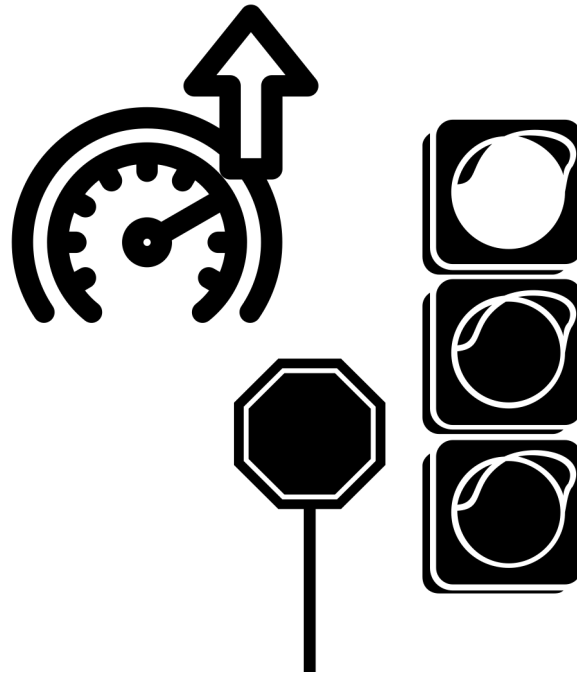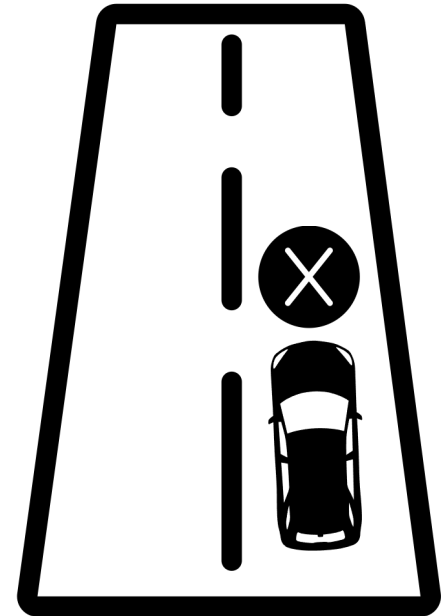
?

Safety-critical vehicular misbehaviors

Collision   Infraction   Immobility

Feedback-driven fuzzing for ADS

Input scenario

ADS
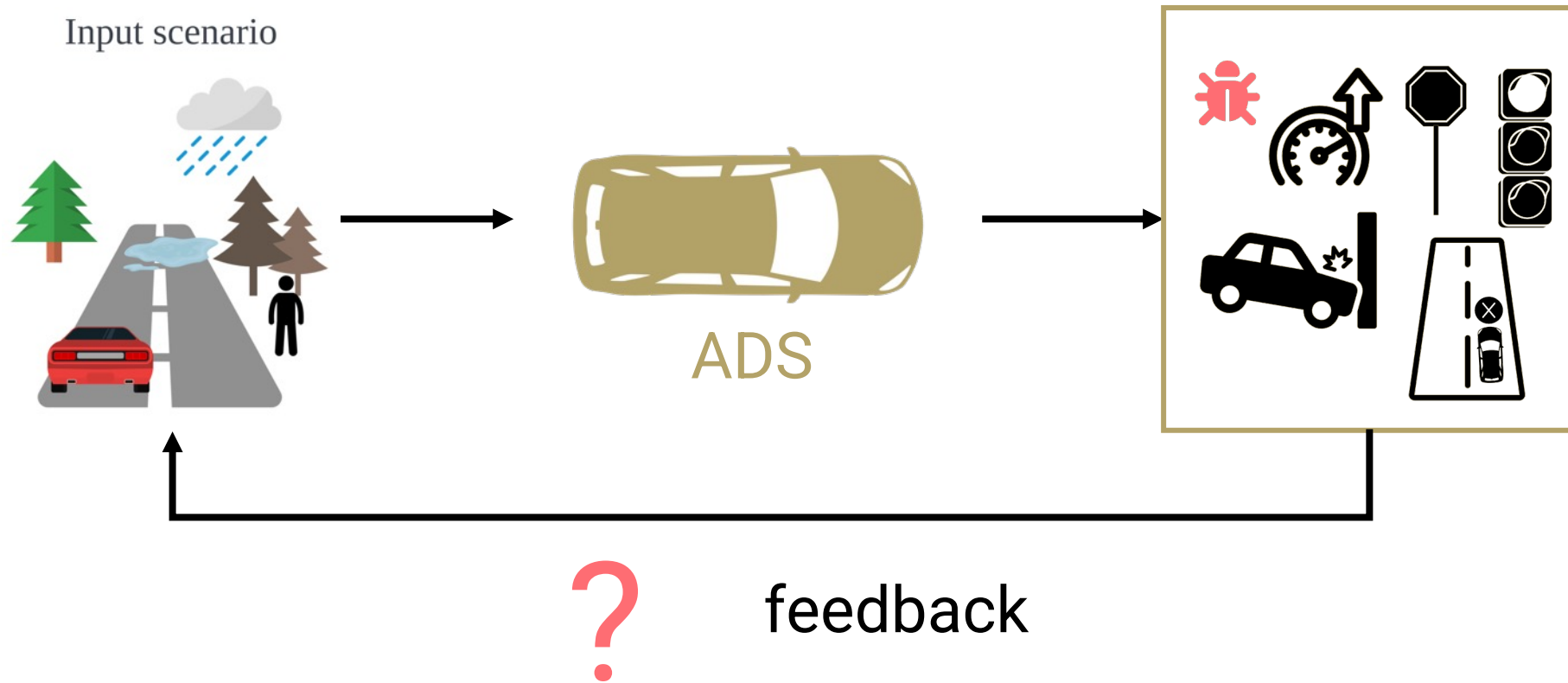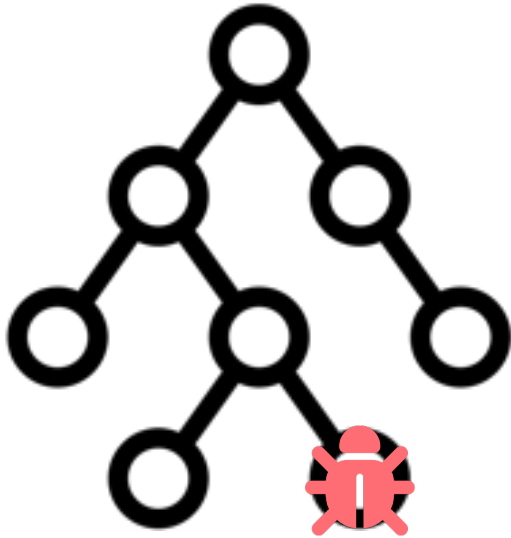
? feedback

A need for a new feedback mechanism

General software programs

- Diverse, linear code paths
- More code paths ≃ more bugs found

Autonomous driving system

Sensing   Perception

Actuation   Planning
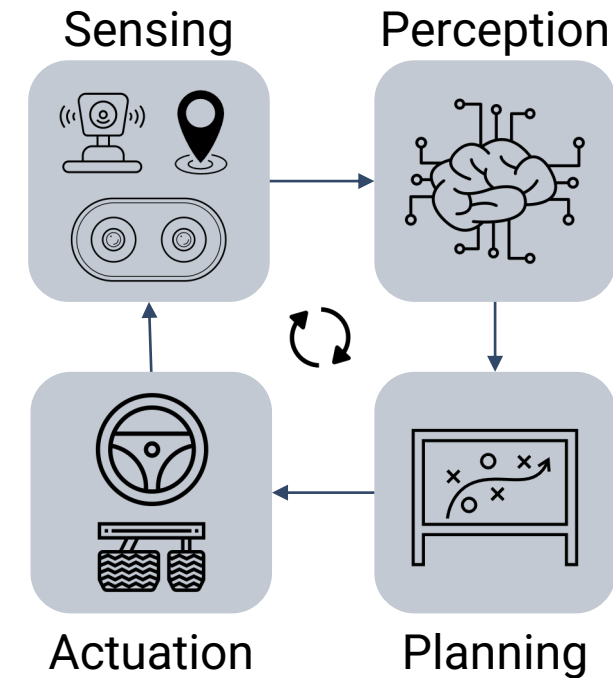
- Distributed system
- Behavior is driven by state changes in a loop, not code paths

A need for a new feedback mechanism

General software programs

Autonomous driving system

Sensing

Perception

Actuation

Planning

Need proper metrics to quantify
the quality of input driving scenarios
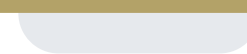
- Diverse, linear code paths
- More code paths ≃ more bugs found

- Distributed system
- Behavior is driven by state changes in a loop, not code paths

Solution: Driving quality feedback

- Intuition
  - Quality of driving $\simeq$ likelihood of misbehaviors

Hard acceleration, braking, and turns	Oversteer and understeer	Minimum distance to other actors

- Metric auto insurance companies use
- #1 cause of motorsport accidents
- Near-missed collisions

DriveFuzz overview

Input scenario

Mutation engine (Section 4.2)
- Mission mutator
- Weather mutator
- Actor mutator
- *Puddle** mutator

Seed pool

Driving quality feedback score
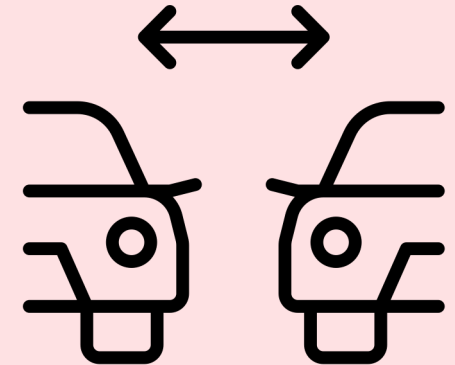
Puddle is invisible (It is visible in the illustrations for presentation)

Mutated scenario

Test executor (Section 4.3)
- Test bridge
- Autonomous driving system (ADS)
- Driving simulator

Vehicle states (position, velocity, acceleration, ...)

Misbehavior detector (Section 4.4)
- Driving test oracles
  - Collison
  - Infraction
  - Immobility

Bug → Bug report

Not a bug

Driving quality feedback engine (Section 4.5)
Quantifying driving quality via vehicle states
1. Hard acceleration/braking
2. Hard turn
3. Over/understeer
4. Minimum distance

DriveFuzz in action

- Seed scenario
  - Map
  - Initial position
  - Destination

DriveFuzz in action

- Round 1

Mutation #1

score: 100

Mutation #2

score: 100

No misbehavior detected

Mutation #3

score: 100

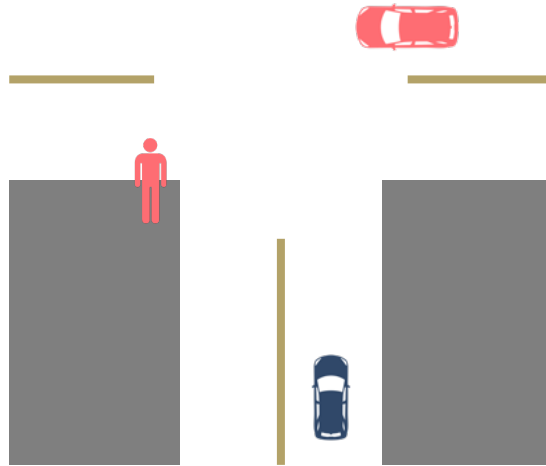Check driving quality scores

Mutation #4

score: 88

SELECT
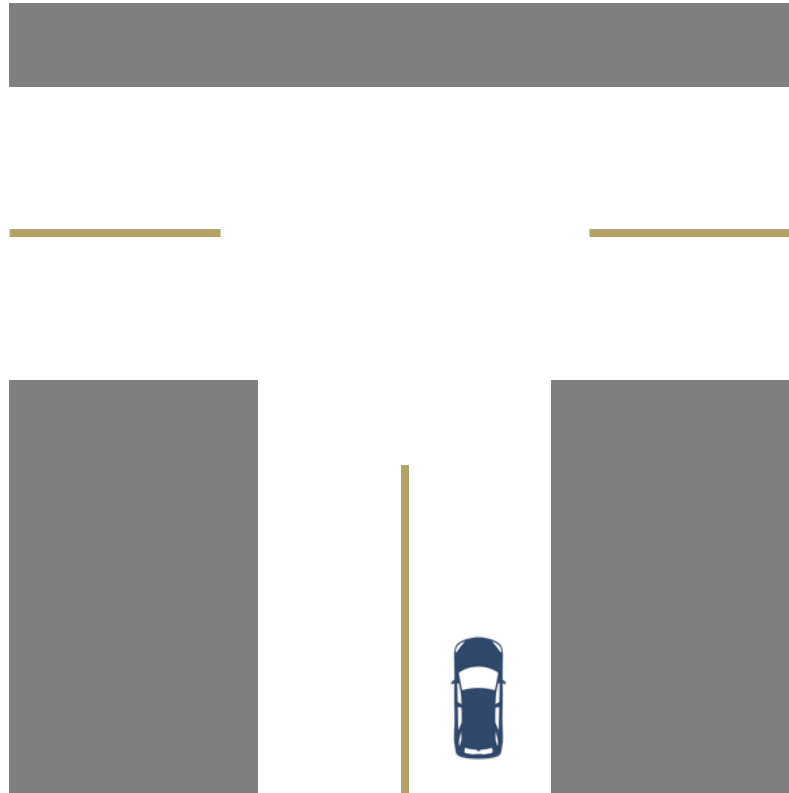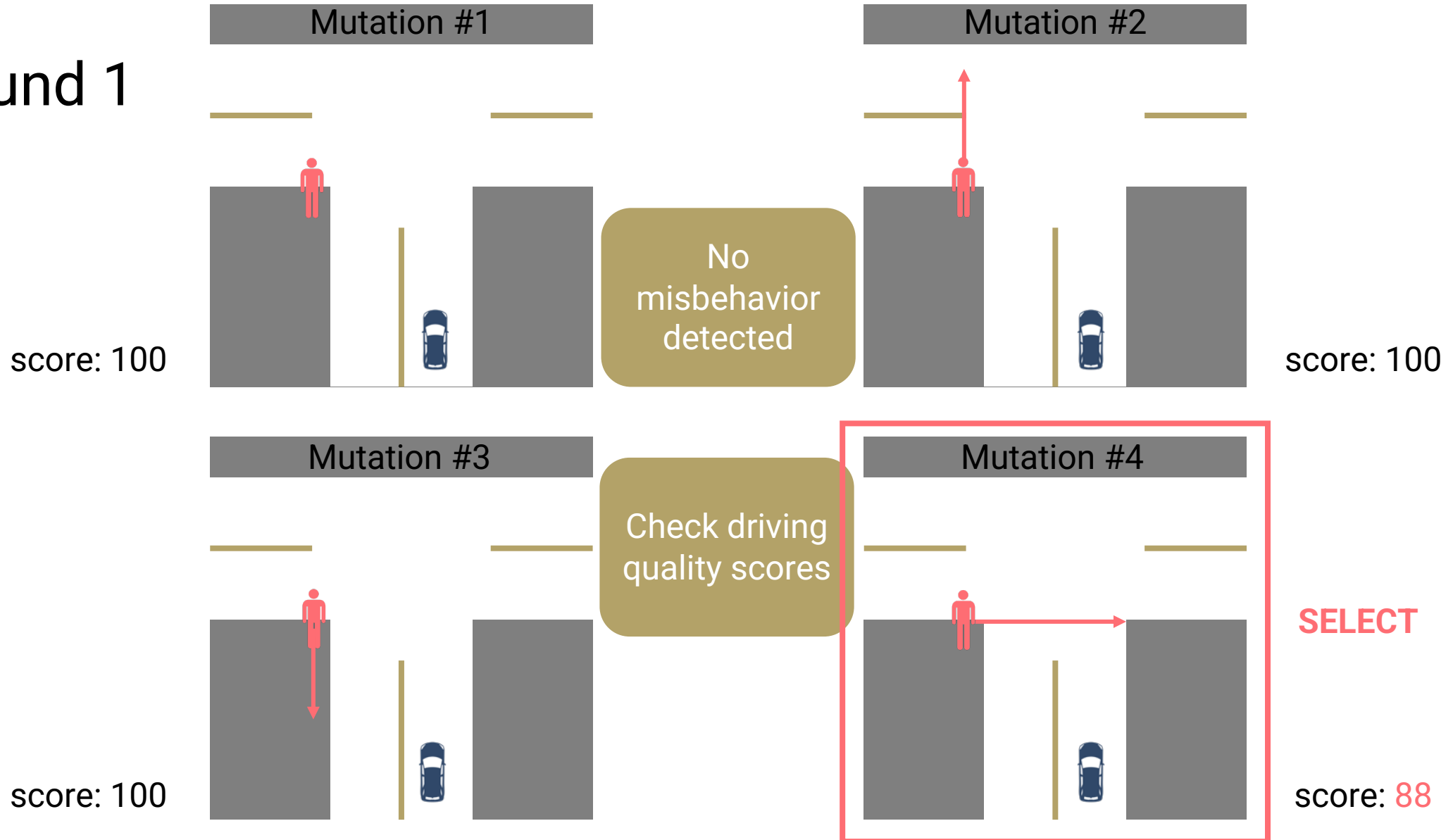
DriveFuzz in action

- Round 2

Mutation #4-1

Mutation #4-2

Misbehavior detected (collision)
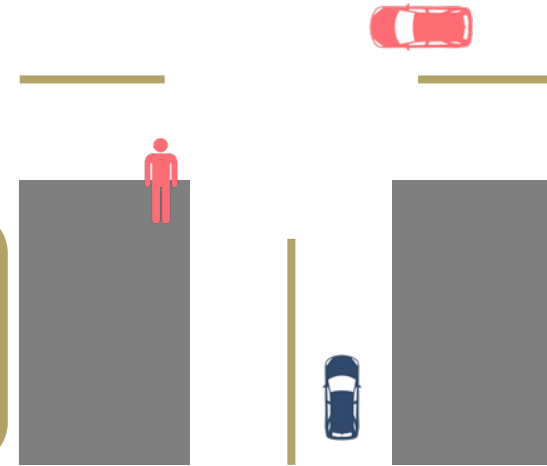
Mutation #4-3
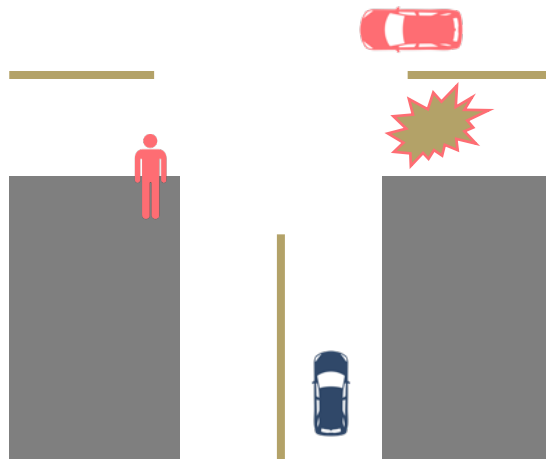
Save states and report

Evaluation

- Targeted two autonomous driving systems
  - Autoware
    - A full-fledged ADS with active development status
    - Internationally adopted by well-known auto manufactures (e.g., BMW)
    - Qualified to run driverless vehicles on public roads in Japan (2017~)
  - Behavior Agent
    - A rudimentary ADS developed by CARLA
    - Implements path planning and feedback-based PID control
    - Complies with traffic laws and avoids collisions

Detected 33 new bugs throughout ADS layers

|	Bug #	Layer	Component	Description	Impact	Strategy	Root cause	ACK
Autoware	01	Sensing	Fusion	LiDAR & camera fusion misses small objects on road	C	all	Logic err	
	02	Perception	Detection	Perceives the road ahead as an obstacle at a steep downhill	I	all	Logic err	✓
	03	Perception	Detection	Fails to semantically tag detected traffic lights and cannot take corresponding actions	C, V	all	Logic err	
	04	Perception	Detection	Fails to semantically tag detected stop signs and cannot take corresponding actions	C, V	all	Logic err	
	05	Perception	Detection	Fails to semantically tag detected speed signs and cannot take corresponding actions	V	all	Logic err	
	06	Perception	Localization	Faulty localization of the base frame while turning	C, L	all	Logic err	✓
	07	Perception	Localization	Localization error when moving underneath bridges and intersections	C, L	all	Logic err	✓
	08	Planning	Global planner	Generates infeasible path if the given goal is unreachable	C, L	all	Logic err	✓
	09	Planning	Global planner	Generates infeasible path if the goal's orientation is not aligned with lane direction	C, I, L	all	Logic err	✓
	10	Planning	Global planner	Global path starts too far from the vehicle's current location	C, I, L	all	Logic err	✓
	11	Planning	Local planner	Target speed keeps increasing at certain roads, overriding the speed configuration	S, C	all	Logic err	✓
	12	Planning	Local planner	Fails to avoid forward collision with a moving object	C	all	Logic err	
	13	Planning	Local planner	Fails to avoid lateral collision (ADS perceives the approaching actor before collision)	C	ent	Not impl	
	14	Planning	Local planner	Fails to avoid rear-end collision (ADS perceives the approaching actor before collision)	C	ent	Not impl	
	15	Planning	Local planner	While turning, ego-vehicle hits an immobile actor partially blocking the intersection	C	ent	Logic err	
	16	Actuation	Pure pursuit	Ego-vehicle keeps moving after reaching the destination	C, L	all	Logic err	✓
	17	Actuation	Pure pursuit	Fails to handle sharp right turns, driving over curbs	C, L	all	Faulty conf	
Behavior Agent	18	Perception	Detection	Indefinitely stops if an actor vehicle is stopped on a sidewalk	I	ent	Logic err	
	19	Perception	Detection	Flawed obstacle detection logic; lateral movement of an object is ignored	C	con	Logic err	
	20	Planning	Global planner	Generates inappropriate trajectory when initial position is given within an intersection	C, L, V	all	Logic err	
	21	Planning	Local planner	Improper lane changing, cutting off and hitting an actor vehicle	C	man	Logic err	
	22	Planning	Local planner	Vehicle indefinitely stops at stop signs as planner treats stop signs as red lights and waits for green	I	all	Logic err	
	23	Planning	Local planner	Vehicle does not preemptively slow down when the speed limit is reduced	S	all	Logic err	
	24	Planning	Local planner	Always stops too far (> 10 m) from the goal due to improper checking of waypoint queue	F	all	Logic err	
	25	Planning	Local planner	Collision prevention does not work at intersections (only checks if actors are on the same lane)	C	all	Logic err	
	26	Planning	Local planner	Fails to avoid lateral collision (ADS perceives the approaching actor before collision)	C	man	Not impl	
	27	Planning	Local planner	Fails to avoid rear-end collision (ADS perceives the approaching actor before collision)	C	man	Not impl	
	28	Planning	Local planner	No dynamic replanning; the vehicle does infeasible maneuvers to go back to missed waypoints	C, L	ins	Not impl	
	29	Actuation	Controller	Keeps over-accelerating to achieve the target speed while slipping, creating jolt back on dry surface	C, L	ins	Not impl	
	30	Actuation	Controller	Motion controller parameters (PID) are poorly tuned, making the vehicle overshoot at turns	C, L	all	Faulty conf	
CARLA	31		Simulator	Simulation does not properly apply control commands	C, L, V	all	Logic err	✓
	32		Simulator	Vector map contains a dead end blocked by objects as a valid lane	I, V	all	Data err	
	33		Simulator	Occasionally inconsistent simulation result	I, V	all	Logic err	✓

[Impact] **C**: Collision / **F**: Fails to complete a mission / **I**: Vehicle becomes Immobile / **L**: Lane invasion / **S**: Speeding / **V**: Miscellaneous traffic Violation
[Strategy] **all**: all strategies / **man**: Adversarial **man**euver-based / **con**: **con**gestion-based / **ent**: **ent**ropy-based / **ins**: **ins**tability-based

- Fuzzing with and without driving quality feedback
  - Approximately 2x bugs detected with the feedback
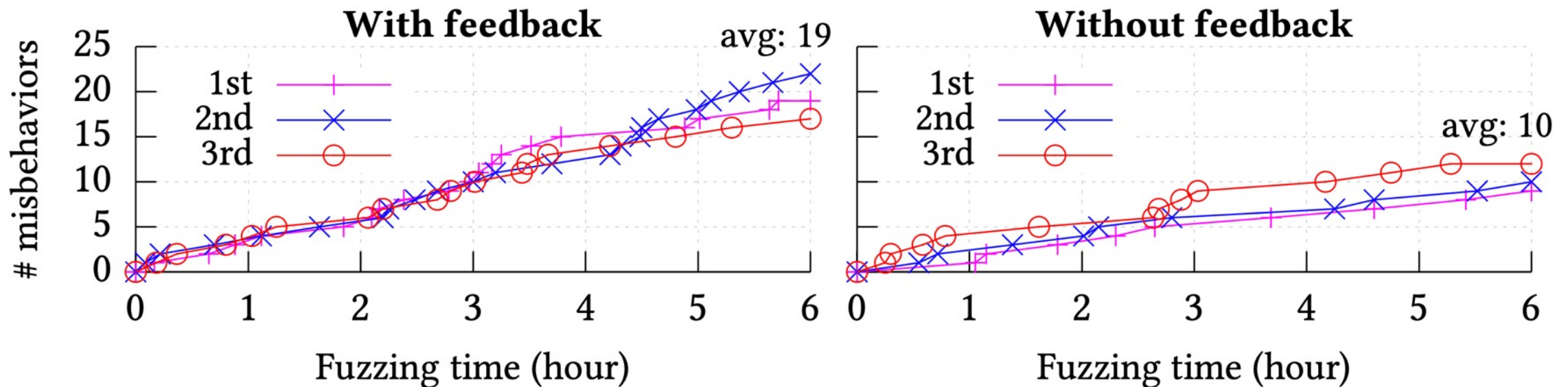
An interesting bug

Multi-layer faults

- Sensing & Perception
  - Fails to perceive the puddle

- Planning
  - Fails to consider the slipping state
  - Keeps commanding speed-up

- Actuation
  - Missing Electronic Stability Control (ESC)
  - Keeps increasing the throttle amount

DriveFuzz summary

- DriveFuzz: End-to-end fuzzing framework for ADS
- Mutate driving scenarios
  - Mission, actors, puddles, weather
- Look for safety-critical misbehaviors
  - Collision, infraction, and immobility
- Leverage semantic feedback using driving quality metrics
- Found 30 bugs in two industry grade ADS
  - Readily exploitable by controlling nearby actors or objects
- Additional materials
  - Website & code: https://drivefuzz.autoinsight.dev/

Q & A