woven planet

May 4, 2022

On the Road

From data to driving with Deep Learning

Ana Ferreira

Senior Software Engineer

ana.ferreira@woven-planet.global

Christian S. Perone

Senior Machine Learning Engineer

christian.perone@woven-planet.global

Photo by Ian Beckley

Talk agenda

Introduction
Imitation Learning
Data-Driven Approach for Planning
Our Architecture
Our Datasets
Results and Deployment
What's Next
We're hiring

Woven Planet

Building an automated driving system to increase access to **safe and reliable transportation for people** everywhere.

2000+ people
3+ global offices

Story of Level 5 and Woven Planet

Level 5

woven planet

2017 — 2018 — 2020 — 2021

2017: **Level 5** launched Autonomous Driving division at **Lyft**

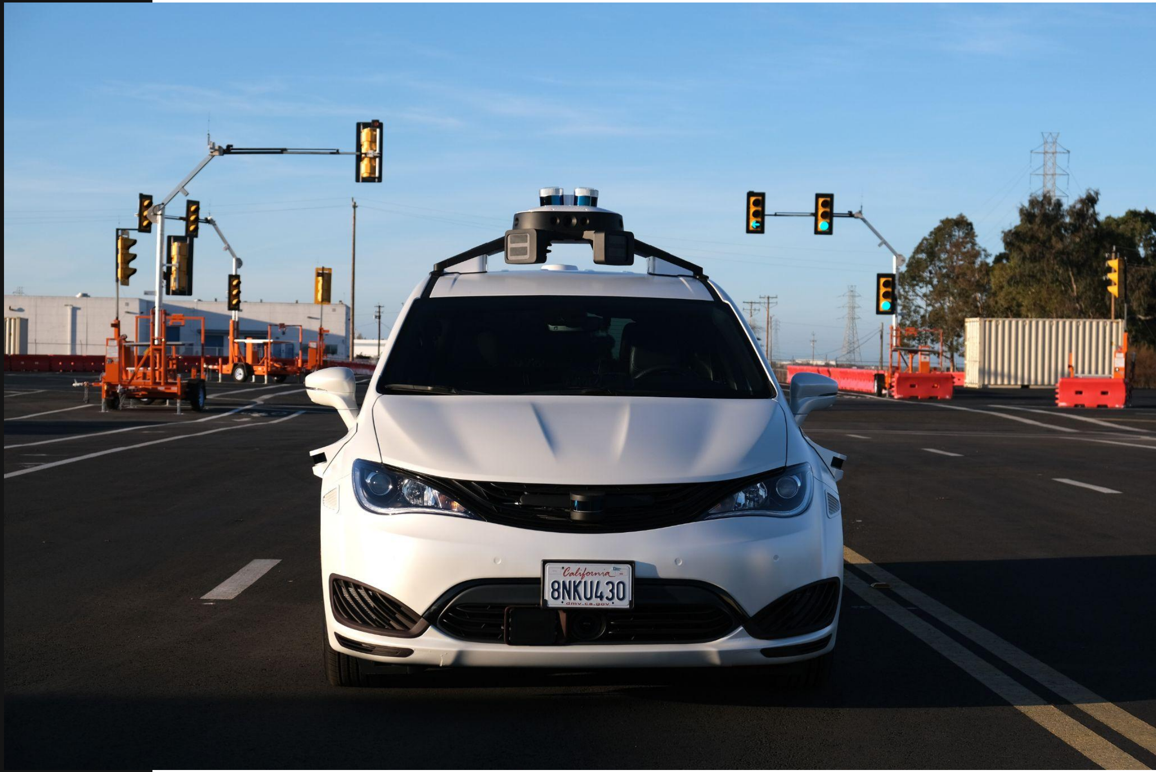
2018: Began testing on **public road**

2020: Began testing **fourth generation vehicle**

Deployed Data - Driven Planner in San Francisco

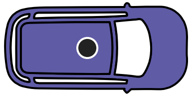
2021: Acquired by **Woven Planet**, a subsidiary of **Toyota Motor Corporation**

Introduction



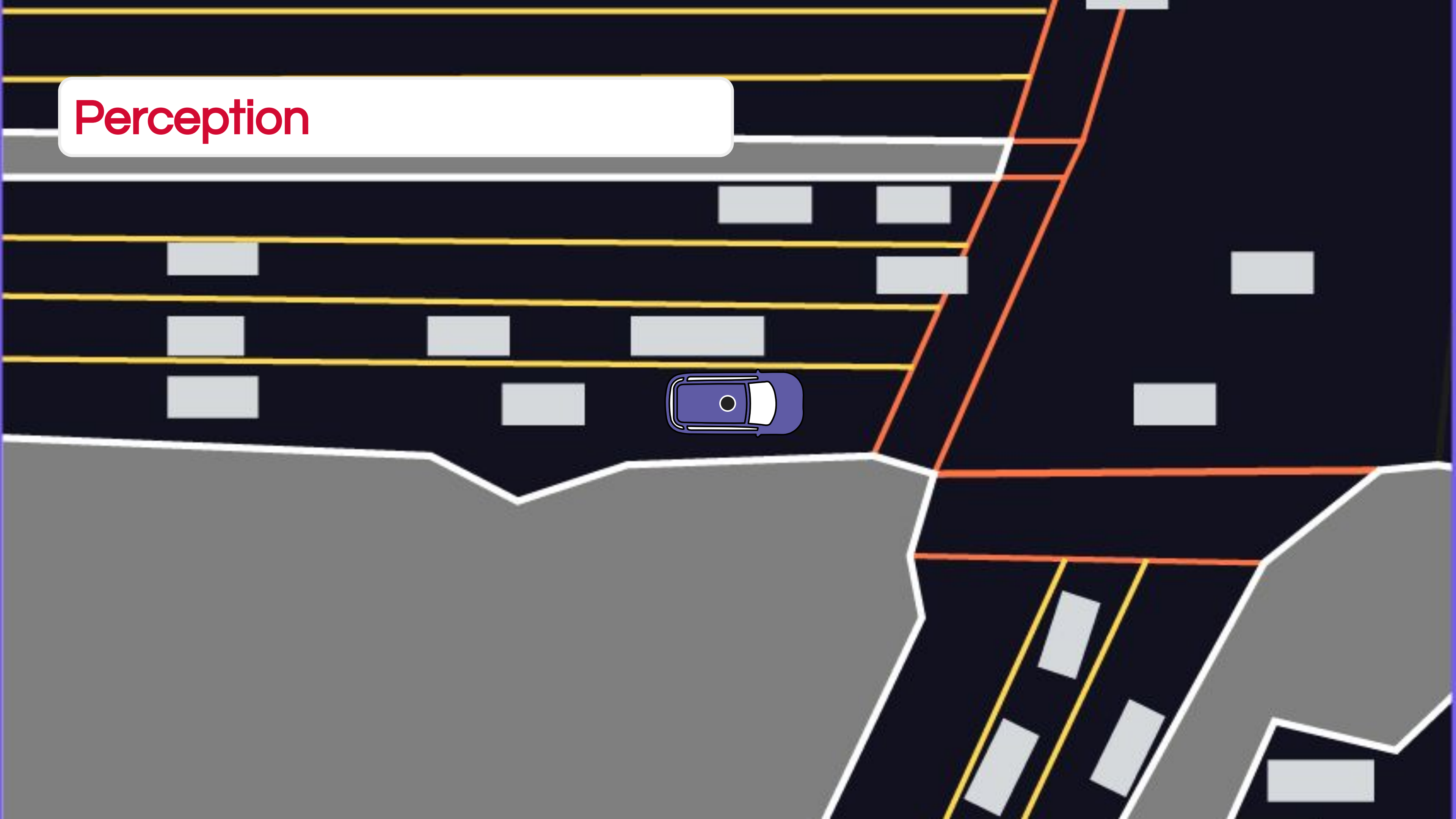
woven planet

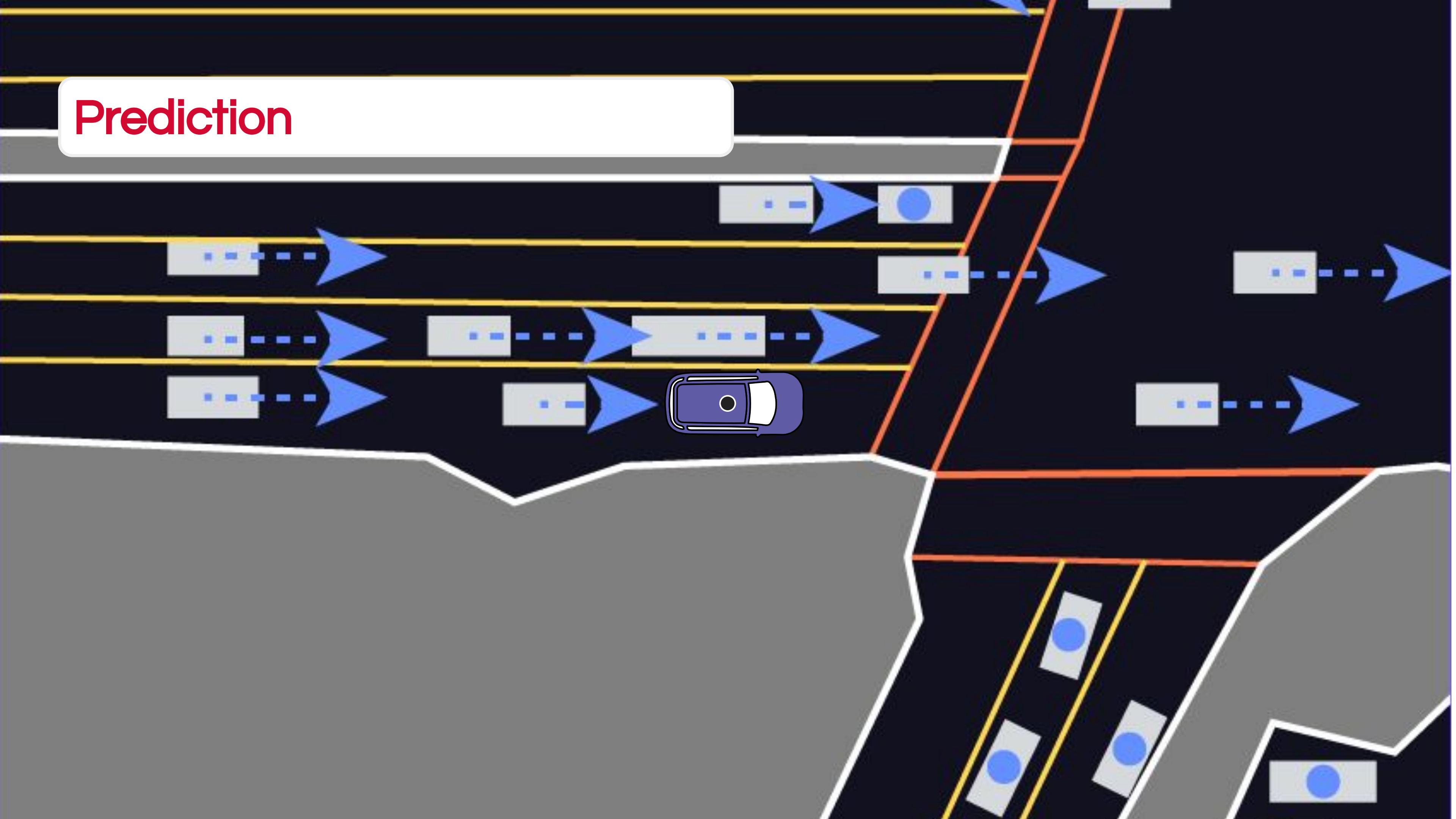
Traditional stack



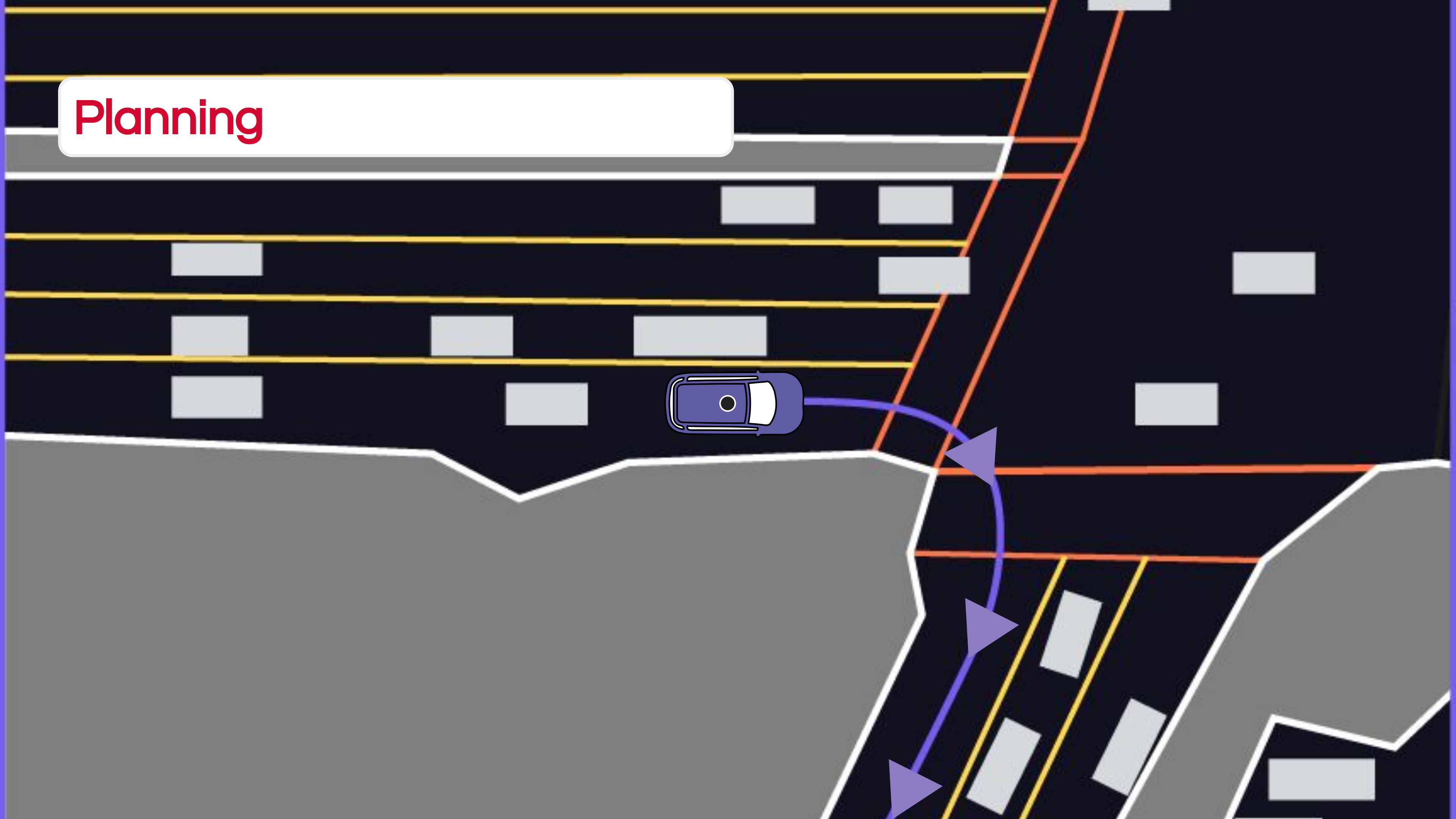
Sensors

Perception



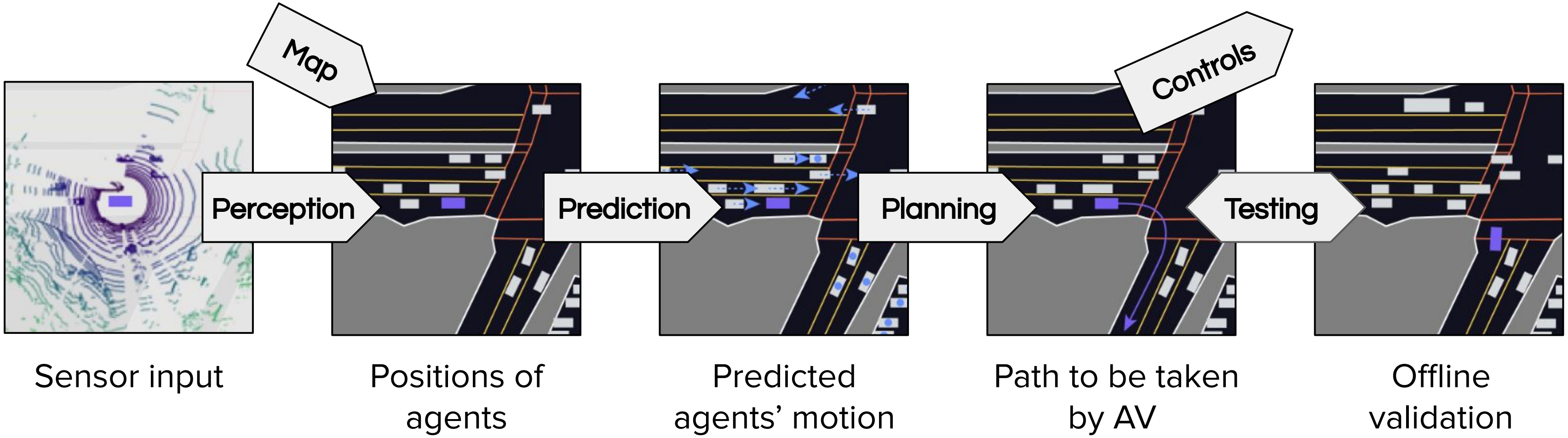

Prediction

Planning

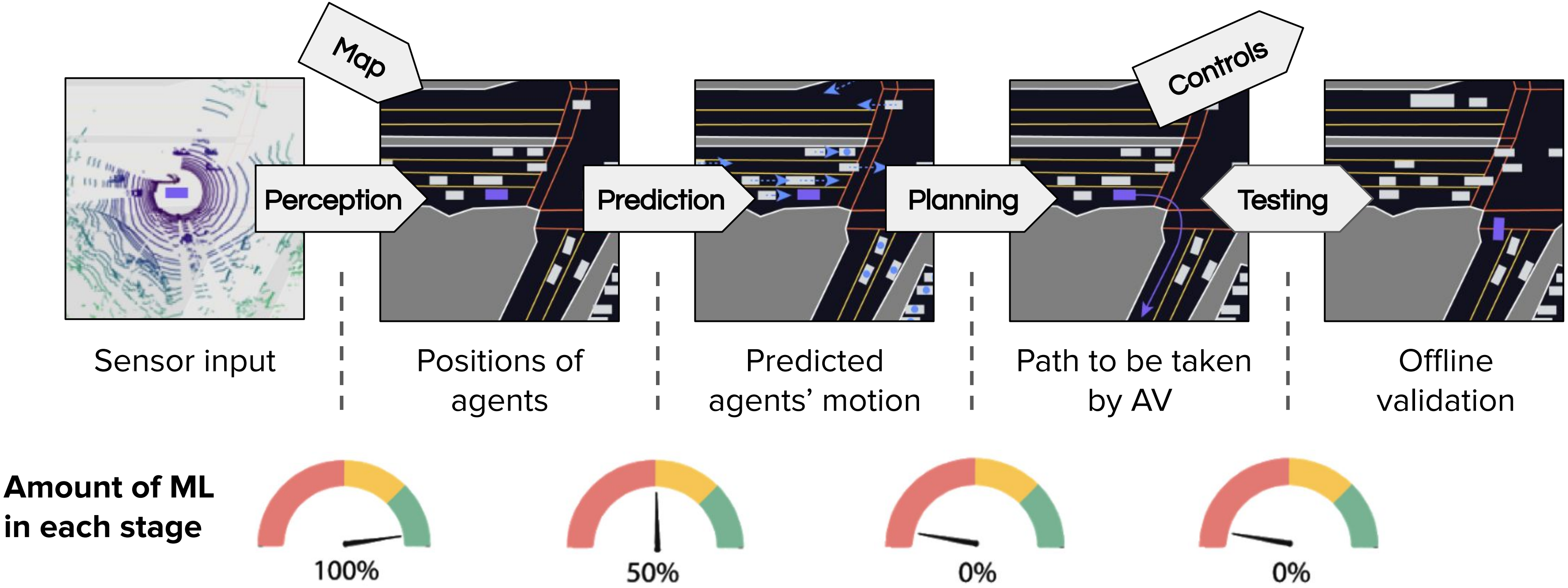


Control

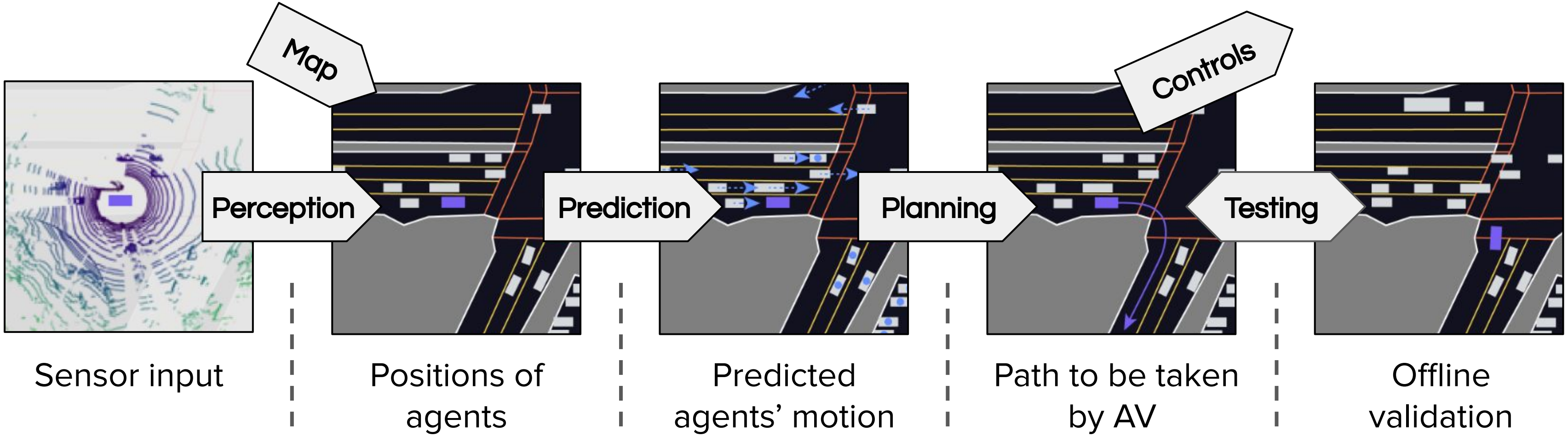
Traditional stack



Traditional stack

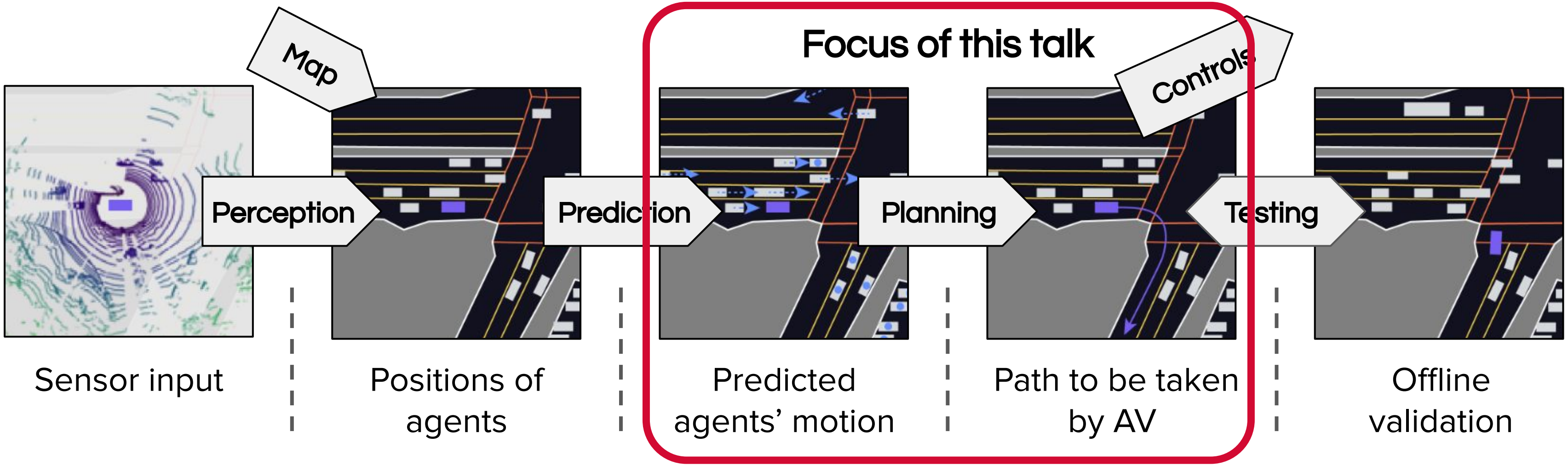


Towards data-driven decisions



Amount of ML in each stage

Towards data-driven decisions

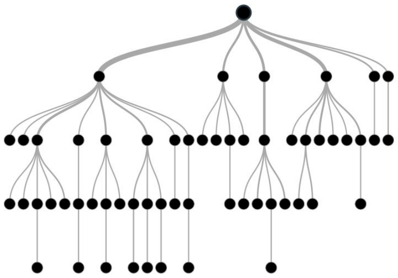


Amount of ML in each stage

Rule-based vs Data-Driven Systems

Rule-Based systems

- Hand-crafted by experts
- Difficult to scale
- More interpretable



Data-driven systems

- Scales with data
- Easy to adapt
- Less interpretable

Imitation Learning

woven planet

Decision Making

Sequential decision making is often formalized under the **MDP (Markov Decision Process)** framework.

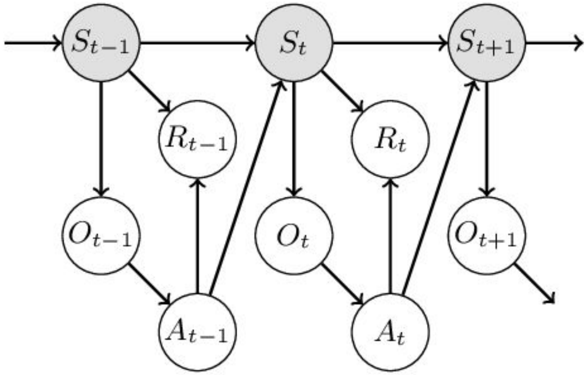
MDP is formalized as a 4-tuple **<S, A, P, C>**, where:

S = State space $\longrightarrow$ $s_t \in S$

A = Action space $\longrightarrow$ $a_t \in A$

P = Transition dynamics $\longrightarrow$ $P(S_{t+1} | s_t, a_t)$

C = Cost $\longrightarrow$ $c(s_t, a_t)$



Source: Learning Partially Observable Markov Decision Processes Using Coupled Canonical Polyadic Decomposition. Kejun Huang et al. June 2019.

Decision Making

The goal is to find a policy **π** that will map states to actions:

$$\pi : s_t \mapsto a_t$$

such that:

$$\underset{\pi}{\operatorname{argmin}}\, J(\pi) = \underset{s_t, a_t \sim \pi}{\mathbb{E}} \sum_{t=1}^{T} c(s_t, a_t)$$

Decision Making

The goal is to find a policy **π** that will map states to actions:

$$\pi : s_t \mapsto a_t$$

such that:

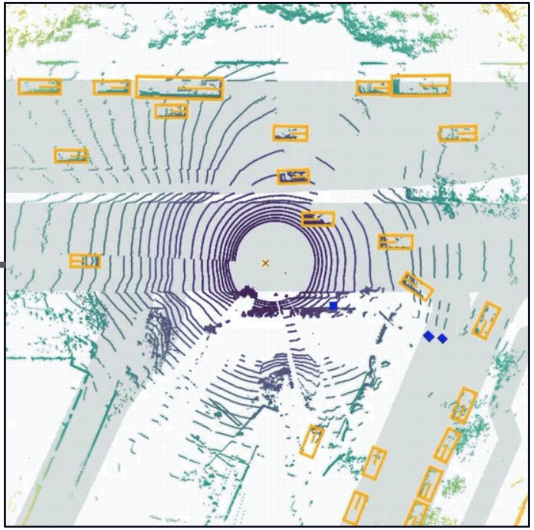
$$\underset{\pi}{\operatorname{argmin}}\, J(\pi) = \mathop{\mathbb{E}}_{s_t, a_t \sim \pi} \sum_{t=1}^{T} \boxed{c(s_t, a_t)}$$

We actually don't know this

Imitation Learning

We can get away from knowing the cost by imitating experts (humans or systems). Let's say we collect a dataset:

$$D = \{(s, a)\}$$



Then, we use traditional empirical risk minimization (**ERM**) to learn to imitate:

$$\pi' = \underset{\pi}{\operatorname{argmin}} \underset{s,a \sim D}{\mathbb{E}} \mathcal{L}(a, \pi(s))$$

* This approach (and its limitations) dates back to 1989 ALVINN paper !

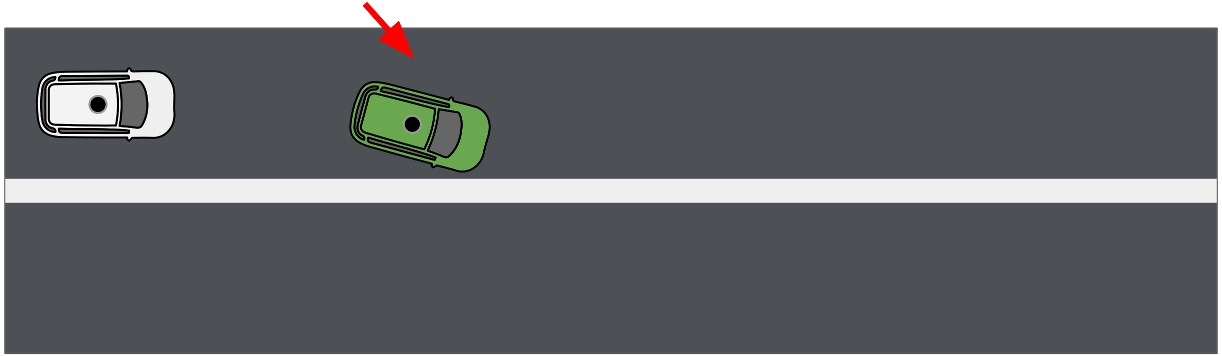
Imitation Learning

Imitation Learning

$P_\theta(a_t|s_{t-1})$

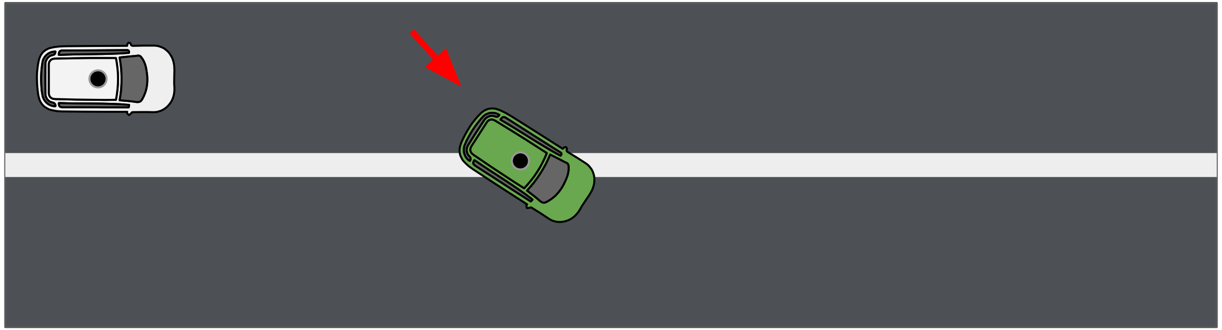
Imitation Learning

$P_\theta(a_t|s_{t-1})$

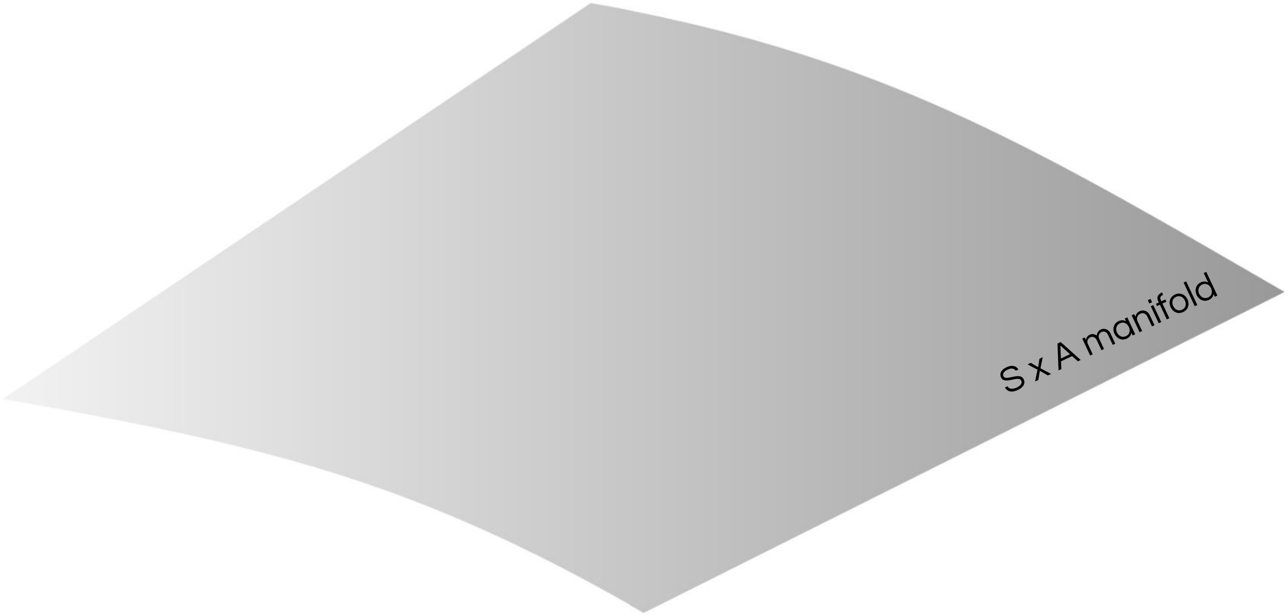


Imitation Learning

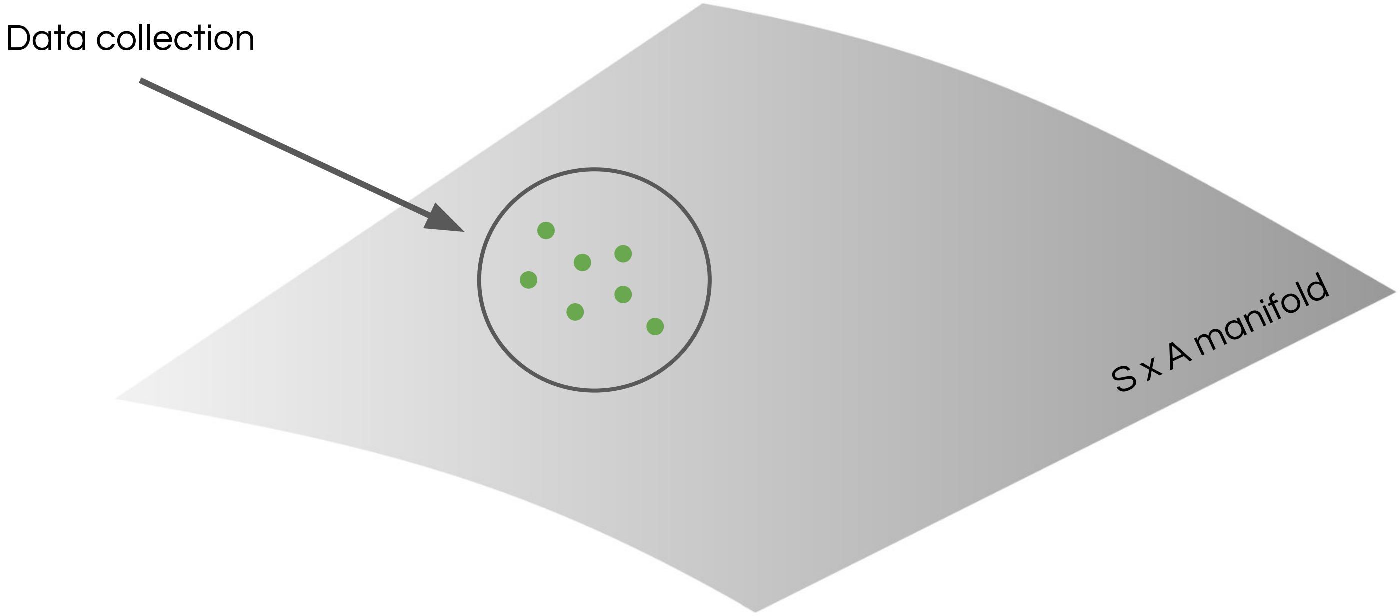
$P_\theta(a_t|s_{t-1})$



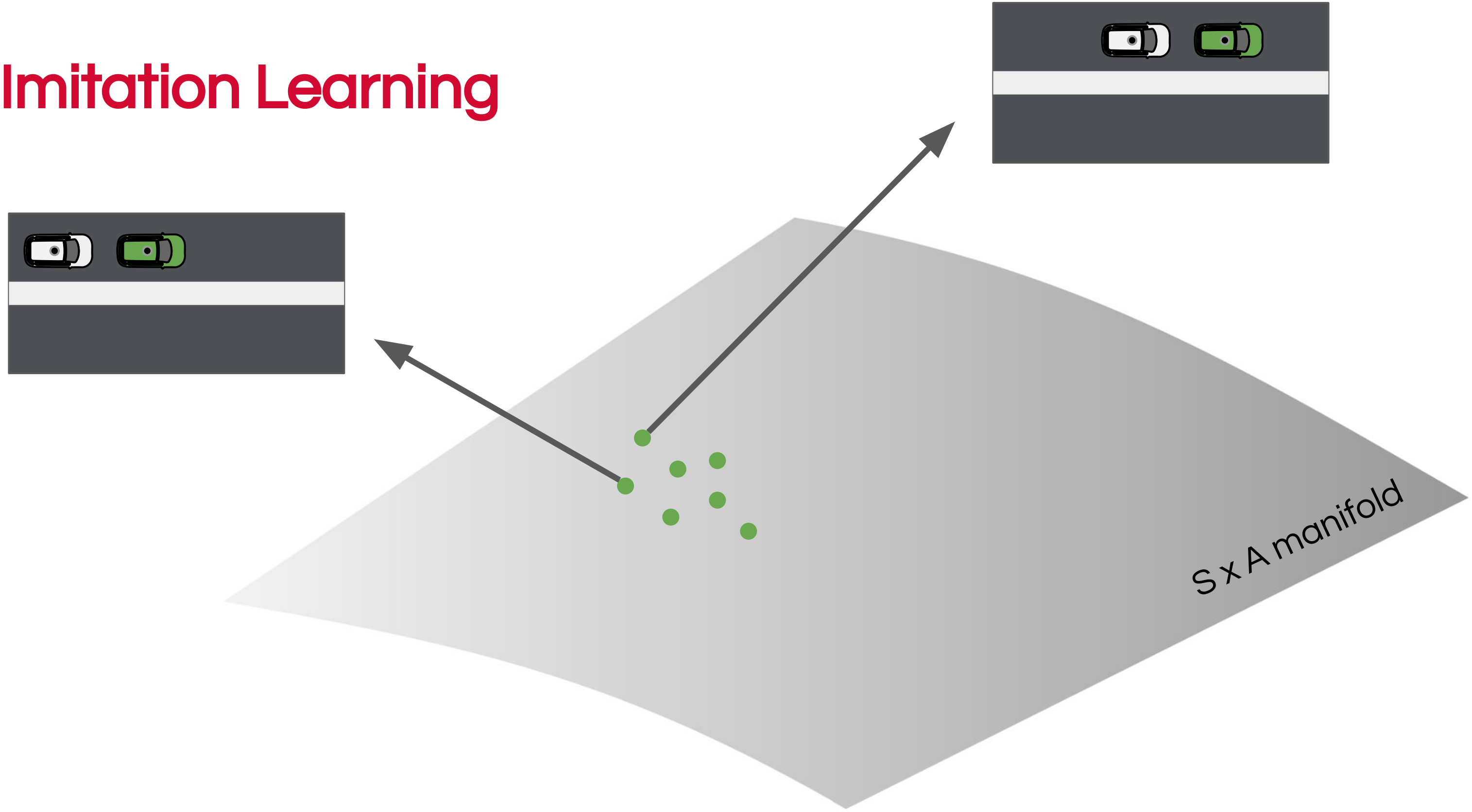
Imitation Learning



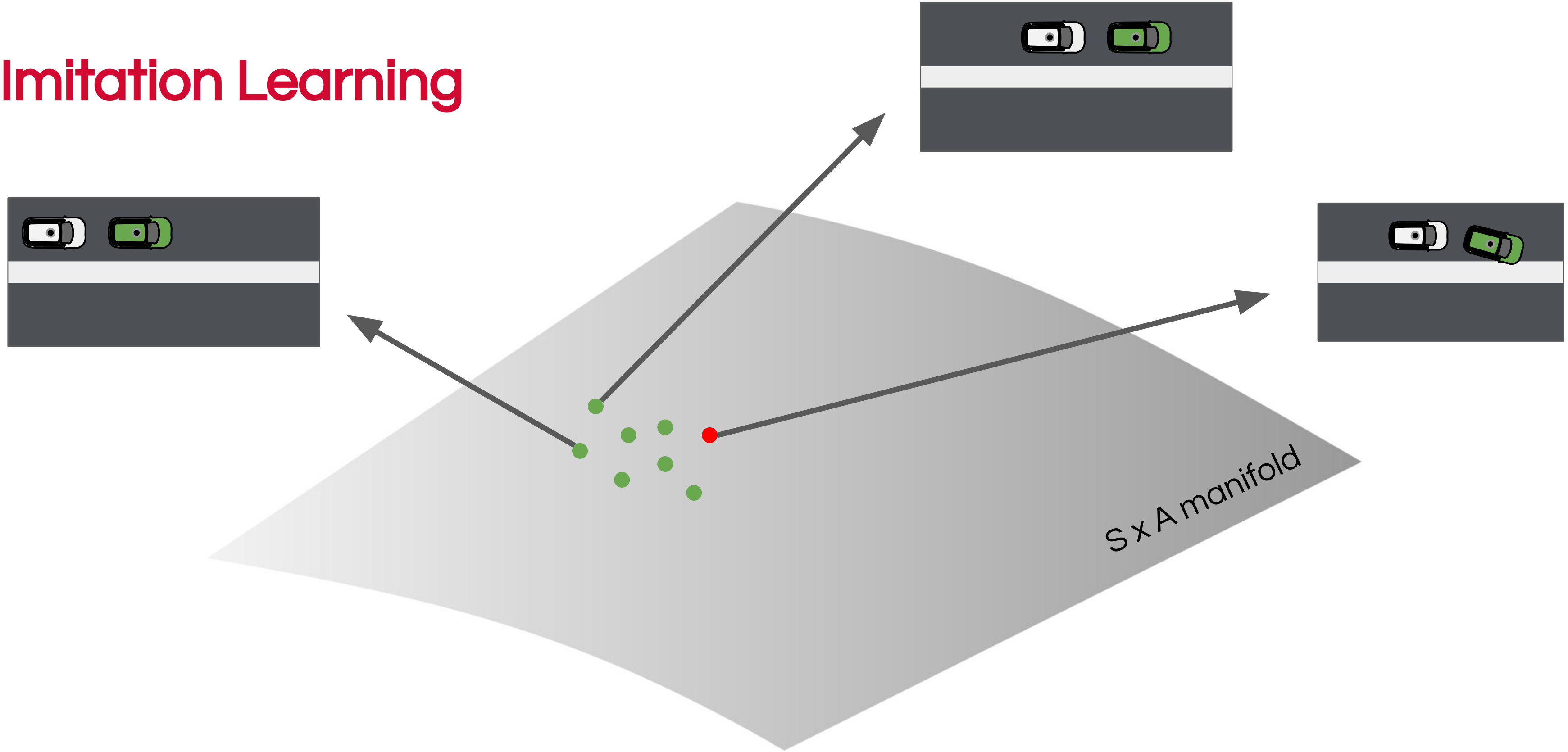
Imitation Learning



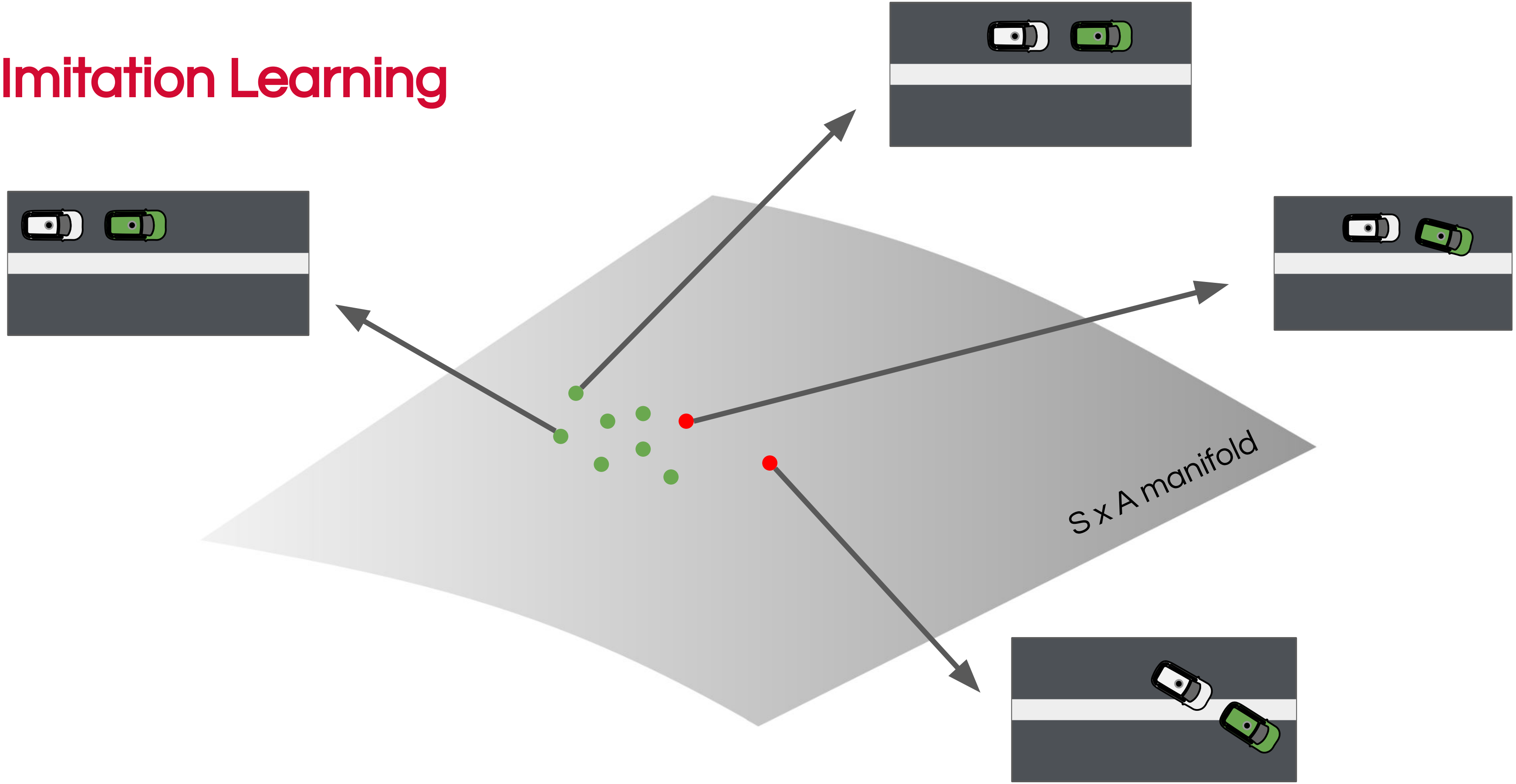
Imitation Learning



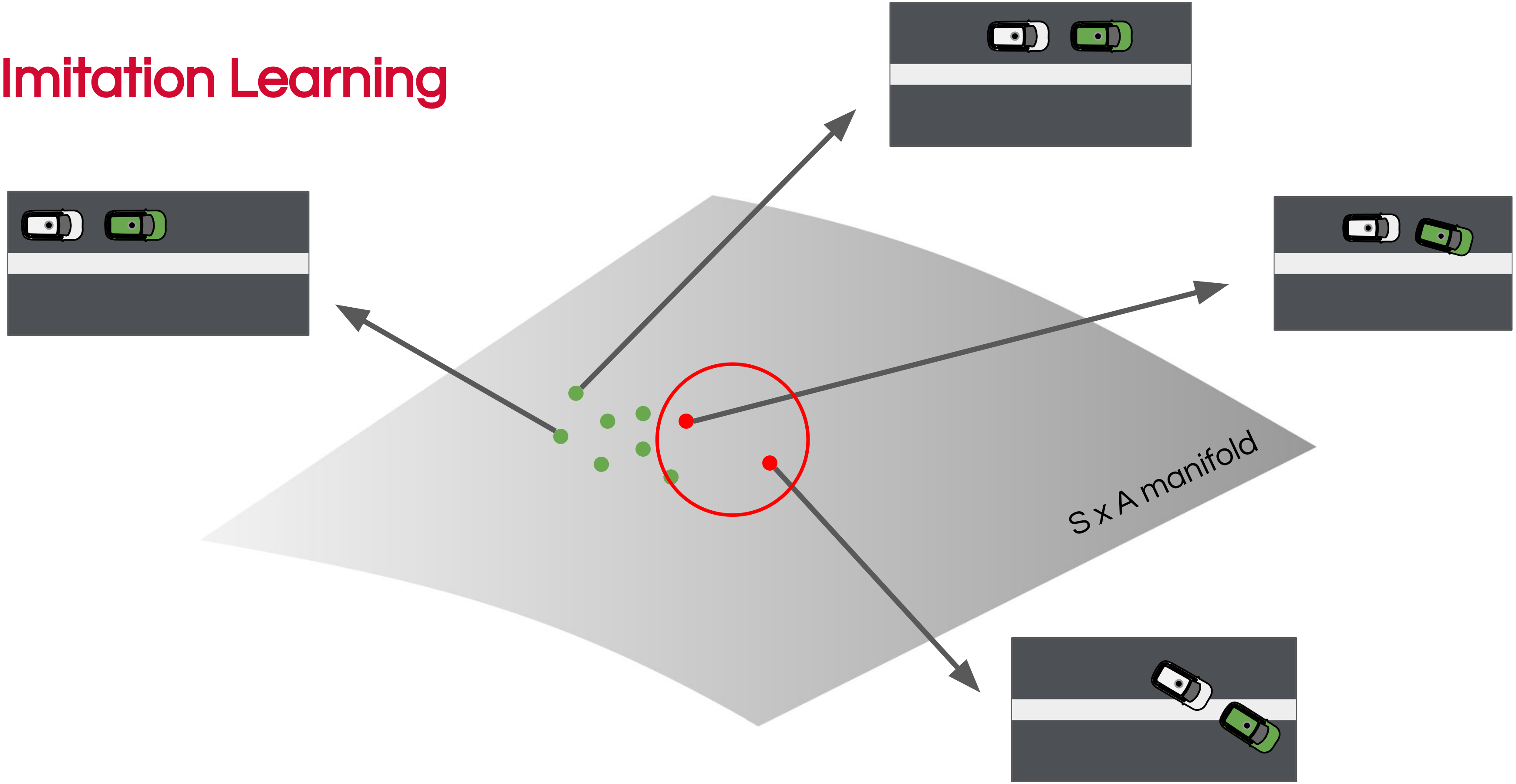
Imitation Learning



Imitation Learning



Imitation Learning



Imitation Learning

When you collect data: $D = \{(s_t, a_t)\} \mid \underbrace{s_t \sim d^t_{\pi^h}}_{\text{state distribution}}, \underbrace{a_t \sim \pi^h}_{\text{action distribution}}$

Mismatch

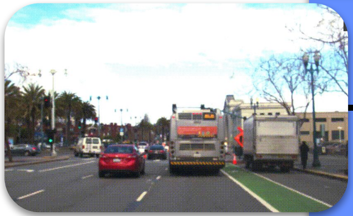
When you deploy: $s_t \sim d^t_{\pi_\theta}$ $a_t \sim \pi_\theta$

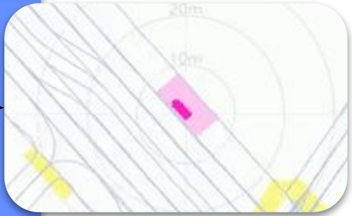
Data-Driven Approach for Planning

woven planet

Two Pillars to Data Driven

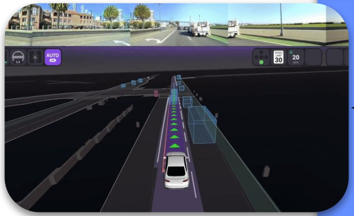
Autonomous driving

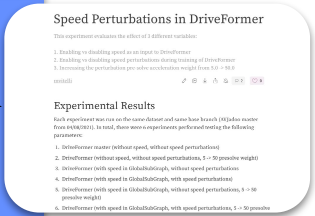




Massive harvesting of
real-world behavioural data

To feed data-hungry
data-driven systems




Speed Perturbations in DriveFormer

This experiment evaluates the effect of 3 different variables:

1. Enabling vs disabling speed as an input to DriveFormer
2. Enabling vs disabling speed perturbations during training of DriveFormer
3. Increasing the perturbation pre-solve acceleration weight from 5.0 -> 50.0

mvitelli

2 0

Experimental Results

Each experiment was run on the same dataset and same base branch (AVJadoo master from 04/08/2021). In total, there were 6 experiments performed testing the following parameters:

1. DriveFormer master (without speed, without speed perturbations)
2. DriveFormer (without speed, without speed perturbations, 5 -> 50 presolve weight)
3. DriveFormer (with speed in GlobalSubGraph, without speed perturbations
4. DriveFormer (with speed in GlobalSubGraph, with speed perturbations)
5. DriveFormer (with speed in GlobalSubGraph, without speed perturbations, 5 -> 50 presolve weight)
6. DriveFormer (with speed in GlobalSubGraph, with speed perturbations, 5 -> 50 presolve

A self-driving system
that improves with data

Data driven planner & simulation
replace rule-based systems

Tradeoff: data quantity vs perception accuracy

AV with **HD sensors**

Low quantity of data

High perception accuracy

Fleet with **commodity sensors**

High quantity of data

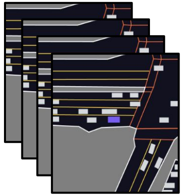
Low perception accuracy

Understanding this tradeoff is key to answering:

"How do we feed a data-first self-driving stack?"

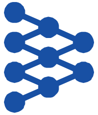
Can we transfer learnings between sensors?

Fleet with commodity sensors



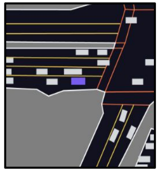
Large amounts of lower-accuracy sensor data

Use HD sensors at test time



Can we train on a mix of data despite the domain shift?

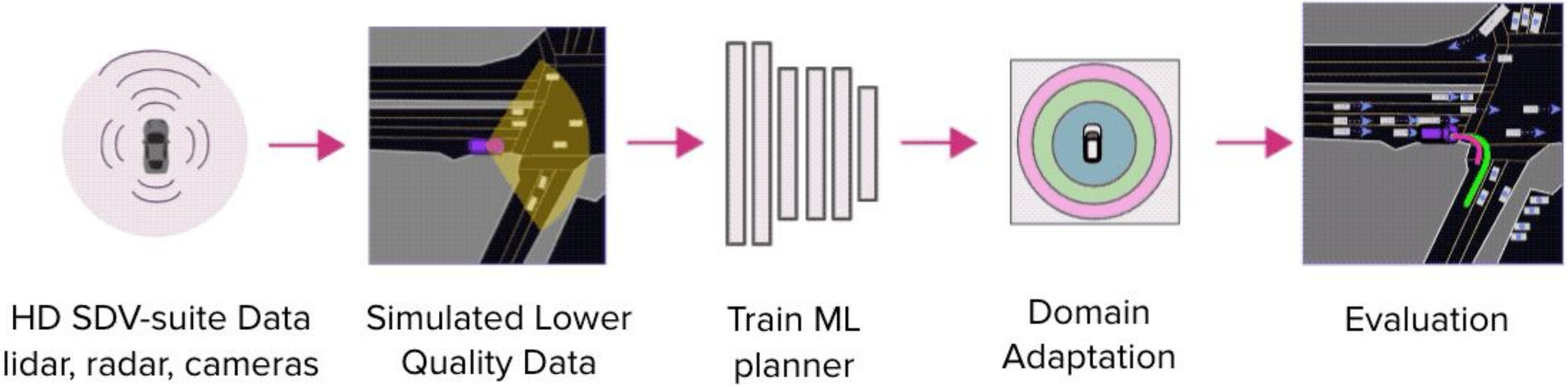
Traditional HD sensors

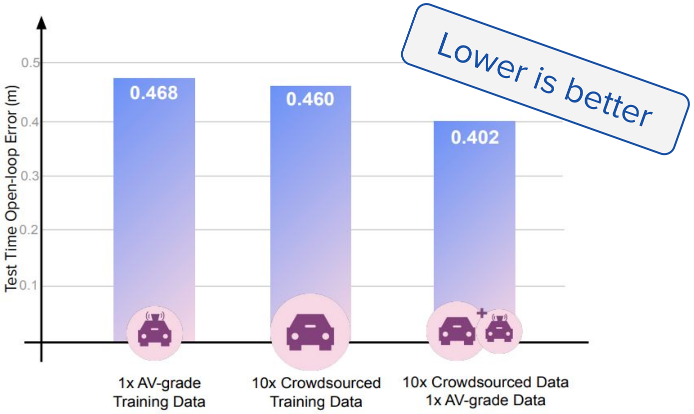


Small amounts of higher-accuracy sensor data

Chen et. al, *What data do we need for training an AV motion planner?*, ICRA 2021
Muller et al, *Driving Policy Transfer via Modularity and Abstraction*, CoRL 2018

Method and Results





- We can use larger amounts of lower quality data to train an AV[1]
- Team showed similar results on real fleet data using commodity sensors[2]

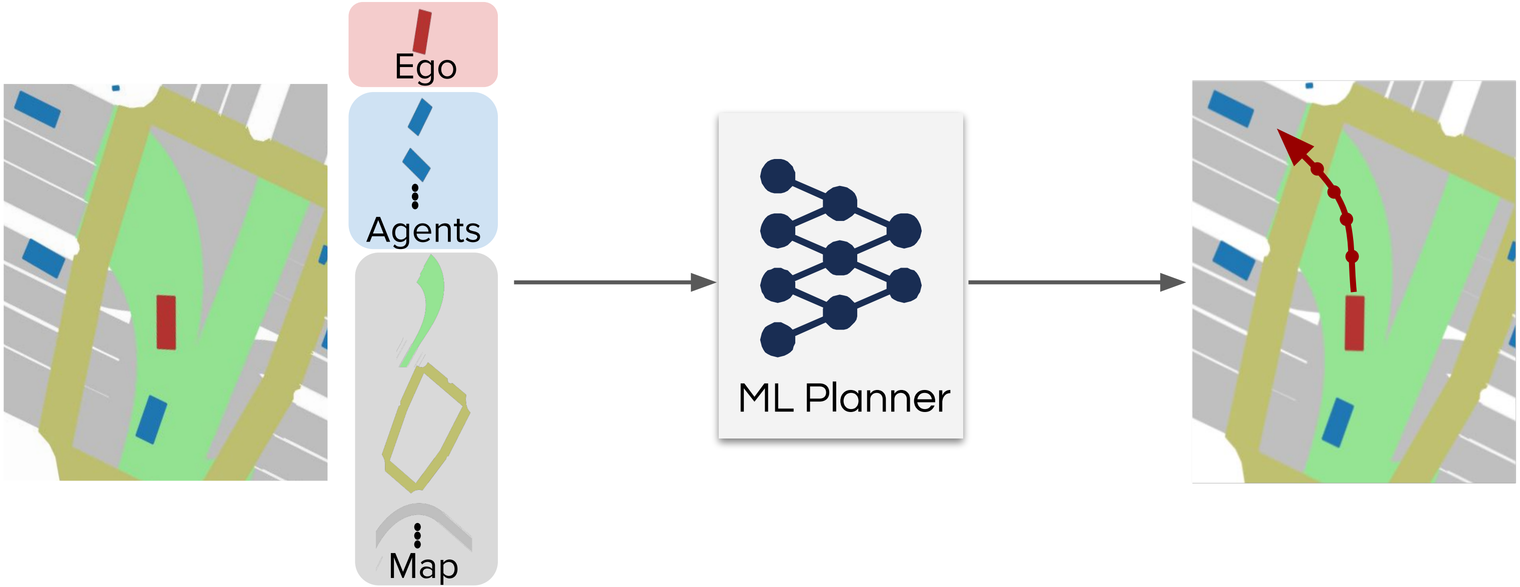
[1]Chen et. al, *What data do we need for training an AV motion planner?*, ICRA 2021

[2]Platinsky et al, *Quantity Over Quality: Training an AV Motion Planner with Large Scale Commodity Vision Data*, 2022

Our Architecture

woven planet

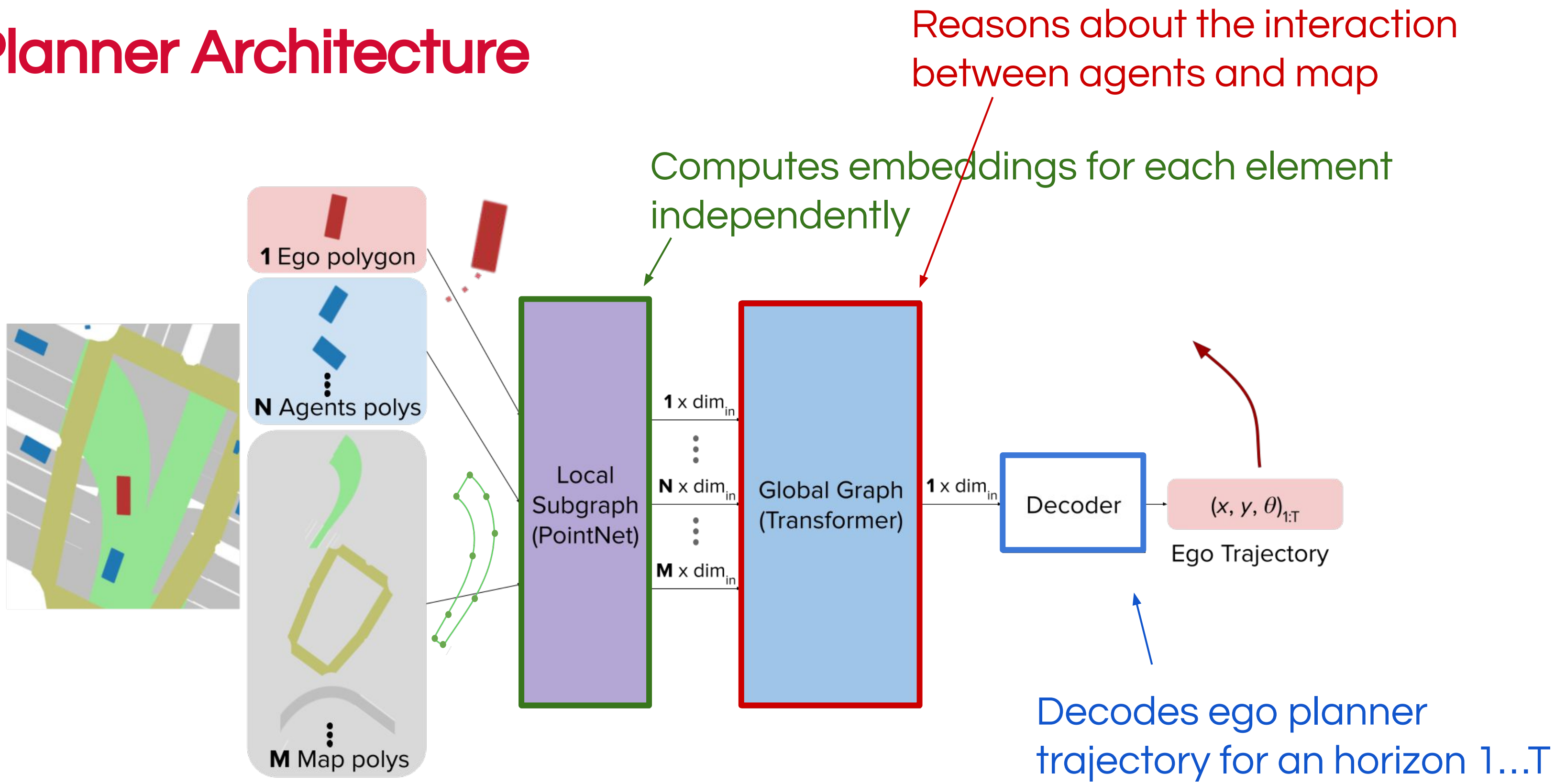
ML Planner



Inputs: localization, perception agents, traffic light status, & HD map

Output: ML planned trajectory

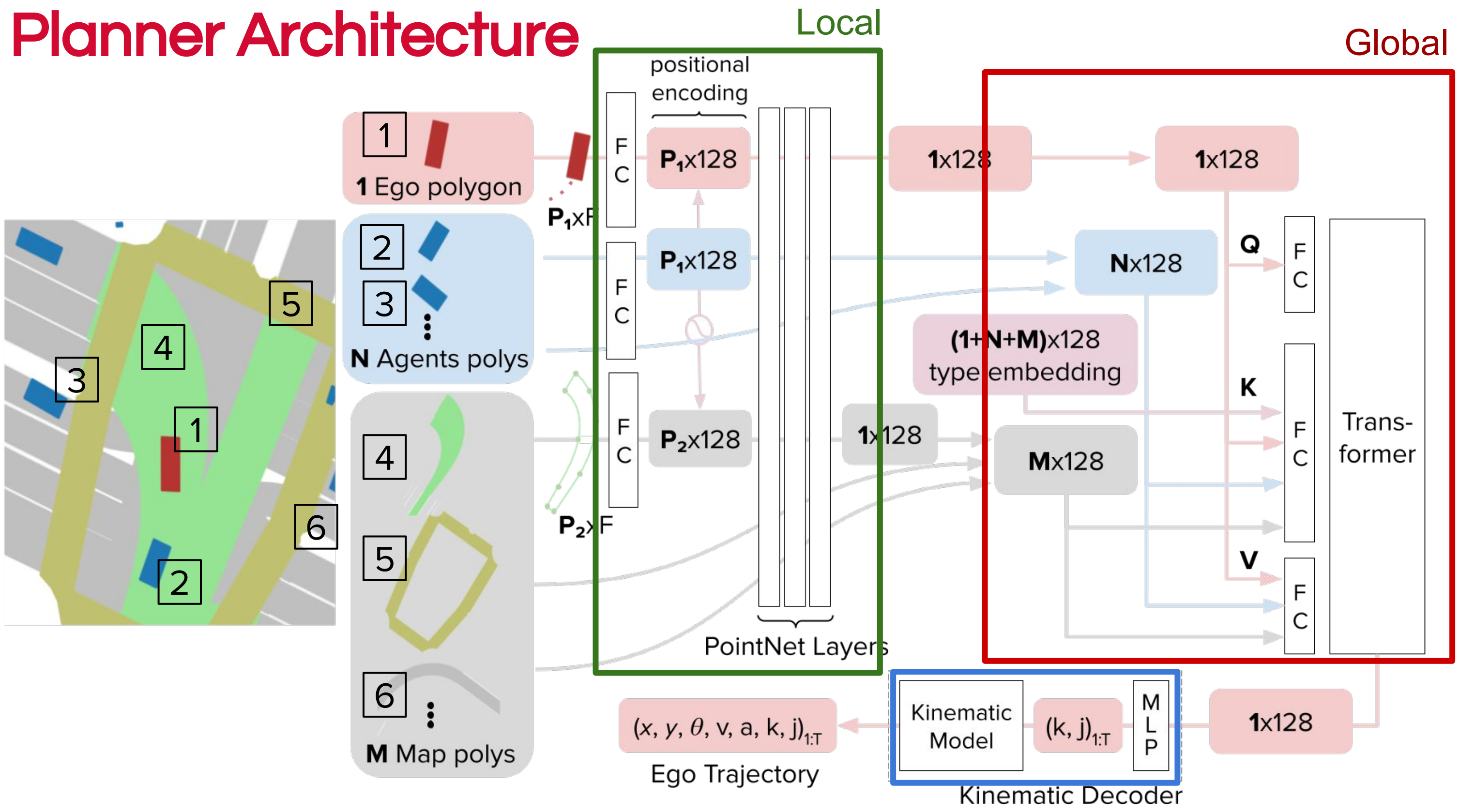
ML Planner Architecture



[1]PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation (Qi, 2017)

[2]Attention Is All You Need (Vaswani, 2017)

ML Planner Architecture



[1]SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies (Vitelli, 2021)

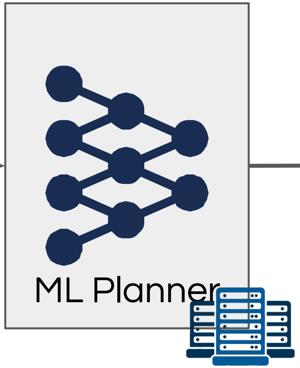
ML Planner Training

Behavior Cloning:
Minimize distance

ML
output

Logged
Trajectory

300 hours
driving logs



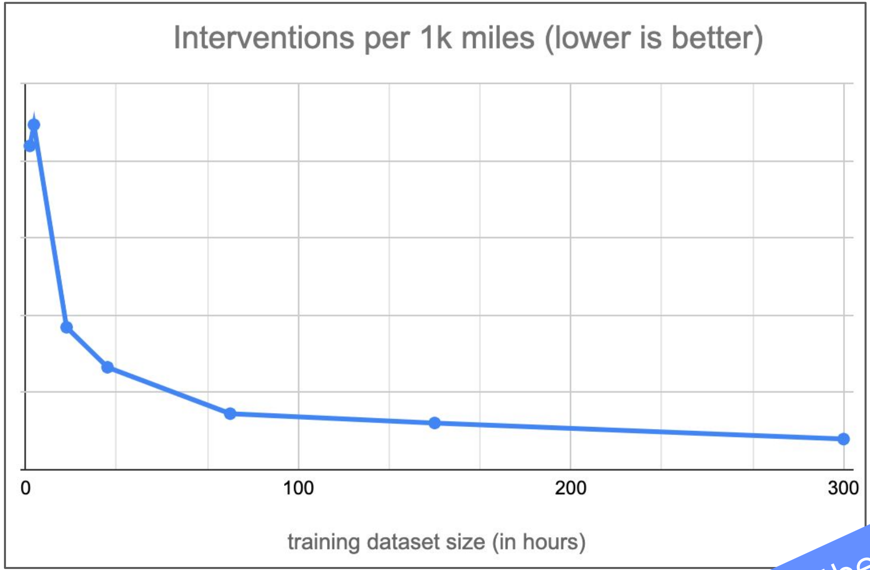
64 GPUs
~5hr cloud
training

$(x, y, \theta)_{1:T}$

Ego Trajectory

[1]SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies (Vitelli, 2021)

ML Planner Training

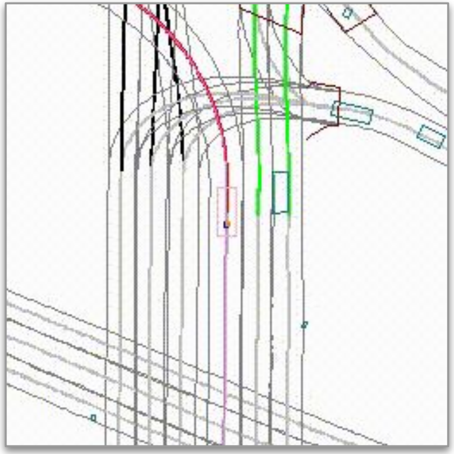


Key property: the ML planner improves with more data

[1]SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies (Vitelli, 2021)

Closed Loop Performance

Compounding Errors and Covariate Shift



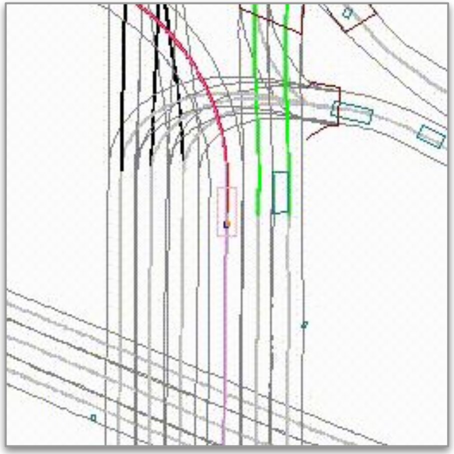
— Logged Ego
— ML planner

woven planet

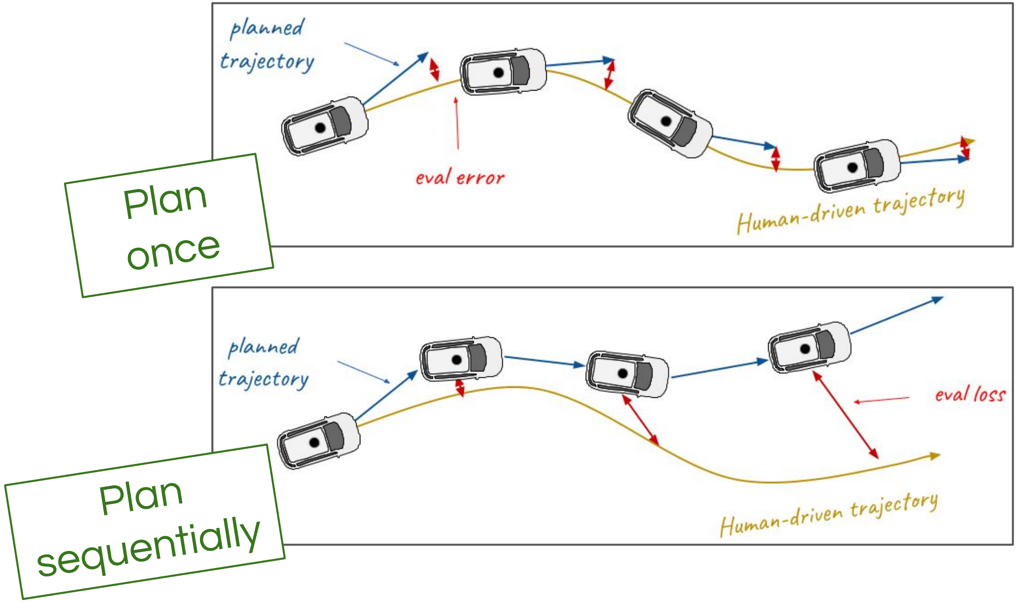
[1]ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst (Bansal, 2018)
[2]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Closed Loop Performance

Compounding Errors and Covariate Shift

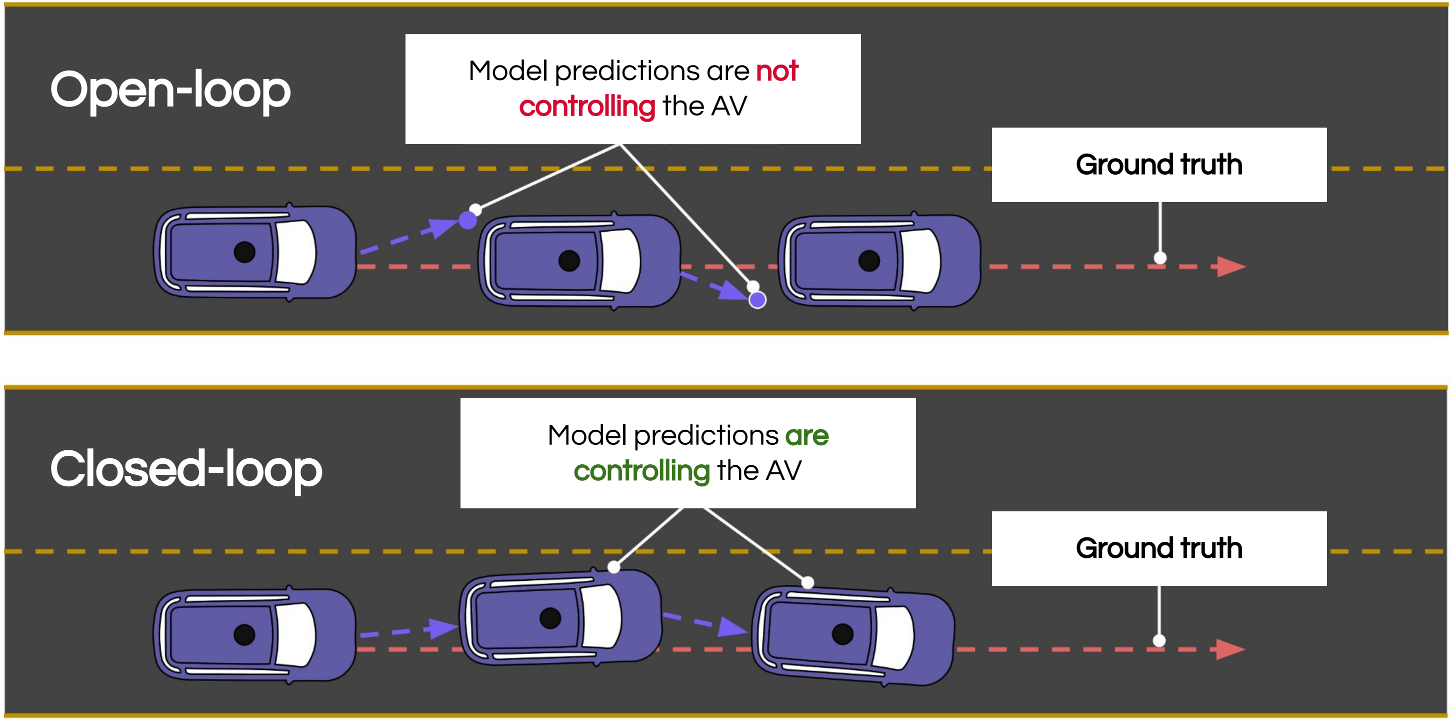


Logged Ego
ML planner



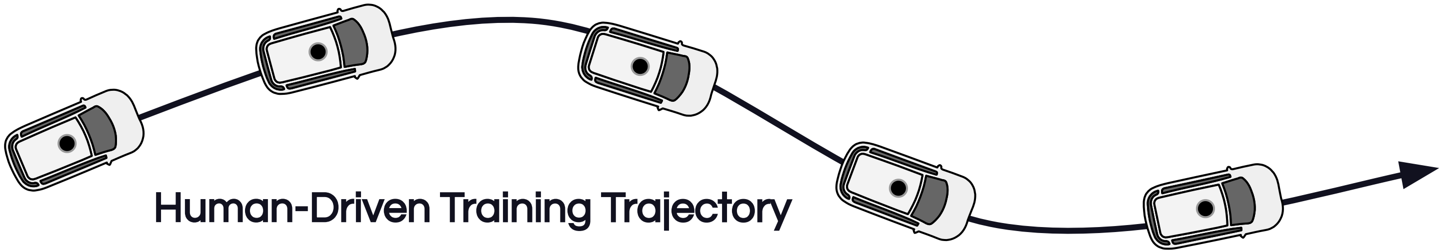
[1]ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst (Bansal, 2018)
[2]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Open-Loop vs Closed-Loop



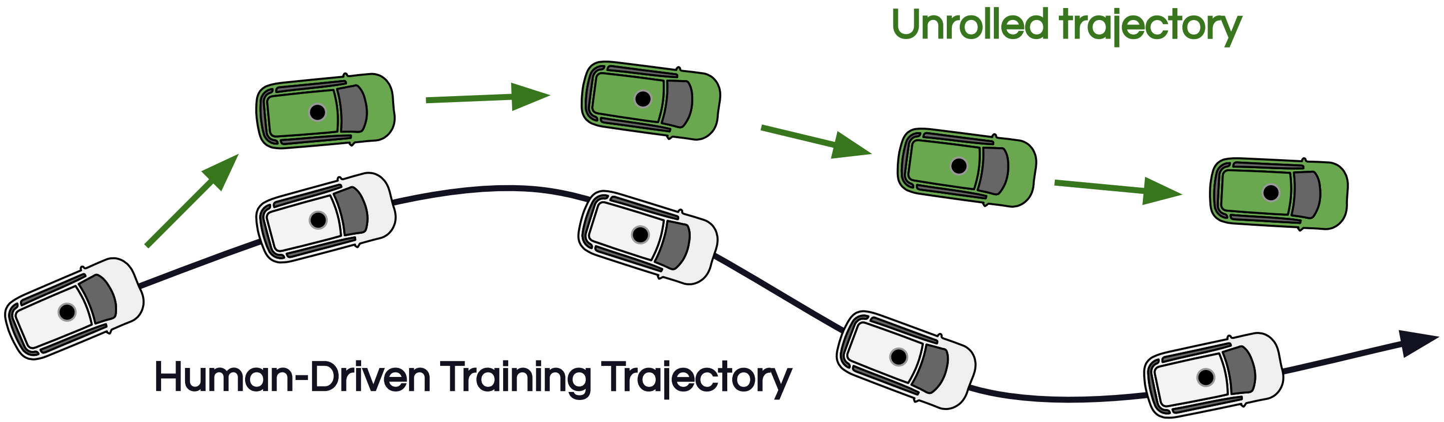
[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Closed Loop Training



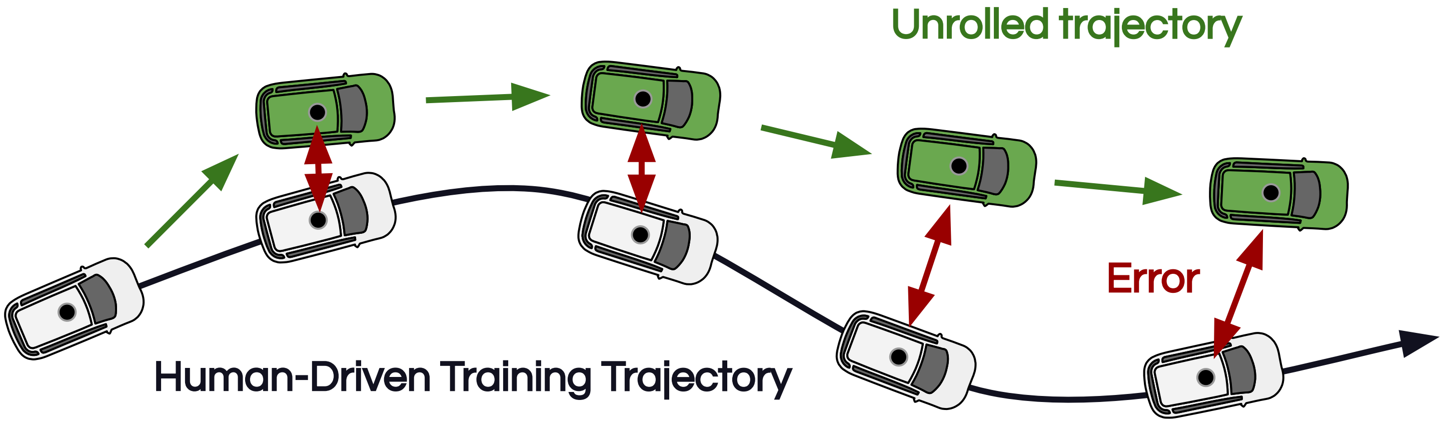
[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Closed Loop Training



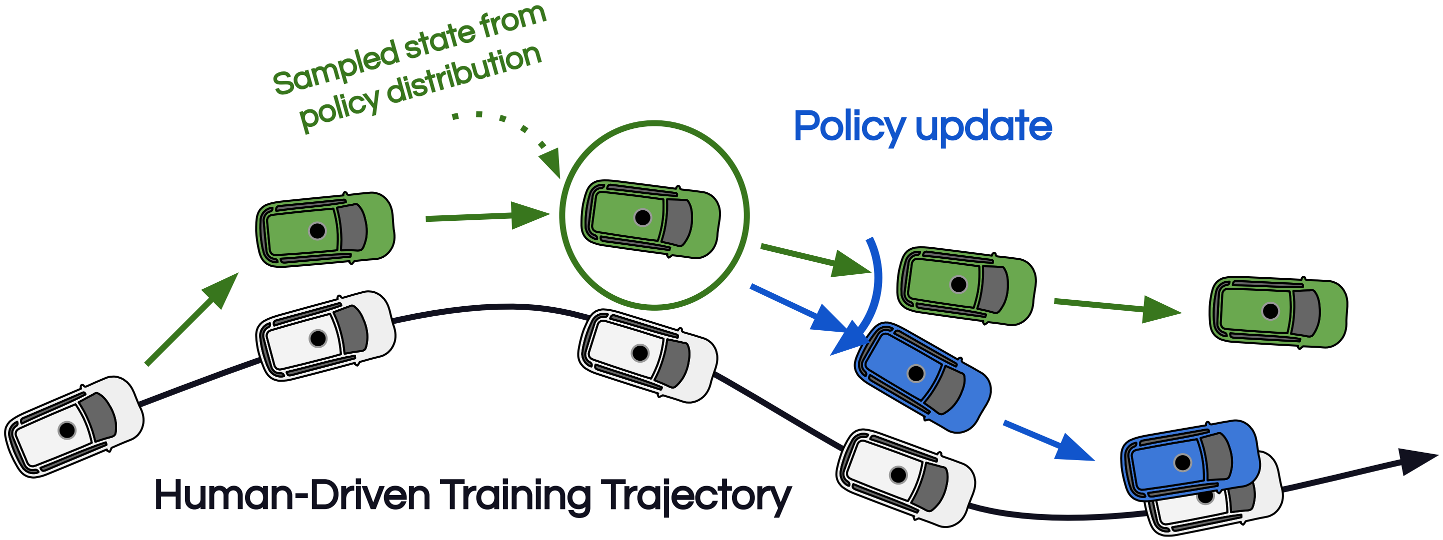
[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Closed Loop Training



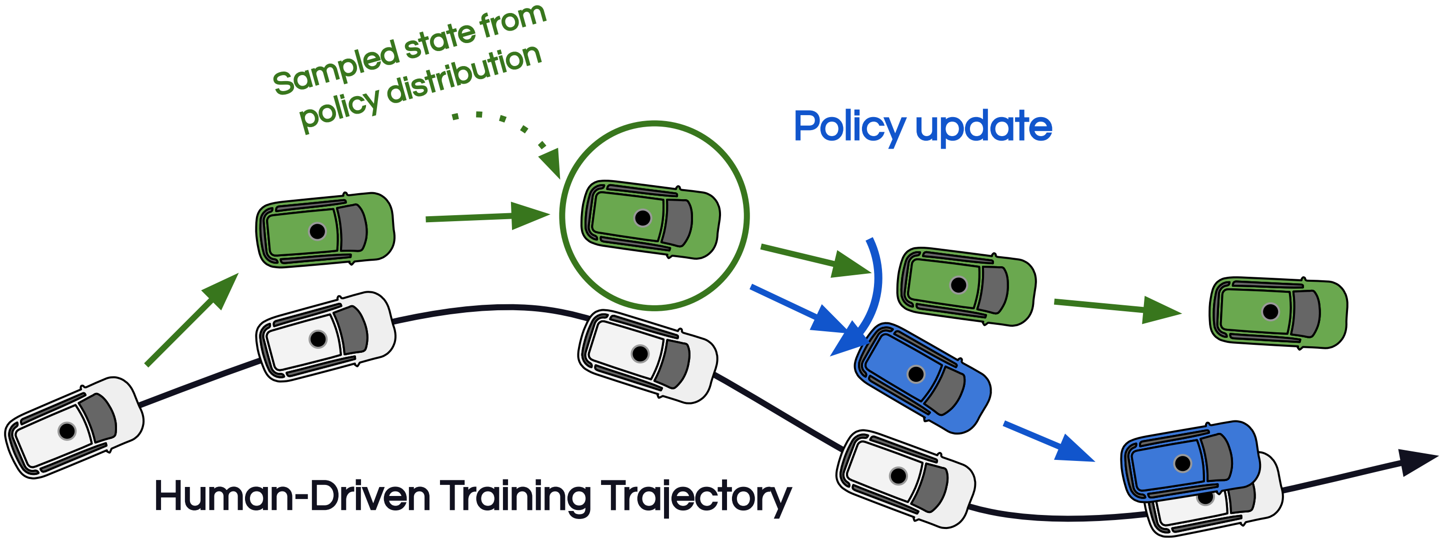
[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Closed Loop Training



[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

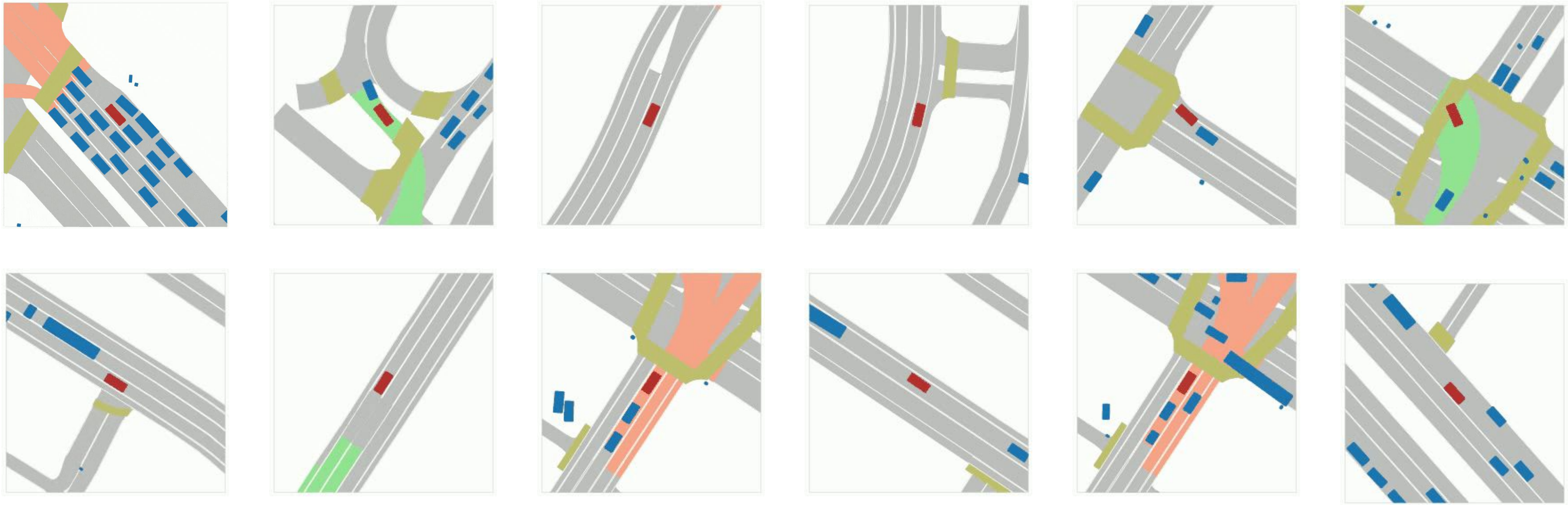
Closed Loop Training



- Unrolls bring the training closer to inference-time (less DBs and passiveness)
- Can backpropagate through it
- Slower to train and scale
- Agents are still logged / not really interacting or exploring / still imitation loss

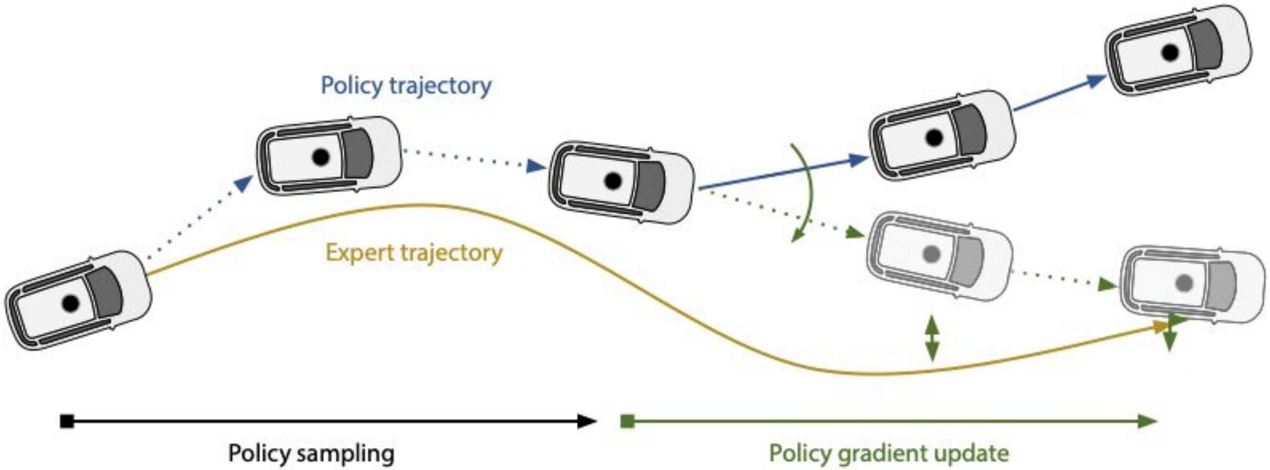
[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

Urban Driver: Qualitative Results



[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

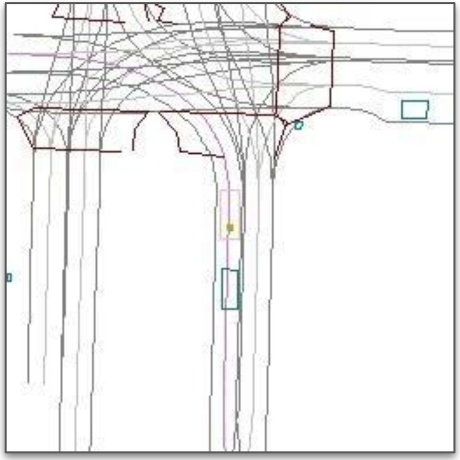
Closed Loop Training



[1]Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients (Scheel, 2021)

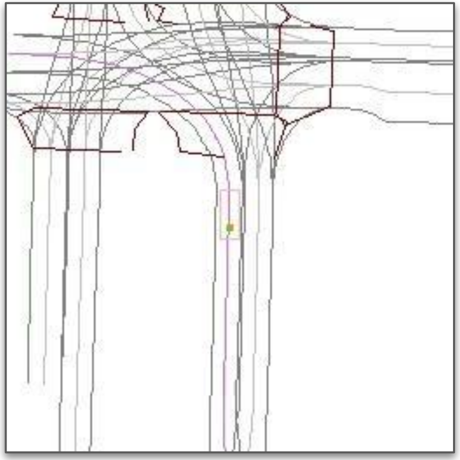
Causal stability

Example: at a red light, the ML Planner learned “**chase vehicle movement means go**” instead of “**green light means go**”



Agents from log

Same driving experience

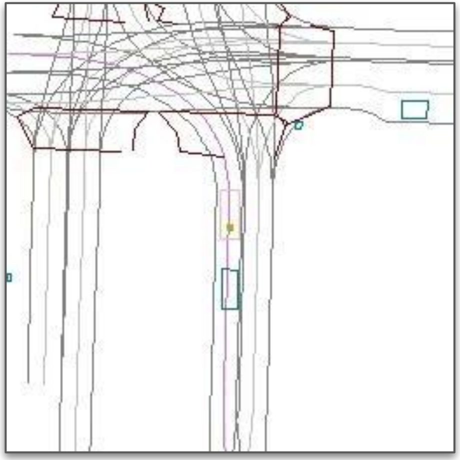


Agents removed
“Ghost town”

Logged Ego
ML planner

Causal stability

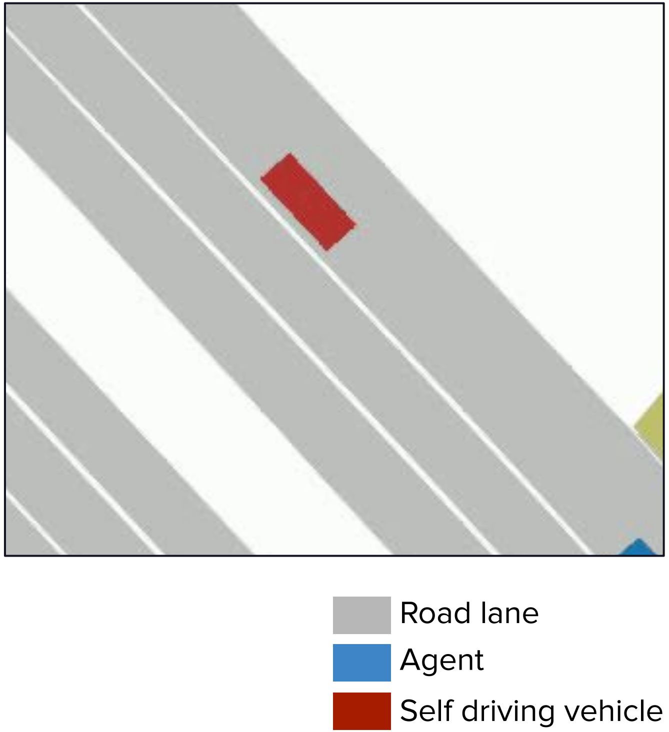
Example: at a red light, the ML Planner learned “**chase vehicle movement means go**” instead of “**green light means go**”



Agents from log

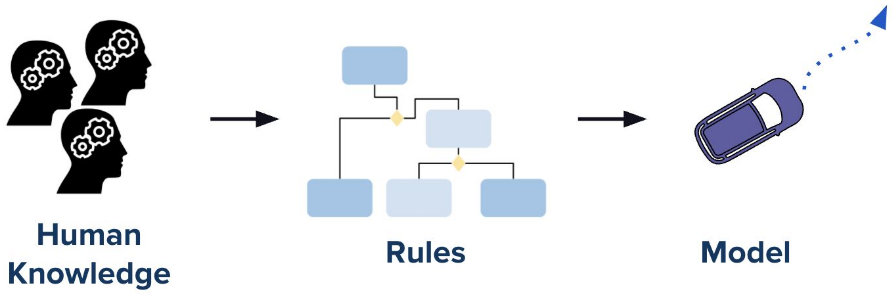
- Model is doing the **right thing**, but for the **wrong reasons,** important having means to **uncover it**
- **Causal instability** if naively trained, common problem in ML
- **Hint**: what would be the cause of these issues ? Would that happen if the model is trained with data that breaks this **spurious correlation** ?

Simulation

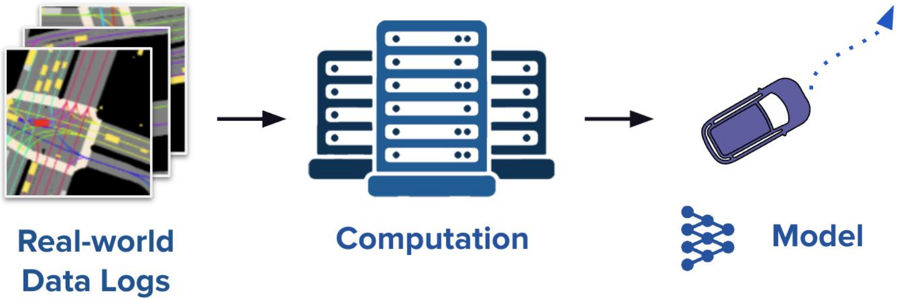


- Agent replay during evaluation is **non-reactive**
- To uncover diverse issues, you need **diverse scenarios**
- Simulating diverse scenes can **uncover failure modes and causal instability** of the policy

We can apply same principles to simulation



Rule based
Manually code the "rules of the road" for all the agents in the simulation

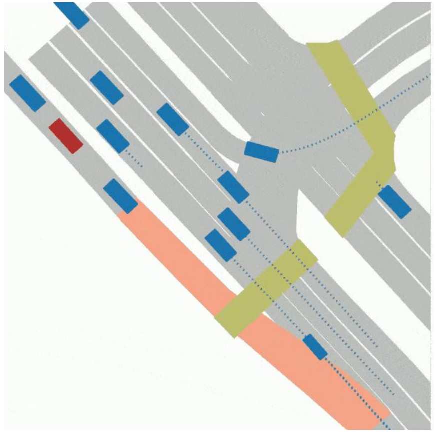


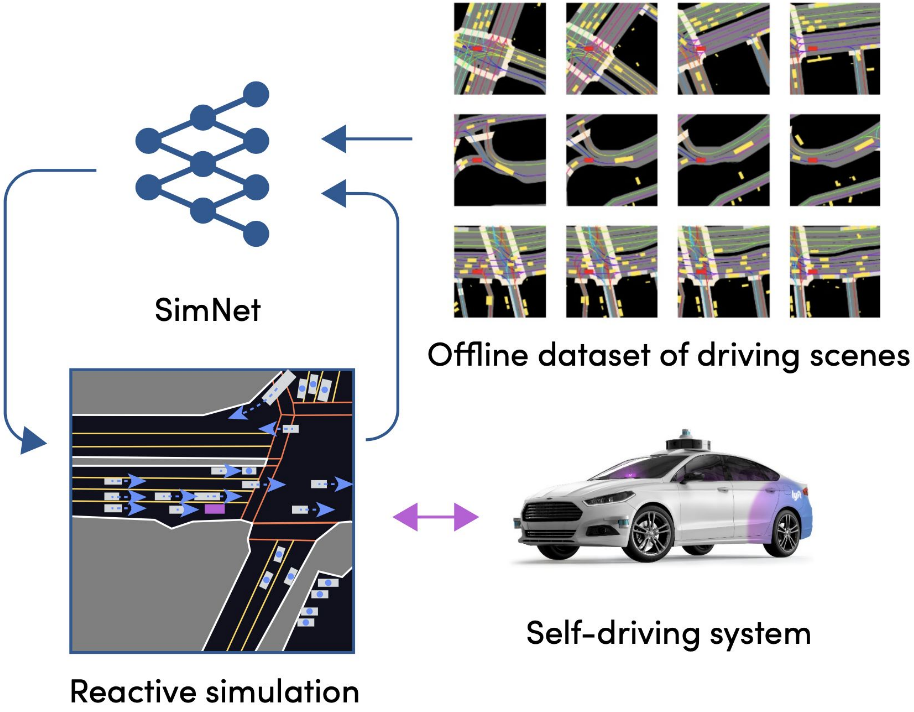
Data-first - ML based
Learn reactive simulation directly from data[1] without hand-crafted features

Data-driven simulation with SimNet

Key Features:

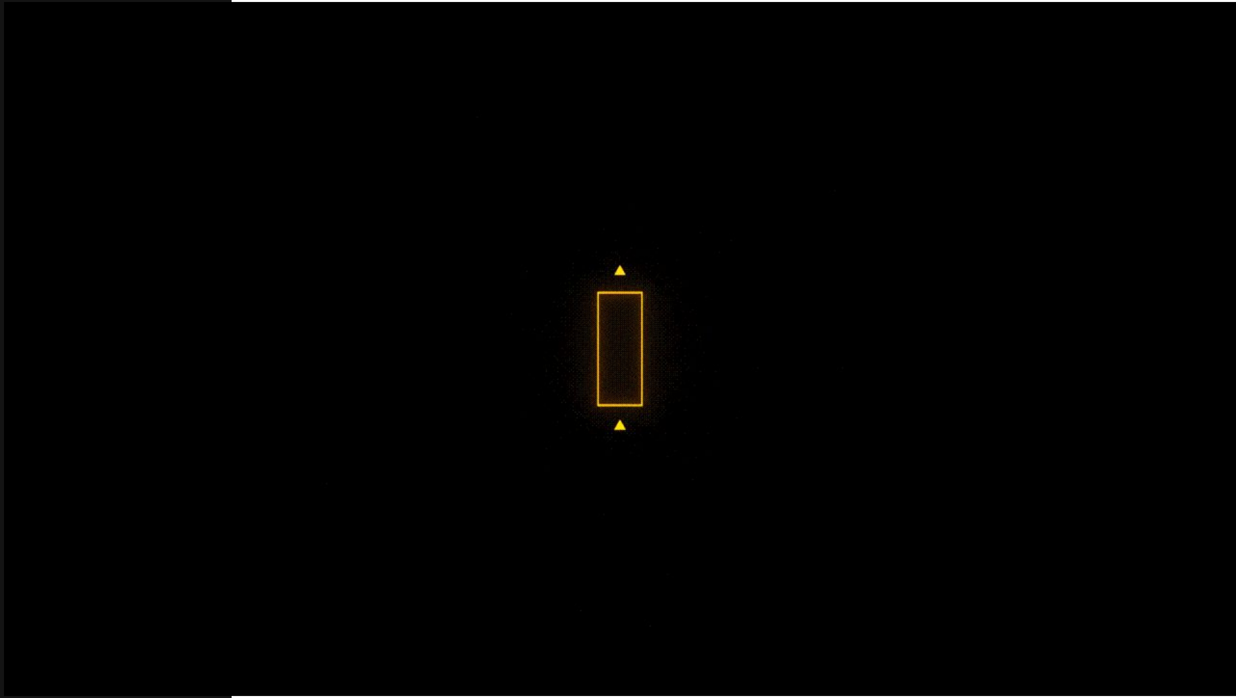
- Data-driven approach to simulation
- Reactive agents



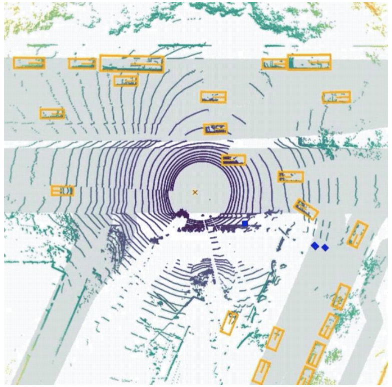


SimNet: Learning Reactive Self-driving Simulations from Real-world Observations, Bergamini et al.

Our Datasets

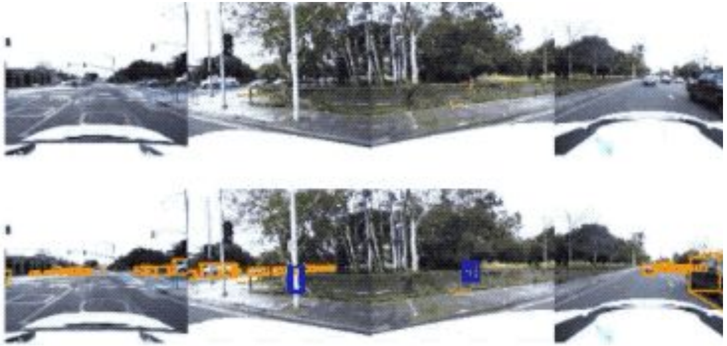


WP Level 5 Perception Dataset



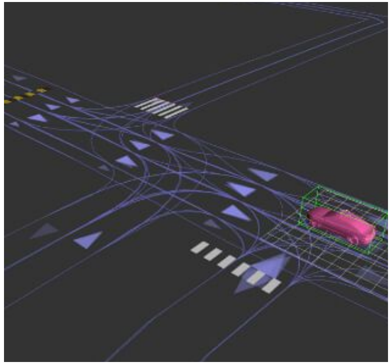
30k

Annotated LiDAR scans



1.3M

Annotated images



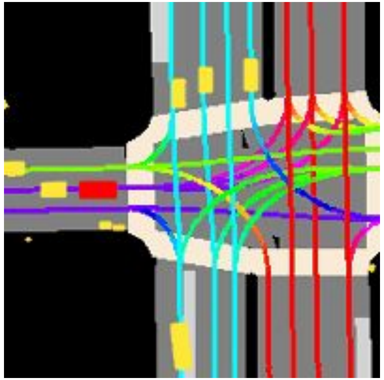
HD Map

Semantic map of the area

https://level-5.global/data

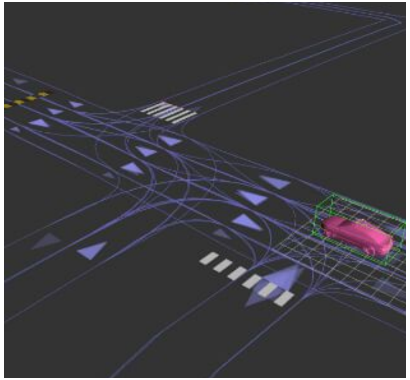
Kesten, Robert et al. "Lyft Level 5 Perception Dataset 2020.

WP Level 5 Prediction/Planning Dataset



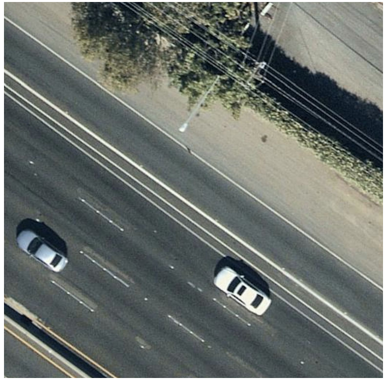
1,000+

Hours of traffic logs



15,000+

HD semantic map annotations



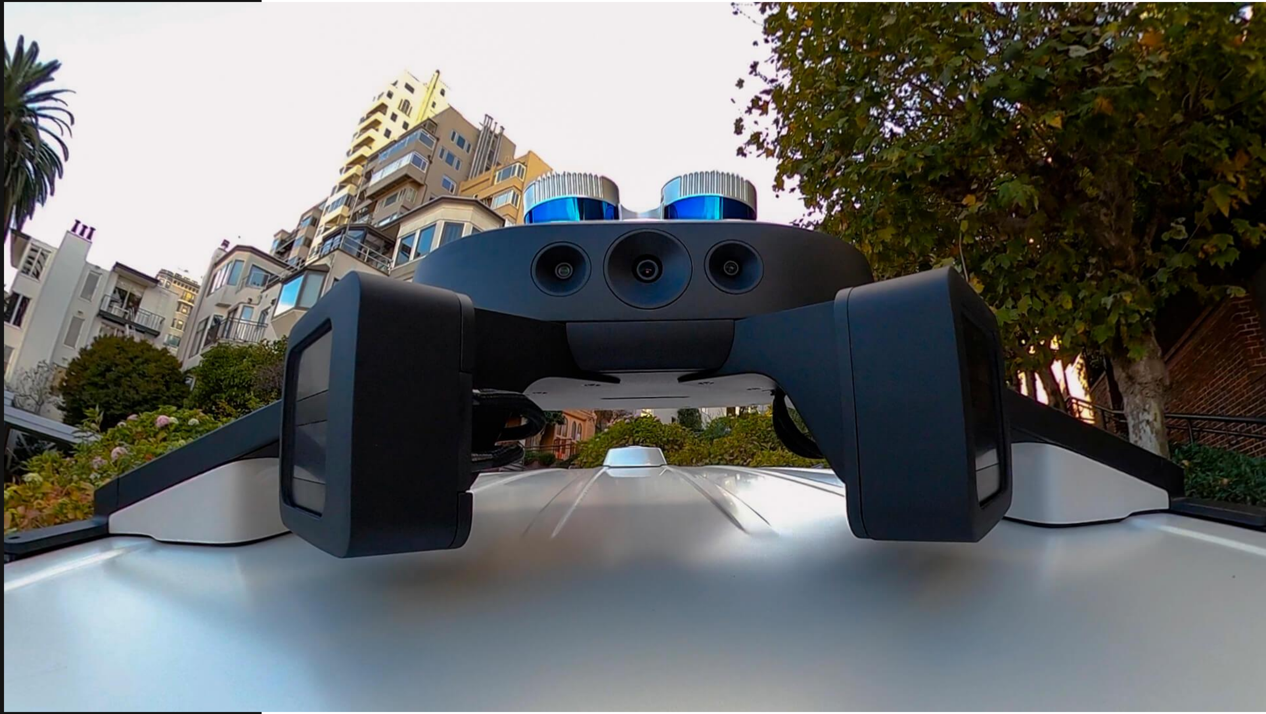
6 cm/pix

High resolution aerial map

https://level-5.global/data

Houston, John, et al. "One thousand and one hours: Self-driving motion prediction dataset." The Conference on Robot Learning (CoRL). 2020.

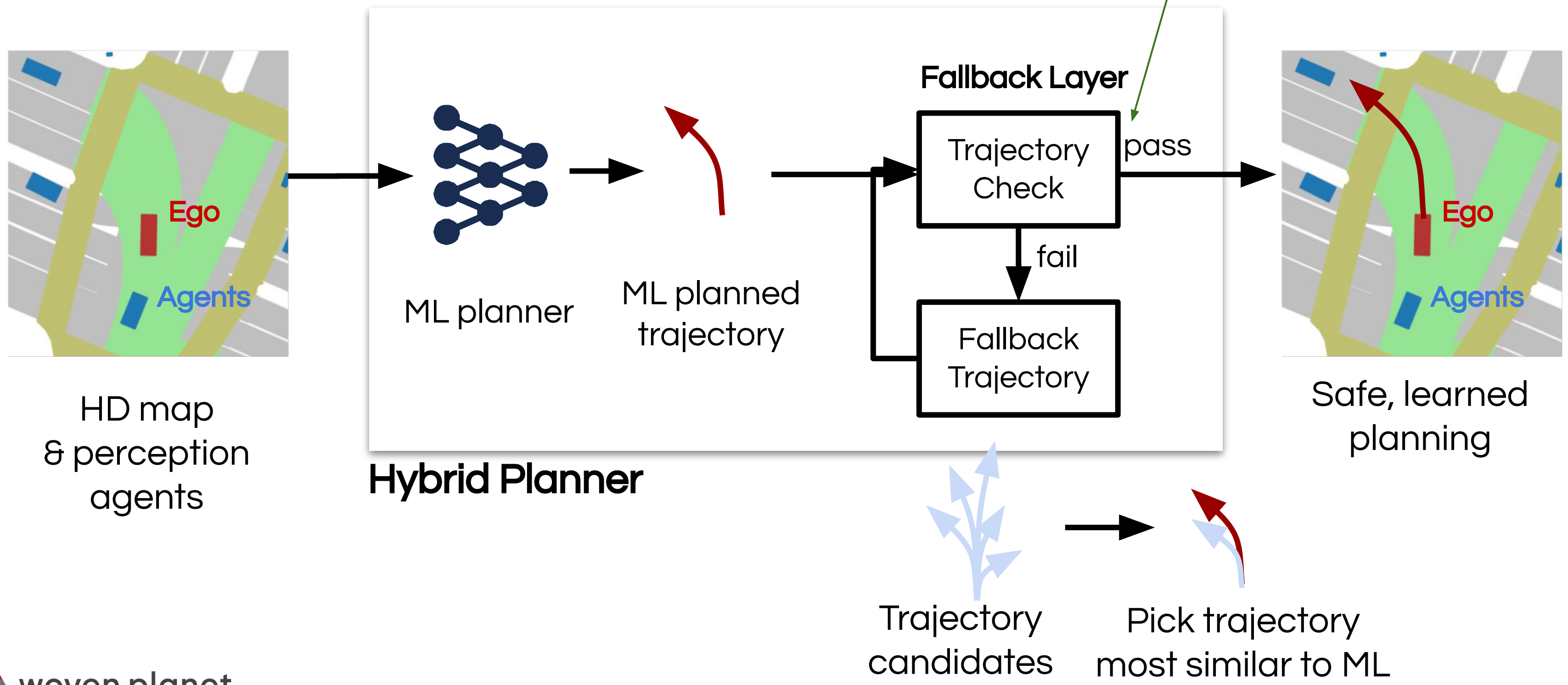
On the Road: Deployment



woven planet

SafetyNet - Hybrid Planner

✅ Comfort
✅ Safety
✅ Legality



[1]SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies (Vitelli, 2021)

SafetyNet - Hybrid Planner

Hybrid Planner can reduce safety issues, in simulation, in tough urban ODDs

- L4 system evaluation
- Evaluated on scenes from Palo Alto and San Francisco

	Number of simulated events per 1000 miles			
Planner Type	**Simulated Collisions**	**Close Calls**	**Discomfort Braking**	**Passiveness**
ML Planner-Only	**92**	75	313	214
Hybrid Planner	**5**	45	24	277
Change Rate	***-95%***	*-40%*	*-92.3%*	*+29.5%*

Key property: we can make ML planning safer

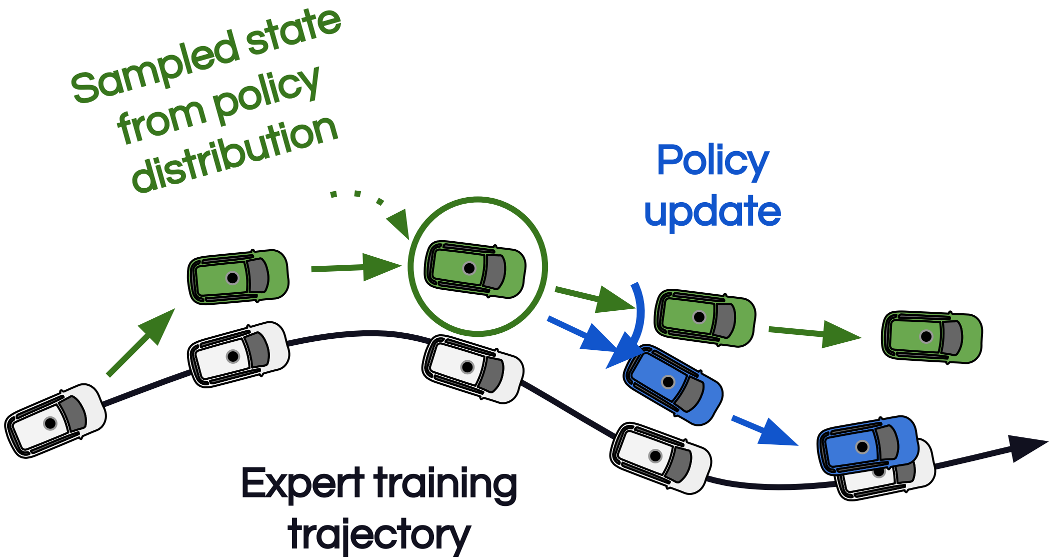
[1]SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies (Vitelli, 2021)

Road testing in busy San Francisco

What’s Next

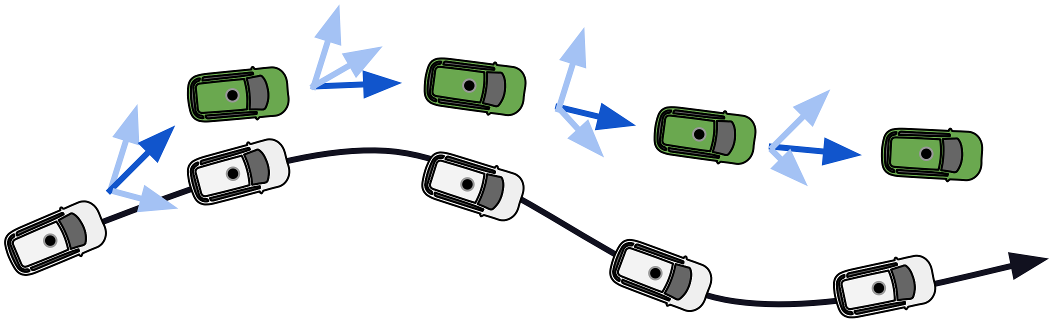
What's Next

- **Close the gap of sequential decision making**
- Learning to search
- Scaling data
- Leveraging low-cost sensor data at scale



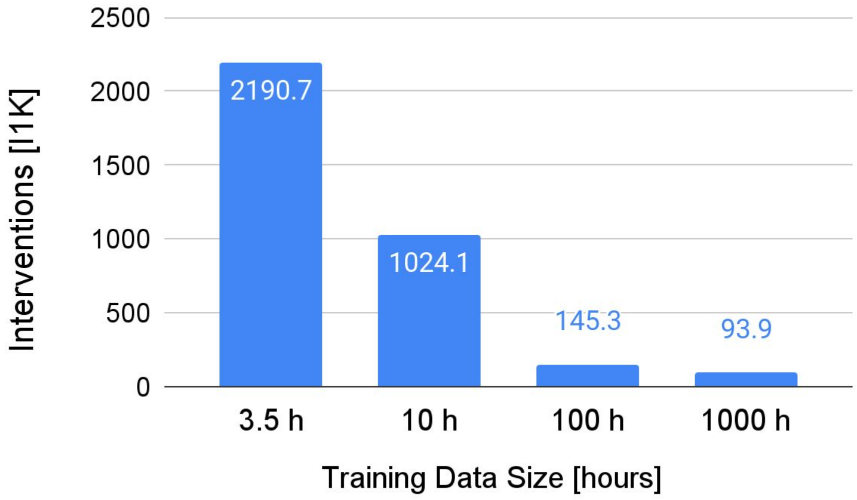
What's Next

- Close the gap of sequential decision making
- **Learning to search**
- Scaling data
- Leveraging low-cost sensor data at scale



What's Next

- Close the gap of sequential decision making
- Learning to search
- **Scaling data**
- Leveraging low-cost sensor data at scale



What's Next

- Close the gap of sequential decision making
- Learning to search
- Scaling data
- **Leveraging low-cost sensor data at scale**

Thanks ! We are hiring!

Offices

Palo Alto, London, Tokyo

Roles

Software Engineer
Deep Learning Engineer
Research Engineer
Research Scientist

Careers

https://boards.greenhouse.io/l5

2000+ people, 3 global offices and remote

woven planet