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]

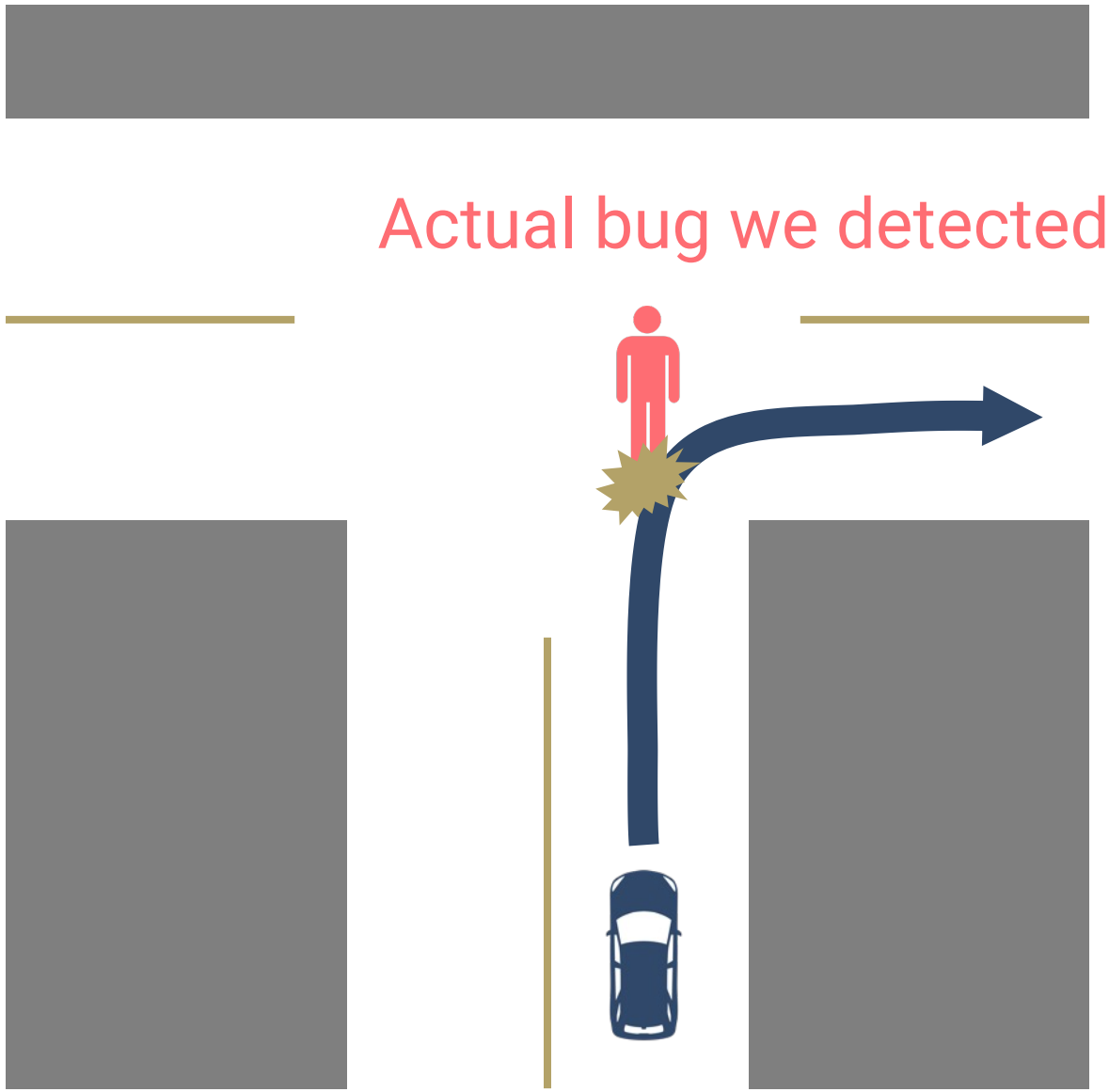
[1] Georgia Institute of Technology, [2] University of Texas at Dallas,
[3] University of Central Oklahoma, [4] UNIST, [5] University of Virginia

Can we trust autonomous driving systems?

- Expectation

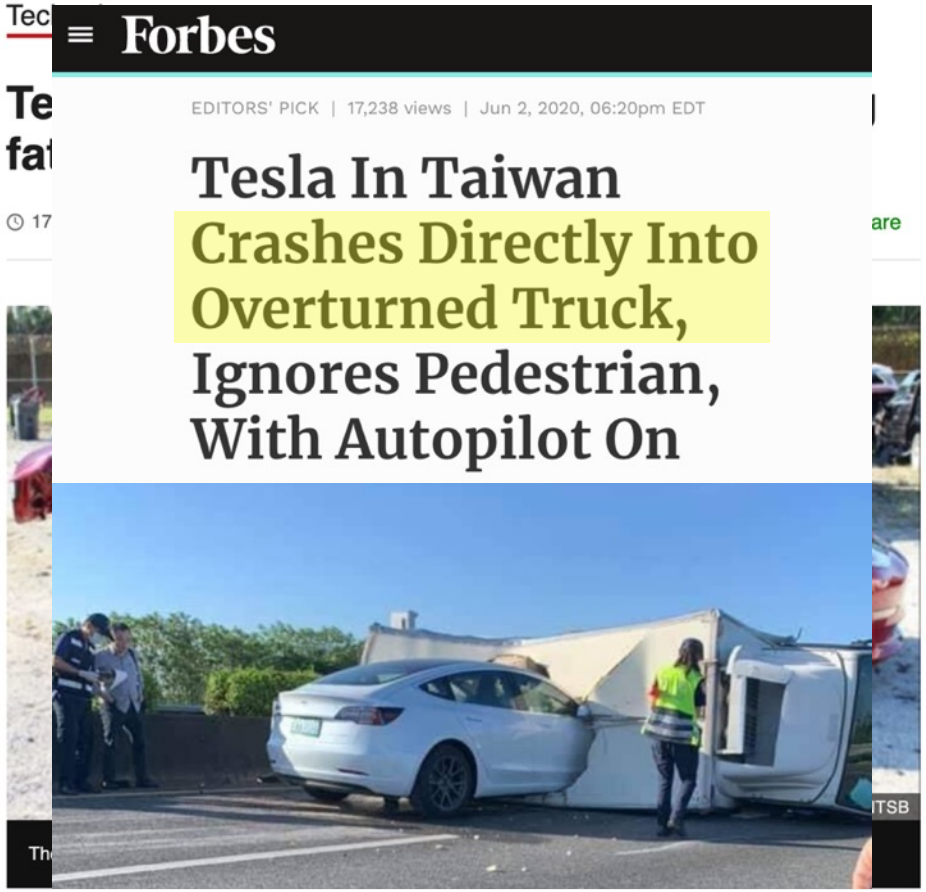
  vs

  Reality



Can we trust autonomous driving systems?

- Fatal autopilot accidents continue



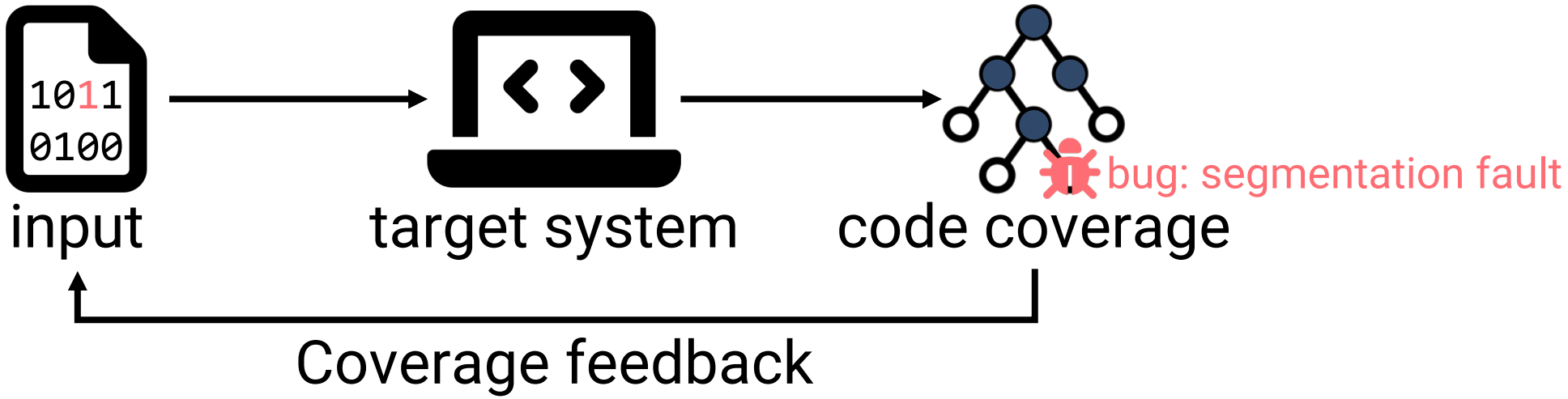
Finding bugs via manual testing

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

Source: “Will a Tesla KILL a cat?”
– Youtube Carwow

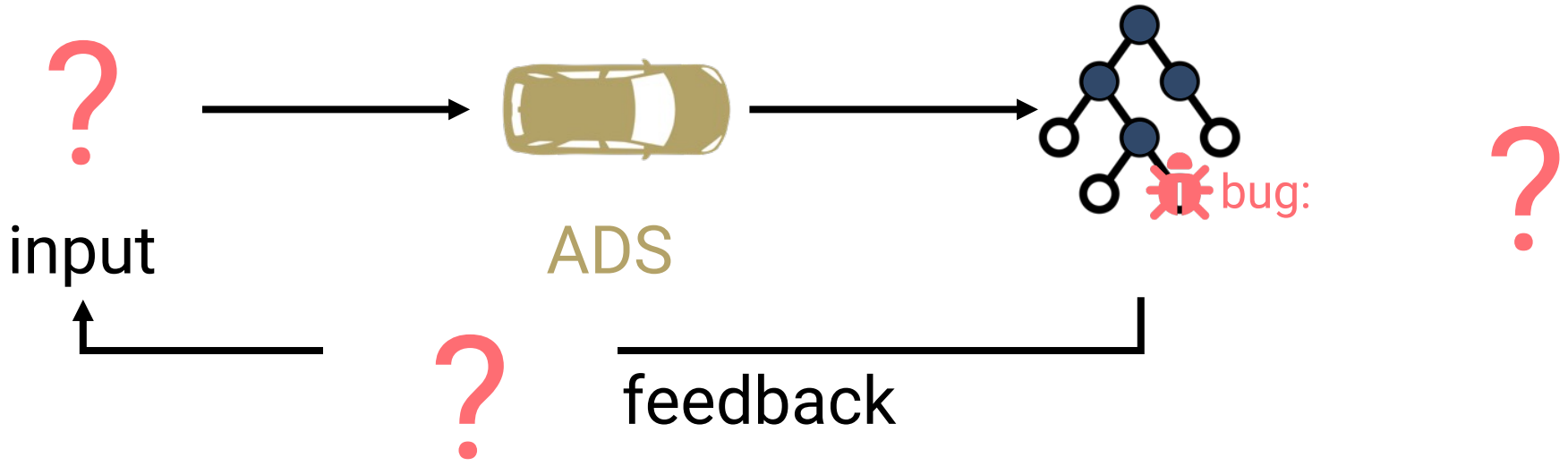
Finding bugs via automated testing

- Feedback-driven fuzzing for traditional software

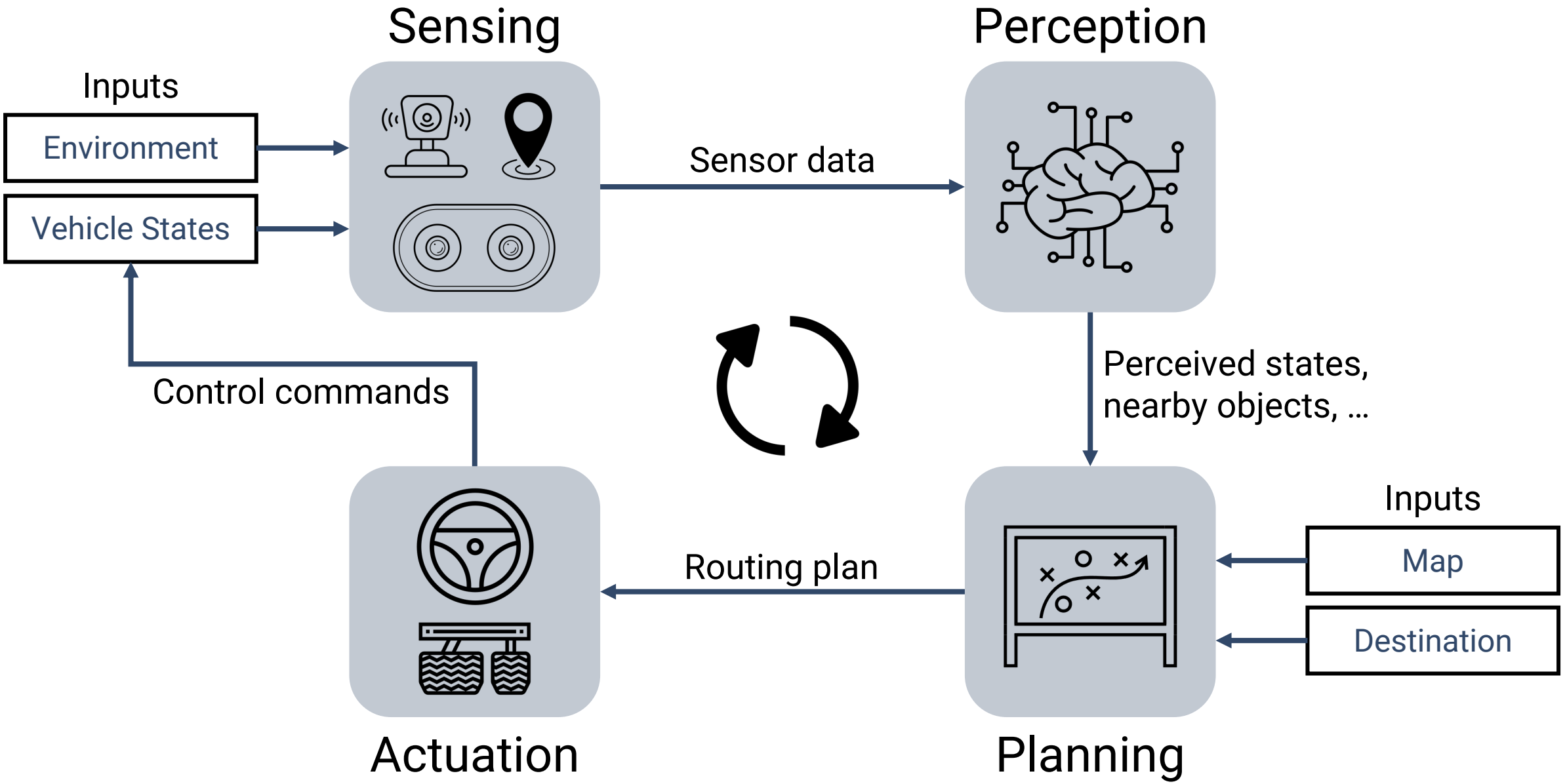


Finding bugs via automated testing

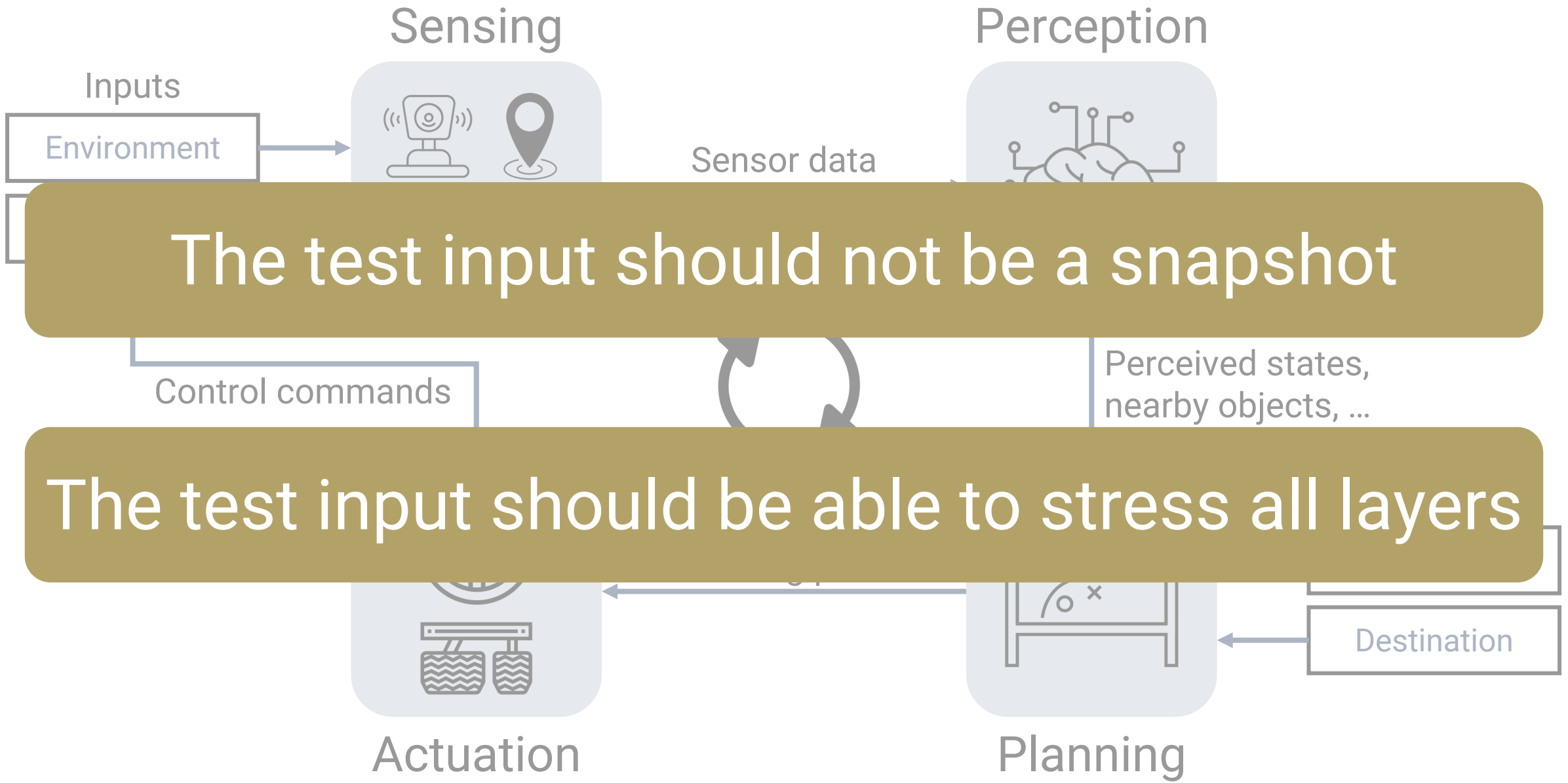
- Feedback-driven fuzzing for autonomous driving systems?



Layers and workflow of Autonomous Driving System (ADS)



Considerations in designing test inputs

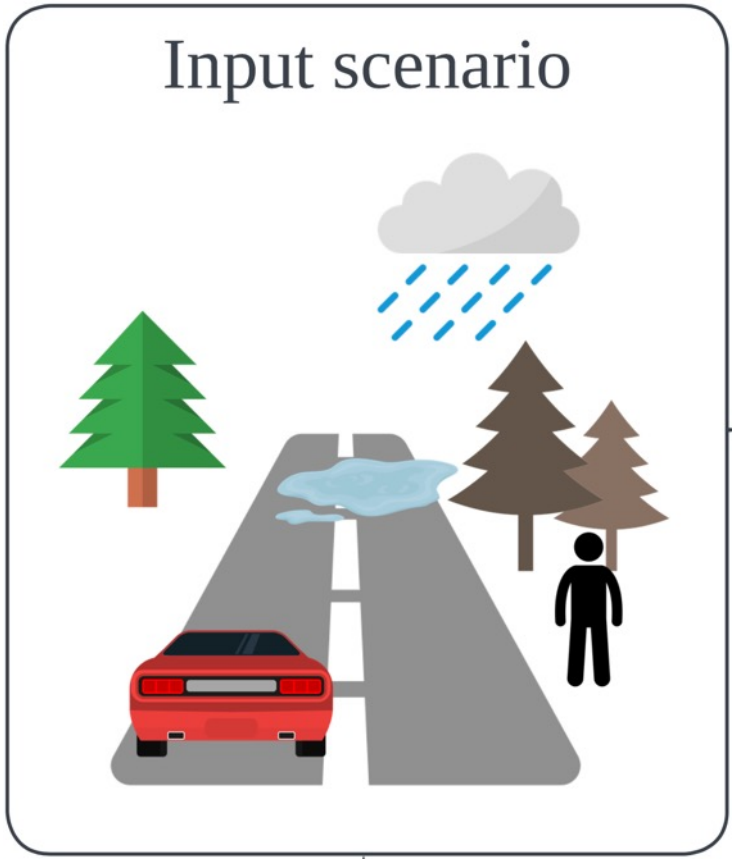


The test input should not be a snapshot

The test input should be able to stress all layers

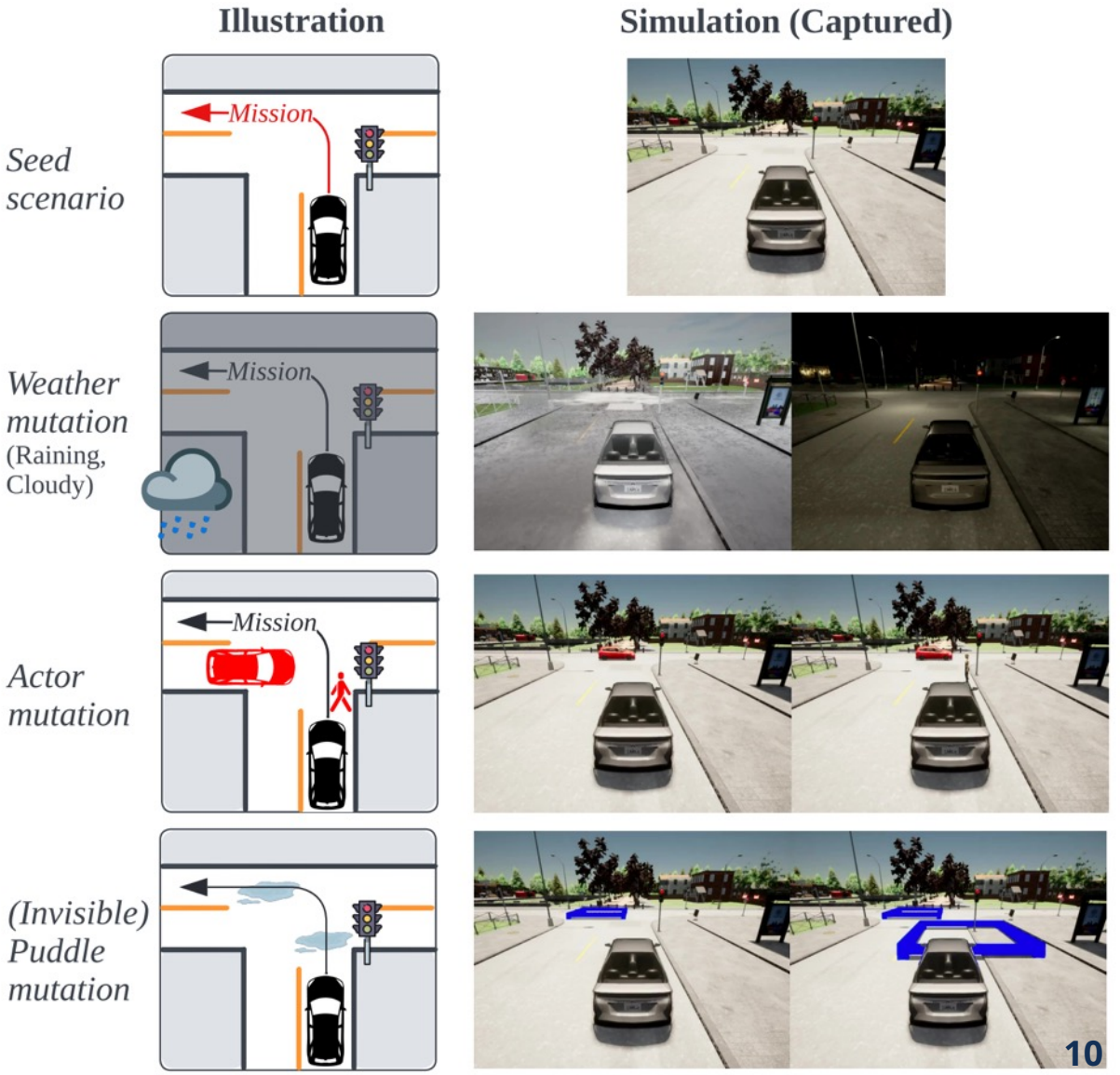
Our input space: Driving scenarios

- Representing temporal and spatial domains of real world
- Consists of
  1) 3D map
  2) Mission (initial and goal positions)
  3) Actors (vehicles or pedestrians)
  4) Puddles (e.g., black ice)
  5) Weather conditions



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

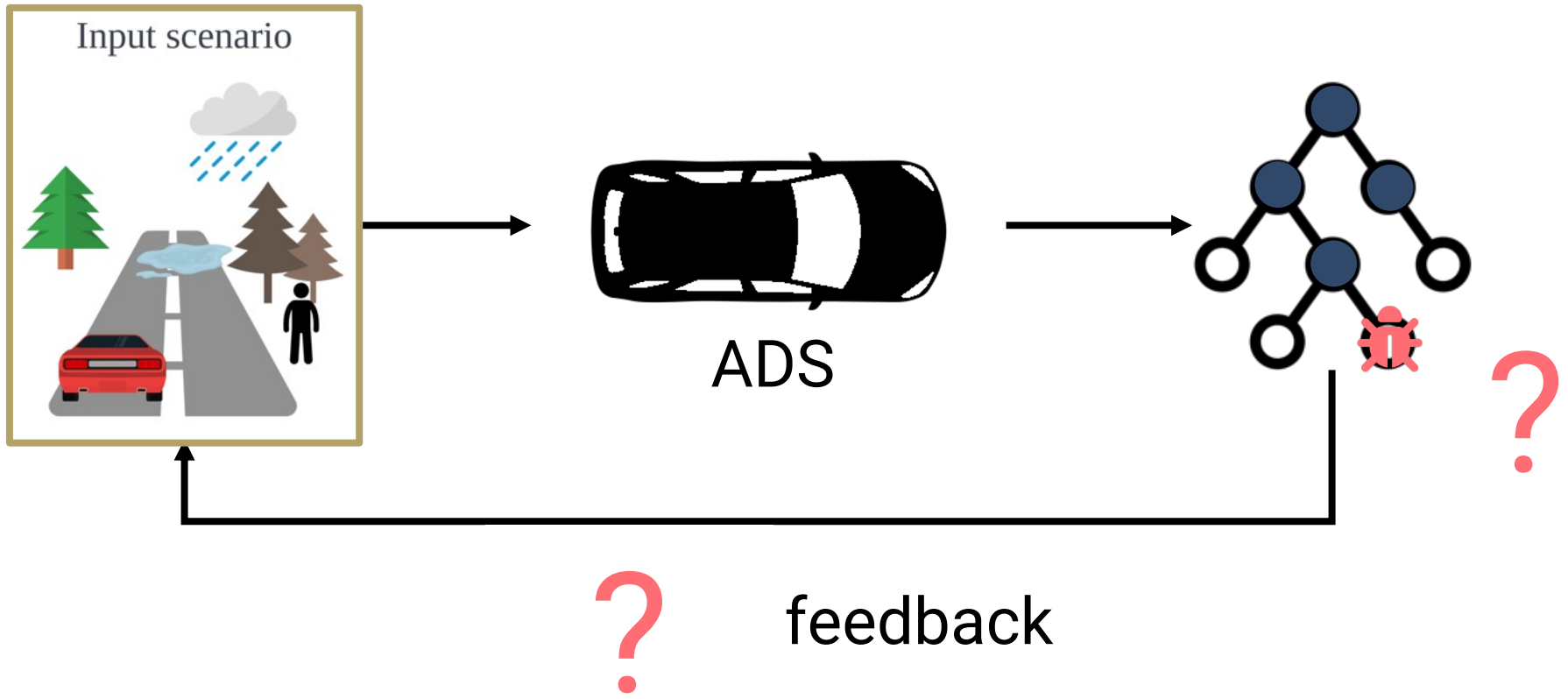


Confining mutation to feasible scenarios

- Two constraints to ensure physically valid mutation
  1) Spatial constraint
     - Initial positions of all actors and objects are spread away (e.g., 5 m)
     - Prevent unrealistic jams (e.g., vehicles overlapping)
  2) Temporal constraint
     - Maximum speed of actors are conservatively set
     - Prevent unrealistic behaviors
       - (e.g., a person running into a vehicle too quickly)

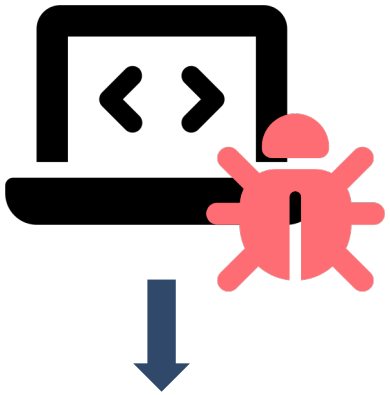
- Both constraints are configurable

Feedback-driven fuzzing for ADS

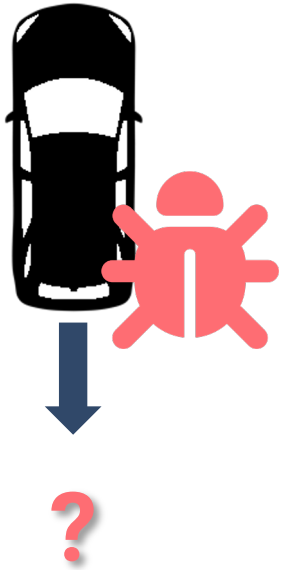


Defining autonomous driving bugs

- Question: What happens to a buggy ADS?



```
$ ./buggy_program
[1]    3541023 segmentation fault  ./buggy_program
```



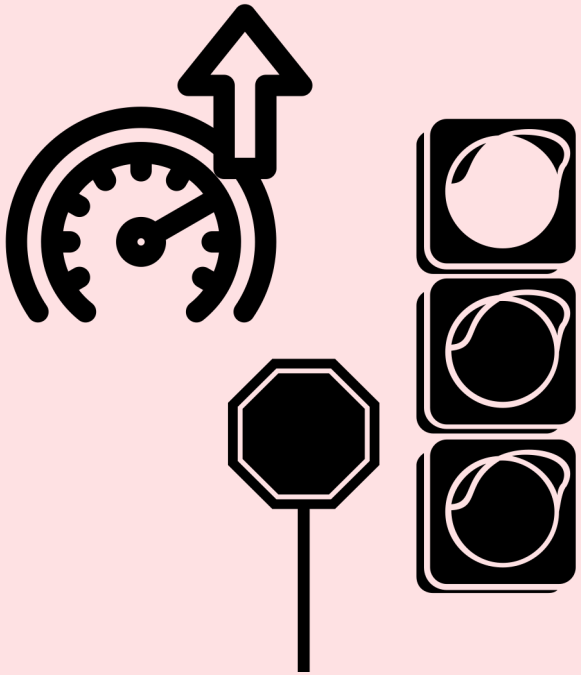
Classic software bugs

Safety-critical vehicular misbehaviors

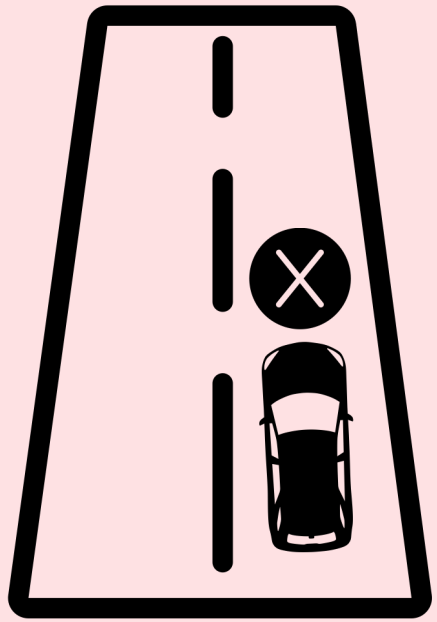
- ADS must comply with traffic rules & regulations

Collision

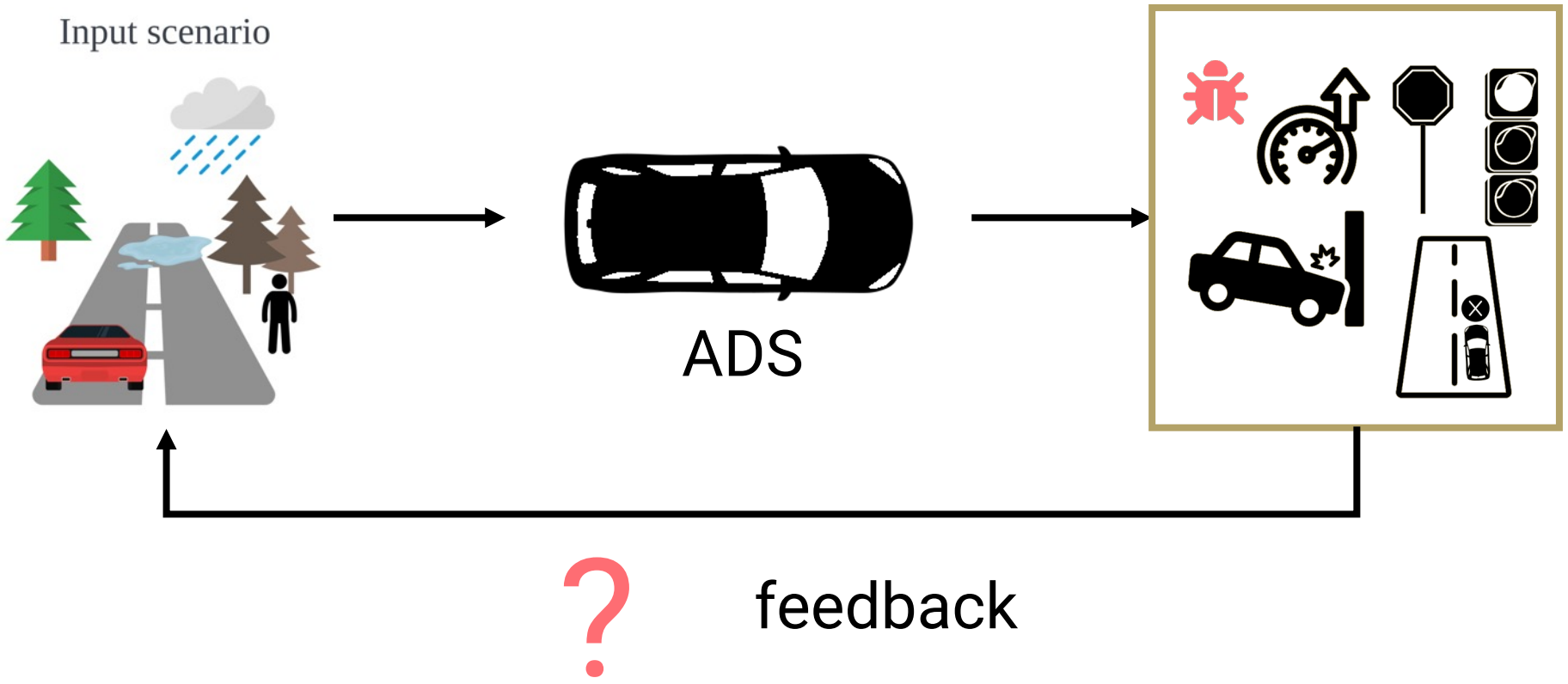
Infraction



Immobility

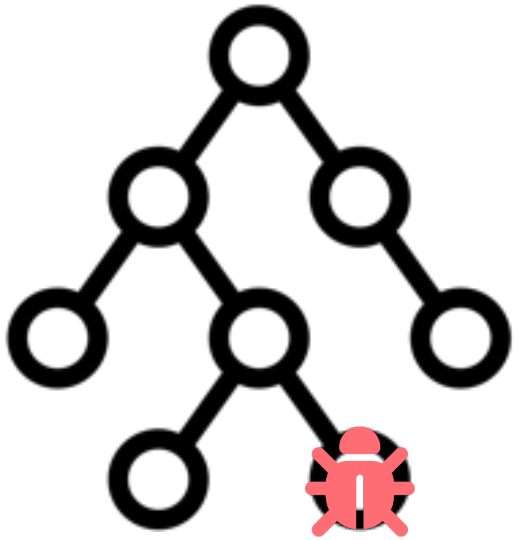


Feedback-driven fuzzing for ADS



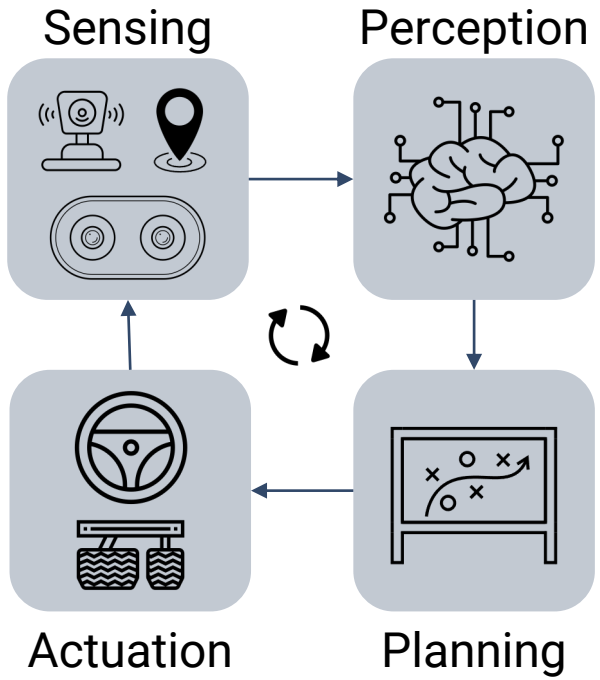
A need for a new feedback mechanism

General software programs



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

Autonomous driving system



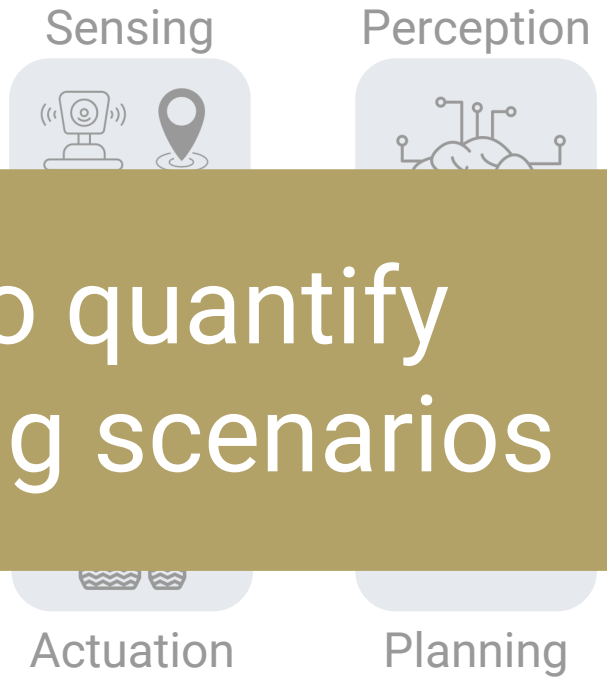
- Distributed system
- Behavior is driven by state changes in a loop, not code paths

A need for a new feedback mechanism

General software programs

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

Autonomous driving system



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

Need proper metrics to quantify the quality of input driving scenarios

Solution: Driving quality feedback

- Intuition
  - Quality of driving ≃ likelihood of misbehaviors

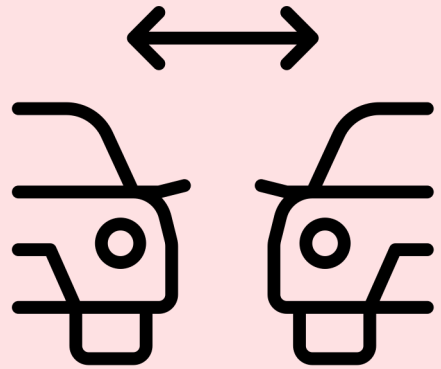
Hard acceleration, braking, and turns

- Metric auto insurance companies use

Oversteer and understeer

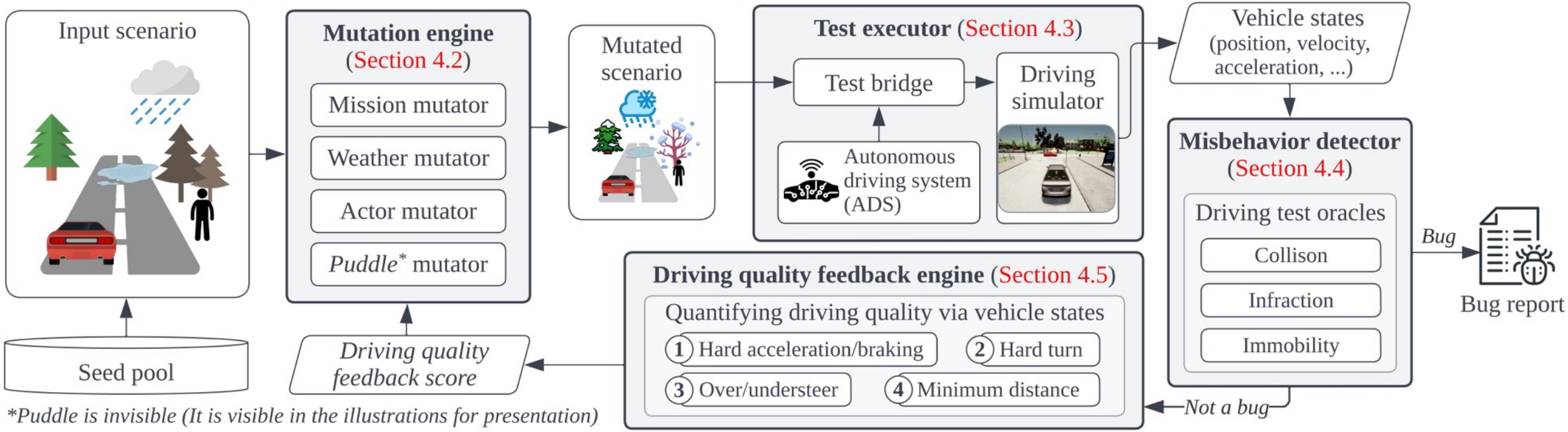
- #1 cause of motorsport accidents

Minimum distance to other actors



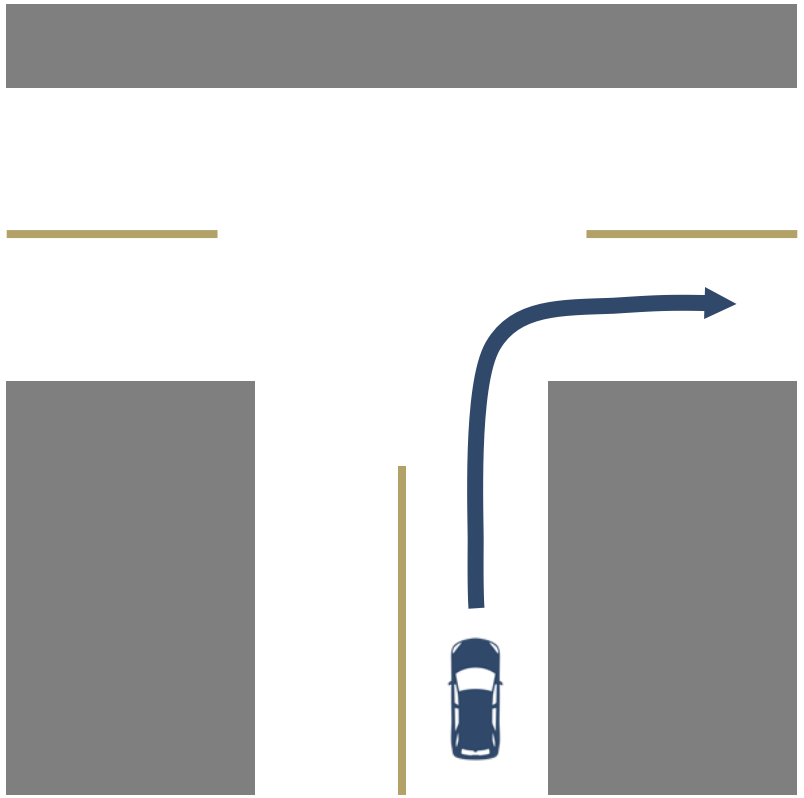
- Near-missed collisions

DriveFuzz overview



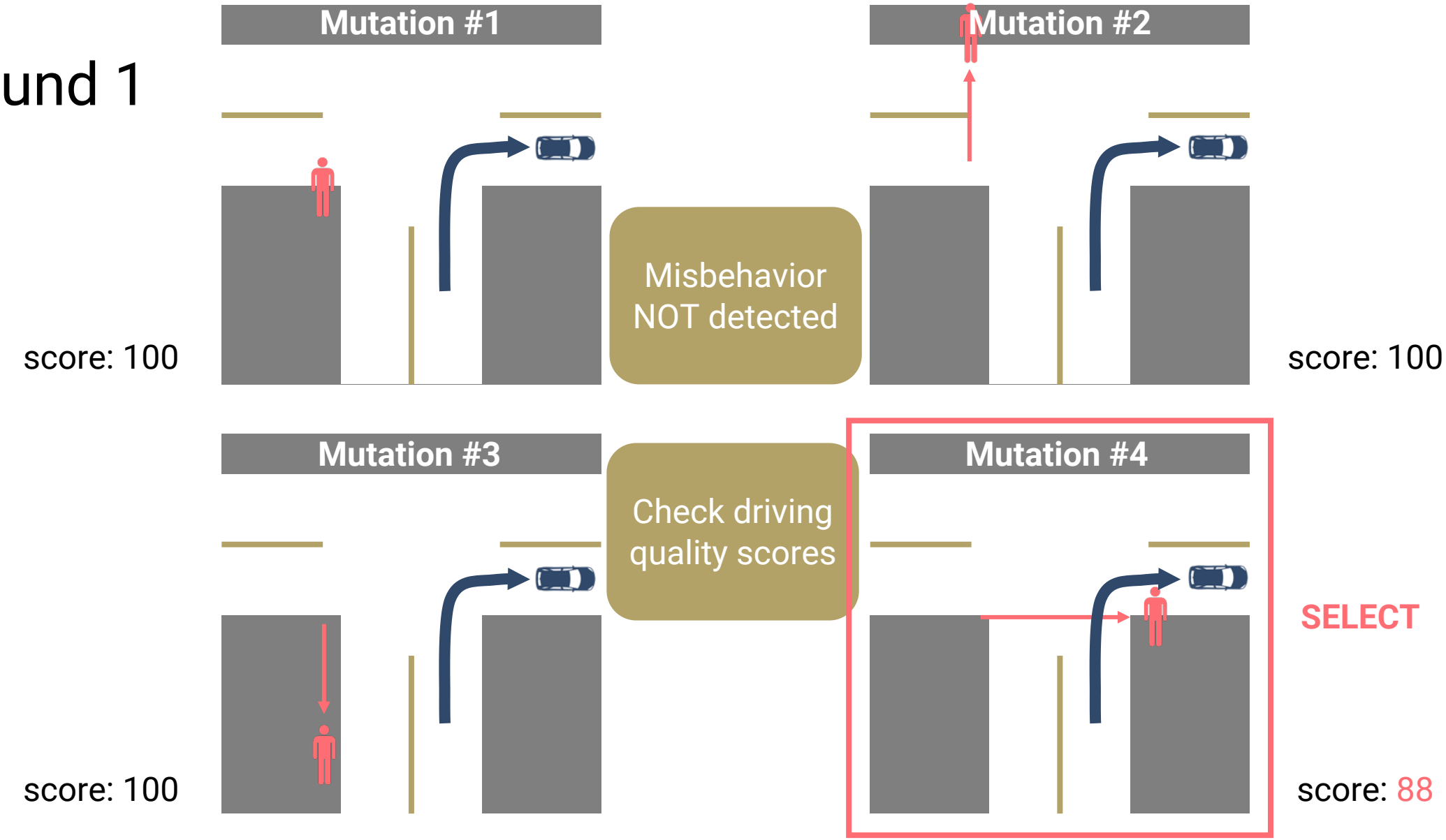
DriveFuzz in action

- Seed scenario
  - Map
  - Initial position
  - Destination



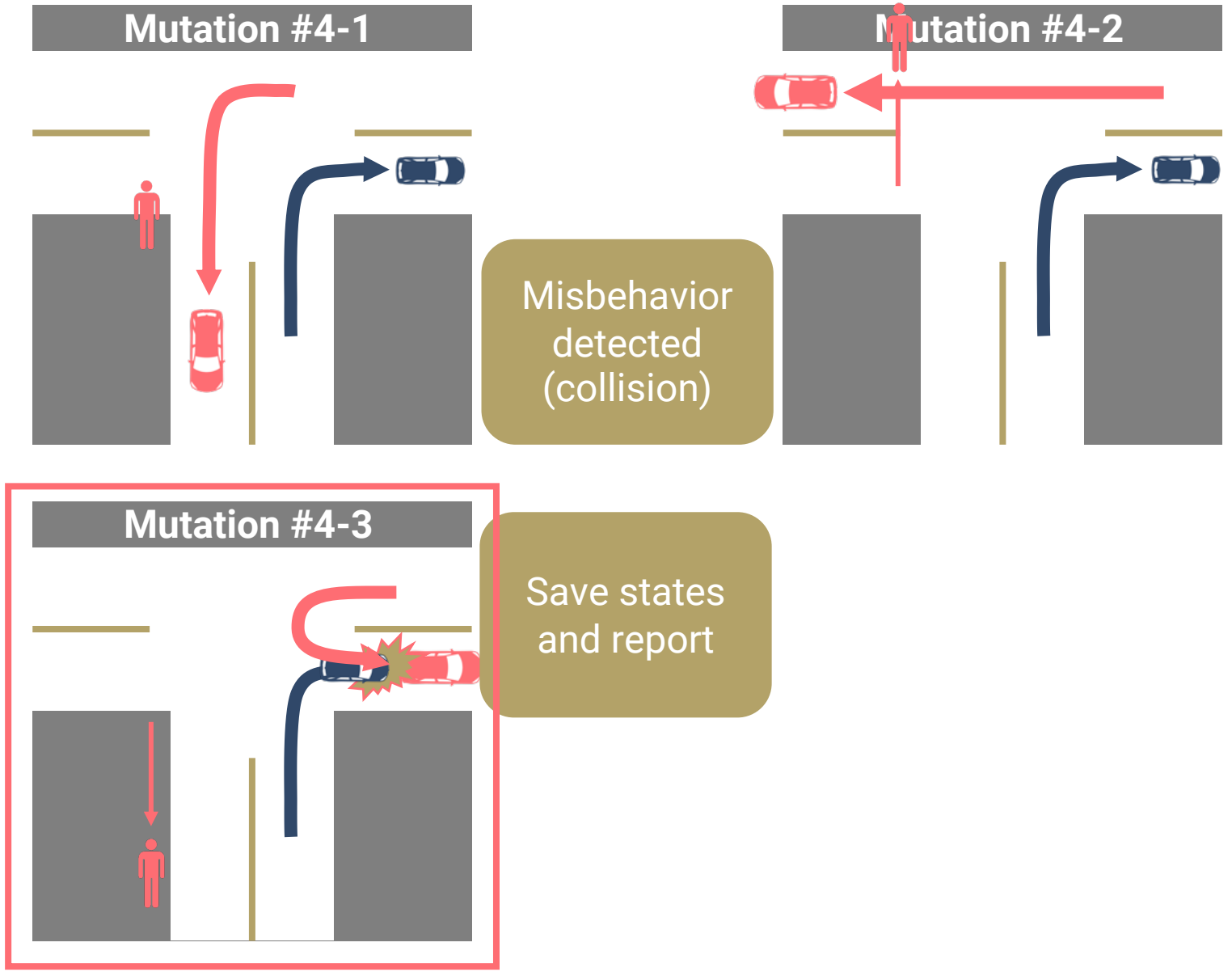
DriveFuzz in action

- Round 1



DriveFuzz in action

- Round 2



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

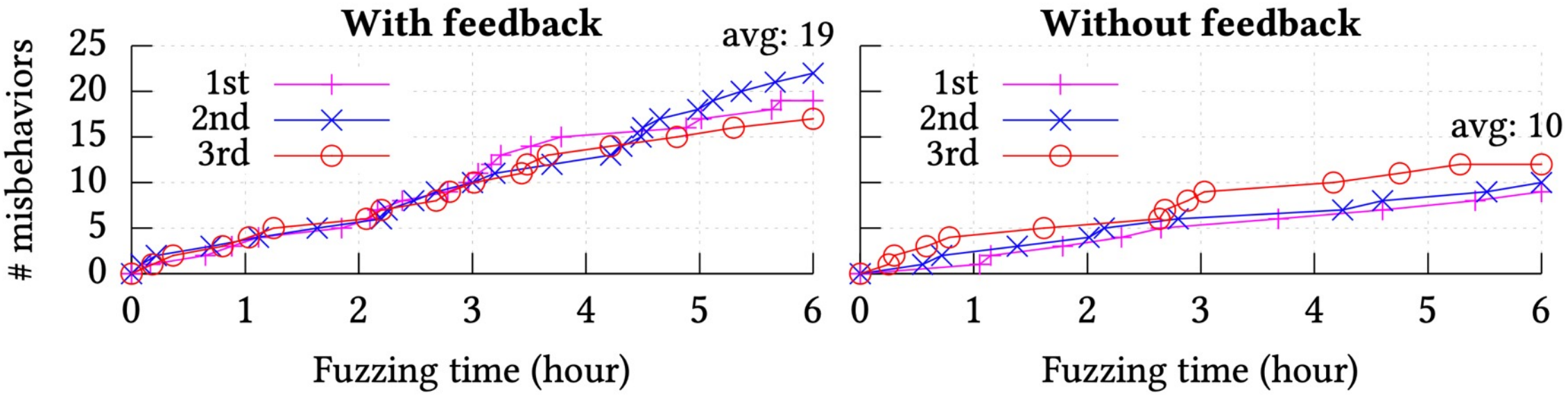
Detected 33 new bugs throughout ADS layers

	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**: **C**ollision / **F**: **F**ails to complete a mission / **I**: Vehicle becomes **I**mmobile / **L**: **L**ane invasion / **S**: **S**peeding / **V**: Miscellaneous traffic **V**iolation
[Strategy] **all**: all strategies / **man**: Adversarial **man**euver-based / **con**: **con**gestion-based / **ent**: **ent**ropy-based / **ins**: **ins**tability-based

The impact of driving quality feedback

- 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

Contact: Seulbae Kim
- seulbae@gatech.edu
- https://squizz617.github.io