Cybersecurity in Autonomous Driving

Yue Duan

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

The rise of Autonomous Driving (AD) vehicles

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

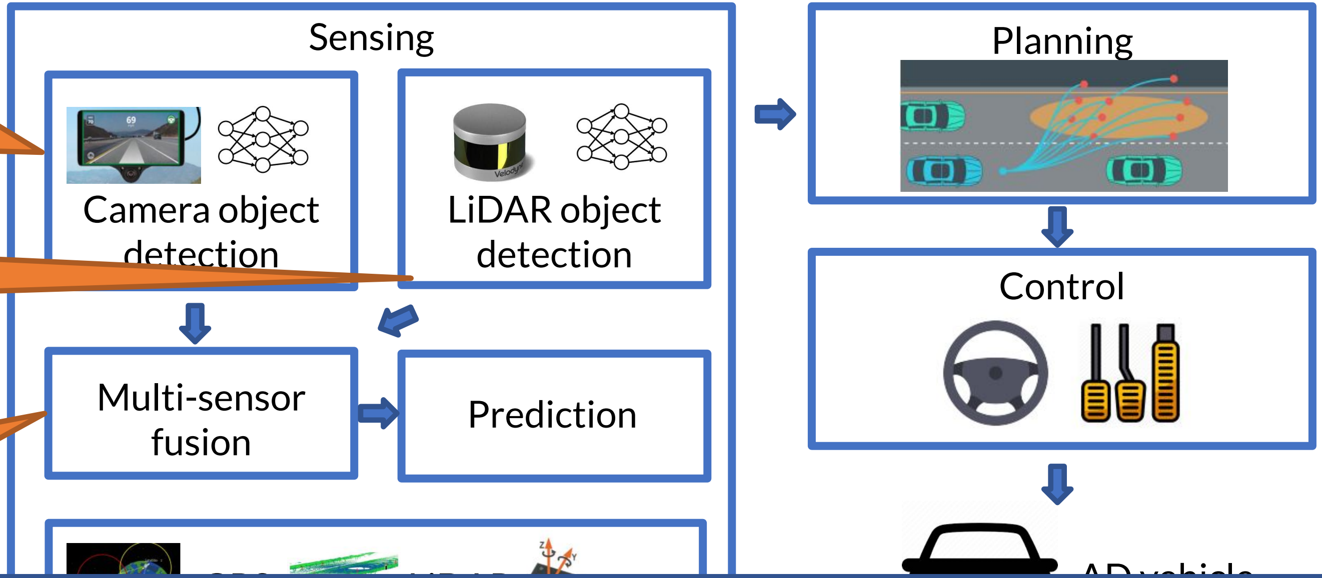
Current status of AD security research

- 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

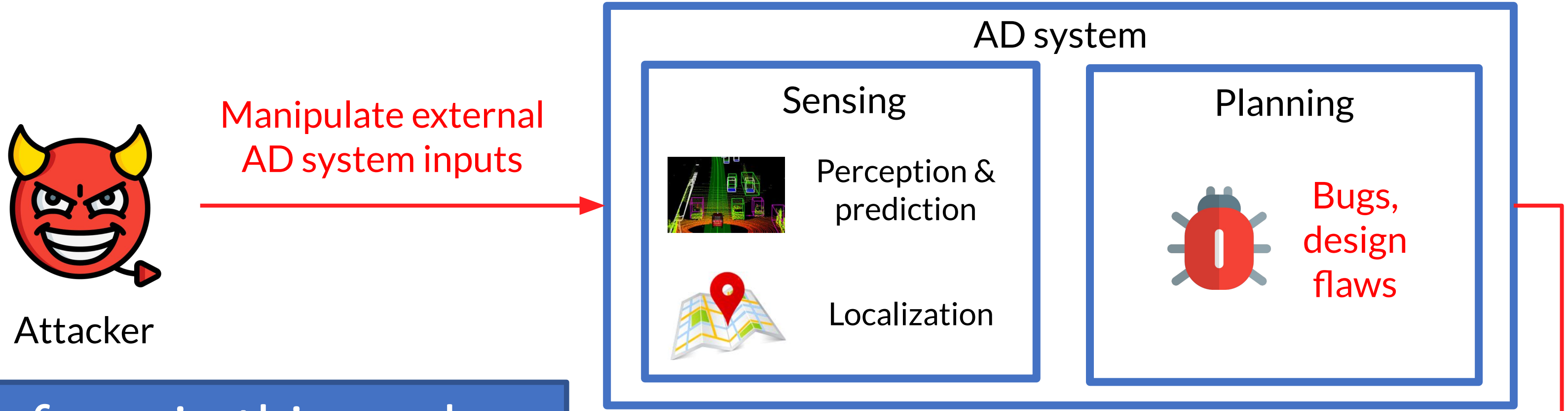
Cao et al. Oakland'21



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 <u>*unexpected*</u> one

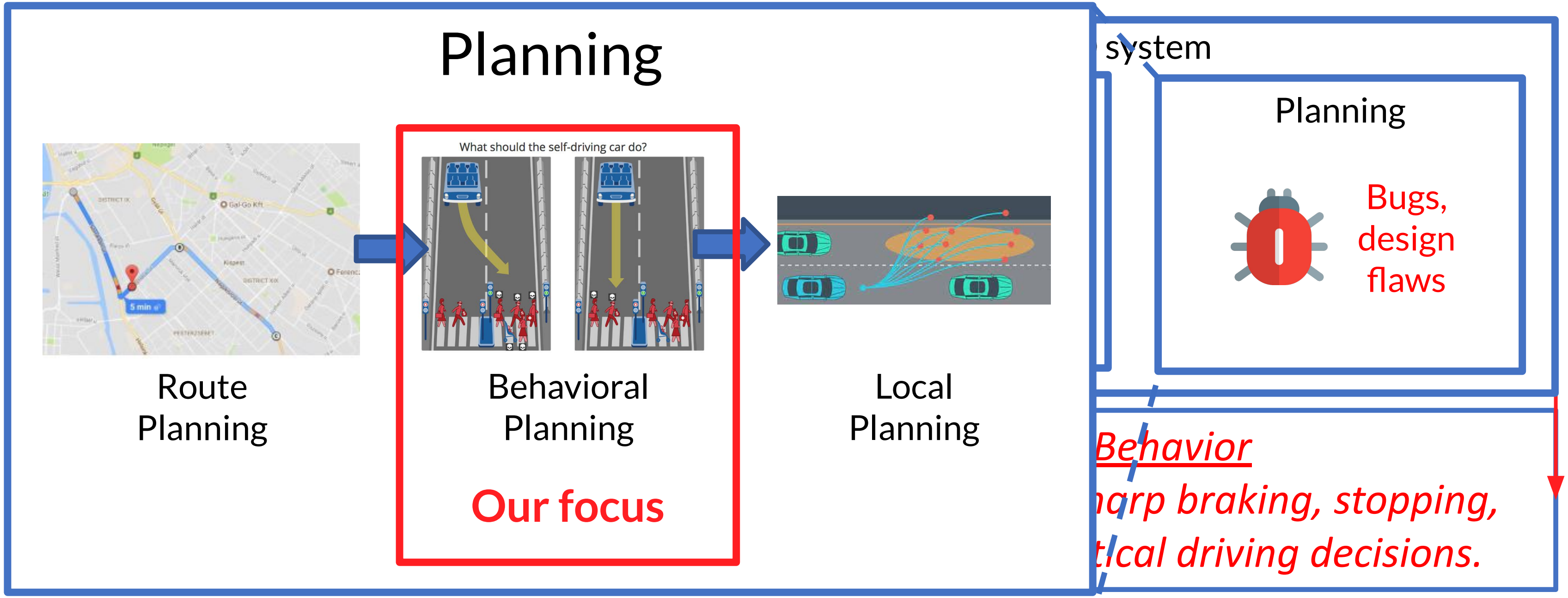


Our <u>*focus*</u> in this work.
Also referred to as
<u>*semantic DoS vulnerability*</u>

<u>*Overly-Conservative Behavior*</u>
- e.g., unnecessary sharp braking, stopping, giving up mission-critical driving decisions.

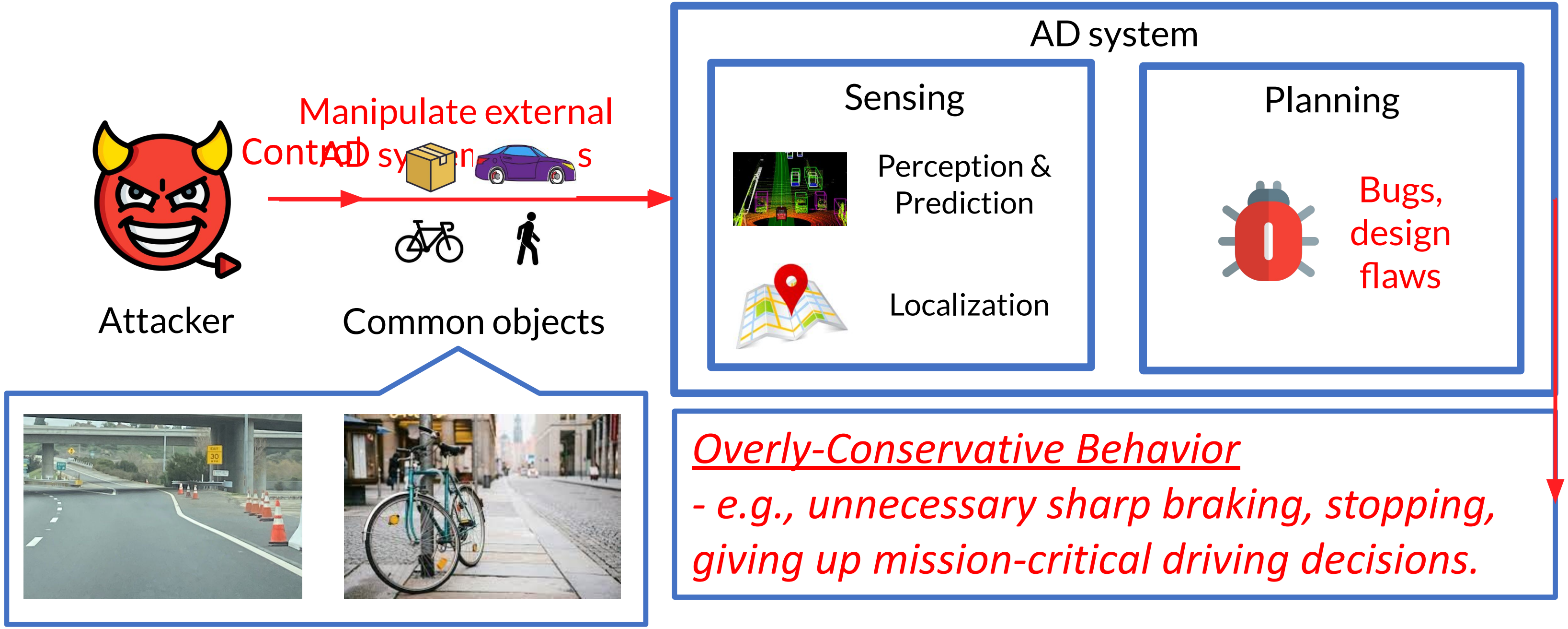
Our target: Behavioral Planning (BP)

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



Threat model

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



Consequence of semantic DoS vulnerability

Consequences

Bad user experience

Safety

Block traffic

Law violation in specific places

<u>Overly-Conservative Behavior</u>

- 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?

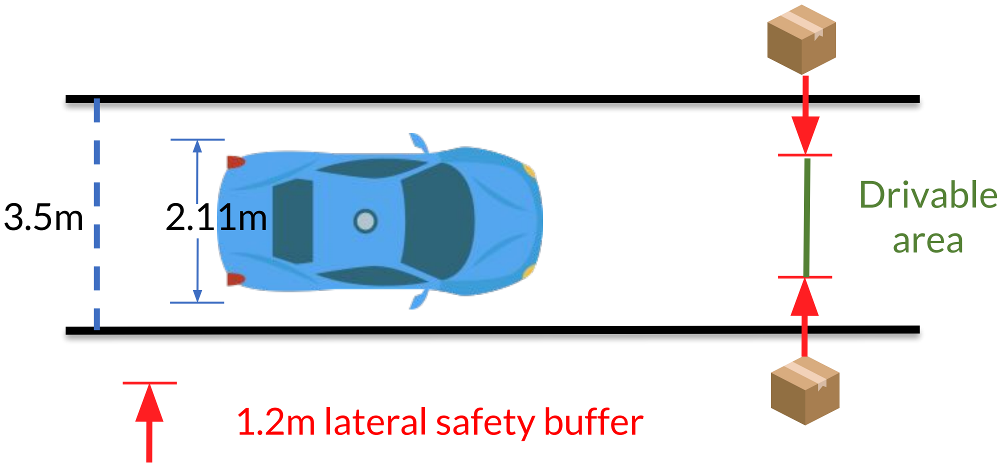
- Ignore them?
- Slightly slow down?

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

Two pictures around our campus.

Attack scenario setup

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

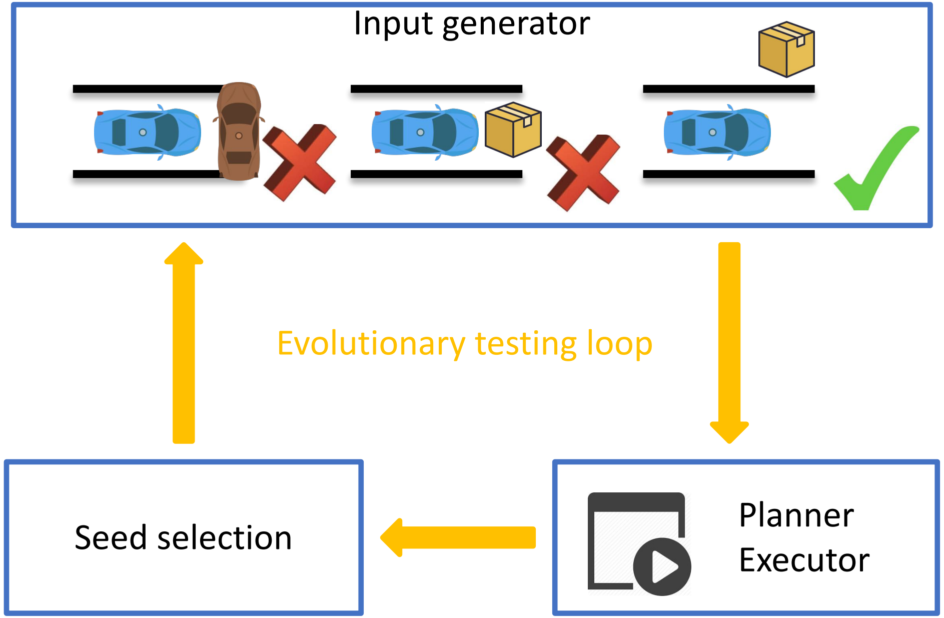
PlanFuzz: Design challenges

- 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?

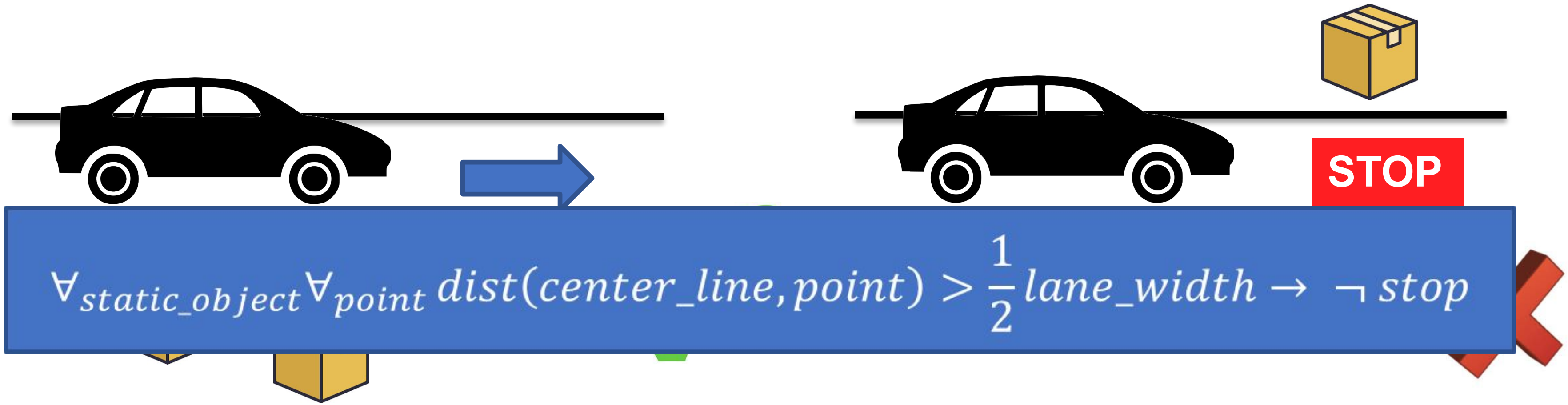
Challenge 1: How to judge a driving decision is *overly-conservative*?

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



Solution: Planning Invariant (PI)

- To address challenge 1 (lack of testing oracles for semantic DoS vuln), we design planning invariant
  - Planning Invariants (PI) = <u>planning scenario</u> + <u>desired planning behavior</u> + <u>attacker-controllable changes</u>
  - **Systematically** define PIs under 8 diverse scenarios with **<u>temporal logic</u>** to constraint static objects, and **moving** pedestrian/vehicles

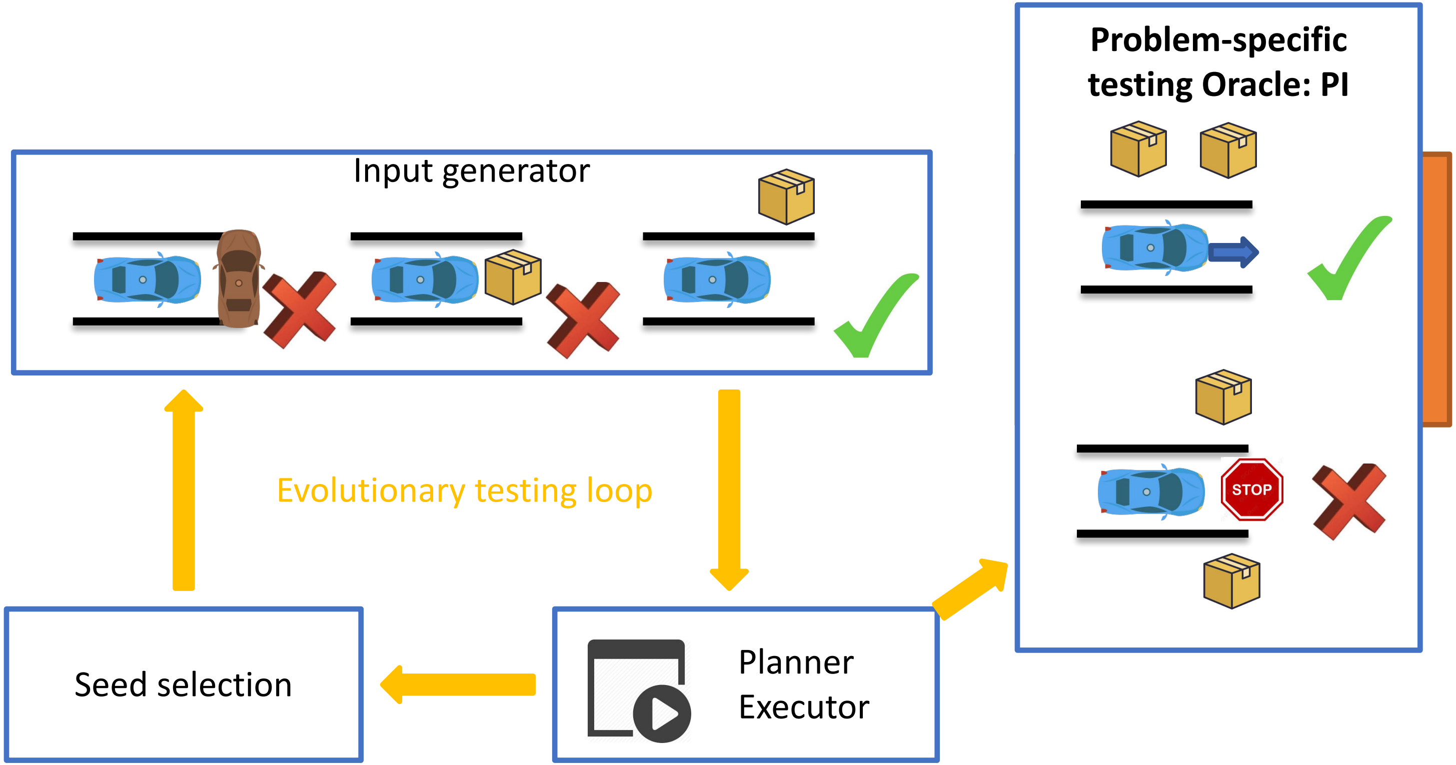


$$\forall_{static\_object} \forall_{point}\, dist(center\_line, point) > \frac{1}{2} lane\_width \rightarrow \neg stop$$

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?



Solution: PI-aware physical-object generation

<u>Input generation:</u>

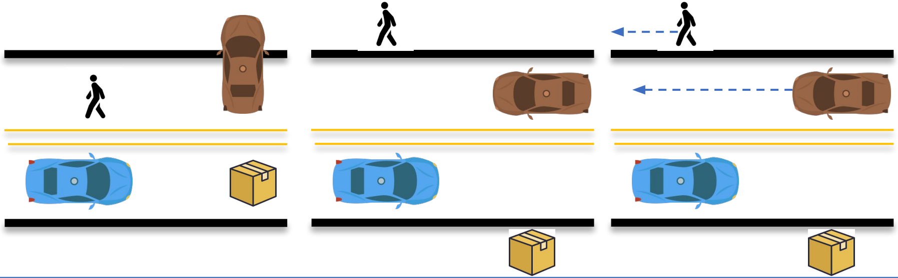
- Satisfy domain-specific constraints
- Maintain diversity and inheritance during mutation

PI-aware physical-object generation

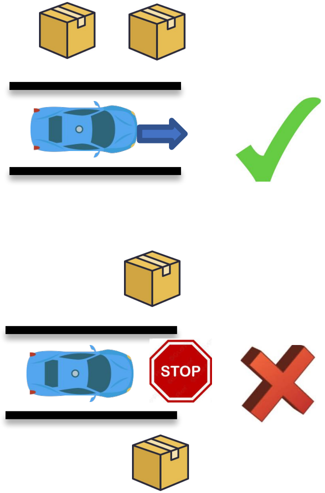
Static property generation

PI-constraint enforcement

Dynamic property generation



Problem-specific testing Oracle: PI



<u>Challenge 3:</u> How to design feedback to efficiently guide the testing?

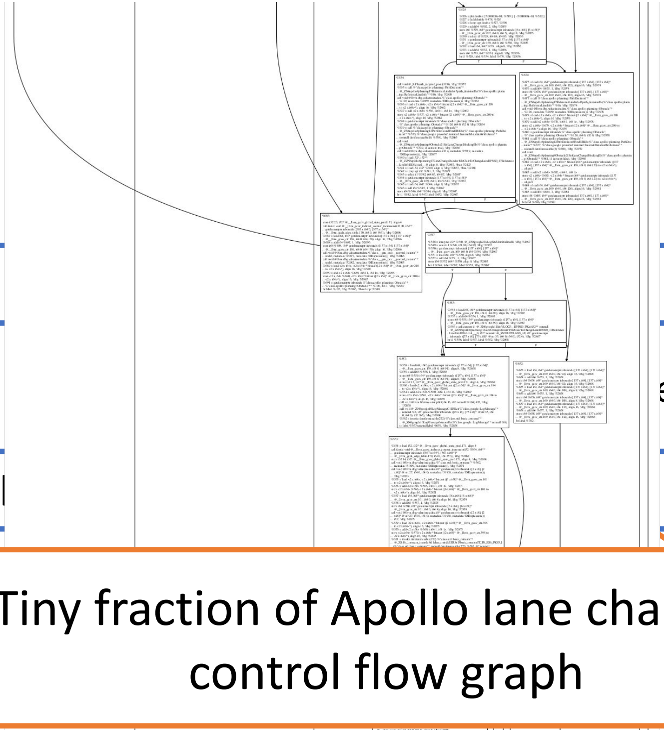
Evolutionary testing loop

Seed selection

Planner Executor

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

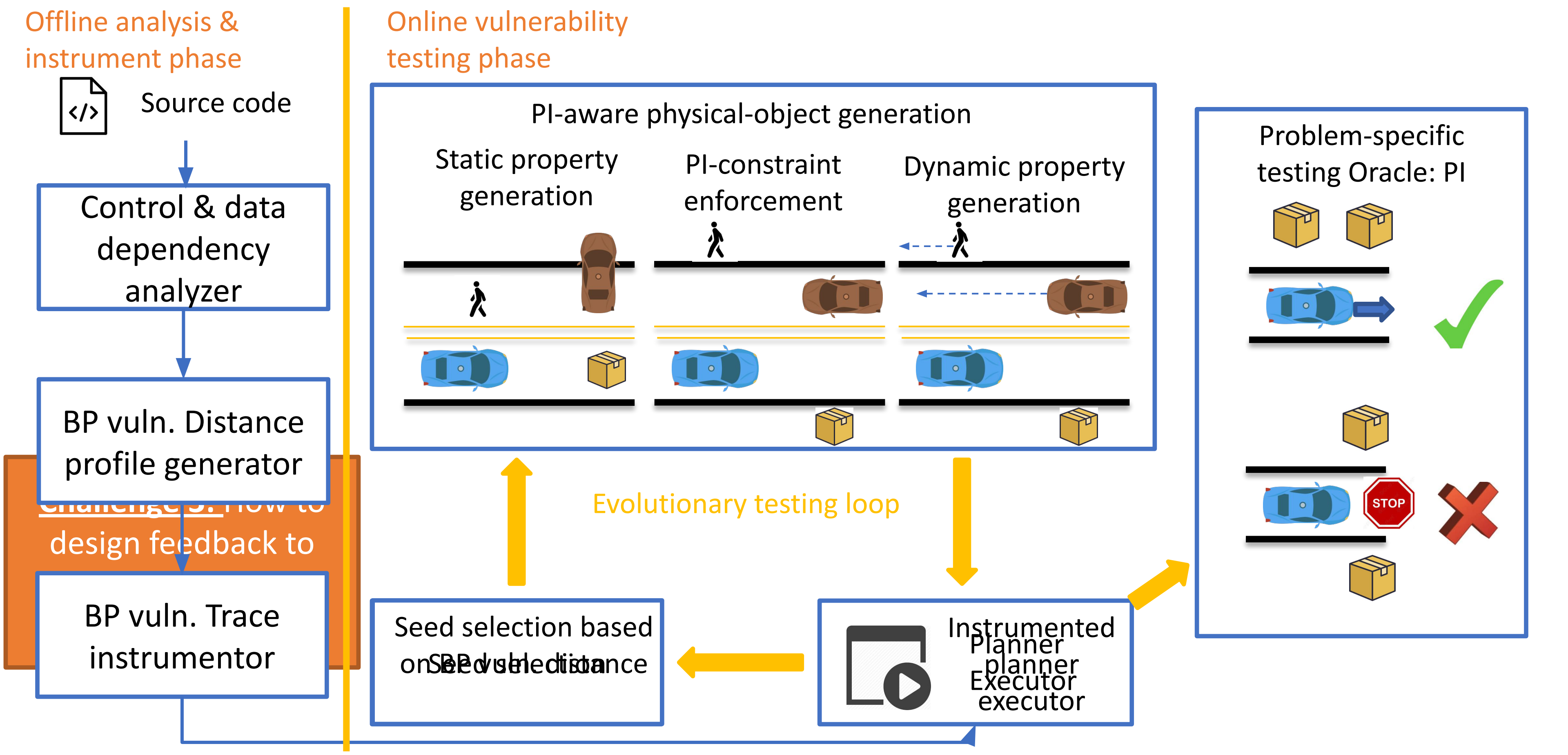
Offline static analysis:

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

Online dynamic analysis:

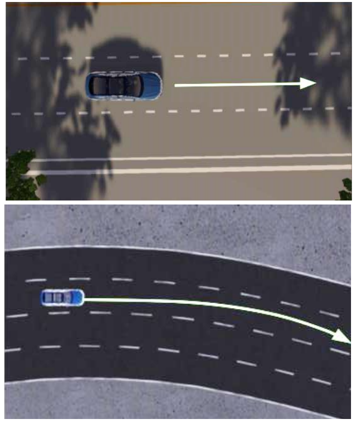
- Calculate BP vuln. dist. at runtime

Solution: BP vulnerability distance

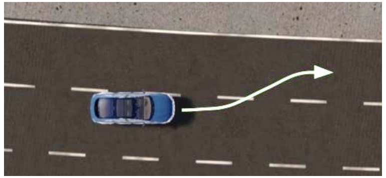


Evaluations: DoS semantic vulnerabilities discovery

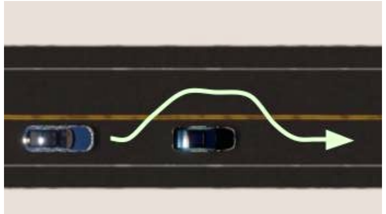
- **9 previously unknown** semantic DoS vulnerabilities from **3 BP implementations** of Baidu Apollo and Autoware.AI (full-stack open-source AD software)
  - Causes: 1 due to implementation bug, 8 due to overly-conservative planning parameters (e.g., safety buffer, angle threshold) & overly-conservative 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

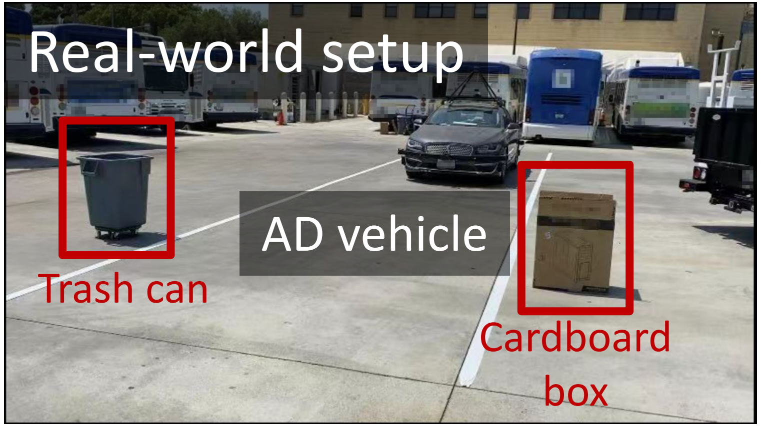


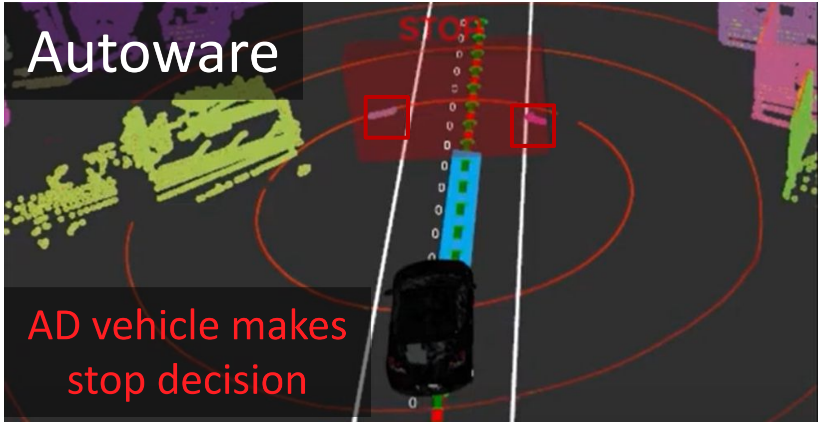
Lane borrowing

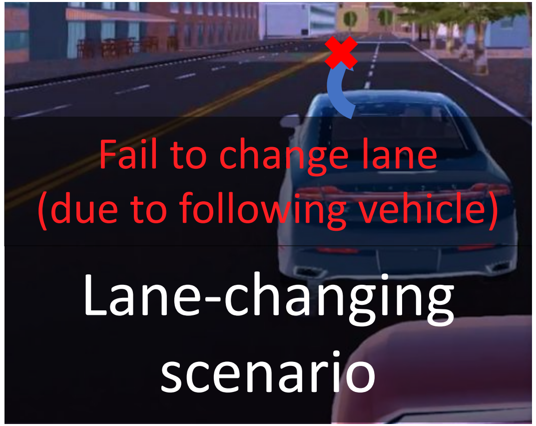
Intersection passing

More evaluations in the paper…

Exploitation case studies







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!

Question?