Forecasting and Decision Making in self-driving

Sanjiban Choudhury

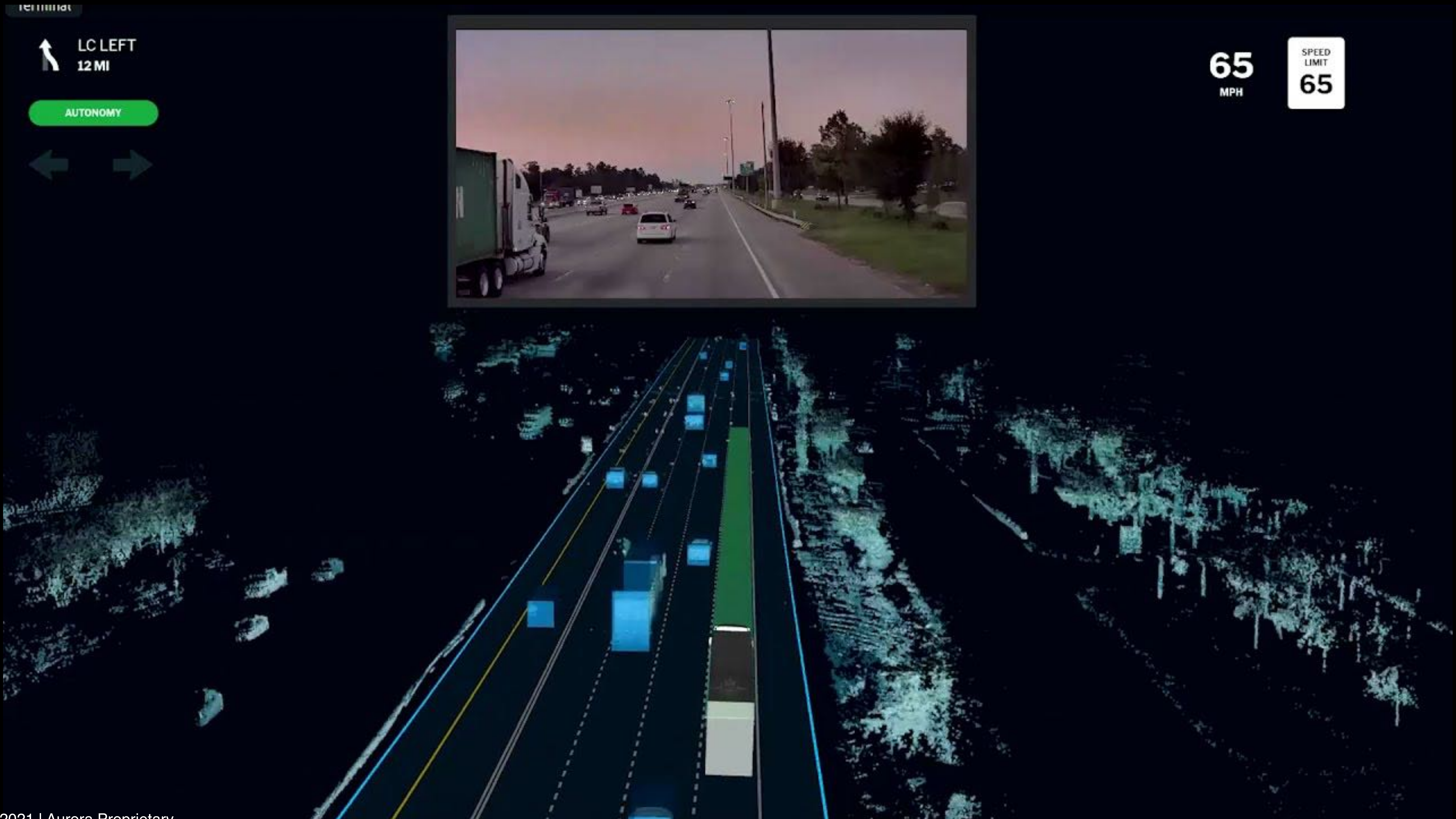
Cornell Bowers CIS
Computer Science

“Trying to predict the future is a mug’s game...

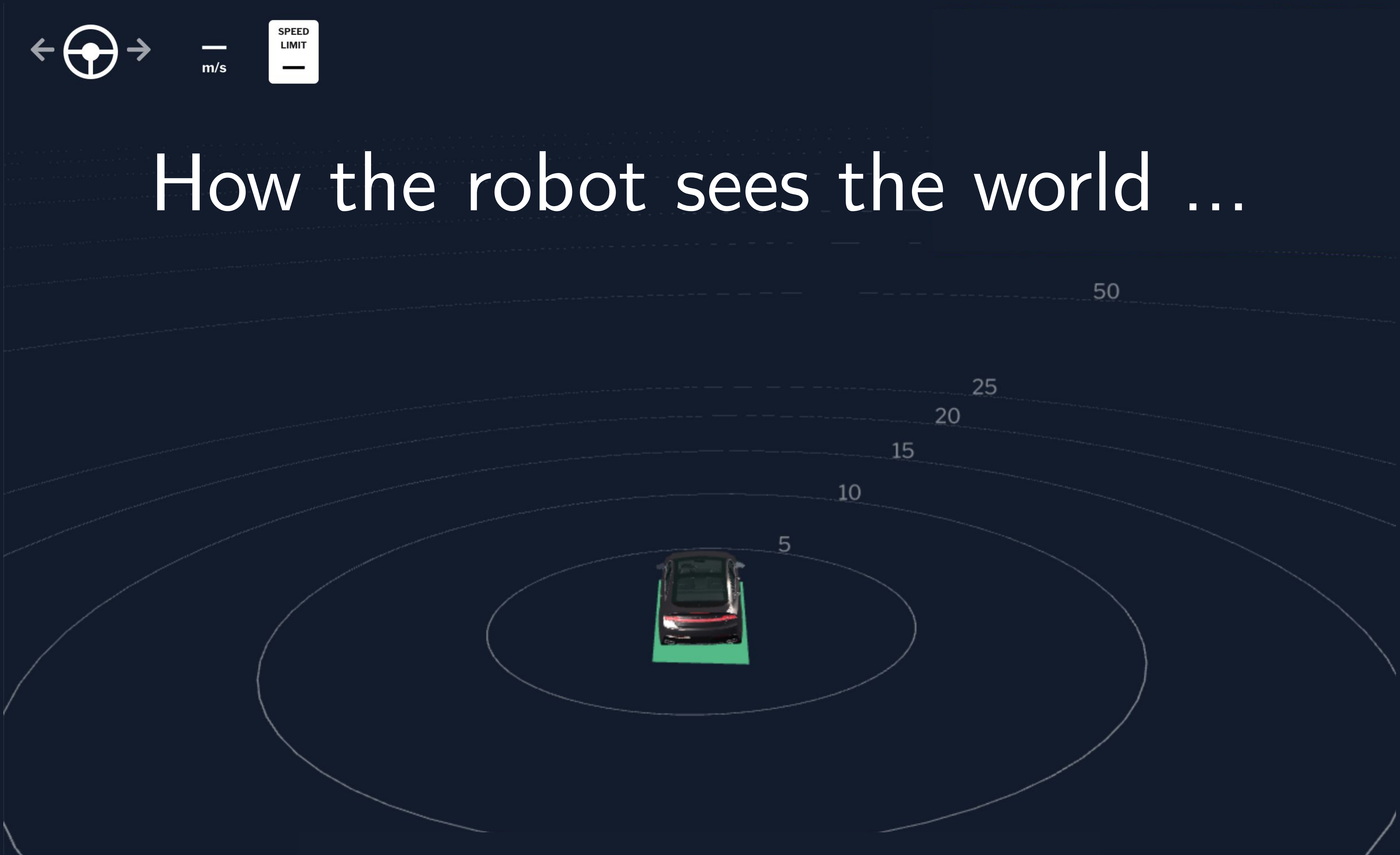
... But increasingly it’s a game we all have to play because the world is changing so fast and we need to have some sort of idea of what the future’s actually going to be like because we are going to have to live there, probably next week.”

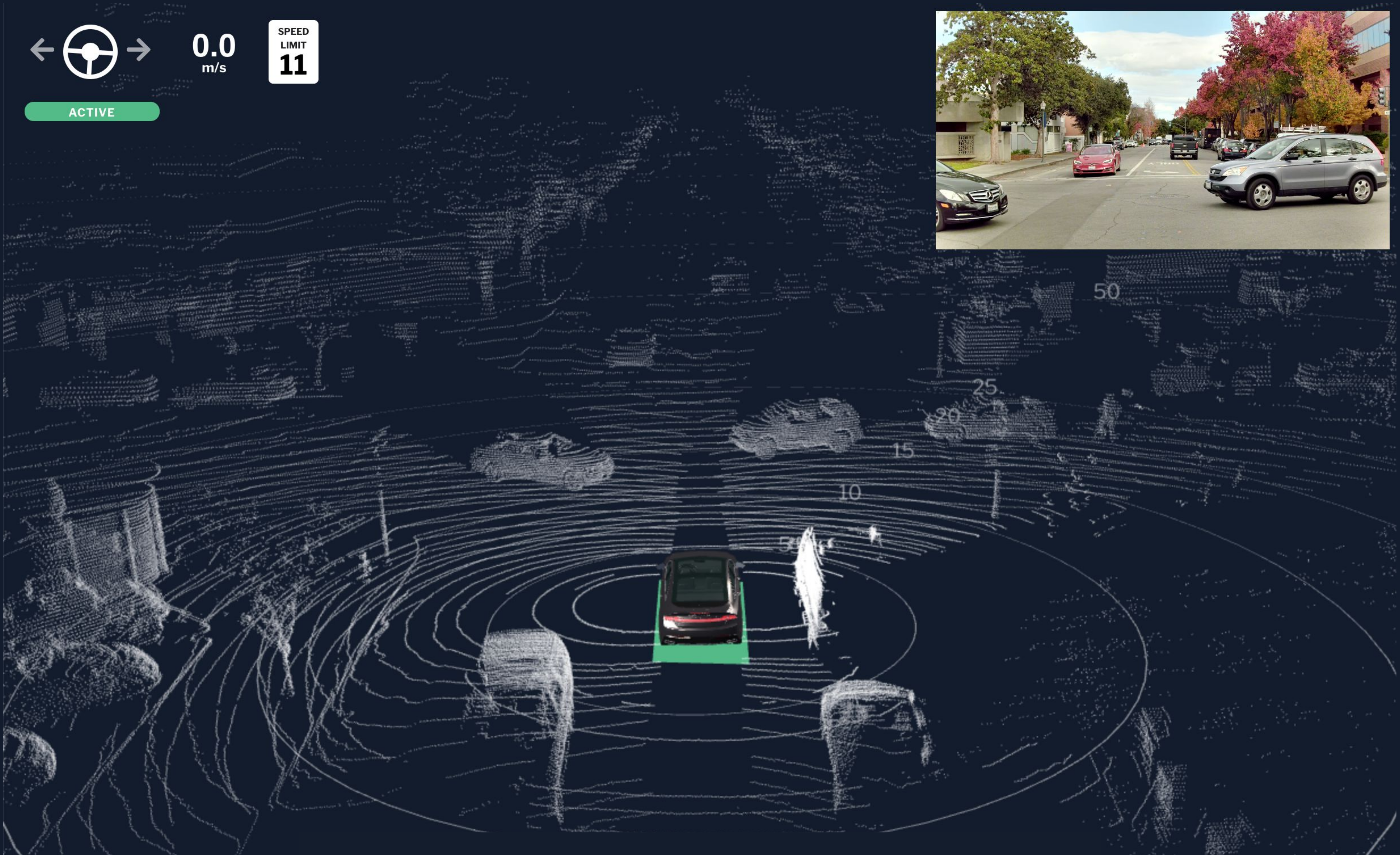
Douglas Adams

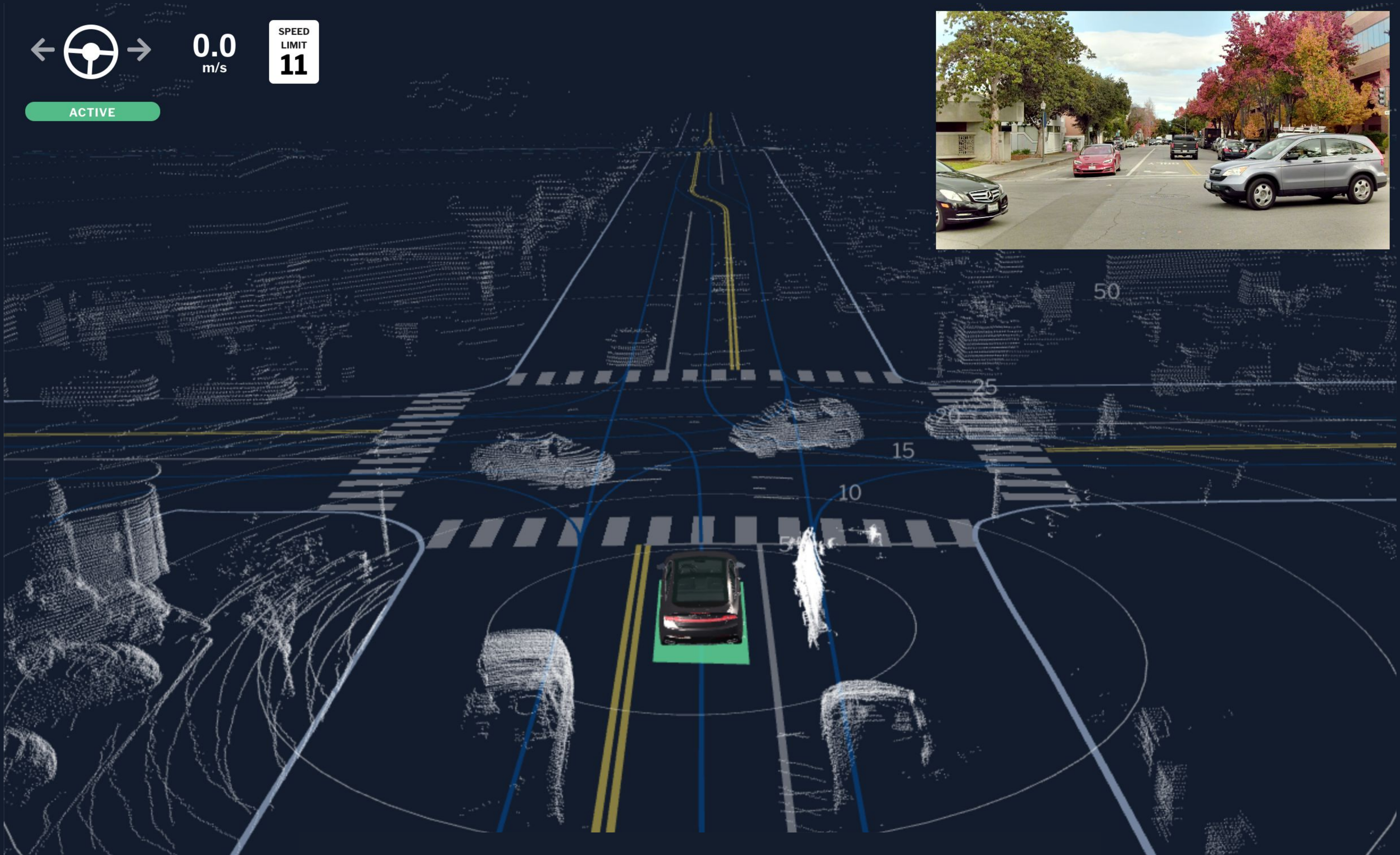
The Salmon on of Doubt

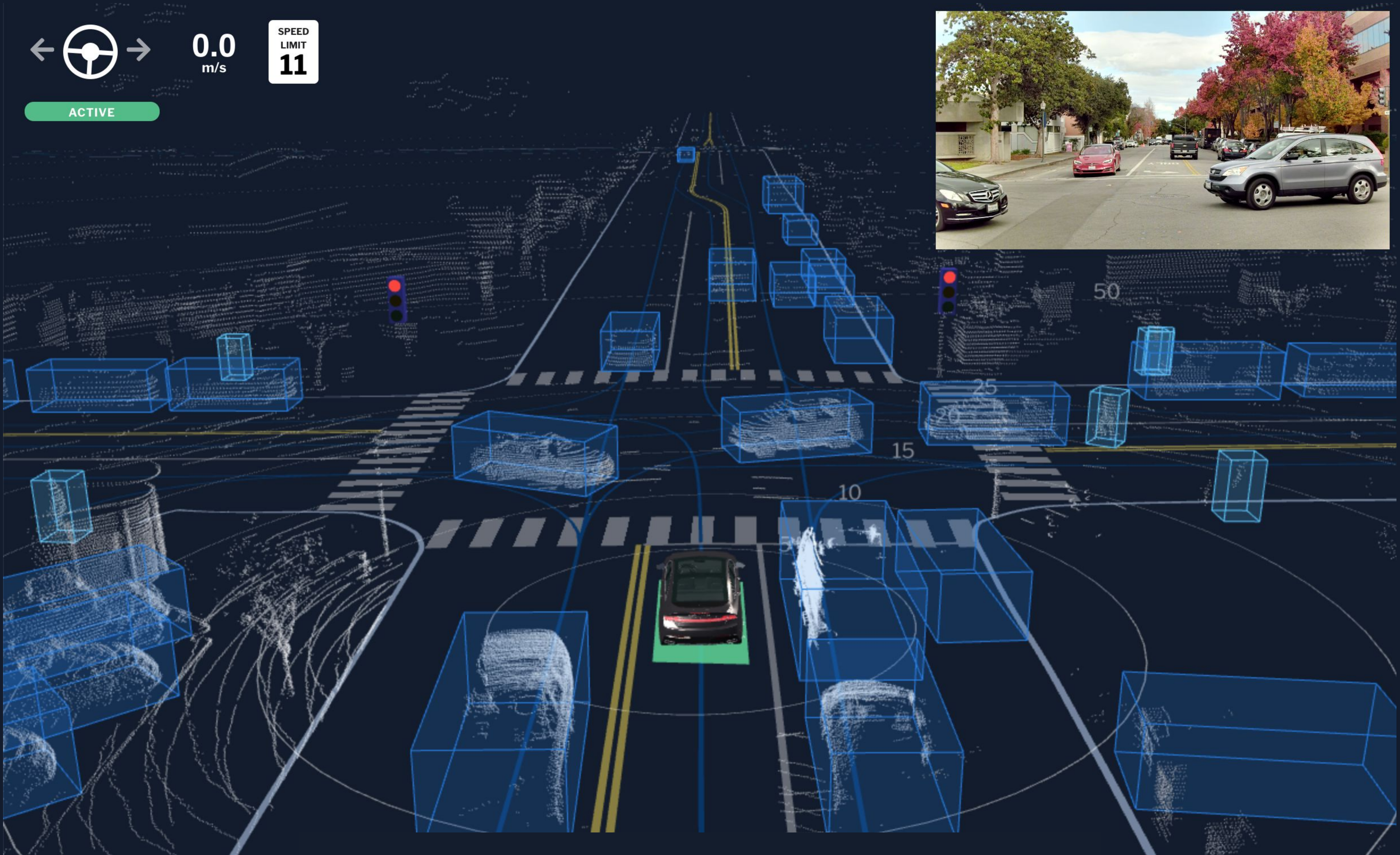

Terminal
LC LEFT
12 MI
AUTONOMY
65
MPH
SPEED LIMIT
65

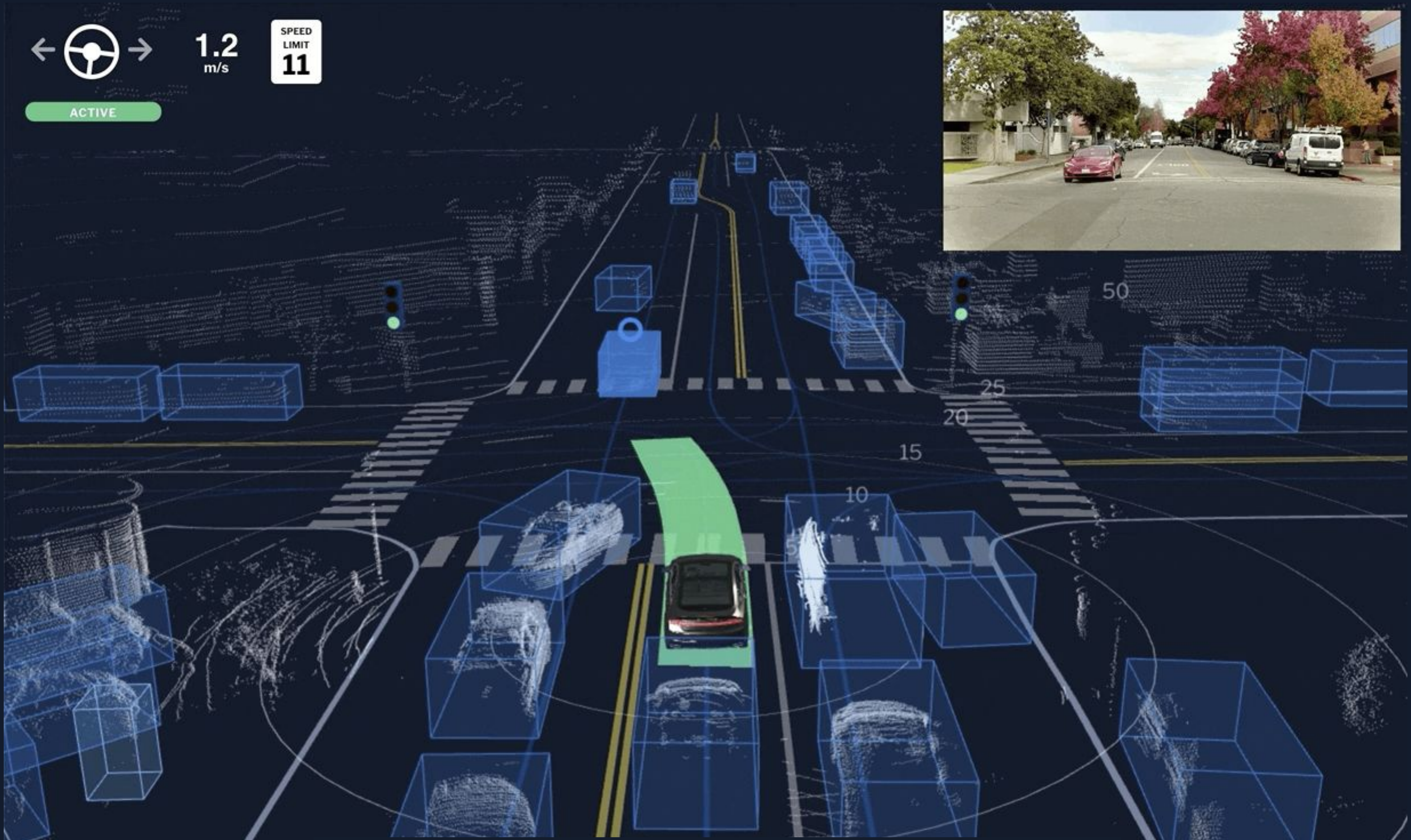
How the robot sees the world …



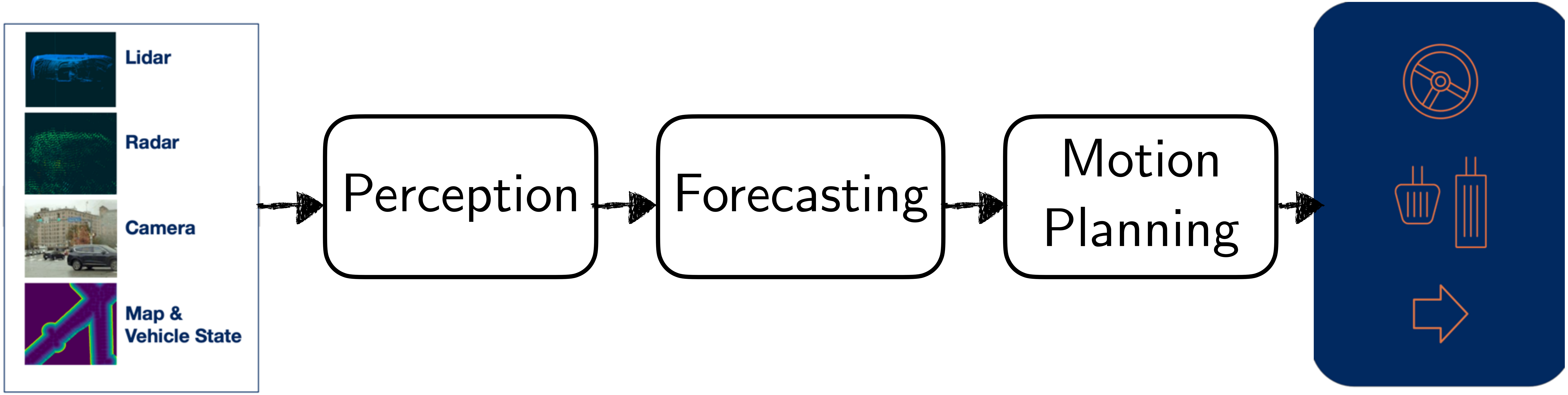

0.0
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10
5


0.0
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10
5


0.0
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10
5


1.2
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10
5

Traditional Architecture



Lidar

Radar

Camera

Map & Vehicle State

Perception

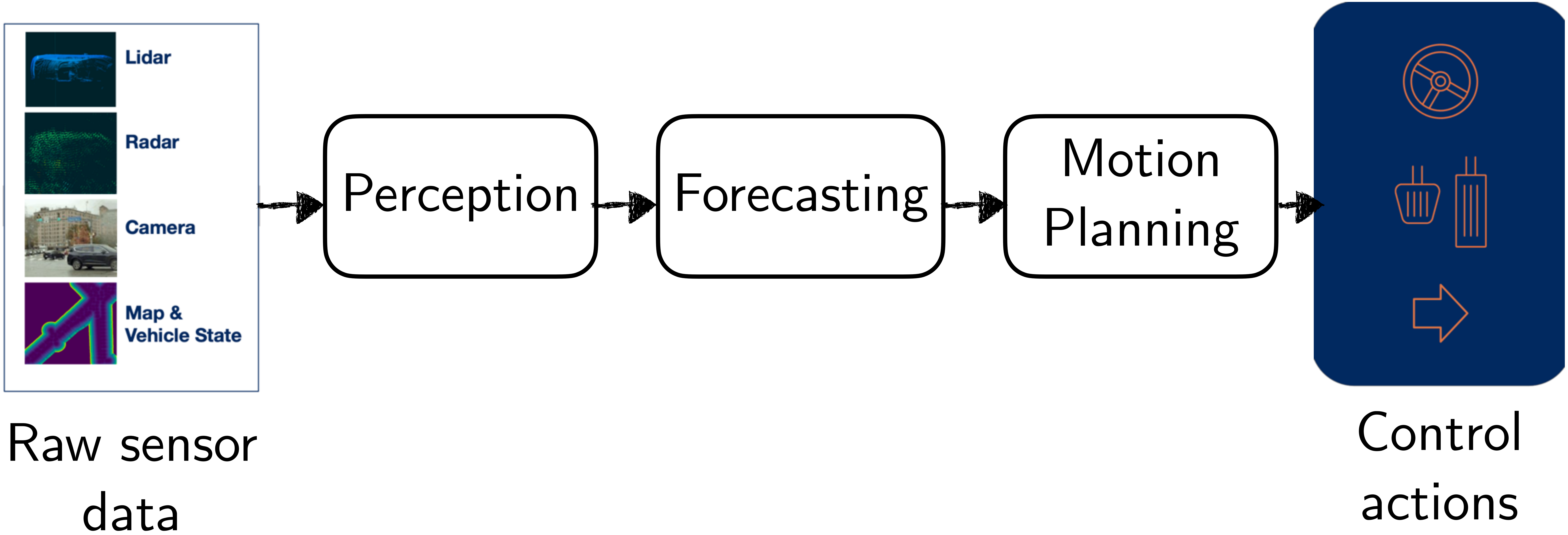
Forecasting

Motion Planning

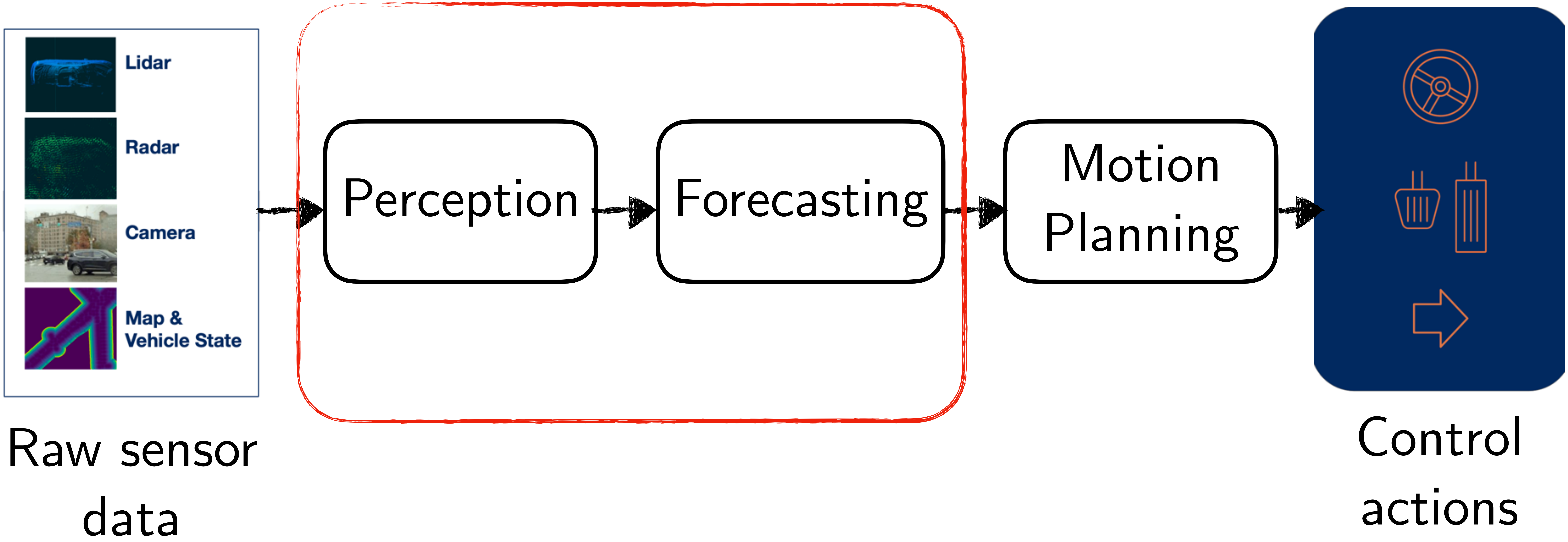
Raw sensor data

Control actions

Is having cascaded blocks a good idea?



Lots of recent work on unifying perception and forecasting



Lidar

Radar

Camera

Map & Vehicle State

Raw sensor data

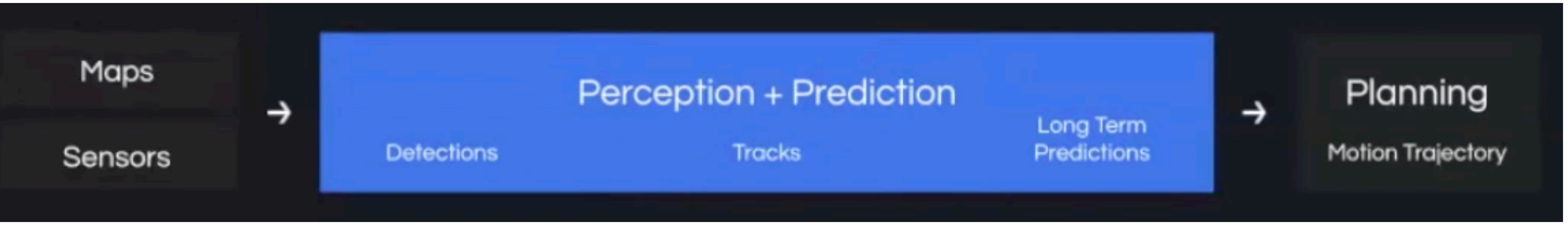
Perception

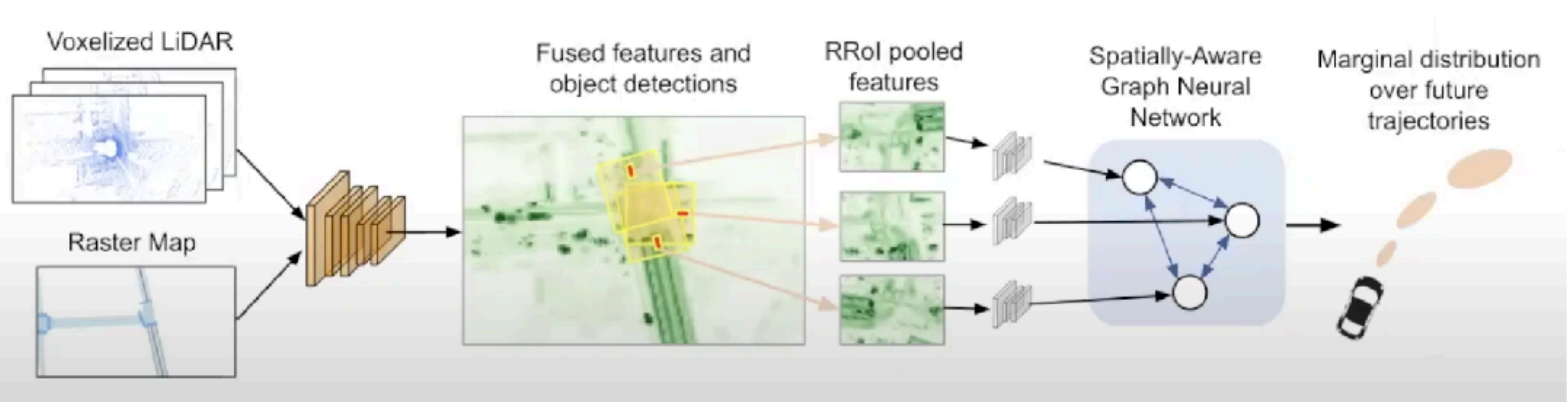
Forecasting

Motion Planning

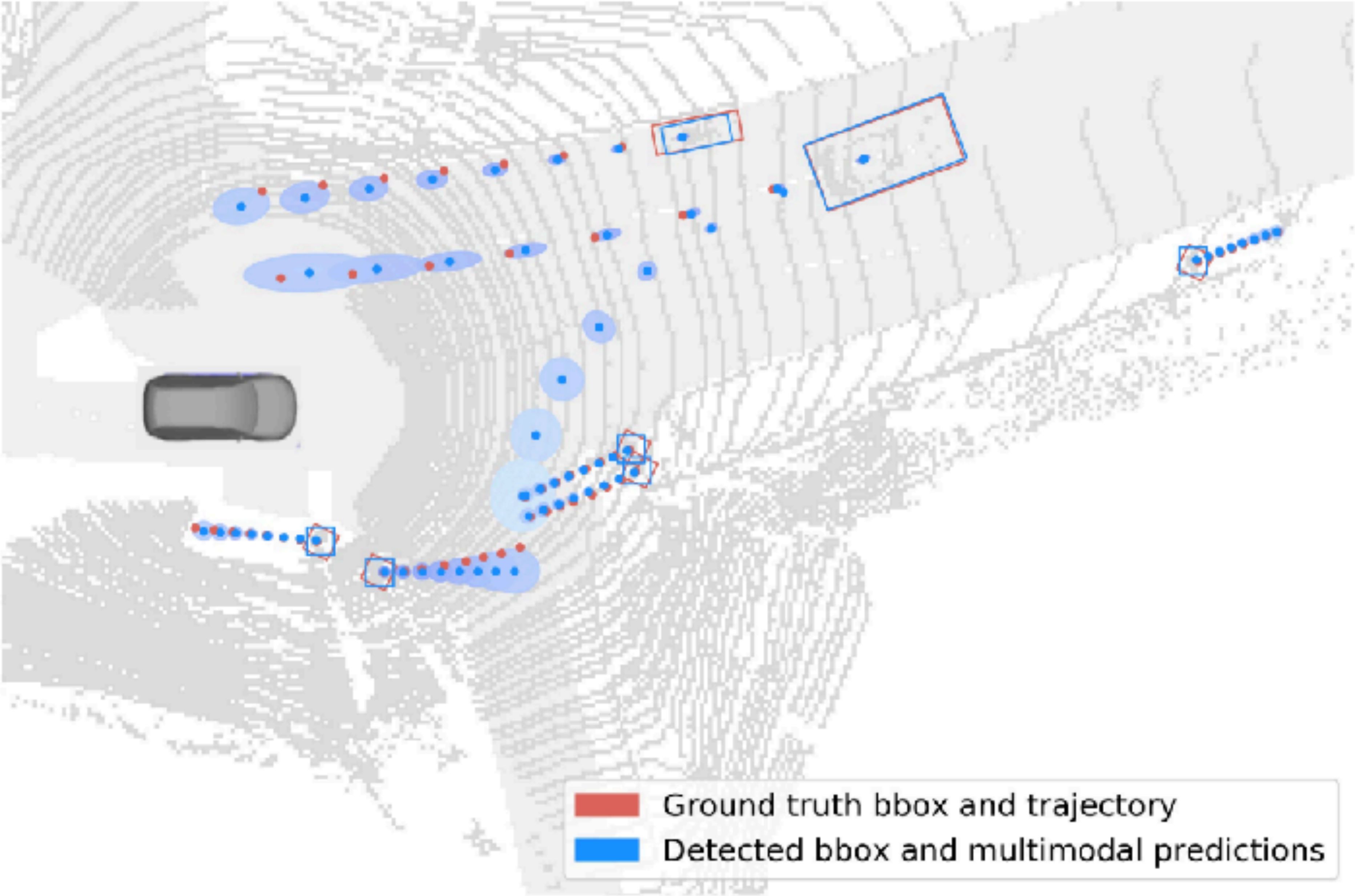
Control actions

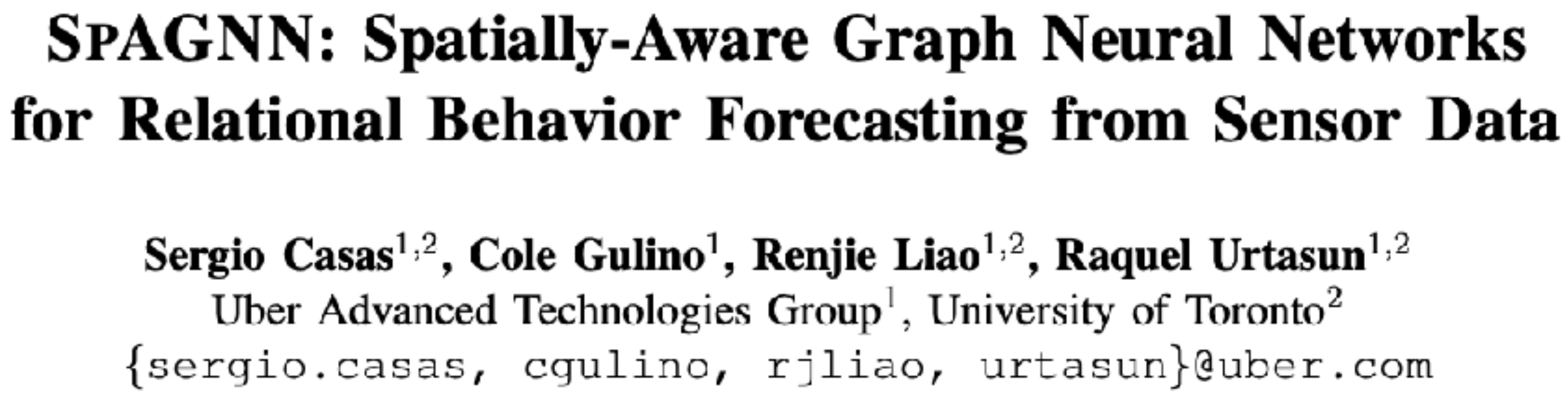
Lots of recent work on unifying perception and forecasting





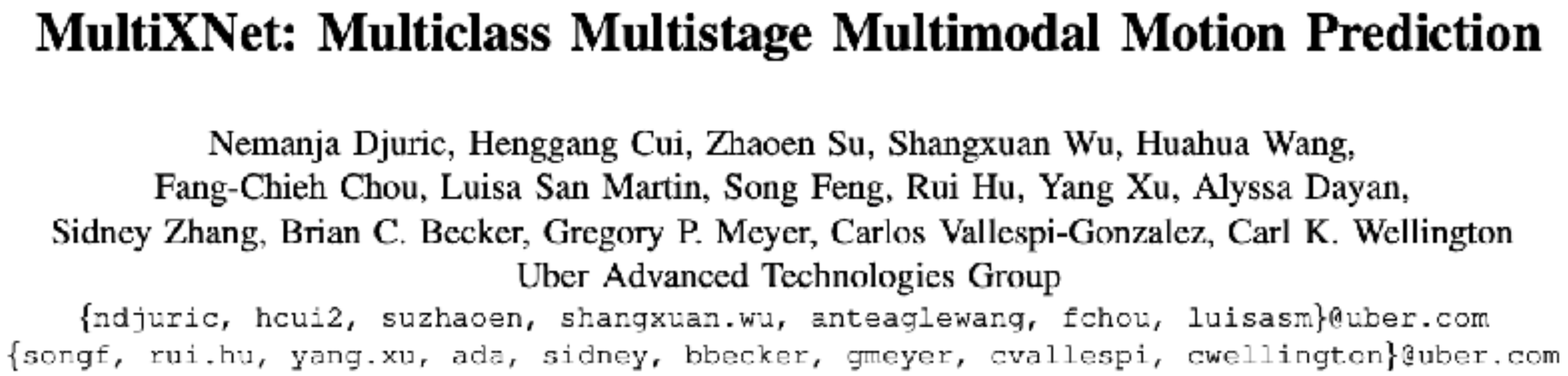
Voxelized LiDAR
Raster Map
Fused features and object detections
RRoI pooled features
Spatially-Aware Graph Neural Network
Marginal distribution over future trajectories





SpAGNN: Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting from Sensor Data

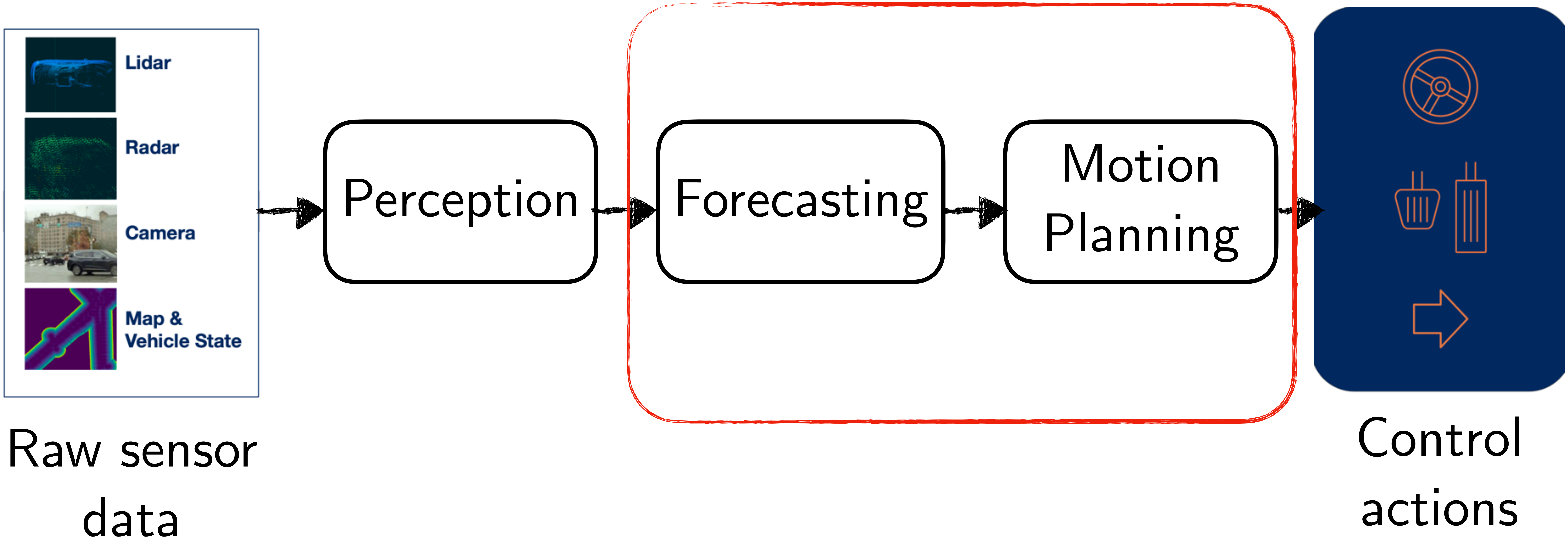
Sergio Casas[1,2], Cole Gulino[1], Renjie Liao[1,2], Raquel Urtasun[1,2]
Uber Advanced Technologies Group[1], University of Toronto[2]
{sergio.casas, cgulino, rjliao, urtasun}@uber.com



MultiXNet: Multiclass Multistage Multimodal Motion Prediction

Nemanja Djuric, Henggang Cui, Zhaoen Su, Shangxuan Wu, Huahua Wang,
Fang-Chieh Chou, Luisa San Martin, Song Feng, Rui Hu, Yang Xu, Alyssa Dayan,
Sidney Zhang, Brian C. Becker, Gregory P. Meyer, Carlos Vallespi-Gonzalez, Carl K. Wellington
Uber Advanced Technologies Group
{ndjuric, hcui2, suzhaoen, shangxuan.wu, anteaglewang, fchou, luisasm}@uber.com
{songf, rui.hu, yang.xu, ada, sidney, bbecker, gmeyer, cvallespi, cwellington}@uber.com

But what about forecasting and motion planning?



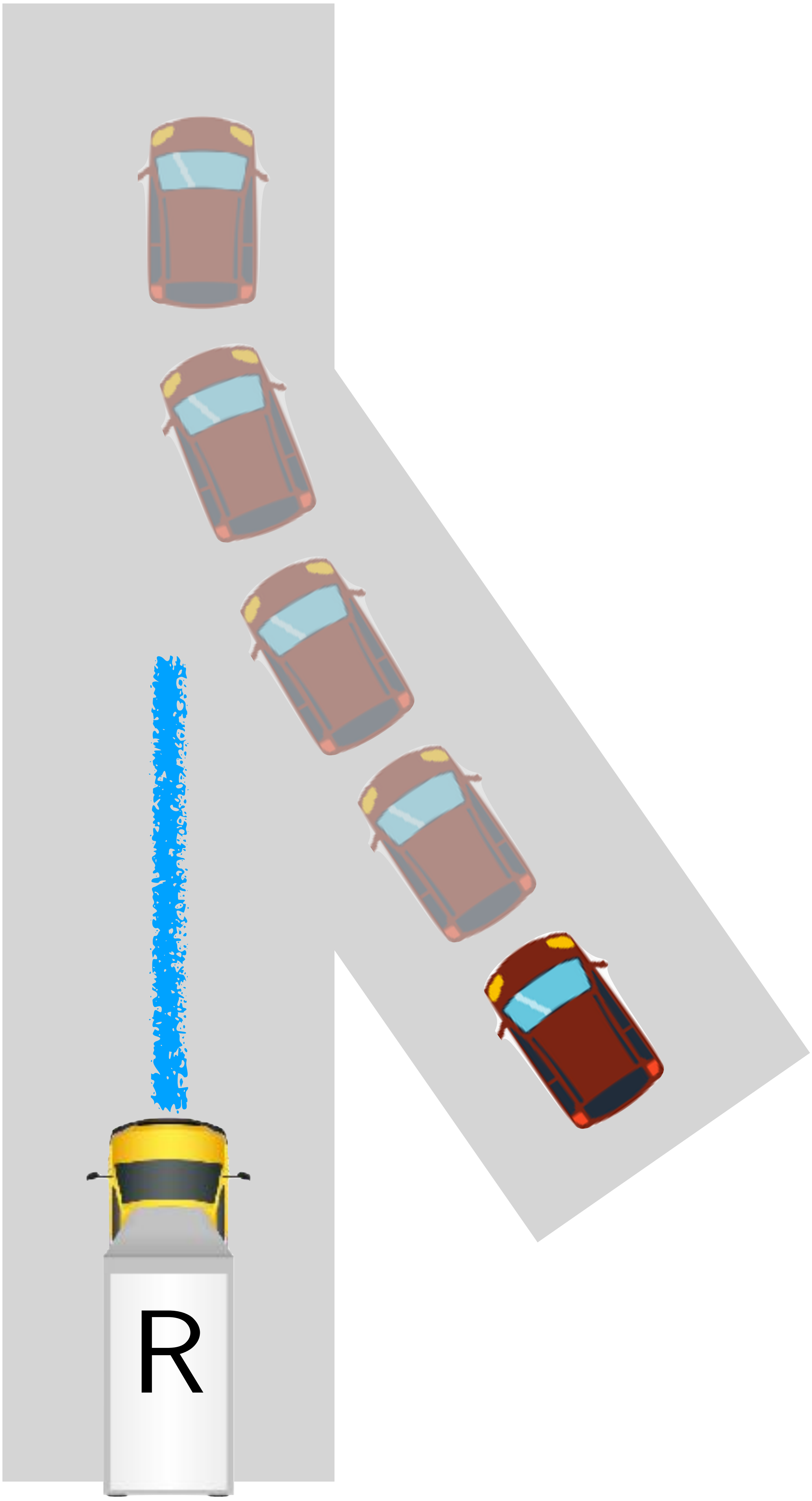
Shaky foundations of forecasting

Are we using the right model?

Are we collecting data correctly?

Are we using the right loss?

Example: Learning forecasts for merging actors



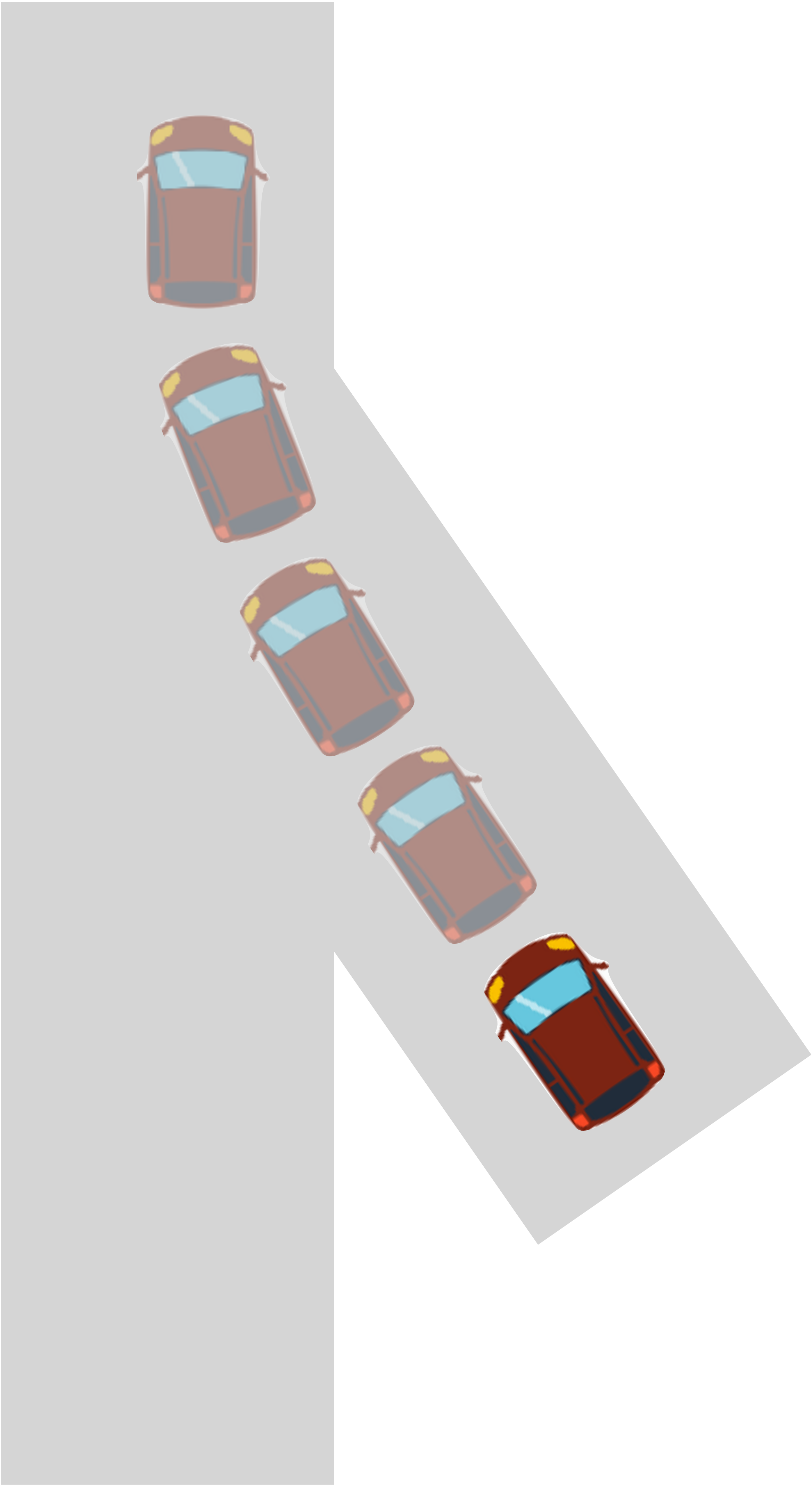
Goal

1. Predict 5s future trajectory

2. Plan with 5s future trajectory

Activity!

Example: Learning forecasts for merging actors



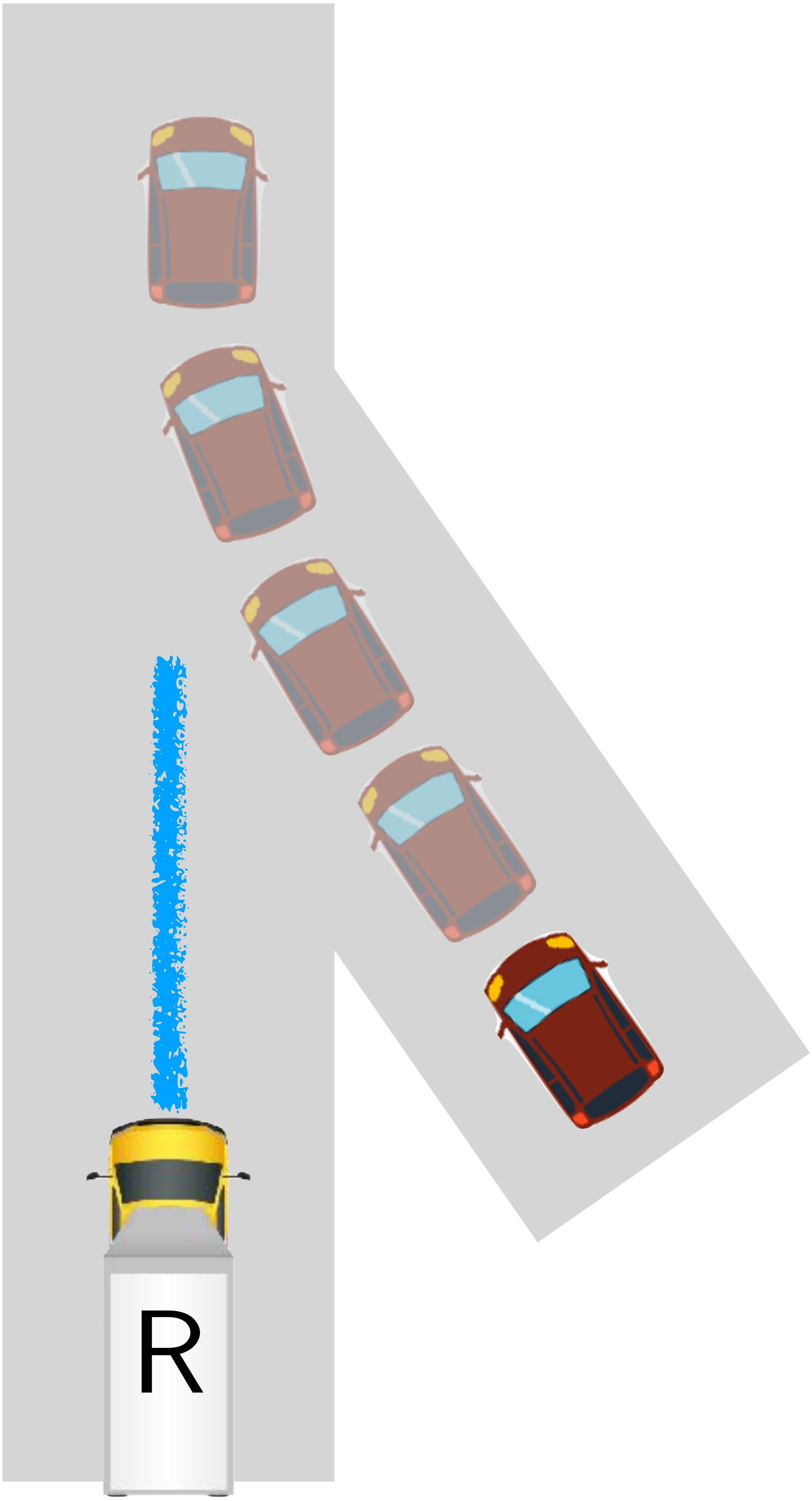
1. Predict 5s future trajectory

Data?

Model?

Loss?

Example: Learning forecasts for merging actors



2. Plan with 5s future trajectory

Cost function?

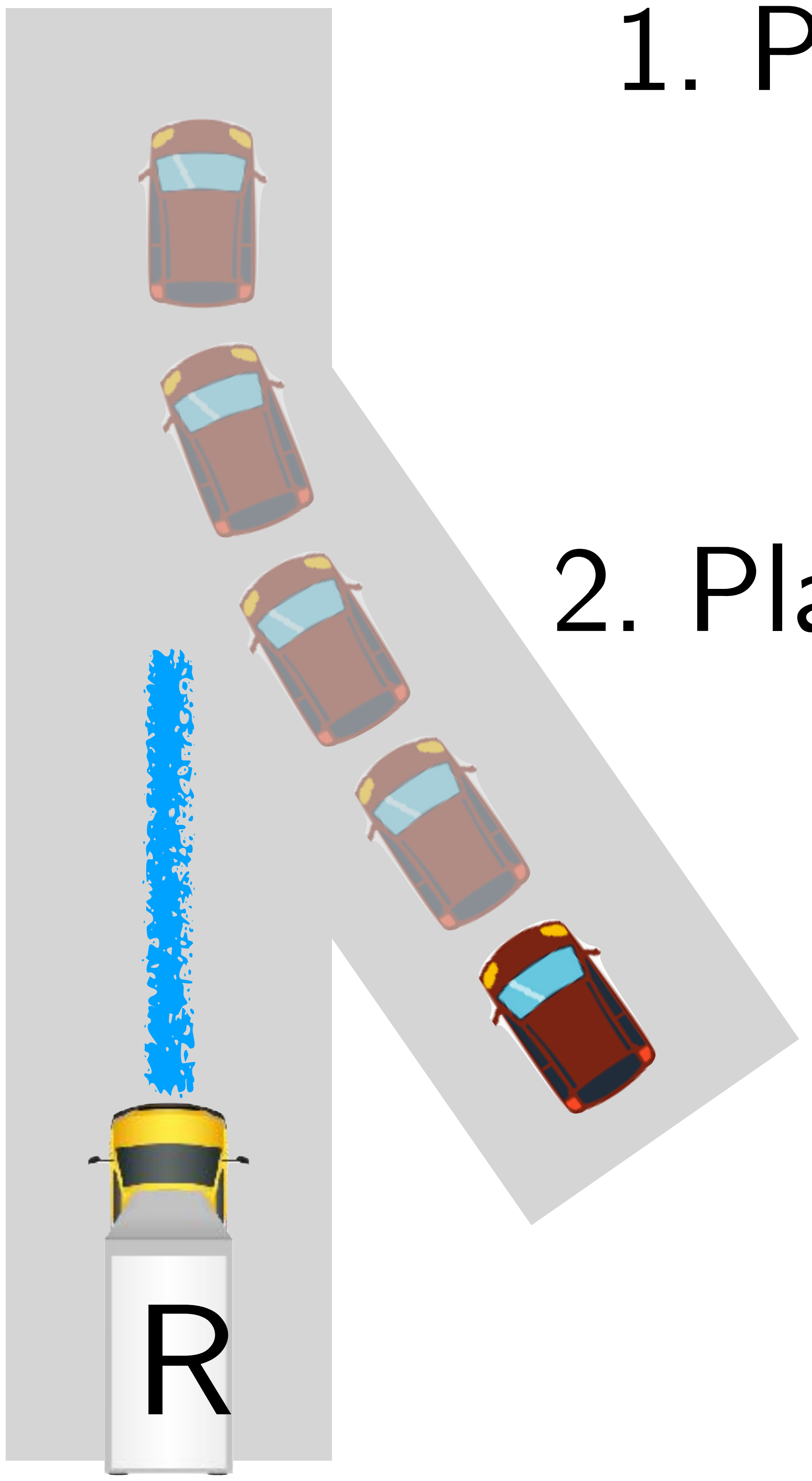
Planner?

Think-Pair-Share!

Think (30 sec): Design choices for forecasting and motion planning

Pair: Find a partner

Share (45 sec): Partners exchange ideas



1. Predict 5s future trajectory

Data? Model? Loss?

2. Plan with 5s future trajectory

Cost Function?
Planner?

Why is current state insufficient to predict future?

Simple latent variables:

Velocity, Acceleration may not be observable

Complex latent variables:

Intent (turning left, making a lane change) are
not observable and must be inferred from past actions

Sequence Model

A very brief history of sequence prediction in robotics

Kalman Filter + Prediction

Hand design observation models, infer latent states, forward predict.

RNN, LSTMs

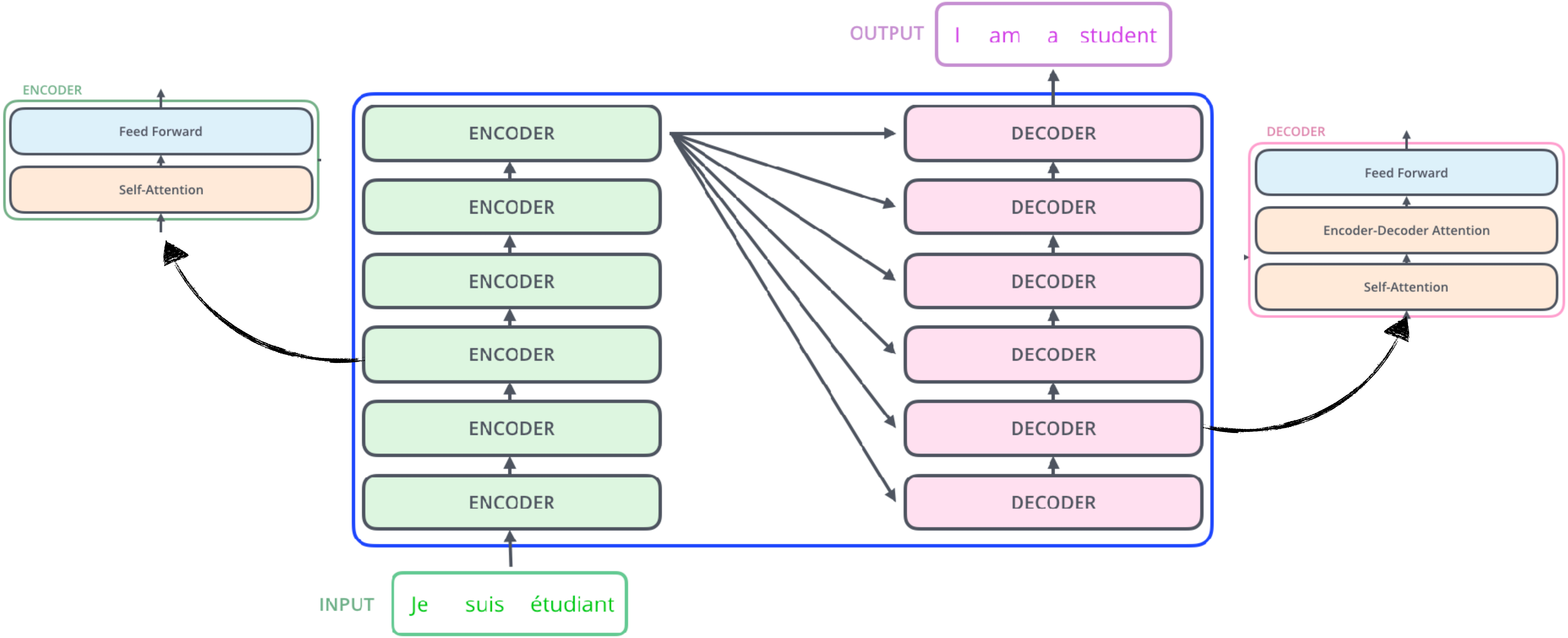
Learn the filter! Problem - forget long sequences since only one hidden state vector passed from one time step to next

Transformers

Retain all hidden state, pay O(H^2) computation

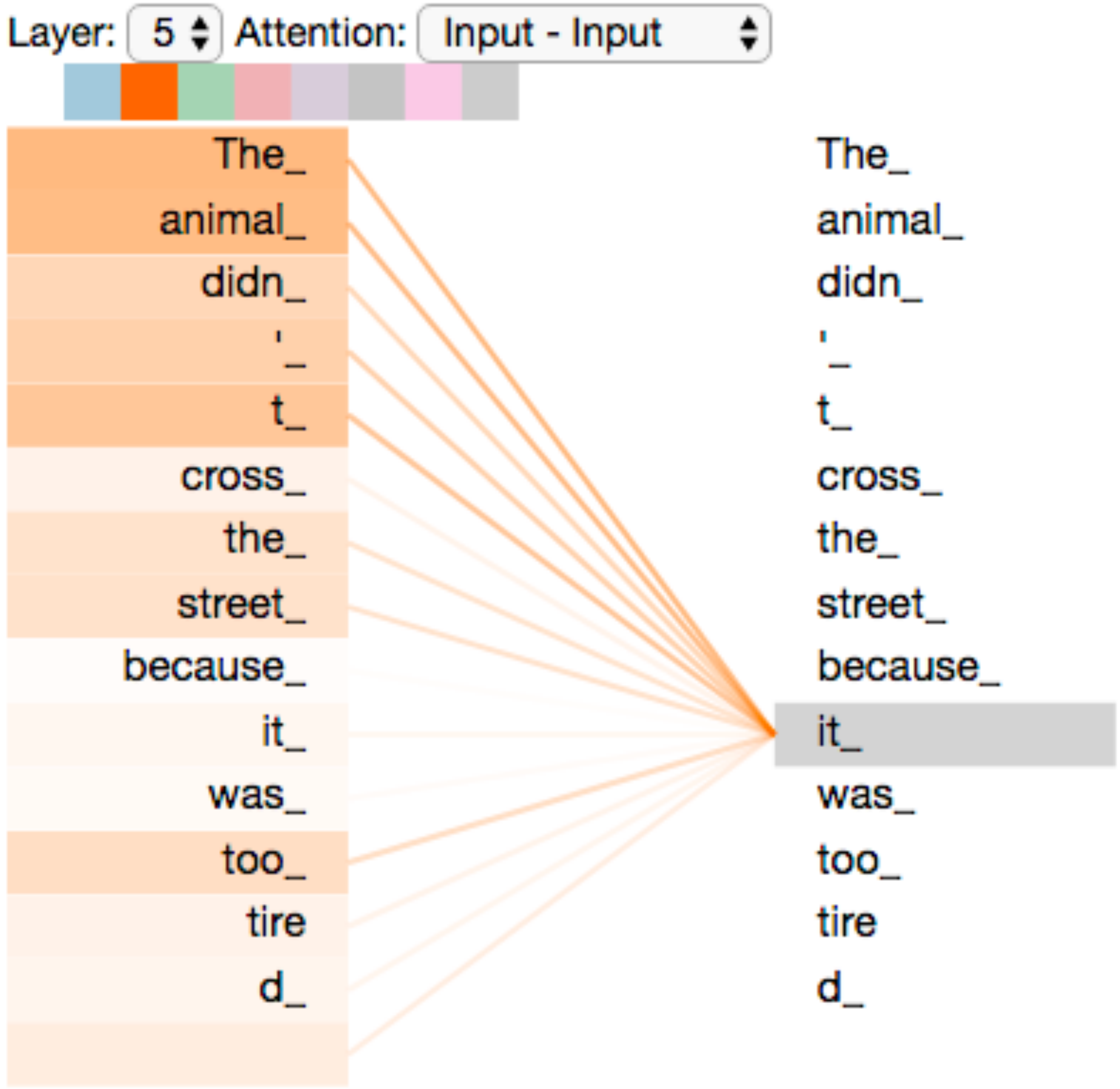
Given sequence of English words, predict sequence of French

Transformer Architecture



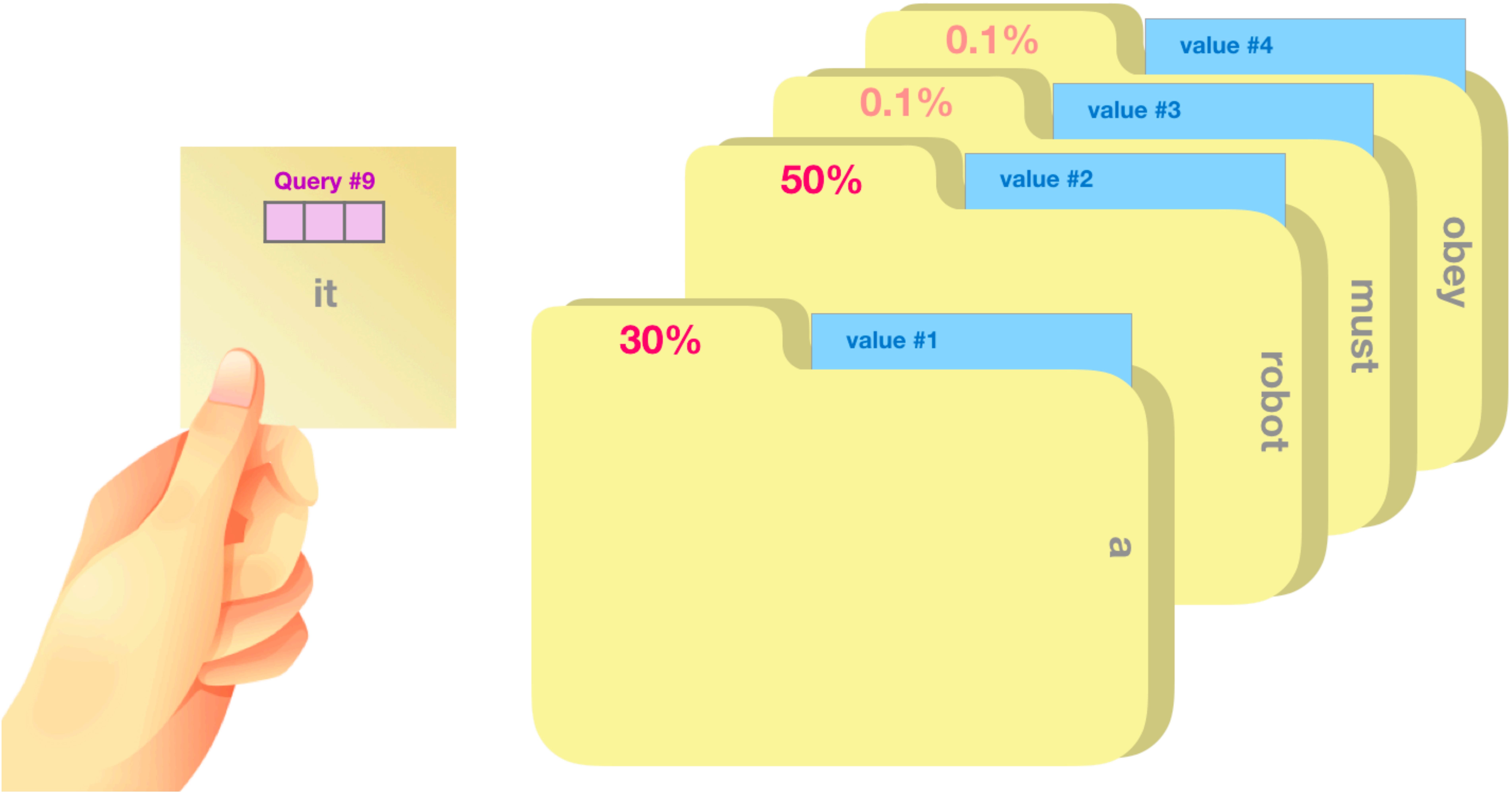
Visualizing attentions

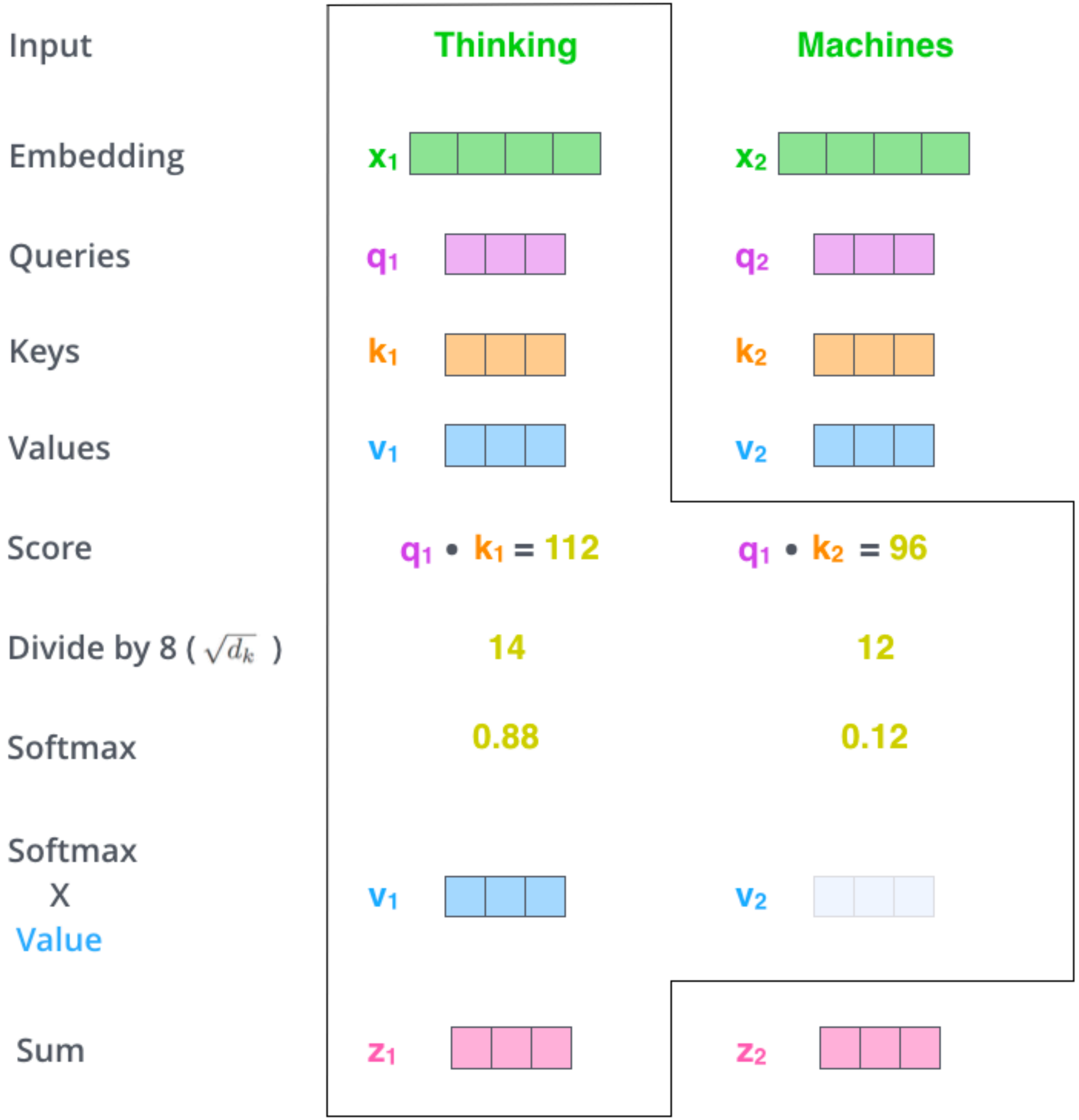
”`The animal didn't cross the street because it was too tired`”



Layer: 5 Attention: Input - Input

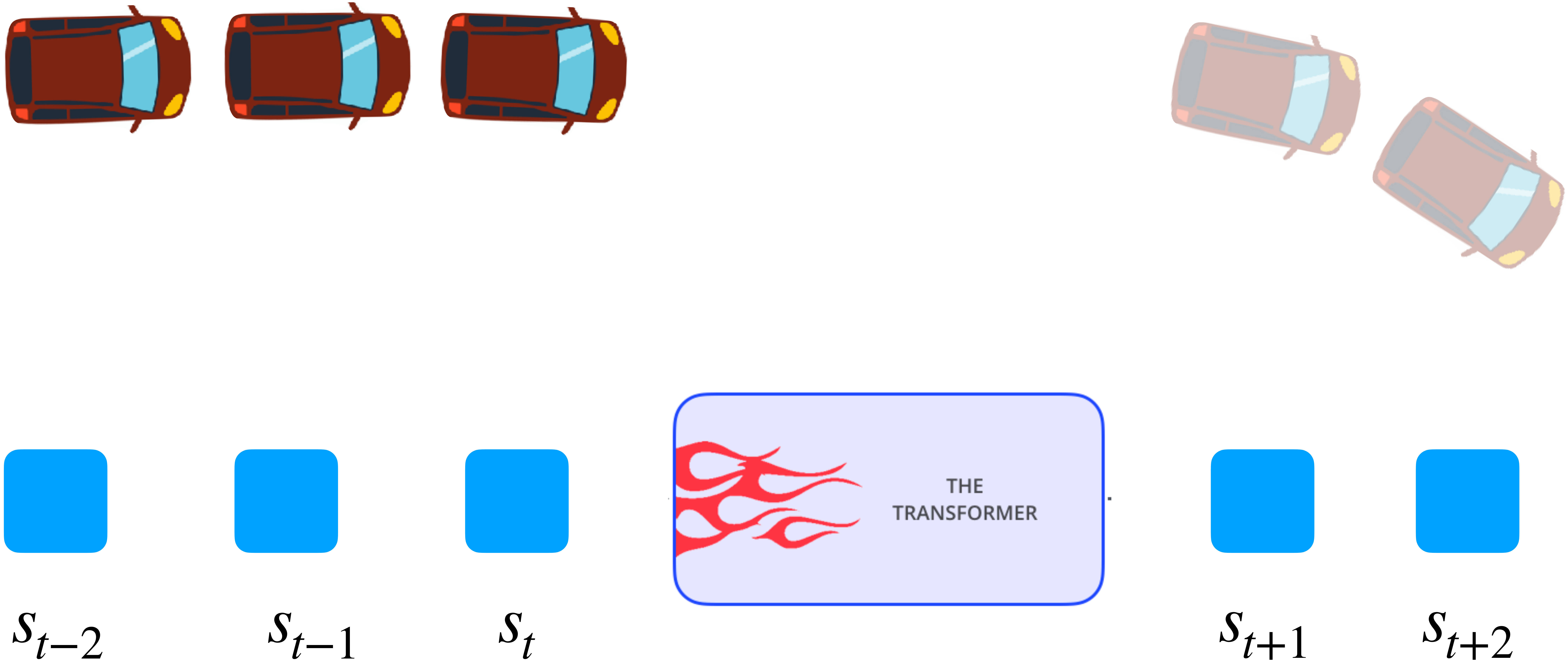
Attention as a soft-memory look up





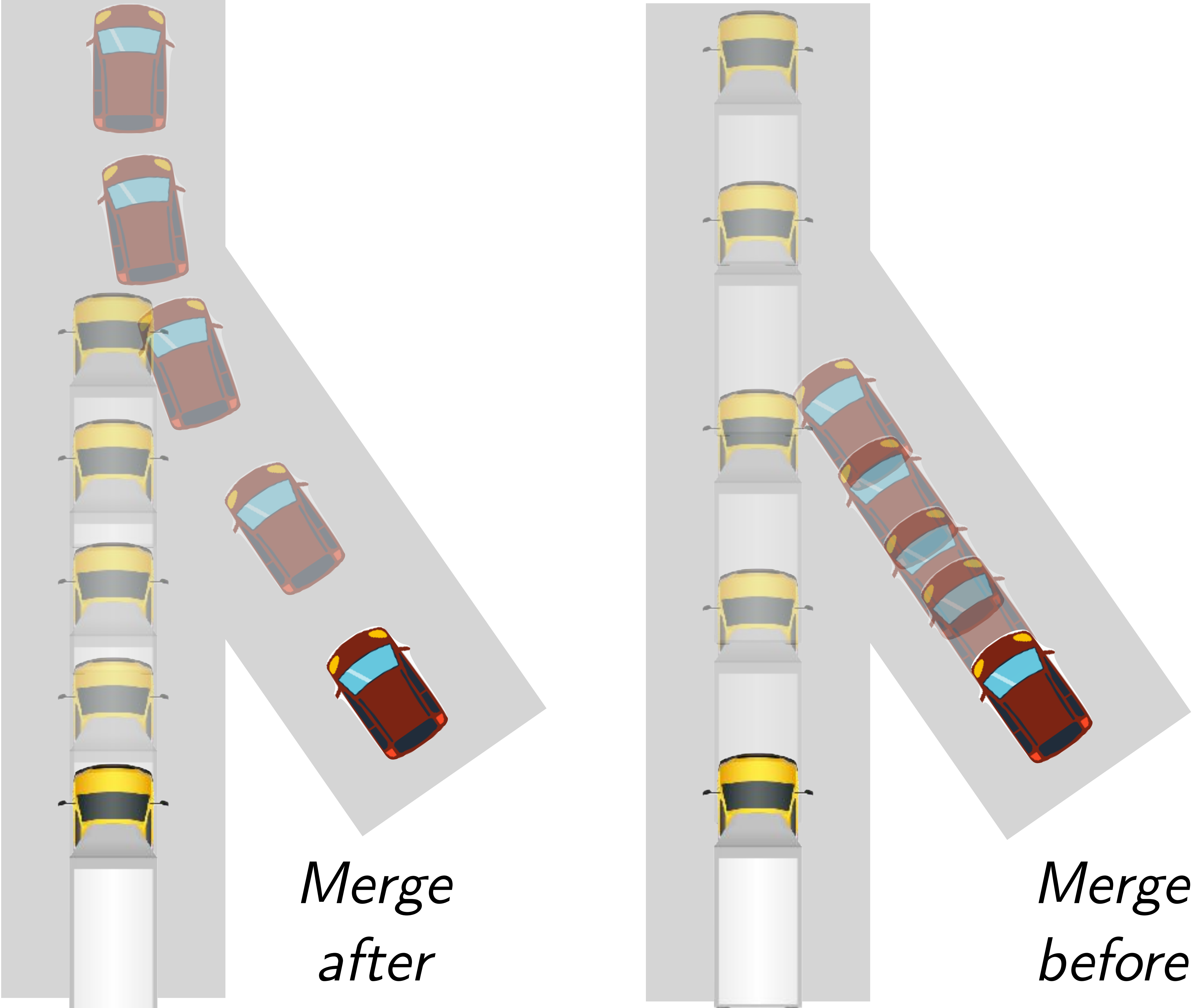
Back to forecasting

Transformers for motion prediction



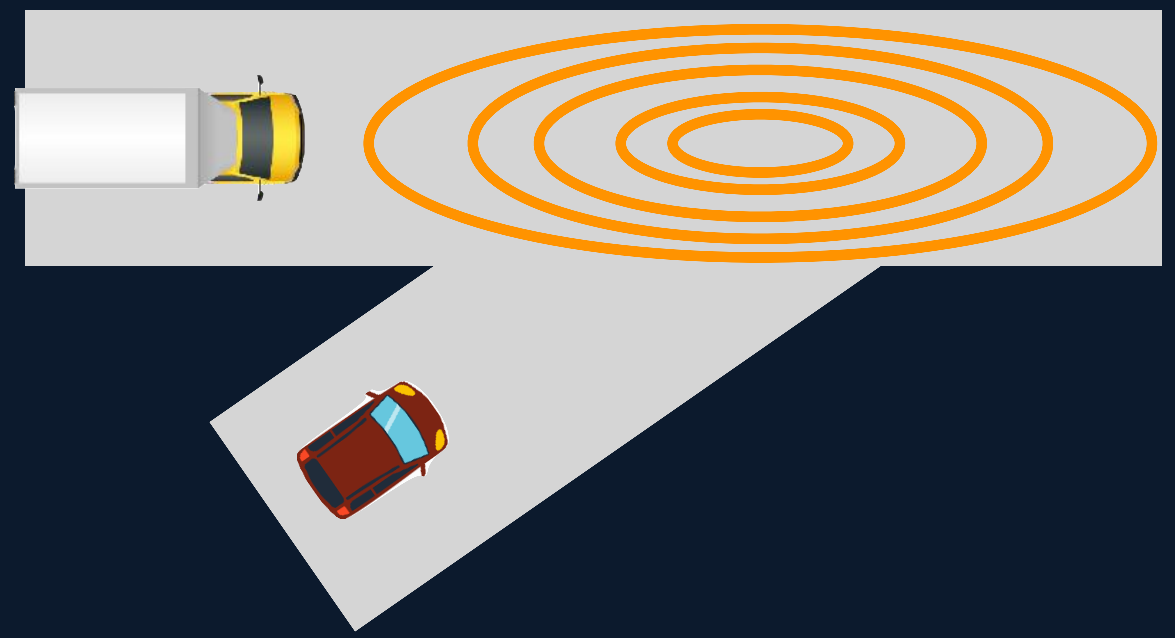
What happens with a typical forecasting approach?

Train Data



1. Collect lots of driving data of actors merging

2. Train a forecast model to predict actor future



Forecasts have huge variance!

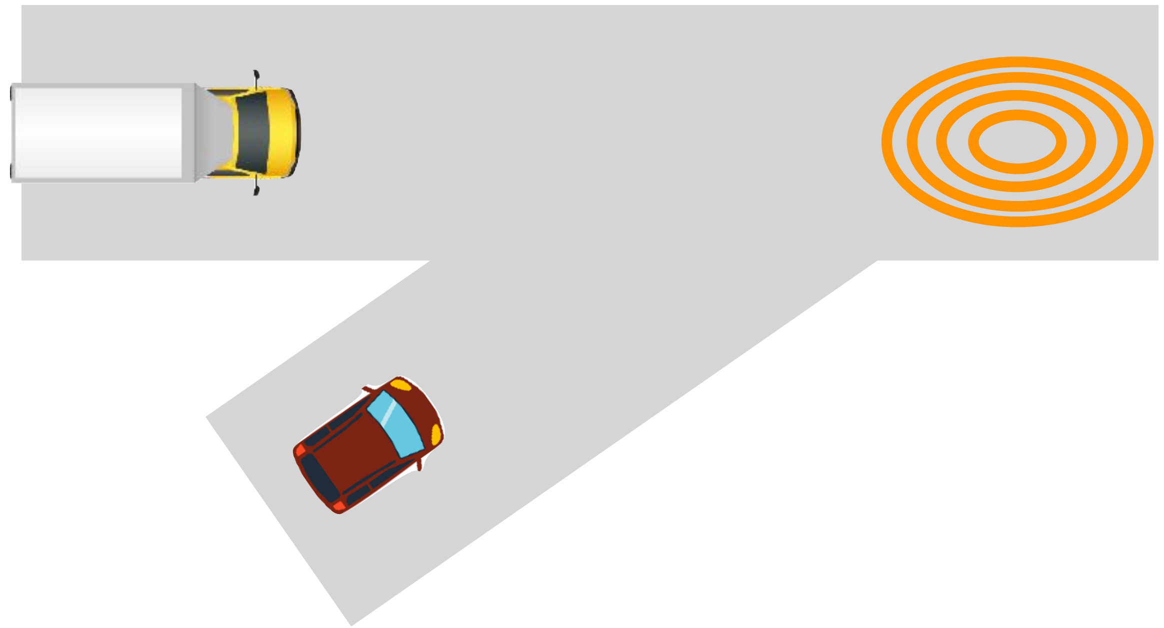
Planner brakes aggressively!

Why is the forecast so whacky?

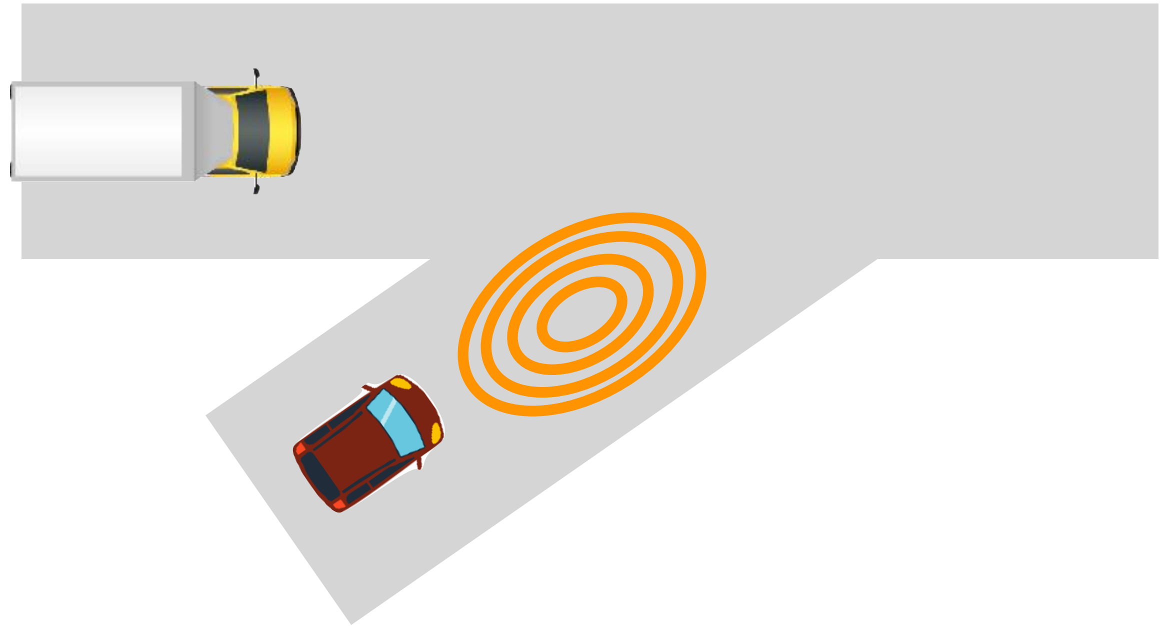
Why is the forecast so whacky?

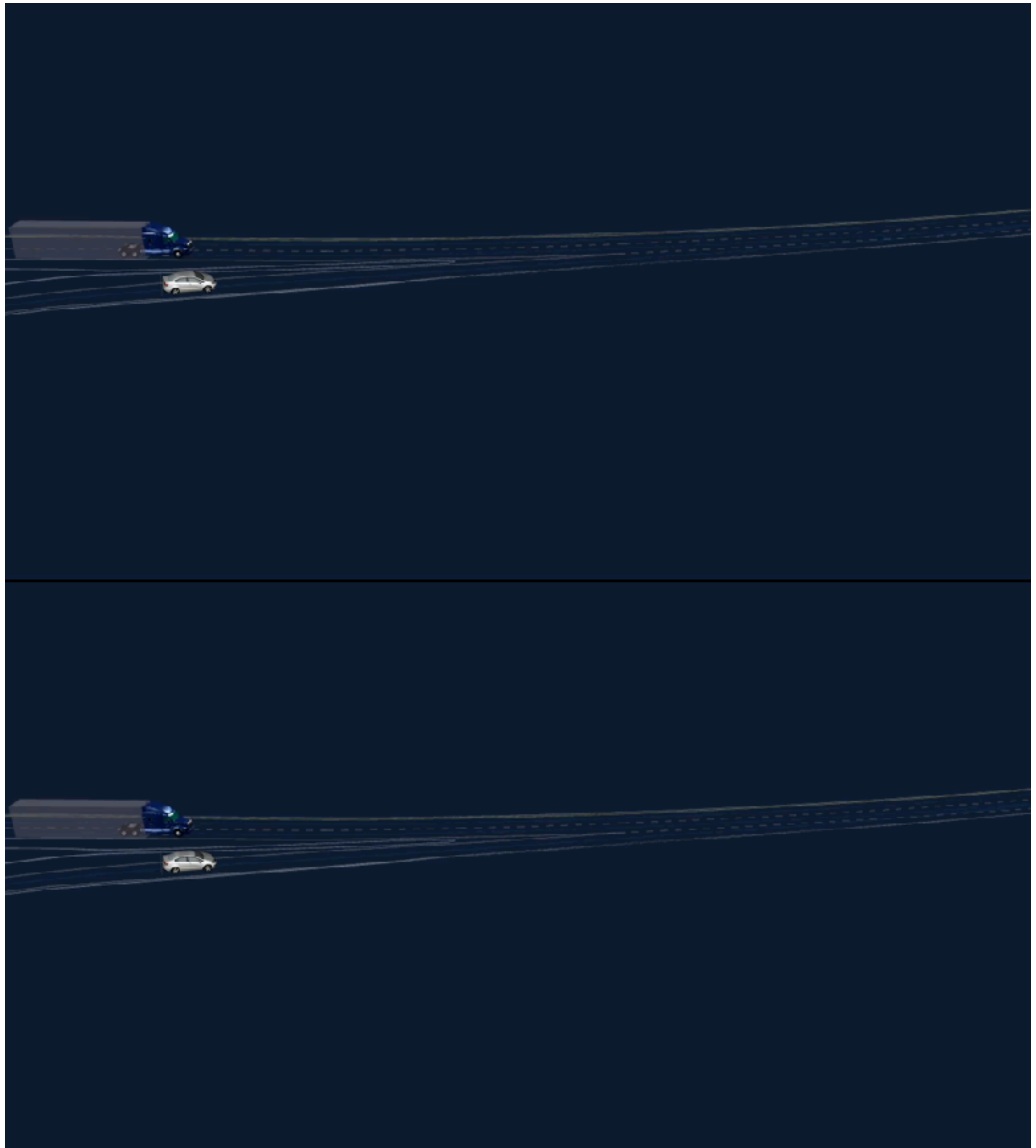
Marginalizing over multiple modes!

Mode A:
Robot merges
after



Mode B:
Robot merges
before



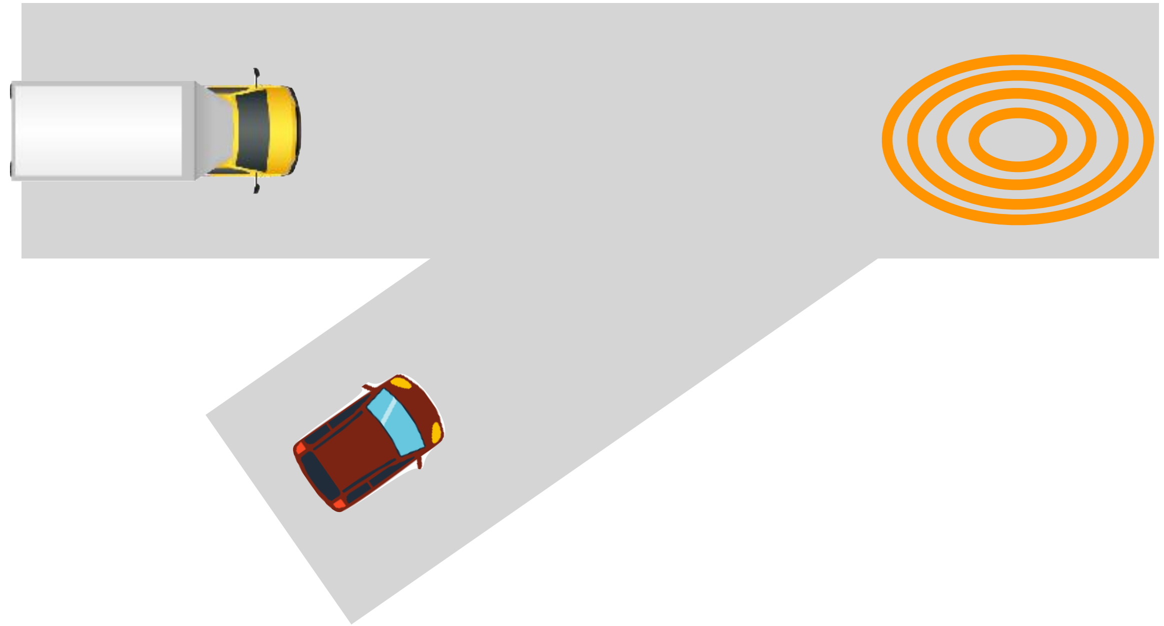


Okay .. so why can't we just predict multi-modal distributions?

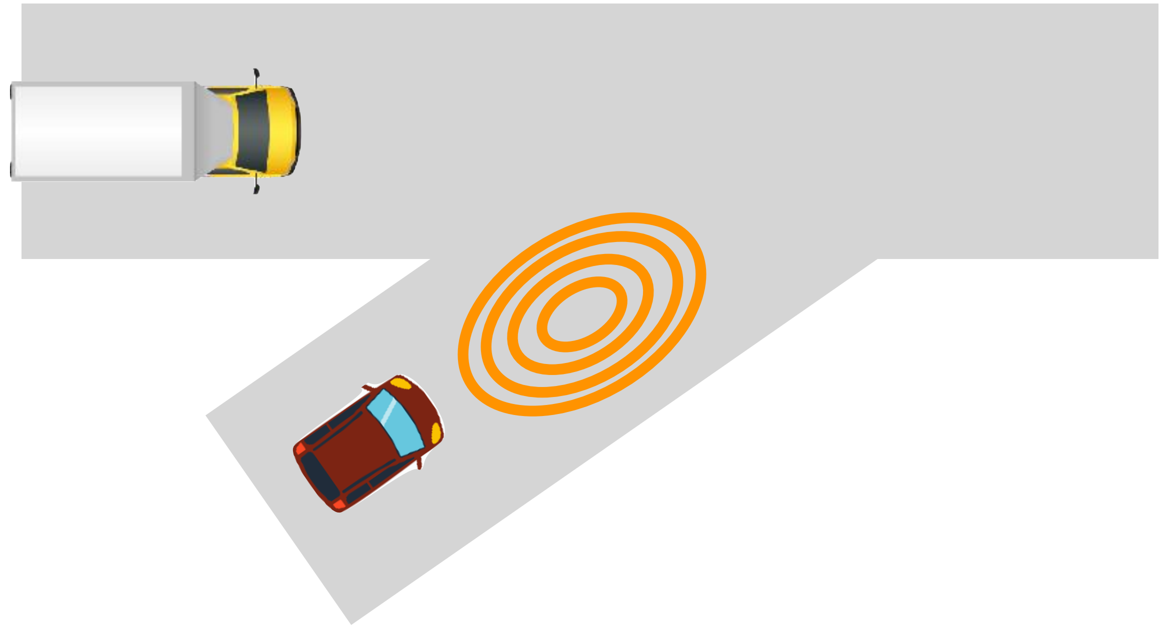
Multi-modal forecasts do not solve the issue!

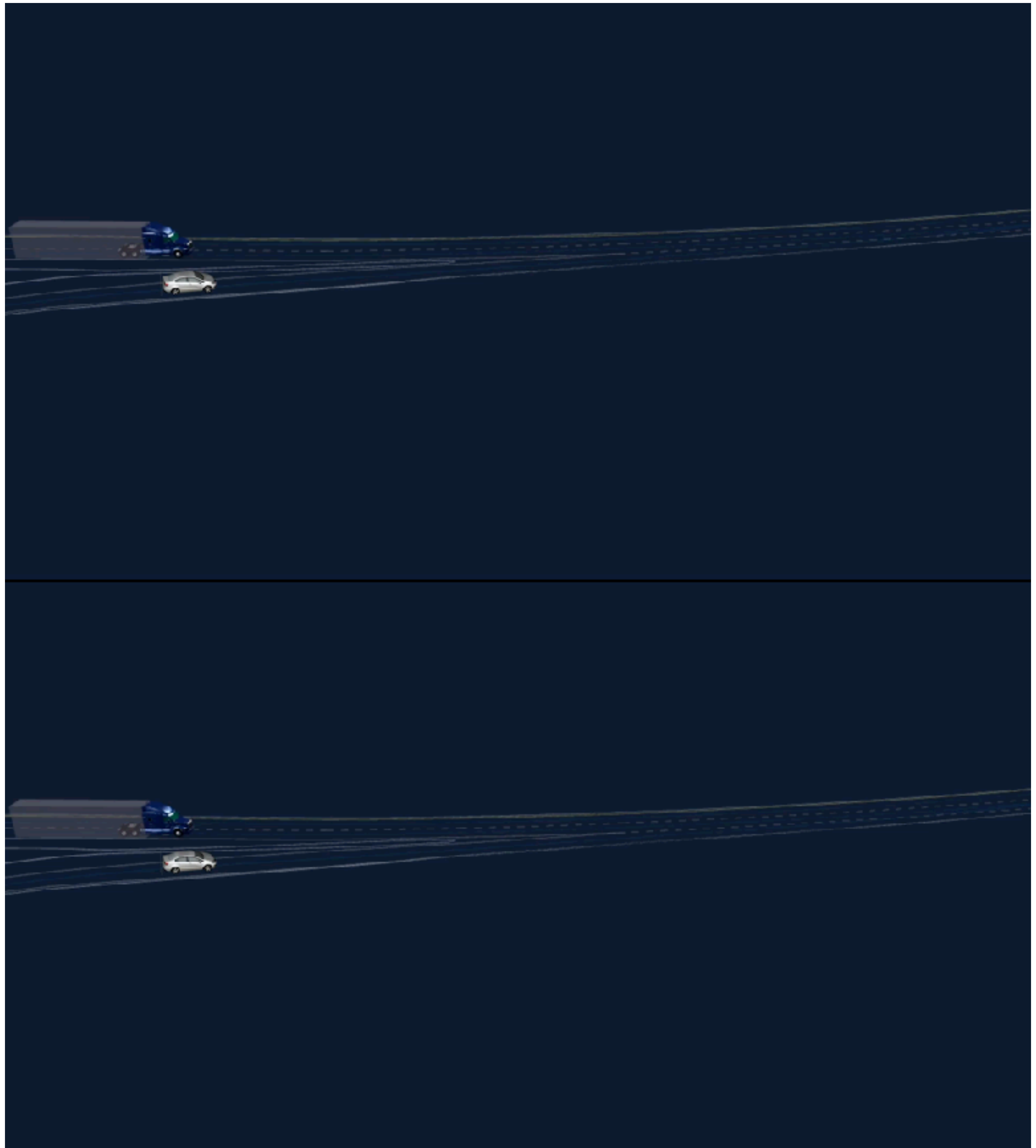
We are (incorrectly) telling the planner both modes can happen!

Mode A:
Robot merges after



Mode B:
Robot merges before

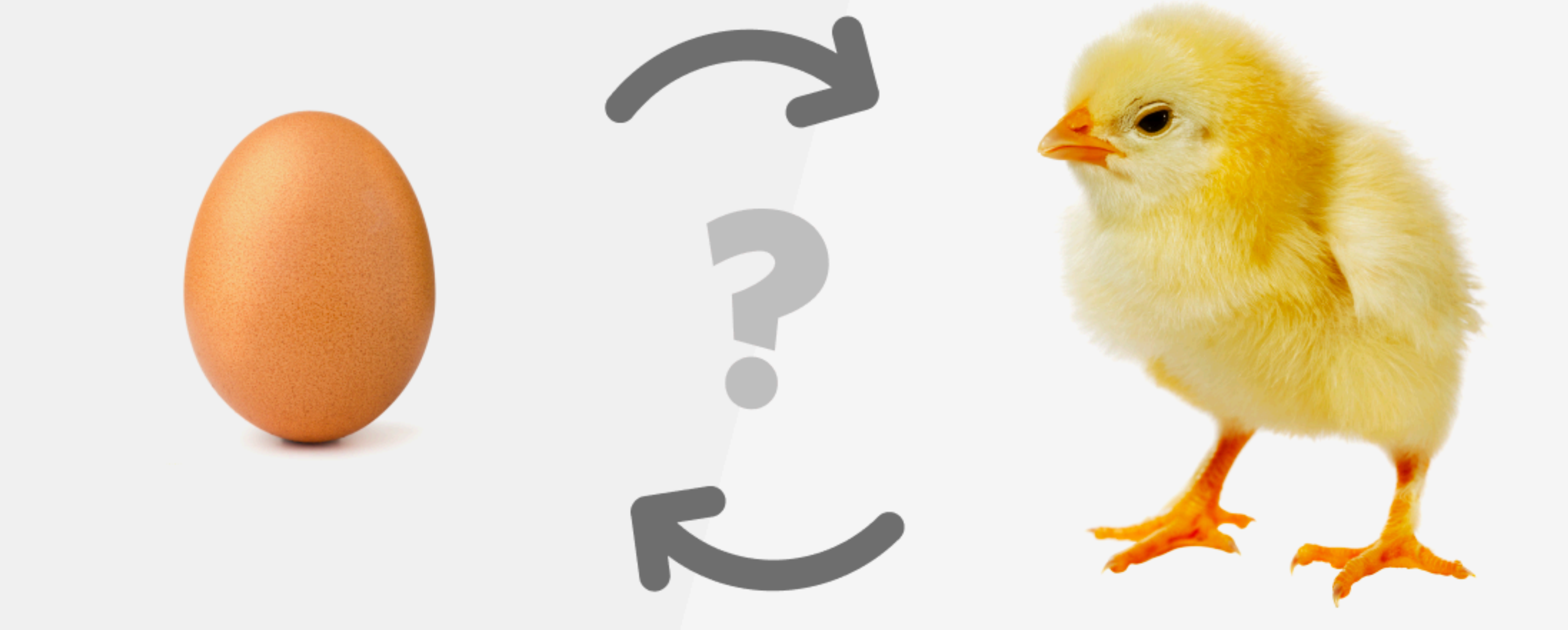




What robot does depends on other humans

What other humans do depends on the robot

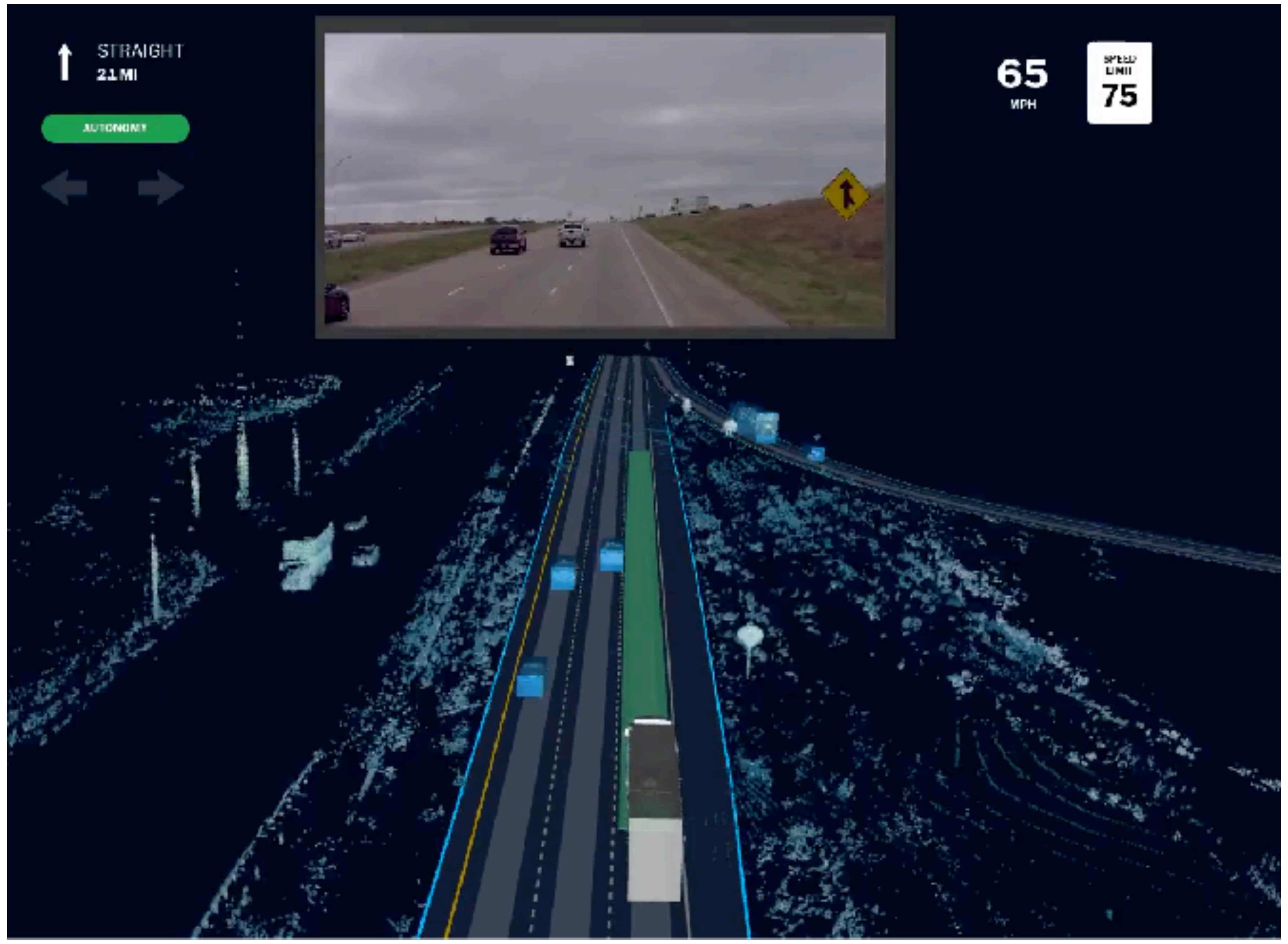
Forecasting-or-planning: a chicken-or-egg problem



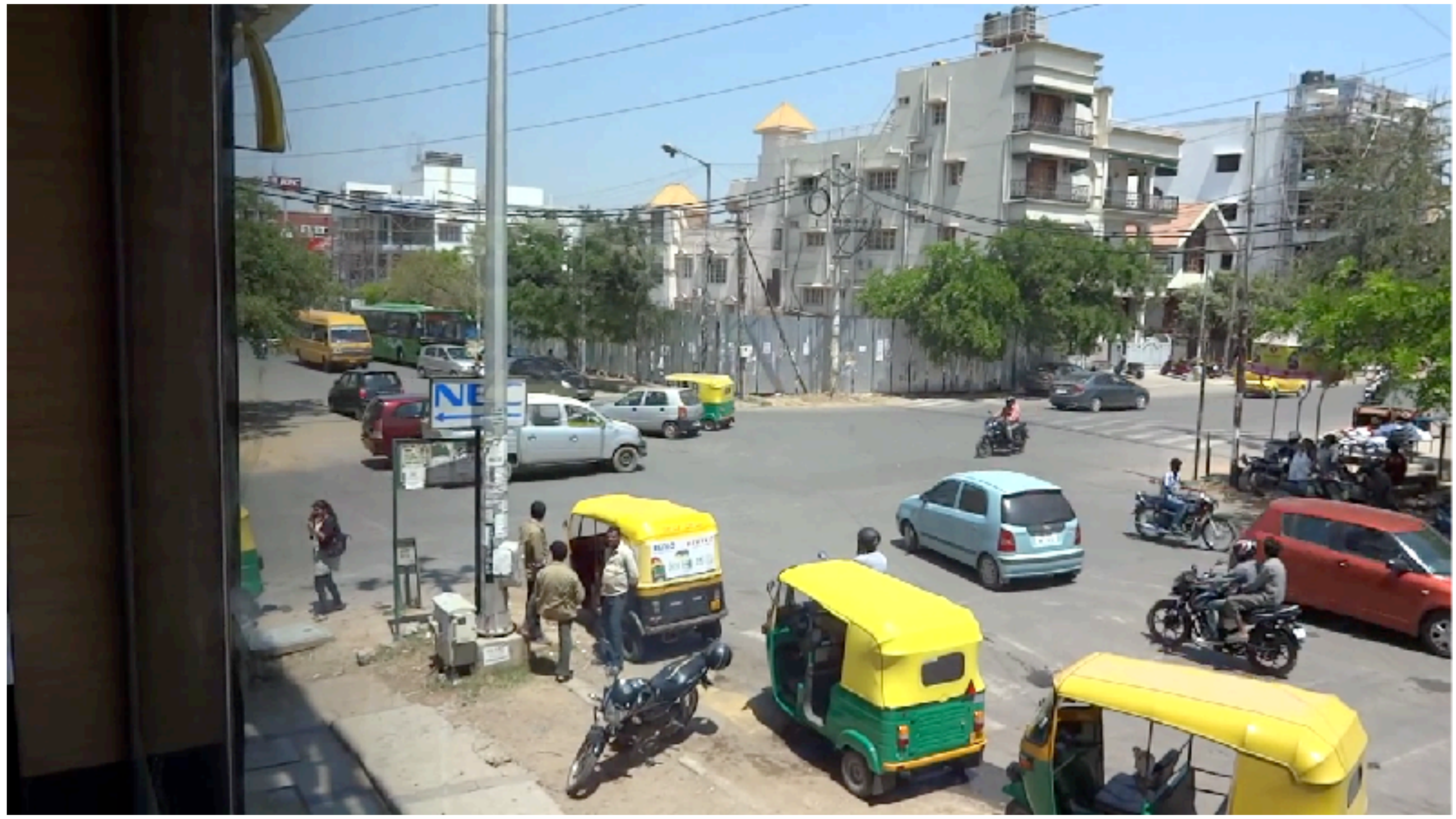
Why can't we just forecast the robot motion?

Planning is NOT merely forecasting

Suppose you collected data from this



vs data from this



Which data is useful for forecasting? For imitation learning?

Solving the chicken-or-egg problem

Train a *conditional* forecasting model

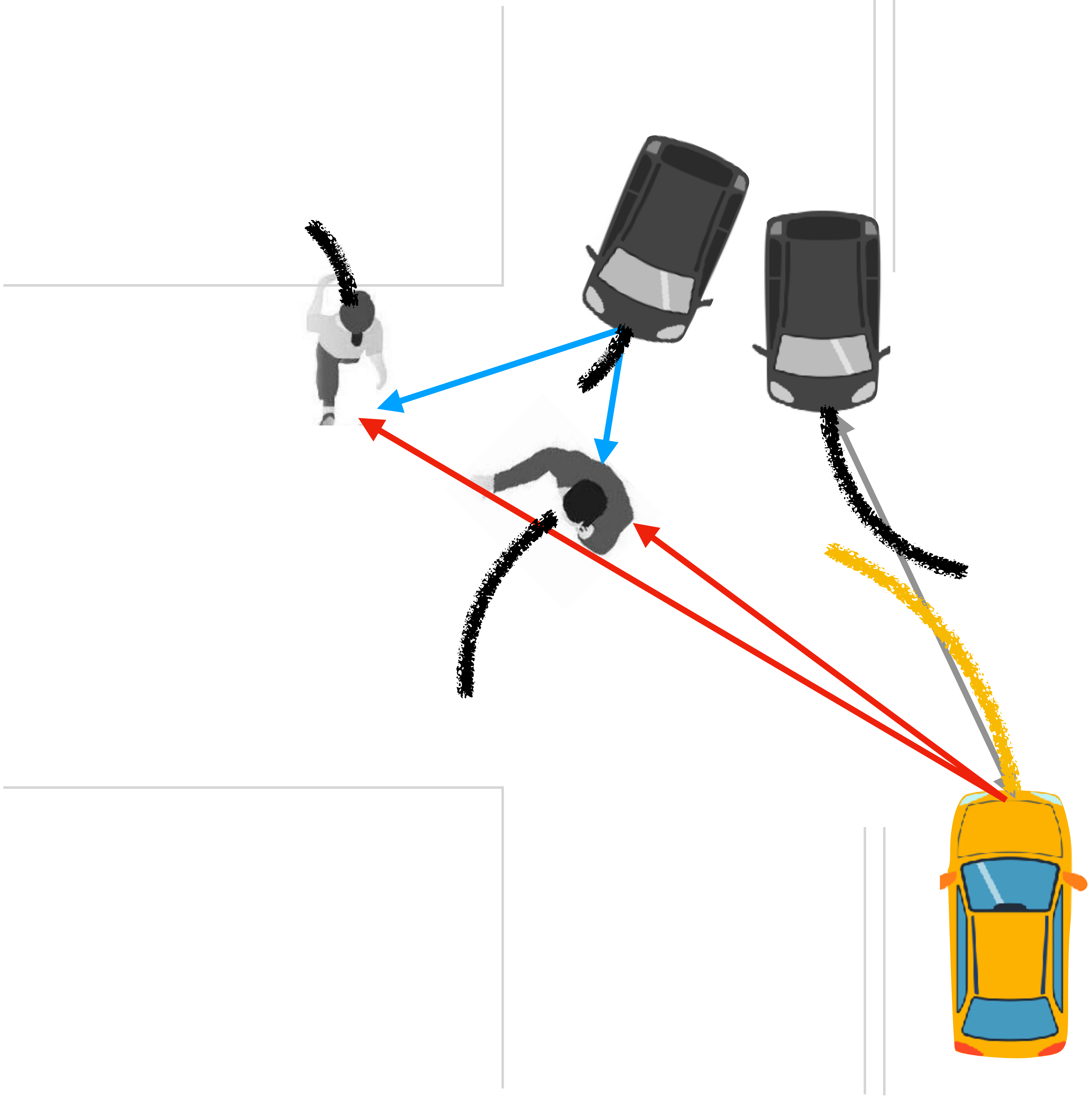
Normal forecasting

$P(s_{t:t+k} \mid s_{t:t-k})$

Conditional forecasting

$P(s_{t:t+k} \mid s_{t:t-k}, \xi_{plan})$

All actors in a scene influence each other



Robot is simply one actor among many in a scene

Need to jointly reason over all actors to produce forecasts

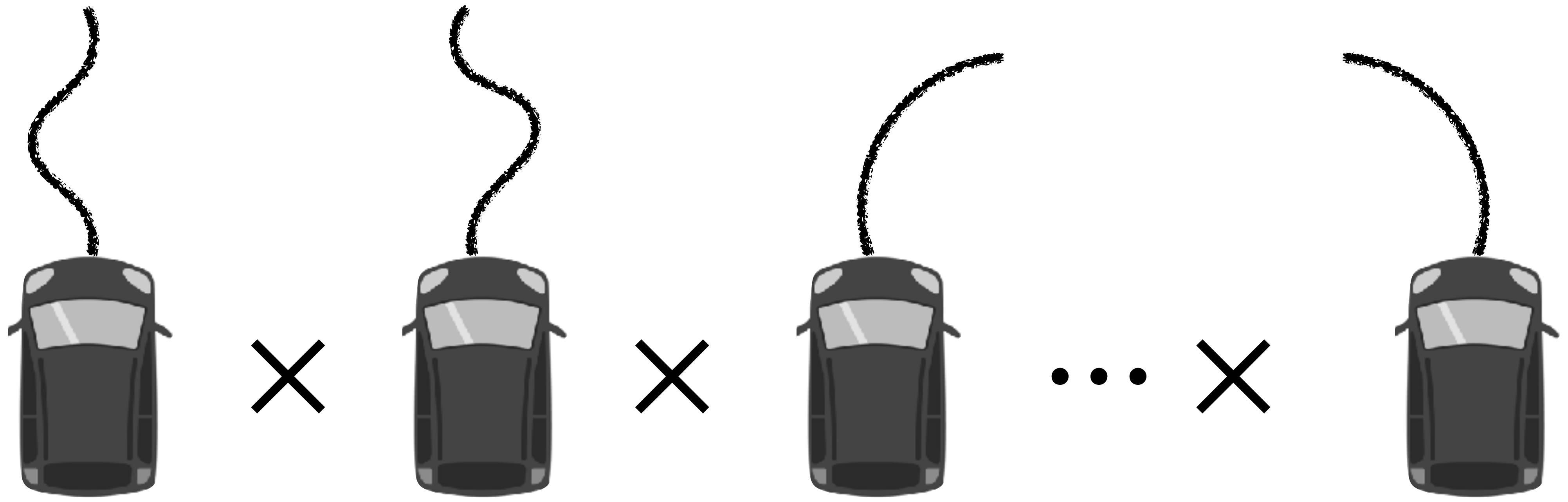
Problem: Space of joint trajectories is massive

©2021 I Aurora Proprietary

Continuous space of trajectories

+

Exponentially with in actors

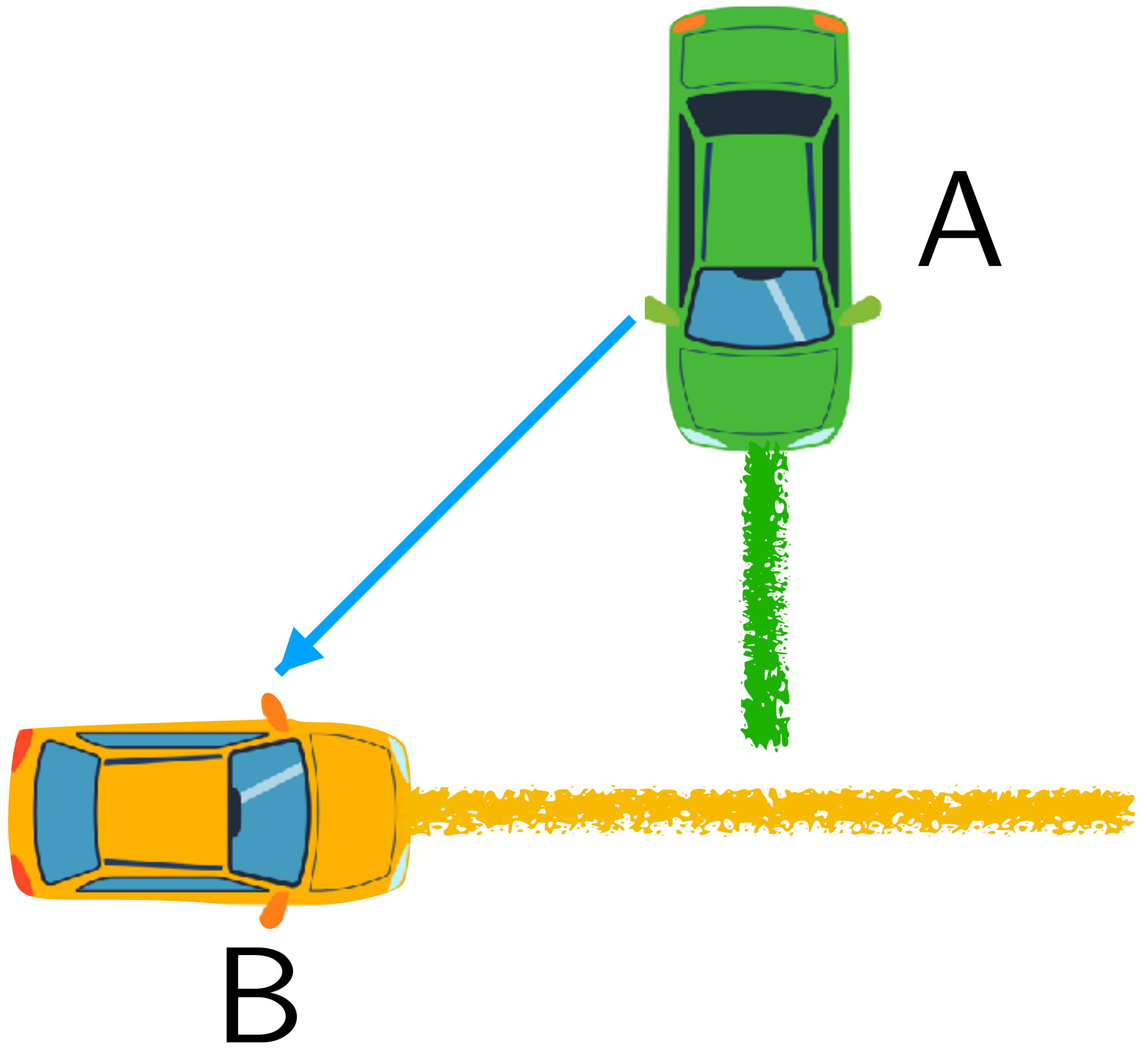


Conditional forecasting just makes this even harder

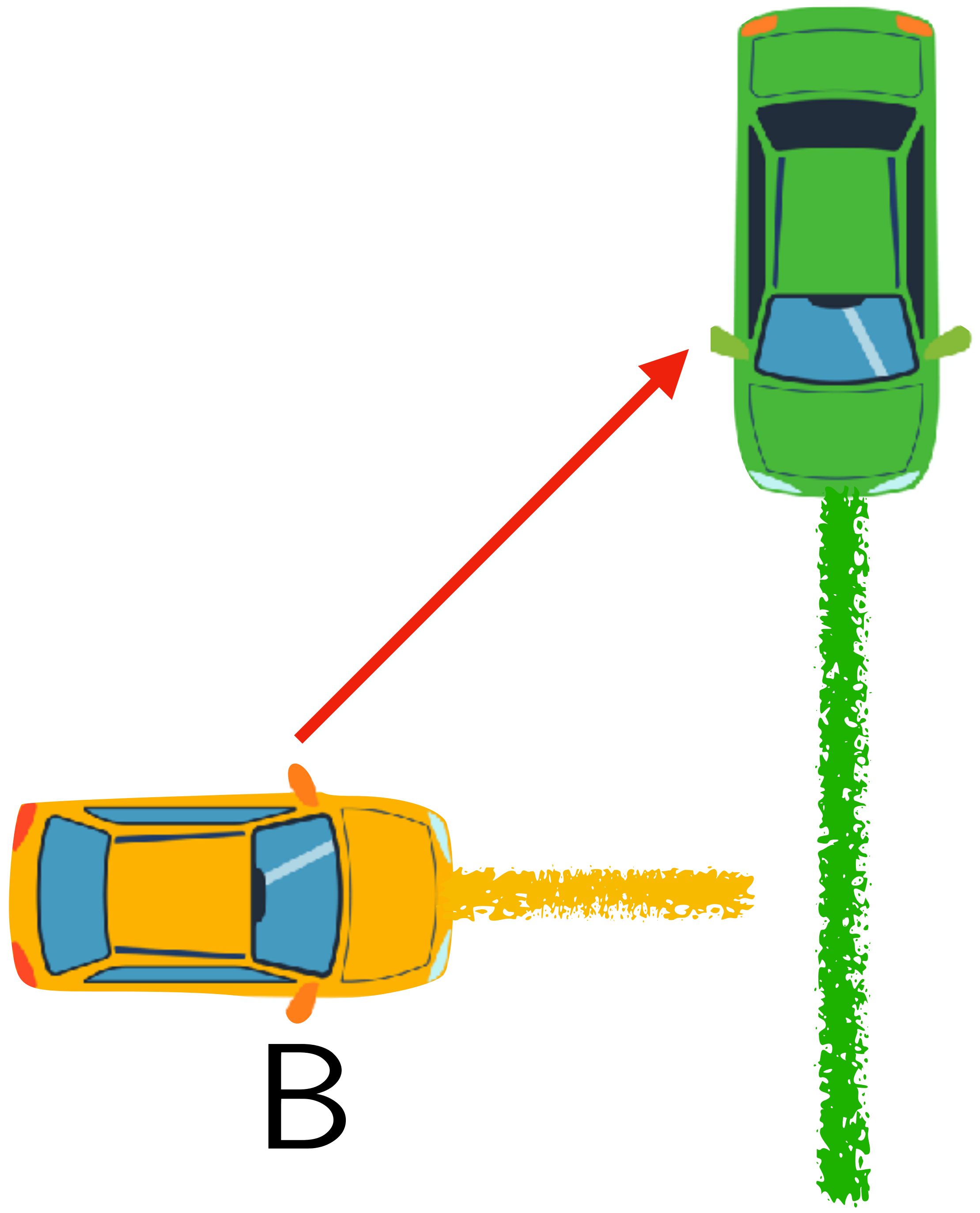
Reason in a space of discrete "modes"

3 fundamental modes of space-time paths

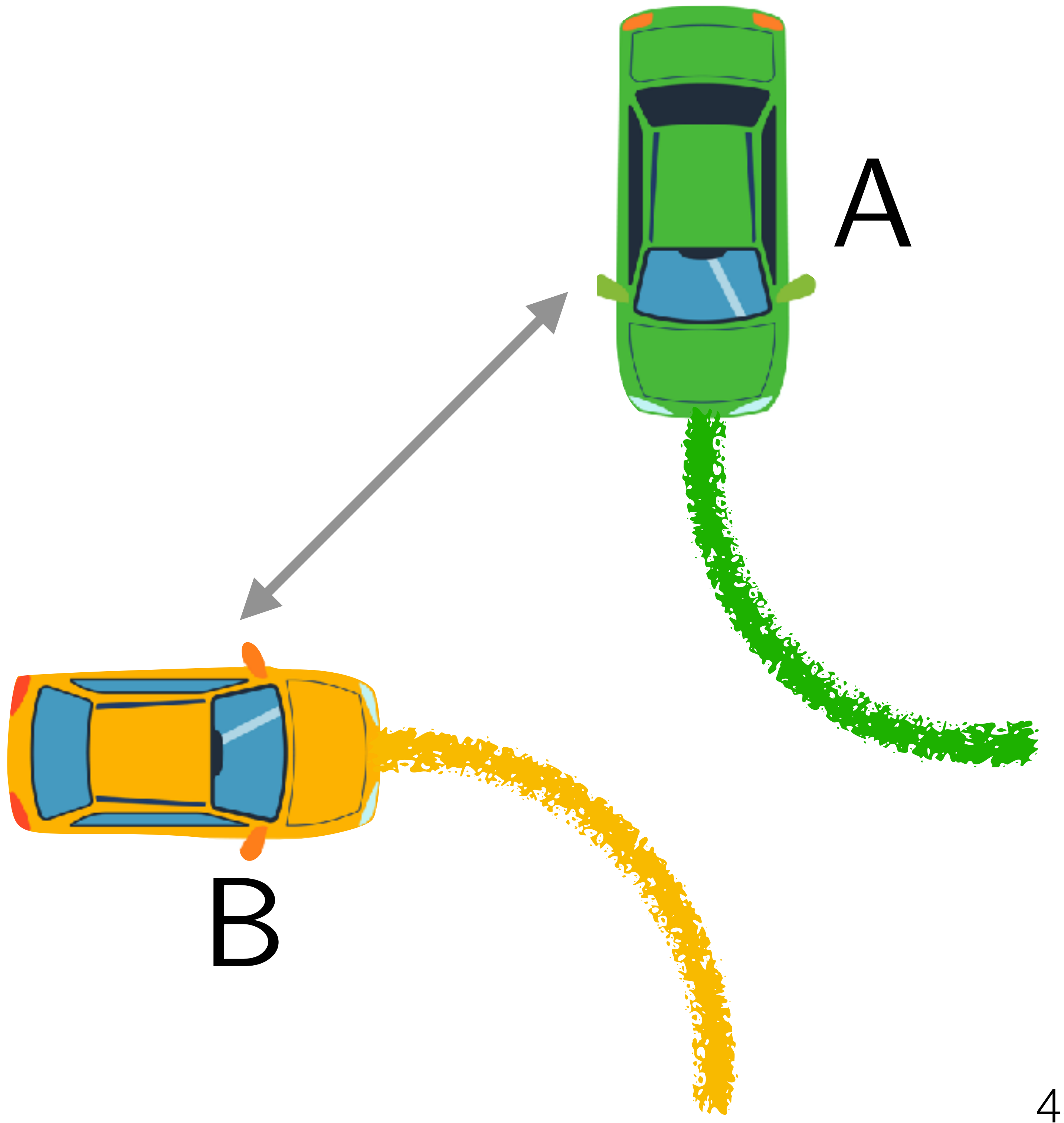
A Yields to B



B Yields to A



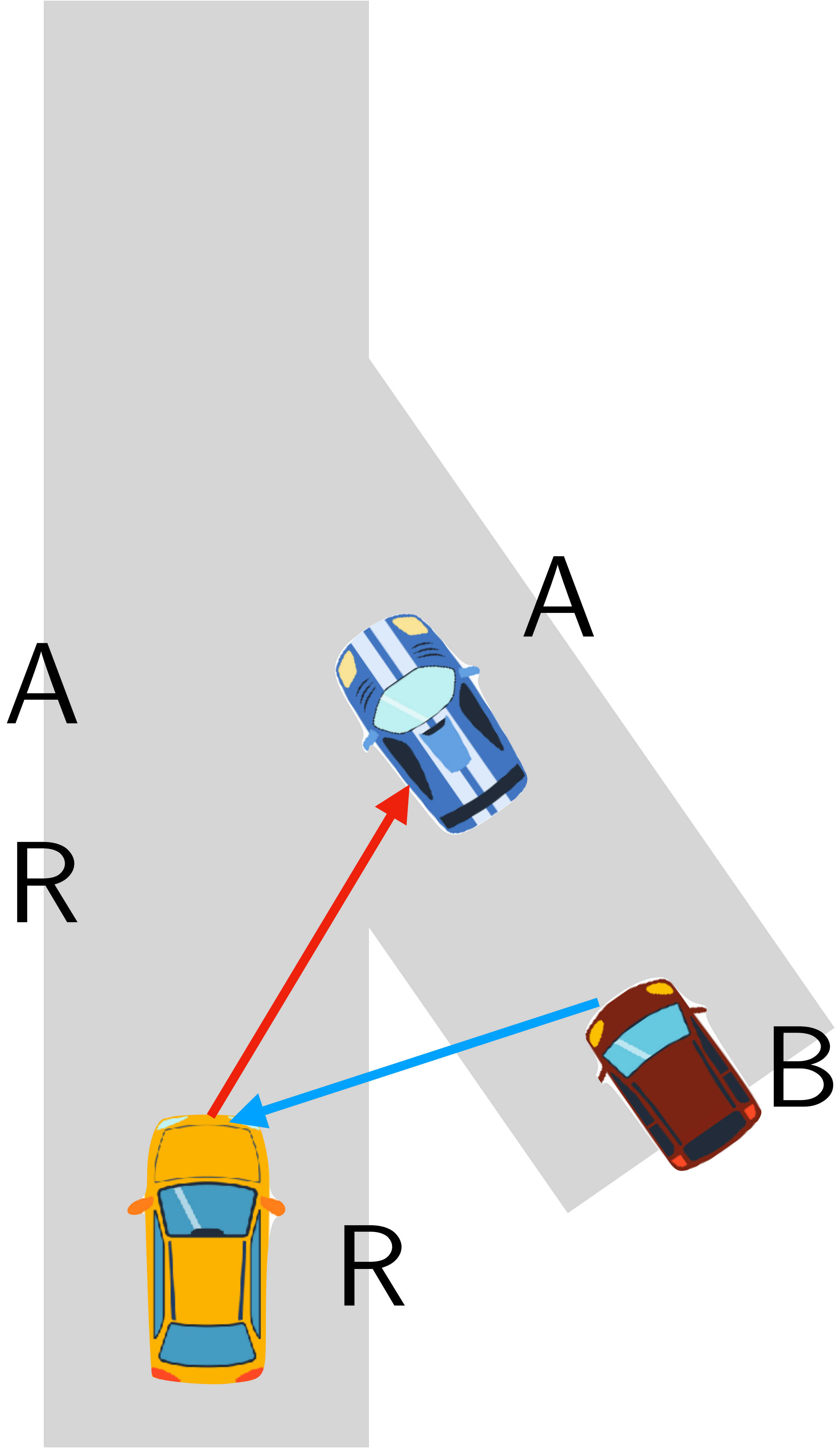
Not Yield



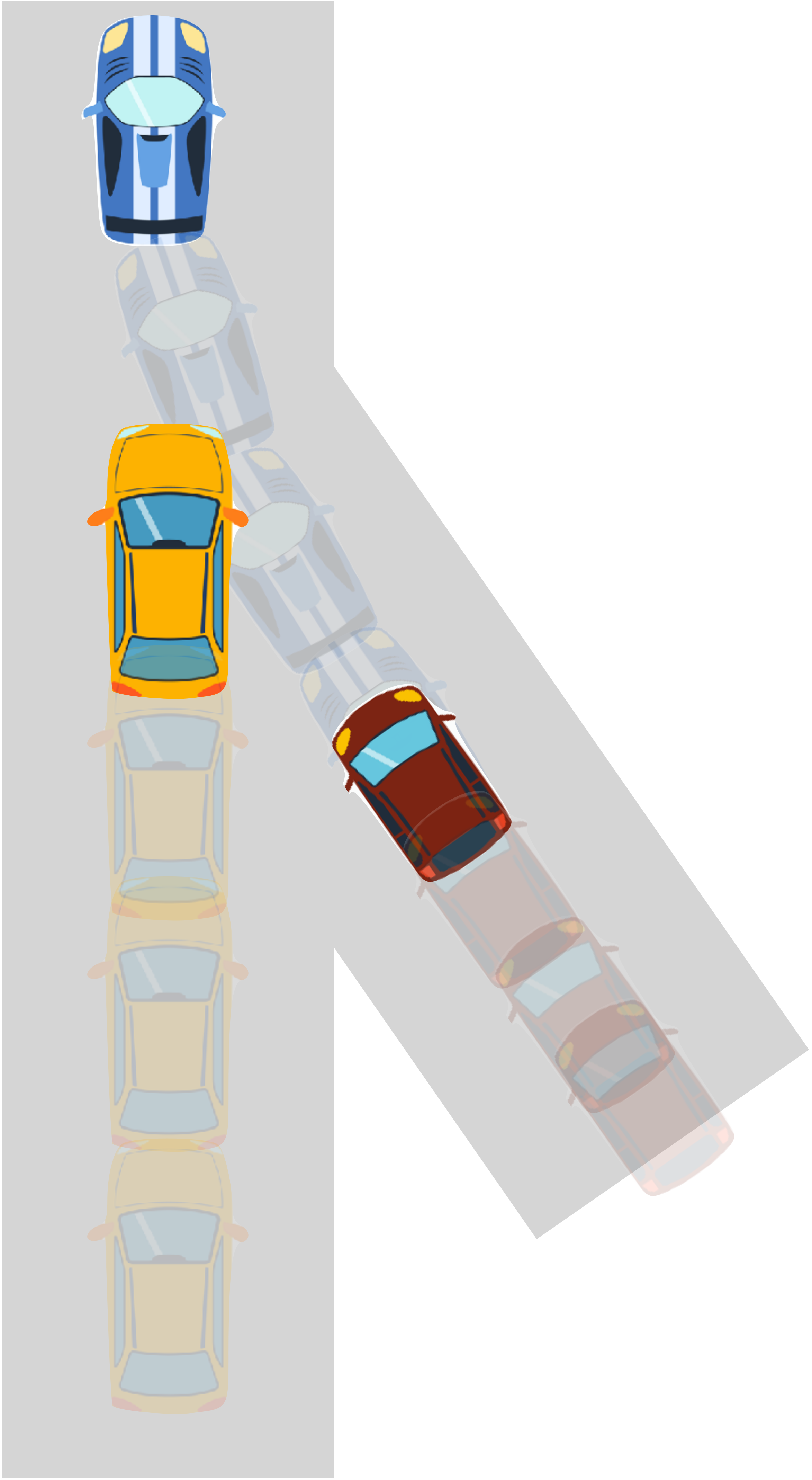
Mode ≡ A single basin of forecast

R Yields to A

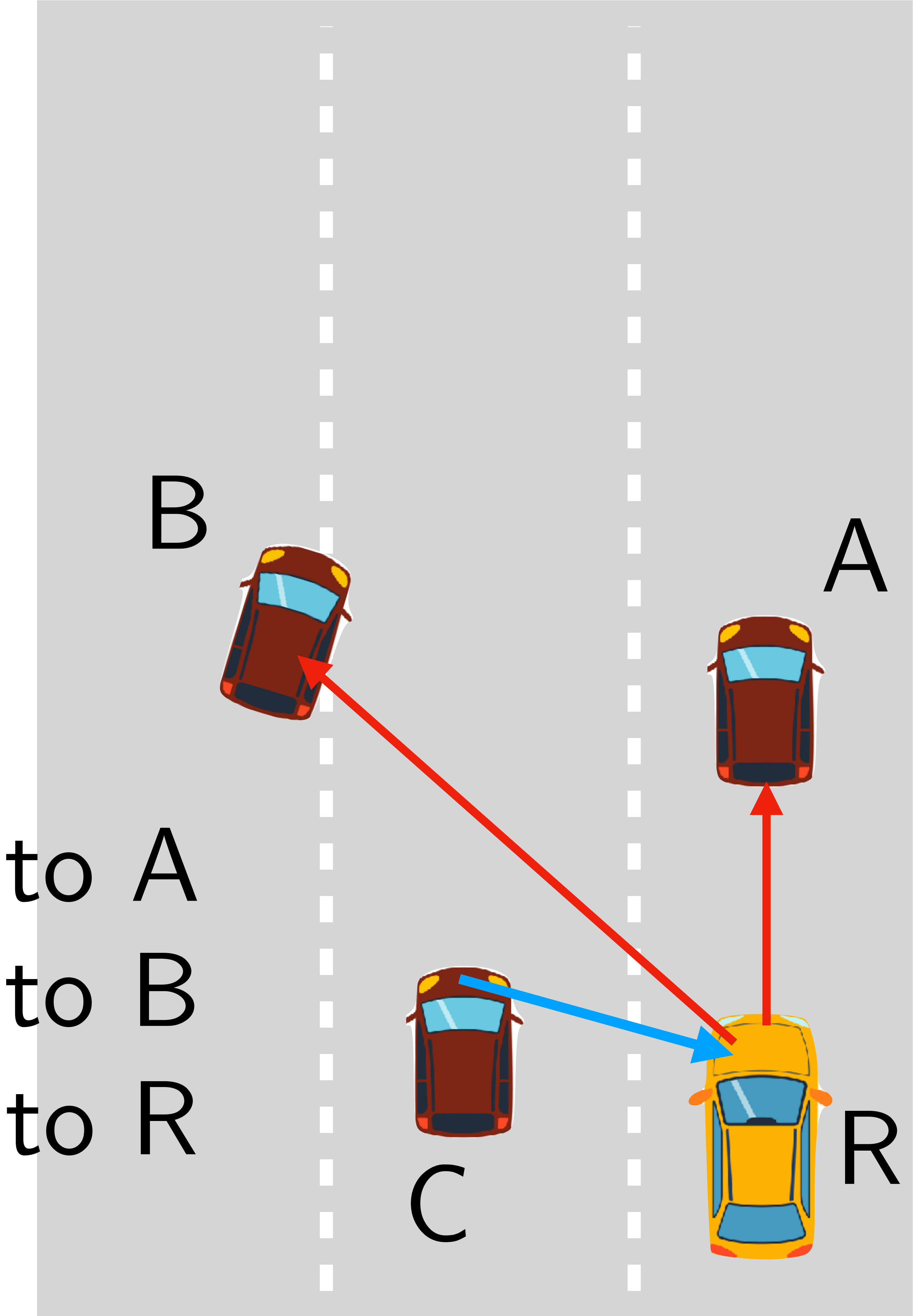
B Yields to R



≡

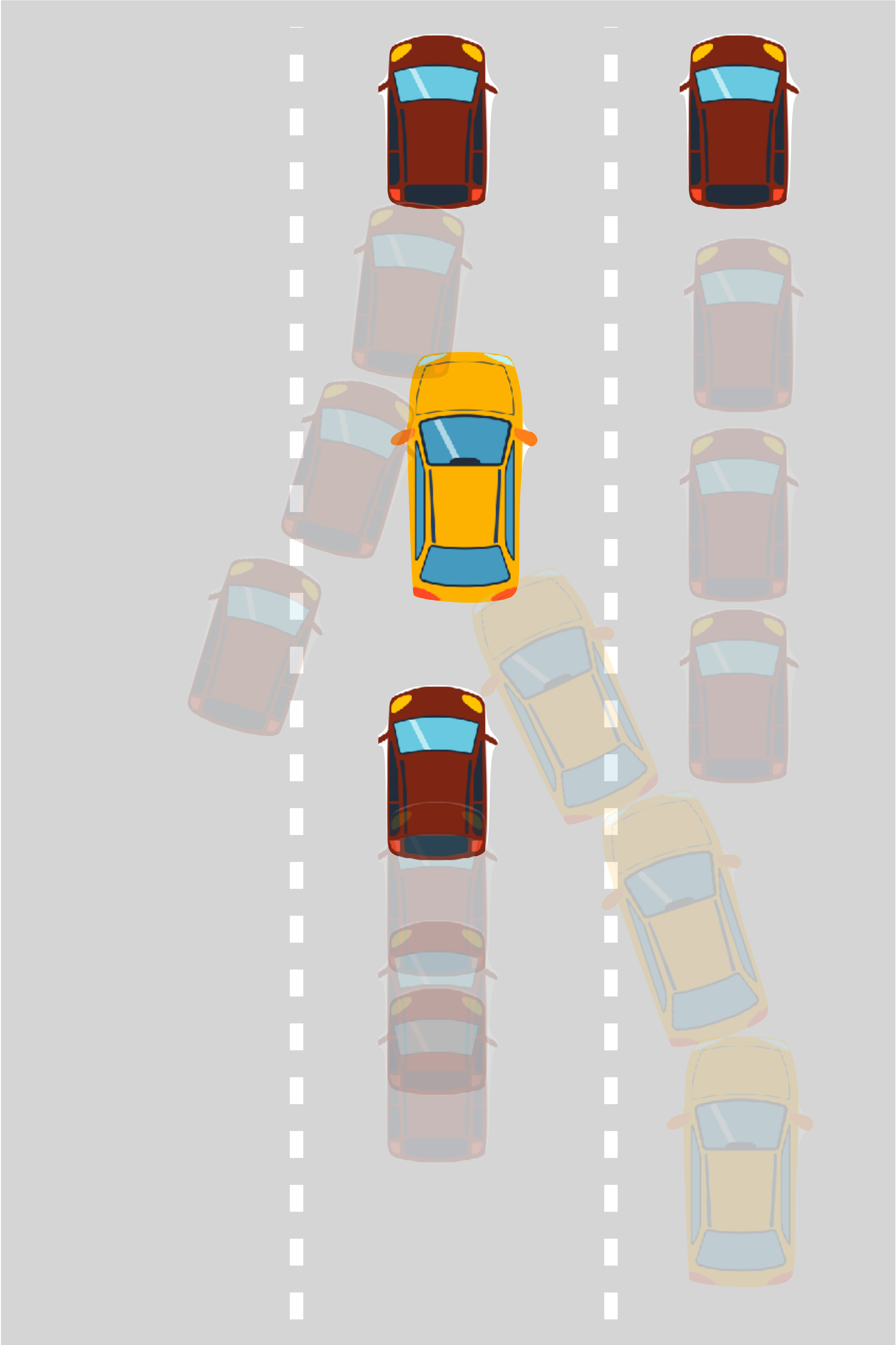


Mode ≡ A single basin of forecast

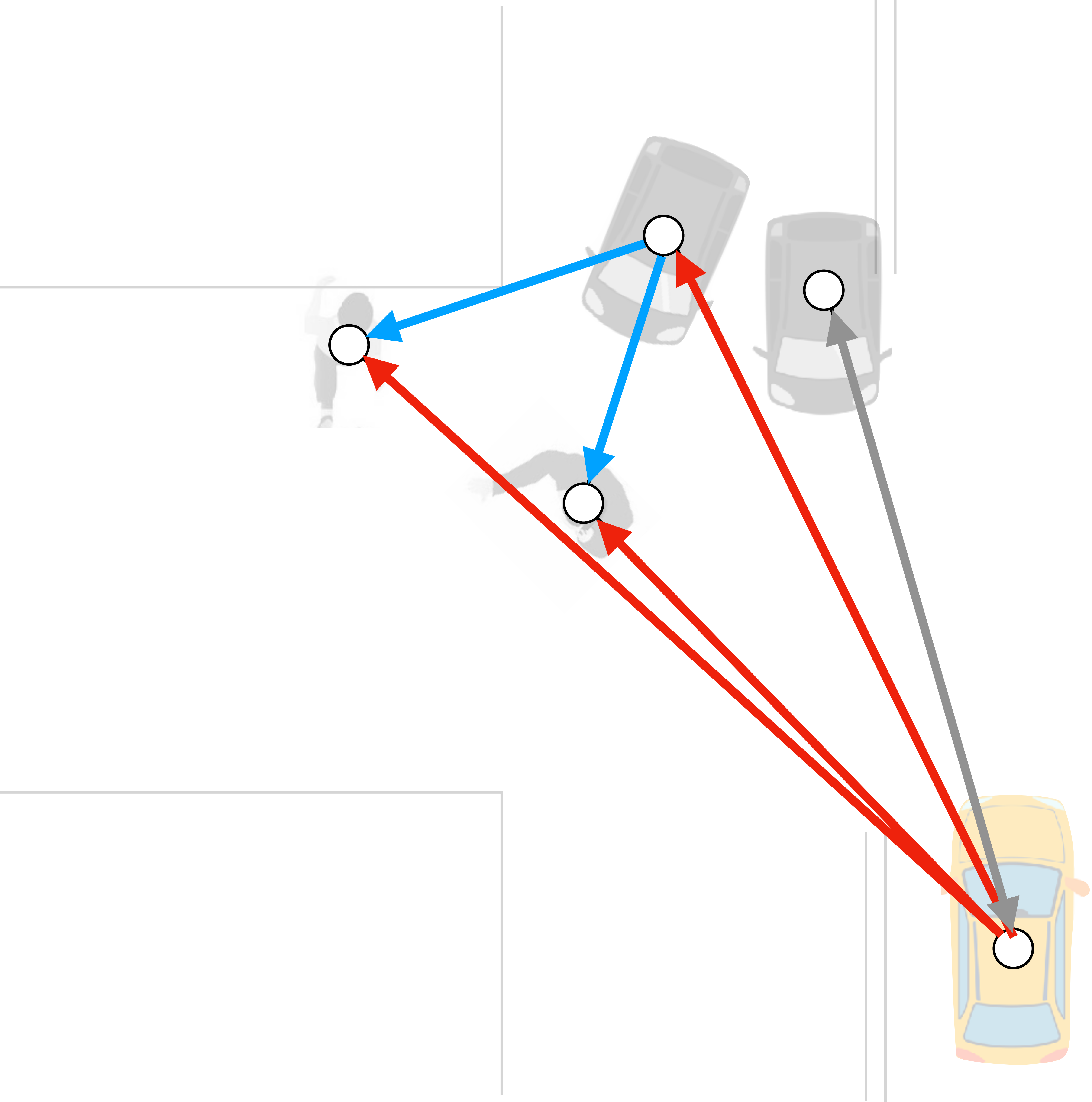


R Yields to A
R Yields to B
C Yields to R

≡



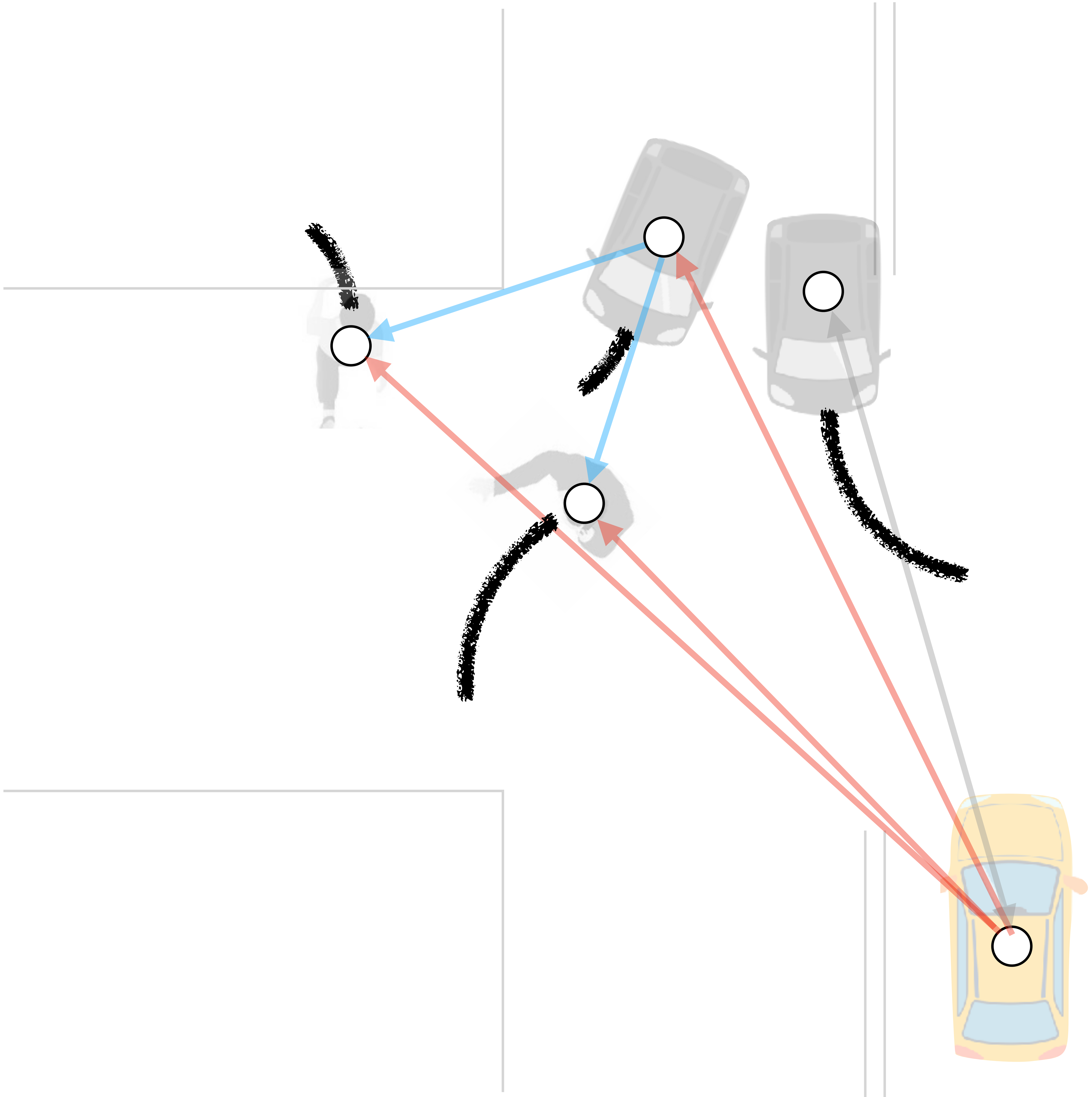
Message Passing on a Graph



Given a set of modes chosen by the robot

Infer what modes others are likely to choose

Message Passing on a Graph

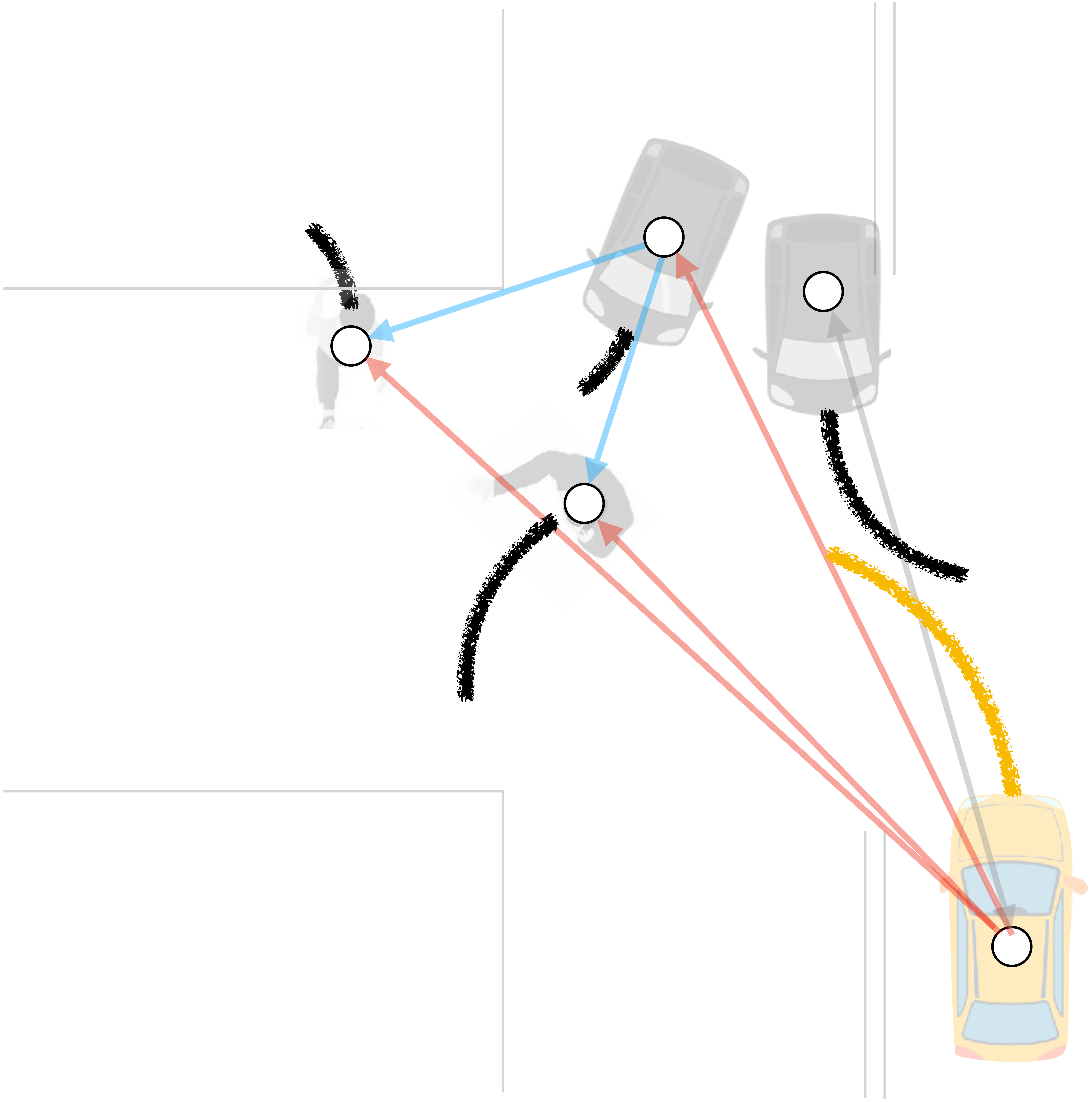


Given a set of modes chosen by the robot

Infer what modes others are likely to choose

Forecast actors given modes

Message Passing on a Graph



Given a set of modes chosen by the robot

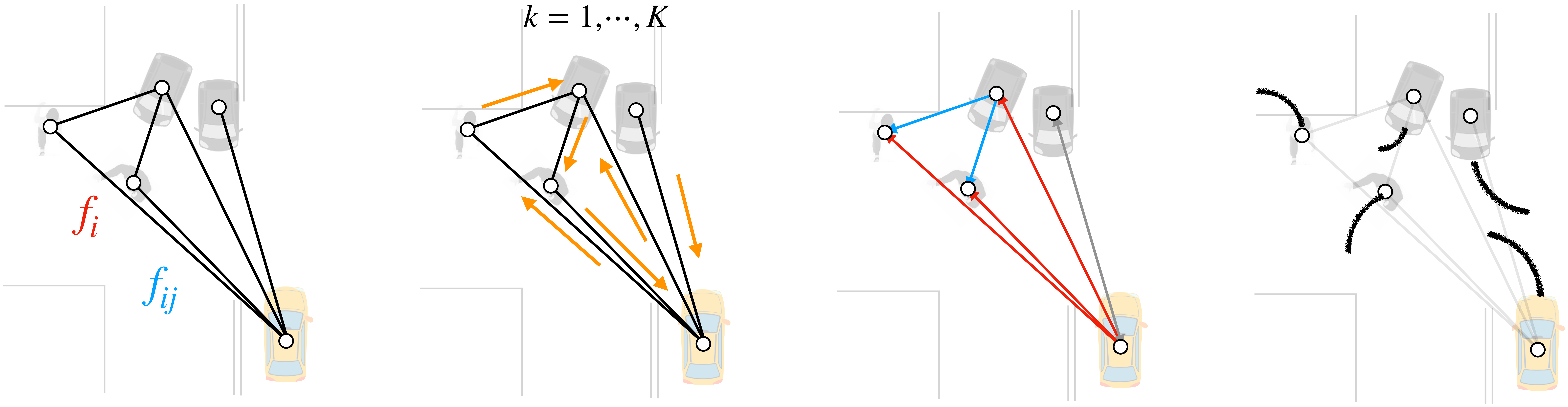
Infer what modes others are likely to choose

Forecast actors given modes

Plan given forecast

Geometric XformerNet

Builds on [Kumar et al. '20]



<u>Input</u>

Node features f_i

state+history of each actor in different path frames

Edge features f_{ij}

source actor state+history in destination actor frame

Encoder
+
K steps of message passing

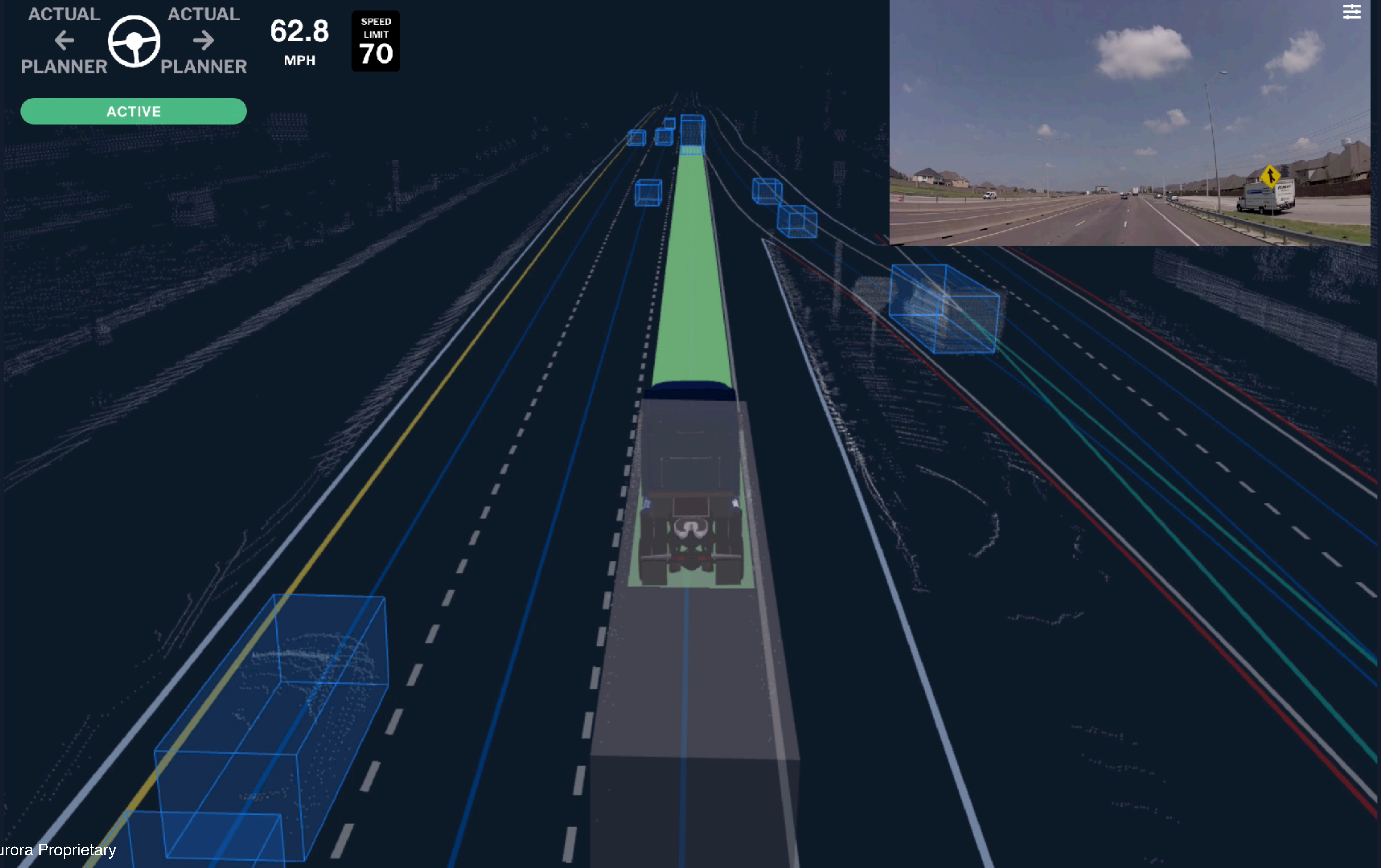
<u>Output</u>

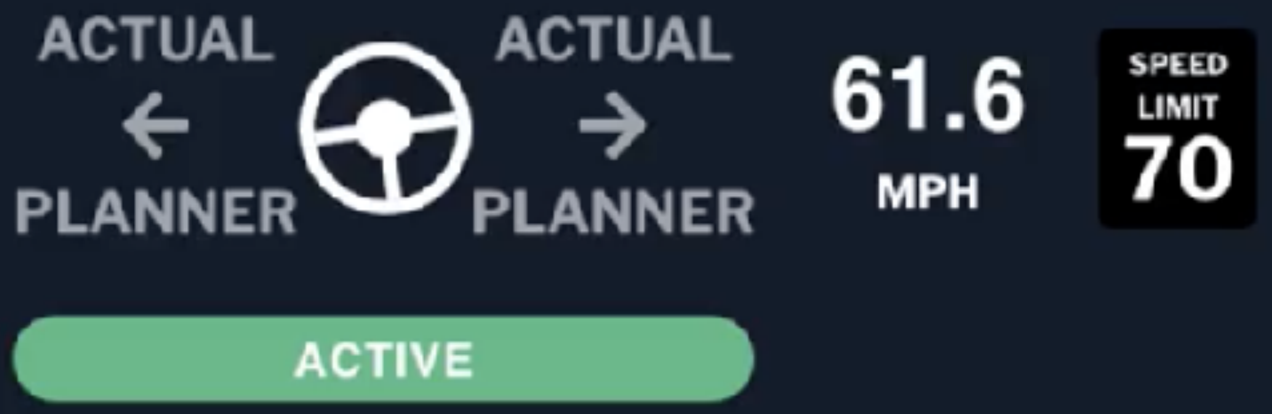
Edge output e_{ij}

Predict discrete modes

Node output n_i

Predict T-step trajectories

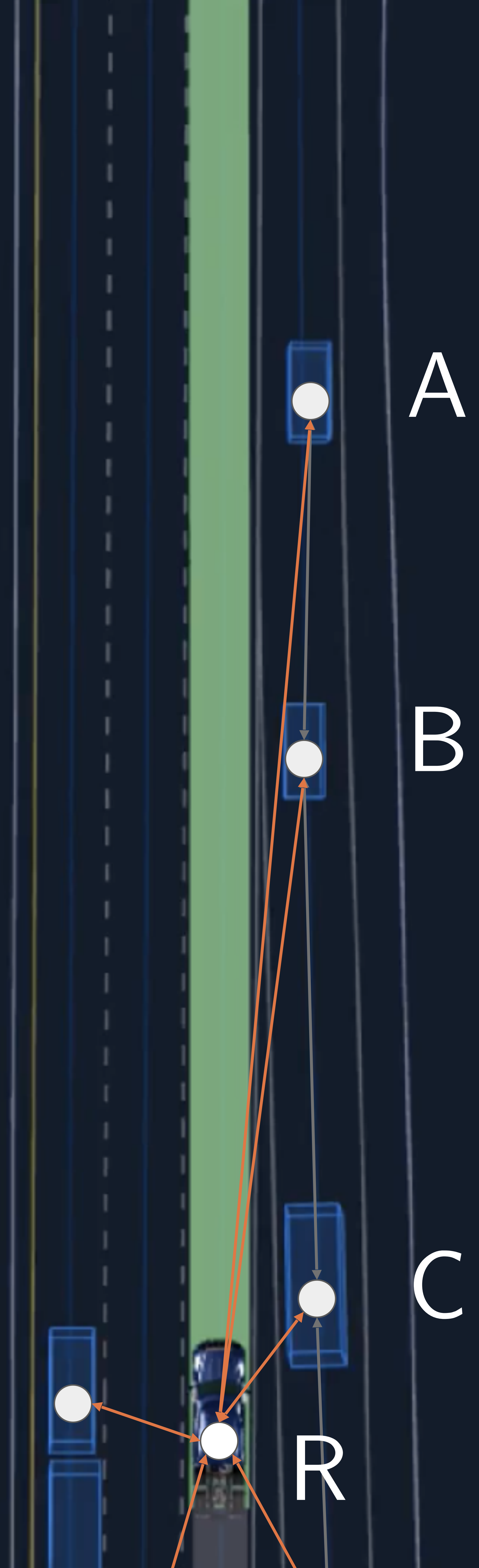

ACTUAL
←
PLANNER
ACTUAL
→
PLANNER
62.8
MPH
SPEED LIMIT
70
ACTIVE

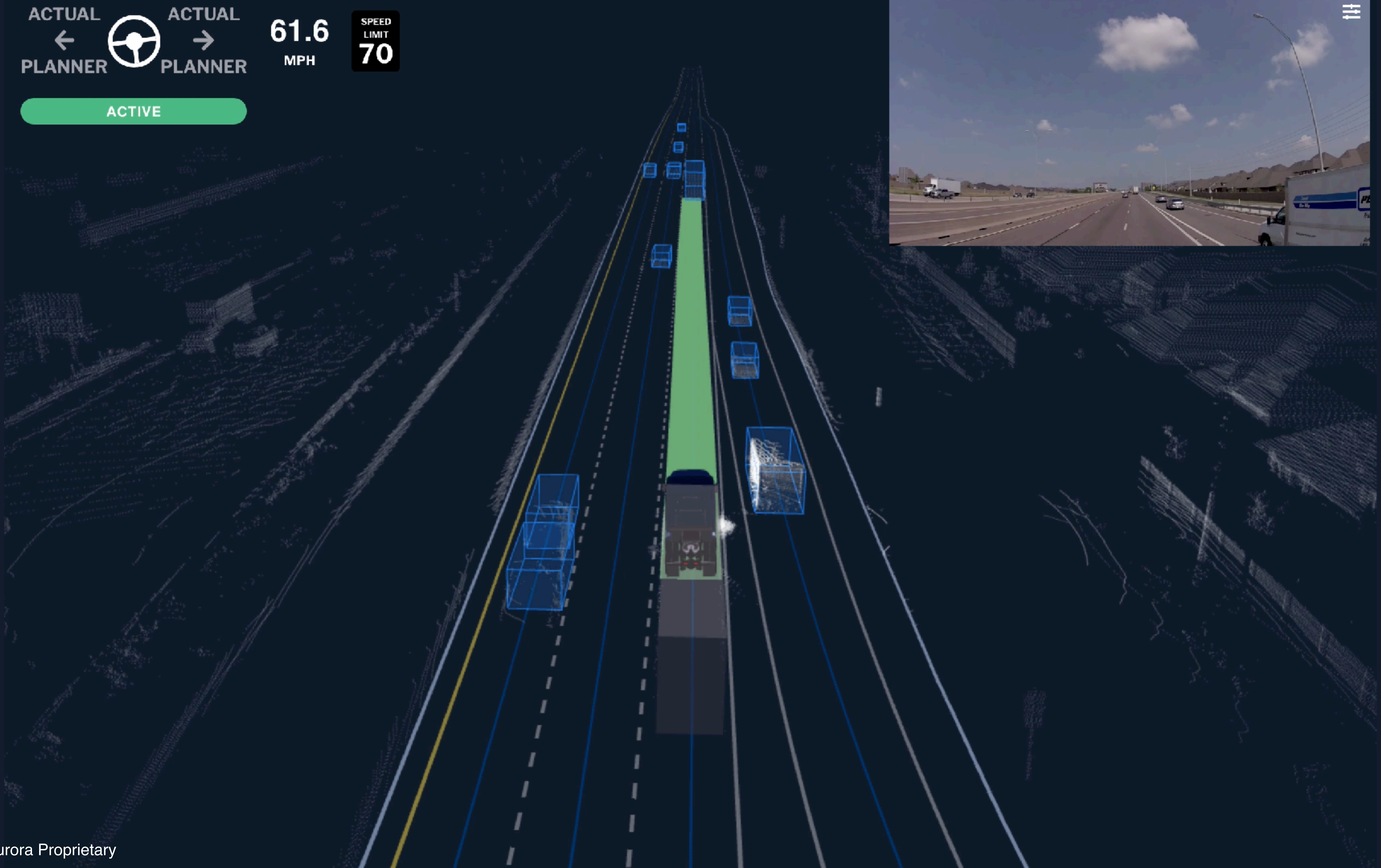


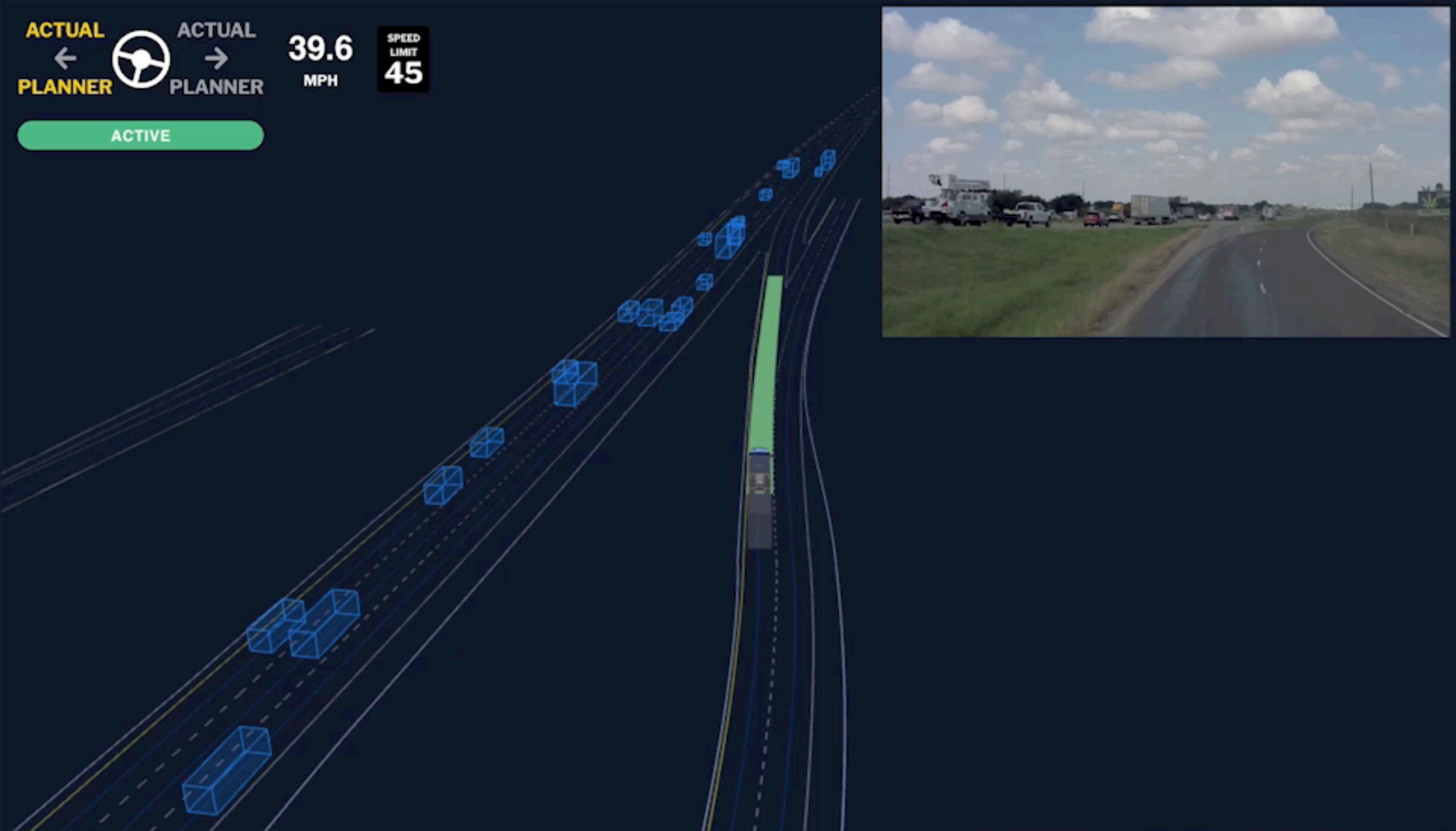
R Yields to A

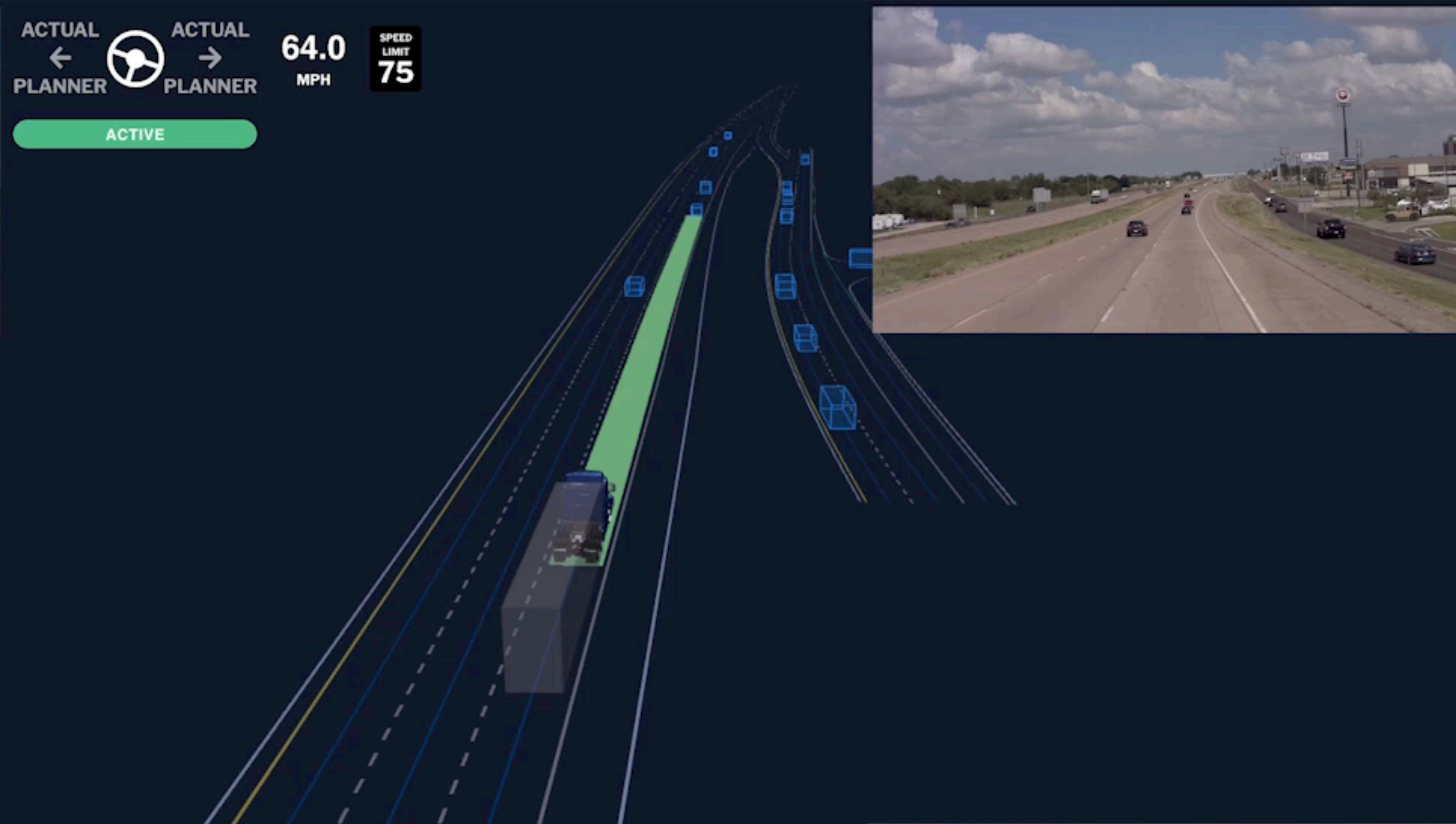
R Yields to B

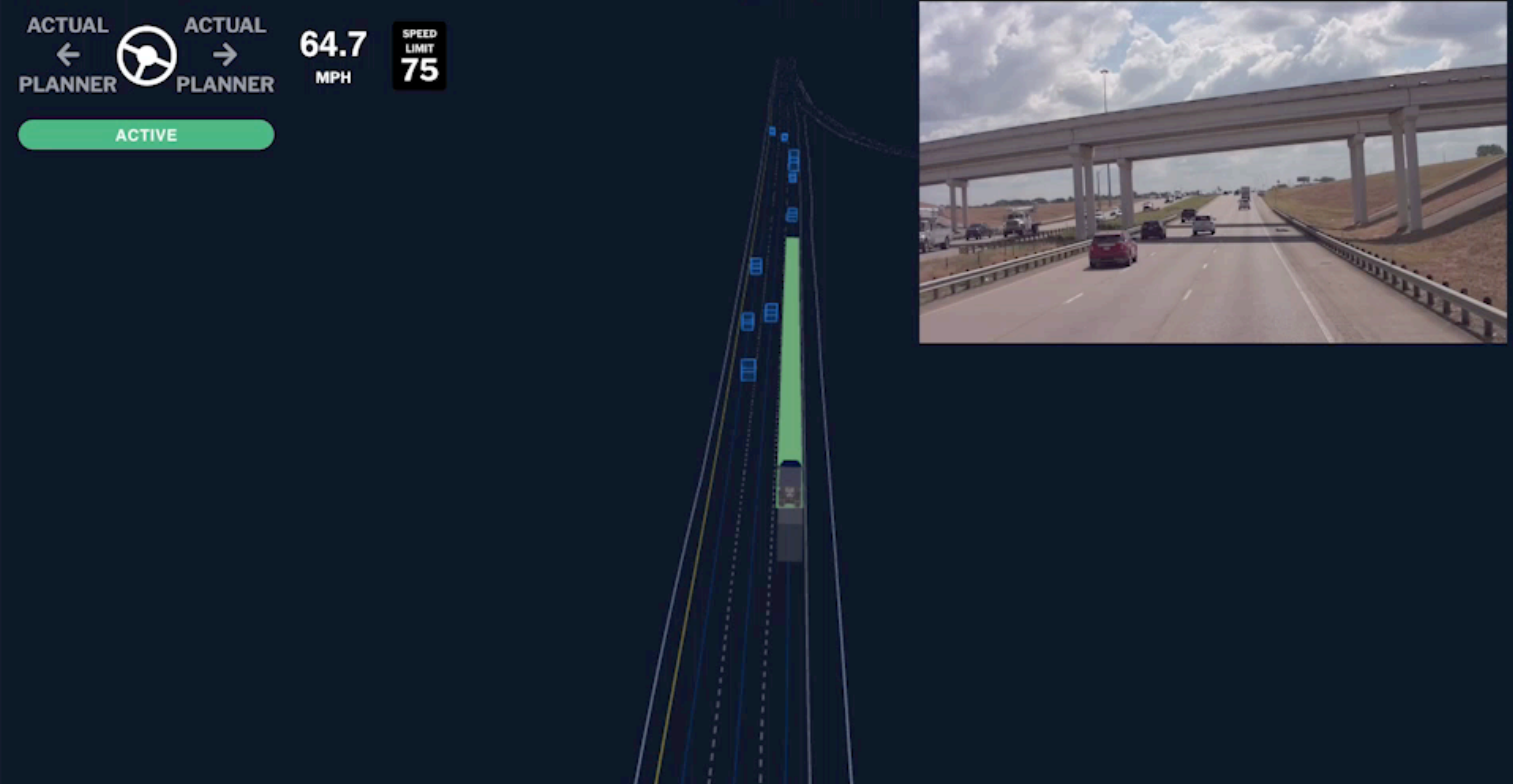
C Yields to R

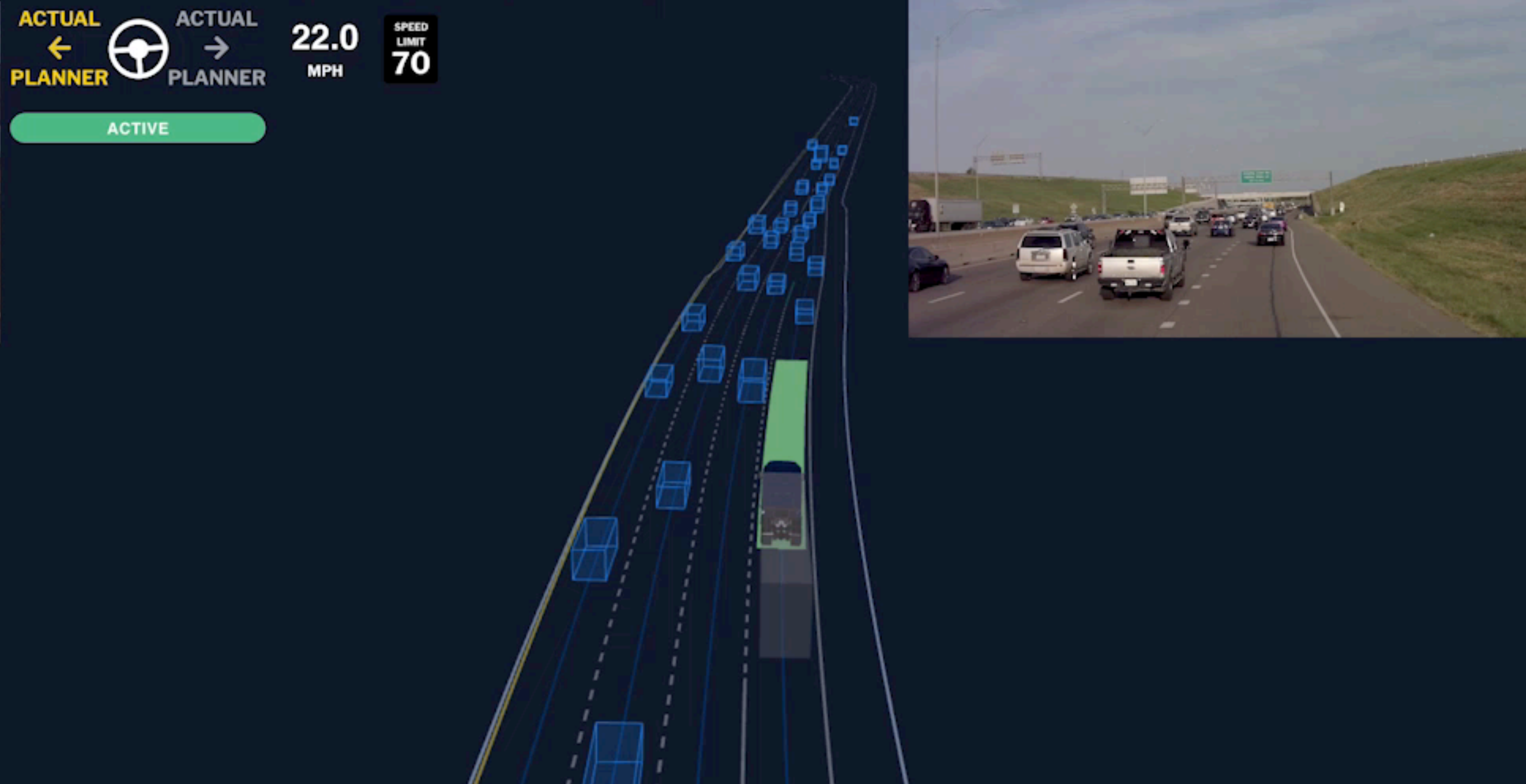



ACTUAL
PLANNER
ACTUAL
PLANNER
61.6
MPH
SPEED LIMIT
70
ACTIVE


ACTUAL
PLANNER
ACTUAL
PLANNER
39.6
MPH
SPEED LIMIT
45
ACTIVE


ACTUAL
PLANNER
ACTUAL
PLANNER
64.0
MPH
SPEED LIMIT
75
ACTIVE


ACTUAL
PLANNER
ACTUAL
PLANNER
64.7
MPH
SPEED LIMIT
75
ACTIVE


ACTUAL
PLANNER
ACTUAL
PLANNER
22.0
MPH
SPEED LIMIT
70
ACTIVE

Shaky foundations of forecasting

Are we using the right model?

Conditional forecasting

Are we collecting data correctly?

Are we using the right loss?

What happens when we deploy model?

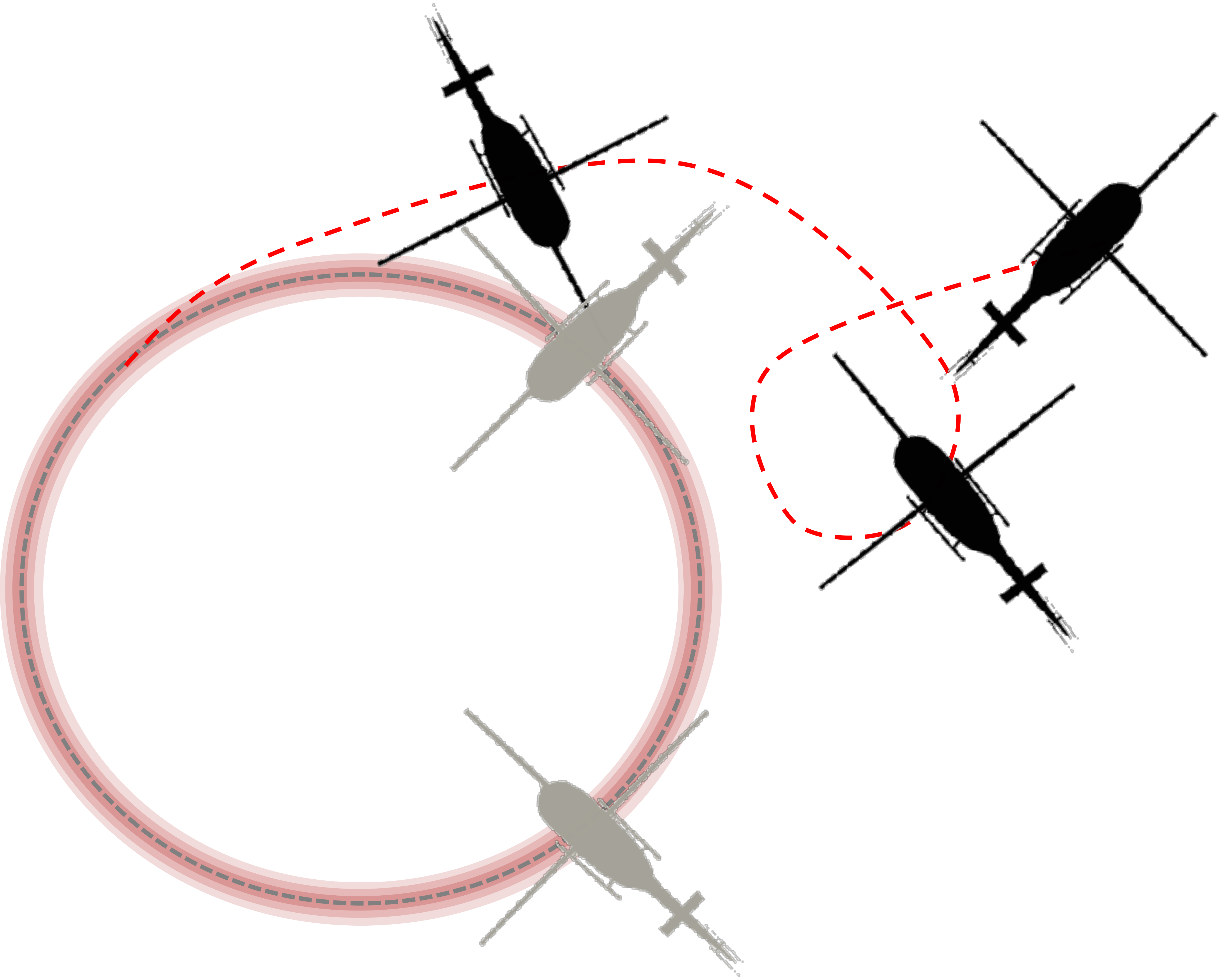
"The car will probably merge ahead, so I can slow down very smoothly ..."

"What the heck does this truck want to do, go ahead or behind ?!?!"

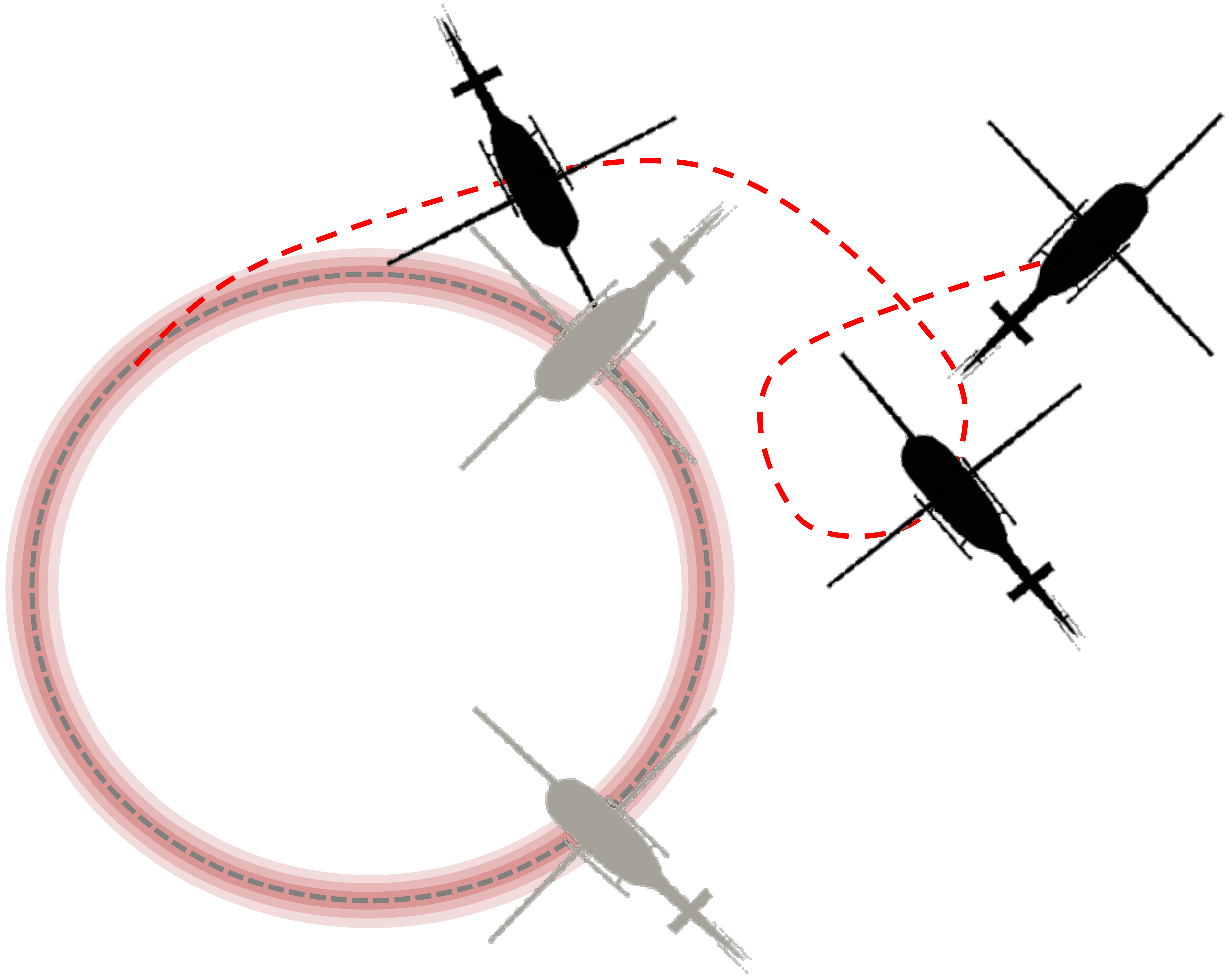
"?!@#!@"

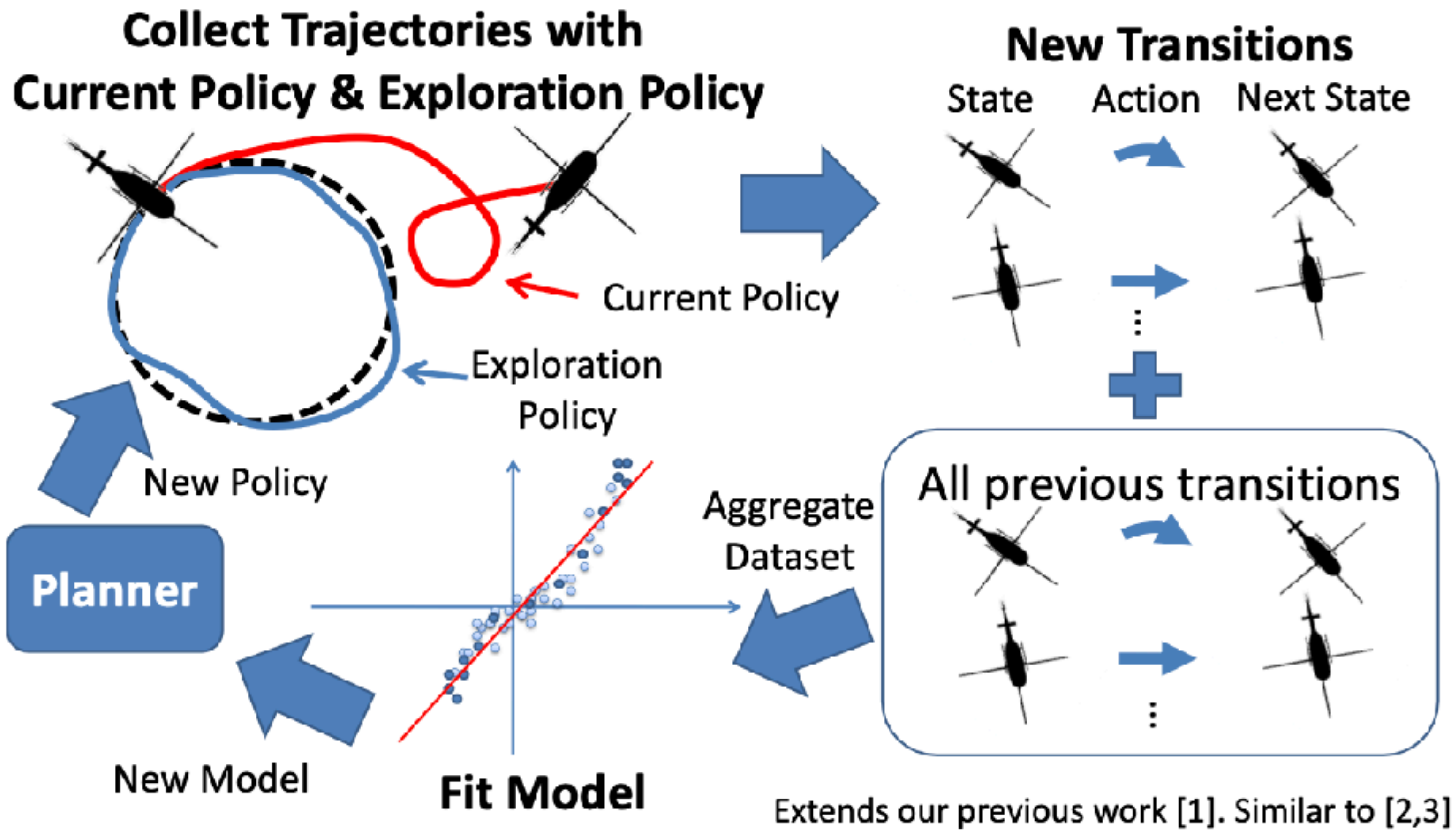
"?!@#!@"

We have seen this problem before!



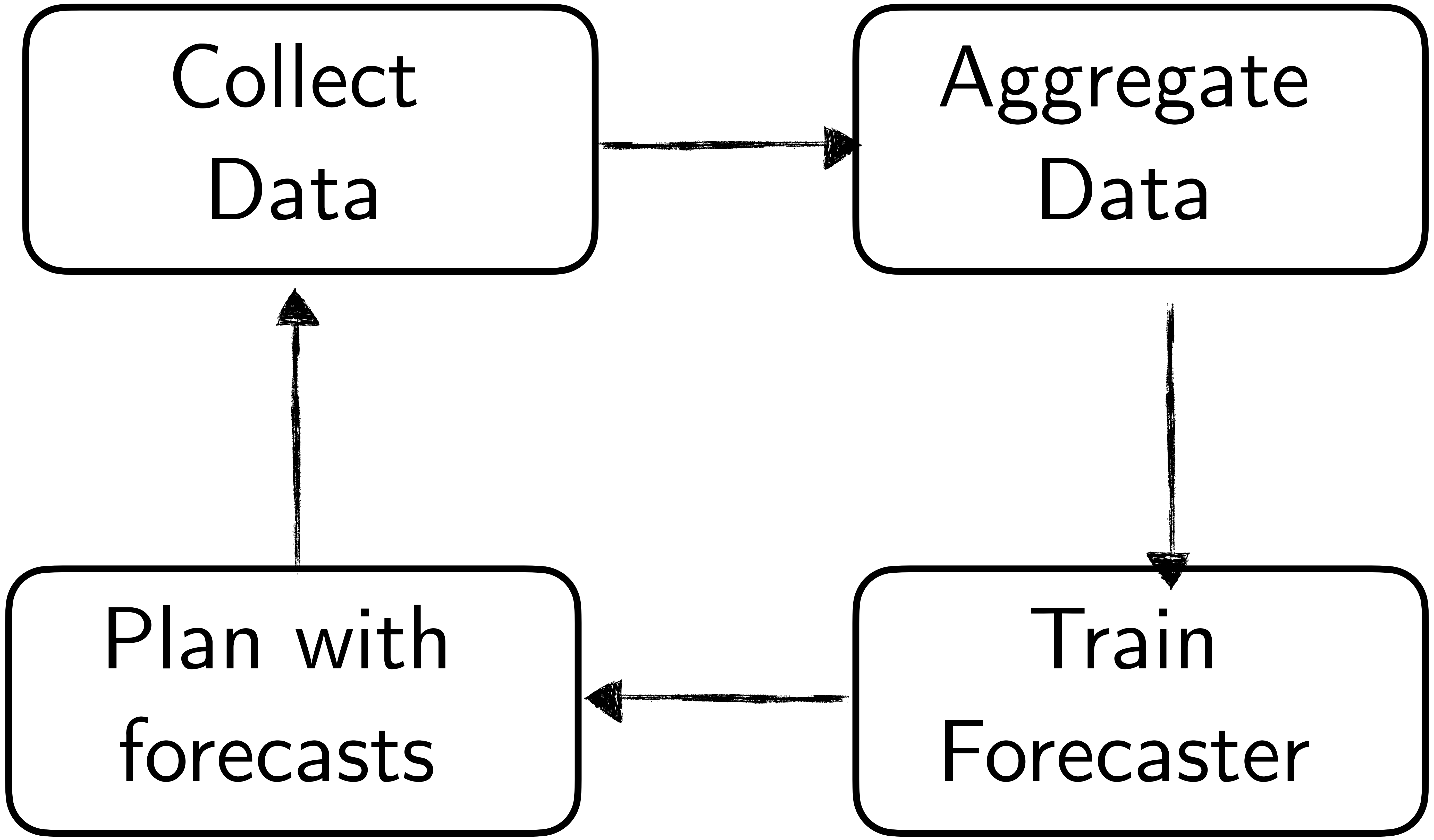
Solution: DAGGER for SysID





Extends our previous work [1]. Similar to [2,3]

DAGGER for Forecasting!



Collect Data

Aggregate Data

Train Forecaster

Plan with forecasts

Shaky foundations of forecasting

Are we using the right model?

Conditional forecasting

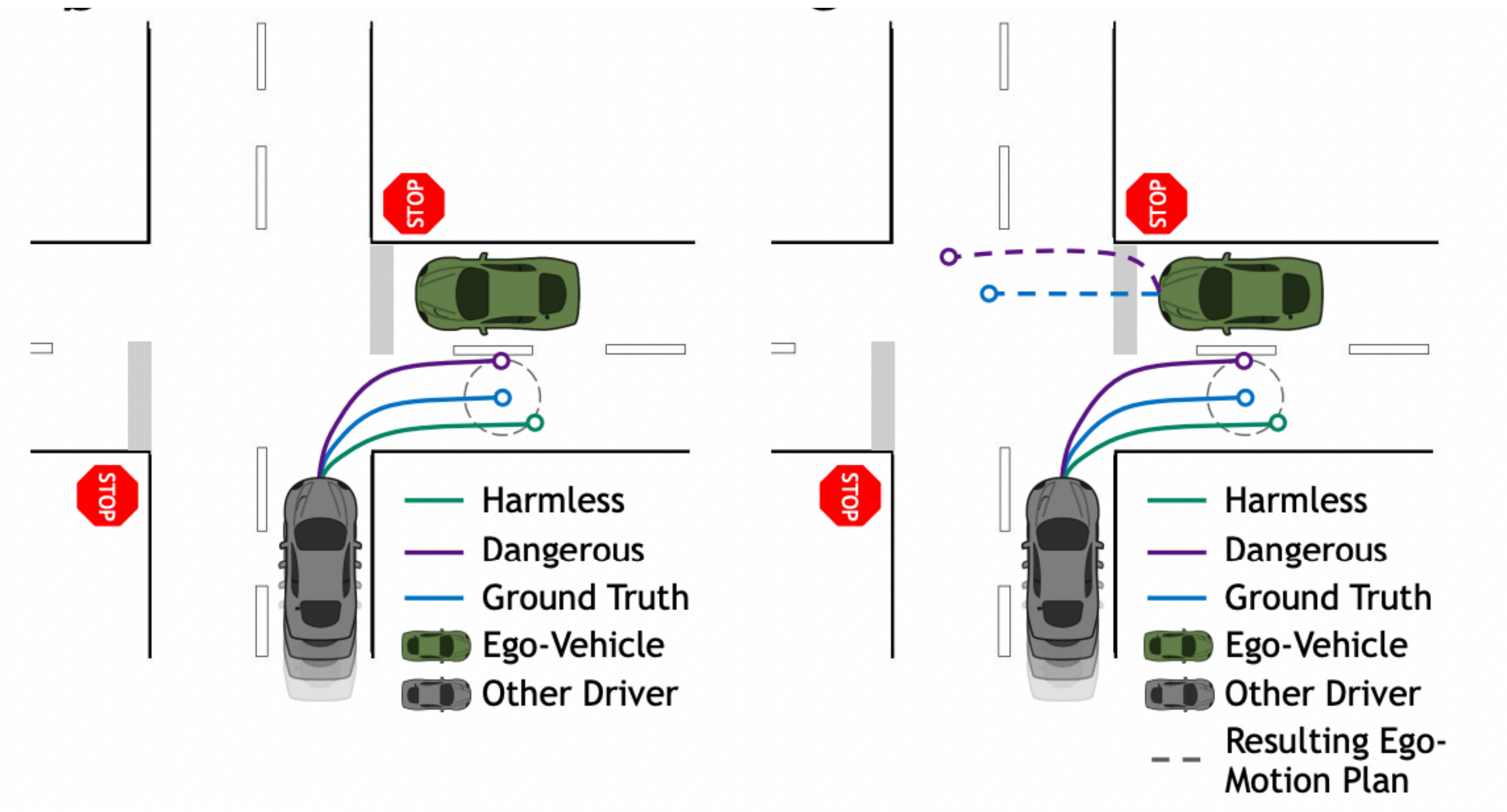
Are we collecting data correctly?

Interactively collect data

Are we using the right loss?

What makes a forecast good?

What makes forecasts good?



Rethinking Trajectory Forecasting Evaluation

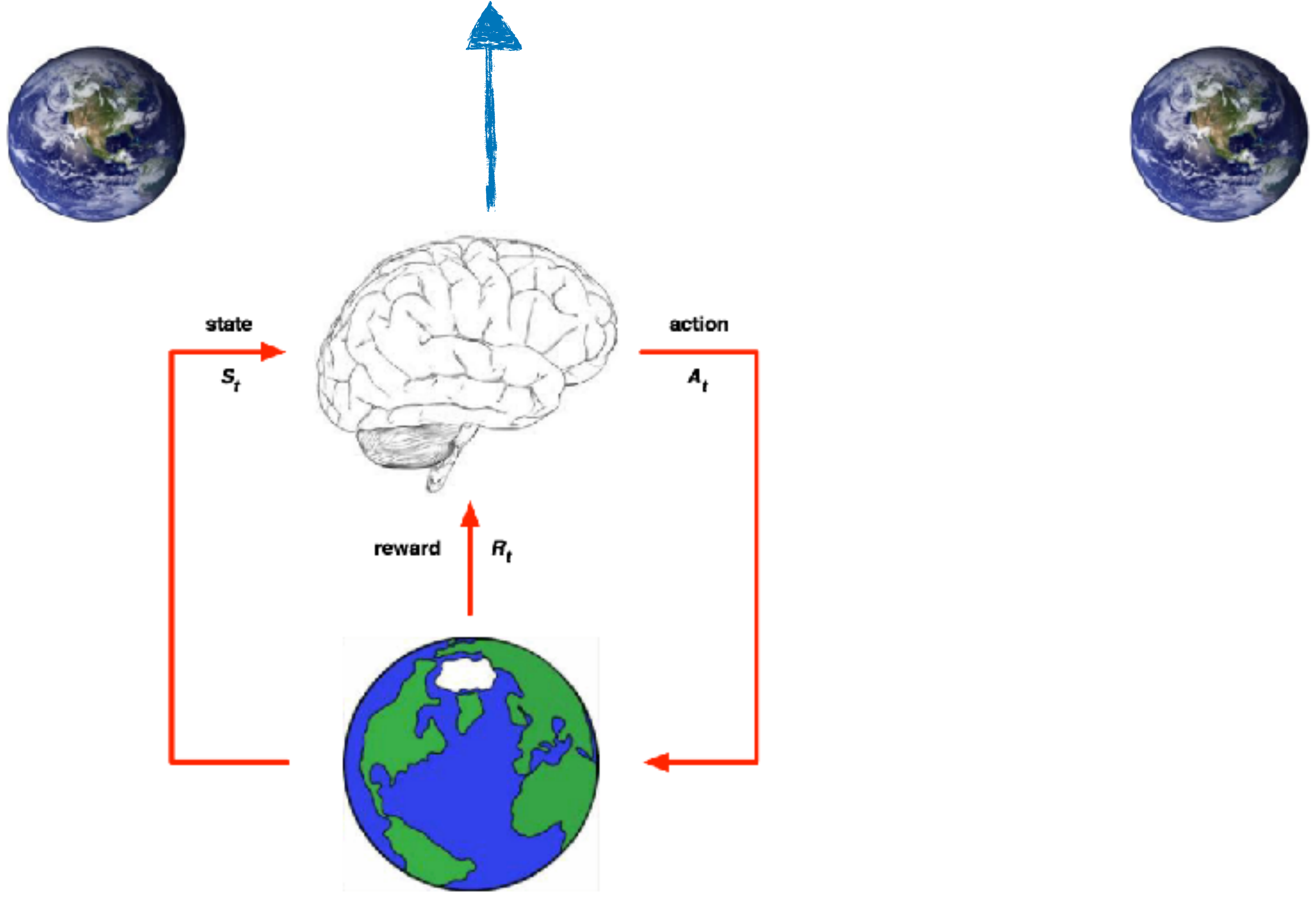
Boris Ivanovic Marco Pavone
NVIDIA Research
{bivanovic, mpavone}@nvidia.com

Forecasting is just a model

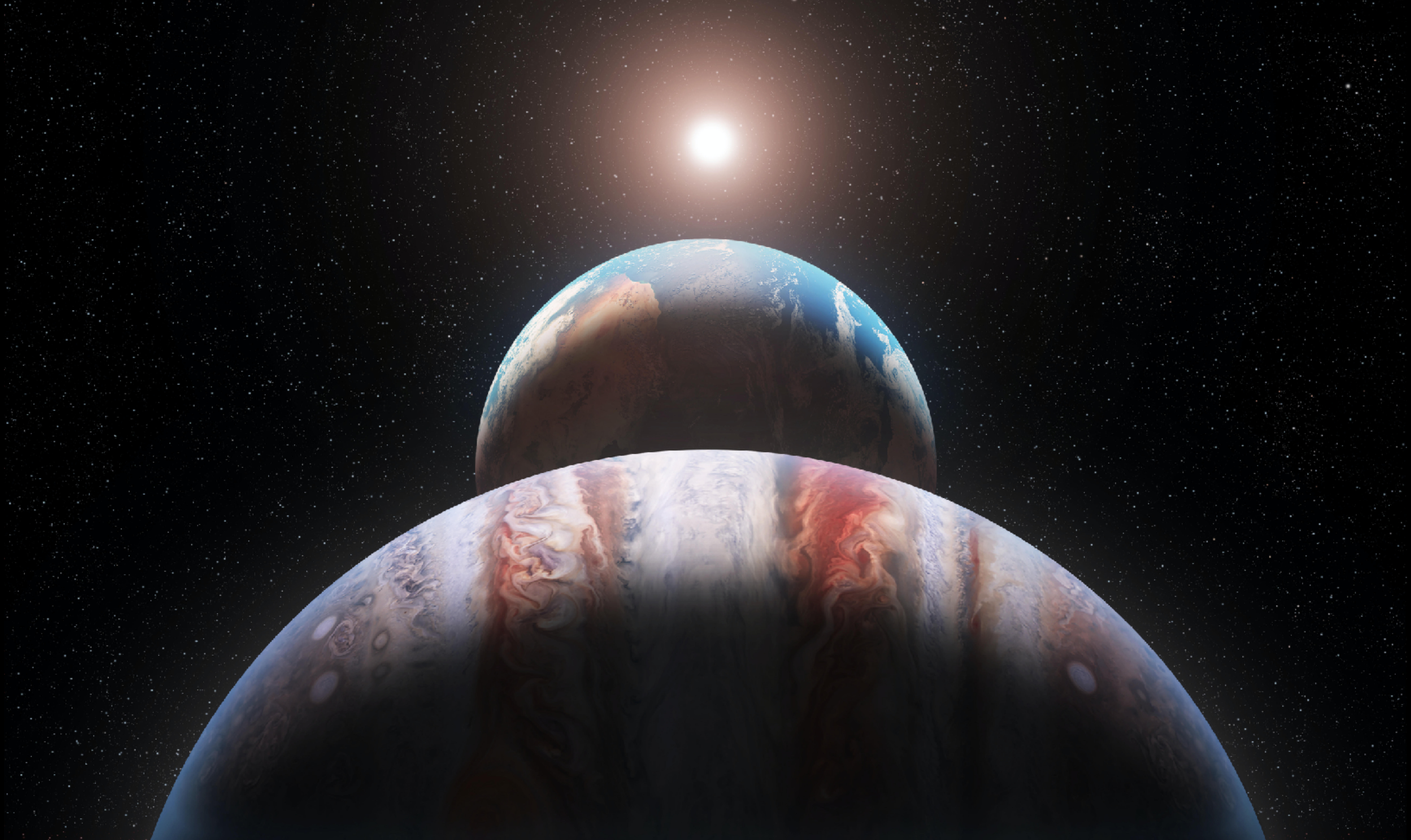
Models are useful fictions

What makes a ~~forecast~~ model good?

$$J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^*)$$



The *Double* Simulation Lemma



~~Forecast~~ Model Learning: It's only a game!

$$\left| J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^*) \right|$$

$$= \min_{\hat{M}} \max_{V} \sum_{t=1}^{T} \mathop{E}_{\substack{s_t, a_t \\ d^{\hat{\pi}}_{M^*}}} \left| \sum_{s_{t+1}} M^*(s_{t+1}|s_t,a_t) V(s_{t+1}) - \hat{M}(s_{t+1}|s_t,a_t) V(s_{t+1}) \right|$$

$$+$$

$$\sum_{t=1}^{T} \mathop{E}_{\substack{s_t, a_t \\ d^{\pi^*}_{M^*}}} \left| \sum_{s_{t+1}} M^*(s_{t+1}|s_t,a_t) V(s_{t+1}) - \hat{M}(s_{t+1}|s_t,a_t) V(s_{t+1}) \right|$$

😇 😈

where $\hat{\pi} = \text{PLANNER}(\hat{M})$

Shaky foundations of forecasting

Are we using the right model?

Conditional forecasting

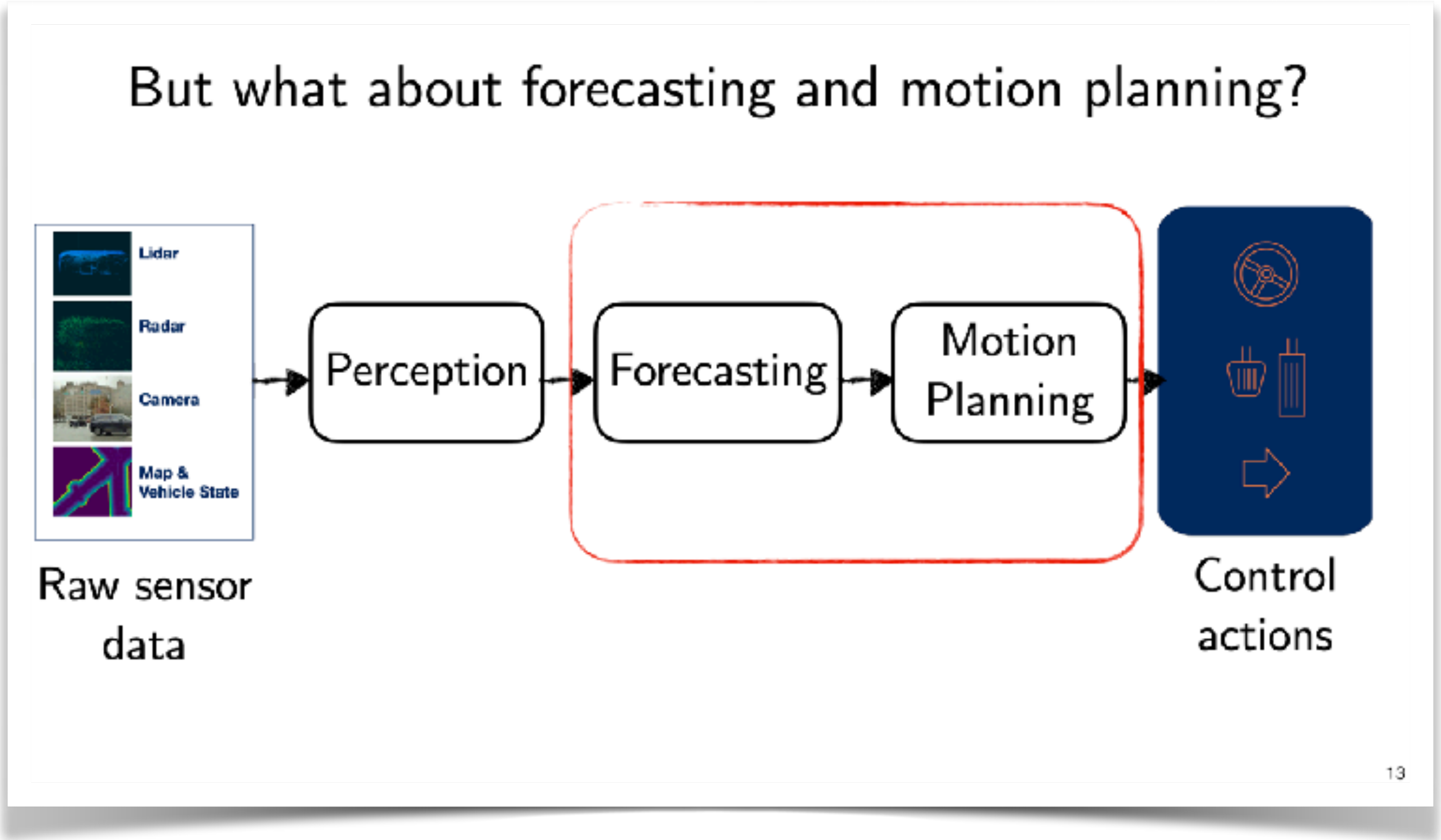
Are we collecting data correctly?

Interactively collect data

Are we using the right loss?

Performance Difference

tl;dr



But what about forecasting and motion planning?

Lidar

Radar

Camera

Map & Vehicle State

Raw sensor data

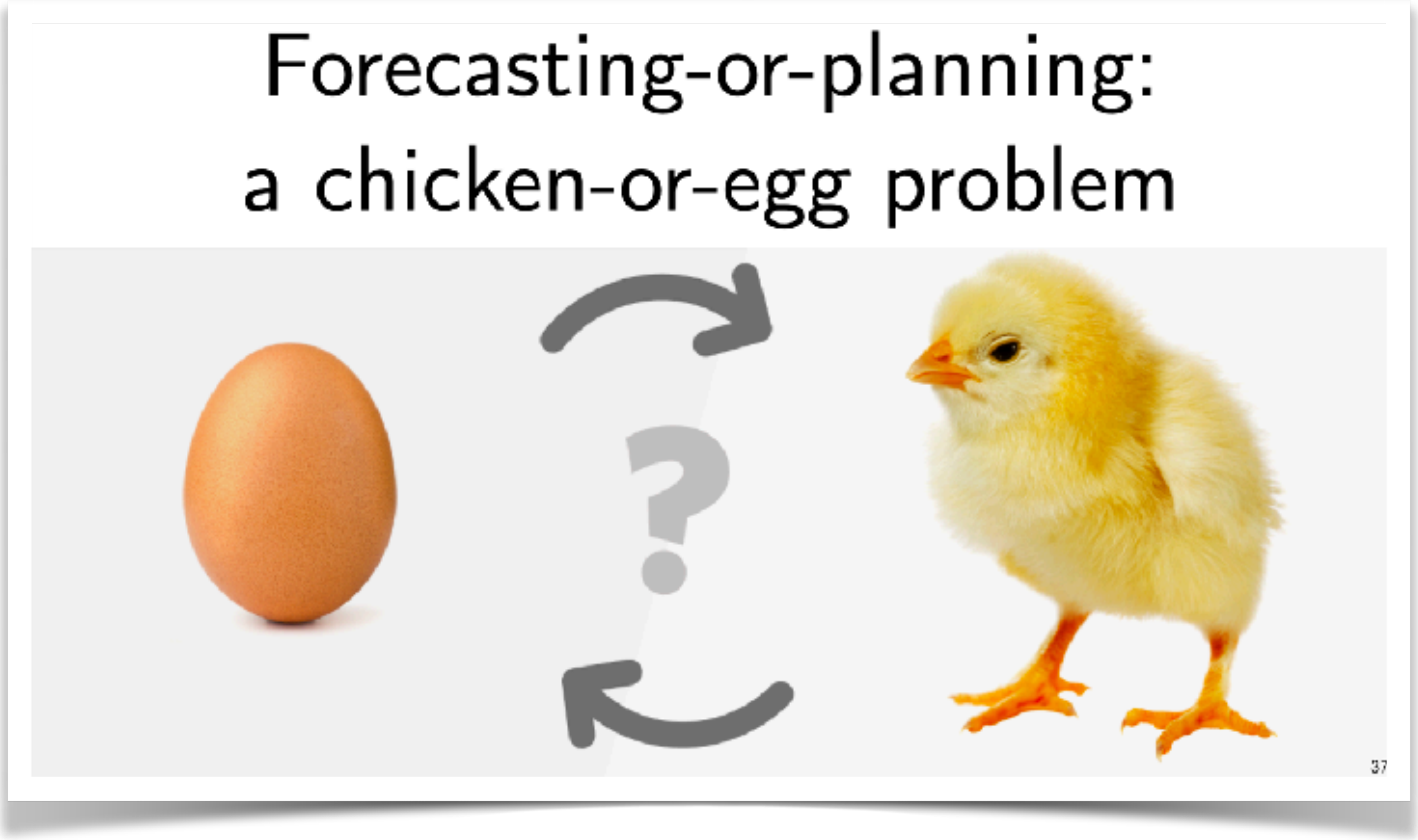
Perception

Forecasting

Motion Planning

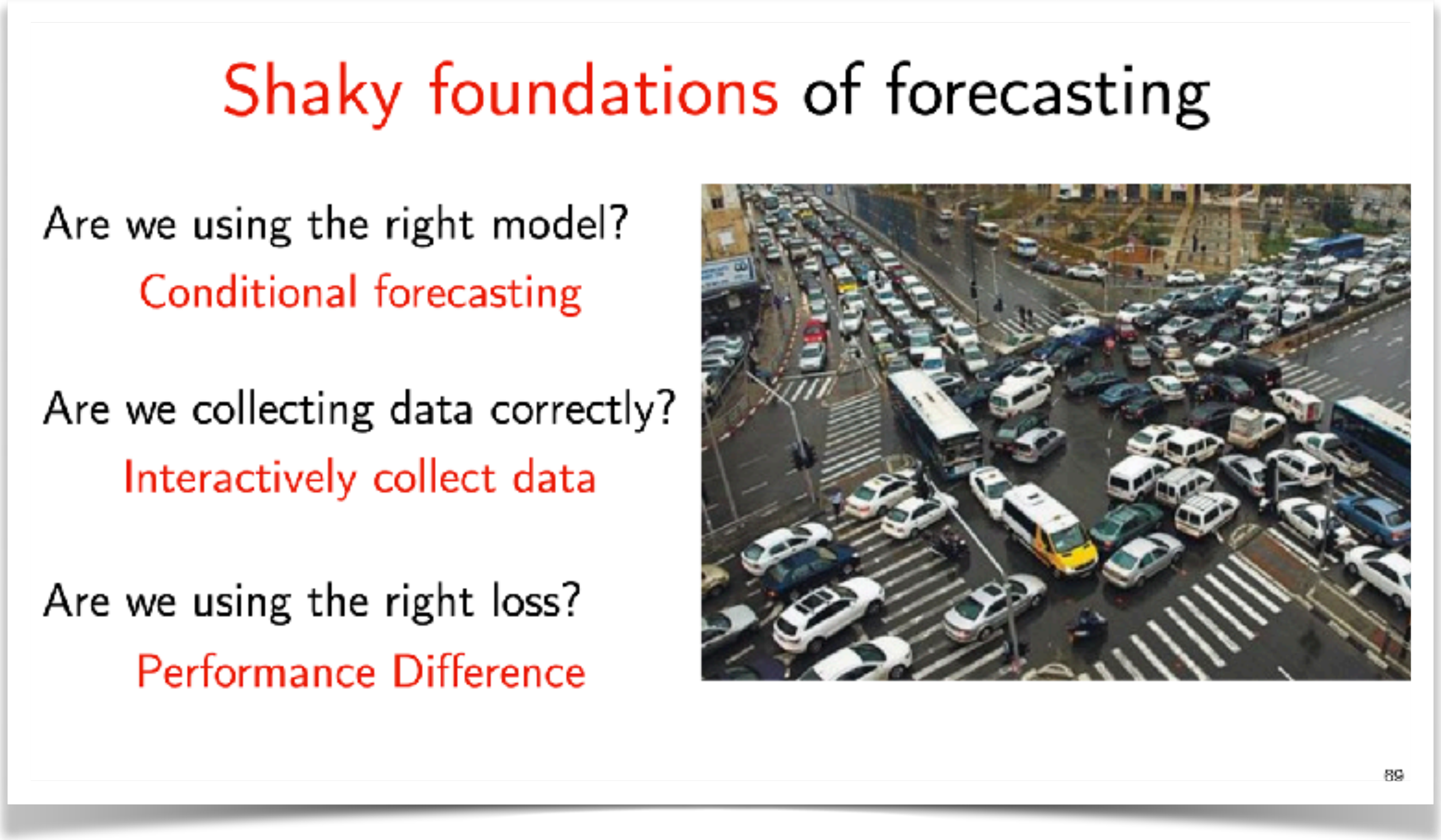
Control actions

13



Forecasting-or-planning:
a chicken-or-egg problem

37



Shaky foundations of forecasting

Are we using the right model?
Conditional forecasting

Are we collecting data correctly?
Interactively collect data

Are we using the right loss?
Performance Difference

89