# Recent advances on testing autonomous vehicle systems: an industry perspective

Jonathan Sadeghi

7 September 2023

Hi-Drive summer school, Porto Heli, Greece

FIVE BOSCH

# A brief history of Five


Five was started in 2016:

- Raised 62M EUR from VCs.
- One of Europe's largest startups in automated driving.
- Strong academic links, particularly with University of Oxford and University of Edinburgh

- Circa 130 staff
  - Comp Sci and other STEM
  - Many PhDs
- Offices in: Cambridge, Bristol, Edinburgh, London, Oxford

**Acquired by Bosch in June 2022**




FIVE BOSCH

# Phase 1: Five's Prototype Automated Driving System
## Fleet of 8 Ford Fusion vehicles

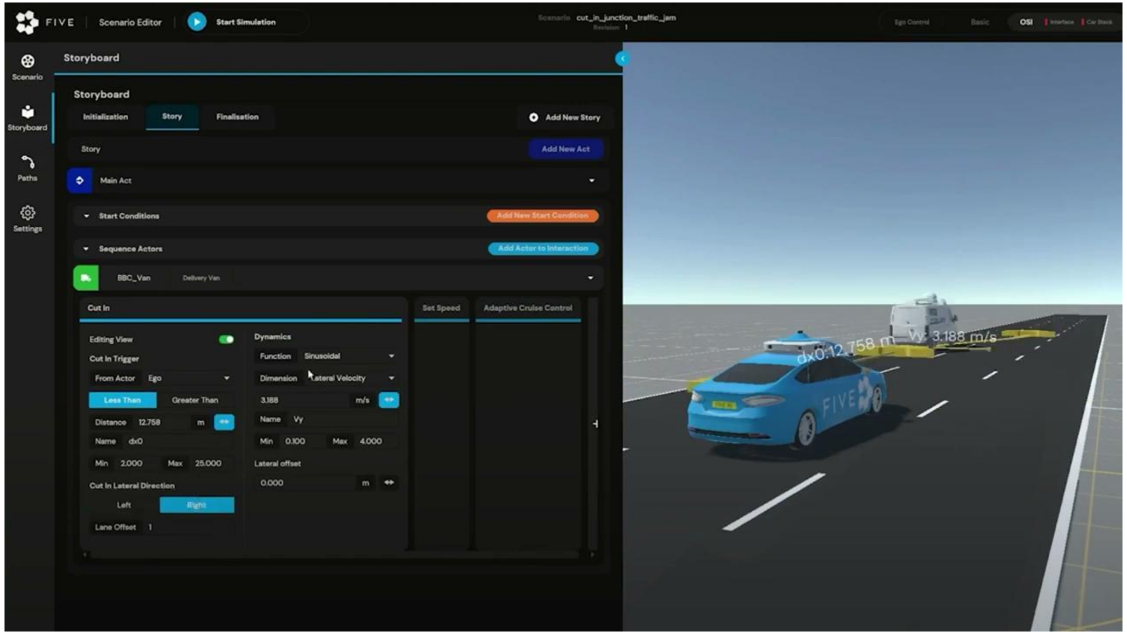

- GPUs
- Sensors
  - Cameras
  - Lidar
  - Radar
  - GPS
  - IMUs, wheel encoders, etc.

# Why do we test in simulation?

- AV systems are *complex* and potentially have many *different failure modes*

- Debugging these in the real world is prohibitively time consuming and expensive
  - Typically $10^9$ hours between failures in the aerospace industry — testing by “brute force” for an AV with an equivalent mean failure time, i.e. driving for $10^9$ hours to mine each failure, is infeasible [6]

- Solution: <u>develop in simulation!</u>



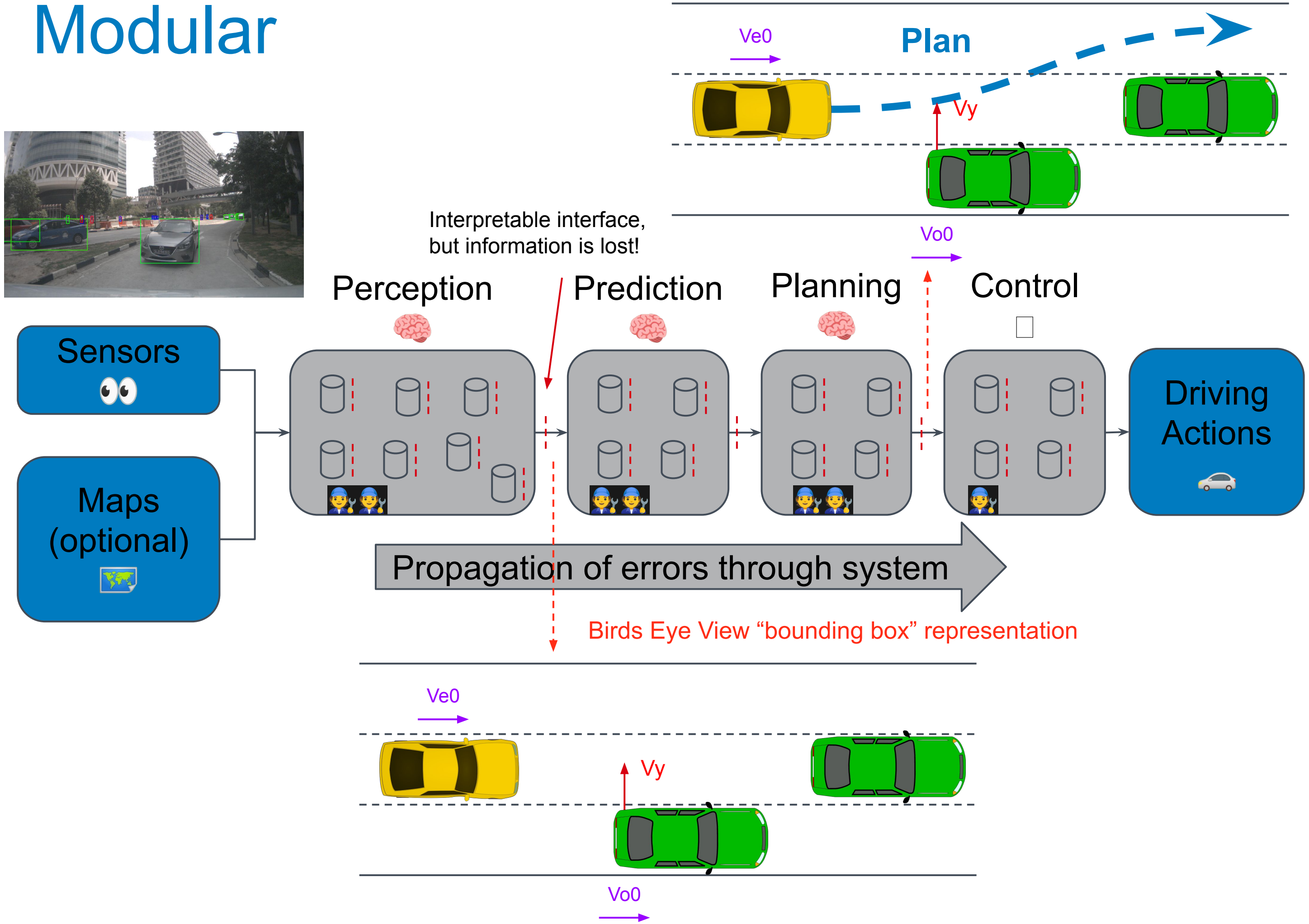

How do we build a good testing system?




FIVE BOSCH

# Self-driving systems architecture types

## Modular




### Pros

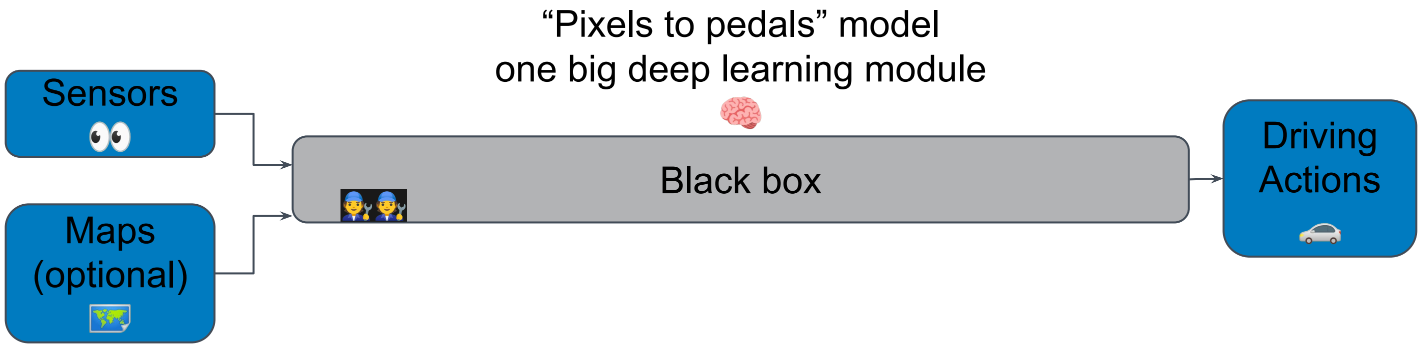
- *Interpretable* ⇒ easier to debug
- *Verifiable* planners can be used
    - (as an alternative to black-box differentiable planners)
- *Divide and conquer*: development of components can be parallelised between teams!

### Cons

- Information is lost at interface between modules
    - e.g. pedestrian head orientation indicates likely future motion but this is lost if using bounding boxes
- Overall performance can suffer as a result compared to end-to-end approaches

# Self-driving systems architecture types

## End to end



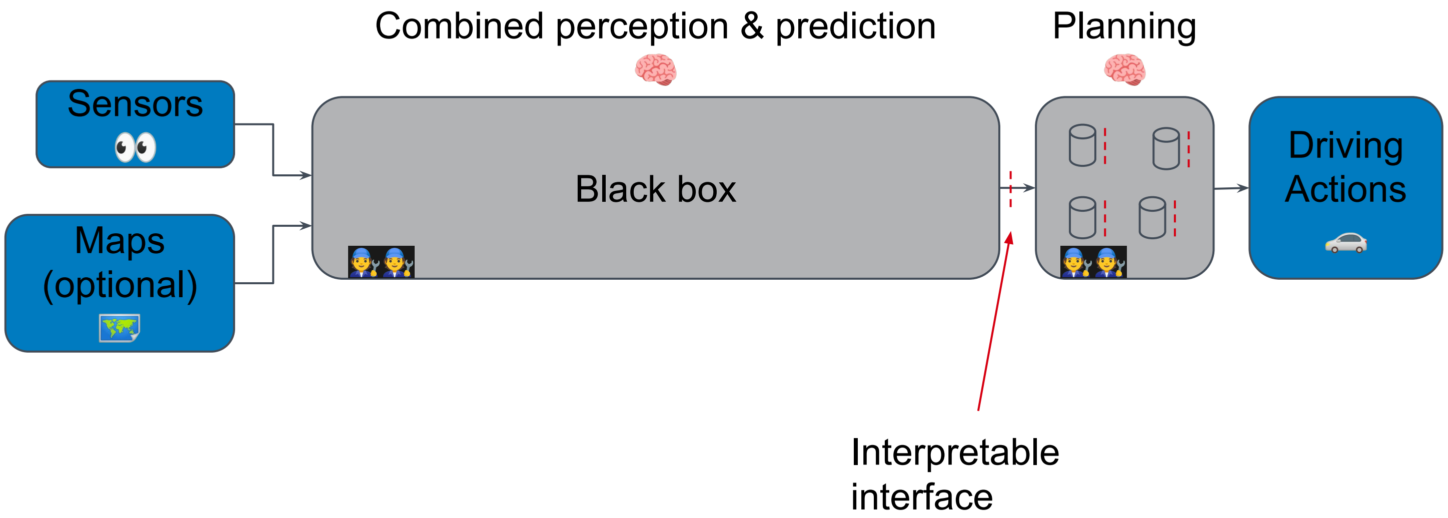

**Pros**

- *No lossy interfaces* information can be preserved throughout the stack leading to better performance (maybe)
- *Fewer components*: perhaps easier to develop with smaller teams
- For these reasons most successful carla leaderboard competition entries use this style of system

**Cons**

- *Less Interpretable* ⇒ difficult to debug
- *Fewer components*: more difficult to share work between teams in larger organisations

# Self-driving systems architecture types
## Hybrid - the best of both worlds




**Pros**

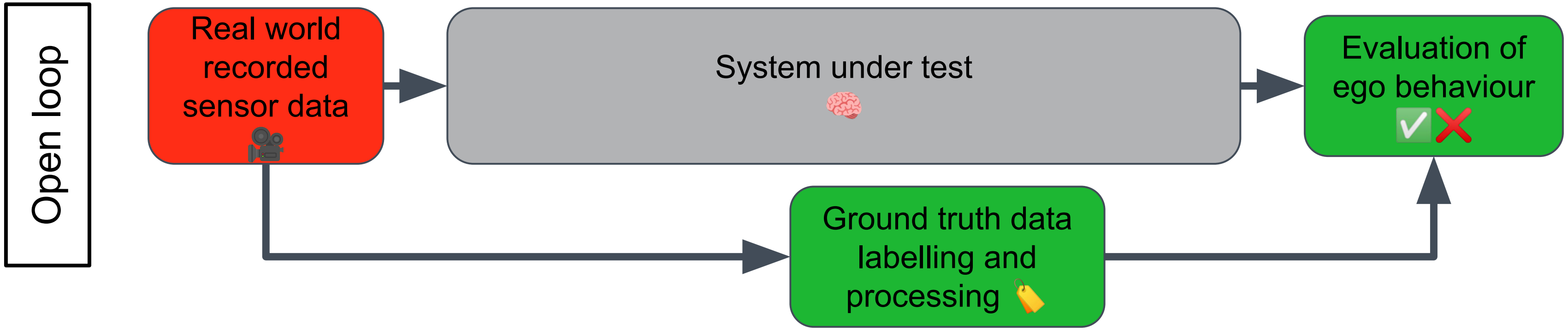
- *Fewer lossy interfaces* information can be preserved throughout the stack leading to better performance (maybe)
- *Verifiable* planners can be used
- *Fewer components*: perhaps easier to develop with smaller teams
- For these reasons this approach is becoming more common in many companies

**Cons**

- *Less Interpretable* ⇒ difficult to debug
- *Fewer components*: more difficult to share work between teams in larger organisations

# Testing styles

## Open loop vs closed loop?

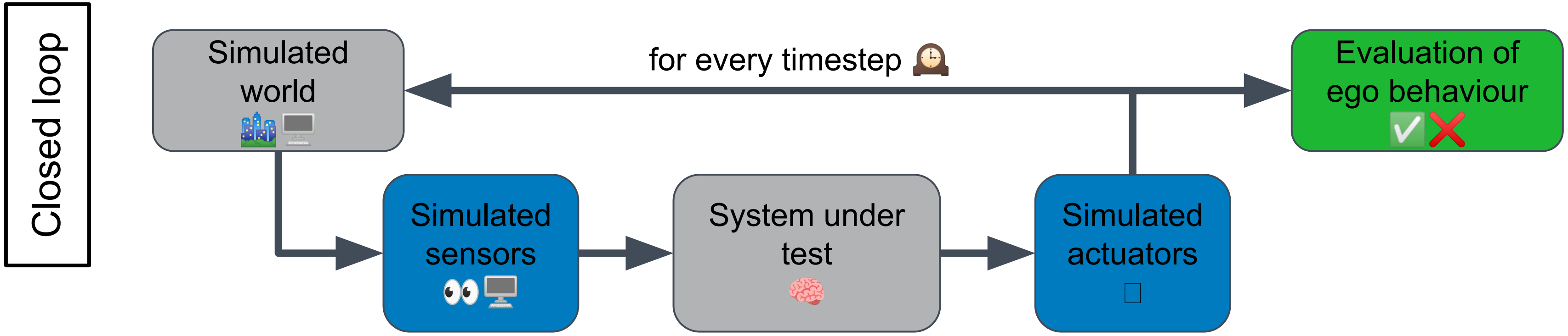



**Pros**

- *Simpler* to implement
- Every timestep evaluated independently

**Cons**

- *Less realistic*: no interaction between agents, etc.




**Pros**

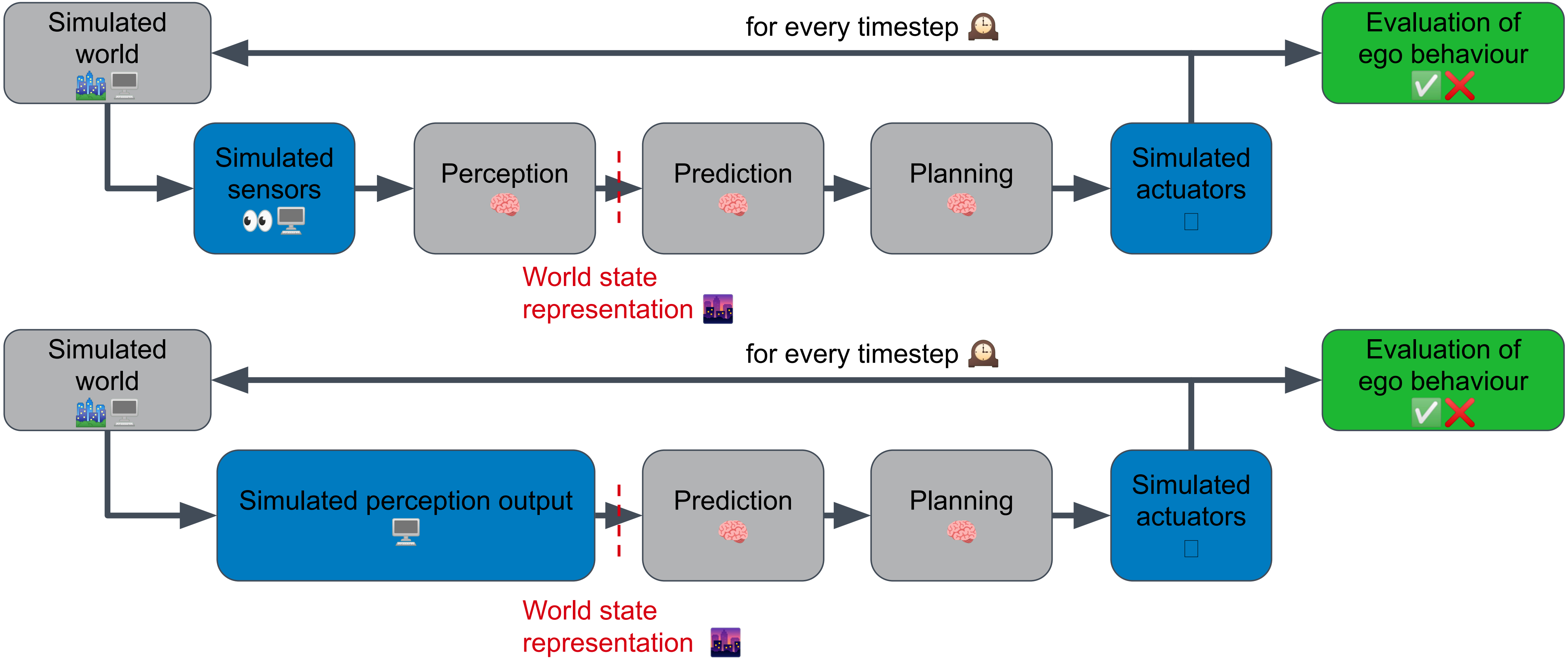
- *More realistic*: errors accumulate over time

**Cons**

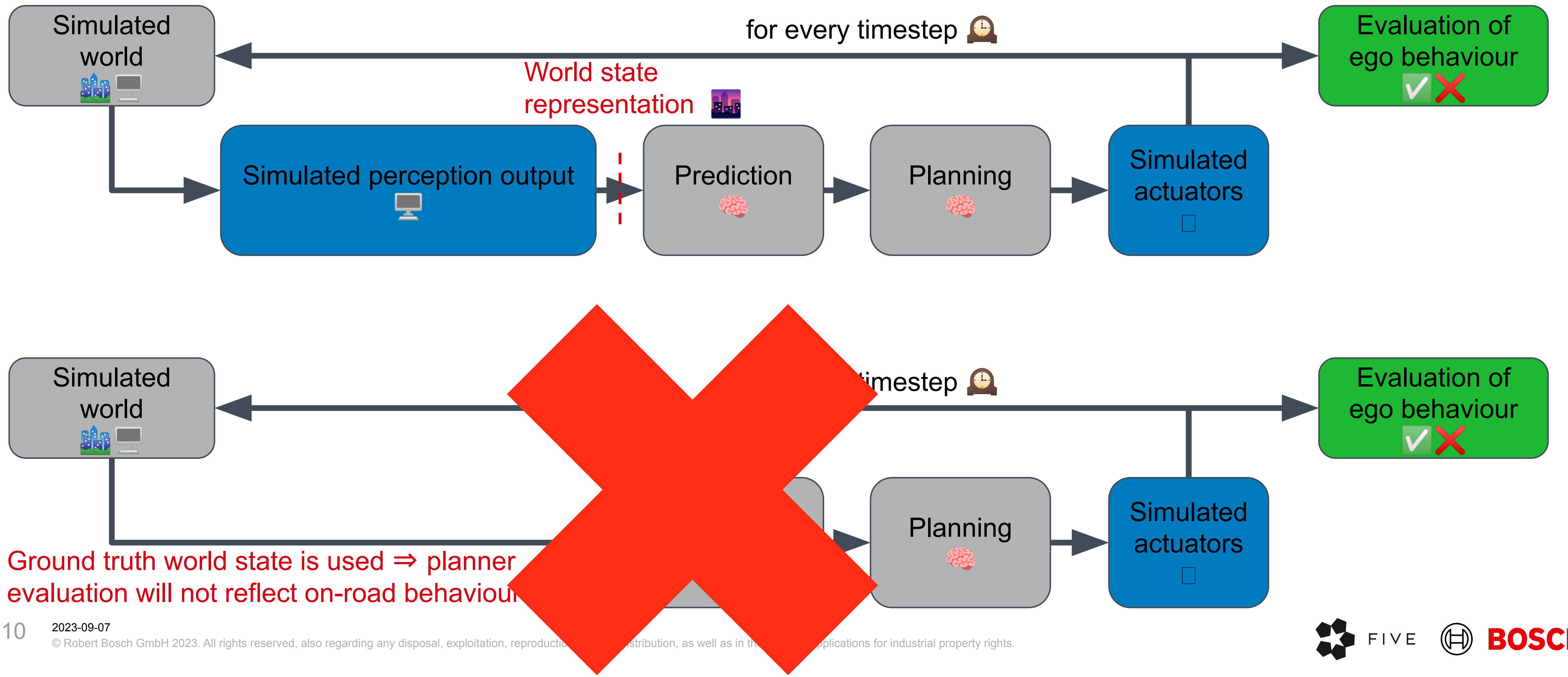
- *Greater complexity* and computational expense
- *Problematic*: we will see that realistic sensor simulation is difficult!

# Testing styles

## Exploiting modularity of system - closed loop



# Testing styles
## Exploiting modularity of system - closed loop





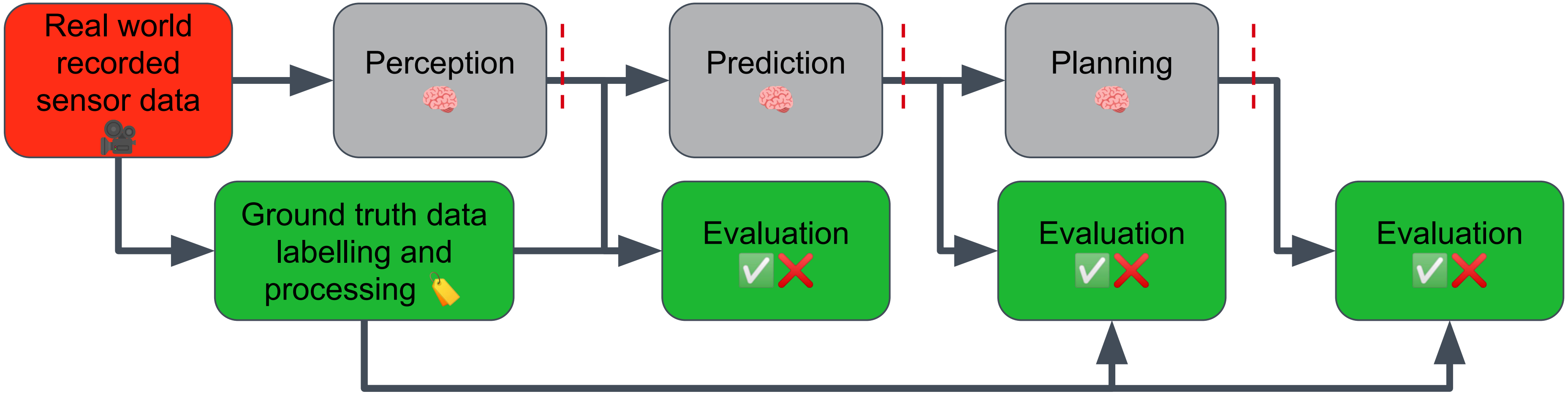
2023-09-07


FIVE BOSCH

# Testing styles

## Exploiting modularity of system - open loop

All module outputs here are human interpretable and can be easily evaluated against ground truth





2023-09-07

FIVE BOSCH

# Testing styles
## Summary

- We can test systems using
  - Open loop evaluation (recorded data)
  - Closed loop evaluation (full world simulation)

- Testing can take place
  - End-to-end
  - On the modular level

- Testing may not be trustworthy due to
  - Inaccurate sensor simulation
  - Not testing the entirety of the system
    - i.e. testing the planner only with ground truth inputs
  - No interaction between agents in open loop

- Clearly open loop and closed loop testing are both be useful tools for developing the AV stack

- We will return to open loop testing strategies at the end of the talk

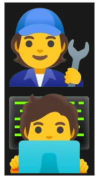
- For now, let us discuss in more detail how the simulator and testing procedure is actually designed for the closed loop evaluation

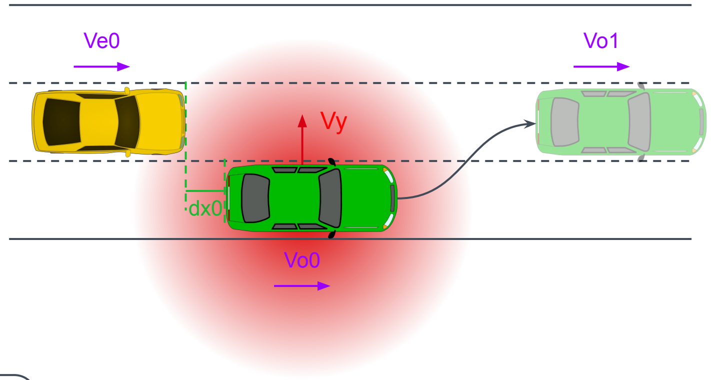


FIVE BOSCH

# Scenario based testing

## What and Why?



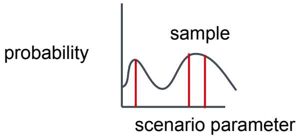



Human knowledge and data collection
“the big data loop”

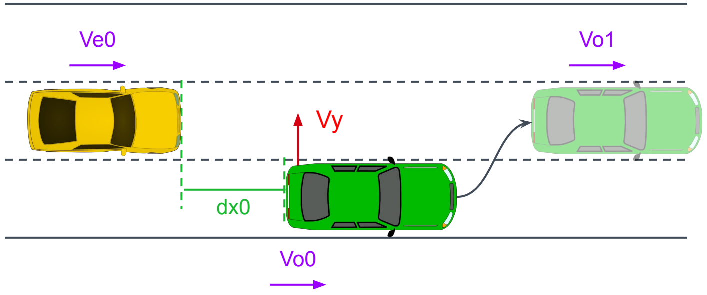
Enables creation of

Logical Scenario

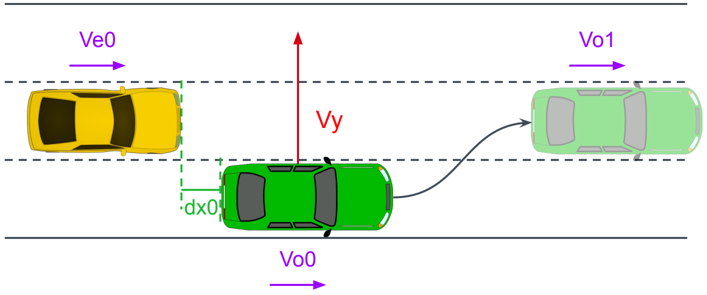
Sample from distribution over scenario parameters to initialize concrete scenarios




Concrete Scenario




Concrete Scenario




Concrete Scenario

...

- Results are easily human interpretable
- Enables fairness to be measured (e.g. risk concentration on different groups of road users)

ref [3, 4]

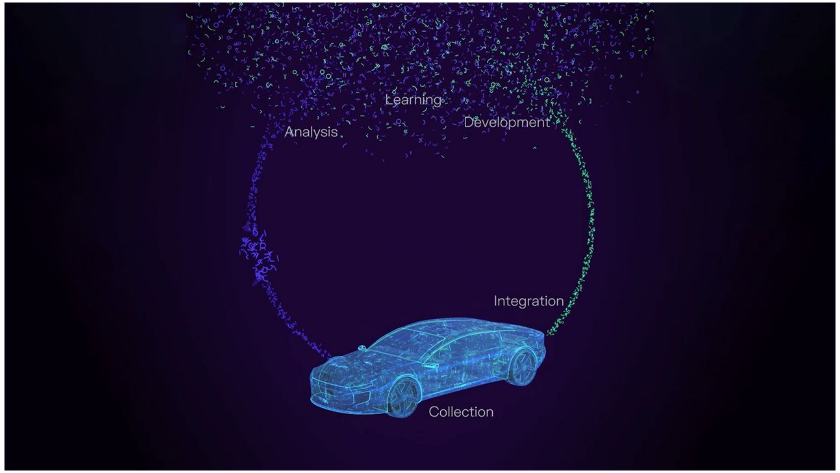


2023-09-07

# Data collection
## Ensure scenario model is correct

- Companies collecting vast datasets to identify all possible scenarios and agent behaviours

- Iteratively test new functionality whilst collecting data to improve the realism of the simulator and stack performance




CARIAD - the Big Loop
https://cariad.technology/de/en/news/stories/big-loop-introduction.html


2023-09-07

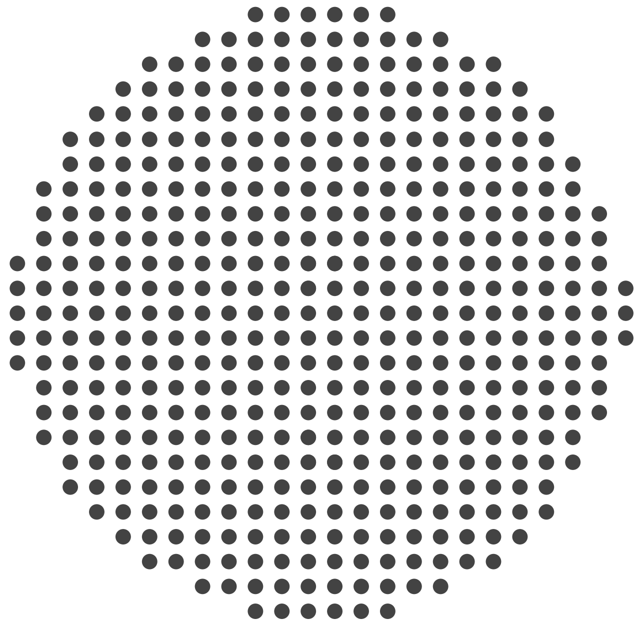
FIVE BOSCH

# Scenario based testing

## How do we know if the system is safe enough to deploy?

unknown safe scenarios

Scenario space





2023-09-07



FIVE BOSCH

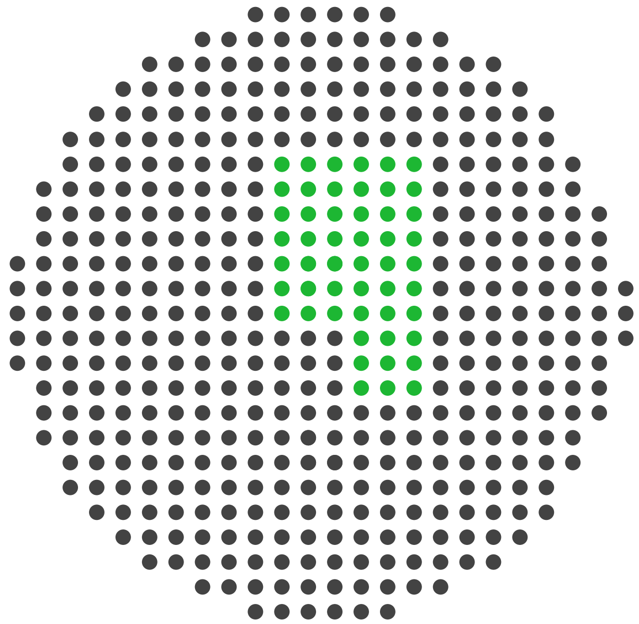
# Scenario based testing

## How do we know if the system is safe enough to deploy?

unknown safe scenarios

known safe scenarios

Scenario space



# Scenario based testing

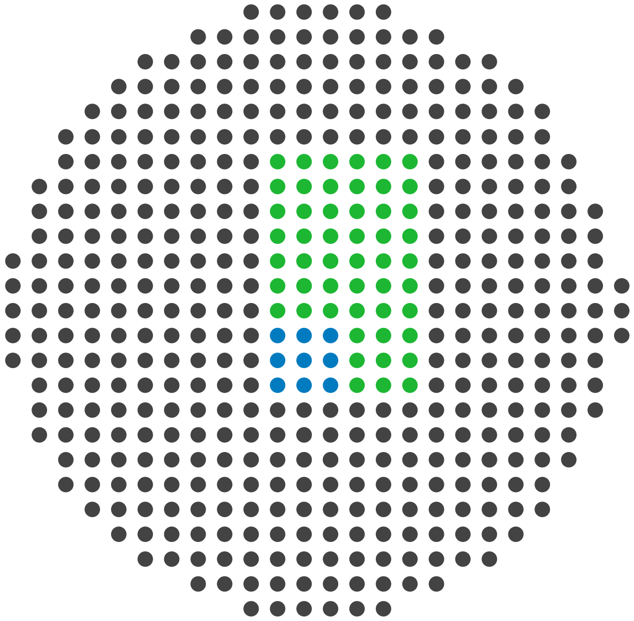
## How do we know if the system is safe enough to deploy?

unknown safe scenarios

known safe scenarios

known unsafe scenarios

Scenario space





2023-09-07



FIVE BOSCH

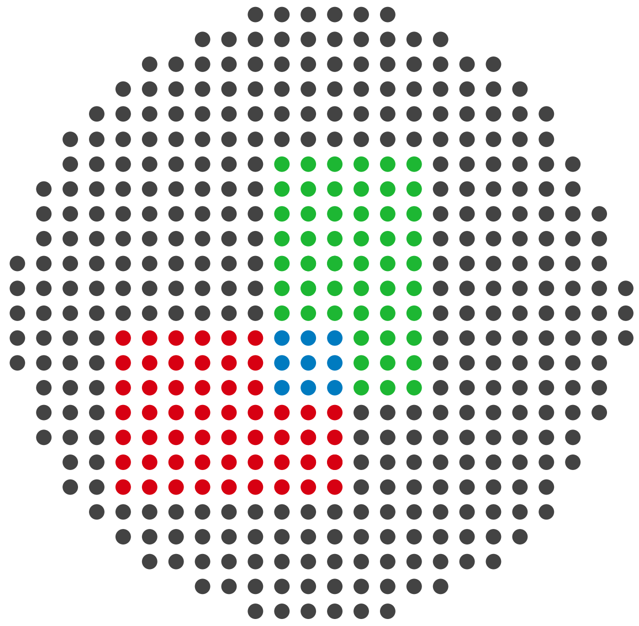
# Scenario based testing

## How do we know if the system is safe enough to deploy?

- unknown safe scenarios
- known safe scenarios
- known unsafe scenarios
- unknown unsafe scenarios

- <u>Verification</u>: calculating the size of these regions and ensuring red/blue are small (*checking the system against requirements*)
- <u>Validation</u>: ensuring the regions are defined correctly (*check if the system meets the needs of the user*)
  - the same in simulation and reality?
  - green matches ODD?

Scenario space




2023-09-07

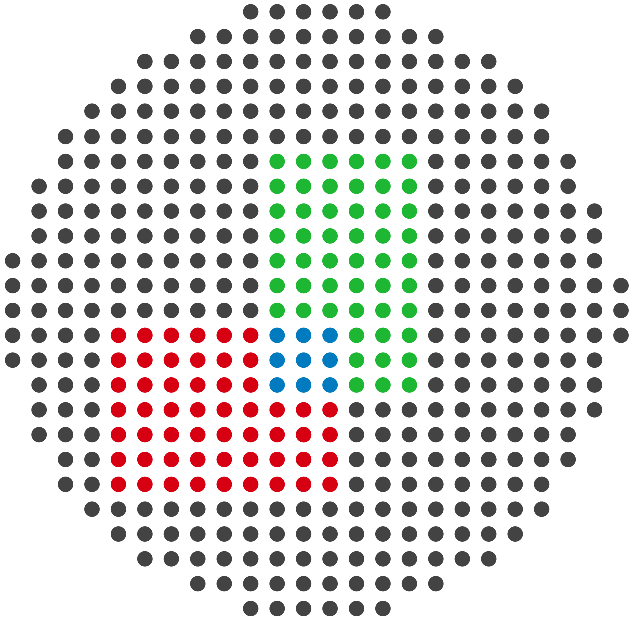
# Scenario based testing

## How do we know if the system is safe enough to deploy?

- unknown safe scenarios
- known safe scenarios
- known unsafe scenarios
- unknown unsafe scenarios

- The regions are defined by:
  - Logical scenarios
  - The behaviour of the ego vehicle
  - Behaviour of other agents
  - Rules, i.e. a digital highway code which define system failure ref [5]

Scenario space

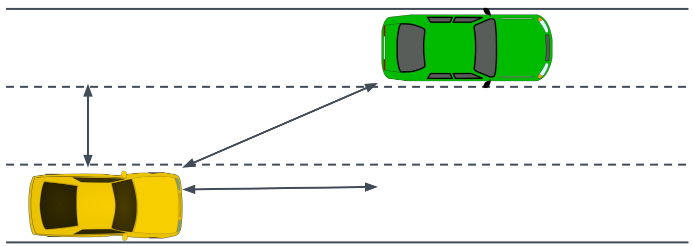




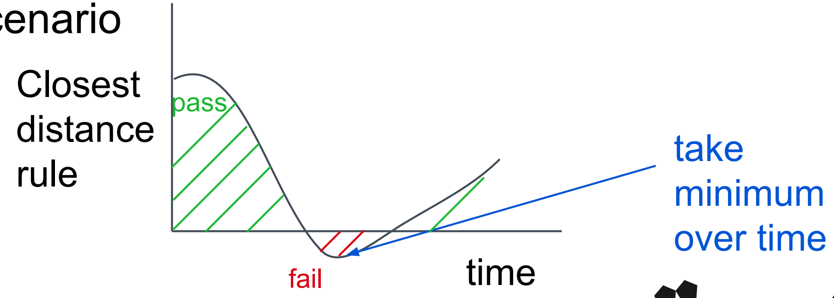
2023-09-07

# Rules
## What are they?

- Define success by measuring behaviours we care about, e.g.
  - Ego distance to other agents
    - Only in front of ego
    - Longitudinally
    - Laterally
    - With respect to time, i.e. time to collision, safe stopping distance
    - Rules are defined agent-wise hence agents breaking rules can have attributed responsibility for collisions



- Conventionally they are real valued functions of simulation history, and negative if the rule is broken
- Usually defined over time
  - the minimum over time gives "worst" time frame in scenario



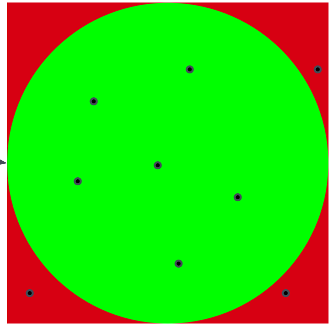

- If you need to implement these check out ref [5]

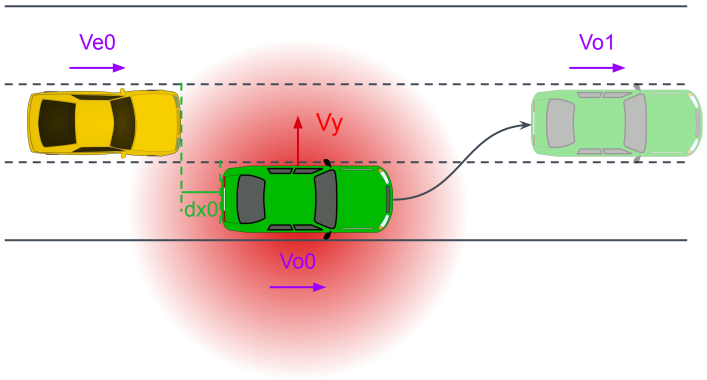

2023-09-07

# Monte Carlo Simulation
## A primer

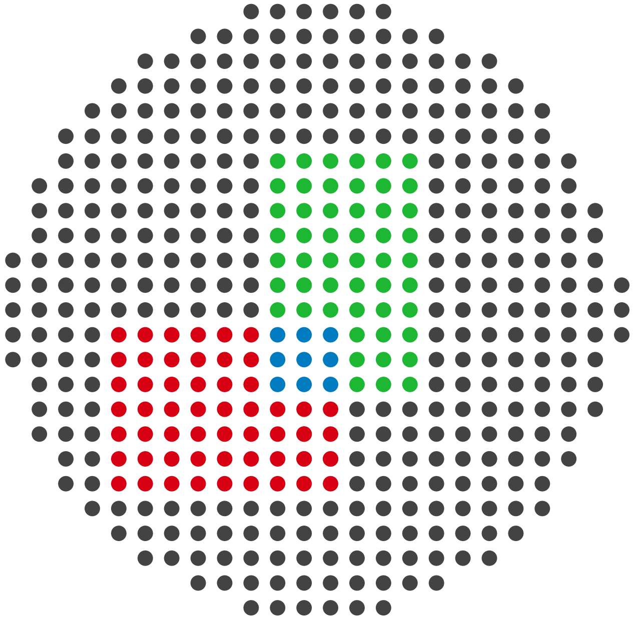
$$P_f = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(g(x_i) < 0) \qquad x_i \sim f(x)$$

Equivalent to throwing darts uniformly to estimate green area






Scenario space



- ε evaluations for 1/ε level of safety and each simulation is expensive - how can we reduce the cost? [6]
  - Typically ε ≅ $10^9$ hours for aircraft

# Overview of current industry challenges
## Problems and potential solutions

- Let's try to *verify* our AV system. To test the system against requirements we calculate the size of fail/pass regions of scenario space. How do we do this while…
  - Ensuring the regions in simulation accurately represent reality (validation)? *sim2real/domain gap*
  - Without too much computational cost? *Identifying the right scenarios to test*

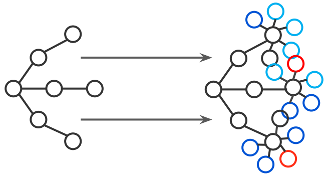
- Modular system testing - can we exploit the modular structure of the system so each team can get high performance on their module independently?
  - *You should also test the perception/prediction/localisation stack separately*

- What is the best approach we could deploy in industry right now?
  - *This is changing rapidly due to the fast evolving landscape of AV development*

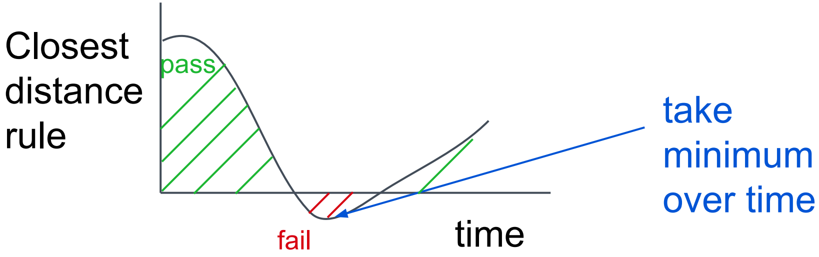

2023-09-07

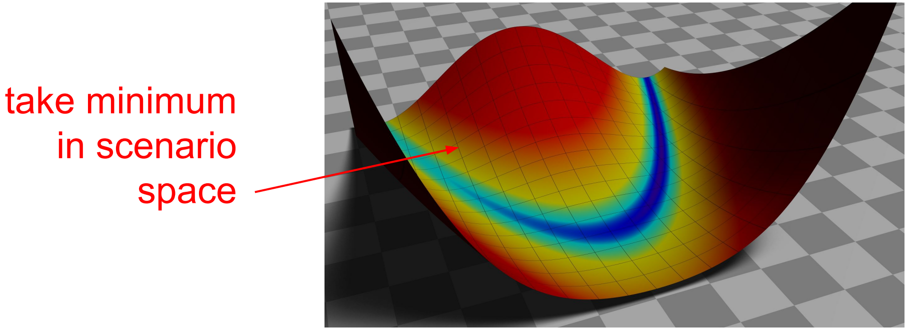
FIVE BOSCH

# Adaptive search techniques
## Adversarial scenario approach

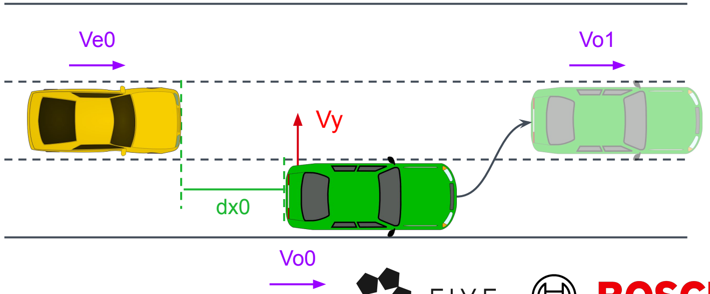


- Task: find the worst possible scenarios for the planner by optimising for rule breaking behaviour
- Approach: Minimise some loss representing the "rule breaking margin" over concrete scenarios in a specific logical scenario



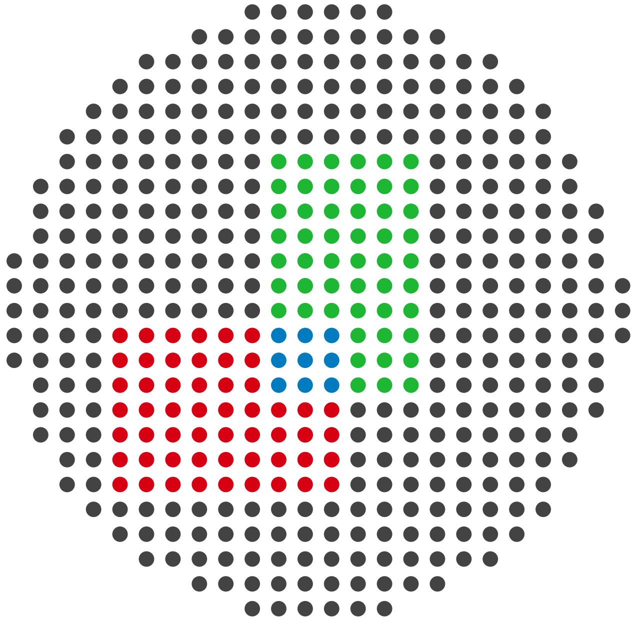



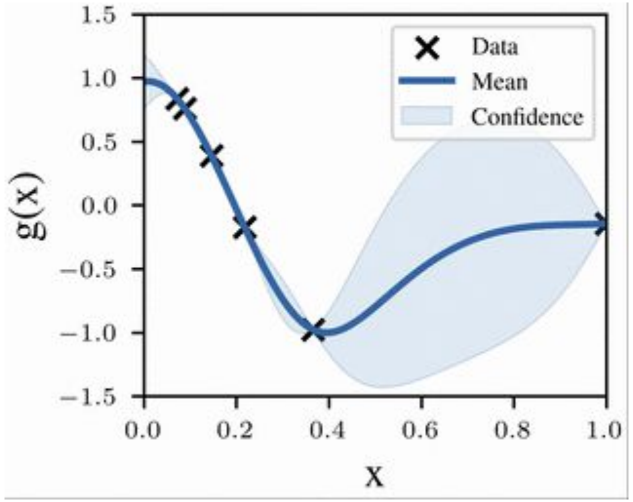

- Can be applied for a wide variety of simulator setups
  - Usually optimising over simulator parameters for a single logical scenario
- The optimisation used can be global or local
  - e.g. surrogate based or approximate gradients
- ref [7, 8, 9, 10]



# Adaptive search techniques
## Residual risk with surrogate models

- Adversarial scenario approaches don't give the size of the whole failure region
- Instead we can iteratively train a surrogate model to "zoom in" on the failure region boundary
  - Then estimate failure region using surrogate
- Gaussian processes are function approximators which also estimate their uncertainty
- Acquire data by running simulations where the ratio between g(x) and uncertainty is low
  - Iteratively retrain model 🔁



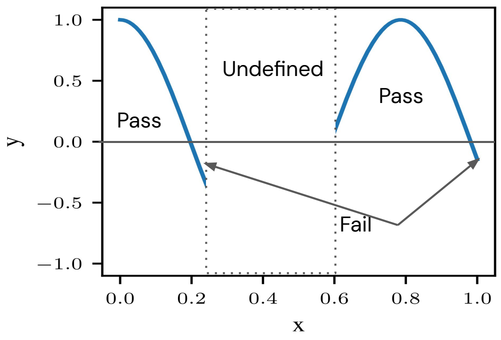


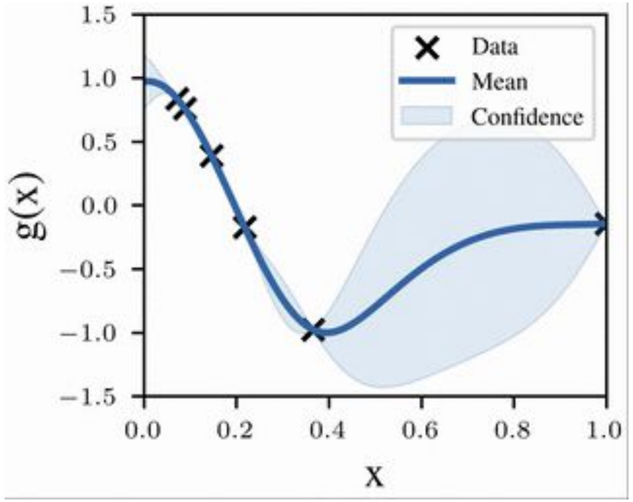


2023-09-07

# Adaptive search techniques
## Residual risk with surrogate models

- Similar approach used by Daimler (Mercedes) for residual risk
- In autonomous driving, rules can sometimes be undefined, e.g. if they do not apply in certain situations
  - Discontinuity in parameter space ⚠️ will break GP
- Our approach (hierarchical GP) [11] still correctly estimates uncertainty and identifies the failure boundary even for partially defined rules



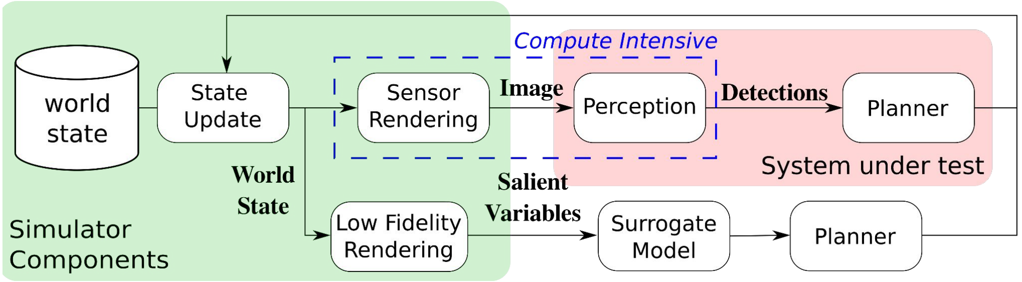


2023-09-07

FIVE BOSCH

# Domain gap
## Addressing the sim2real perception gap with PEMs

- Perception error models (PEM)
  - Train a neural network to approximate the sensor rendering and perception systems [13, 14, 15]
  - Try to predict the perception outputs given the world state
    - Can add extra “salient” variables to world state, e.g. occluded-ness
  - Can give 100x improvement in eval time
  - How similar is the PEM to the actual perception system outputs?
    - See our paper



# sim2real gap

## Addressing the sim2real perception gap for end to end systems

- For modular systems, bound discrepancies after each component [12]
  - Good preliminary results

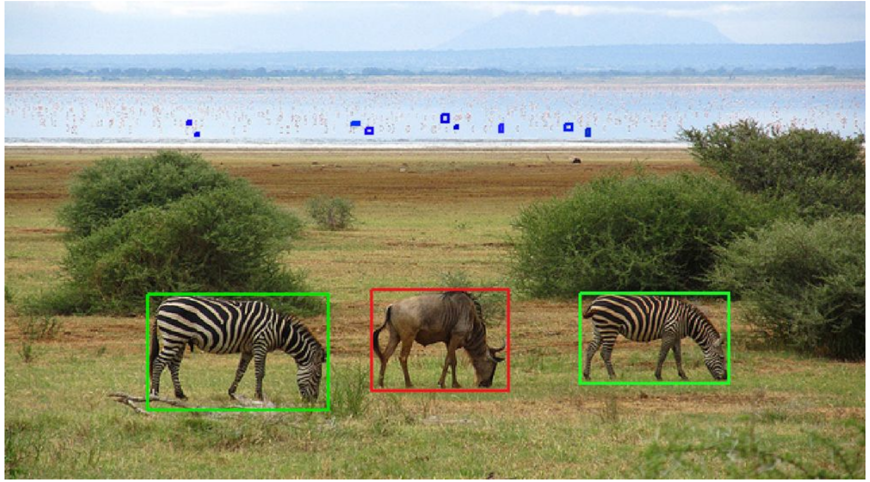
- For end to end
  - Improve the simulator by introducing greater diversity in simulated objects [16]
    - Known as domain randomisation
    - Many tricks which seem to improve real world performance in practice when training in simulators

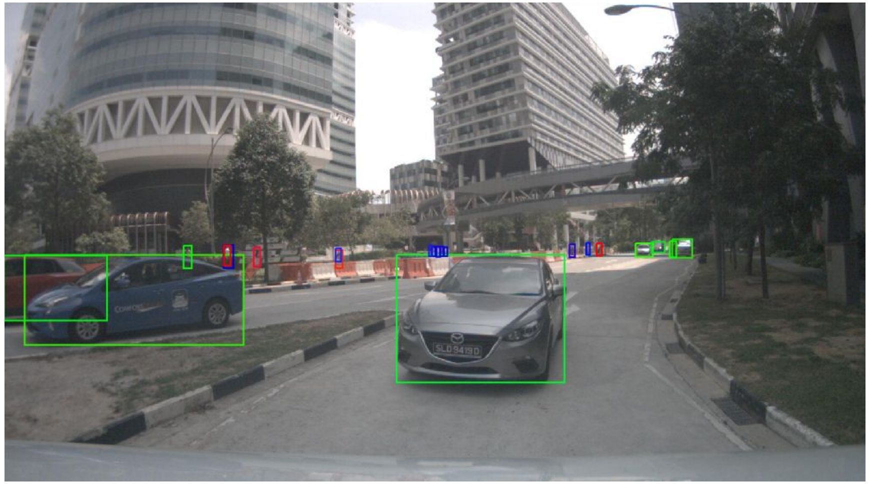
  - Sophisticated machine learning adaptation strategies [17]
    - Transfer learning to address the *domain gap*
    - Essentially trying to create a neural network with similar performance on both types of data

# Testing perception systems (open loop)

## Divide and conquer!

- If testing a modular system you can add separate tests to help debug the perception component (see open loop testing)
- Improving detector results in fewer downstream errors
- Several approaches
  - Vanilla approach: total precision, recall, mAP, box error
  - Query based hardness
    - Hardness depends on what you care about, e.g. pedestrians or cars ⇒ measure this [18]
  - More complex evaluation
    - Are all errors equally severe? ⇒ measure hierarchical errors [19]
  - Active learning approaches
    - iteratively retrain detector on most uncertain examples, which are labelled sequentially 🔁 [20]






2023-09-07

FIVE BOSCH

# Summary

## Recent advances on testing autonomous vehicle systems: an industry perspective

- Different approaches to simulation: open loop/closed loop, modular/end-to-end systems
  - Can test modules separately ⇒ strategies to describe errors in more detail
  - Scenario based approach with evaluation rules enables interpretable testing
- Some of the issues we encounter when trying to test autonomous vehicles in simulation
  - Domain gap (sim2real) ⇒ PEMs, discrepancy propagation, domain randomization
  - Computational cost ⇒ adversarial scenario approach, efficient sampling strategies

- There are many unsolved problems, but the technology to solve these problems is rapidly maturing

- Get in touch for internships/collaboration

- My site: https://jcsadeghi.github.io
- Hiring: https://www.five.ai/careers

# References

[ours in **bold**]


1. **Pulver, Henry, et al. "PILOT: Efficient planning by imitation learning and optimisation for safe autonomous driving." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021.**
2. **Hawasly, Majd, et al. "Perspectives on the system-level design of a safe autonomous driving stack." AI Communications Preprint (2022): 1-10.**
3. Menzel, Till, Gerrit Bagschik, and Markus Maurer. "Scenarios for development, test and validation of automated vehicles." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.
4. Menzel, Till, et al. "From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment." 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019.
5. Shalev-Shwartz, Shai, Shaked Shammah, and Amnon Shashua. "On a formal model of safe and scalable self-driving cars." (2017)
6. Koopman, Philip, and Michael Wagner. "Challenges in autonomous vehicle testing and validation." SAE International Journal of Transportation Safety 4.1 (2016): 15-24.
7. Wang, Jingkang, et al. "Advsim: Generating safety-critical scenarios for self-driving vehicles." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
8. Hanselmann, Niklas, et al. "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.
9. H. Beglerovic, M. Stolz and M. Horn, "Testing of autonomous vehicles using surrogate models and stochastic optimization," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 2017, pp. 1-6, doi: 10.1109/ITSC.2017.8317768.
10. Koren, Mark, et al. "Adaptive stress testing for autonomous vehicles." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.
11. **Sadeghi, Jonathan, Romain Mueller, and John Redford. "Bayesian Quadrature for Probability Threshold Robustness of Partially Undefined Functions." arXiv preprint arXiv:2210.02168 (2022).**
12. Reeb, David, et al. "Validation of composite systems by discrepancy propagation." Uncertainty in Artificial Intelligence. PMLR, 2023.
13. Piazzoni, Andrea, et al. "Modeling perception errors towards robust decision making in autonomous vehicles." arXiv preprint arXiv:2001.11695 (2020).
14. **Sadeghi, Jonathan, et al. "A step towards efficient evaluation of complex perception tasks in simulation." arXiv preprint arXiv:2110.02739 (2021).**
15. **Innes, Craig, and Subramanian Ramamoorthy. "Testing rare downstream safety violations via upstream adaptive sampling of perception error models." 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023.**
16. Pouyanfar, Samira, et al. "Roads: Randomization for obstacle avoidance and driving in simulation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019.
17. Bewley, Alex, et al. "Learning to drive from simulation without real world labels." 2019 International conference on robotics and automation (ICRA). IEEE, 2019.
18. **Ayers, Edward, et al. "Query-based Hard-Image Retrieval for Object Detection at Test Time." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37. No. 12. 2023.**
19. **Bertinetto, Luca, et al. "Making better mistakes: Leveraging class hierarchies with deep networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.**
20. Philion, Jonah, Amlan Kar, and Sanja Fidler. "Learning to evaluate perception models using planner-centric metrics." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.
21. Yu, Weiping, et al. "Consistency-based active learning for object detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

# Thank you

FIVE BOSCH