Focus

- Level 4 HAVs are only required to operate autonomously within **specific conditions** under which the system is intended to function .
  - Operational Design Domain (ODD)

Summary

- On-road testing is not enough
- A layered approach is proposed.

Role of Vehicle Test and Simulation

On-road testing is impractical

- takes **a huge number of miles** to make a credible statistical argument.
- is potentially **undermined with each software change** (training data updates)
- what if **an HAV is not living up** to its hoped-for safety goal on-road testing?
  - another round after fixing the bugs?
- unusual situations must be handled safely, but such **situations are comparatively rare** in normal driving

Need a way to build a **methodical, defensible safety argument** that can be evaluated by an independent party despite any unique validation challenges.

Vehicle-Level Testing and Simulation

- Closed-course testing: set known rare events up as explicitly designed test scenarios.
- Software-based vehicle simulation can scale up coverage of test scenarios to acceralate evaluation
  - involves a tradeoff of **fidelity vs. run-time cost** as well as questions about **completeness and accuracy** of software models
    - The level of fidelity in a simulation is the degree to which **it makes simplifications and assumptions** about the behavior of the system
  - unknown safety-relevant rare events still happen.

The key to improving testing efficiency is realizing that not all realism is actually useful for all tests.

- e.g., the coefficient of road surface friction is generally irrelevant to determining if a computer vision capability can see a child in the road.

Effective and efficient simulation

- that the HAV system model is sufficiently accurate,
- the assumptions made by the various-fidelity models of the system and operational environments.

Any practical validation effort should be considered as **a hierarchical series of models of varying levels of abstraction and fidelity.**

> Closed-course testing is a form of simulation, because even though obstacles and vehicles involved might be real, the scenarios are "simulated."

A robust safety validation plan must address at least

- Requirements defects:
  - the system is required to do the wrong thing (defect), is not required to do the right thing (gap), or has an ODD description gap.
  - **a complete set of behavioral requirements needs to be developed**
    .
    - systems that use on-road data as the basis for training machine learning do not ever identify requirements per se
- Design defects:
  - the system fails to meet its safety requirements or fails to respond properly to violations of the defined ODD.
- Testing plan defects:
  - the test plan fails to exercise corner cases in requirements or design, or has other gaps.
- Robustness problems:
  - invalid inputs or corrupted system state cause unsafe system behavior or failure

Incomplete Requirement

A key challenge for HAV validation is that a complete set of behavioral requirements needs to be developed before behavioral correctness can be measured to provide pass/fail criteria for testing .

Vehicle Testing as Requirements Discovery

On-road testing is that accumulating miles in a search for missing requirements

- Detecting and evading novel road hazards
- Emergent traffic effects due to HAV behaviors

Encountering some unexpected scenarios will result in a **requirements update**, while others result in a modification either of ODD parameters or ODD violation detection requirements.

Separating Requirements Discovery and Design Testing

- On-road testing should primarily **emphasize requirements validation**,
- while lower level simulation and testing should **emphasize the validation of design and implementation.**

> Example
>
> - Smple coding defects should be found in subsystem simulation
> - On the other hand, rare event requirements gaps might be best found in on- road testing if they are due to unforeseeable factors.

A Layered Residual Risk Approach

To approach the problem of missing safety requirements is to start with simple set of rules and elaborate them over time in response to tests that violate those simplistic rules

Optimal performance may not be needed , simpler requirements are likely to be sufficient to define safe operation . A list of unsafe behaviors that are forbidden based on safety envelopes can be sufficient for some autonomous vehicle behaviors

> Safety envelope for lane-keeping could be that the vehicle stays within its lane boundaries plus some safety margin. This is much simpler than checking perfect implementation.

If an HAV design team attempts to **determine safety requirements via machine learning-based approaches**, it will be important for them to **express the results in a way that is interpretable** to human safety argument reviewers .

Safety envelope approach can simplify the complexity of creating a model of requirements to use for pass/fail criteria,

HAV testing will still need to run a huge number of scenarios to attain reasonable coverage

Managing Residual Risks in Simulations

The important relationship between **high- and low-fidelity simulation** is emphasizing validating the correctness of assumptions and simplifications made at lower fidelity levels.

- If a particular level of fidelity model is "wrong", a higher fidelity simulation should assume the burden of mitigating that residual safety validation risk.

Roles of high fidelity model

To mitigate that residual risk by

- not only checking the accuracy of lower fidelity simulation results,
- but also by checking whether assumptions made by lower fidelity models are violated when the higher fidelity simulation is performed.

Example

> if a simplified model assumes 80% of radar pulses detect a target, a higher fidelity model or vehicle test should **flag a fault if only 75% of pulses detect a target** – even if the vehicle happens to perform safely according to the higher fidelity model.
>
> The assumption of 80% detection rates is a residual risk of the lower fidelity simulation that makes that assumption. Violating that assumption invalidates the safety argument, even if a particular test scenario happens to get lucky and avoid a mishap.

An Example of Residual Risks

Validation Activity	Residual Risks (Threats to Validity)
Pre-deployment road tests	Unexpected scenarios, environment
Closed course testing	As above, plus: Unexpected human driver behavior, degraded infrastructure, road hazards
Full vehicle & environment simulation	As above, plus: simulation inaccuracies, simulation simplifications (e.g., road friction, sensor noise, actuator noise)
Simplified vehicle & environment simulation	As above, plus: inaccurate vehicle dynamics, simplified sensor data quality (texture, reflection, shadows), simplified actuator effects (control loop time constants)
Subsystem simulation	As above, plus: subsystem interactions

Obstacle detection example,

> Higher fidelity levels such as physical vehicle testing should **not primarily focus on different sizes and placement of obstacles**.
>
> Rather, they should **focus on things such as dirt on objects and sensors, and other aspects that might not be handled by software-only simulation tools**.

Improving Observability

Given a thorough simulation- and vehicle-based test plan, sufficient **controllability and observability** must be provided to yield a credible safety validation outcome.

- Controllability is the ability of a tester to **control the initial state and the workload** executed by a system under test.

  Controlling test scenarios to elicit a particular autonomous system behavior is difficult because of combination of the use of stochastic methods (randomized path planners)

  A useful approach to improving controllability is

  - to use **simulation** that can avoid physical world randomness and constraints.
  - a system testing interface can be provided that forces the system into an initial state for testing.
  - A path planner might be tested in a repeatable manner if its **internal pseudo-random number generator can be set to a predetermined seed value.**

- Observability is the ability of the tester to **observe the state of the system** to determine whether a test passed or failed. [33]

Observability can be a more difficult problem.

> In a vehicle-level obstacle test if the system "passes" a test by not colliding,
>
> - simply be due to the system **getting lucky** in avoiding an obstacle it did not even know was there.
> - The system might **hit the obstacle on the next test run** – or perhaps hit it 2000 test runs later.
> - This lack of observability is one facet of the robot legibility problem, which recognizes the difficulty of humans understanding the design,

Passing Tests for the Right Reason—interpretability

When a human takes a driver test, the test examiner has a fairly accurate mental model because misbehavior can be observed .

> I f the driver changes lanes without making eye contact with a rear-view mirror or otherwise checking for vehicles in the destination lane, the examiner knows that the driver got lucky in executing a collision-free lane change instead of behaving properly.

For HAV, it is unclear what the “tells” are for a machine exhibiting safe behavior vs. getting lucky with unsafe behavior. Being able to reasonably infer causality of actions from explicit system information can reduce testing costs compared to a brute force statistical approach

Having an HAV self-report regions of saliency

> As an example, rather than just performing a vehicle lane change when it can, the vehicle might report: “I want to change lanes … I am checking the next lane and there is a car there but it is sufficiently far behind me that I am clear … I am starting to change lanes … I am continuing to monitor that the lane is still clear … the car behind me is speeding up to close the gap …” and so on

The advantage of an explicit explanation is that the validity of that mechanism can be made falsifiable if it is required to match the test plan narrative

- In designing safety- critical systems, we prefer explicit, verifiable, simple patterns that might be less performant over those that are highly- optimized but opaque.

Coping with Uncertainty

While approaches such as safety envelopes can help, in the end, there is **no way to completely mitigate residual risks from unknown types of defects**.

- A confidence assessment framework [40] that has been extended to **include unknown unknowns** is one approach that could provide a way to manage residual risks.
- Each time a surprise causes a safety problem, additional steps should be taken to address underlying system and safety argument assumptions that are invalidated by the newly discovered issue

How to measure HAV “maturity” to ensure that this desirable outcome is fully achieved‘:

- The first way is ensuring that the **HAV passes a detailed technical driving skill test for the right reasons**,

- the second way is monitoring **whether the HAV validation assumptions and residual risk monitoring hold up** when it is deployed in the real world.
    - Ensuring that there are no vehicle operational situations that invalidate assumptions. If a high rate of assumption violations is detected by runtime monitoring, that can provide valuable feedback to the design team of an impaired safety margin.