Agenda

Session 1: 1:00 PM to 1:50 PM

Introduction to NVIDIA Halos - Strategy for AV Safety
- Chapter 1: Overview
- Chapter 2: Design-time safety guardrails

Session 2: 2:00 PM to 2:50 PM

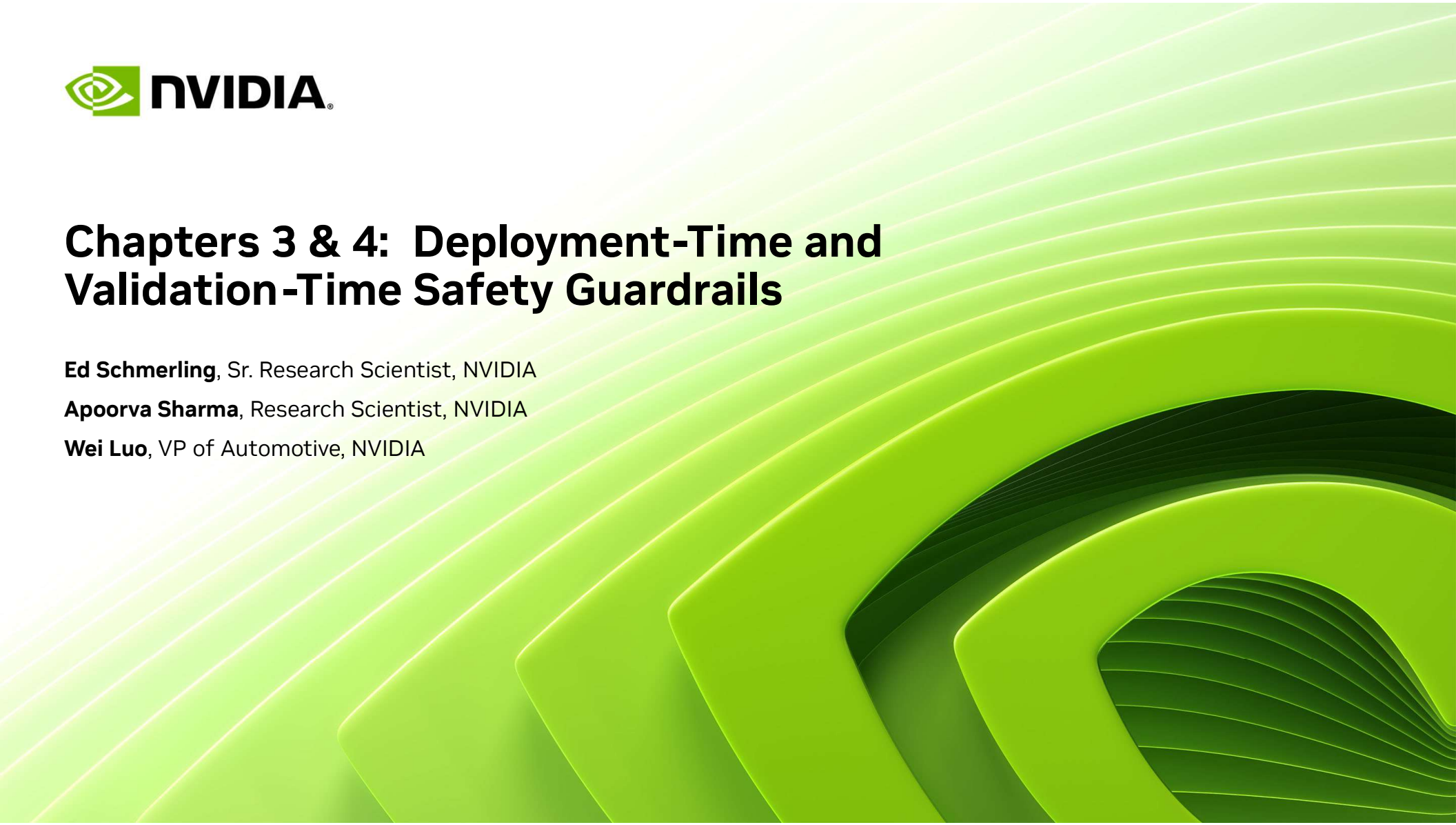
Guardrails for NVIDIA Halos Across the Product Life Cycle
- Chapter 3: Deployment-time guardrails
- Chapter 4: Validation-time guardrails

Session 3: 3:00 PM to 3:50 PM

Safety Regulation and Standardization in the Era of AI-Based AV
- Chapter 5: Safety regulation and standardization
- Chapter 6: From AVs to general Physical AI

Session 4: 4:00 to 5:00 PM

Navigating the High-Stakes Safety Challenges of AVs



Chapters 3 & 4: Deployment-Time and Validation-Time Safety Guardrails

Ed Schmerling, Sr. Research Scientist, NVIDIA

Apoorva Sharma, Research Scientist, NVIDIA

Wei Luo, VP of Automotive, NVIDIA

Chapter 3: Deployment-Time Guardrails

Run-Time Monitoring and Arbitration

Synoptic View of NVIDIA Halos AV Safety Day

Design-time

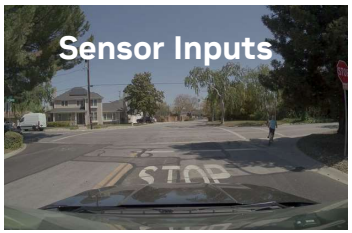
Design-time safety (Chapter 2):
- Safety architecture
- AI train-time safety
- AV platform safety
- Data flywheel and processes

Large-Scale Data

E2E

Validation-time guardrails (Chapter 4):
- Metrics
- Coverage – top-down
- Coverage – bottom-up
- Data flywheel and processes

Run-time



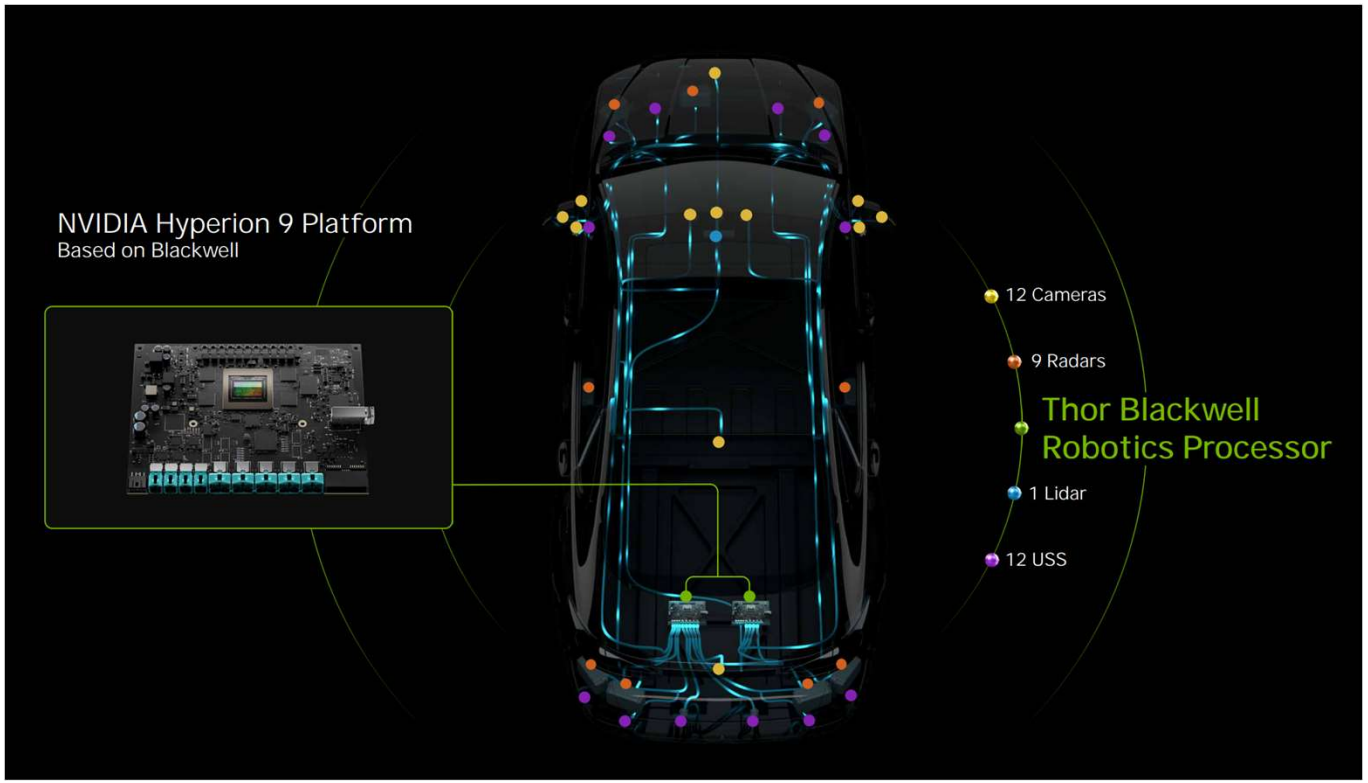
Deployment-time guardrails (Chapter 3):
- Run-time monitoring – HW
- Run-time monitoring – SW
- Arbitration
- Data flywheel and processes

Safety Regulation and Standardization in the Era of AI-Based AV (Chapter 5):
- Standardization challenges
- Regulatory challenges
- NVIDIA AI Systems Inspection Lab

From AVs to general Physical AI (Chapter 6):
- How Halos extends to Physical AI
- NVIDIA IGX elements
- Outside-in safety

Key Problem: Accounting for Real-World Variability & Uncertainty

- Safety starts with design-time guardrails:
  - Diverse redundancy
  - Safety data flywheel
- Now at deployment time:
  - What happens if your diverse sensors apparently disagree?
  - What happens if you encounter an unfamiliar situation you aren't prepared to handle?
- Despite our best laid plans, the real world happens
- Run-time monitoring provides an avenue to plan for the unknowns you expect to encounter on the road!

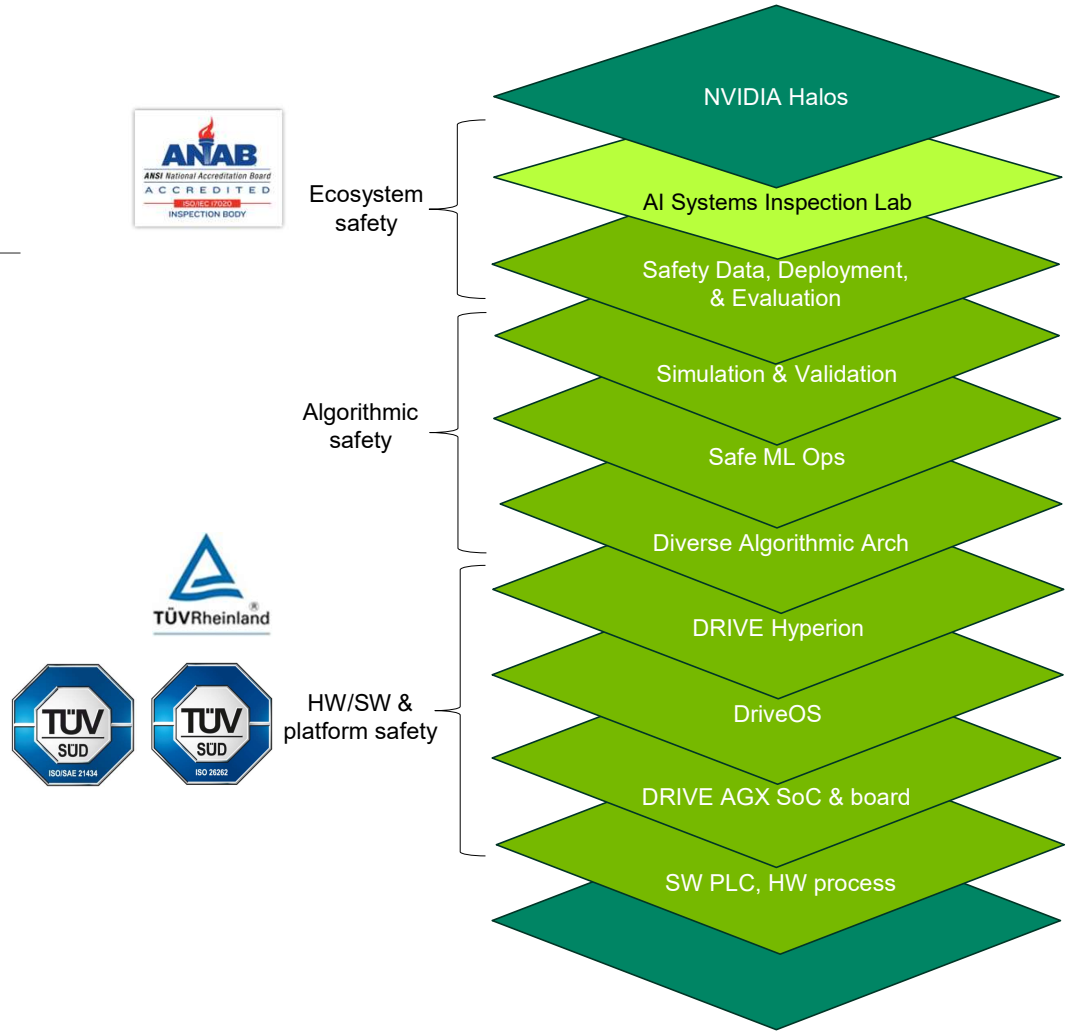


NVIDIA DRIVE Hyperion – Reference Architecture

NVIDIA Halos

A Full-Stack Comprehensive Safety System for Autonomous Vehicles

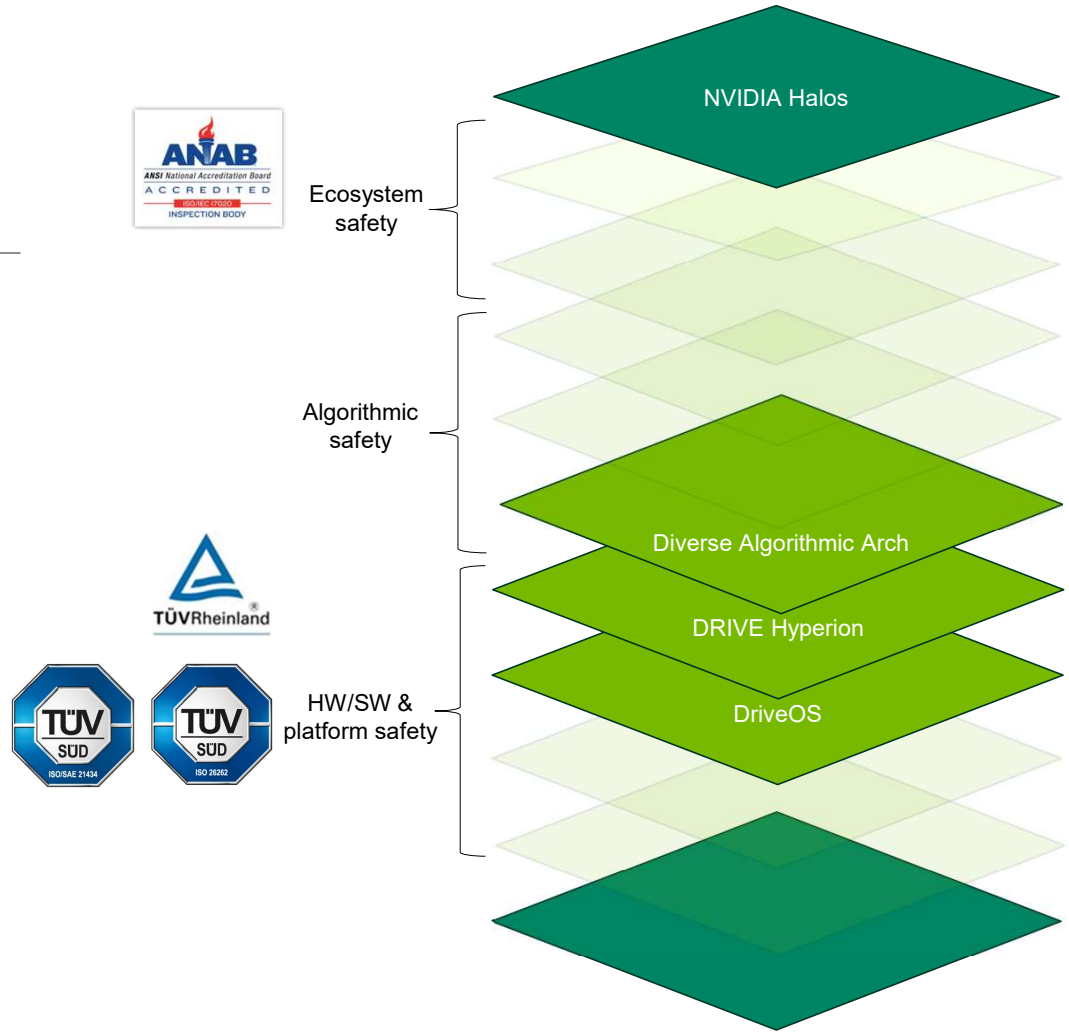
- Halos is a full-stack comprehensive safety system for Autonomous Vehicles that unifies safety elements from vehicle architecture to AI models.
- It comprises HW and SW elements, tools, models, and the design principles for combining them to safeguard AI-based, end-to-end AV stacks



NVIDIA Halos

A Full-Stack Comprehensive Safety System for Autonomous Vehicles

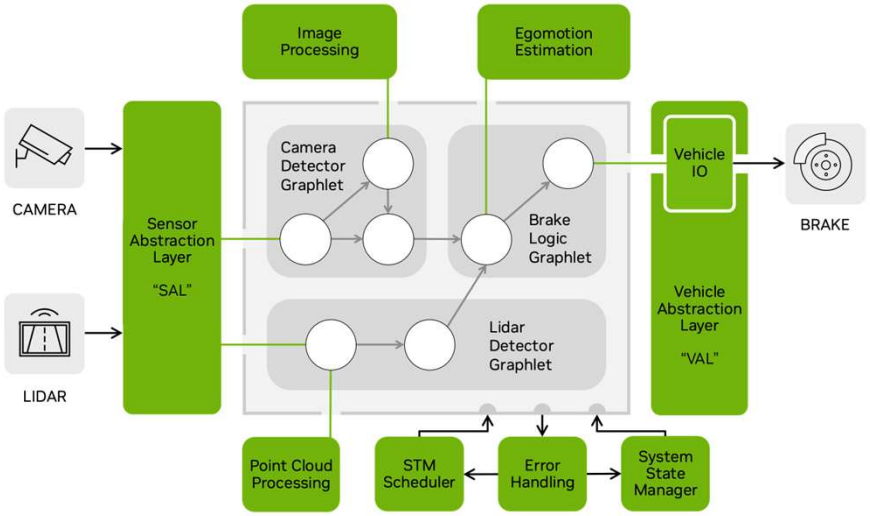
- Halos is a full-stack comprehensive safety system for Autonomous Vehicles that unifies safety elements from vehicle architecture to AI models.
- It comprises HW and SW elements, tools, models, and the design principles for combining them to safeguard AI-based, end-to-end AV stacks
- Deployment-time guardrails are co-designed (and validated) with the full-stack Halos system, but are particularly focused on the intersection of platform and algorithmic safety



Deployment-Time Safety Assurance for "Classical" AV Stacks

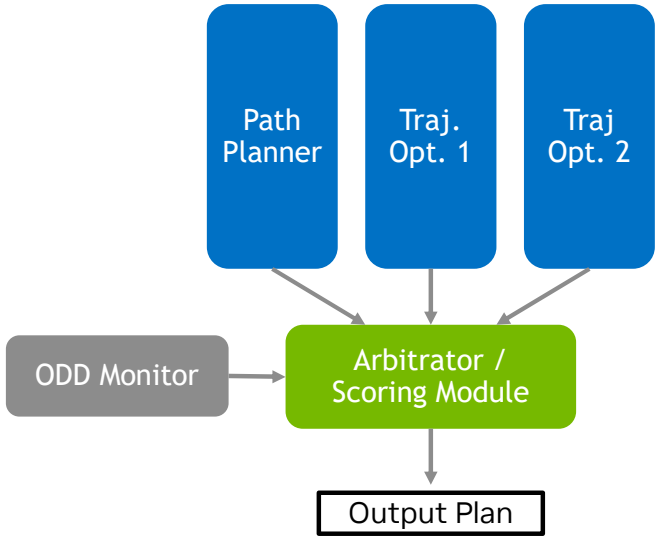
Run-Time Monitoring is Implemented as Modules Throughout the Stack

- **Component-Level Robustness**
  - Health monitoring of individual components is incorporated within the compute graph, e.g., with discrete state machine logic and/or Bayesian filtering of uncertainties
  - Diverse redundancy provides a traceable contribution to safety case (e.g., ASIL decomposition)



Example sensor fusion for Automatic Emergency Braking (AEB)

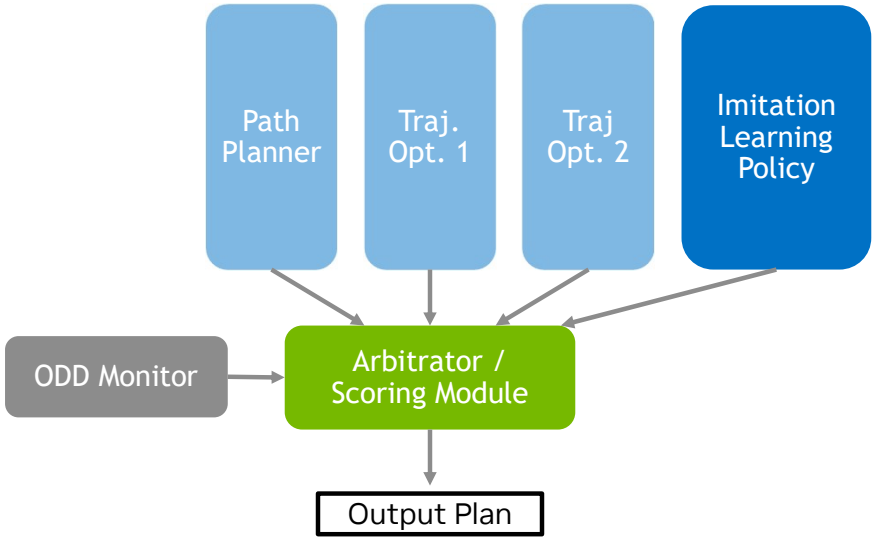
- **System-Level Safety**
  - Various modules, e.g., planner(s), reason explicitly about hazards in the scene
  - To accommodate increasing complexity in target operational design domain (ODD), multiple specialized implementations are often composed through arbitration



Planner decomposition + Monitor-based Arbitration

Progressing Towards Safe End-to-End Systems

- Arbitration = Avenue for E2E
  - In practice, deployment of learned planners has often been staged through the plan arbitration module

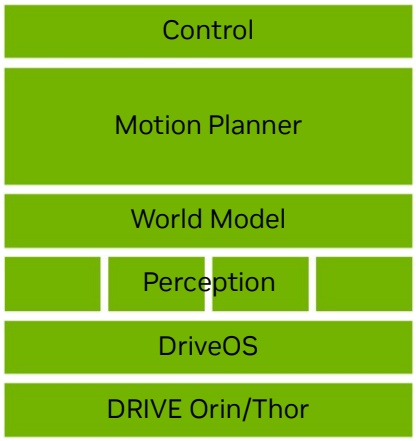


  - This construction places the bulk of the deployment-time safety burden for ML planning onto the arbitrator

➡ Run-time monitoring as a reframing of classical decomposition-based safety arguments.

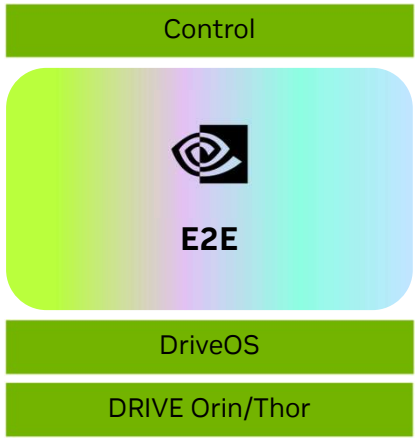
Key Challenges in Run-Time Monitoring for E2E-Empowered AV

Modular AV Stack



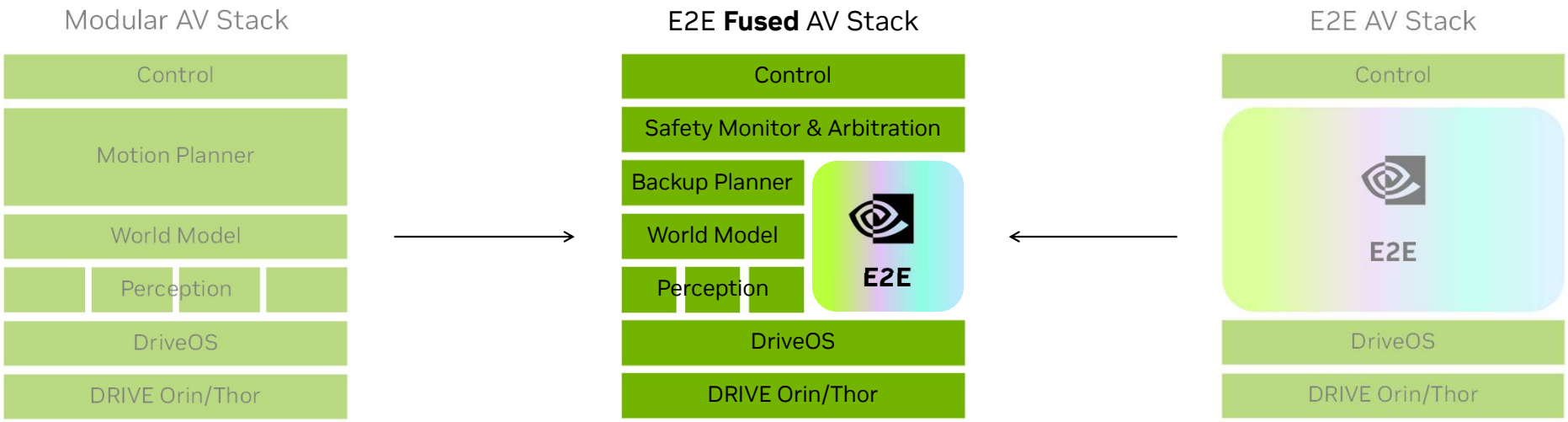
- Rich environment of interpretable signals: every component has checkable input/output
- Assessing system status can be complicated for faults that cross multiple components (often requiring a human's holistic view)
- Algorithmically, what is there to monitor? E2E monitoring problem is as hard as E2E driving itself.
- The power of E2E AI is to provide holistic solutions – but it may still benefit from "sanity checking"!

E2E AV Stack



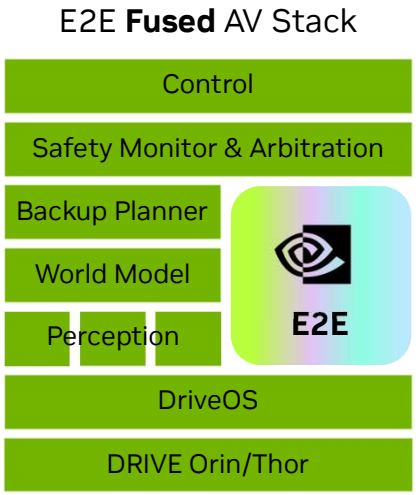
- Engaging the appropriate fallback (including alerting the driver if necessary) is a key monitoring challenge for both stacks.

Key Challenges in Run-Time Monitoring for E2E-Empowered AV



Can a fused approach provide complementary benefits, retaining compositional monitoring from the modular portion while adding system-level safety monitoring from the E2E portion?

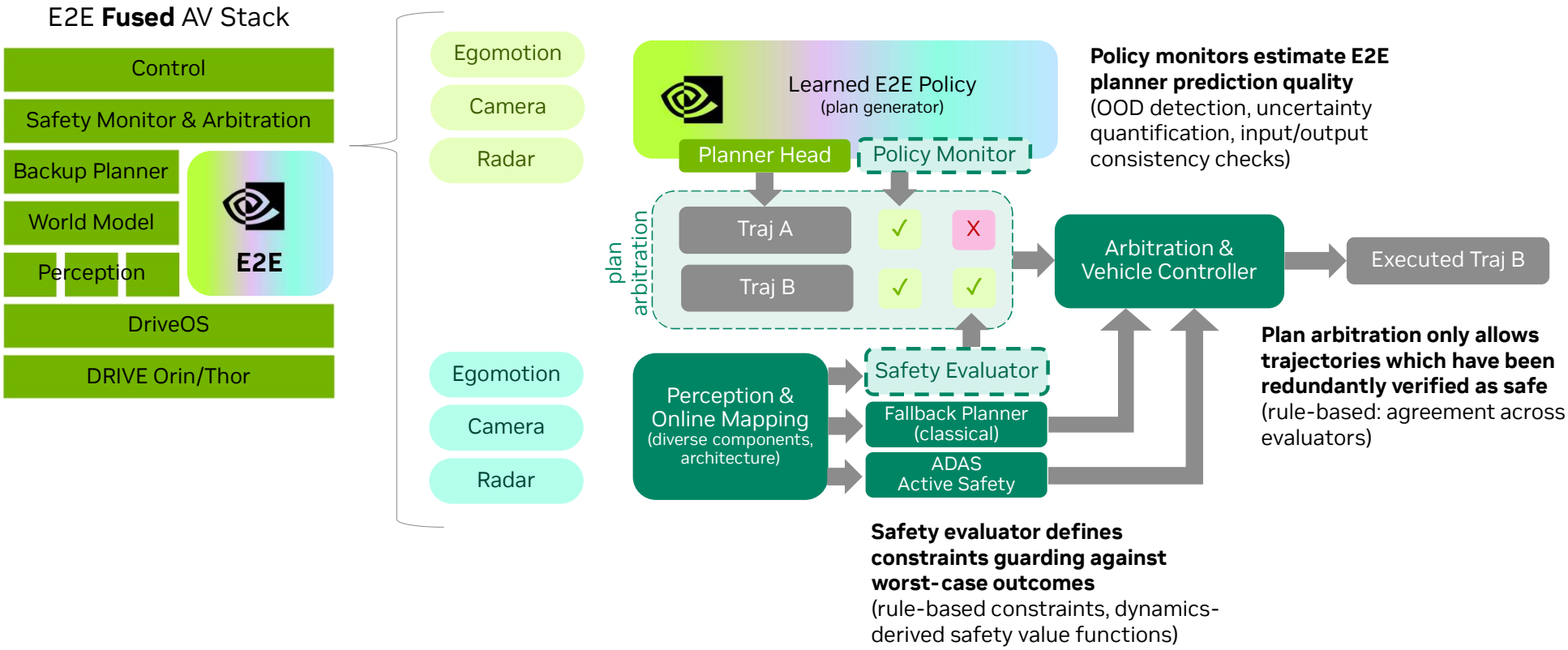
Key Tenets of Run-Time Monitoring for AV Stacks



- **Built upon a foundation of HW/component-level safety**
  - Extending proven safety architecture to E2E, not reinventing approach to safety entirely
- **Mitigate common causes of failure**
  - A monitor must provide additional safety signal beyond the systems it's monitoring
- **Limited in scope**
  - Monitor implementation (correspondingly, validity) may be specific to certain ODDs, with growing coverage over the product lifecycle
  - Restricted to safety *constraints*, not optimal behavior; validating a monitor must be simpler than validating the full system
- **Compositional safety argument**
  - Monitoring should not itself exclusively span E2E, but make use of intermediate "component-level" information
  - Composition is necessary to ensure validation data requirements are tractable
- **Flexibility in monitor implementation**
  - Monitoring framework should be compatible existing safety concepts, allow for future advances

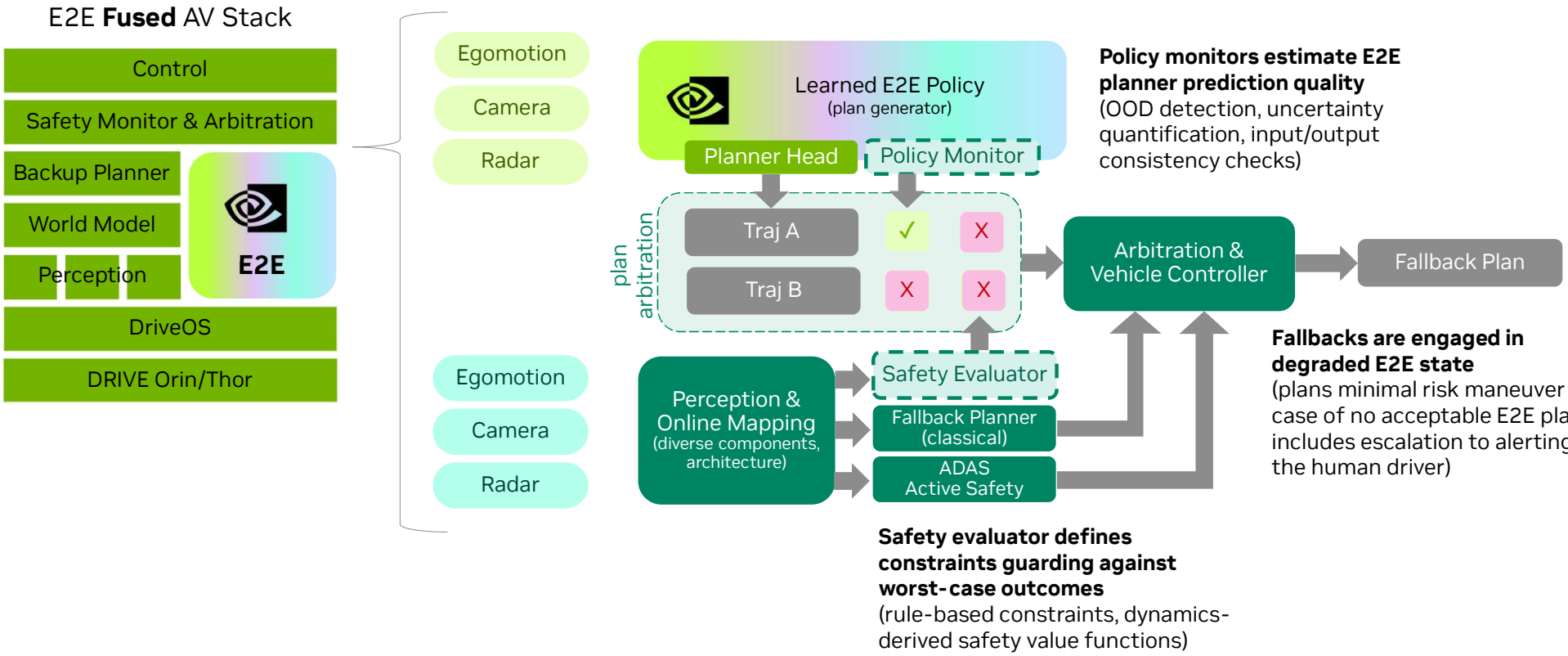
Run-Time Monitoring With an E2E Fused AV Stack

Decomposition Across Diverse and Redundant Systems Enables Tractable Validation of Planner Safety



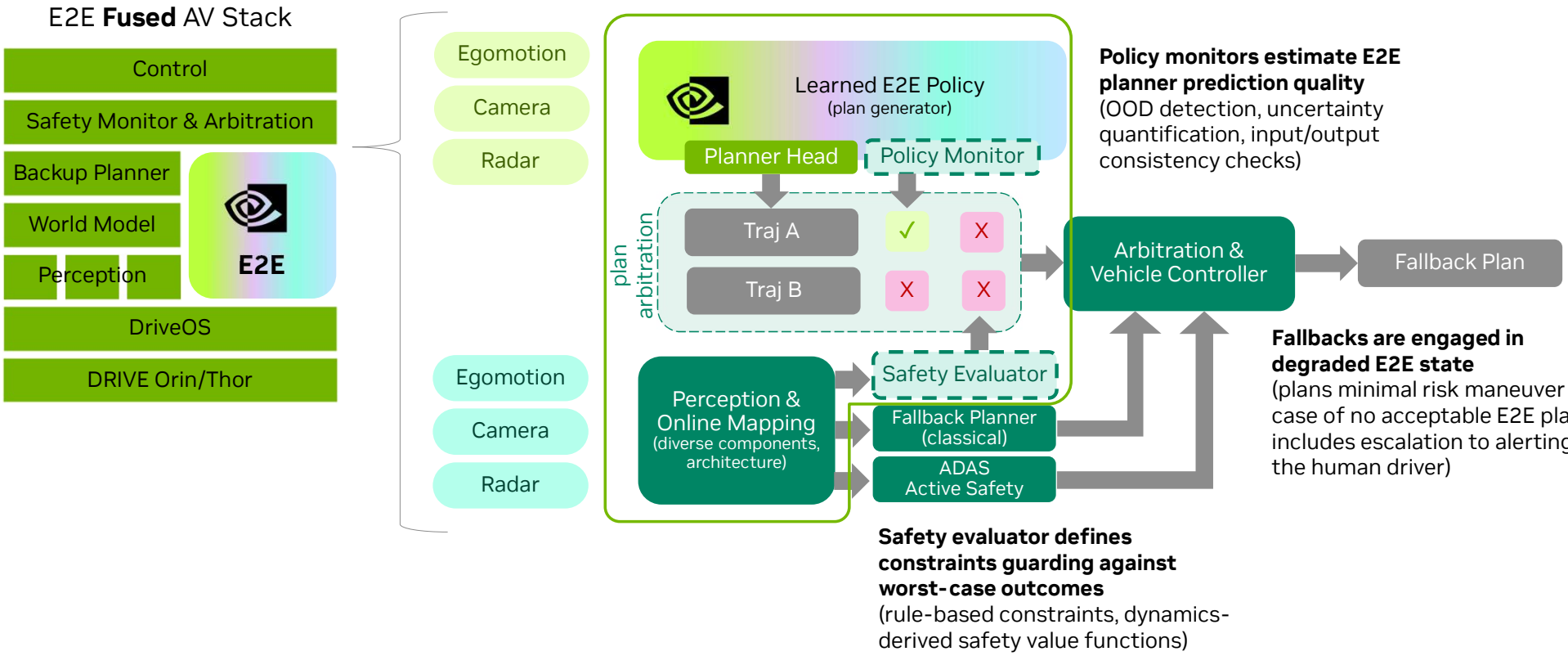
Run-Time Monitoring With an E2E Fused AV Stack

Decomposition Across Diverse and Redundant Systems Enables Tractable Validation of Planner Safety

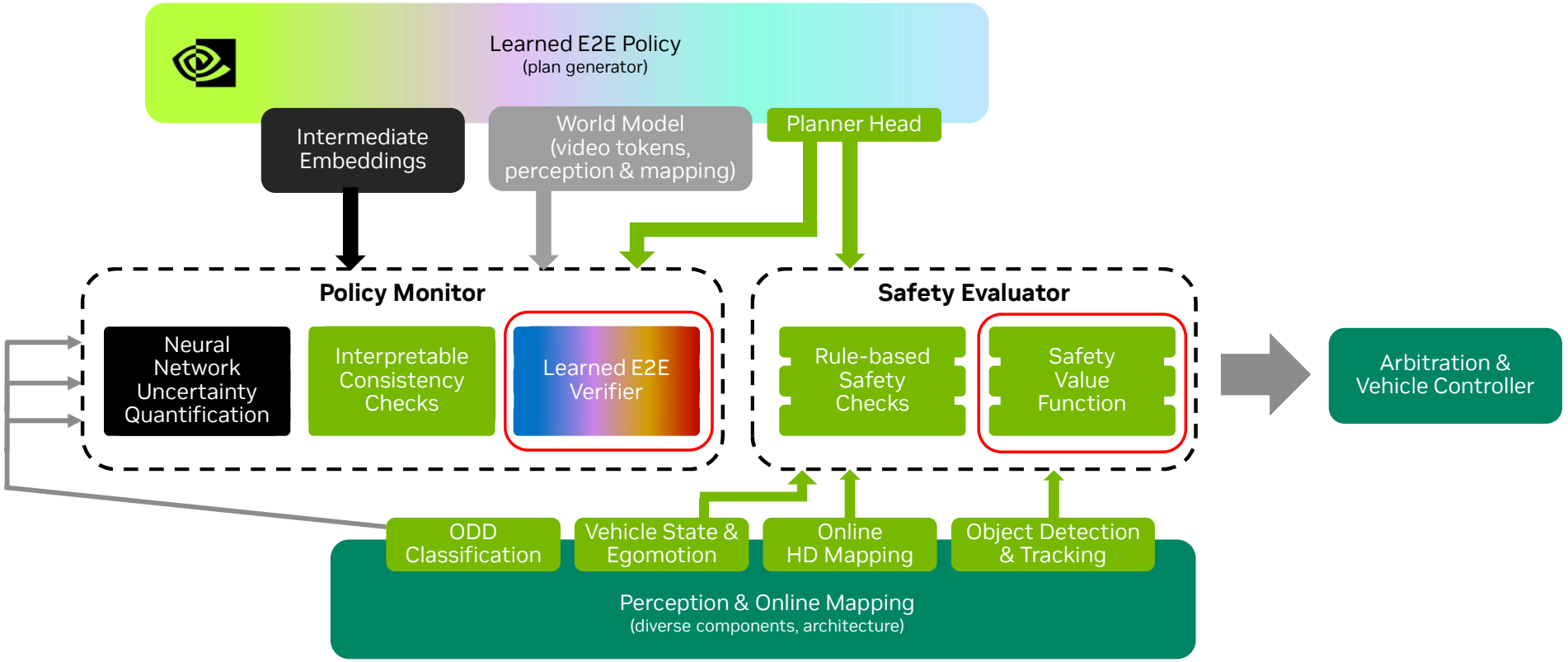


Run-Time Monitoring With an E2E Fused AV Stack

Decomposition Across Diverse and Redundant Systems Enables Tractable Validation of Planner Safety



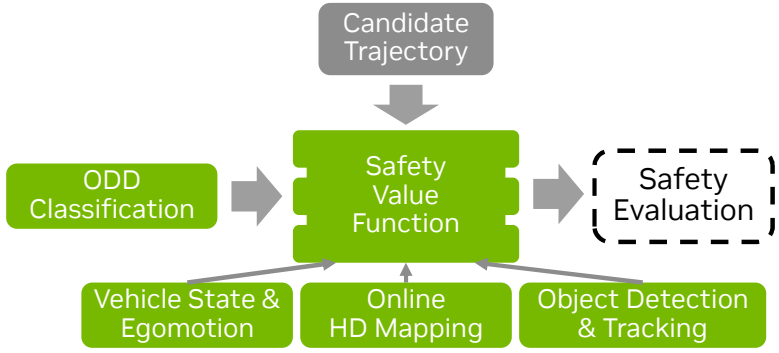
Run-Time Monitor Design

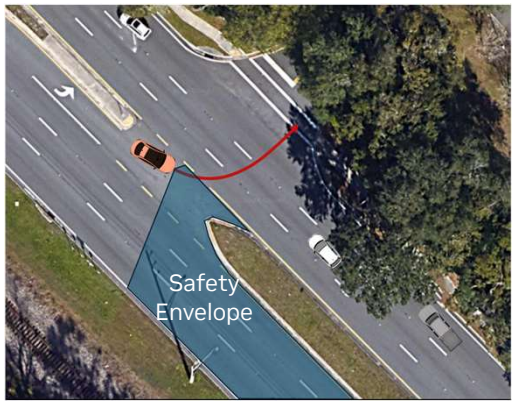


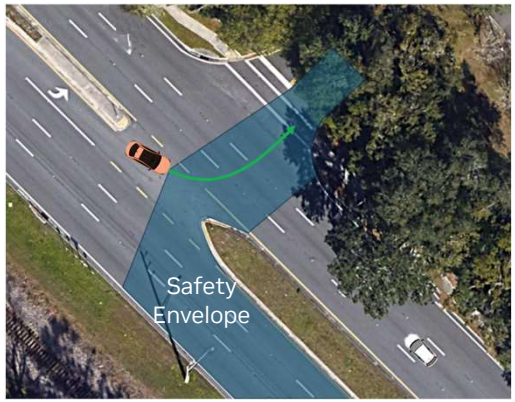
Safety Evaluation: Dynamics-Derived Safety Value Functions

Incorporating Safety From First Principles

- Value function: represents relative risk of future collision
- Perception-based
  - Maintaining an independent perception component enables explicit reasoning about safety of individual road users
- Dynamics-derived
  - Vehicle dynamics and ODD-specific assumptions may be used to compute a value function, defining a safety envelope
- Examples
  - Mobileye's Responsibility Sensitive Safety (RSS)
  - NVIDIA's Safety Force Field (SFF)
  - Reachability: a unifying analysis framework



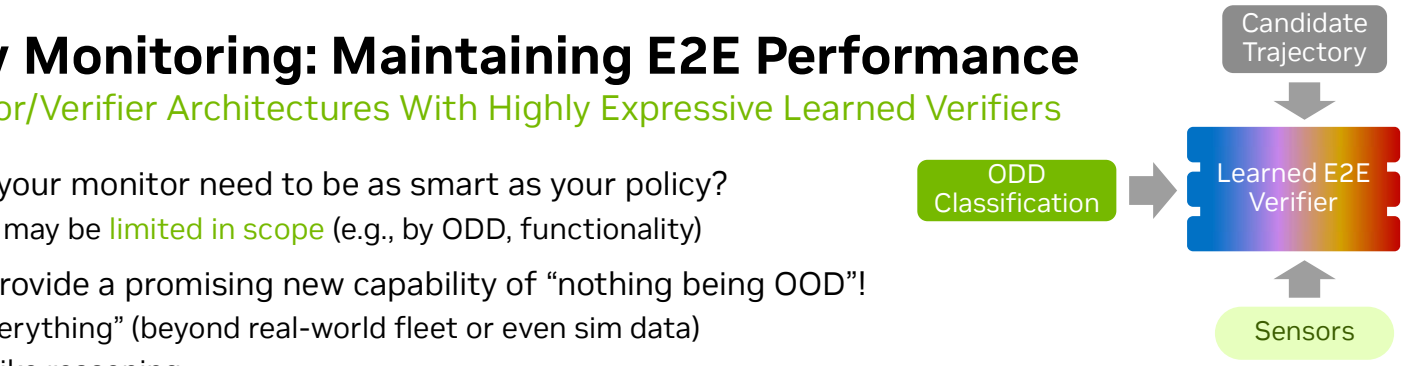


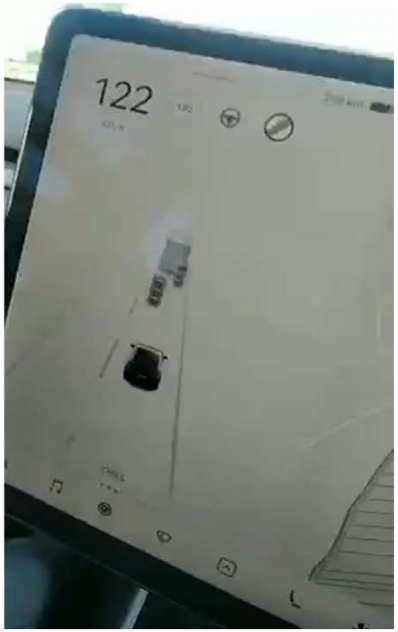


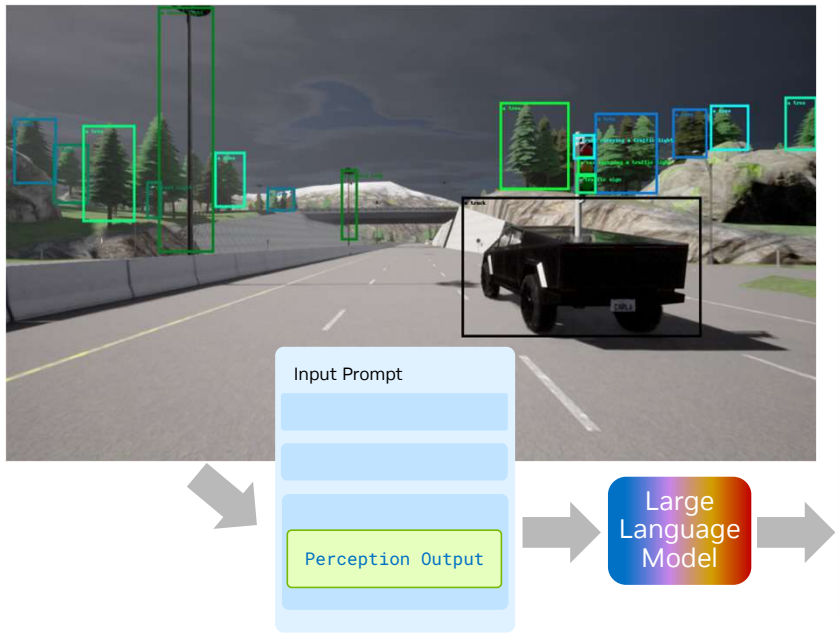
Safety Monitoring: Maintaining E2E Performance

Generator/Verifier Architectures With Highly Expressive Learned Verifiers

- To realize E2E benefits, does your monitor need to be as smart as your policy?
  - Yes in general, but monitors may be limited in scope (e.g., by ODD, functionality)
- Internet-pretrained models provide a promising new capability of “nothing being OOD”!
  - Monitors that have seen “everything” (beyond real-world fleet or even sim data)
  - Capability to mimic human-like reasoning







LLM Output

```
Truck: …
```

```
Tree:
1. Is this common to see while driving? Yes…
2. Can this influence the vehicle's behavior?
No, trees do not influence the vehicle's
behavior…

Classification: Normal.
```

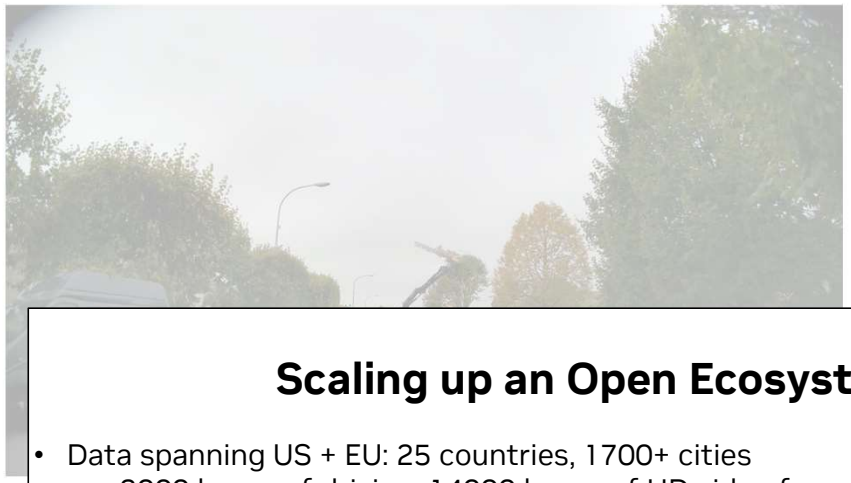
…

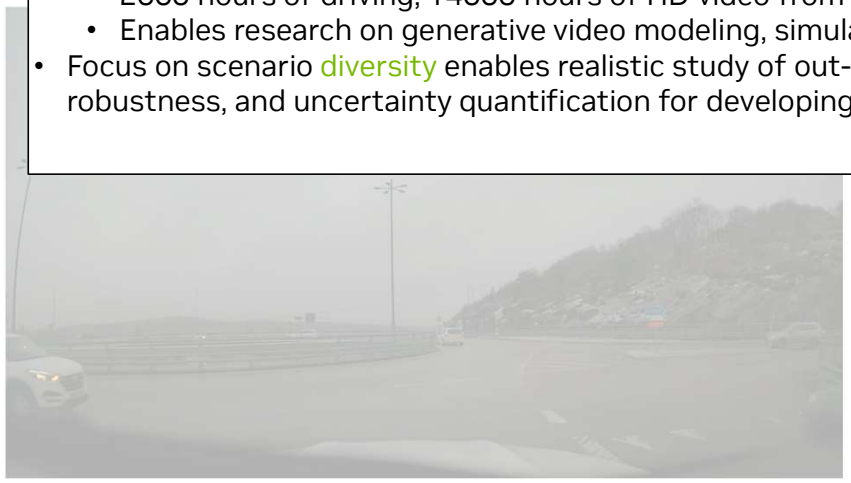
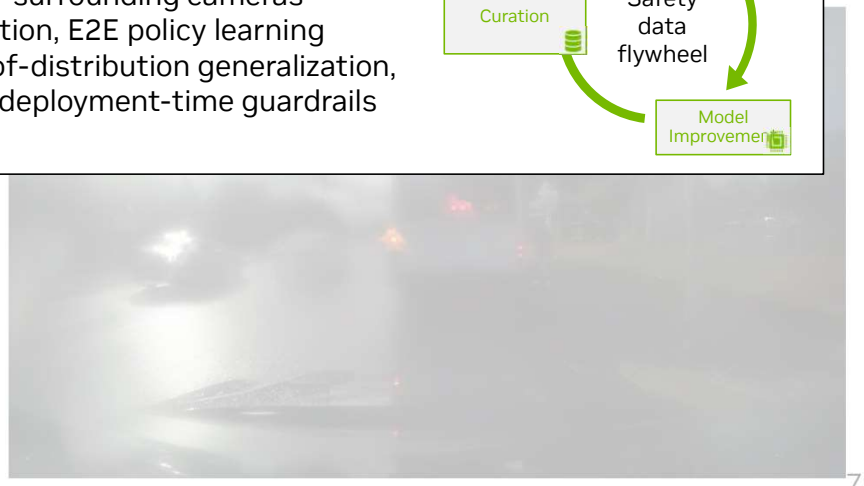
```
Truck carrying a traffic light:
1. Is this common to see while driving? No…
2. Can this influence the vehicle's behavior?
Yes, the autonomous vehicle may mistake the
traffic light for a real traffic signal…
3. Can the vehicle drive safely in its
presence? No, this could deceive the vehicle
into interpreting the traffic light as a legal
traffic signal.
4. Can this cause the vehicle to make
unpredictable or unsafe maneuvers? Yes, this
could deceive the autonomous vehicle into
driving forward when it should otherwise be
stopped (e.g., at a red light).
Classification: Anomaly.
```

```
Overall Scenario Classification: Anomaly.
```


Elhafsi, Sinha, Agia, Schmerling, Nesnas, Pavone, *Semantic anomaly detection with large language models*, Autonomous Robots, 2023
Sinha, Elhafsi, Agia, Foutter, Schmerling, Pavone, *Real-time anomaly detection and reactive planning with large language models*, Robotics: Science and Systems, 2024

NVIDIA Physical AI Dataset – AV Safety



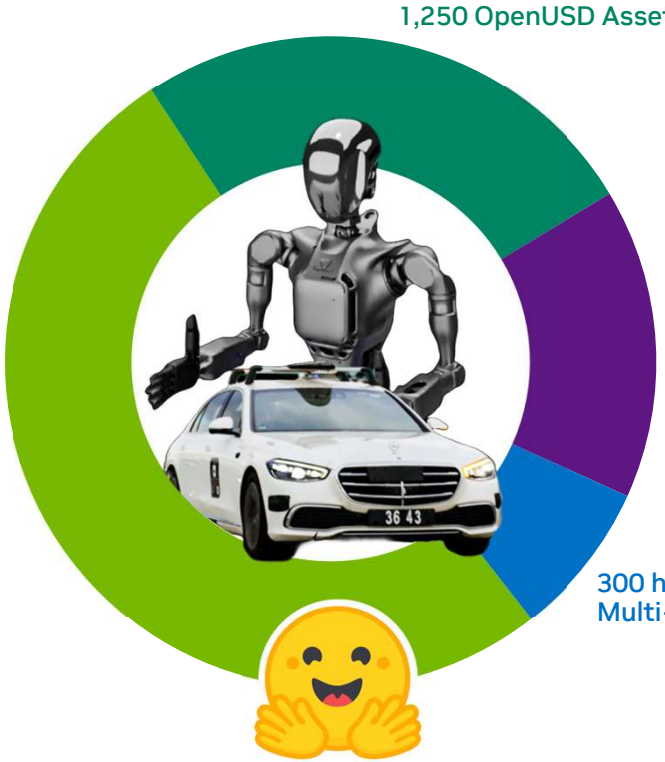



Scaling up an Open Ecosystem for E2E AV Development

- Data spanning US + EU: 25 countries, 1700+ cities
  - 2000 hours of driving, 14000 hours of HD video from 7 surrounding cameras
  - Enables research on generative video modeling, simulation, E2E policy learning
- Focus on scenario diversity enables realistic study of out-of-distribution generalization, robustness, and uncertainty quantification for developing deployment-time guardrails

Announcing New Open Source Physical AI Dataset

Massive Dataset Open-Source Dataset to Advance Robotics, AV Development



- Validated SimReady real and synthetic data
- Commercial-use
- More data to be released over time
- Available at GTC on Hugging Face

Chapter 4: Validation-Time Guardrails

Metrics, Coverage, and Data Flywheel and Processes

Synoptic View of NVIDIA Halos AV Safety Day

Design-time

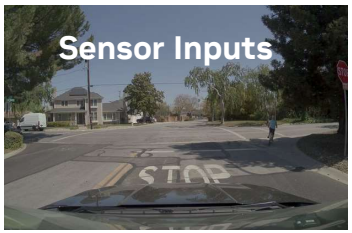
Large-Scale Data

Design-time safety (Chapter 2):
- Safety architecture
- AI train-time safety
- AV platform safety
- Data flywheel and processes

E2E

Validation-time guardrails (Chapter 4):
- Metrics
- Coverage – top-down
- Coverage – bottom-up
- Data flywheel and processes

Run-time



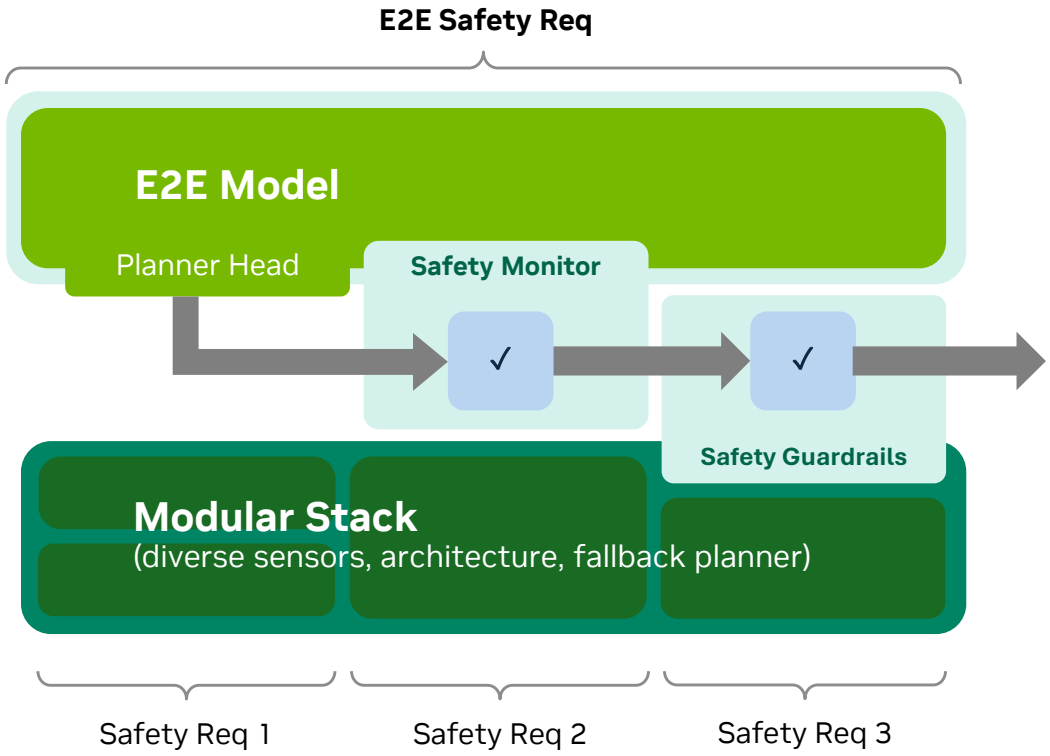
Deployment-time guardrails (Chapter 3):
- Run-time monitoring – HW
- Run-time monitoring – SW
- Arbitration
- Data flywheel and processes

Safety Regulation and Standardization in the Era of AI-Based AV (Chapter 5):
- Standardization challenges
- Regulatory challenges
- NVIDIA AI Systems Inspection Lab

From AVs to general Physical AI (Chapter 6):
- How Halos extends to Physical AI
- NVIDIA IGX elements
- Outside-in safety

Key Challenge: Validating E2E Safety

Measurability & Coverage

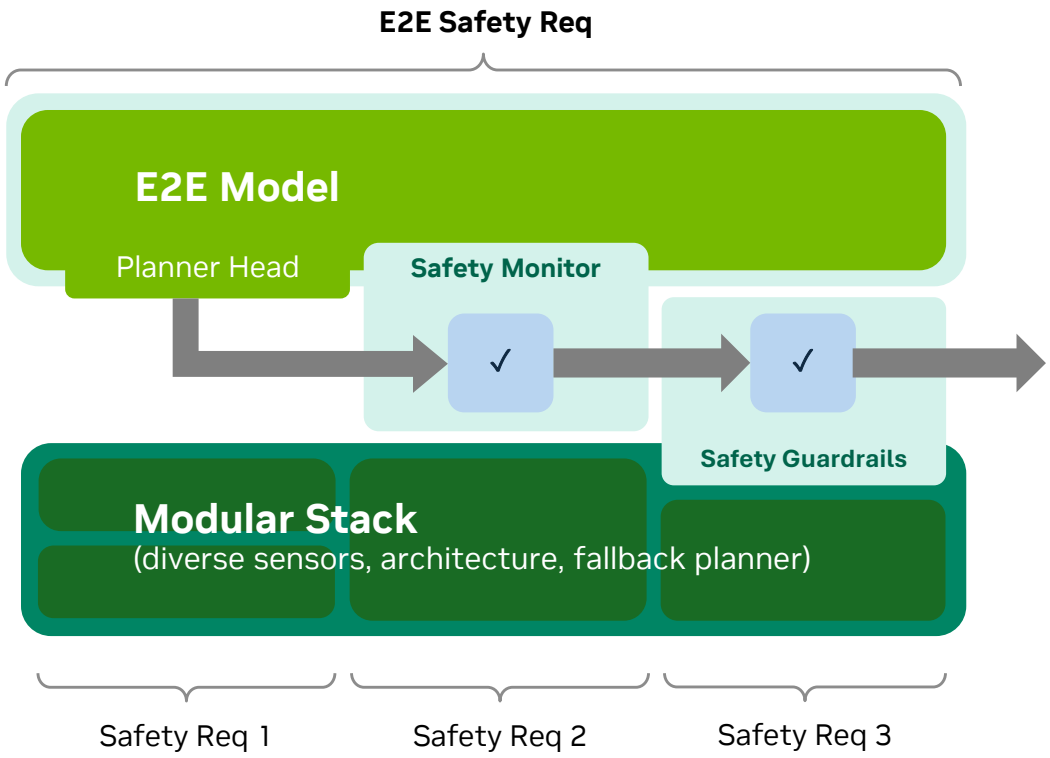


Defining E2E Safety
Relying on E2E components increase the weight on E2E behavioral metrics, but how do we define safe behavior in complex driving situations?

Achieving Test Coverage
How do we measure and ensure we achieve good coverage of the high-dimensional input space in closed-loop testing?

Key Challenge: Validating E2E Safety

Measurability & Coverage



Defining E2E Safety

Relying on E2E components increase the weight on E2E behavioral metrics, but how do we define safe behavior in complex driving situations?

Achieving Test Coverage

How do we measure and ensure we achieve good coverage of the high-dimensional input space in closed-loop testing?

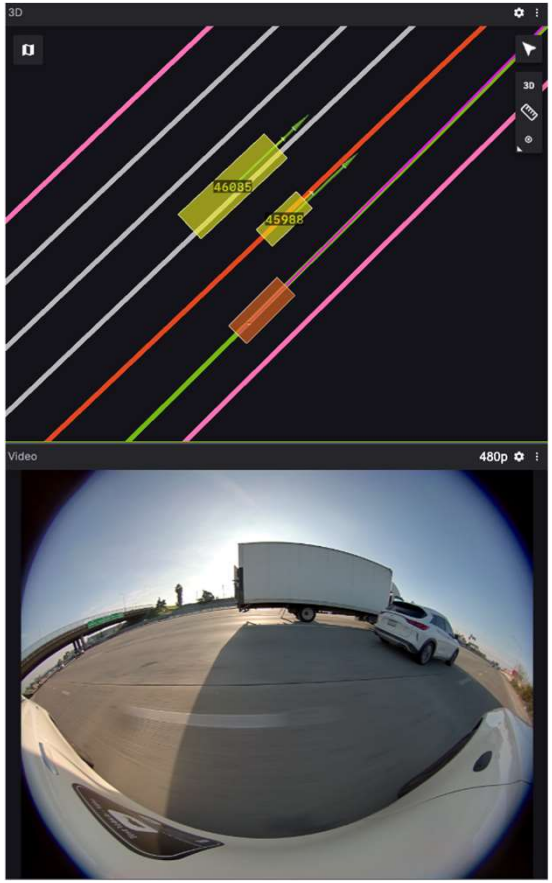
Defining E2E Safety: Quantifying Risk as a KPI

Reasoning About Possible Future Behavior

Goal: Identify when the car is in a risky state

- Ego's action requires unrealistic reactions by other road users
  - "If **the ego accelerates**, the only way to avoid a collision is if **the contender responds by accelerating, which is unrealistic.**"
- Other's actions could require unrealistic reactions by the ego
  - "If the **ego doesn't start braking now**, then it will likely **need to slam on the brakes** to avoid collision in the future."

Core Challenge: How will road agents behave in the future?



Reachability of Dynamical Systems

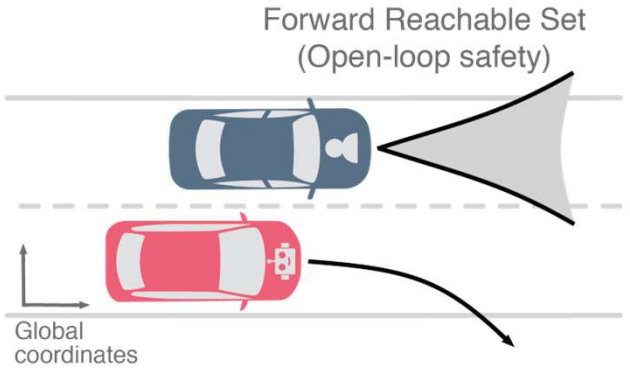
Reasoning About the Future

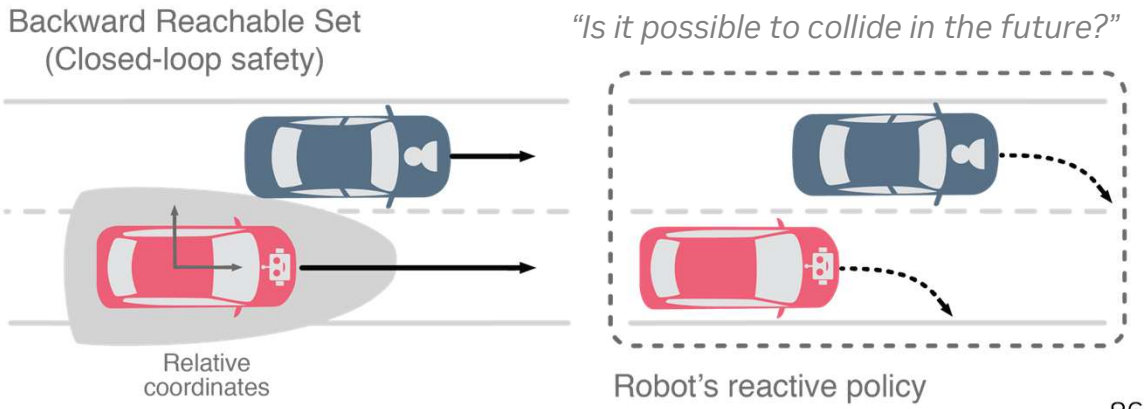
Estimating **forward looking risk** from the **current state** can be framed as an **optimal control problem.**

Dynamical systems reachability theory provides a mathematical framework for reasoning about a range of possible outcomes, not just a single prediction.

Safety concepts like Mobileye Responsibility Sensitive Safety (RSS) and NVIDIA's Safety Force Field (SFF) can be viewed as instantiations of this mathematical framework.

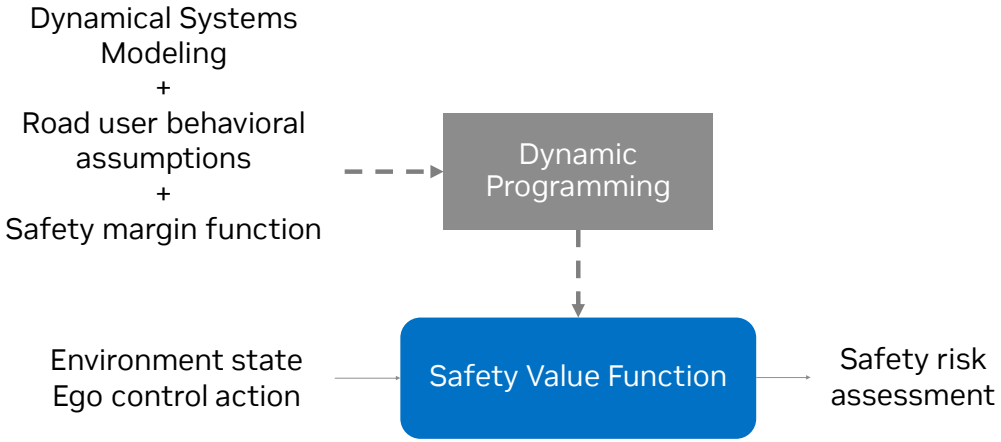
"Where can an agent reach in the future?"





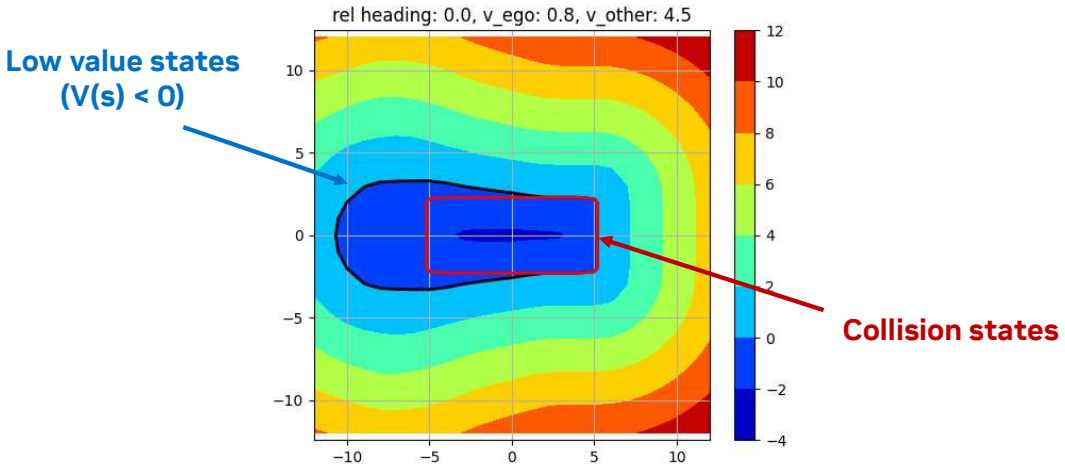
"Is it possible to collide in the future?"

Computing a Safety Value Function



Reachability lets us translate first principles knowledge of vehicle dynamics, assumptions on road user behavior, and a definition unsafe outcomes into a **safety value function**.

This value function maps the environment state and an ego control action to a safety risk assessment: **is it possible for the system to reach an unsafe outcome** (e.g. collision).

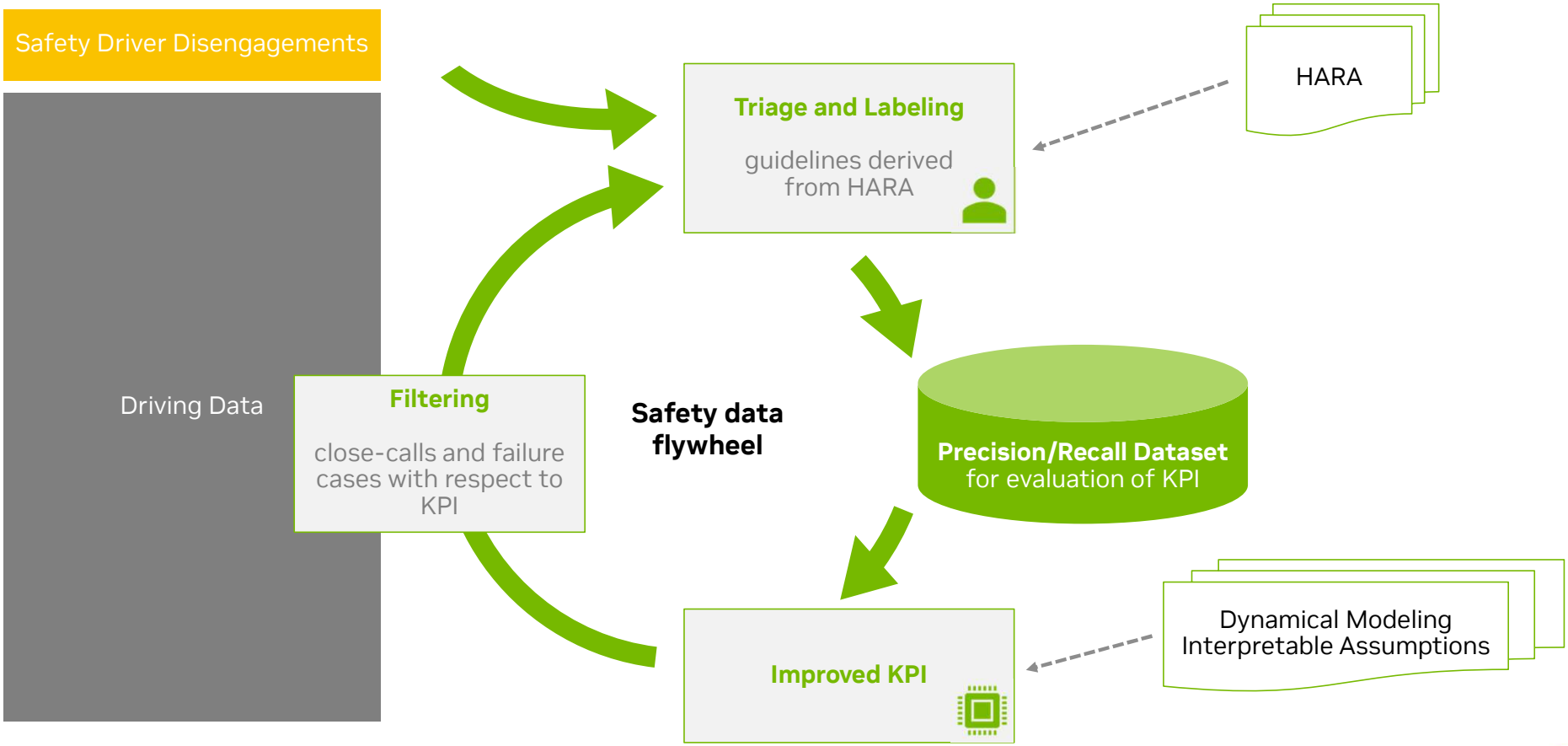


Design Choices:

- Assumptions on road-user future behavior
  - e.g., Reaction time, control limits
  - Data driven prediction or rule-based constraints
- Environment state representation
  - Factoring in aspects like ODD, map geometry, etc.

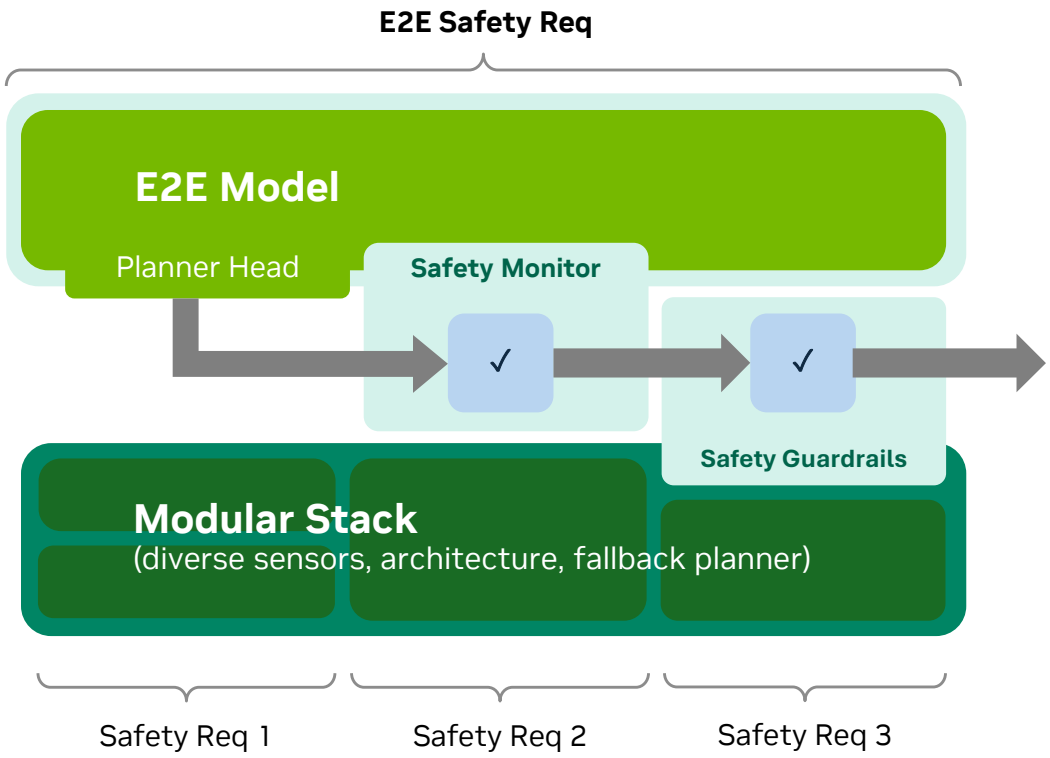
Measuring and Improving Alignment of Safety KPIs

Accelerating Triaging and Human Safety Assessment



Key Challenge: Validating E2E Safety

Measurability & Coverage



Defining E2E Safety

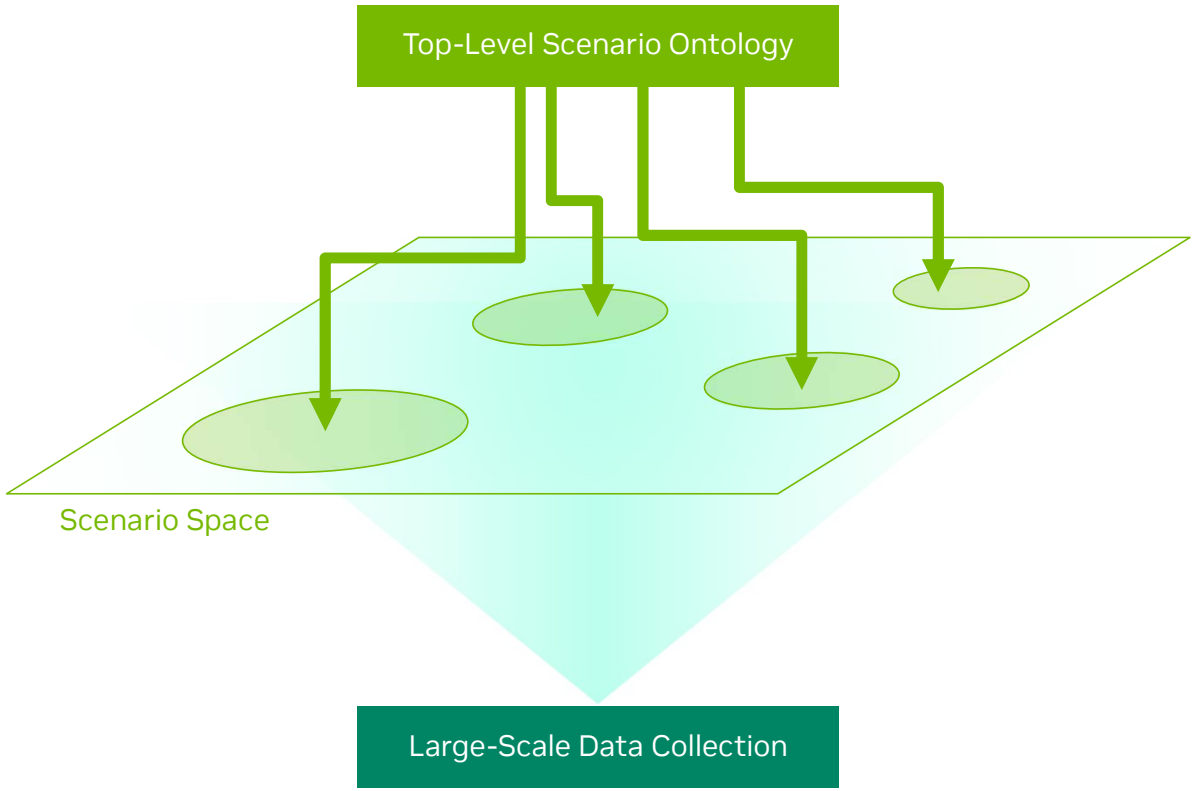
Relying on E2E components increase the weight on E2E behavioral metrics, but how do we define safe behavior in complex driving situations?

Achieving Test Coverage

How do we measure and ensure we achieve good coverage of the high-dimensional input space in closed-loop testing?

E2E Systems Demand a Holistic Approach to Coverage Analysis

Top-Down & Bottom-Up



Top-Down Coverage:

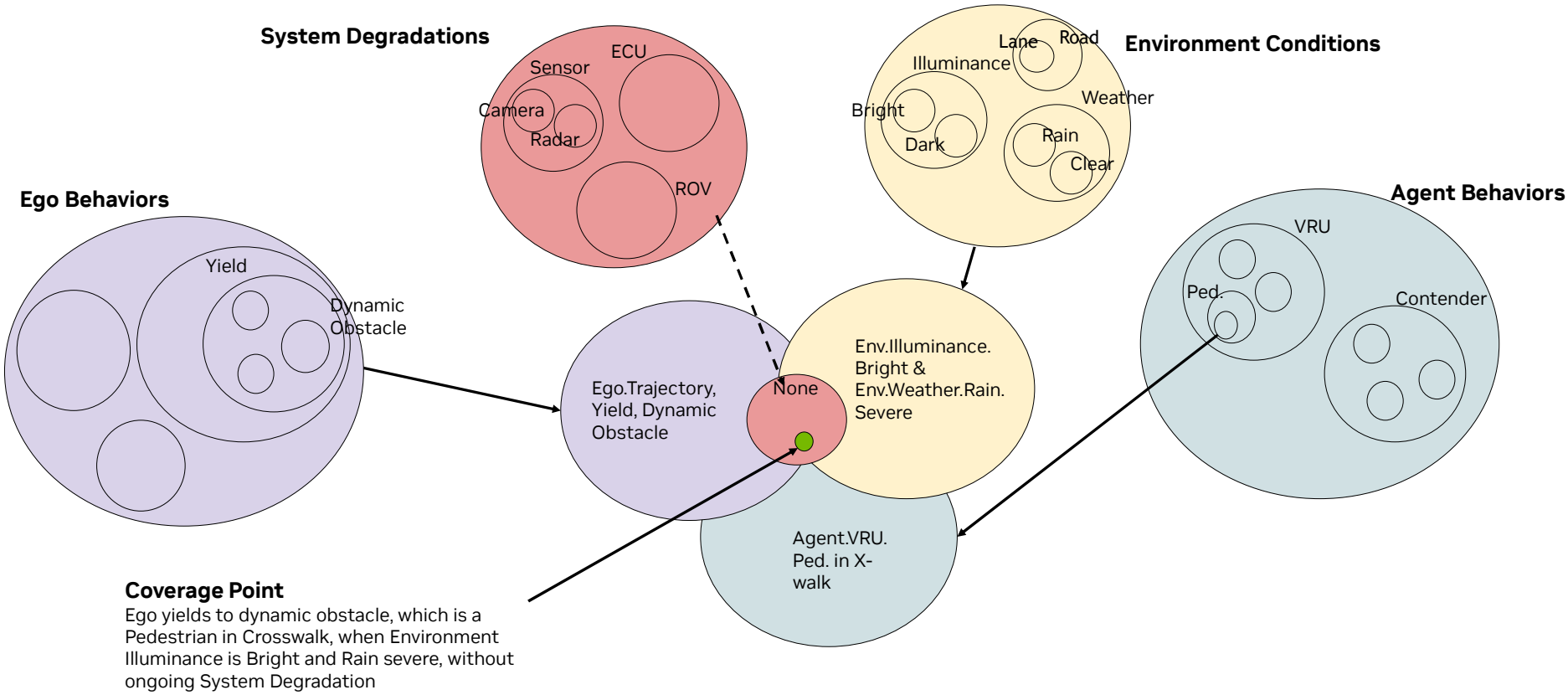
Top-level, first-principles safety analysis guides targeted testing of safety critical scenarios

Bottom-Up Coverage:

Large scale, broad sampling of input space surfaces additional failure modes and addresses gaps in top-down hazard analysis

Validation Input Space is Infinite

There May Be Hundreds of Dimensions In Practice



Coverage Point
Ego yields to dynamic obstacle, which is a Pedestrian in Crosswalk, when Environment Illuminance is Bright and Rain severe, without ongoing System Degradation

Top-Down Coverage Analysis: Challenges

Unbounded, Sparse Input Space

- Many dimensions, each of them large.
- Exponential scaling of scenarios to cover
- Infeasible to validate all combinations of scenarios

A Tree of Labels

Ontology Tree to Map and Abstract Input Space

- Connections between branches / dimensions constrained by semantic rules & relationships
- Trace along branches or along rule chains to find related coverage groups
- Identify areas which are highly correlated or invariant to some dimensions to reduce the total number of coverage points needed

➔ Pruned scenario tree that's feasible to validate against

An Ontology

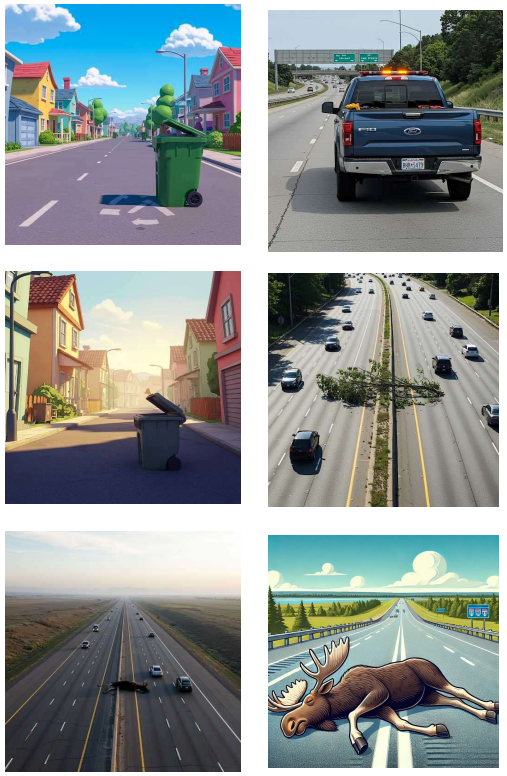
Example Scenario

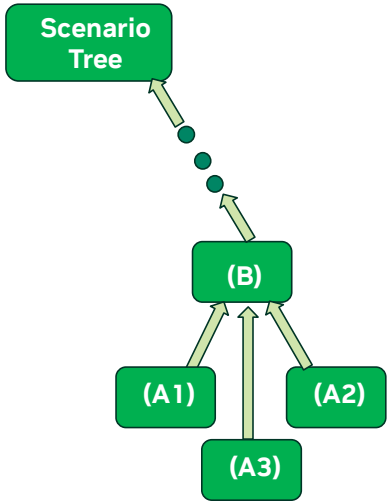
- Breakdown of the scenario
  - Subject = ego vehicle
  - Action = avoiding
  - Object = stopped construction vehicle
  - ODD = highway
  - Conditions/constraints
    - Ego is driving on the same lane as the stopped vehicle
    - The current lane is not the only available lane

Partitioning Scenarios

- There are other scenarios demanding the same *behavior*
  - Subject = ego vehicle
  - Action = Avoiding
  - Object = **Unexpected Obstacle**
    - stopped vehicle, garbage can, fallen tree, big dead animal etc.
  - ODD = highway, urban, residential, etc.
  - Conditions/constraints
    - Ego is driving on the same lane as the **Unexpected Obstacle**
    - The current lane is not the only available lane
  - Speed of ego = {…}
  - ….
  - ---

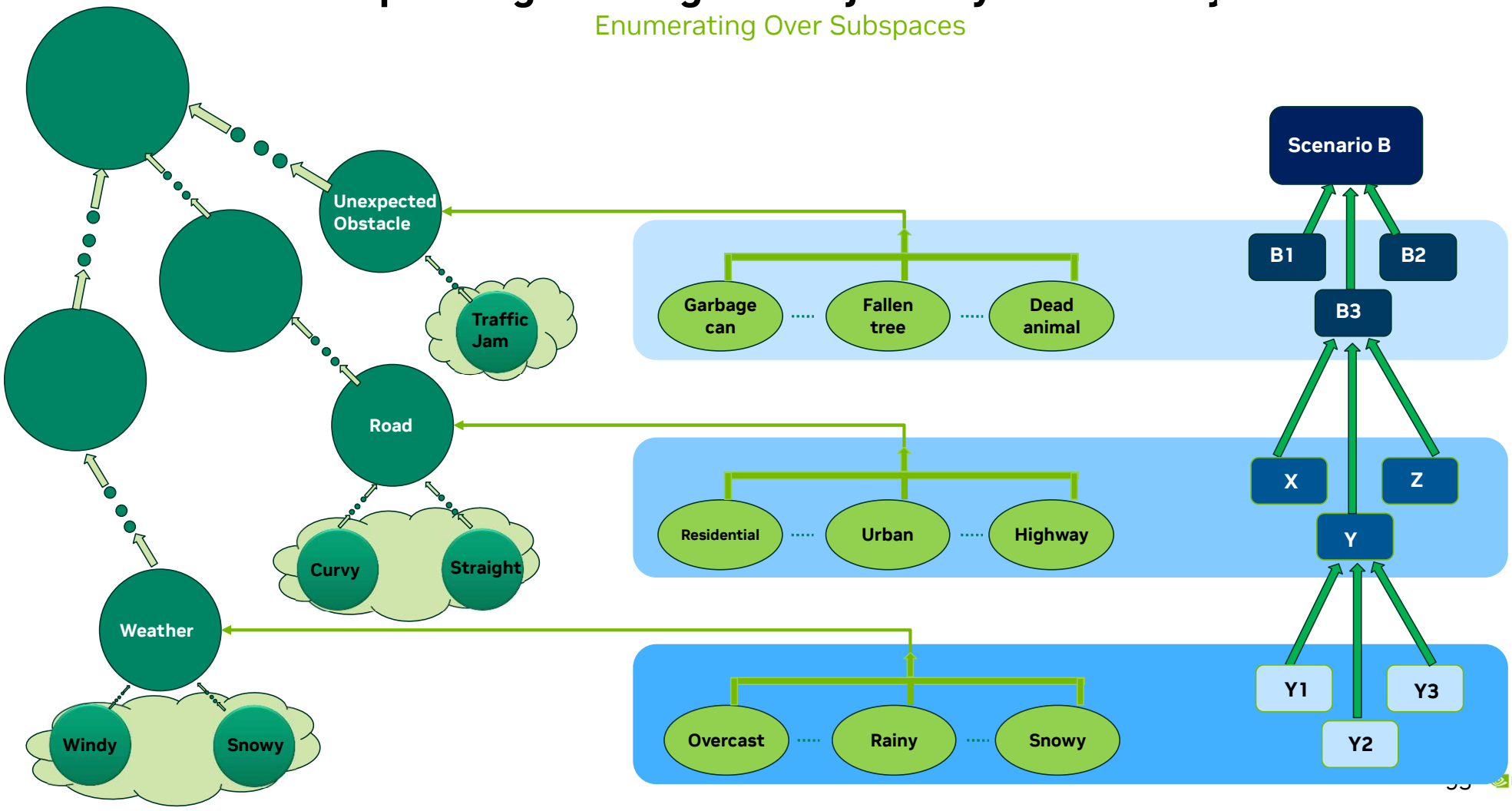
Abstract Generalization





Expanding Coverage on Adjacency & Similarity

Enumerating Over Subspaces



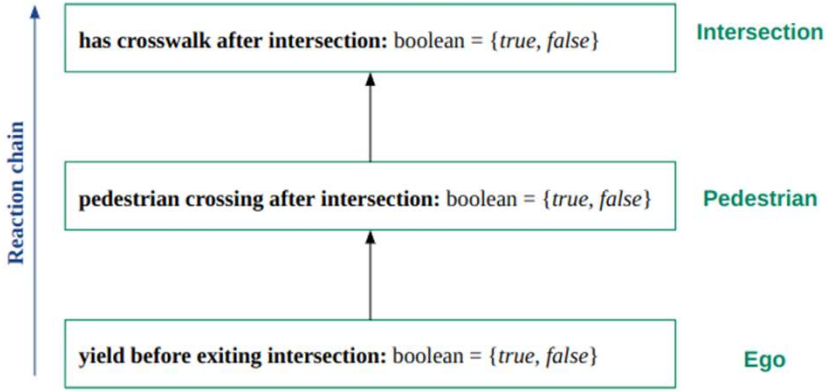
Coverage Dimensions for Traffic Signal Assist

Dimension of Intersections	Cardinality
decision sight distance: integer = range(10,100,10)	10
intersection sight distance: integer = range(10,100,10)	10
stopping sight distance: integer = range(10,100,10)	10
has crosswalk after intersection: boolean = {*true, false*}	2
has crosswalk before intersection: boolean = {*true, false*}	2
num lanes entering intersection: integer = [1,3]	3
num lanes exiting intersection: integer = [1,3]	3
pedestrian crossing after intersection: boolean = {*true, false*}	2
pedestrian crossing before intersection: boolean = {*true, false*}	2
num intersection legs: integer = [3,8]	6
Space size	**864,000**

Dimension of Ego Behavior	Cardinality
approaching direction: Direction = {*Left, Right, Straight*}	3
traffic light status before entering intersection: TrafficLightStatus = {*Red, Green, Yellow*}	3
traffic light status before exiting intersection: TrafficLightStatus = {*Red, Green, Yellow*}	3
exit after entering intersection: boolean = {*true, false*}	2
stopped before entering intersection: boolean = {*true, false*}	2
yield before exiting intersection: boolean = {*true, false*}	2
yield before entering intersection: boolean = {*true, false*}	2
Space size	**432**

Coverage Reduction

Dimension Dependency



Range Reduction

Dimension	Cardinality
decision sight distance: integer = {10,50,100}	3
intersection sight distance: integer = {10,50,100}	3
stopping sight distance: integer = {10,50,100}	3
num intersection legs: integer = [3,5]	3

Dimensions Decoupling

Dimension	Cardinality
has crosswalk after intersection: boolean = {*true, false*} **has crosswalk before intersection:** boolean = {*true, false*}	2
num lanes entering intersection: integer = [1,3] **num lanes exiting intersection:** integer = [1,3]	3
pedestrian crossing after intersection: boolean = {*true, false*} **pedestrian crossing before intersection:** boolean = {*true, false*}	2
num intersection legs: integer = [3,5]	3
Total	**36**

Dimension	Cardinality
decision sight distance: integer = {10,50,100}	3
intersection sight distance: integer = {10,50,100}	3
stopping sight distance: integer = {10,50,100}	3
Total	**27**

Space size = max(36,27) = **36**

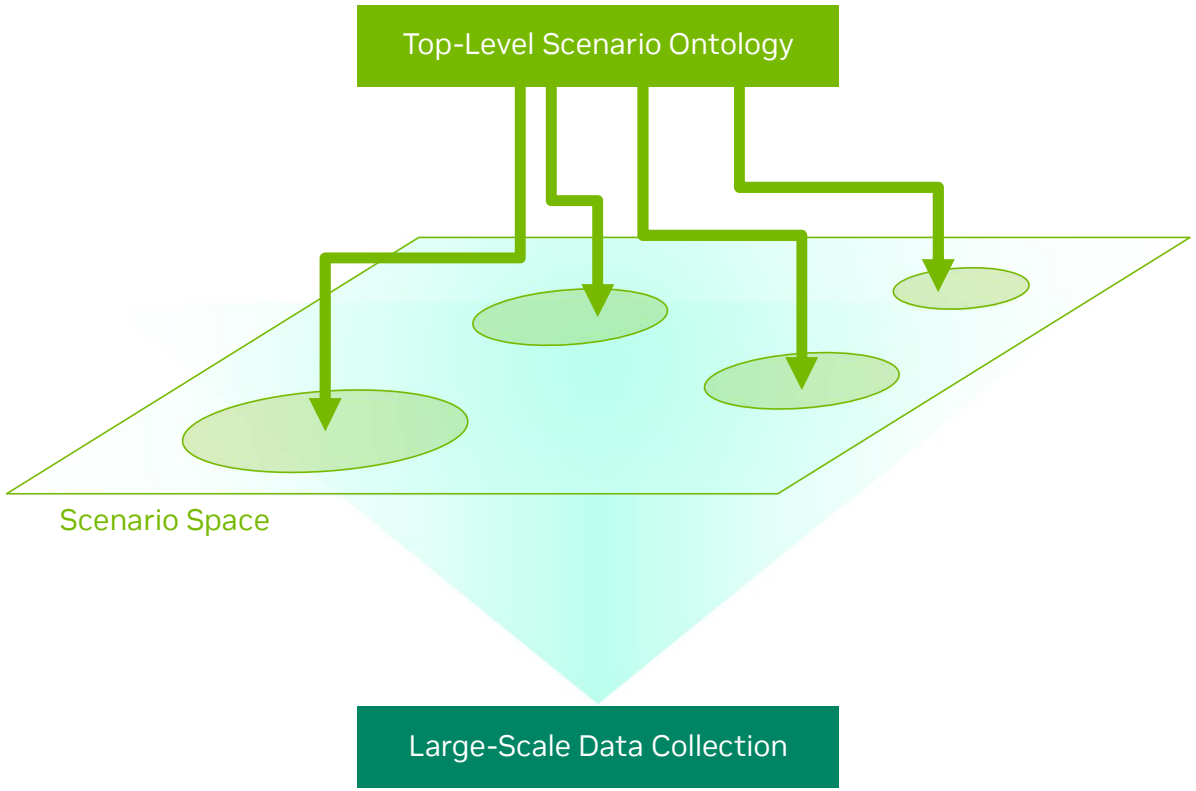
Scaling Coverage Analysis for Safety

- Expand coverage and compress validation input space
  - To ensure comprehensives and make validation feasible
  - Scenario clustering through ontology semantic relationship and based on embedding, PCA, K-means, t-SNE
  - Programmatically generate new scenarios through similarity/adjacency searches for Sim

- Combine with bottom-up coverage analysis with large data analysis
  - Start with safety critical scenarios and expand in related dimensions

E2E Systems Demand a Holistic Approach to Coverage

Top-Down & Bottom-Up



Top-Down Coverage:

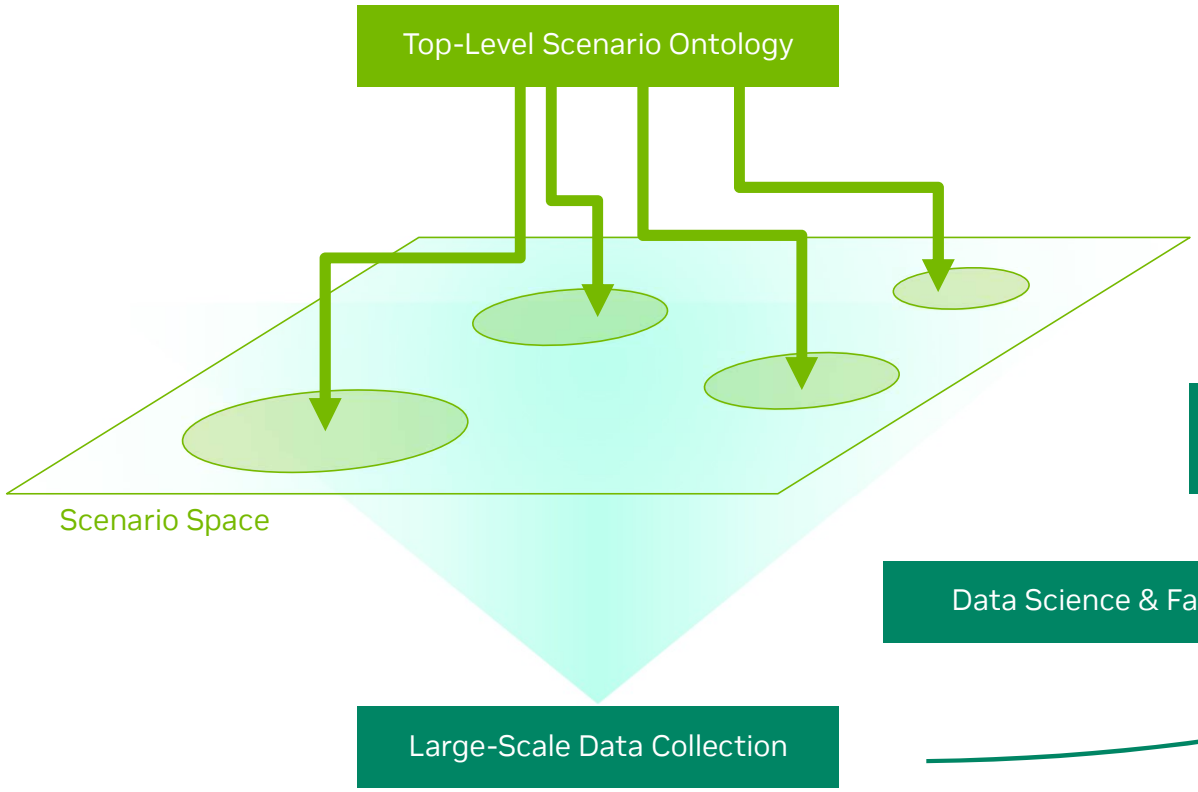
Top-level, first-principles safety analysis guides targeted testing of safety critical scenarios

Bottom-Up Coverage:

Large scale, broad sampling of input space surfaces additional failure modes and enables addressing gaps in top-down hazard analysis

E2E Systems Demand a Holistic Approach to Coverage

Top-Down & Bottom-Up



Bottom-Up Coverage:

Large scale, broad sampling of input space surfaces additional failure modes and enables addressing gaps in top-down hazard analysis

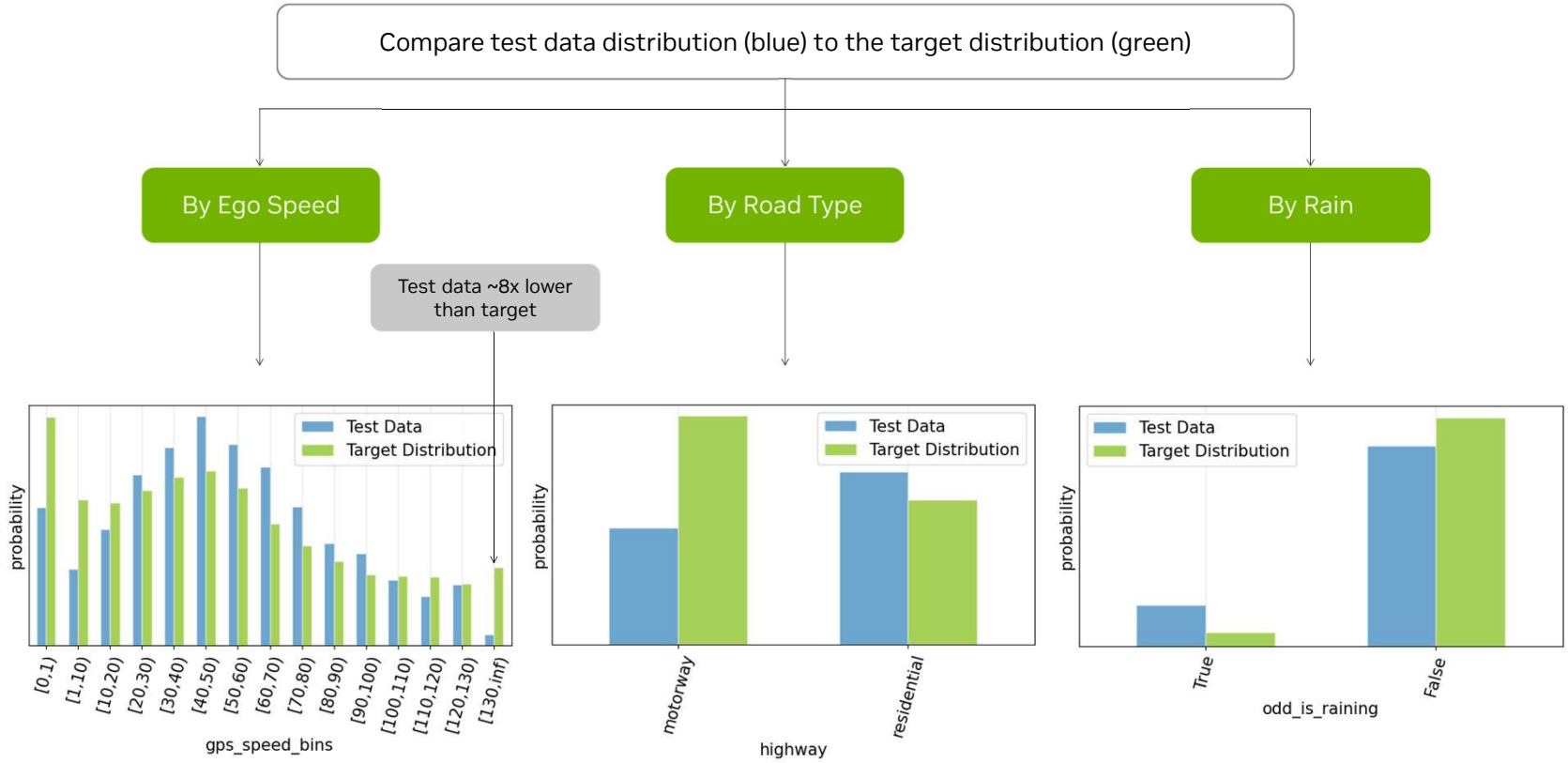
Data Science & Failure Mining

Expanded test-cases with E2E Sim

Improved Hazard Analysis & Risk Assessment

Data Science: ODD (Operational Design Domain) Coverage

Ensuring test data distribution is representative



Data Science & Failure Mining

Data Science: ODD conditioned failure analysis

Where Are We Failing?

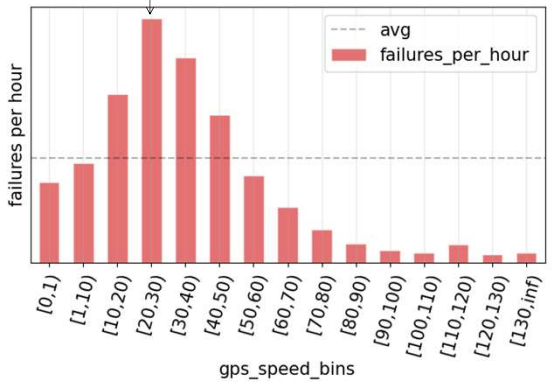
Data Science: ODD Breakdown of Failure Rates

Where Is the Failure Rate High?

In what ODDs are the weaknesses of the system?

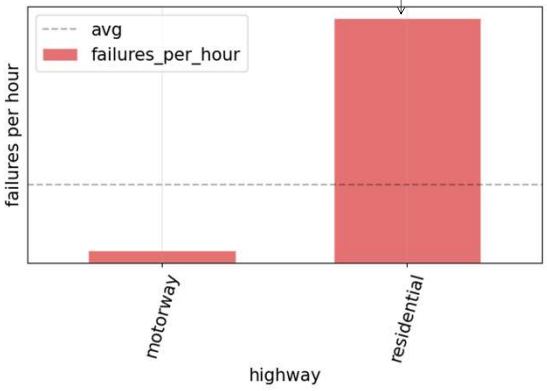
By Ego Speed

~2x higher than average

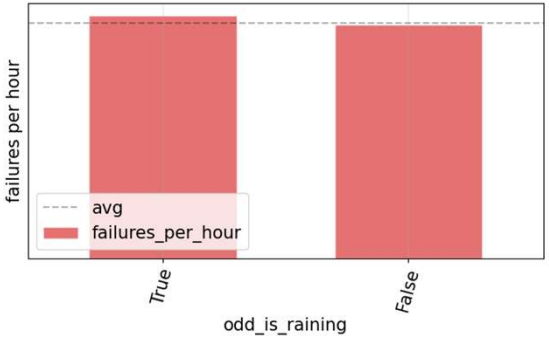


By Road Type

~20x higher than motorway

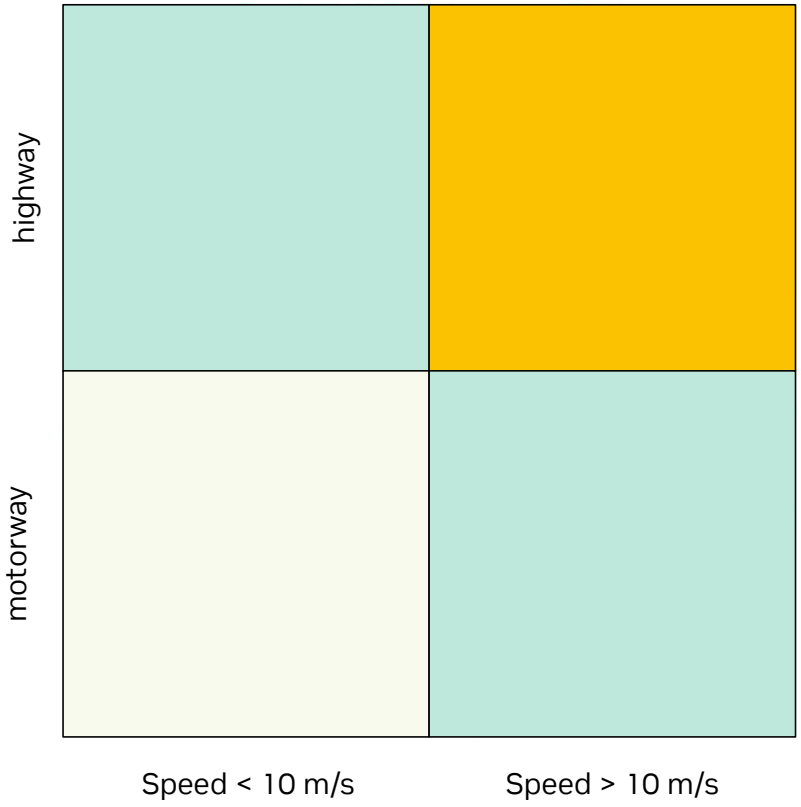


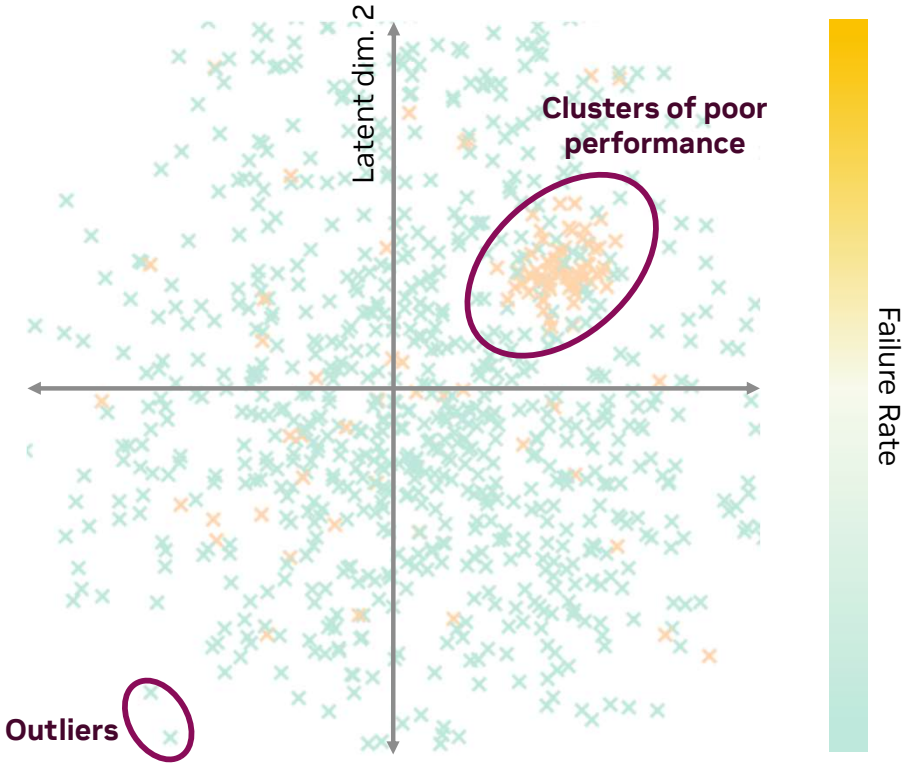
By Rain



Beyond Discrete Metadata: Continuous Embedding Models

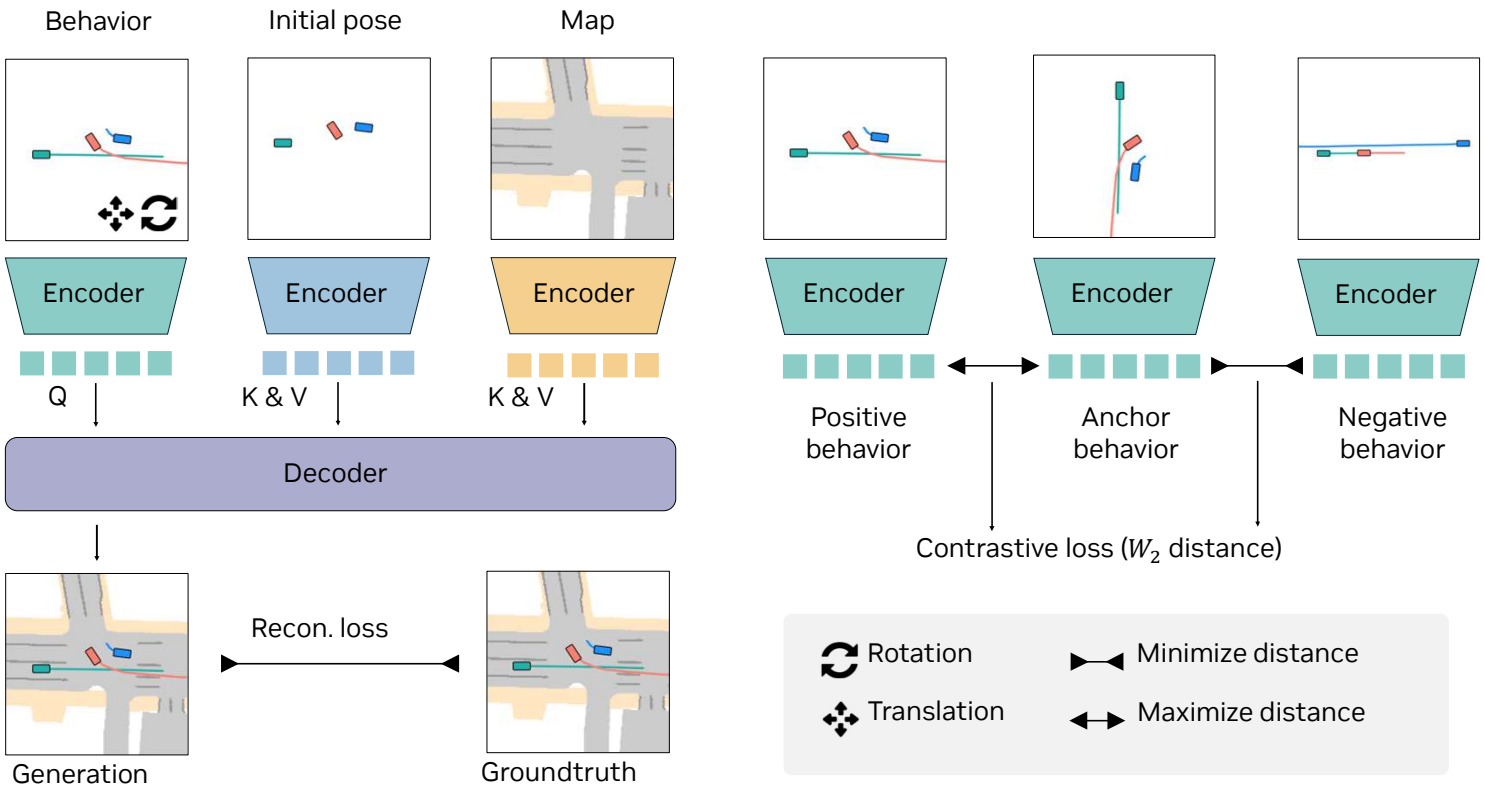
"Open-Vocabulary" Failure Discovery





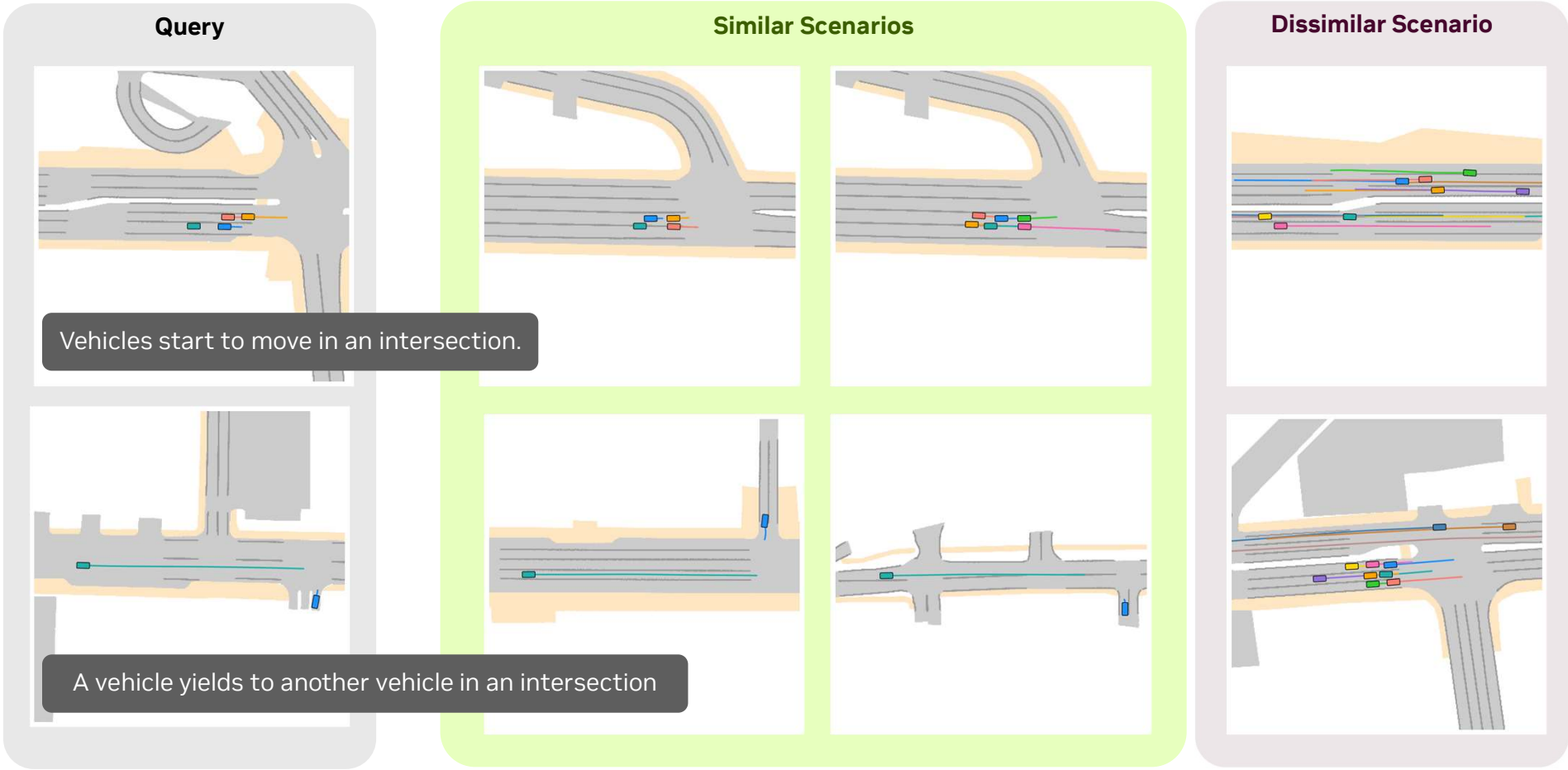
RealGen: A Scenario Embedding Model

Trained with reconstruction and contrastive losses



Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, Marco Pavone, "*RealGen: Retrieval Augmented Generation for Controllable Traffic Scenarios*", ECCV 2024 (Oral) https://realgen.github.io/

RealGen Scenario Embeddings Encode Semantic Similarity



Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, Marco Pavone, "*RealGen: Retrieval Augmented Generation for Controllable Traffic Scenarios*", ECCV 2024 (Oral) https://realgen.github.io/

Expanding Test-Cases With E2E Sim

Observed Rare Failure / Close-Call
(logged data)

E2E Sim Test Suite
closed-loop testing with local perturbations to scenario

An observed failure or close call can reveal rare but critical **triggering conditions**

- Do variations of the scenario also cause stack failures?
- Does an updated stack address this failure?

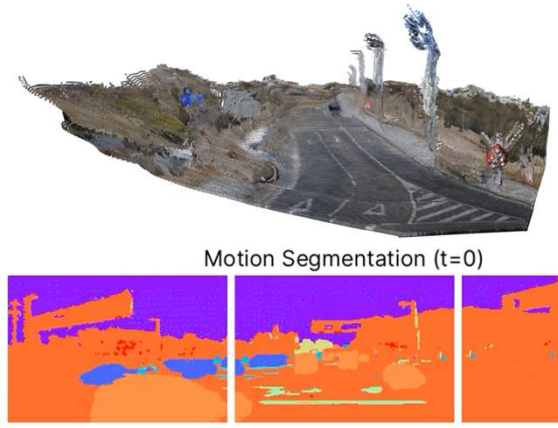
Halos leverages NVIDIA's E2E simulation technologies to answer these questions without waiting to observe the scenario again in real-world testing.

Expanding Test-Cases With E2E Sim

4D Reconstruction & Closed-loop Reactive Resimulation

Observed Rare Failure / Close-Call
(logged data)

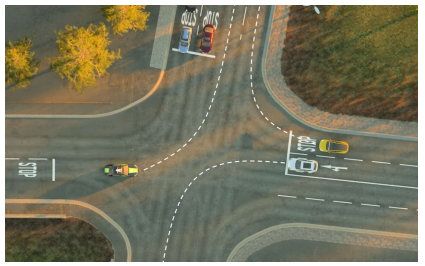
Reconstruction



4D Environment with Static/Dynamic Decomposition

(Optional)
User Prompts

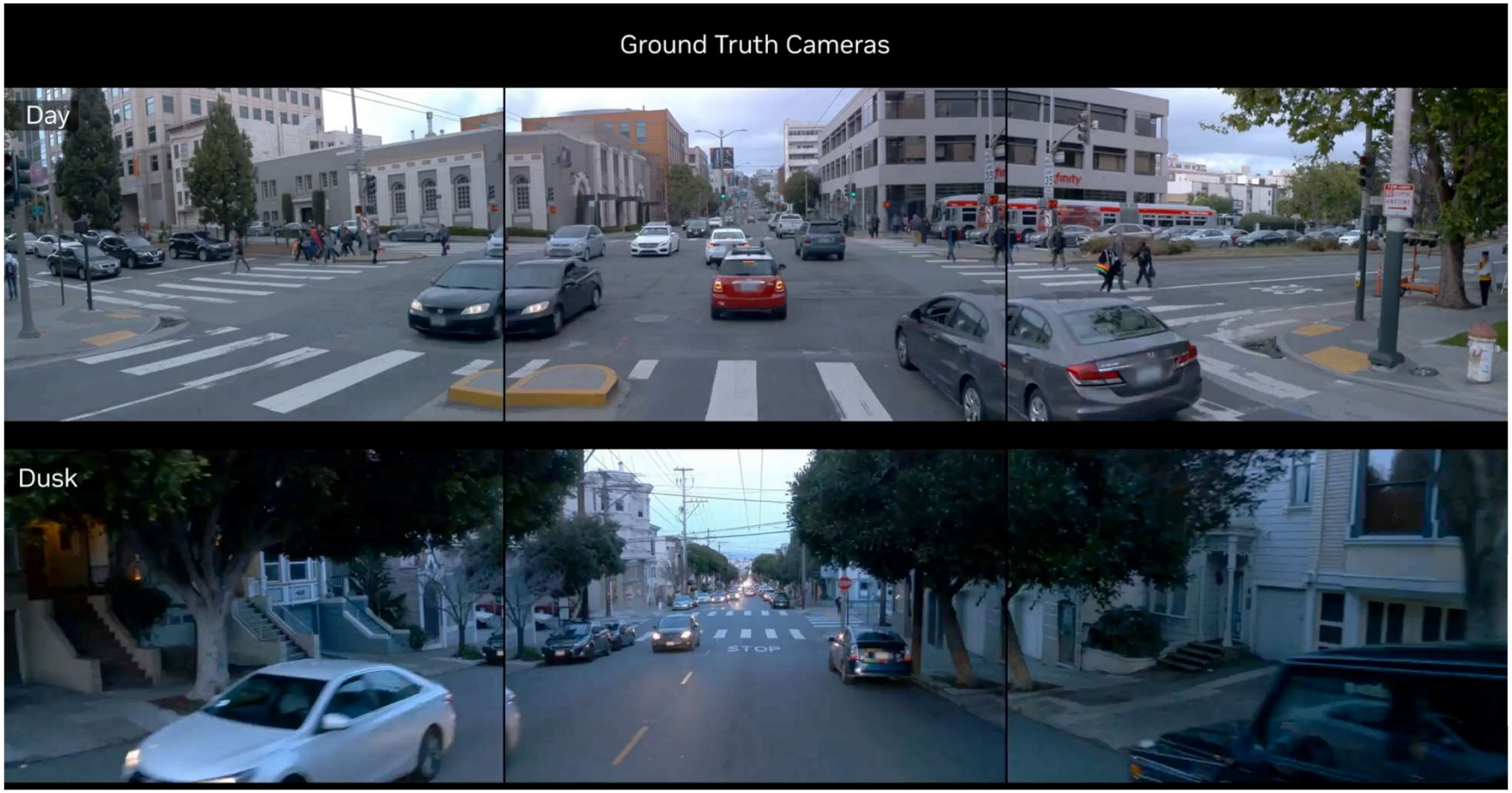
Resimulation



Reactive Traffic Models

E2E Sim Test Suite
closed-loop testing with local perturbations to scenario

Scene Reconstruction and Understanding



Learn more on DRIVE Labs

Yang, Ivanovic, Litany, Weng, Kim, Li, Che, Xu, Fidler, Pavone, Wang, *EmerNeRF: Emergent Spatial-Temporal Scene Decomposition via Self-Supervision*, ICLR 2024 https://emernerf.github.io/

Simultaneous Sensor *and* Traffic Simulation

Camera Log of the Scenario

Rendered Camera Log of New Scenario

Learn more on DRIVE Labs

Expanding Test-Cases With E2E Sim

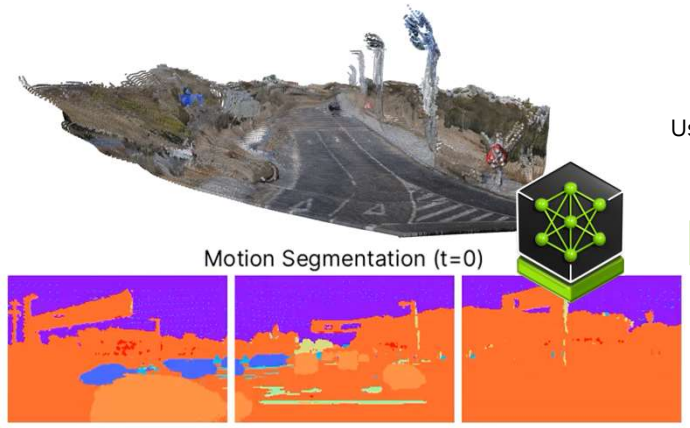
A Large Blueprint With Interchangeable Components Powered by Halos AV NIMs

Observed Rare Failure / Close-Call
(logged data)

and/or

Generated Data

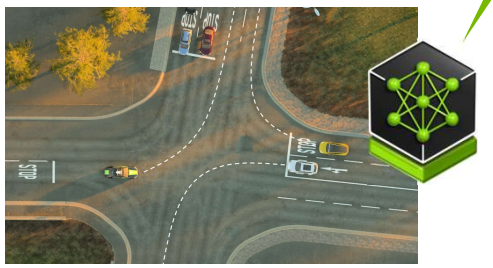
Reconstruction



4D Environment with Static/Dynamic Decomposition

(Optional)
User Prompts

Resimulation

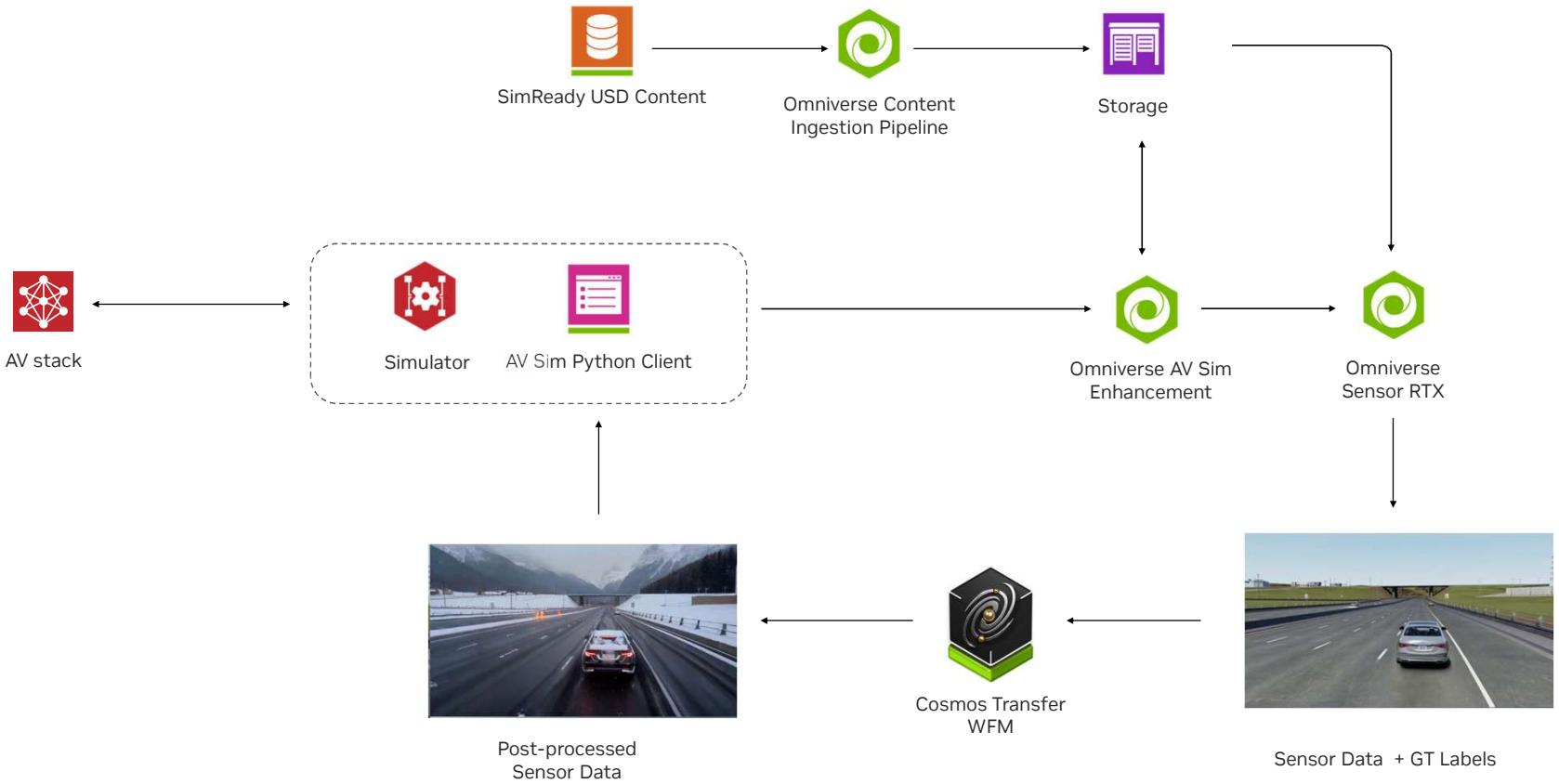


Reactive Traffic Models

E2E Sim Test Suite
closed-loop testing with local perturbations to scenario

Expanding Test-Cases With E2E Sim

NVIDIA Omniverse Blueprint for AV Simulation



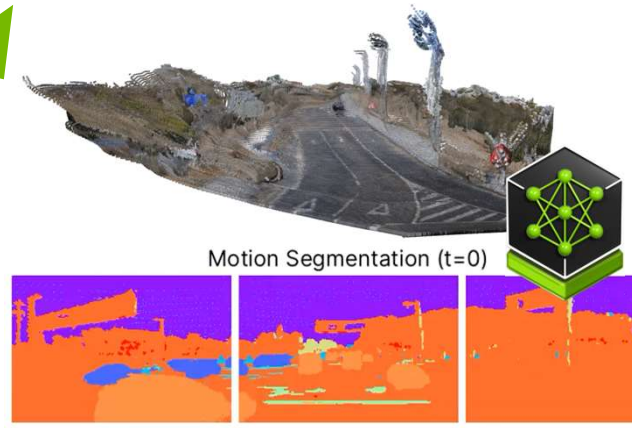
Expanding Test-Cases With E2E Sim

A large Blueprint With Interchangeable Components

Observed Rare Failure / Close-Call

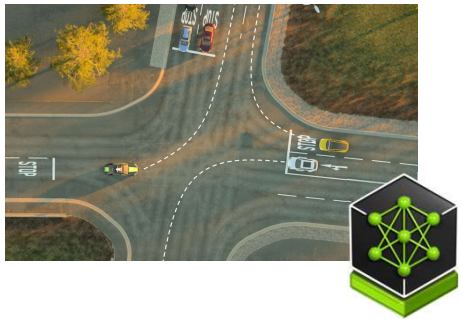
and/or

Generated Data



4D Environment with Static/Dynamic Decomposition

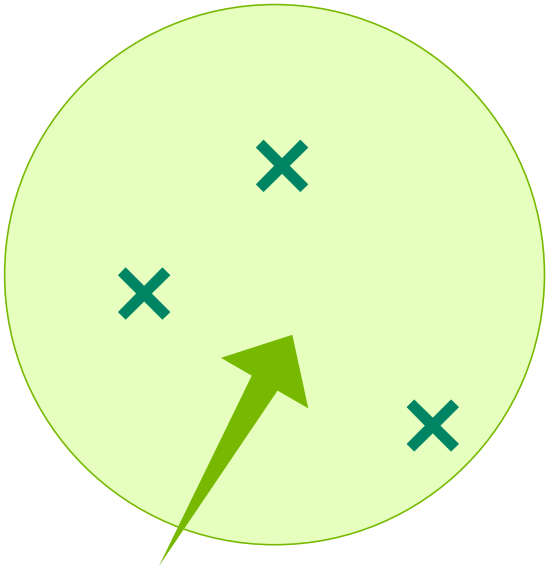
(Optional)
User Prompts



Traffic Models

E2E Sim Test Suite

closed-loop testing with local perturbations to scenario



Validating Simulators to Support Empirical Safety Arguments

Beyond targeted stress testing

- E2E sim is already invaluable for targeted stress testing.
- Can simulators lower real-world test data requirements for empirical data-driven validation arguments?
  - **Need statistical guarantees on correlations across simulation and real-world testing.**

$$\mathbb{E}[f_{\text{real}}] \mid \mathbb{E}[f_{\text{sim}}]$$

Aggregate Correlation
plenty of data,
but correlation is weaker

$$\mathbb{E}[f_{\text{real}} \mid \boldsymbol{z}] \mid \mathbb{E}[f_{\text{sim}} \mid \boldsymbol{z}]$$

Embedding-Conditioned Correlation
stronger correlation,
while remaining tractable

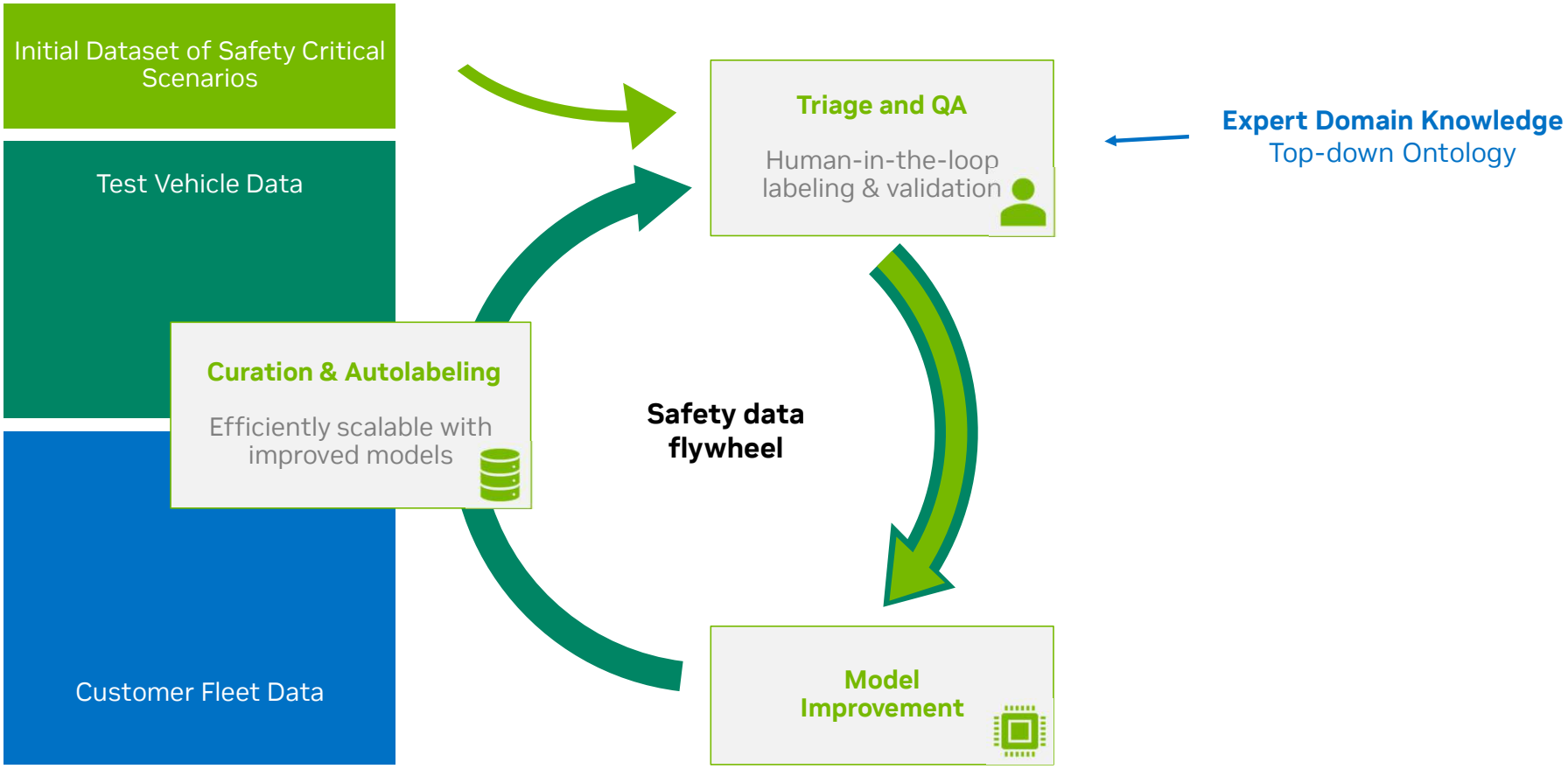
$$f_{\text{real}} \mid f_{\text{sim}}$$

Per-Scene Correlation
gold standard,
but intractable in practice

Scenario and video embedding models enable grouping scenes where metrics are highly correlated across evaluation domains.

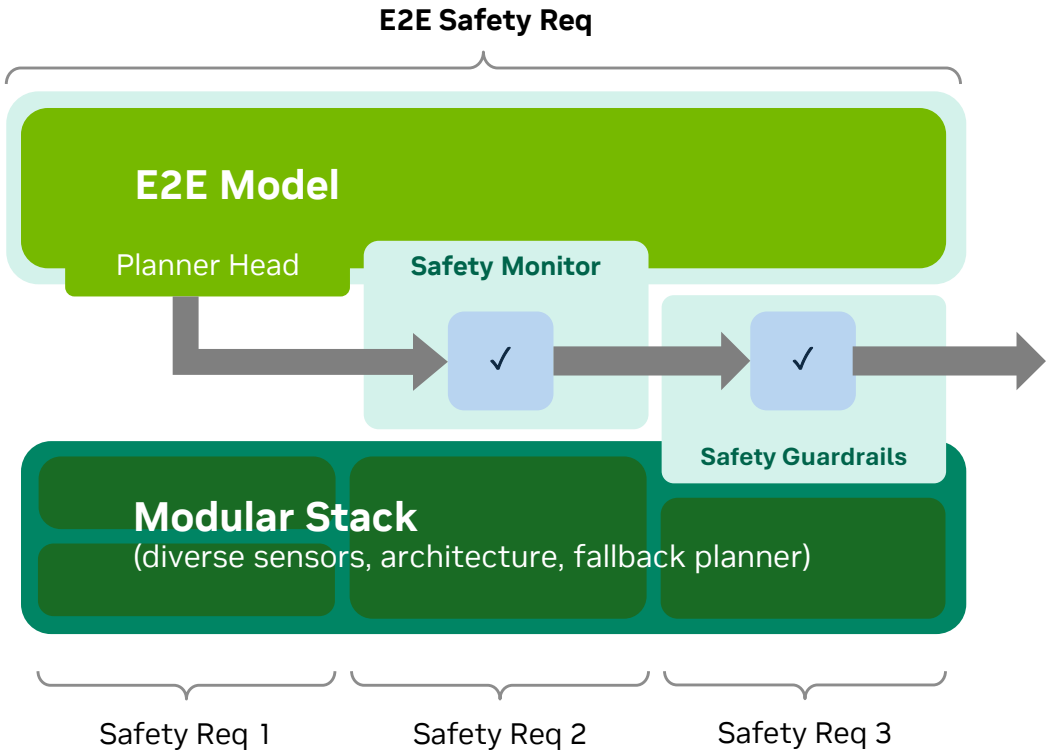
The Safety Data Flywheel

Continuously improving coverage



Key Tenets of Validating E2E Safety

Measurability & Coverage



Defining E2E Safety

- Define safety KPIs that measure **forward looking risks**, instead of solely relying on observed safety events

Achieving Test Coverage

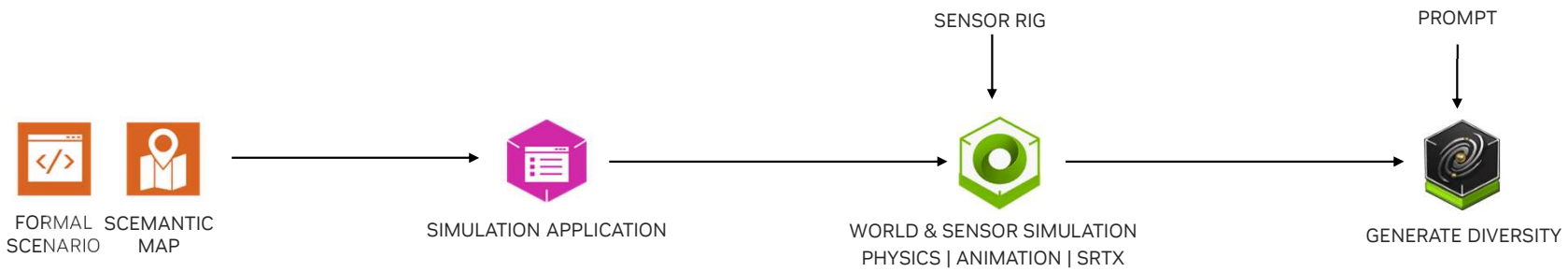
- Combine **ontology-based top-down** and **data-driven bottom-up** approaches to ensure comprehensive coverage.
- Leverage **closed loop E2E simulation** to augment real-world tests and stress test E2E models.
- Enable a **safety data flywheel** to continuously improve validation coverage and efficiency

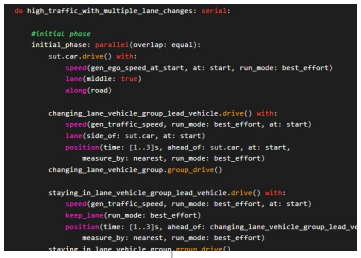
Tenets of E2E System Validation

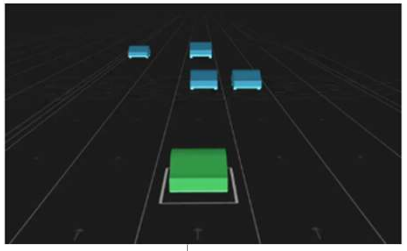
- Measure safety KPIs by measuring forward looking risks, instead of solely relying on observed safety events
- A hybrid coverage analysis ensures comprehensive as well as pragmatism
- Enables a data fly-wheel for creating, curating and evaluating scenarios and safety performance

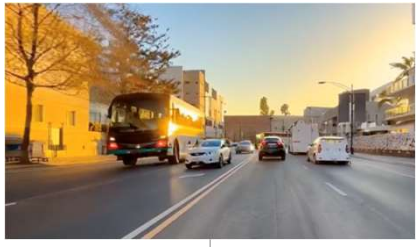
Generating Diverse Sensor Data Grounded in Physics

Using NVIDIA Omniverse Blueprint for AV Simulation with Cosmos Transfer









Author Scenario

- Formal
- Traceable
- Interpretable
- Fine Control

Simulate Agent Behavior

- Reactive
- Repeatable

Simulate World & Sensors

- Physically-based behavior
- Accurate sensor modeling
- Accurate timing
- Ground Truth ready

Generate Diversity

- Weather, ToD, Appearance
- Grounded in Physics

Ecosystem Enablement and Customer Stories

Wei Luo

VP Automotive

NVIDA Halos Elements

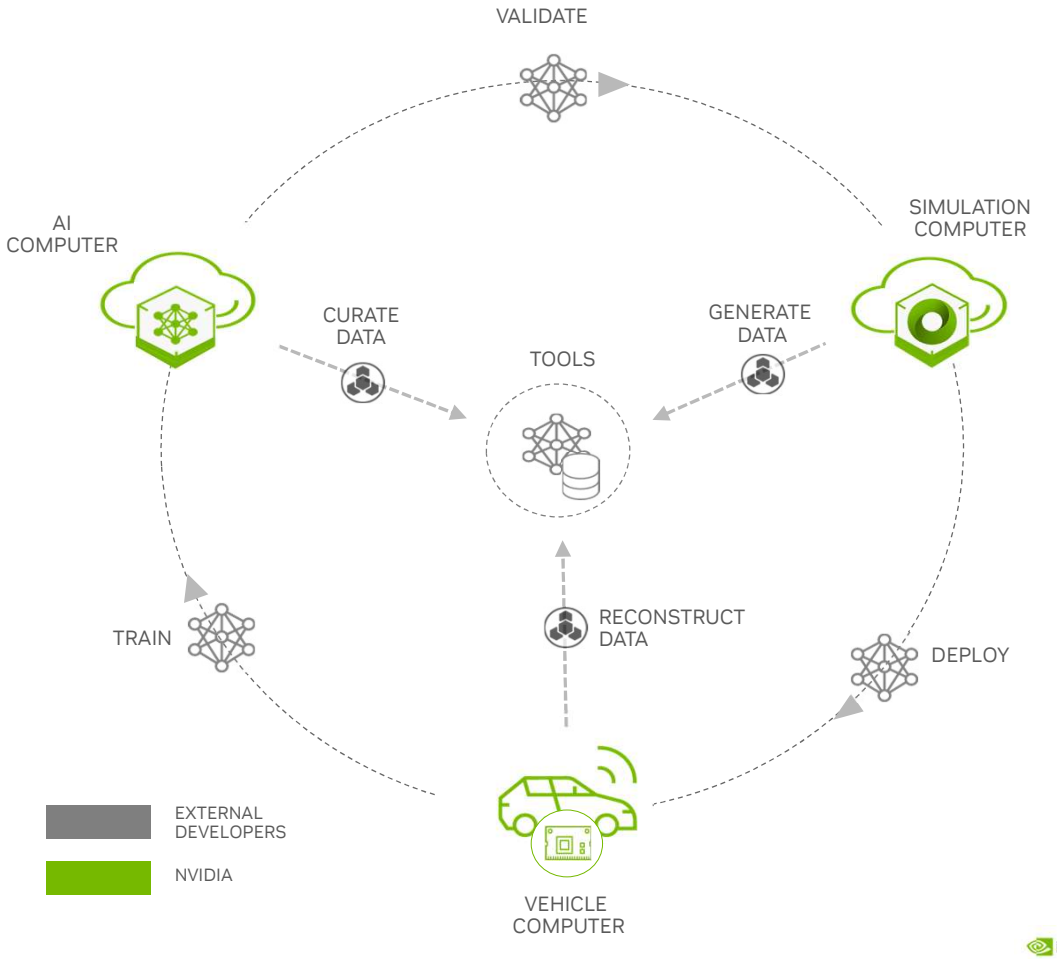
Full-Stack System for Autonomous Vehicle Safety

HW/SW and Platform Safety	Algorithmic Safety	Ecosystem Safety	AI Systems Inspection Lab
• Safety assessed HW (SoC and reference board) • Safety certified DriveOS • Safety assessed base platform • NVIDIA DRIVE AGX Hyperion™ • DriveOS Linux for Safety *(future offering)*	• Libraries for safety data loading and accelerators • API for safety data creation, curation, reconstruction • NVIDIA Omniverse™ and Cosmos for AV Simulation Blueprint to train, test, and validate AVs • Diverse AV stack that combines a modular stack and E2E AI models	• Safety data with diverse, unbiased data • Continual improvements through a safety data flywheel	• Leadership in AV safety standardization and regulation • First of its kind to be accredited by ANAB, Inspects and verifies the integration of partners' products with Halos' safety elements

Accelerating AV E2E Workflow with NVIDIA Cloud

NVIDIA Product Offering

- **Three Computers**
  - AI Computer
  - Simulation Computer
  - Vehicle Computer
- **Cloud APIs**
  - Data Curation and Search
  - Data Generation
  - Scene Reconstruction
- **Libraries**
  - Data Prep and Loading
  - Post Processing
- **Data**
  - Fleet Data
  - Synthetic Data



AV Simulation Blueprint

Tools / Platforms

The AV Simulation APIs in Omniverse are made possible by a large ecosystem of partners who bring valuable capabilities.

Verification and validation tools are helping developers understand coverage of scenarios for AV development, create new scenarios and validate these in simulation. As also to help curate and visualize generated synthetic data.

AV Simulation Blueprint

Sensors: Camera, Lidar, Radar

Developers such as sensor providers are able to use the RTX pipeline to build dedicated sensor models and bring these into Omniverse so that OEMs and AV developers can choose sensor models from specific vendors in their simulations.

AV Simulation Blueprint

Simulation Applications

There are a wide range of simulation applications that are also part of the AV simulation ecosystem including open-source solutions.

Thanks !

Scan the QR code below to visit NVIDIA Halos website and Contact Us to join the Halos community, or be informed on our future initiatives and get involved!

or contact:

mpavone@nvidia.com
rmariani@nvidia.com