Too Afraid to Drive: Systematic Discovery of Semantic DoS Vulnerability in Autonomous Driving Planning under Physical-World Attacks

Ziwen Wan, Junjie Shen, Jalen Chuang, Xin Xia,
Joshua Garcia, Jiaqi Ma, and Qi Alfred Chen

AS²Guard Autonomous & Smart Systems Guard Research Group

UCI

UCLA

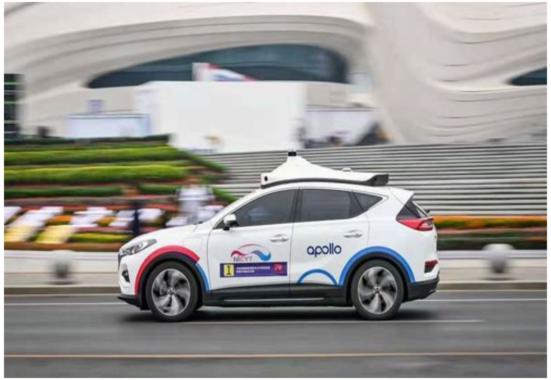
Who is Qi Alfred Chen?

- **Research Interest:** AI/Autonomous Vehicles/Intelligent Transportation Systems

- **Publications**
- Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving (CCS ‘19)
- Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing (Usenix Security ’20)
- Too Afraid to Drive: Systematic Discovery of Semantic DoS Vulnerability in Autonomous Driving Planning under Physical-World Attacks (NDSS ‘22)

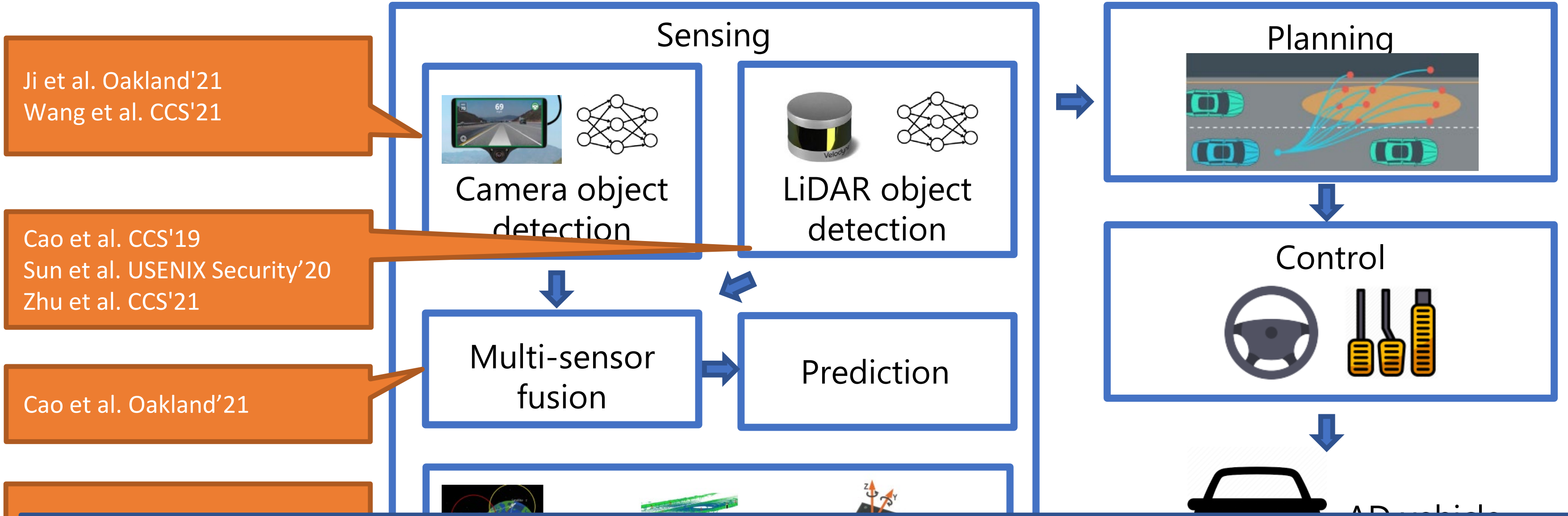
The rise of Autonomous Driving (AD) vehicles

- **High-level** autonomous driving vehicles are already providing services **without safety drivers**.



Current status of AD security research

- We have witnessed security problems in high-level AD systems.



Ji et al. Oakland'21
Wang et al. CCS'21

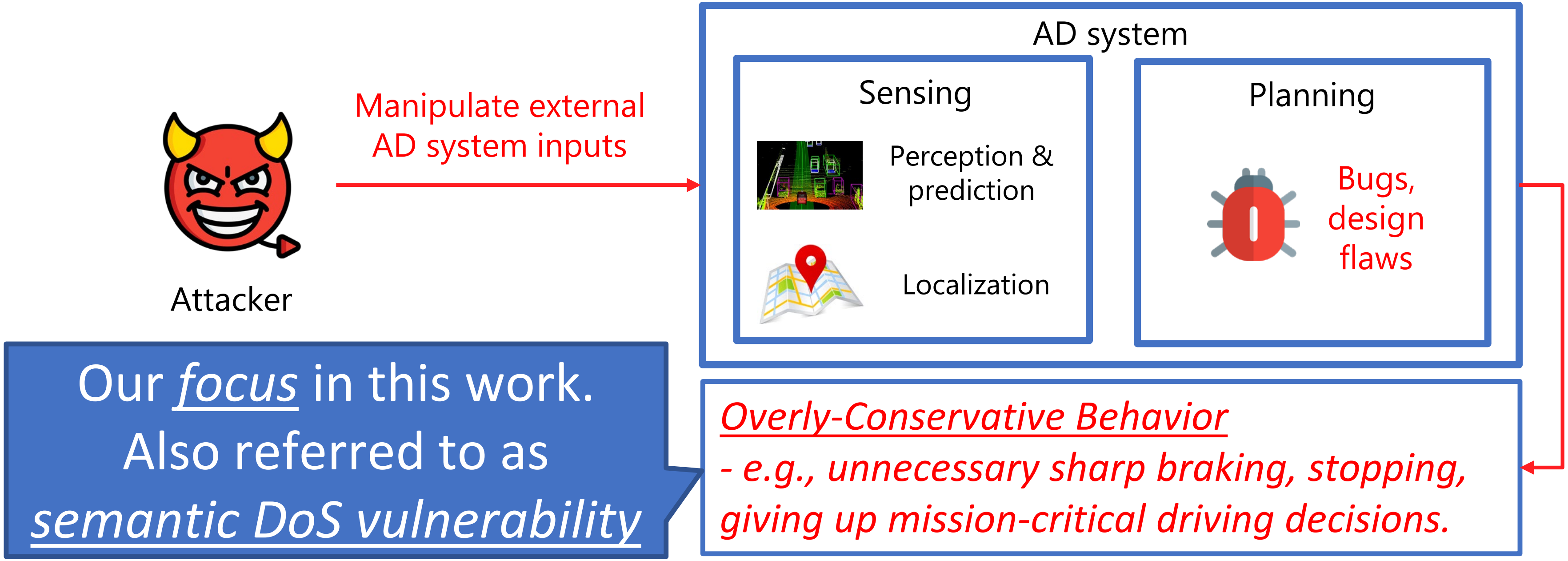
Cao et al. CCS'19
Sun et al. USENIX Security’20
Zhu et al. CCS'21

Cao et al. Oakland’21

Question: Could planning (critical driving decision-making) also be vulnerable and thus exploitable to external attackers?

Our focus: Semantic vulnerability in AD planning

- **Definition**: causing planning to change a normal driving decision to an *unexpected* one



Attacker

Manipulate external AD system inputs

AD system

Sensing

Perception & prediction

Localization

Planning

Bugs, design flaws

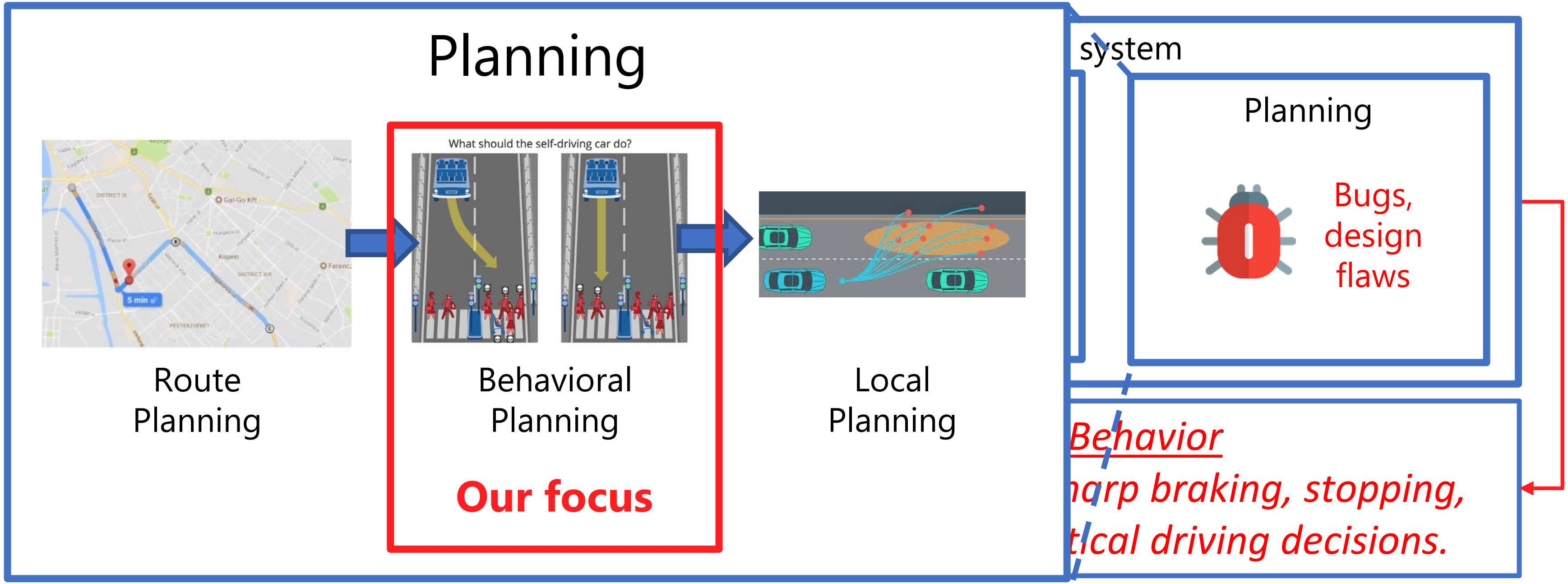
Our *focus* in this work. Also referred to as *semantic DoS vulnerability*

Overly-Conservative Behavior

- e.g., unnecessary sharp braking, stopping, giving up mission-critical driving decisions.

Our target: Behavioral Planning (BP)

- Functionality of BP: Makes mission-critical driving decisions, e.g., collision avoidance, lane changing



Planning

Route Planning

Behavioral Planning

Our focus

Local Planning

system

Planning

Bugs, design flaws

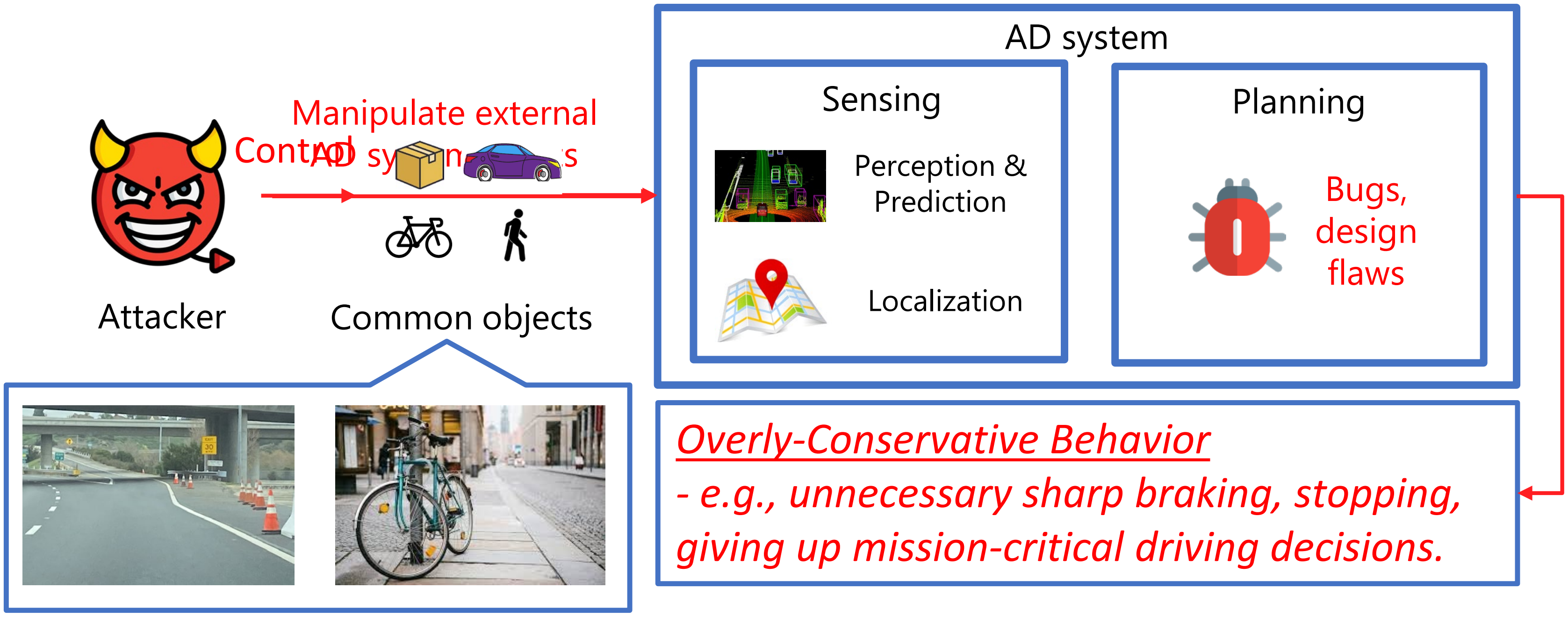
Behavior

arp braking, stopping,

tical driving decisions.

Threat model

- **Attack vector**: *attacker-controllable common* roadside objects
  - e.g., dumped cardboard boxes, parked bikes on the road side



Overly-Conservative Behavior

- e.g., unnecessary sharp braking, stopping, giving up mission-critical driving decisions.

Consequence of semantic DoS vulnerability

Consequences

Bad user experience

Safety

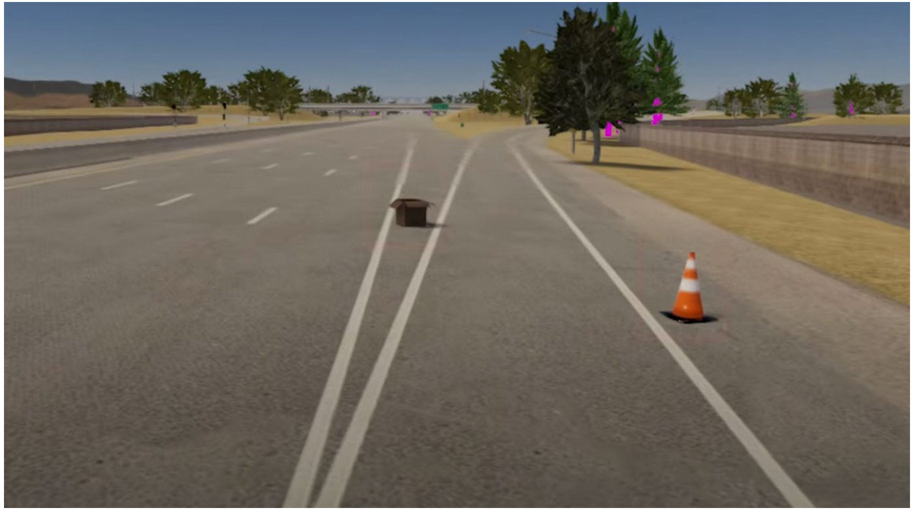
Block traffic

Law violation in specific places

Overly-Conservative Behavior

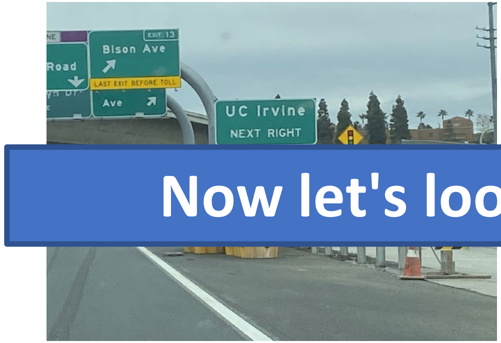
- e.g., unnecessary sharp braking, stopping, giving up mission-critical driving decisions.

Semantic DoS vulnerability demo



As a human driver, how should you react to this scenario at the highway off-ramp?

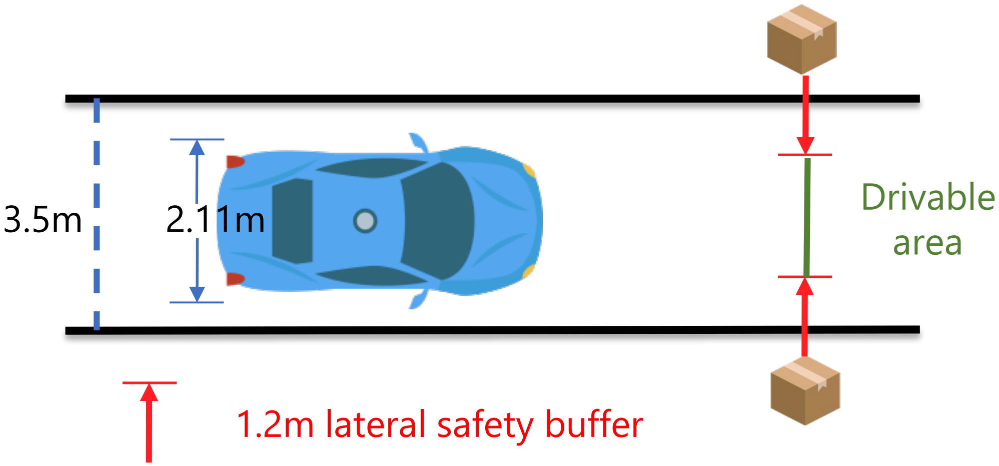
- ➢ Ignore them?
- ➢ Slightly slow down?

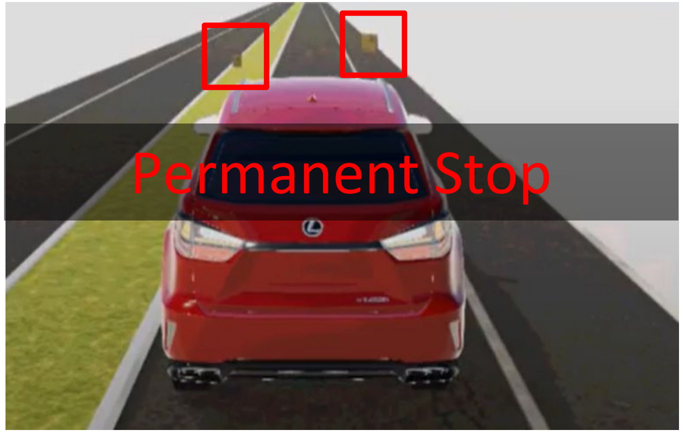


Now let's look into a demo we created with Autoware.AI.

Two pictures around our campus.

Root cause of the DoS vulnerability





Drivable area (minimal value is (3.5 - 2*1.2)) < car width (2.11m)

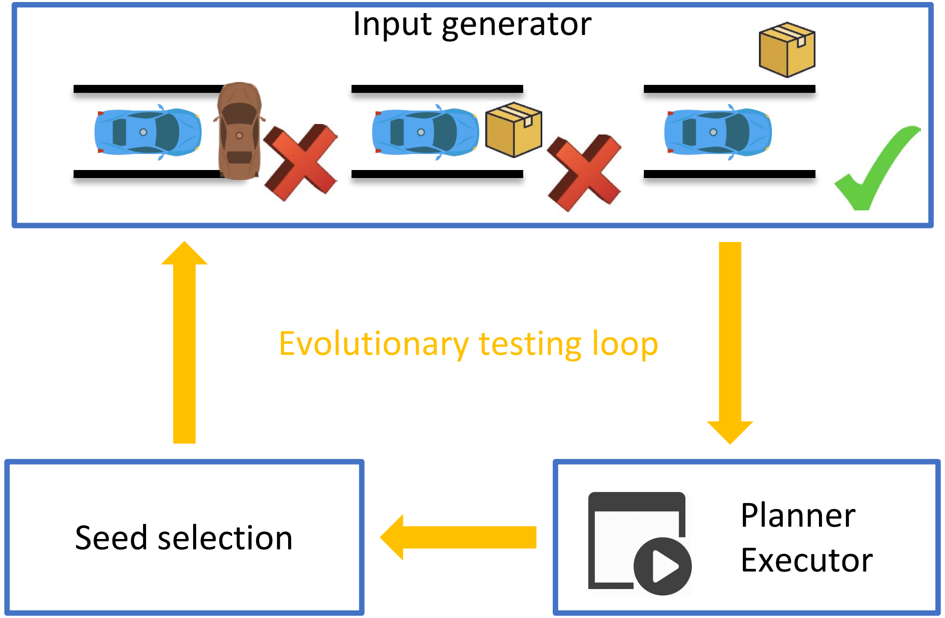
The AD vehicle thinks there is not enough space

PlanFuzz: Design challenges

- We design *PlanFuzz,* a novel dynamic testing tool to automate the semantic DoS vulnerabilities discovery

Challenge 2: How to generate inputs that satisfy domain constraints?

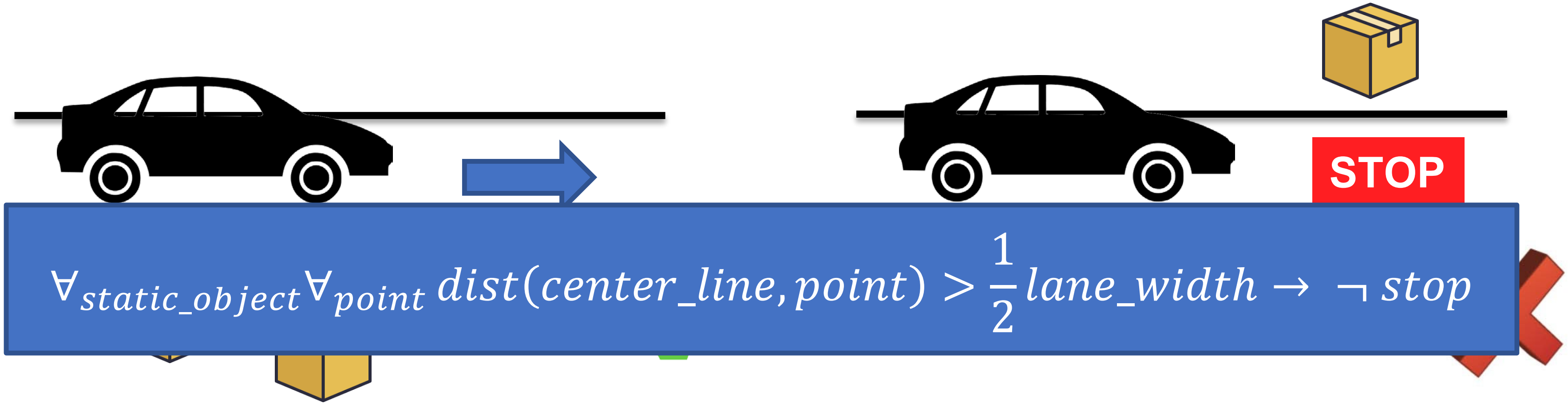
Challenge 3: How to design feedback to efficiently guide the testing ?



Challenge 1: How to judge a driving decision is *overly-conservative*?

Solution: Planning Invariant (PI)

- To address challenge 1 (lack of testing oracles for semantic DoS vuln), we design planning invariant
  - Planning Invariants (PI) = planning scenario + desired planning behavior + attacker-controllable changes



$$\forall_{static\_object} \forall_{point}\, dist(center\_line, point) > \frac{1}{2} lane\_width \rightarrow \neg\, stop$$

Solution: Planning Invariant (PI)

- **Systematically** define PIs under 8 diverse scenarios with **temporal logic** to constraint static objects, and **moving** pedestrian/vehicles

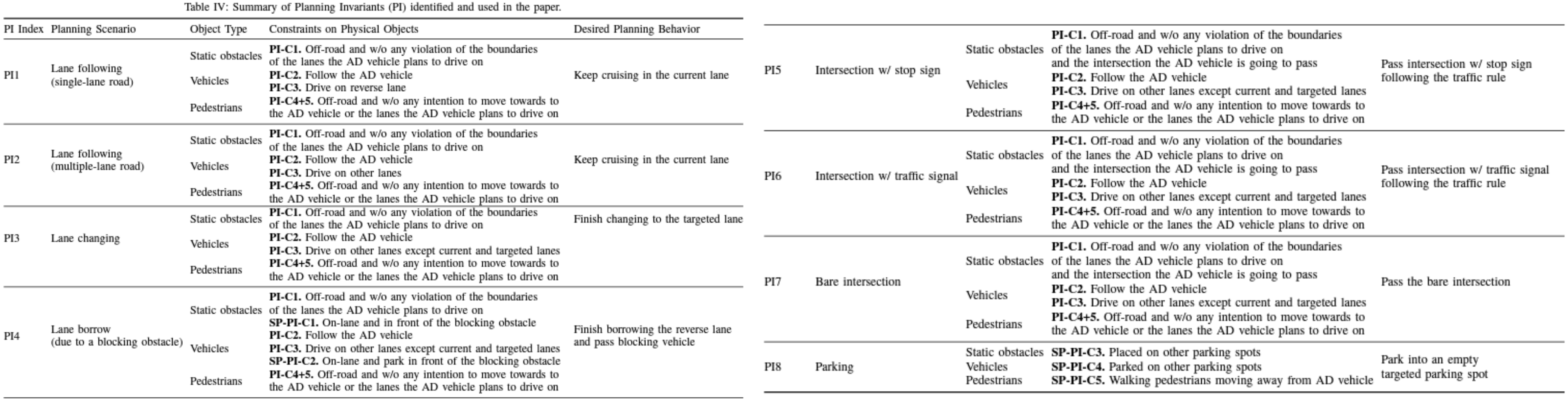


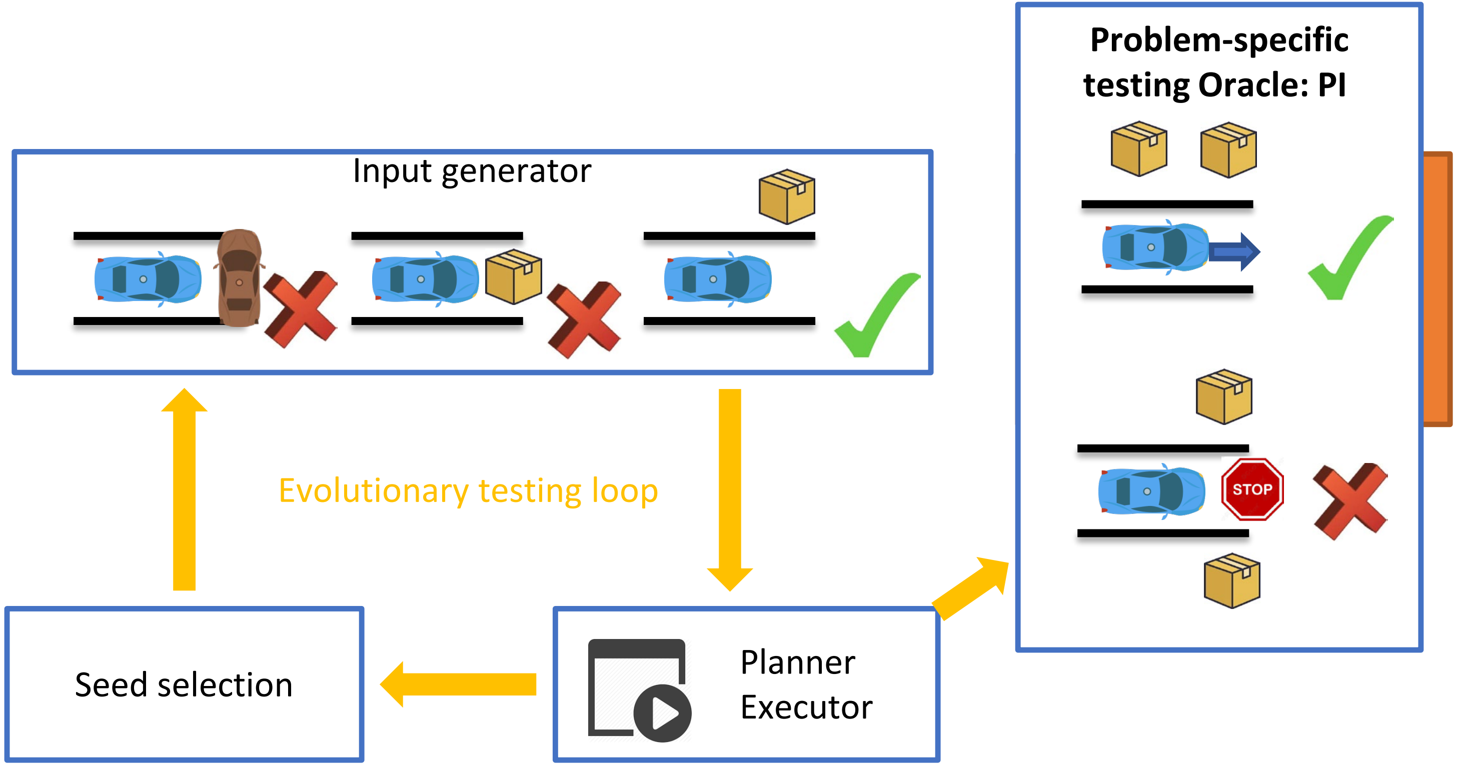
Table IV: Summary of Planning Invariants (PI) identified and used in the paper.

PI Index	Planning Scenario	Object Type	Constraints on Physical Objects	Desired Planning Behavior
PI1	Lane following (single-lane road)	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on reverse lane **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Keep cruising in the current lane
PI2	Lane following (multiple-lane road)	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Keep cruising in the current lane
PI3	Lane changing	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Finish changing to the targeted lane
PI4	Lane borrow (due to a blocking obstacle)	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on **SP-PI-C1.** On-lane and in front of the blocking obstacle **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes **SP-PI-C2.** On-lane and park in front of the blocking obstacle **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Finish borrowing the reverse lane and pass blocking vehicle
PI5	Intersection w/ stop sign	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on and the intersection the AD vehicle is going to pass **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Pass intersection w/ stop sign following the traffic rule
PI6	Intersection w/ traffic signal	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on and the intersection the AD vehicle is going to pass **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Pass intersection w/ traffic signal following the traffic rule
PI7	Bare intersection	Static obstacles Vehicles Pedestrians	**PI-C1.** Off-road and w/o any violation of the boundaries of the lanes the AD vehicle plans to drive on and the intersection the AD vehicle is going to pass **PI-C2.** Follow the AD vehicle **PI-C3.** Drive on other lanes except current and targeted lanes **PI-C4+5.** Off-road and w/o any intention to move towards to the AD vehicle or the lanes the AD vehicle plans to drive on	Pass the bare intersection
PI8	Parking	Static obstacles Vehicles Pedestrians	**SP-PI-C3.** Placed on other parking spots **SP-PI-C4.** Parked on other parking spots **SP-PI-C5.** Walking pedestrians moving away from AD vehicle	Park into an empty targeted parking spot

Solution: Planning Invariant (PI)

Challenge 2: How to generate inputs that satisfy domain constraints?

Challenge 3: How to design feedback to efficiently guide the testing?

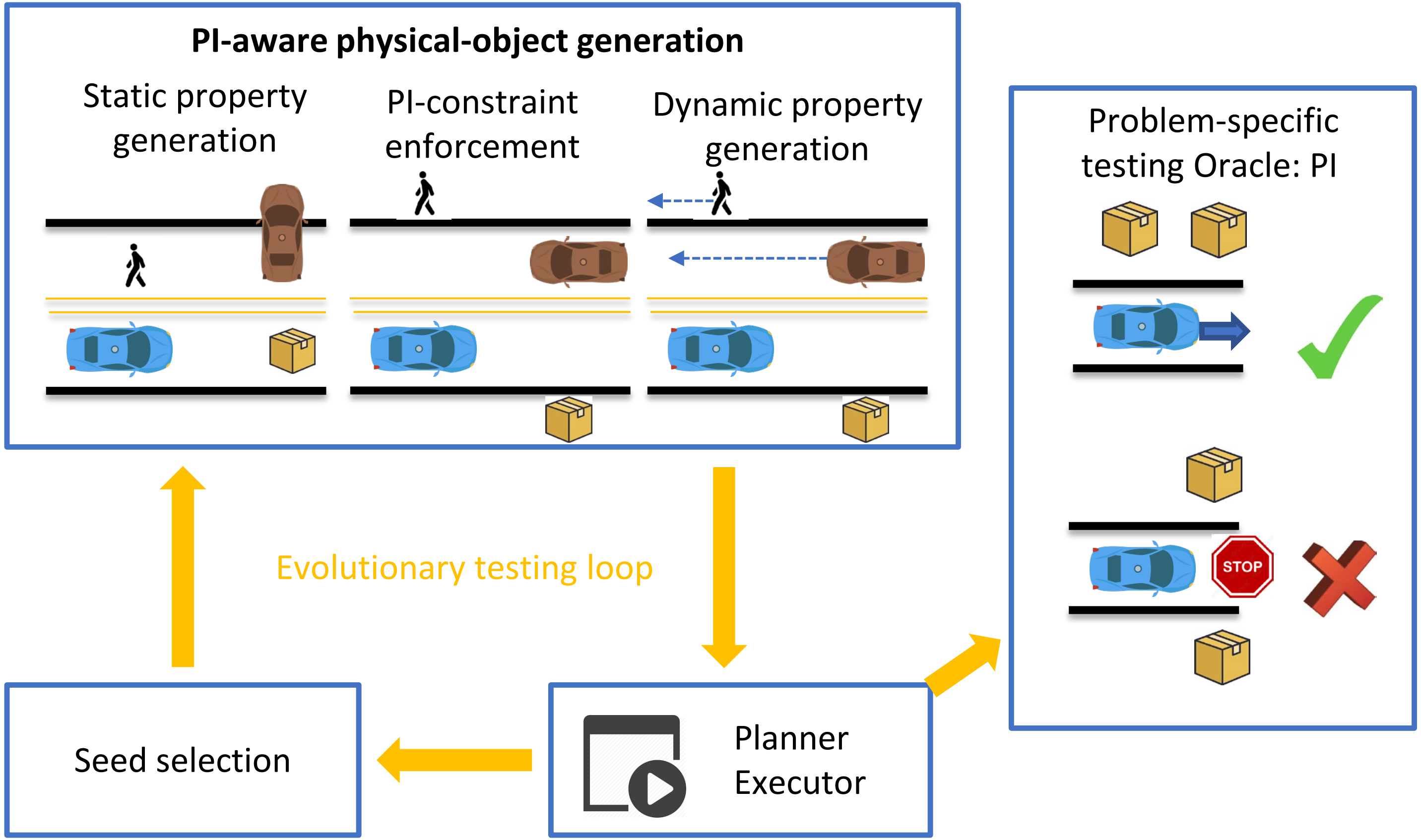

Input generator
Evolutionary testing loop
Seed selection
Planner Executor
Problem-specific testing Oracle: PI
STOP

Solution: PI-aware physical-object generation

Input generation:

- Satisfy domain-specific constraints
- Maintain diversity and inheritance during mutation

Challenge 3: How to design feedback to efficiently guide the testing?



PI-aware physical-object generation

Static property generation

PI-constraint enforcement

Dynamic property generation

Evolutionary testing loop

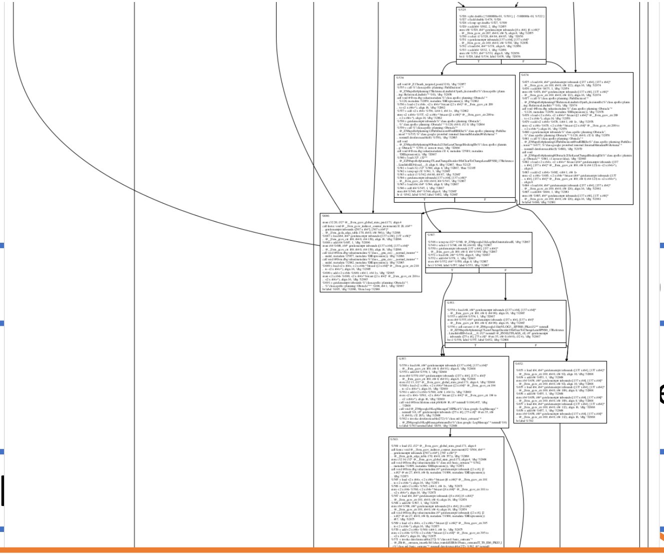
Seed selection

Planner Executor

Problem-specific testing Oracle: PI

Solution: BP vulnerability distance

- To address challenge 3 (lack of efficient guidance)
  - We propose **BP vulnerability distance,** which is a **gray-box** guidance.



Tiny fraction of Apollo lane changing control flow graph

Key idea: Use the distance between operands in decision-related predicates to guide driving decision changes

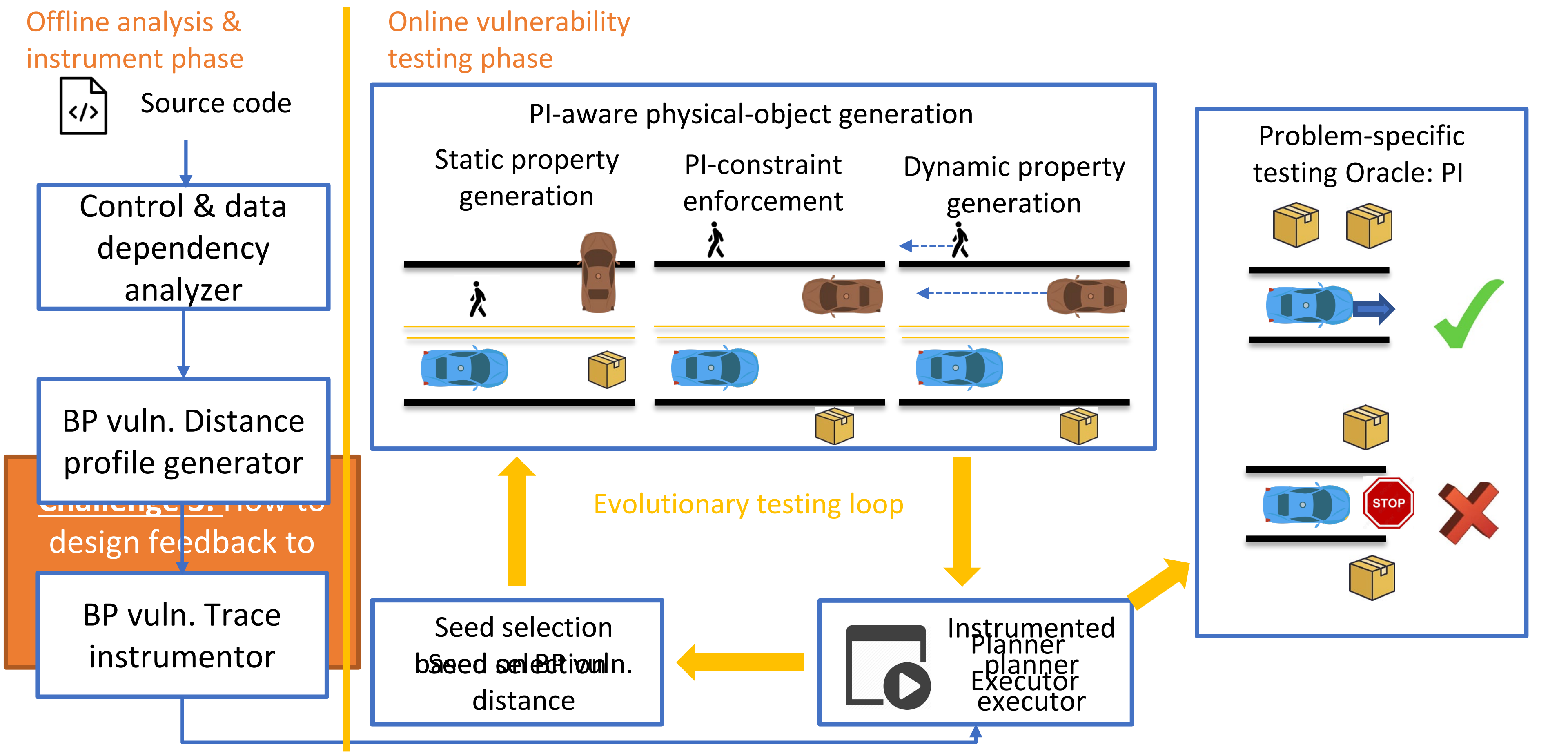
Offline static analysis:

- Extract control/data dependency
- Generate BP vuln. distance profile for instrumentation

Online dynamic analysis:

- Calculate BP vuln. dist. at runtime

Solution: BP vulnerability distance



Offline analysis & instrument phase

Source code

Control & data dependency analyzer

BP vuln. Distance profile generator

Challenge 3: How to design feedback to

BP vuln. Trace instrumentor

Online vulnerability testing phase

PI-aware physical-object generation

Static property generation

PI-constraint enforcement

Dynamic property generation

Evolutionary testing loop

Seed selection based on BP vuln. distance

Seed selection

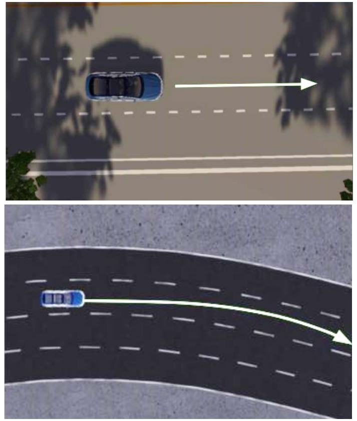
Instrumented planner executor

Planner Executor

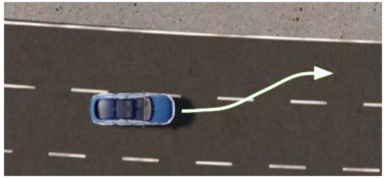
Problem-specific testing Oracle: PI

Evaluations: DoS semantic vulnerabilities discovery

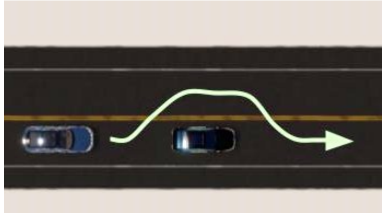
- **9 previously unknown** semantic DoS vulnerabilities from **3 BP implementations** of Baidu Apollo and Autoware.AI (full-stack open-source AD software)
  - Causes: 1 due to implementation bug, 8 due to overly-conservative planning parameters (e.g., safety buffer, angle threshold) & overly-conservative estimation of surrounding object intentions (e.g., from pedestrians, parked bicycles)
- **Diverse** driving scenarios
  - 28,789 BP decision snapshots from 40 driving traces & 8 different scenario types



Lane following



Lane changing

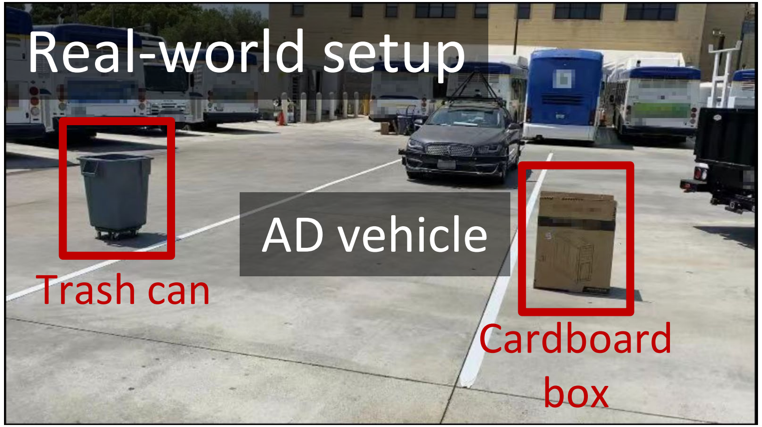


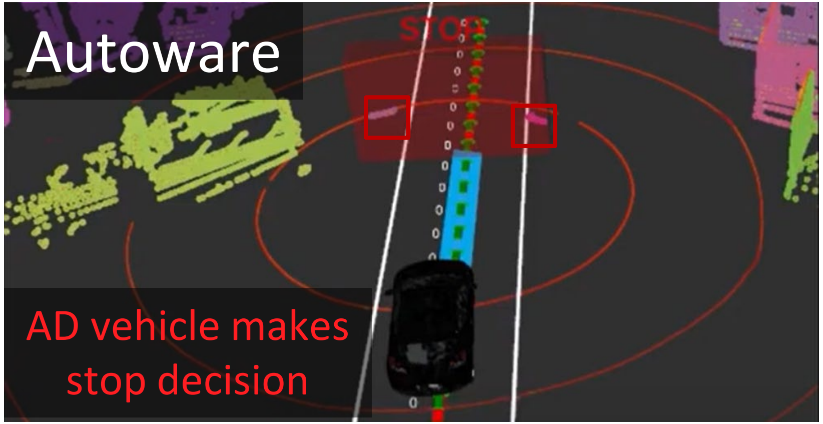
Lane borrowing

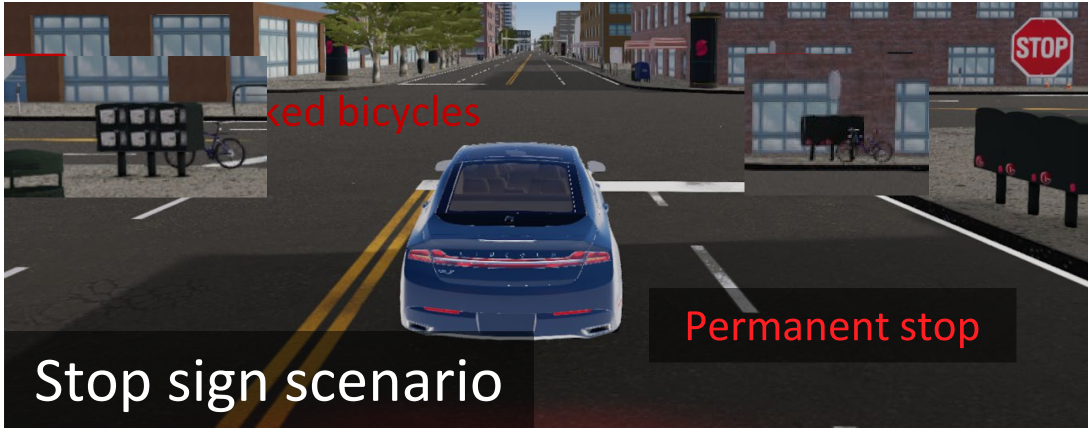
Intersection passing

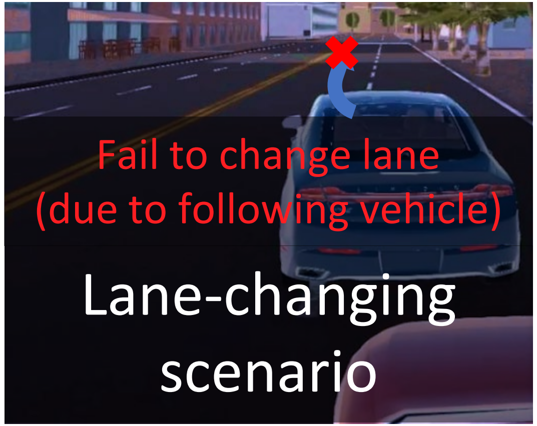
More evaluations in the paper…

Exploitation case studies



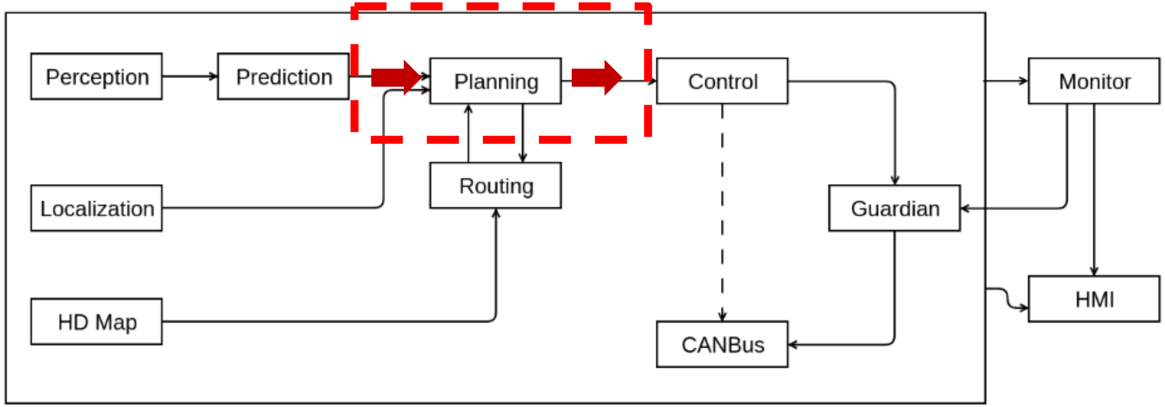






Limitations and Future Work

- **Testing Method: E2E vs Module Testing**
  - Result from module testing ≠ real-world vulnerability



- **Input Generation**
  - 8 driving scenarios with 40 driving traces
  - Uncovered scenario still exists.. (etc. Emergency scenarios in Baidu Apollo)

Conclusion

First to perform AD planning-specific semantic vulnerability discovery with **a domain-specific vulnerability definition** and **a practical threat model**

- Design *PlanFuzz*, a **novel dynamic testing** approach that addresses various problem-specific design challenges
- We evaluate *PlanFuzz* on **two** practical open-source **full-stack** AD systems and discover **9** previously-unknown DoS vulnerabilities
- Perform exploitation case studies of **3 diverse driving scenarios** with simulation and driving traces collected from **a real AD vehicle**
- Inform **24 companies** developing AD vehicles

Thank you!

*For **more demos, source code release, and other details,***
Please visit our project website:

https://sites.google.com/view/cav-sec/planfuzz

Scan to visit our project website

AS^{2}Guard Autonomous & Smart Systems Guard Research Group

UCI

UCLA