Empirically Evaluating Flaky Tests for Autonomous Driving Systems in Simulated Environments

Olek Osikowicz, Phil McMinn, Donghwan Shin

University of Sheffield

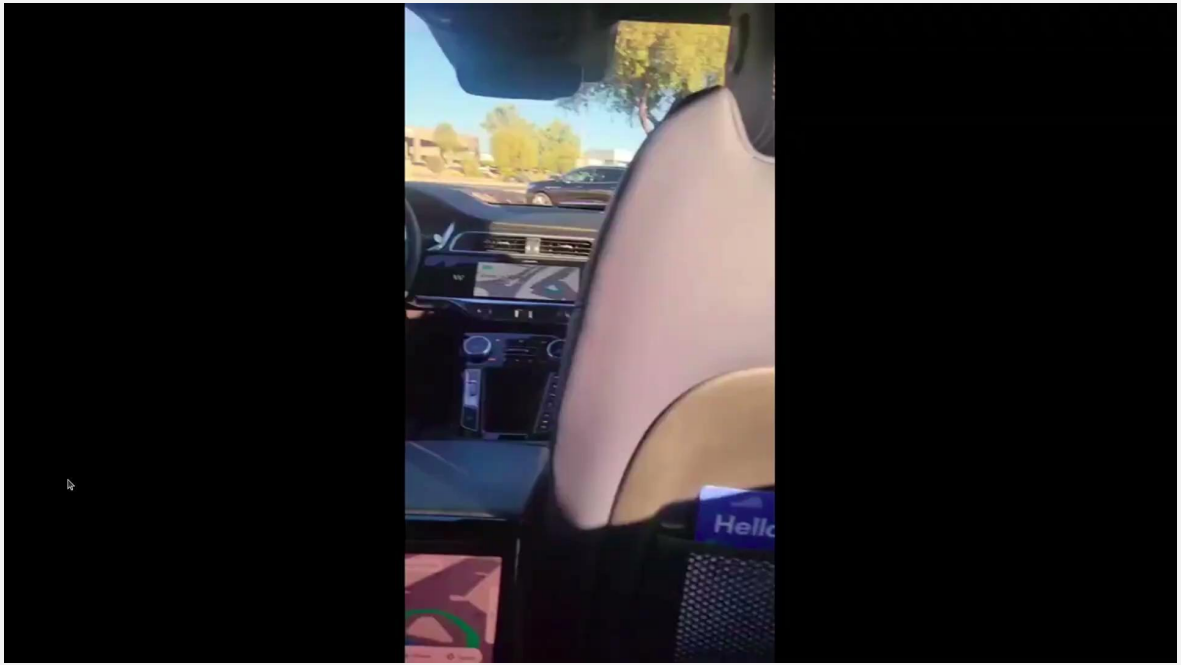
Agenda

- Introduction & Background
- Case study CARLA
- Case study MetaDrive

A bit of context

Why should we test Autonomous Driving Systems (ADS)?

- Autonomous driving systems are far from perfect:



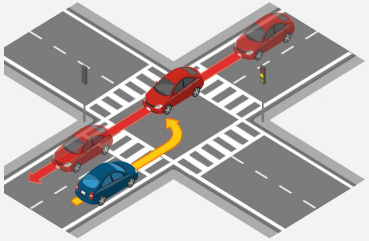
[Man stuck inside circling humanless Waymo car]

Why should we test ADS in simulation?

- Cost-effective
- Safe and convenient
- **Repeatable!**

Scenario-based ADS Testing overview

Scenario based testing



Driving Scenario

Simulation

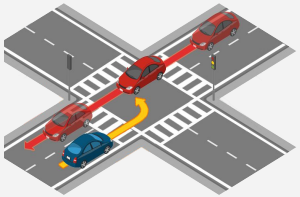
Scenario execution

ADS

Sensor data

Driving command

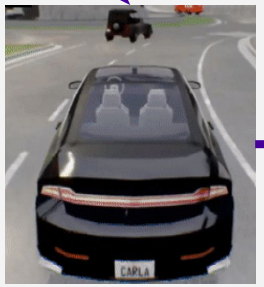
Scenario based testing



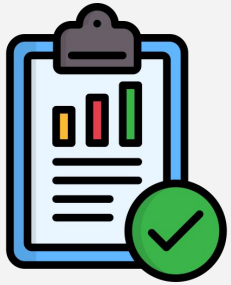
Driving Scenario

test case

Simulation



ADS

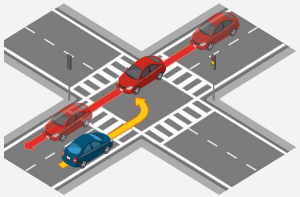


Evaluation Results

test outcome

Scenario based testing

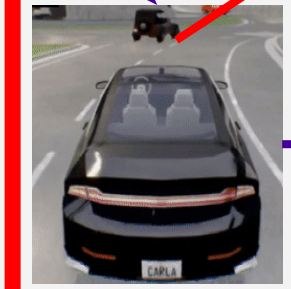
System under test
(deterministic)



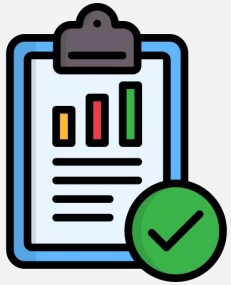
Driving
Scenario

test case

Simulation



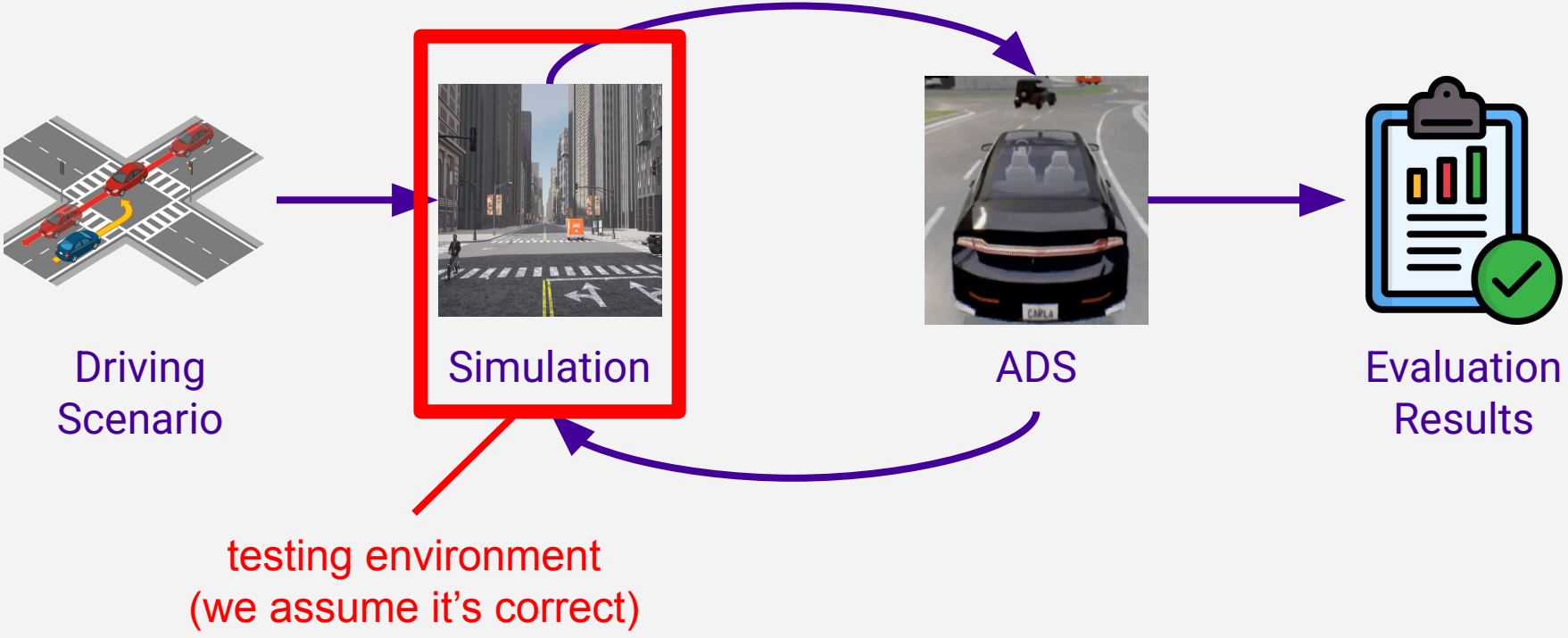
ADS



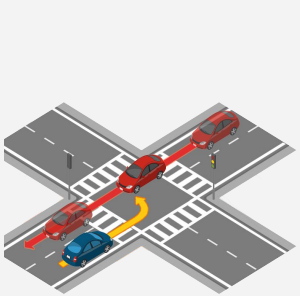
Evaluation
Results

test outcome

Scenario based testing

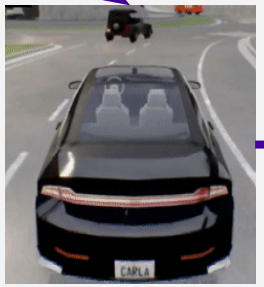


Assumption: simulator is deterministic

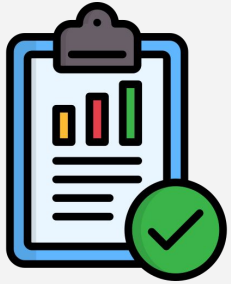


Driving Scenario

Simulation



ADS



Evaluation Results

Same input scenario should yield same evaluation results!

What if simulator is not deterministic?

- Great chance of flaky tests!
- i.e. a test scenario can pass or fail without any changes to the ADS under test

Research questions

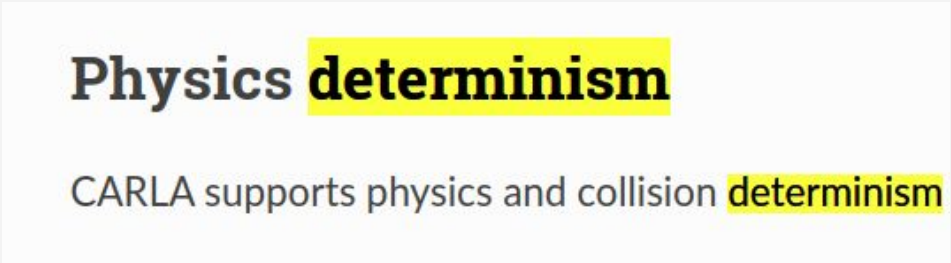
- **RQ1:** How many test scenarios are potentially flaky due to simulators nondeterminism?

- **RQ2:** How flaky are the test results of the potentially flaky test scenarios?

Case study: CARLA

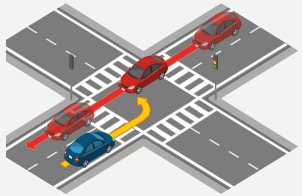
What is CARLA?

- An open-source simulator widely used in the research community
- Claims to be deterministic:

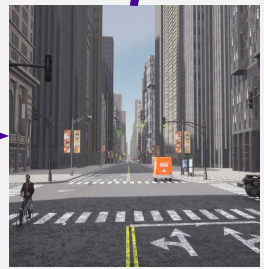


Source, https://carla.readthedocs.io/en/latest/adv_synchrony_timestep/

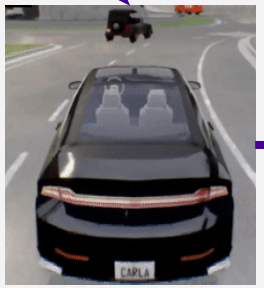
Methodology (1/3)



128 Scenarios



CARLA



TransFuser++

Safety violations

Repeat 10 times

Methodology (2/3)

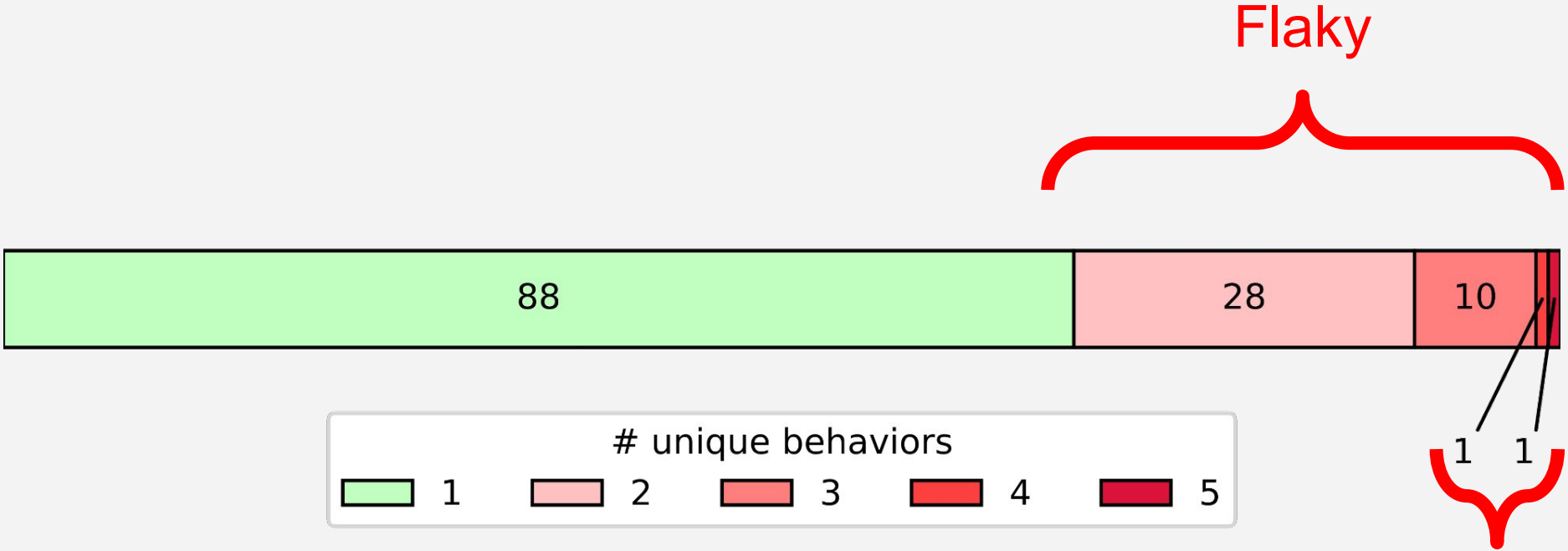
- During scenario execution we count following violations:
  - Collisions with other vehicles, pedestrians and static objects
  - Running a red light/ stop sign
  - A Time-out (car can’t reach the goal in expected time)
  - Vehicle blocked (car can’t move due to deadlock)

Methodology (3/3)

- ADS behaviour for same scenario should be **the same**

- If ADS committed same set of violations ⇒ same behaviour
- Then for each scenario we count number of unique behaviours
- If more then one unique behaviour exhibited ⇒ flakiness

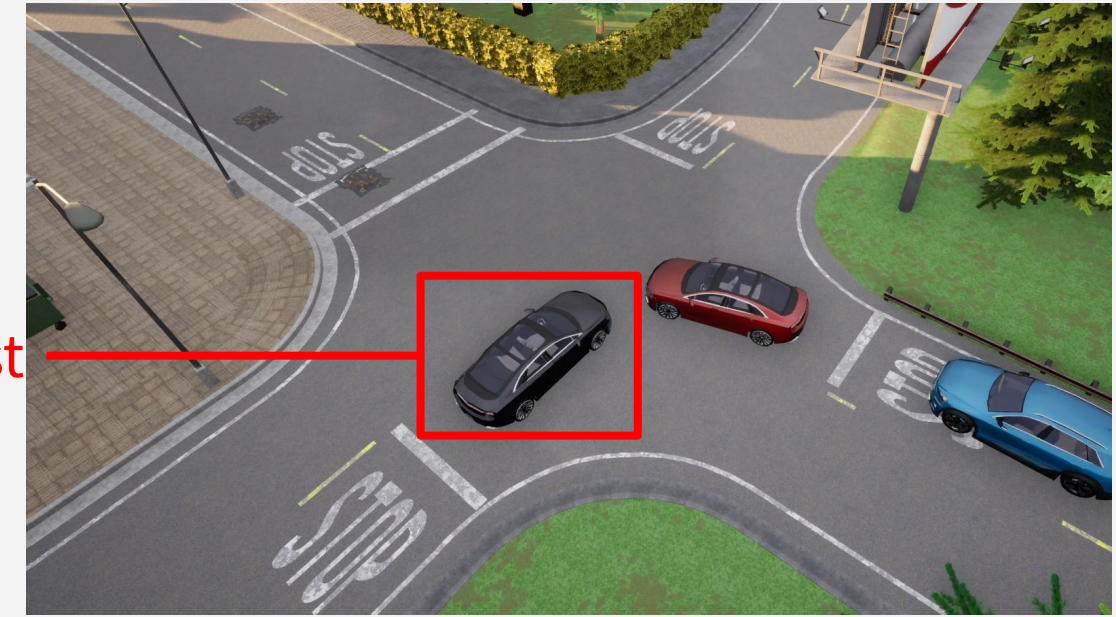
CARLA's result RQ1



What are those? 🤔

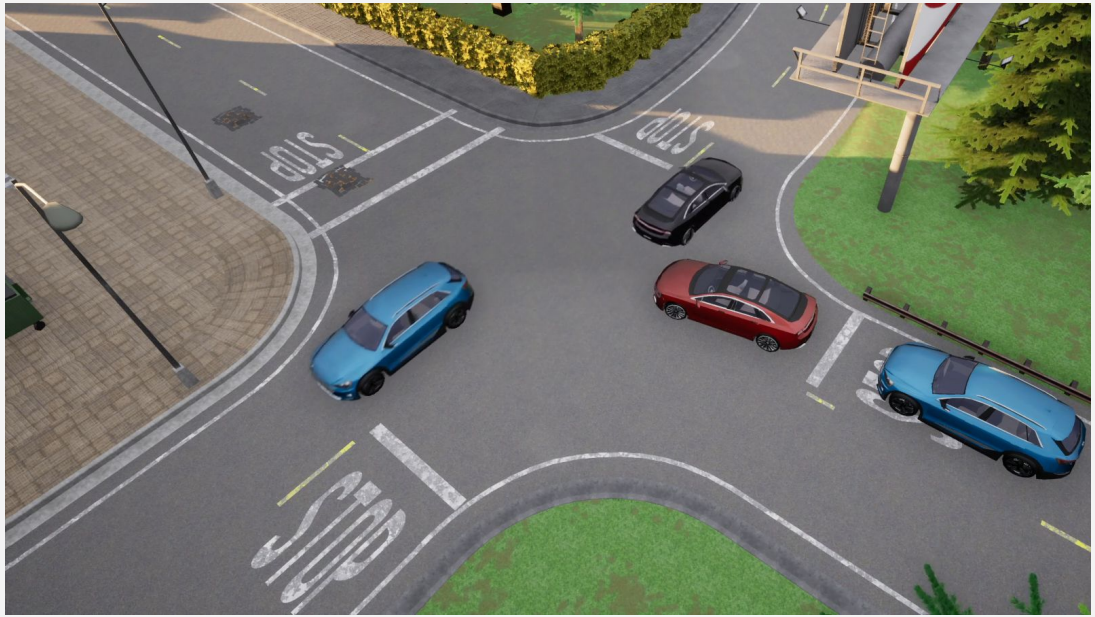
Manually investigating flaky scenarios

- Single scenario 4 different behaviours! - deadlock



Manually investigating flaky scenarios

- Single scenario 4 different behaviours! - pass safely



Manually investigating flaky scenarios

- Single scenario 4 different behaviours! - collision

Manually investigating flaky scenarios

- Single scenario 4 different behaviours! - deadlock & collision

New type of flakiness: simulator flakiness

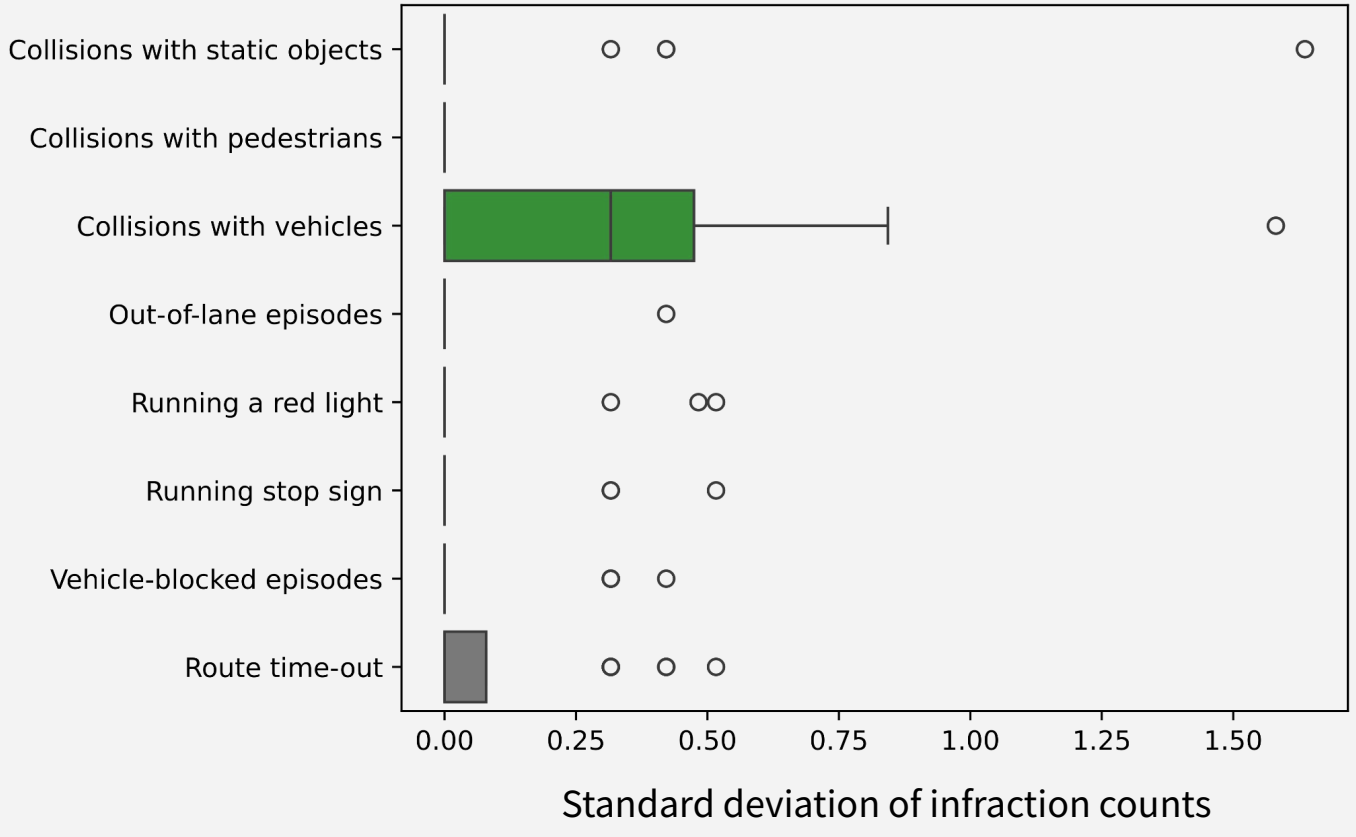
- It’s not the test definitions that are causing flaky behaviours…
- It is the simulation itself!
- Therefore its: **simulator flakiness**

Methodology RQ2

- **RQ2:** How flaky are the test results of the potentially flaky test scenarios?

- Degree of flakiness of each infraction type:
  - standard deviation of each violation counts

Results RQ2

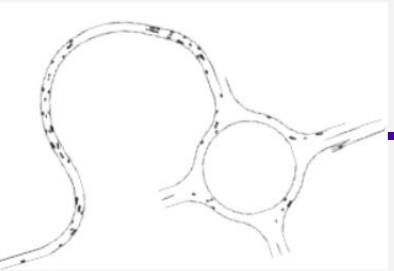

Collisions with static objects
Collisions with pedestrians
Collisions with vehicles
Out-of-lane episodes
Running a red light
Running stop sign
Vehicle-blocked episodes
Route time-out
0.00
0.25
0.50
0.75
1.00
1.25
1.50
Standard deviation of infraction counts

Results RQ2

- Most flaky safety violation: “collision with vehicle”
  - one of the most important requirements
- Highlights importance of considering nondeterminism in the driving simulators

Case study: MetaDrive

Similar Methodology



200 Scenarios

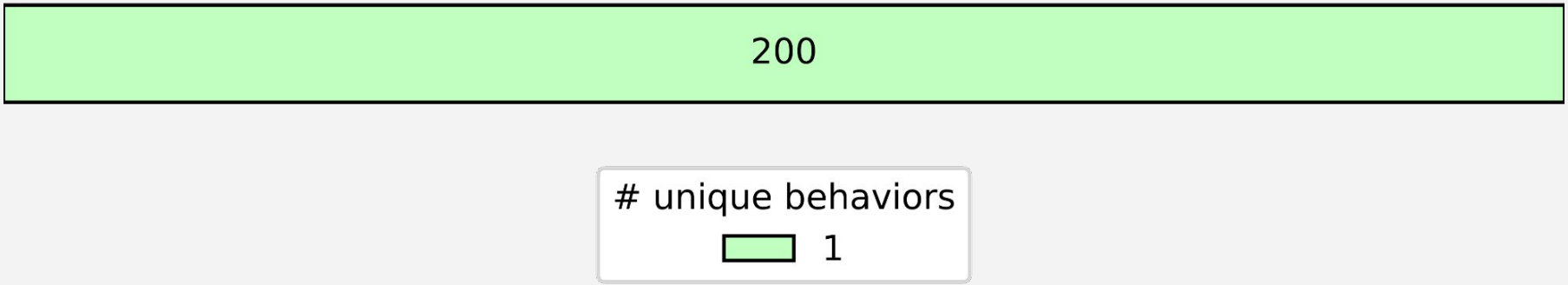
MetaDrive

MetaDrives' expert

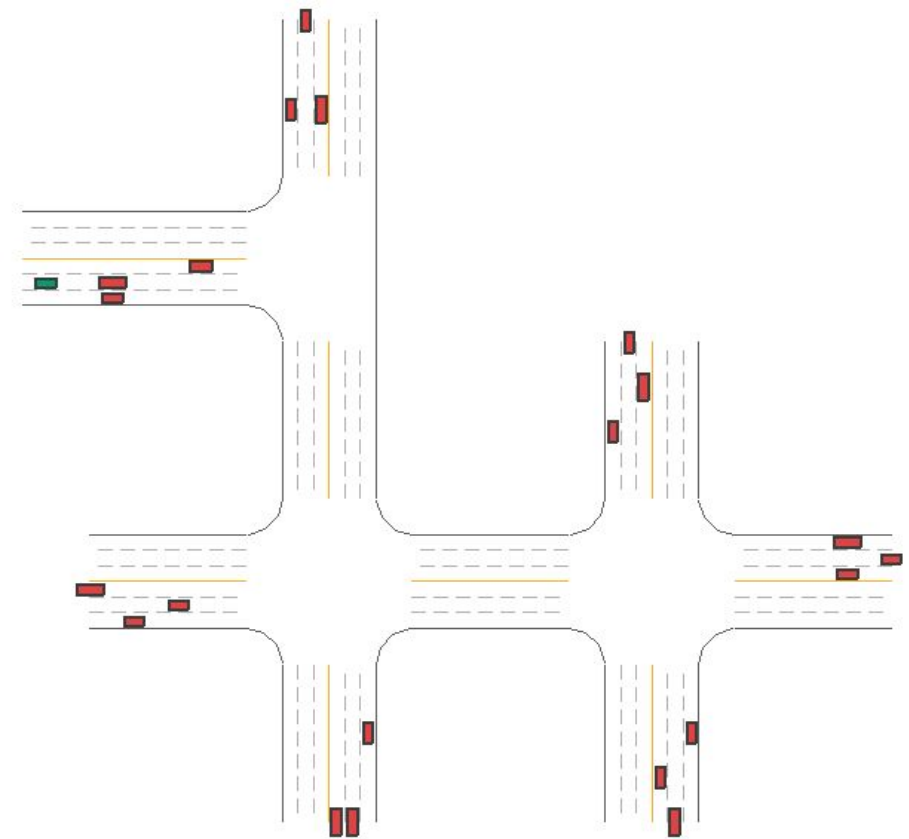
Safety violations

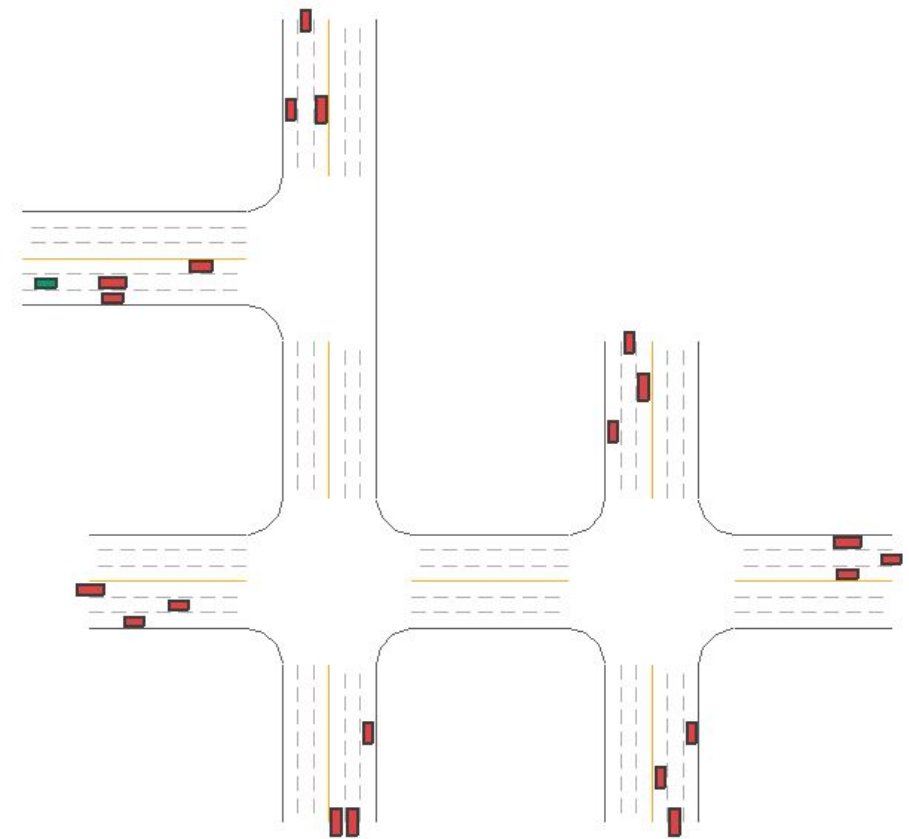
Repeat 10 times

MetaDrives result RQ1



MetaDrive's perfect determinism 💪





Discussion

General mitigation strategy

1. **Acknowledge** flakiness: “simulators can be flaky”
2. **Prepare** for flakiness:
    - Latest updates, fixed simulation time step, seeded experiments e.t.c.
3. **Check** for flakiness:
    - Repeatedly run same scenario and assess the variance
4. **Respond** to flakiness: e.g. results should be compared using statistical test

Conclusions

Summary

- We Empirically evaluated the determinism of CARLA and MetaDrive
  - CARLA has intrinsic nondeterministic
  - MetaDrive is capable of deterministic simulation
- Identified new type of flakiness: **simulator flakiness**
- Provided guidelines to mitigate potentially flaky tests in simulation-based ADS testing

Any questions?

University of Sheffield