OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Software Algorithm Pipeline *and* Deepdive to Industry Solutions

Dr. Hongyang Li

OpenDriveLab / 浦江实验室

2024年3月6日

Outline

First 45 min

- **自动驾驶软件算法体系**
  - 各模块简介 / Task Definition
  - 车辆动力学 / Dynamics

Second 45 min

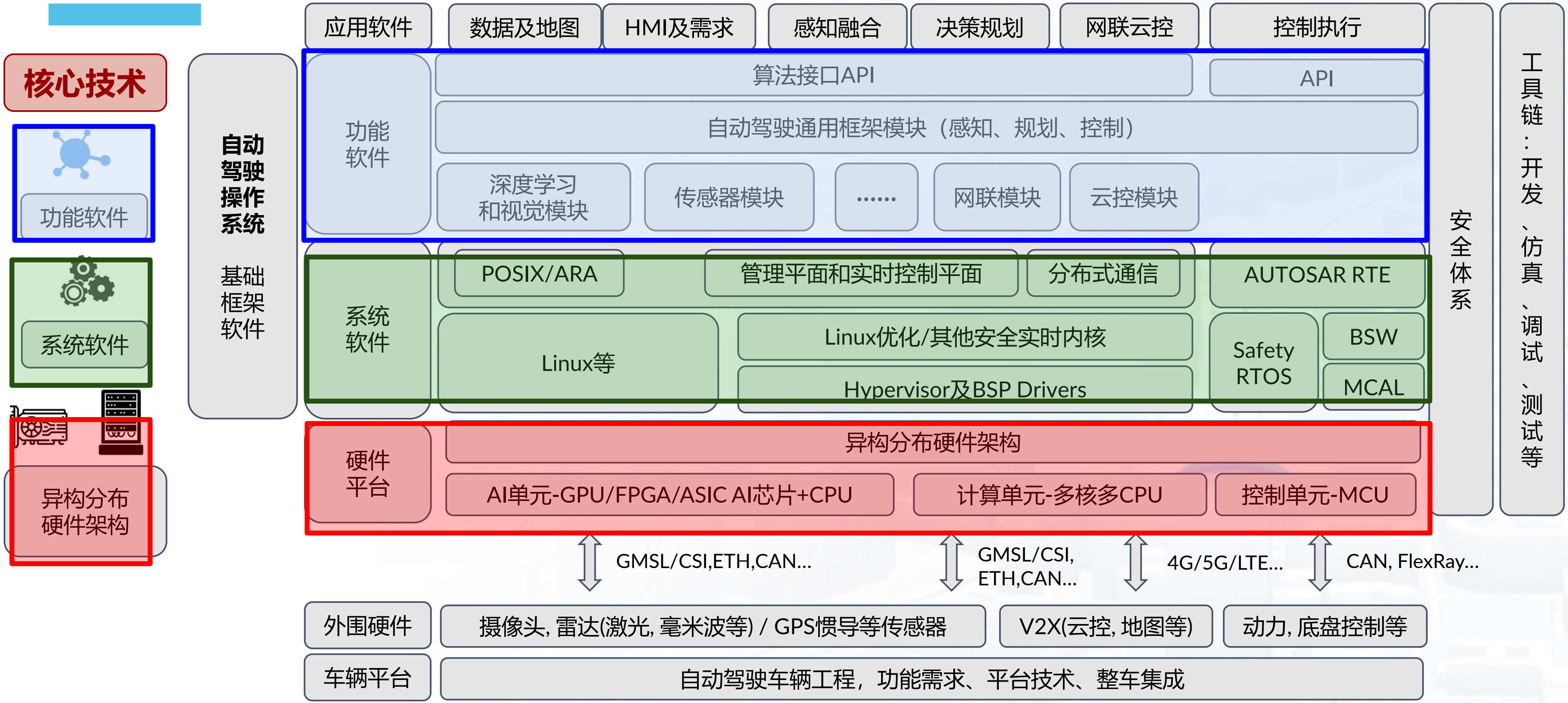
- **端到端自动驾驶简介**
  - 主要工作介绍 / Research Roadmap
  - 挑战 / Challenges

Third 45 min

- **主流车企技术框架**
  - Mobileye - CES 2024
  - 特斯拉 - AI Day 2023

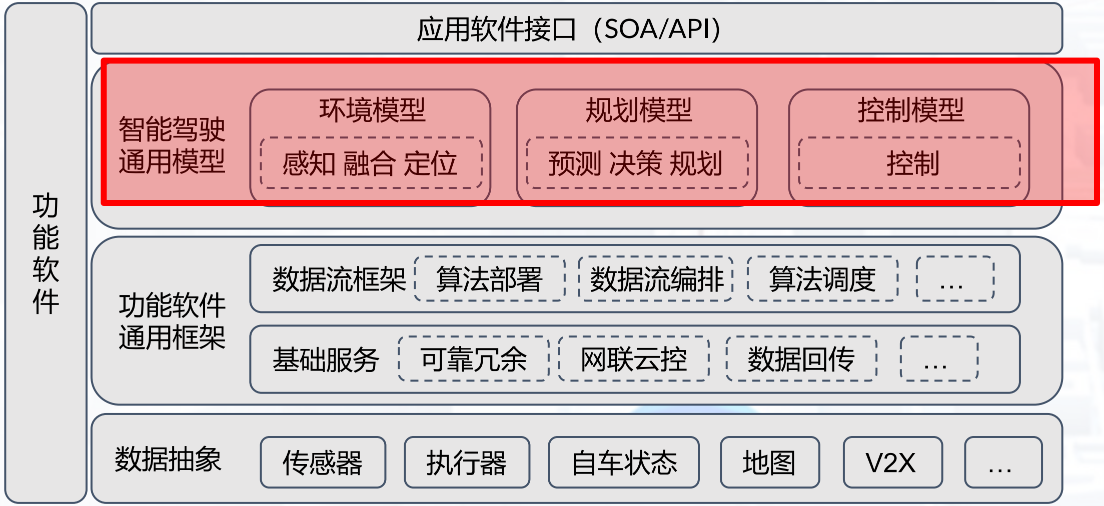
自动驾驶系统

https://www.sohu.com/a/577938734_121124366

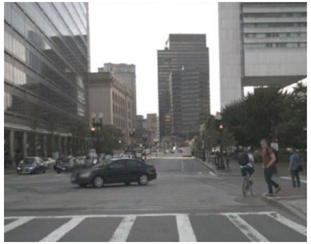


回顾：自动驾驶软件算法系统

智能驾驶通用模型是对智能驾驶中智能认知、智能决策和智能控制等过程的模型化抽象。对应于自动驾驶中环境感知、决策与规划、控制与执行三大部分，通用模型也可以分为**环境模型**、**规划模型**和**控制模型等。**



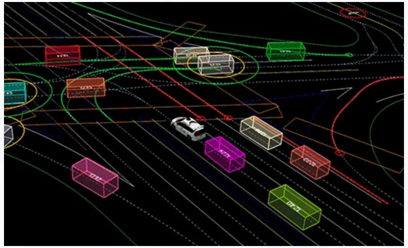
回顾：自动驾驶算法模块



Challenge | Various weather, illuminations, and scenarios

Sensor → Perception → (Bounding boxes) → Prediction → (Waypoints) → Planning → (Trajectory)

What are around?



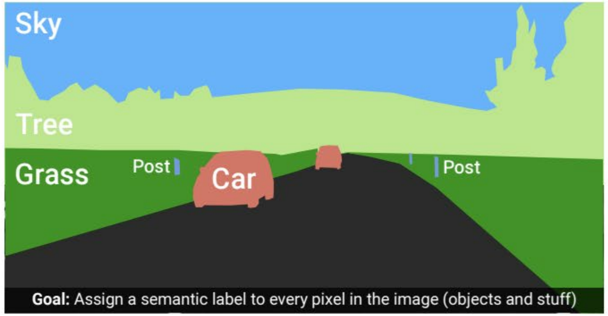
How will they go in the future?

Where should I go?

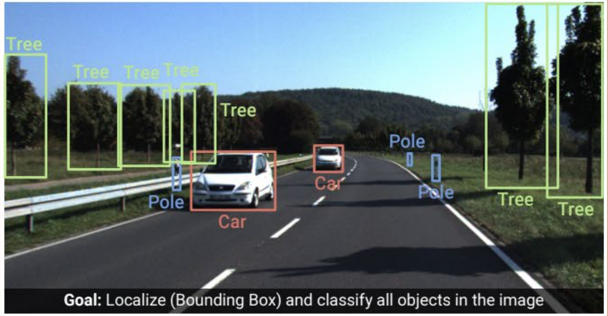
软件算法系统 - 分类、分割

Credit to Prof. Andreas Geiger

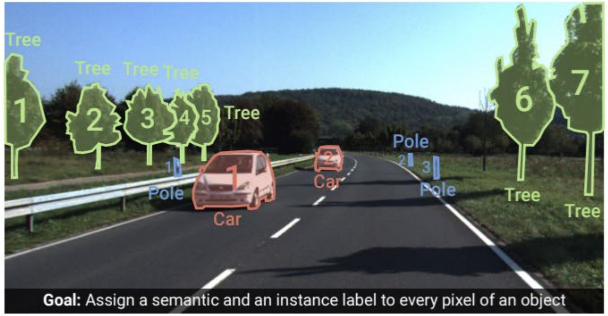
Image Classification



Semantic Segmentation



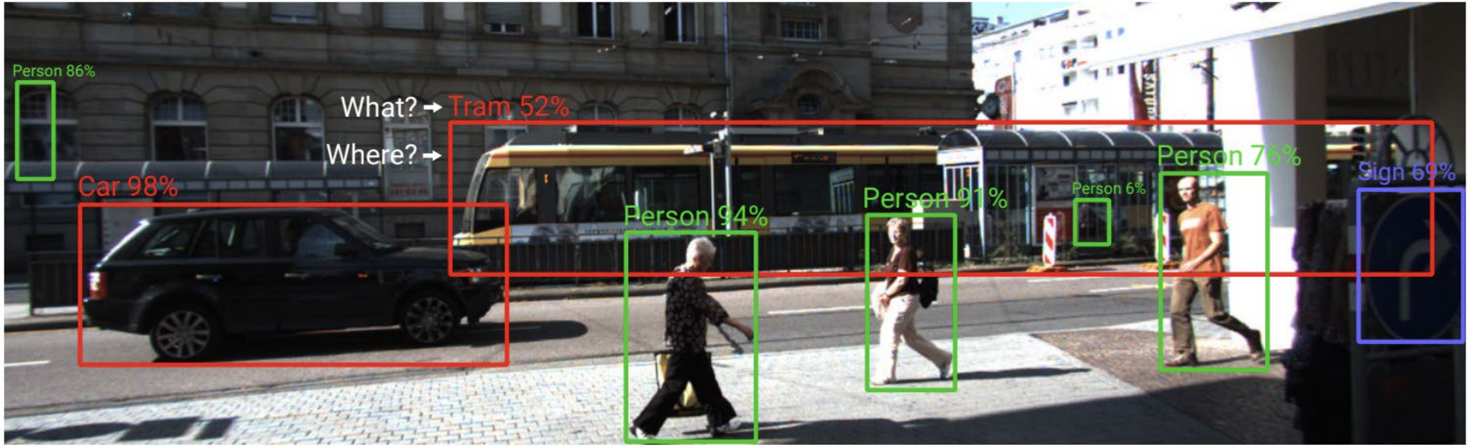
Object Detection



Instance Segmentation

软件算法系统 - 检测（2D）

Credit to Prof. Andreas Geiger



Problem Setting:

- **Input:** RGB Image or laser range scan
- **Output:** Set of 2D/3D bounding boxes with category label and confidence
- There are many(∞) possible boxes and even the number of objects is unknown.

软件算法系统 - 检测（3D）

Credit to Prof. Andreas Geiger

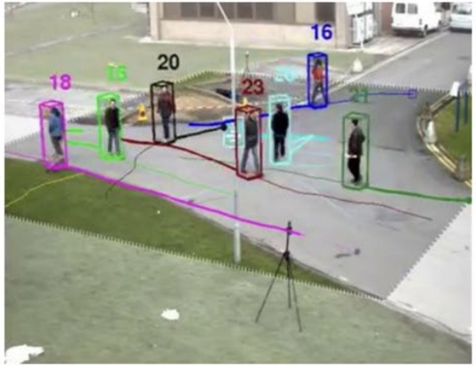
Problem Setting:

- **Input:** RGB Image or laser range scan
- **Output:** Set of 2D/3D bounding boxes with category label and confidence
- There are many(∞) possible boxes and even the number of objects is unknown.

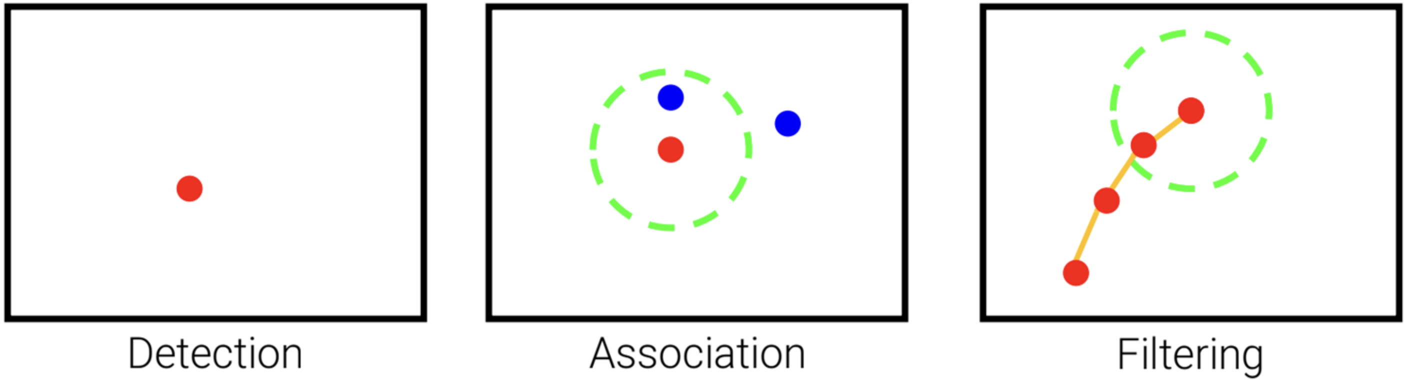
软件算法系统 - 跟踪

Problem Definition:

- Given noisy object detections (e.g., bounding boxes) for each frame of a sequence, **associate** those that belong to the same physical object
- Reject detections that are false alarms; **initiate/delete** object tracks
- Estimate **object state** by considering an observation/ motion model



软件算法系统 - 跟踪



- **Detection:** Where are candidate objects in each frame?
- **Association:** Which detection corresponds to which object?
- **Filtering:** What is the most likely object state, e.g., location and size?

软件算法系统 - 规划与控制

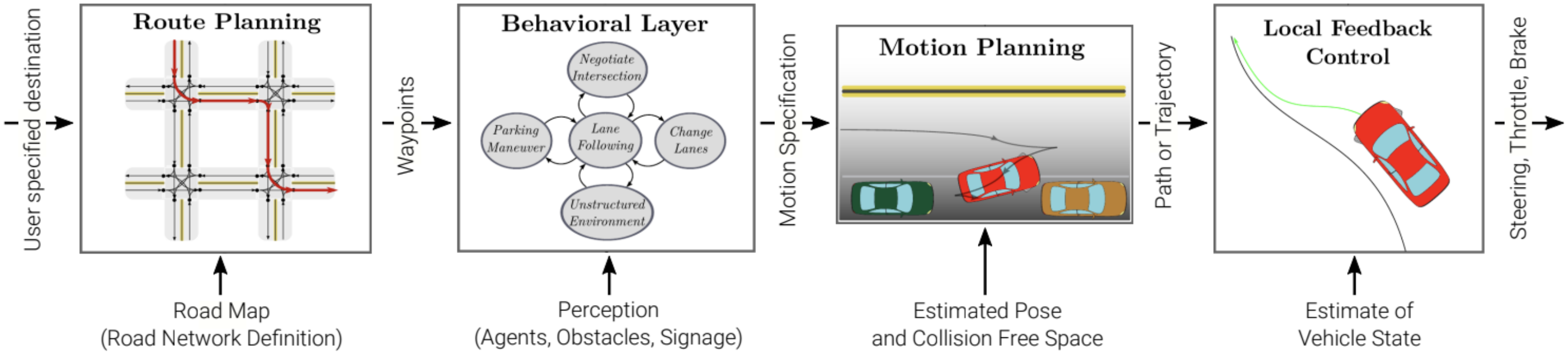
Problem Definition:

- Goal: find and follow path from current location to destination
- Take static infrastructure and dynamic objects into account
- Input: vehicle and environment state (via perception stack)
- Output: path or trajectory as input to vehicle controller

Challenges:

- Driving situations and behaviors are very complex
- Thus difficult to model as a single optimization problem

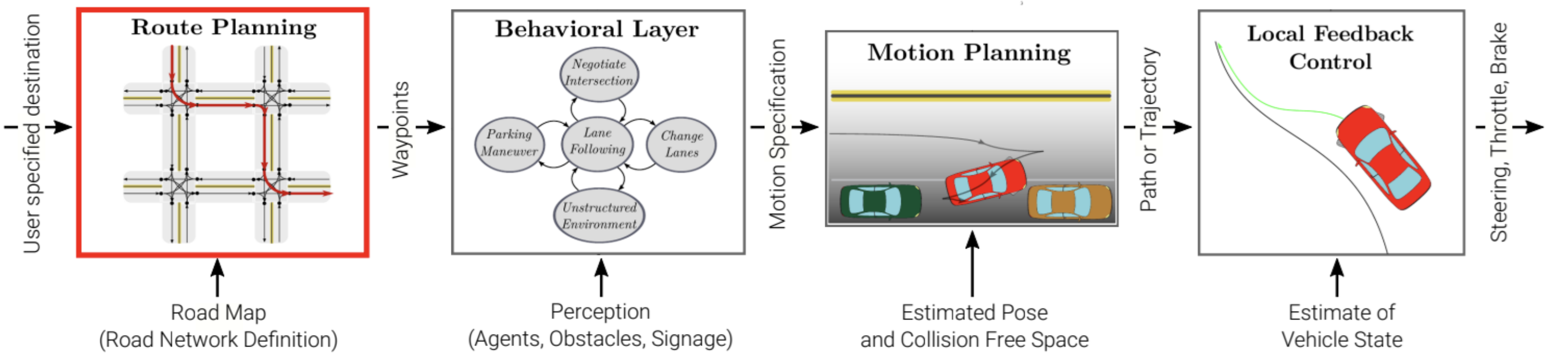
软件算法系统 - 规划与控制



- Idea: Break planning problem into a **hierarchy of simpler problems**
- Each problem tailored to its scope and level of abstraction
- Earlier in this hierarchy means higher level of abstraction
- Each optimization problem will have constraints and objective functions

软件算法系统 - 规划与控制

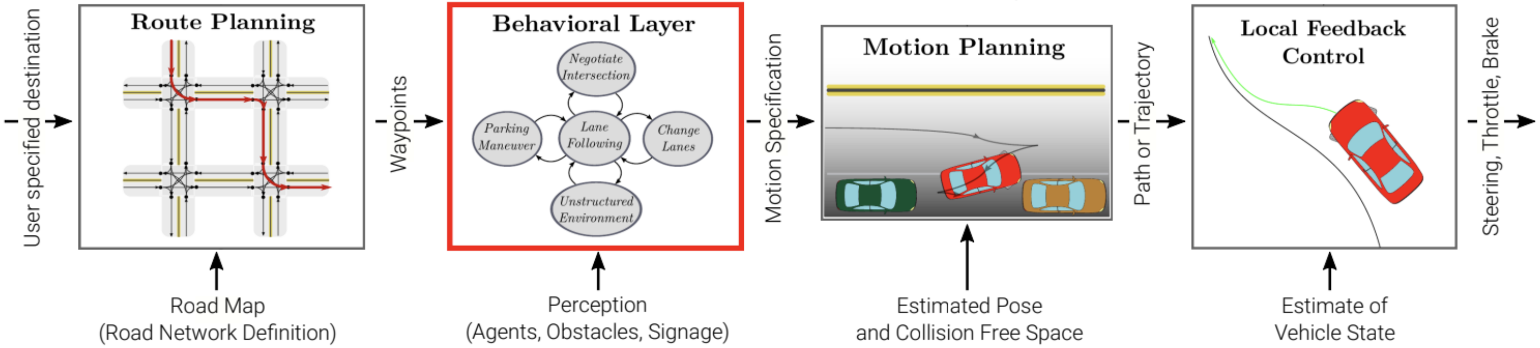
Step 1: Route planning



- Represent **road network** as **directed graph**
- Edge weights correspond to road segment length or travel time
- Problem translates into a minimum-cost graph network problem
- Inference algorithms: Dijkstra, A*, ...

软件算法系统 - 规划与控制

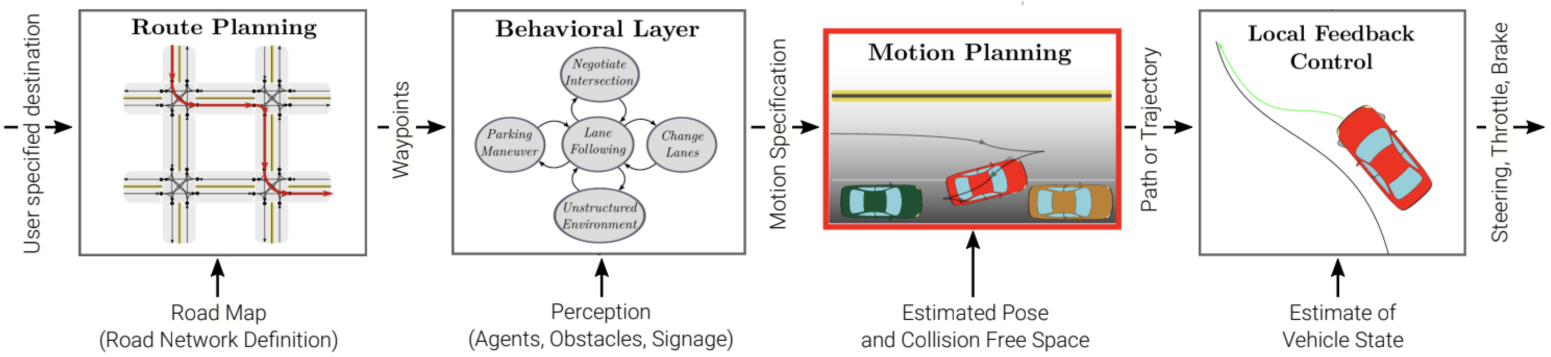
Step 2: Behavior planning



- Select **driving behavior** based on current vehicle/environment state
- E.g. at stop line: stop, observe other traffic participants, traverse
- Often modeled via **finite state machines** (transitions governed by perception)
- Can be modeled probabilistically, e.g., using Markov Decision Processes (MDPs)

软件算法系统 - 规划与控制

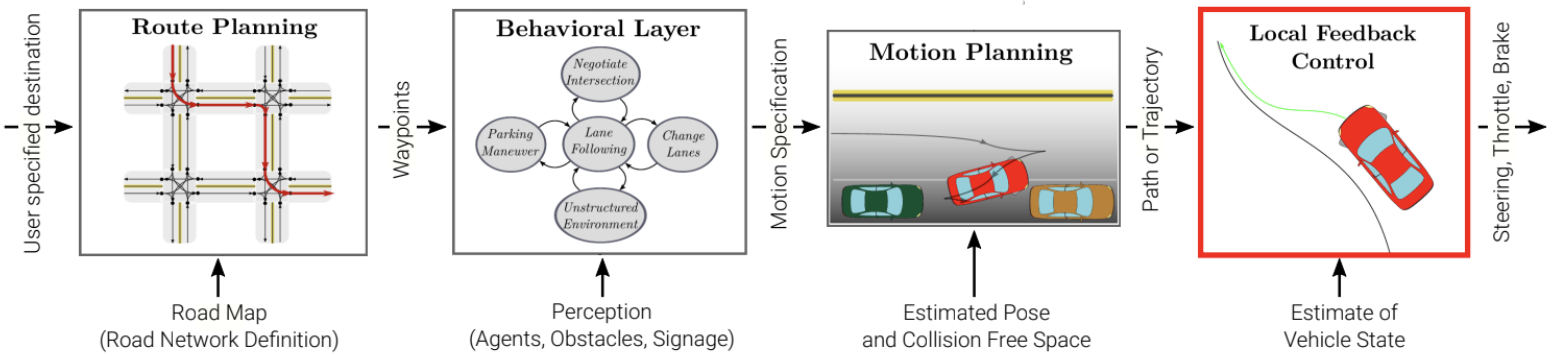
Step 3: Motion planning



- Find feasible, comfortable, safe and fast **vehicle path/trajectory**
- Exact solutions in most cases computationally intractable
- Thus often **numerical approximations** are used
- Approaches: variational methods, graph search, incremental tree-based

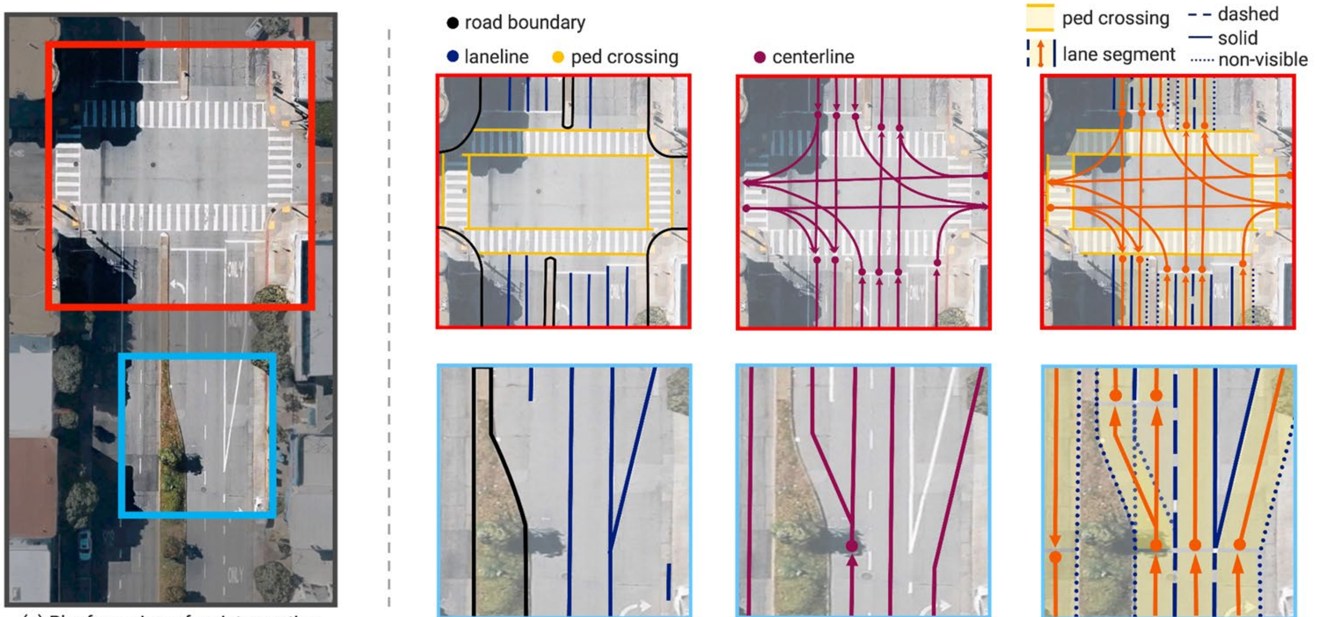
软件算法系统 - 规划与控制

Step 4: Local Feedback Control



- **Feedback controller** executes the path/trajectory from the motion planner
- Corrects errors due to inaccuracies of the vehicle model
- Emphasis on **robustness, stability and comfort**

软件算法系统 - 车道线检测 LaneSegNet



(a) Planform view of an intersection lying in the Bay Area.

(b) Map Element Detection

(c) Centerline Perception

(d) Lane Segment Perception (Ours)

- **Input:** Surrounding Multi-View Images
- **Output:** Lane Segment

Tesla

总结：Tesla AI Day

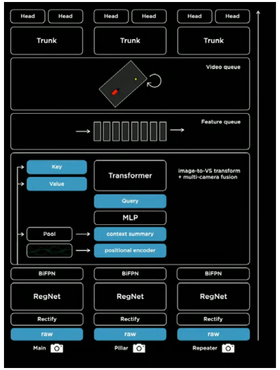
- 2021 - 共享特征多任务网络 Hydranet / 助力FSD的空间理解 BEV Layer/ 增添短期记忆：时空序列feature/ 基于 Neural Network Heuristic 路径规划 / 自研FSD SoC芯片

- 2022 - 栅格网络 Occupancy Network / 车道线及障碍物感知 Lane & Object Perception / 基于Vector Space的FSD路径规划 / 自动标注和数据引擎 AutoLabeler & Data Engine / FSD车端推理计算机 & Dojo训练服务器

Tesla/Waymo/Nvidia对比

绝对领先

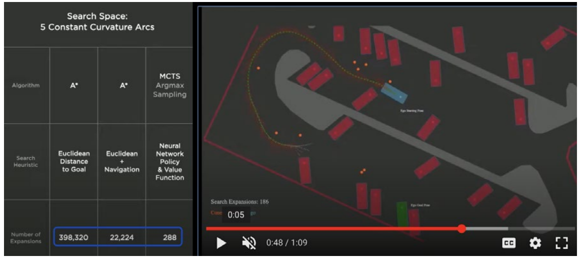
Vision



- Multi-task (still new?)
- **Transformer - nV fusion**
  - 解决遮挡和不同视角投影变换
  - smart summon
- **Spatial-temporal**
  - 增加视频前后信息，更好的帮助感知
  - 有助于后续规控
  - c.f. HDMap工作

Planning/Control

- MCTS + policy net 节省搜索空间

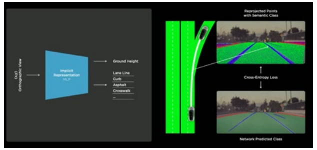


首次提出；Google/Waymo/Uber做的更好

绝对领先/ScaleAI

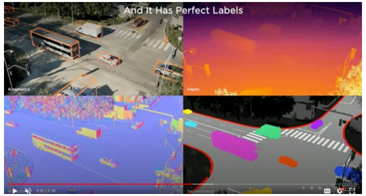
Labelling

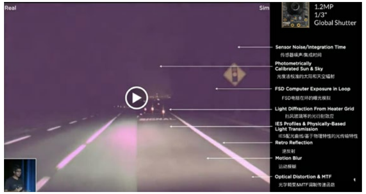
- 手工标注/第三方标注
  - 早期使用；但效率很低
  - 现在：in-house, **1k 标注员工（非外包）**
- 自动标注
  - scalable
  - **GT真值**：借鉴NeRF思想，self-supervised方式
- **数据规模**
  - 60亿label (含vel/depth)
  - 250w video clip
  - 1.5 PB存储
  - 高质量数据(diverse, clean, large)
  - **Ref: waymo**
    - 20w frame, 5 hours
    - 12M labels



英伟达/谷歌/腾讯

Simulation

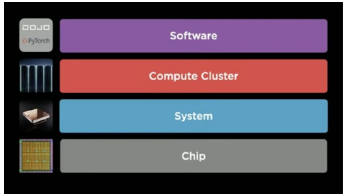




- 仿真提供了大量cornercase数据，帮助算法性能迭代
- 为了仿真和现实场景逼真，做了**五方面**工作；其中场景复原：**neural rendering**

领先

Hardware/Infra/Dojo



- 芯片D1 (DPU)
- 系统
- 计算集群
- 软件架构

各种牛逼数字

- 硬件 3.0 做AI 评估
- 每周运行 100 万次
- 3 个数据中心
- 超过 3000 块 HW3.0 主板组成

Overview

Andrej Karpathy	Tesla vision (core)
Ashok Elluswamy	Planning and Control
Andrej/Ashok	Manual/Auto Labelling (core)
Ashok	Simulation
Milan Kovac	Hardware Integration/Infrastructure
Ganesh Venkataramanan	Dojo
Elon Musk	Tesla Bot (skip)
All	Q&A (skip)

Overview

- **Tesla vision**
- Planning and Control
- Manual/Auto Labelling
- Simulation
- Hardware Integration/Infrastructure
- Dojo

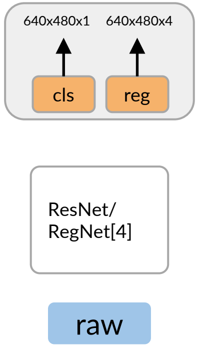
Sum-up: the roadmap in Tesla Vision over the years

- Algorithm
- Product/software/version
- Labelling

2016~

- Regular Network
- 2D detector
- Software 1.0

- manual labeling
- image space labeling



[1] Tesla CVPR 2021 workshop video
[2] Tesla AI day
[3] Smart Summon released on Sep. 26th 2019
[4] Radosavovic, Ilija, et al. "Designing network design spaces." CVPR 2020

Sum-up: the roadmap in Tesla Vision over the years

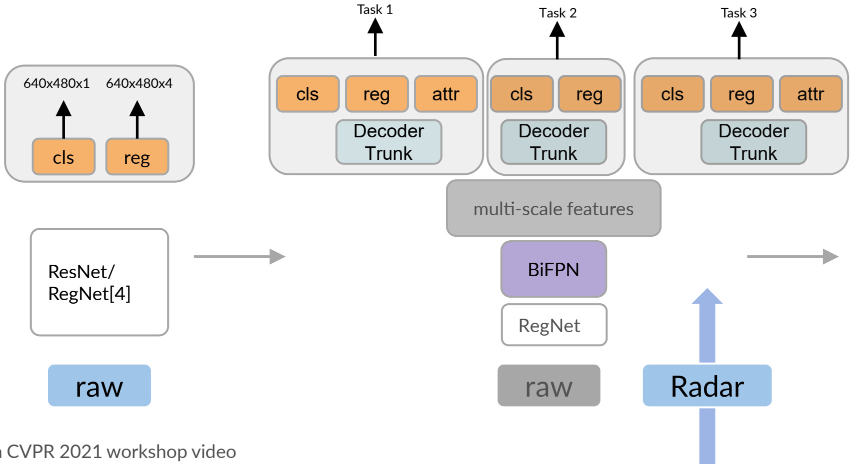
- Algorithm
- Product/software/version
- Labelling

2016~
- Regular Network
- 2D detector
- Software 1.0

- manual labeling
- image space labeling

2018-2019
- **Multi-task learning** - “HydraNets”
- Autopilot 4.0

- manual labeling
- vector space labeling



[1] Tesla CVPR 2021 workshop video
[2] Tesla AI day
[3] Smart Summon released on Sep. 26th 2019
[4] Radosavovic, Ilija, et al. "Designing network design spaces." CVPR 2020

Sum-up: the roadmap in Tesla Vision over the years

- Algorithm
- Product/software/version
- Labelling

2016~
- Regular Network
- 2D detector
- Software 1.0

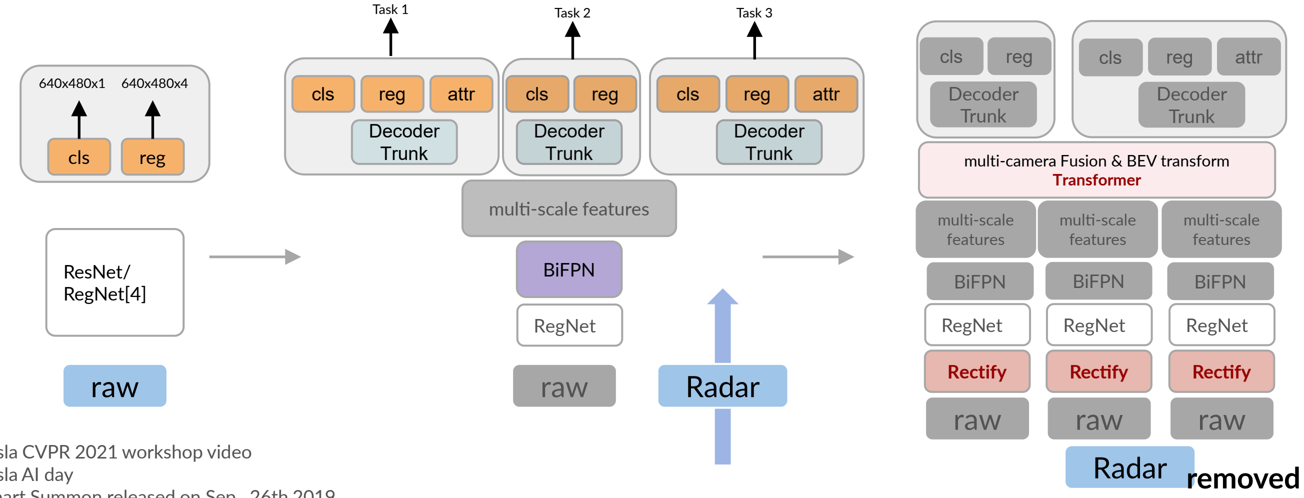
- manual labeling
- image space labeling

2018-2019
- **Multi-task learning** - "HydraNets"
- Autopilot 4.0

- manual labeling
- vector space labeling

2019-2020
- **Fusion** - Smart Summon [3]
- **Transformer**
- Software 2.0
- Radar removed [1]

- auto labeling
- vector space labeling



[1] Tesla CVPR 2021 workshop video
[2] Tesla AI day
[3] Smart Summon released on Sep. 26th 2019
[4] Radosavovic, Ilija, et al. "Designing network design spaces." CVPR 2020

Sum-up: the roadmap in Tesla Vision over the years

- Algorithm
- Product/software/version
- Labelling

2016~
- Regular Network
- 2D detector
- Software 1.0

- manual labeling
- image space labeling

2018-2019
- **Multi-task learning** - "HydraNets"
- Autopilot 4.0

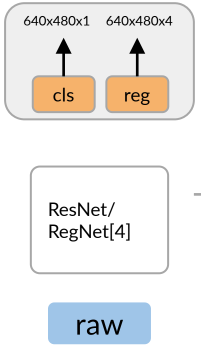
- manual labeling
- vector space labeling

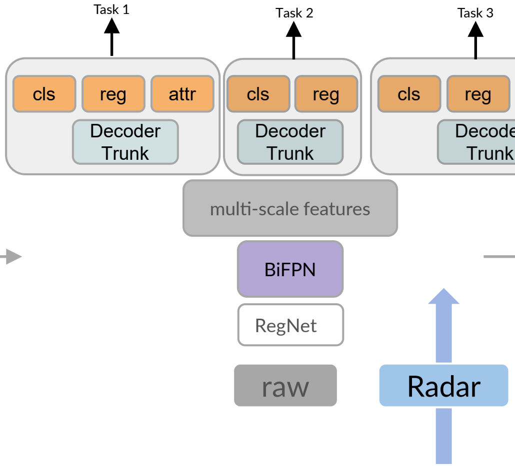
2019-2020
- **Fusion** - Smart Summon [3]
- **Transformer**
- Software 2.0
- Radar removed [1]

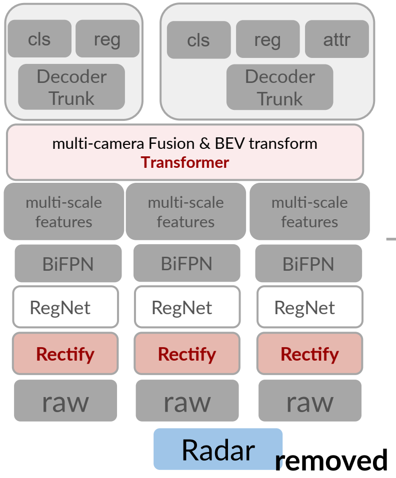
- auto labeling
- vector space labeling

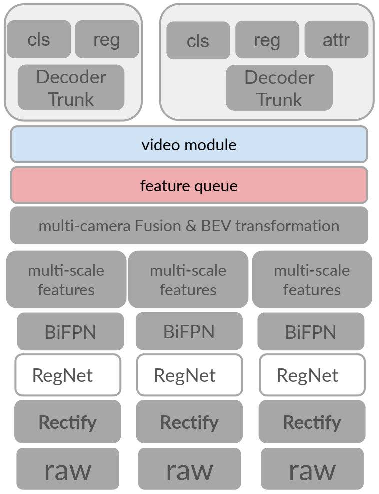
2021 [2]
- Spatial-temporal
- Video module
- feature queue
- **FSD beta 9.0/10.0**

- auto labeling
- vector space labeling









[1] Tesla CVPR 2021 workshop video
[2] Tesla AI day
[3] Smart Summon released on Sep. 26th 2019
[4] Radosavovic, Ilija, et al. "Designing network design spaces." CVPR 2020

User Interface Performance

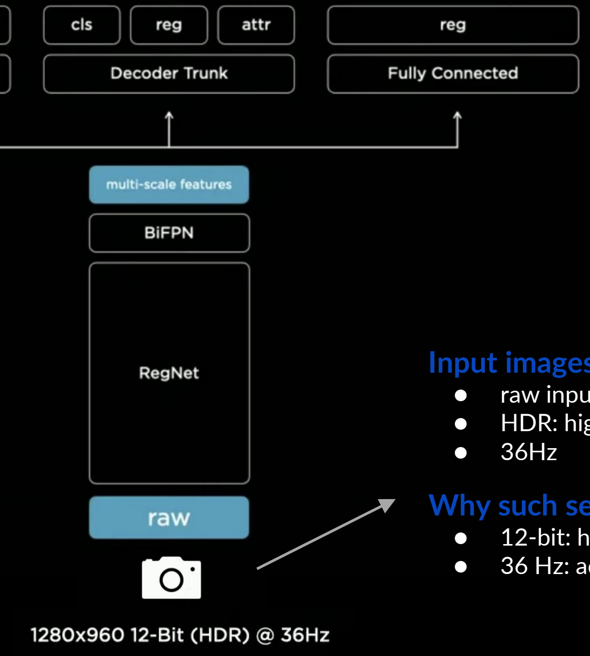
Network Design - at the beginning

ResNet: simply detecting vehicles

[1] Tan, Mingxing, Ruoming Pang, and Quoc V. Le.
"Efficientdet: Scalable and efficient object detection." CVPR 2020.

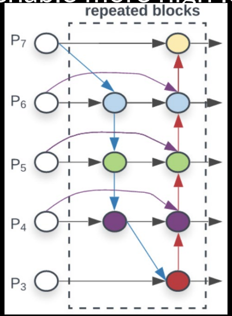
Network Design - at the beginning

Multi-task learning - “HydraNets”



Then to BiFPN [1]. Why?

- fuse different level features
- weighted different features for better fusion
- repeated blocks enable more high level feature fusion



Input images:

- raw input
- HDR: high dynamic range(12-bit)
- 36Hz

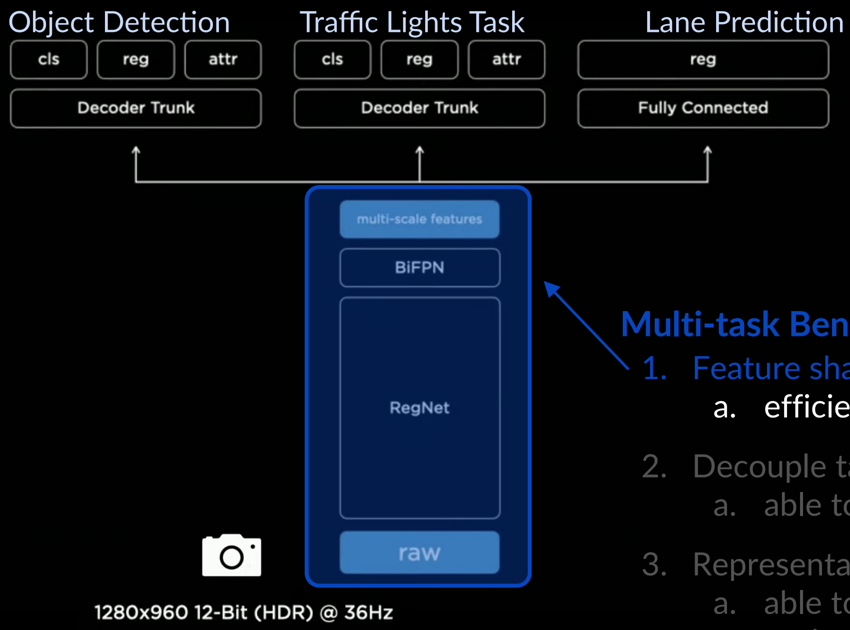
Why such settings?

- 12-bit: have larger illumination representation
- 36 Hz: add temporal information

[1] Tan, Mingxing, Ruoming Pang, and Quoc V. Le.
"Efficientdet: Scalable and efficient object detection." CVPR 2020.

Network Design - four years (roughly) ago

Multi-task learning - “HydraNets”

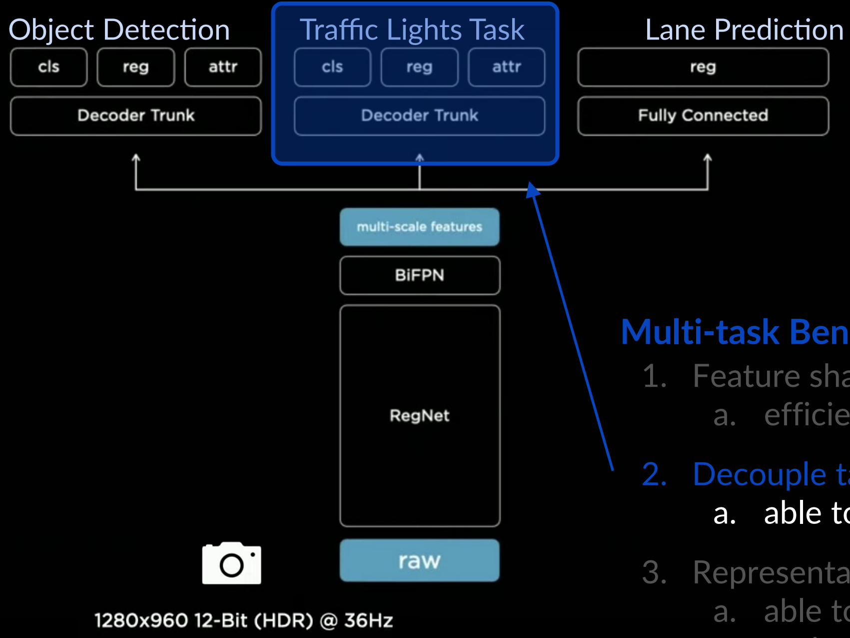


Multi-task Benefits

1. Feature sharing
    a. efficient at test time
2. Decouple tasks
    a. able to fine-tune tasks individually
3. Representation Bottleneck
    a. able to feature cache & speed up fine-tuning

Network Design - four years (roughly) ago

Multi-task learning - “HydraNets”

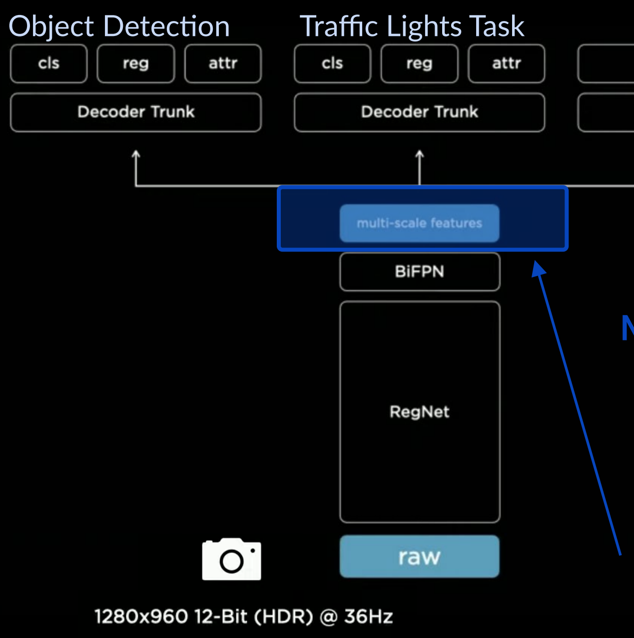


Multi-task Benefits

1. Feature sharing
   a. efficient at test time
2. Decouple tasks
   a. able to fine-tune tasks individually
3. Representation Bottleneck
   a. able to feature cache & speed up fine-tuning

Network Design - four years (roughly) ago

Multi-task learning - "HydraNets"



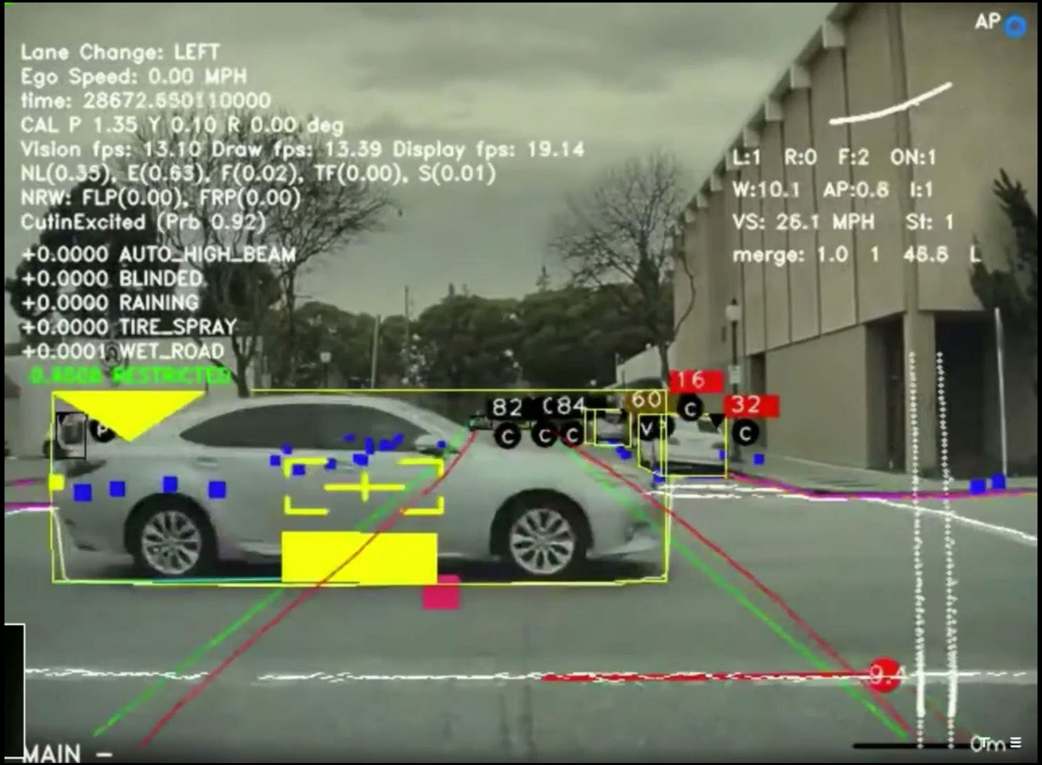
Multi-task Benefits

1. Feature sharing
   a. efficient at test time
2. Decouple tasks
   a. able to fine-tune tasks individually
3. Representation Bottleneck
   a. able to feature cache & speed up fine-tuning

Network Design - four years (roughly) ago

Multi-task learning - “HydraNets” - Experiments

- 自车信息
  - ego speed
  - P/Y/R
  - fps 15~
- Cutin or not
- Scene classification
  - HBA
  - 下雨
  - 道路湿滑
  - 限行区域
- car detection
- lane detection
- stop sign
- distance measurement



Smart Summon - Per-camera detection then fusion (nV)

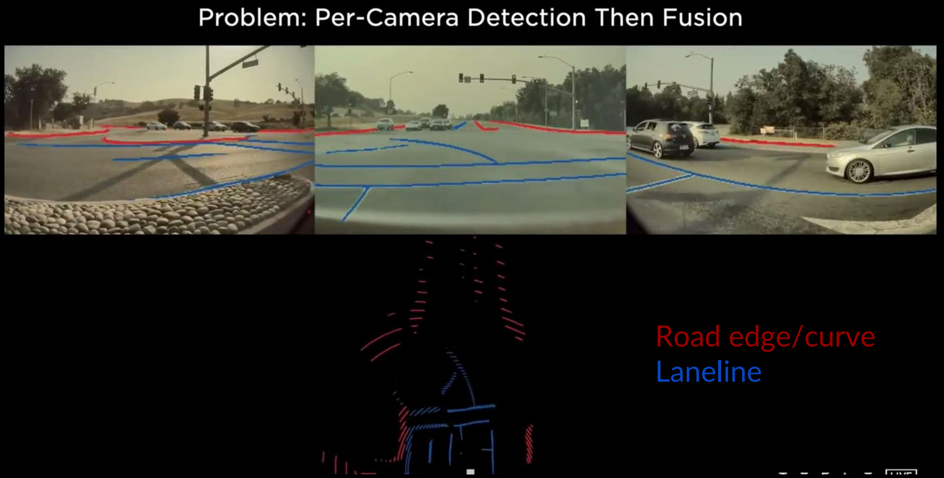
Difficulties:

- fusion process is hard to write explicitly
- image space result is hard to use

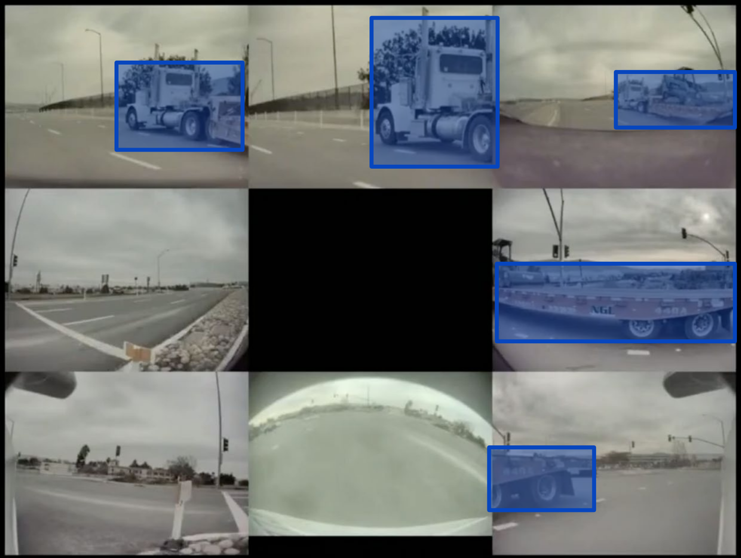
Smart Summon - Per-camera detection then fusion (nV)

Goal: summon vehicle to the person nearby

Cast out image-space predictions onto vector space



Traditional method:
- project from image plane to vector space. → Don't have depth per pixel
- assumption ground is horizontal. → which is not ture



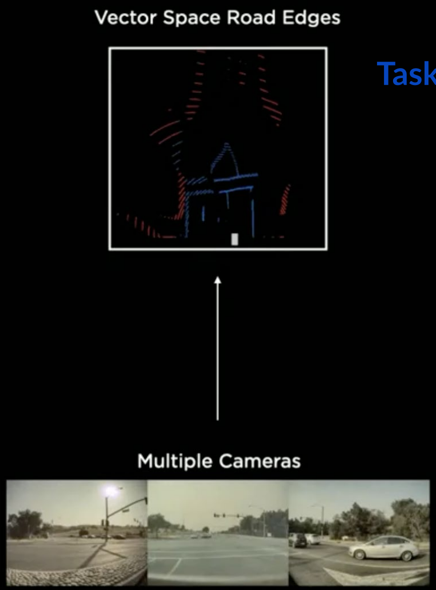
Fusion is difficult as objects span **differently** across images.

Smart Summon - Per-camera detection then fusion (nV)

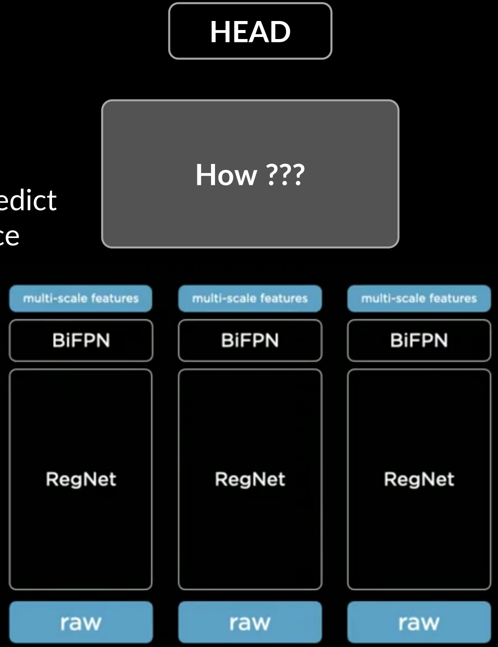
CORE

Goal: summon vehicle to the person nearby

Cast out image-space predictions onto vector space



Task ⟹ **Solution:**
Directly predict vector space results



Caveats

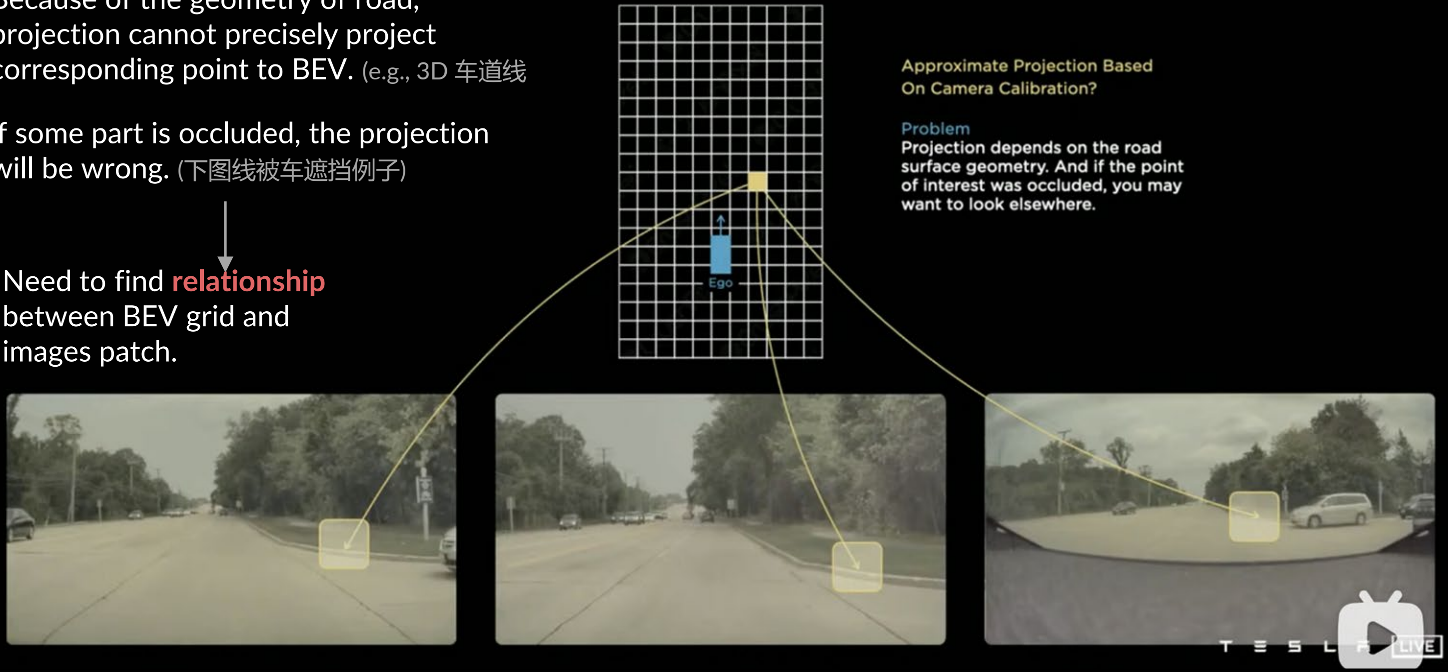
1. How to transform features from image-space to vector space?
   a. differentiable, e2e
   b. camera pose **varies**

1. Vector space dataset
   a. massive labelling (coming up)

Caveat 1

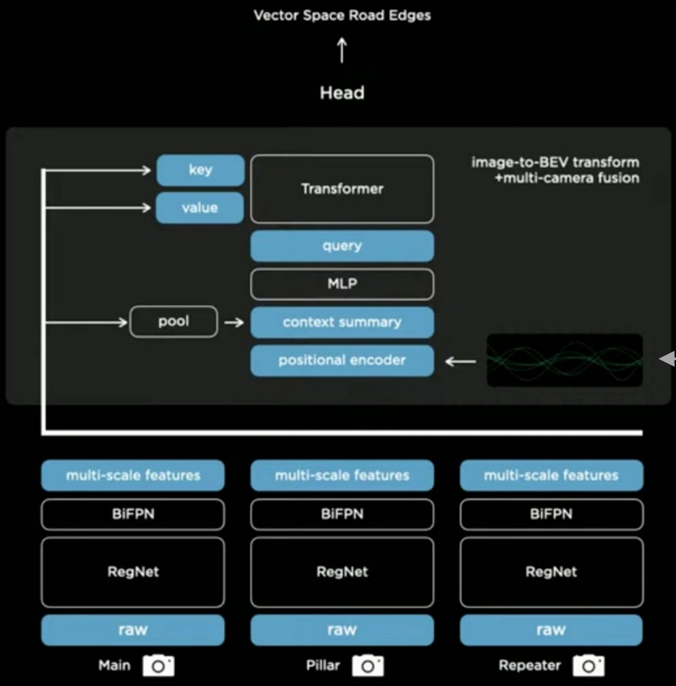
- Because of the geometry of road, projection cannot precisely project corresponding point to BEV. (e.g., 3D 车道线)
- if some part is occluded, the projection will be wrong. (下图线被车遮挡例子)

Need to find **relationship** between BEV grid and images patch.



Attention is all you need,
https://arxiv.org/pdf/1706.03762.pdf 2017

Solution to Caveat 1: Transformer

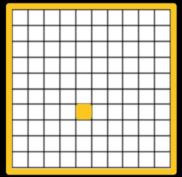


front view

side view

rear view

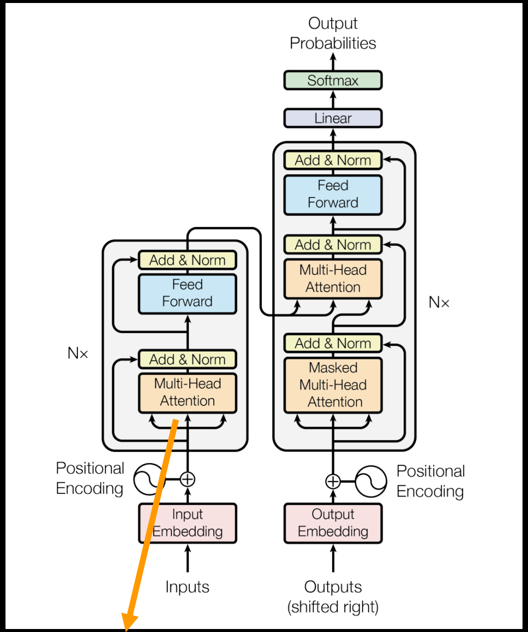
“Raster” position encoding, generate a position encoding vector for every grid on raster.

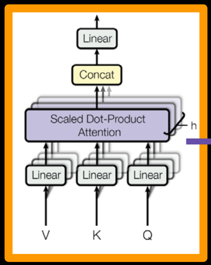


for location *(i,j)* in BEV view

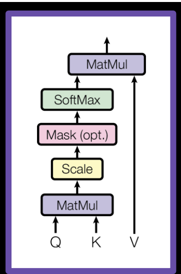
$$A(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Transformer - Model structure





Multi-head attention



Dot-product Attention

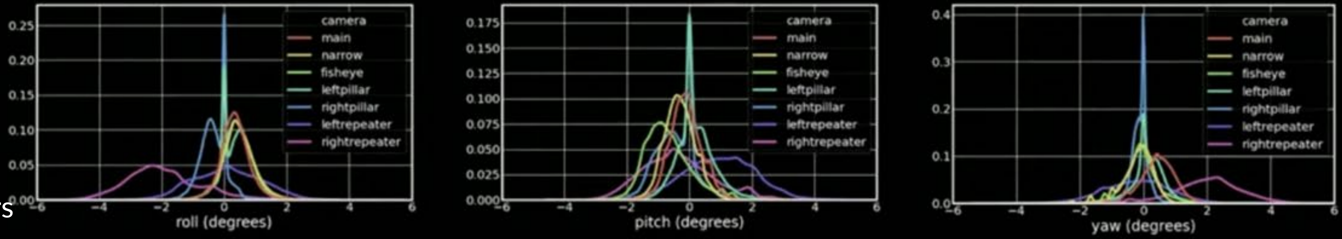
Background on Transformer

- What: a query and a set of key-value pairs to an output
- The output: a **weighted** sum of the **values**, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

Side issue in Caveat 1: Variations in camera

Extrinsics varies in different cars

Neural networks in different cars are the same! But input images are slightly different!



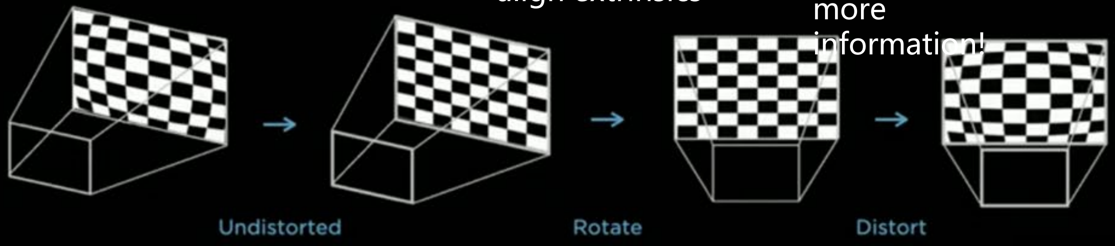
mount error distribution in roll/pitch/yaw

Align cameras parameters

Rectify Images Into a "Virtual Camera"

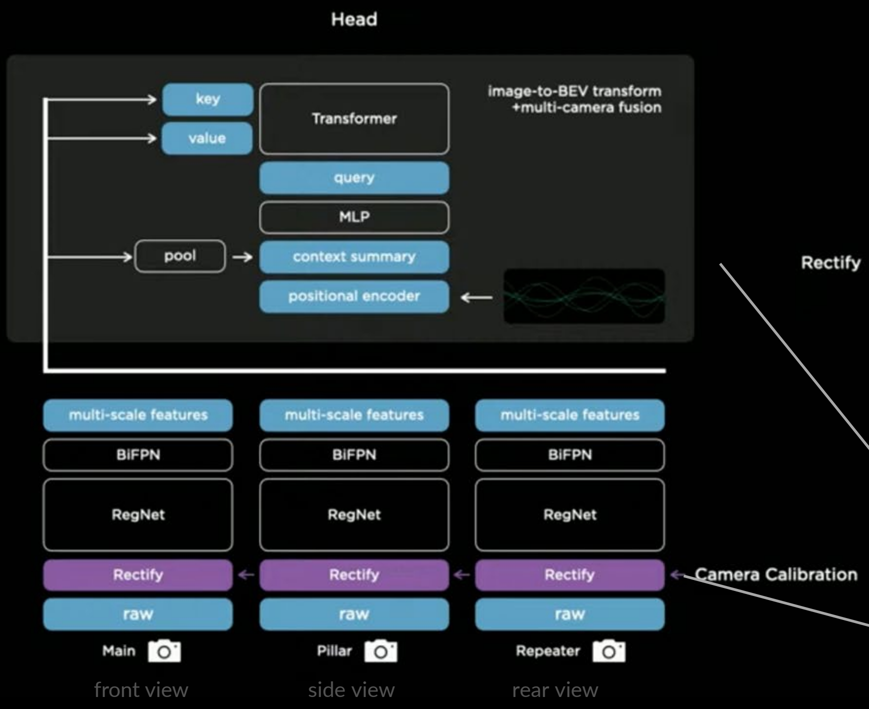
align extrinsics

distort to keep more information!



1. how to determine virtual camera? (Mean of statistics data)
2. why distort again after rectify? (To keep more information)

Rectify to a Common Virtual Camera



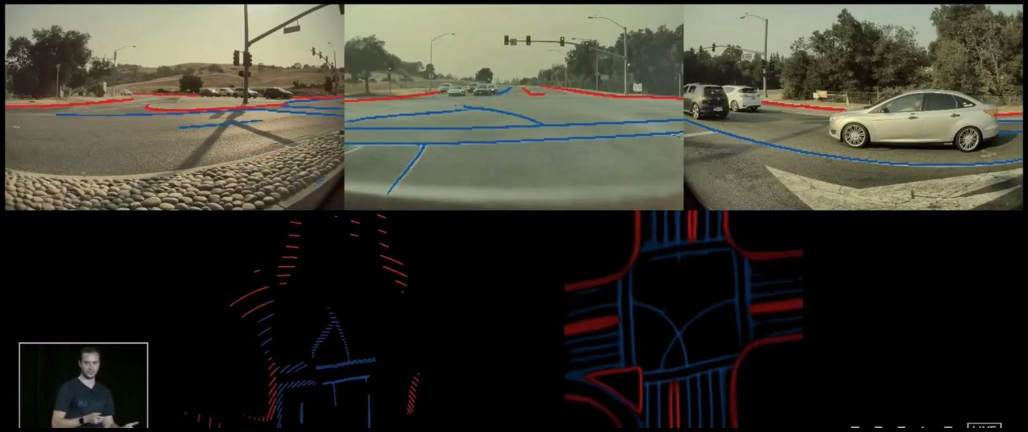
Rectifying images works!

Getting more crisp (清晰) after rectify module

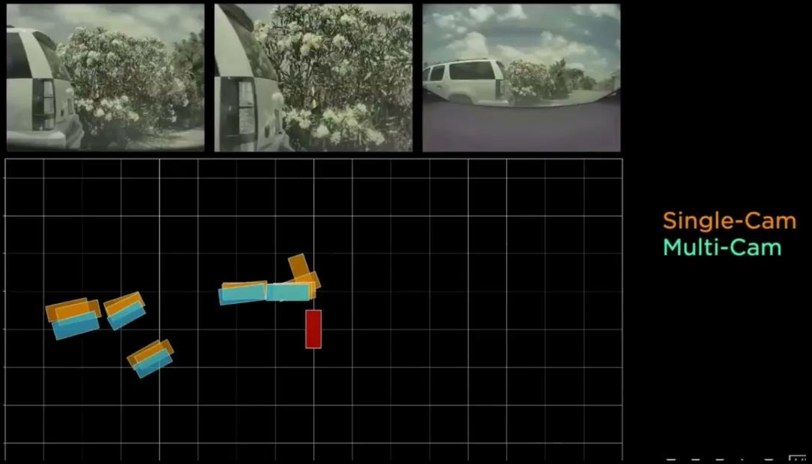
Get rectify layer and transform layer together to complete multi-cam fusion.

Improvement after Transformer and Rectification

Vector Space Edges and Lines



Detections: SingleCam -> MultiCam



Before

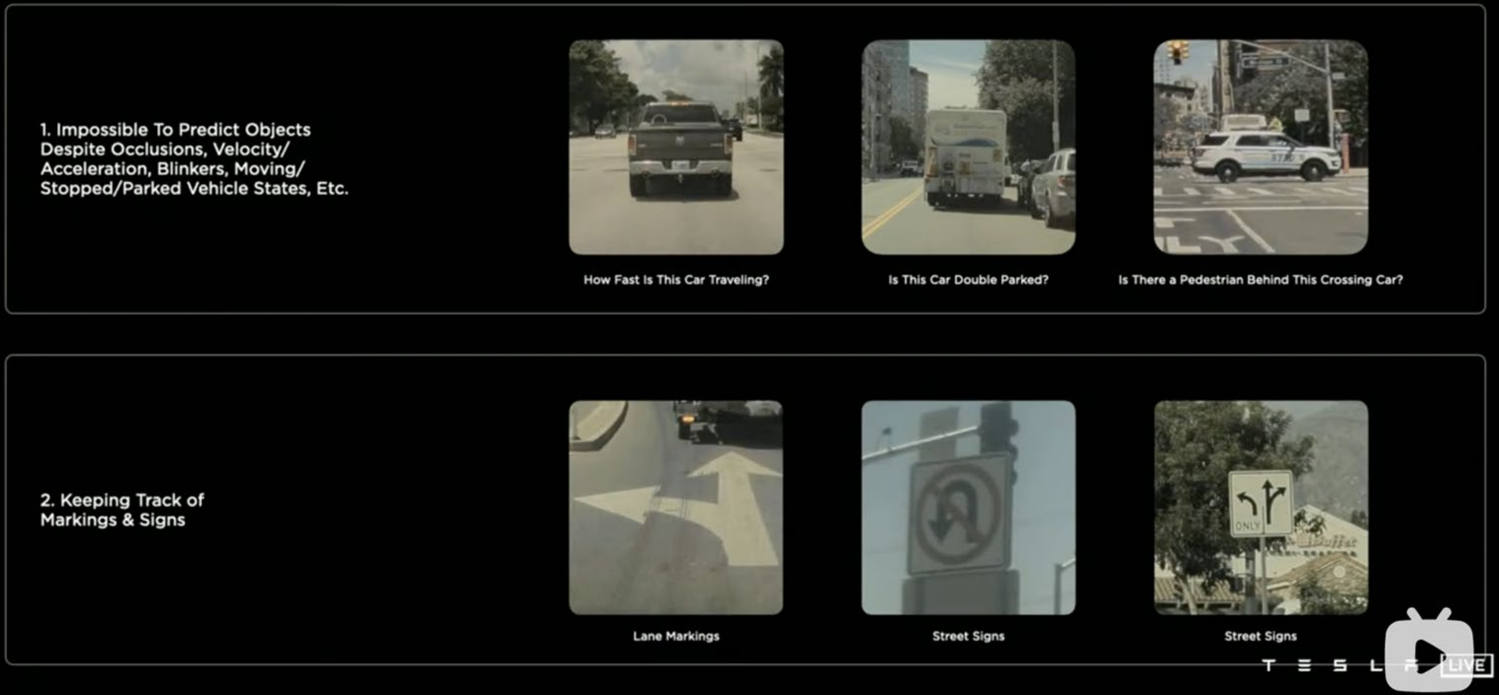
Road edge/curve
Laneline

After (nV, Transformer + Rectify)

It's basically night and day (天差地别)

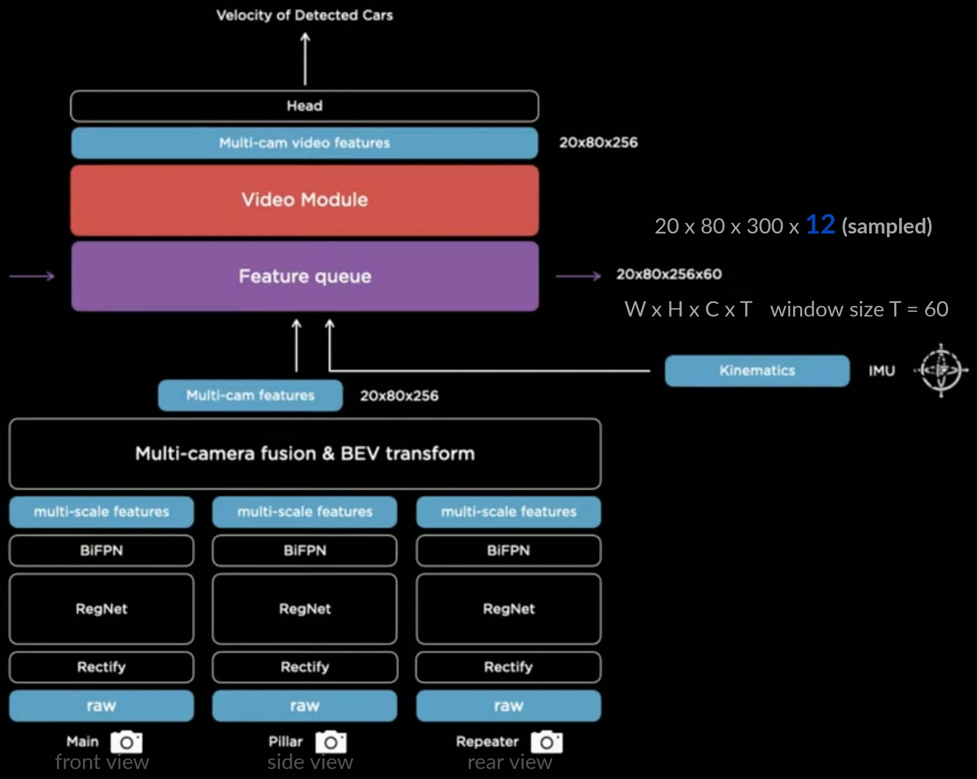
Stepping further - Motivation: Lack of memory

Introducing temporal info

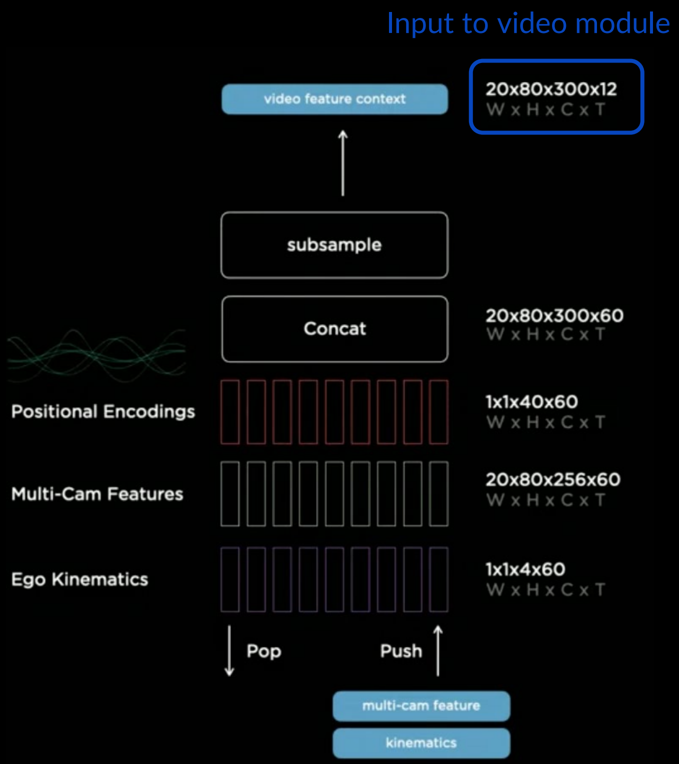


Video Neural Net Architecture

CORE



feature queue

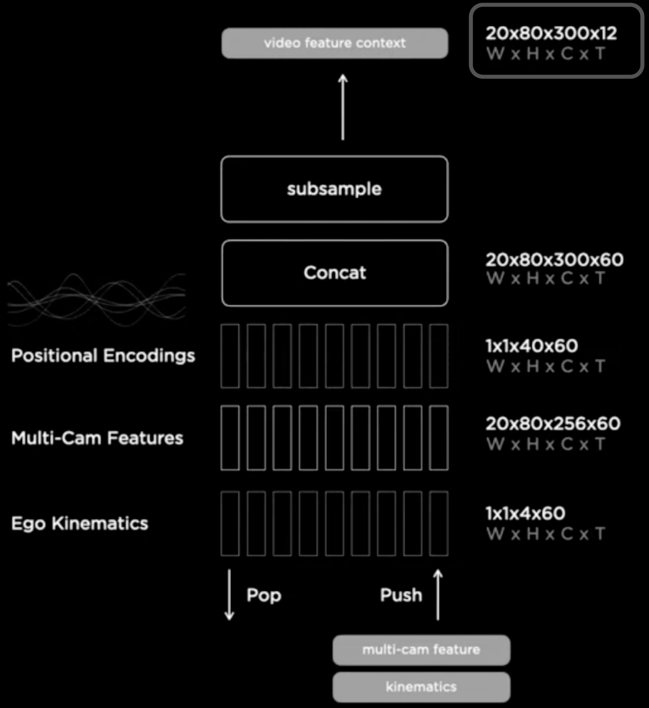


Positional encoding **(40)**: encode (x,y) as does in Transformer paper
Ego kinematics **(4)**: velocity, acc. etc

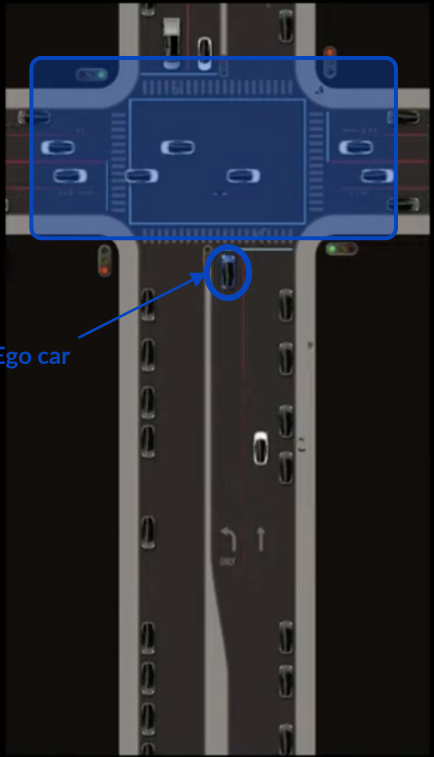
feature queue

Why use/push the queue?

Input to video module



Ego car



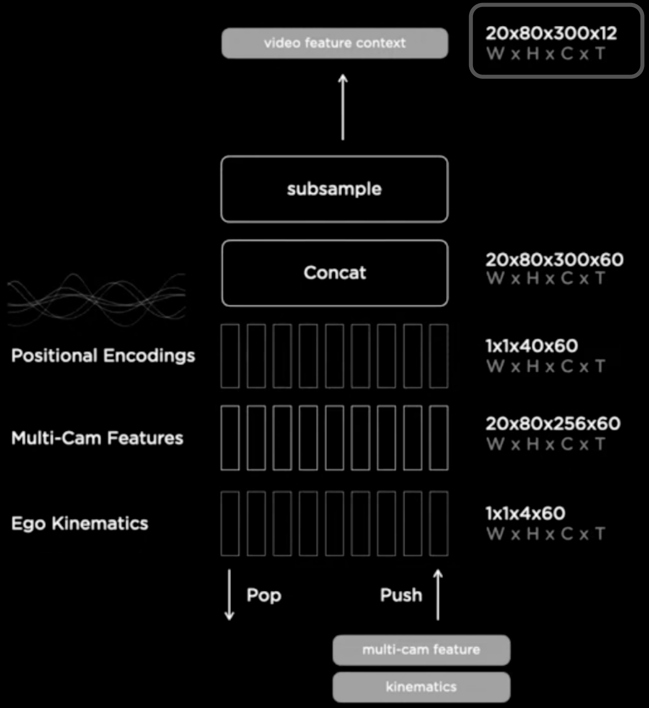
1. Temporary occlusions
=> time-based queue
(e.g. push every 27ms)

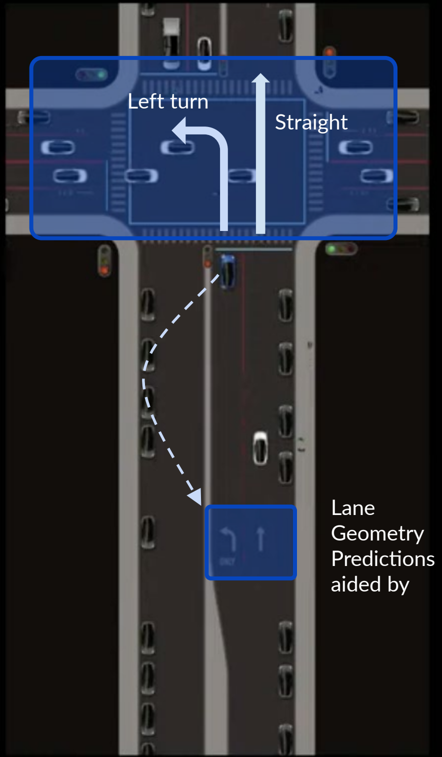
Positional encoding **(40)**: encode (x,y,z) to higher frequency [1]
Ego kinematics **(4)**: velocity, acc. etc

feature queue

Why use/push the queue?

Input to video module





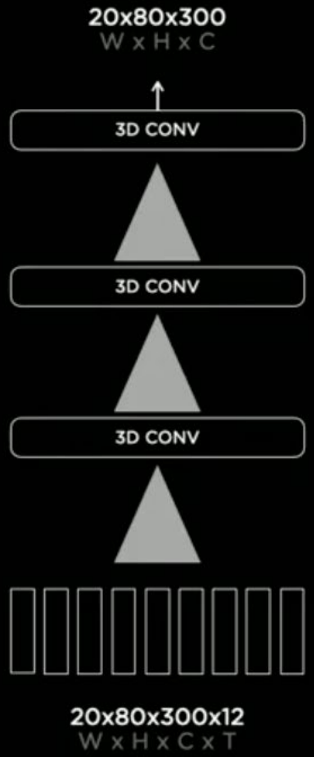
1. Temporary occlusions
=> time-based queue
(e.g. push every 27ms)

2. Signs & Markings Earlier on the Road
=> space-based queue
(e.g. push every 1 meter)

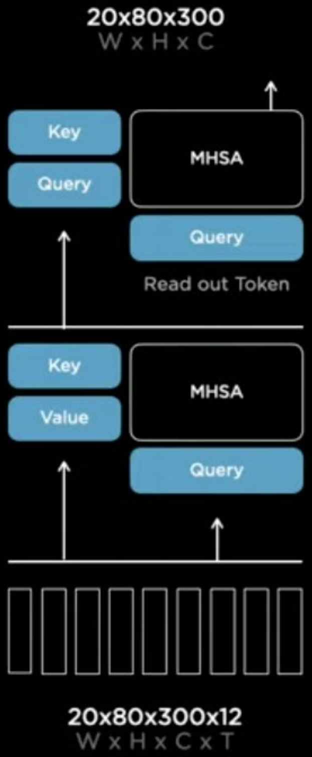
Positional encoding **(40)**: encode (x,y,z) to higher frequency [1]
Ego kinematics **(4)**: velocity, acc. etc

video module

Possible candidates

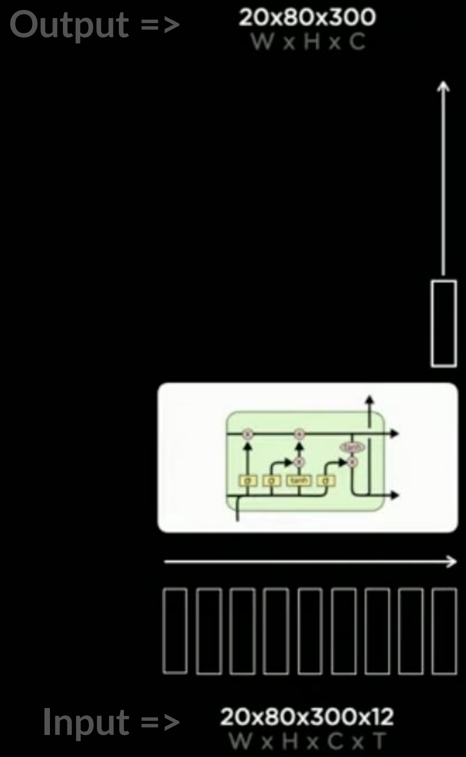


3D conv



Transformer

Axial Transformer [1]



Spatial RNN

[1] https://arxiv.org/abs/1912.12180

video module

Spatial RNN

Hidden state h(t-1)
W x H x C

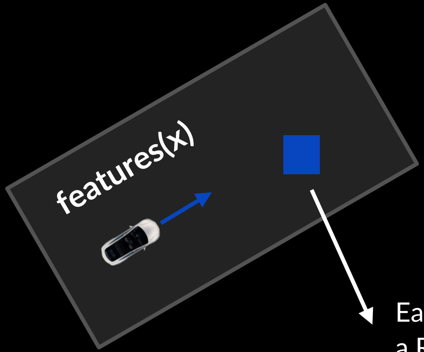
Input x(t)

kinematics

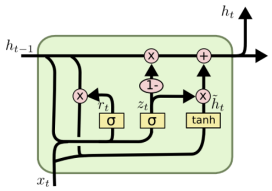
features

20 x 80 x 256
Ego Coordinate System

Spatial Feature Grid: h(t)
W x H x C

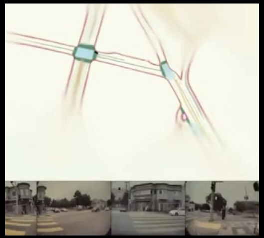


Each cell is a RNN



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

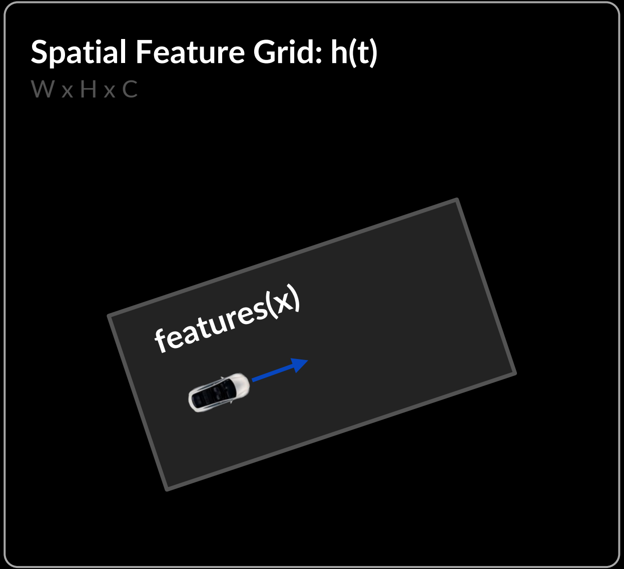
Output h(t)
W x H x C

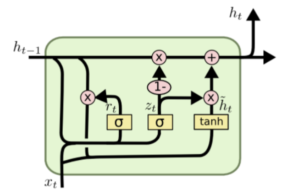


- 尺寸不一致（300/256）
- 20 x 80 - 我们的理解

video module

Spatial RNN





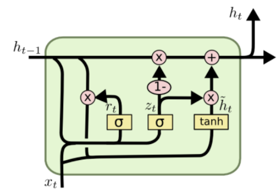
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Only update RNN at the points where they are **nearby** the ego car

- to save computational cost

video module

Spatial RNN



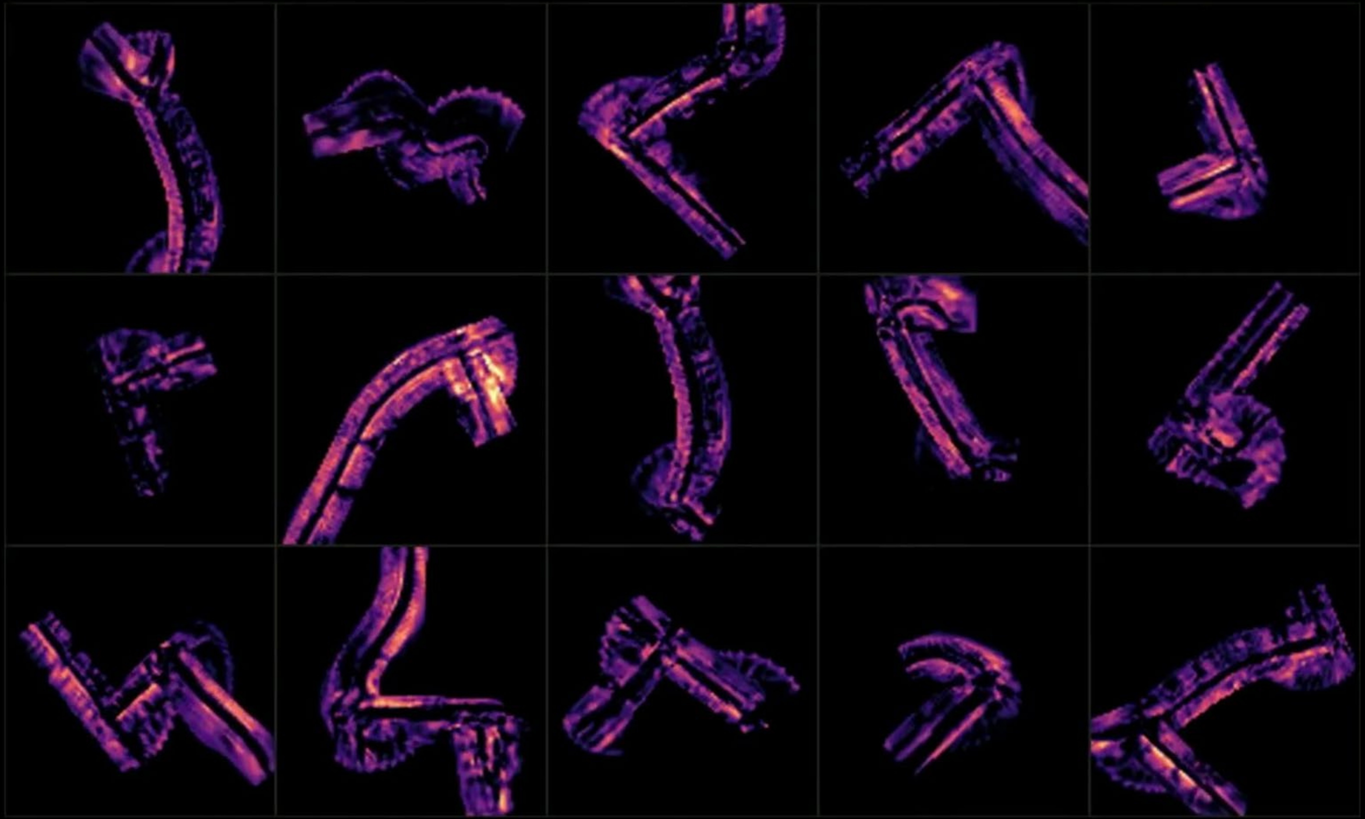
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Only update RNN at the points where they are **nearby** the ego car

- to save computational cost

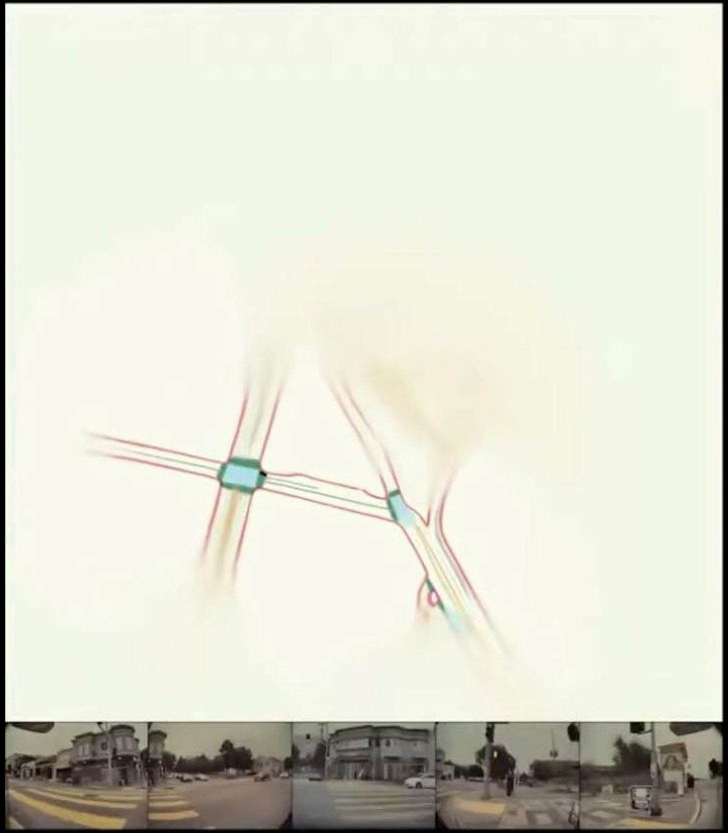
video module

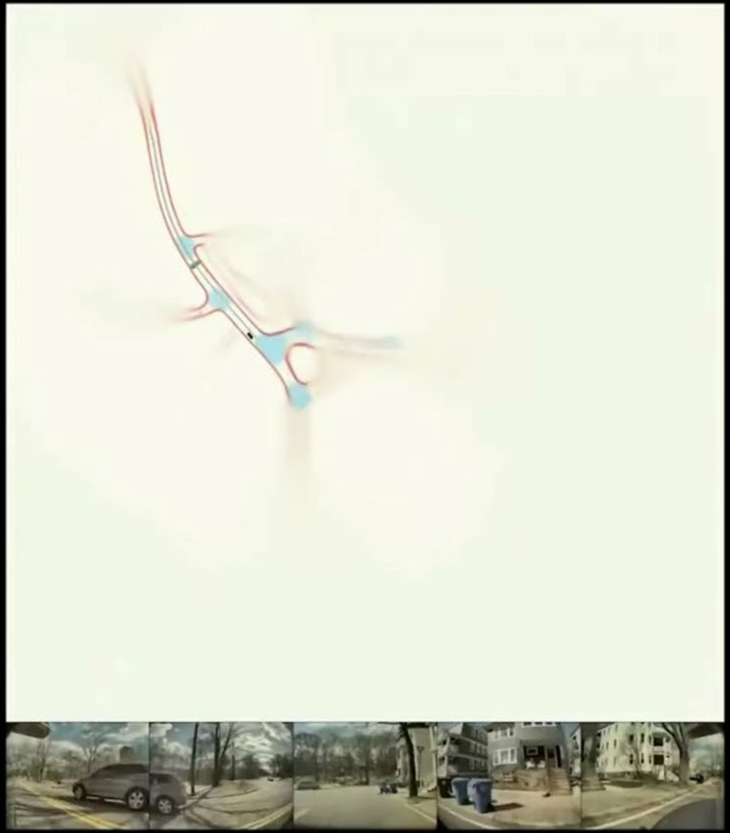
Spatial RNN - Feature Channel Visualization



video module

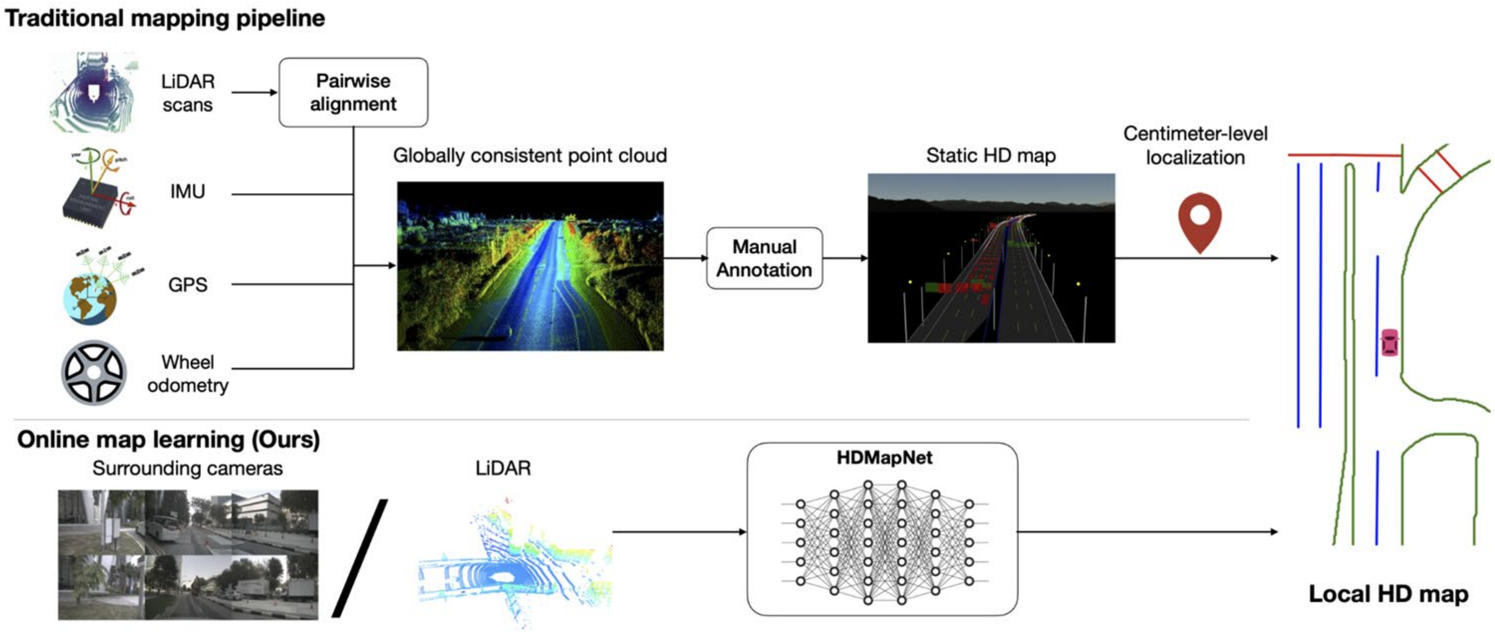
Spatial RNN - Road reconstruction





Road reconstruction(HDMap Net)

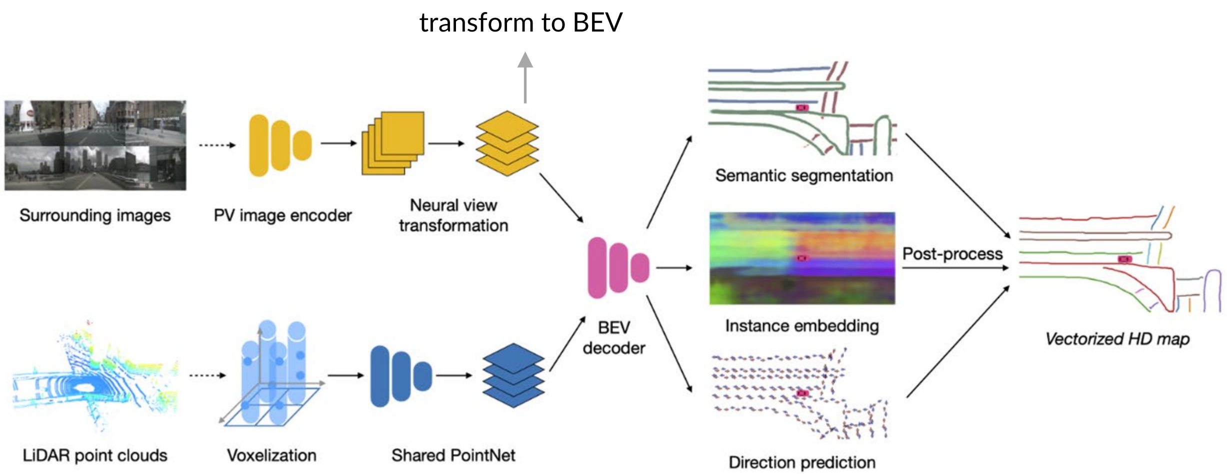
Predicte HD map directly from images/lidar data



Li, Qi, et al. "HDMapNet: An Online HD Map Construction and Evaluation Framework." *arXiv preprint: 2107.06307*

Road reconstruction(other solution)

Predicte HD map directly from images/lidar data



HDMapNet architecture[1]

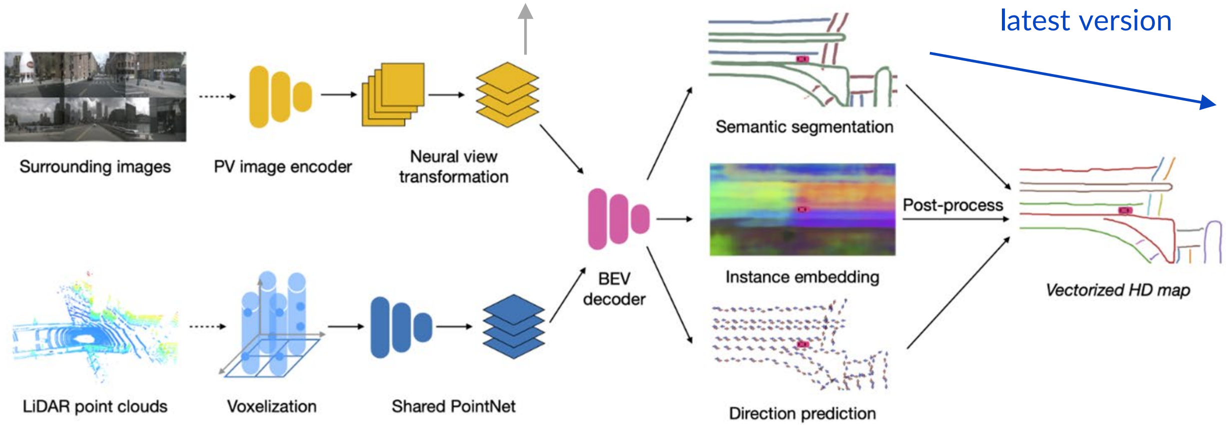
BEV feature [1]

[1] Li, Qi, et al. "HDMapNet: An Online HD Map Construction and Evaluation Framework." *arXiv preprint: 2107.06307*

Road reconstruction(other solution)

Predicte HD map directly from images/lidar data

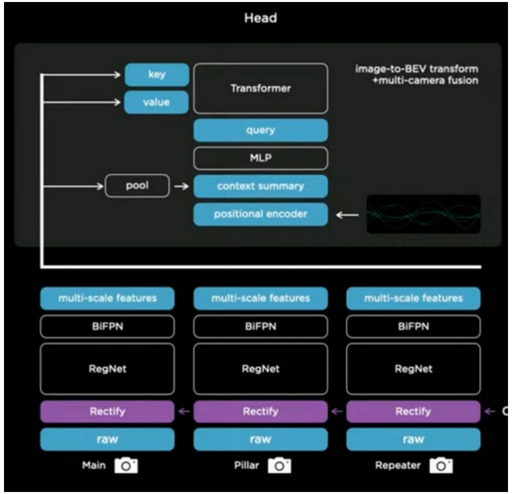
transform to BEV



HDMapNet architecture[1]

similar to Tesla latest version

spatial RNN added temporal and space information



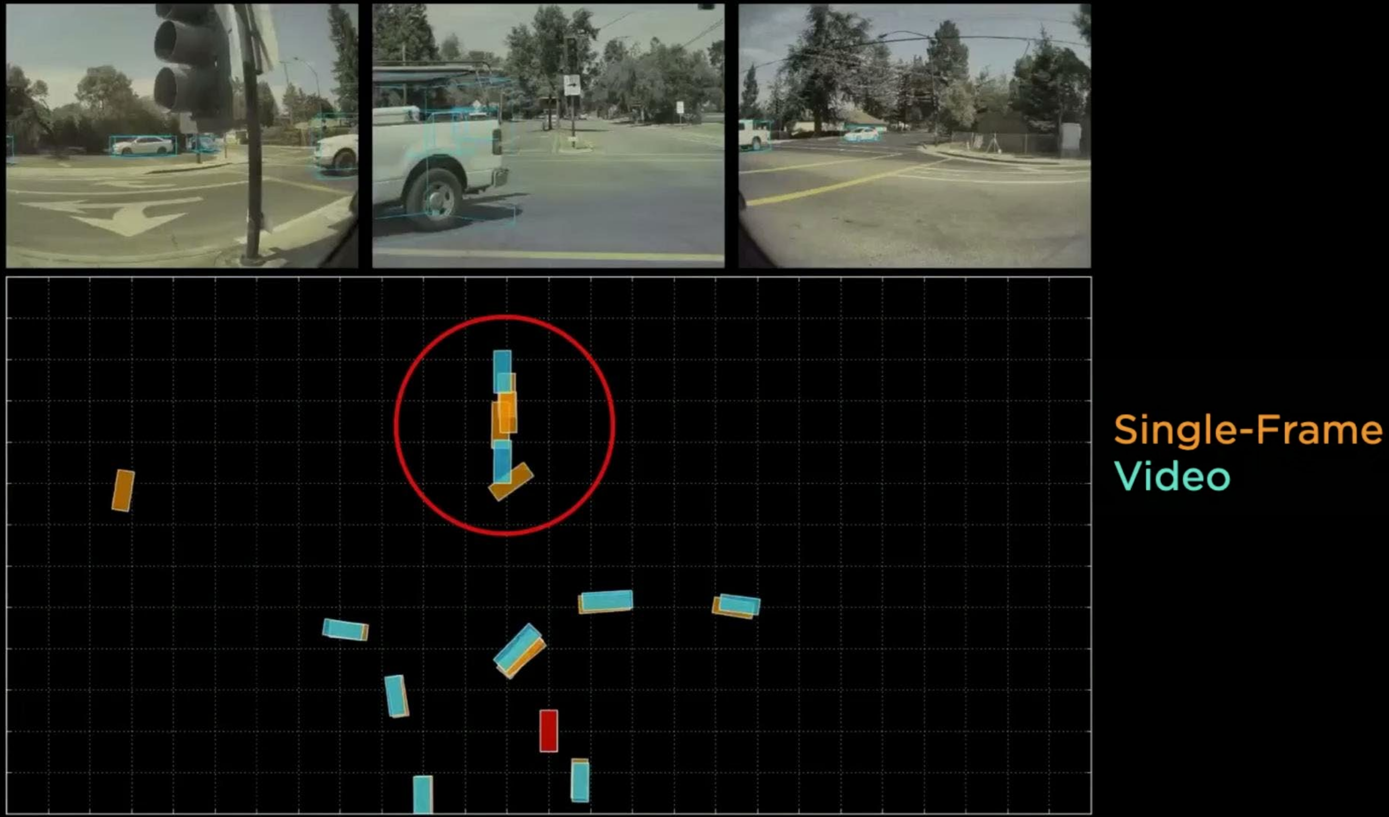
[1] Li, Qi, et al. "HDMapNet: An Online HD Map Construction and Evaluation Framework." *arXiv preprint: 2107.06307*

Road reconstruction(HDMap Net)

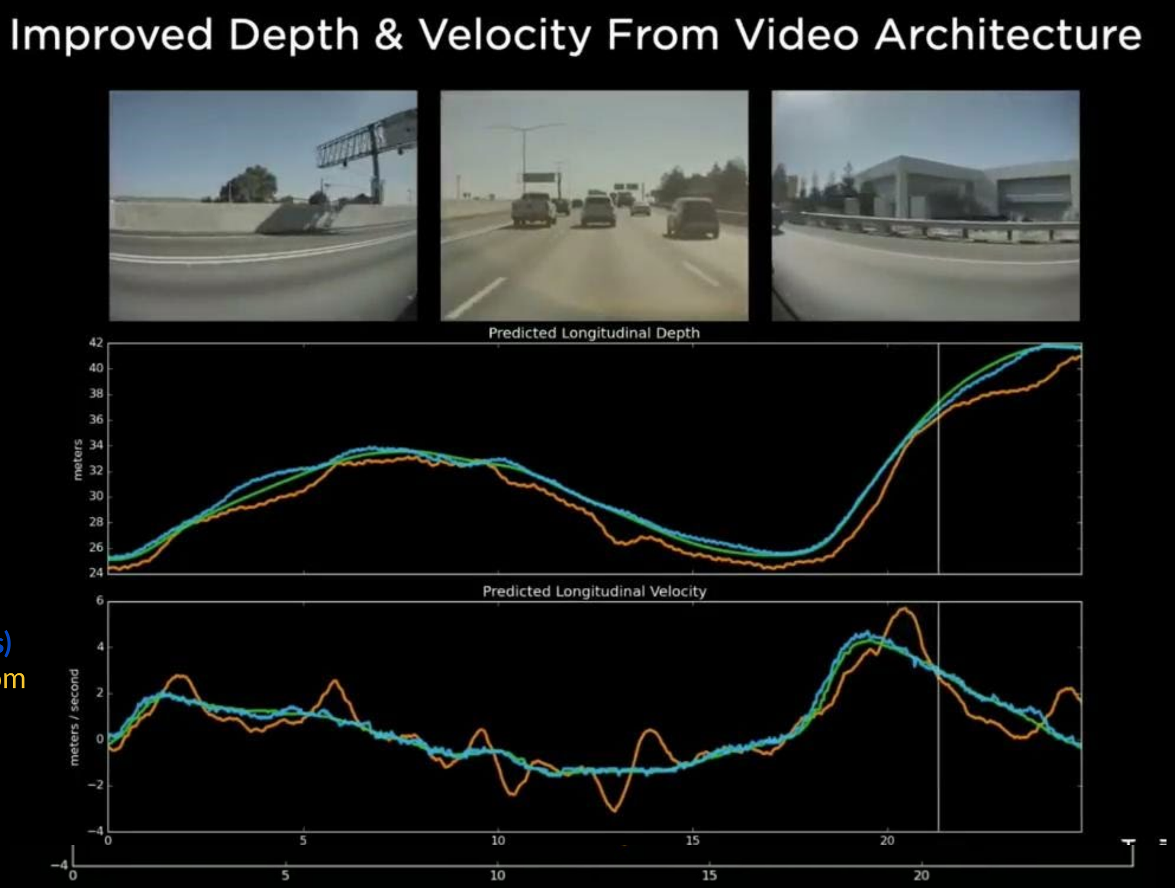
HDMapNet experiment

Li, Qi, et al. "HDMapNet: An Online HD Map Construction and Evaluation Framework." *arXiv preprint: 2107.06307*

Object Detection - Improved Robustness to Temporary Occlusion



Improved Depth & Velocity from Video Architecture



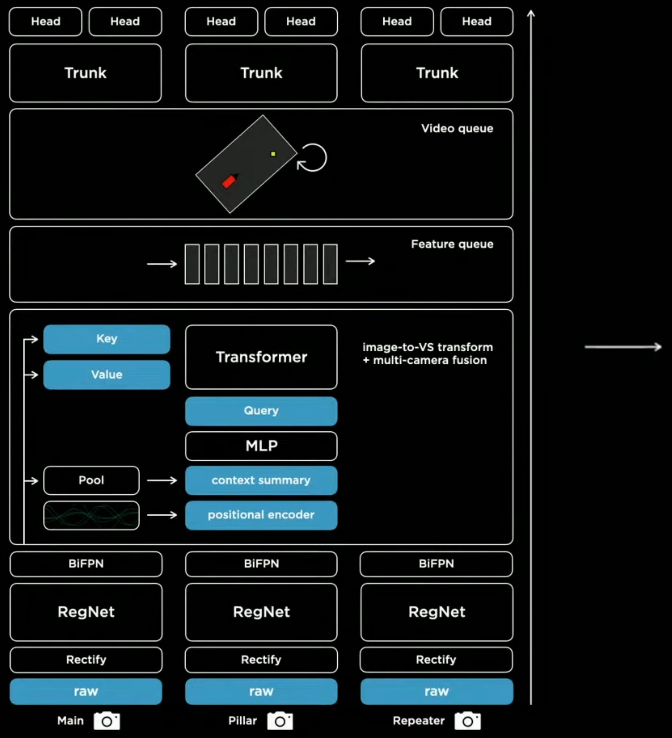
LEGEND

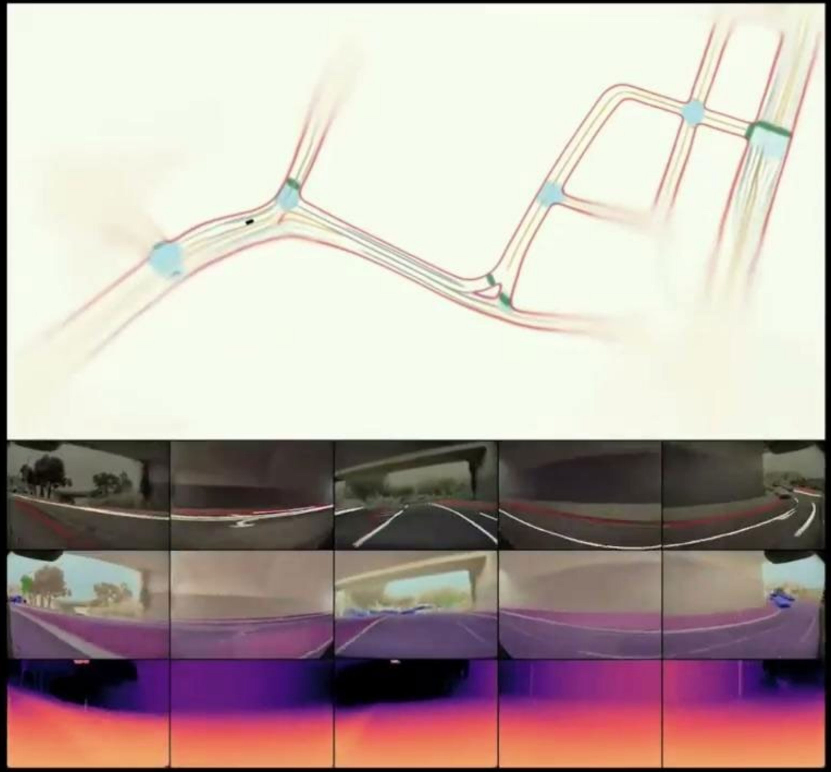
Radar signal (GT)

Video architecture (Ours)

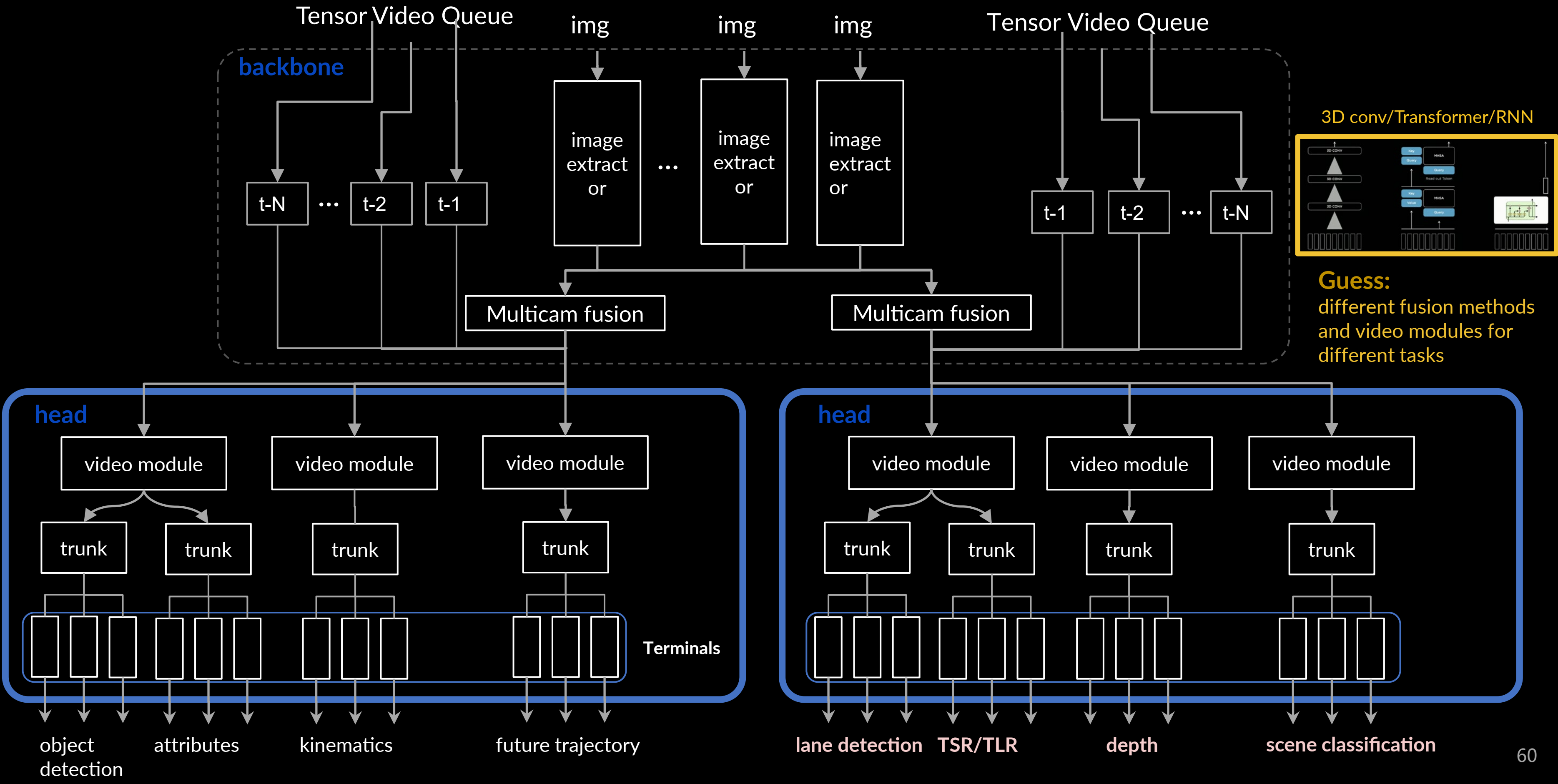
Single frame (velocity from differentiable)

Putting everything together (in Tesla AI Day)

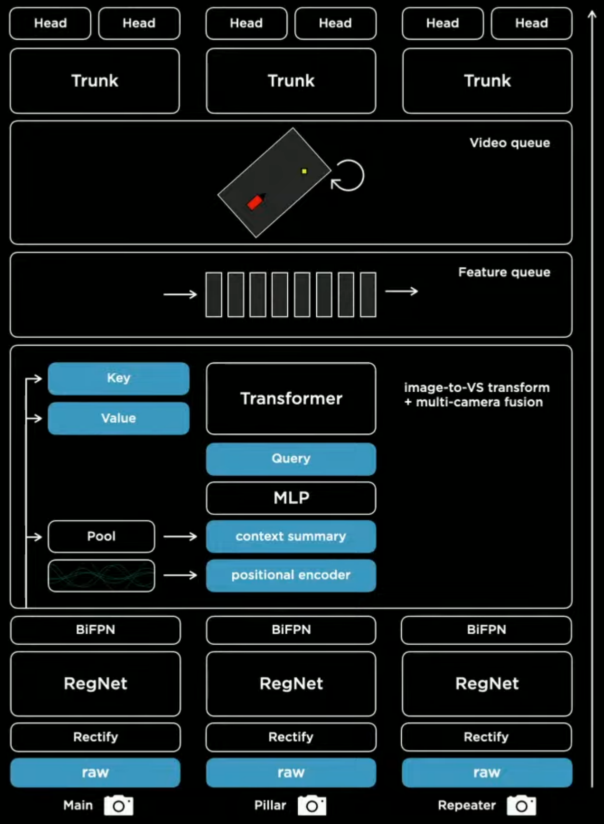




Putting everything together (in CVPR 2021 workshop)



Future Work

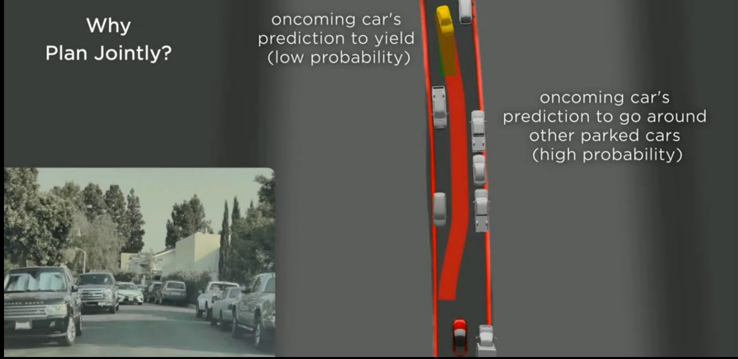


1. Fusion (time & space) at early stage
   a. use cost volumes to obtain optimal flow at the bottom
2. Dense raster outputs, which is expensive (to compute)
   a. since we are under strict latency requirements
   b. aim to produce **sparse** structure of the road

Overview

- Tesla vision
- **Planning and Control**
- Manual/Auto Labelling
- Simulation
- Hardware Integration/Infrastructure
- Dojo

Core issues in Planning



Action Space Is:

1. Non-Convex
   a. Discrete Search
   b. Continuous Function Optimization -- Can Converge to Local Minima

2. High-Dimensional
   a. Discrete Search -- Computationally Intractable
   b. Continuous Function Optimization

Solution:

Hybrid Planning System

Route Planner: a comparison

VS

Baidu Apollo 6.3 EM

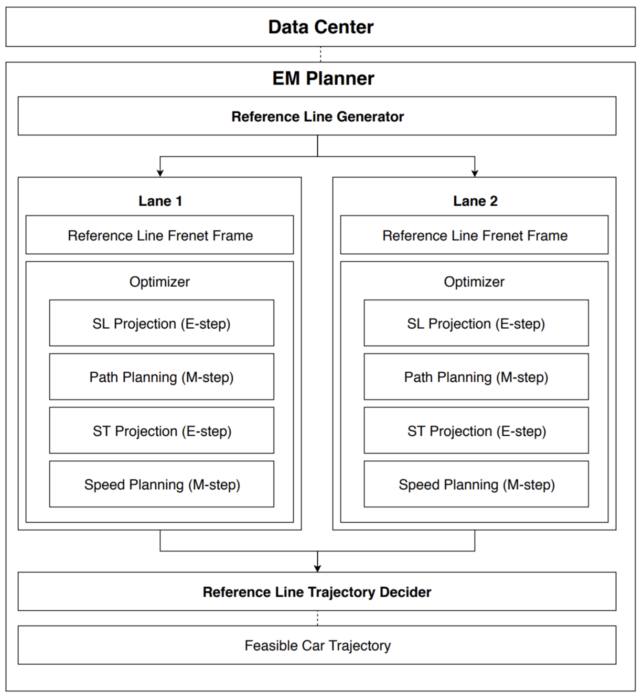
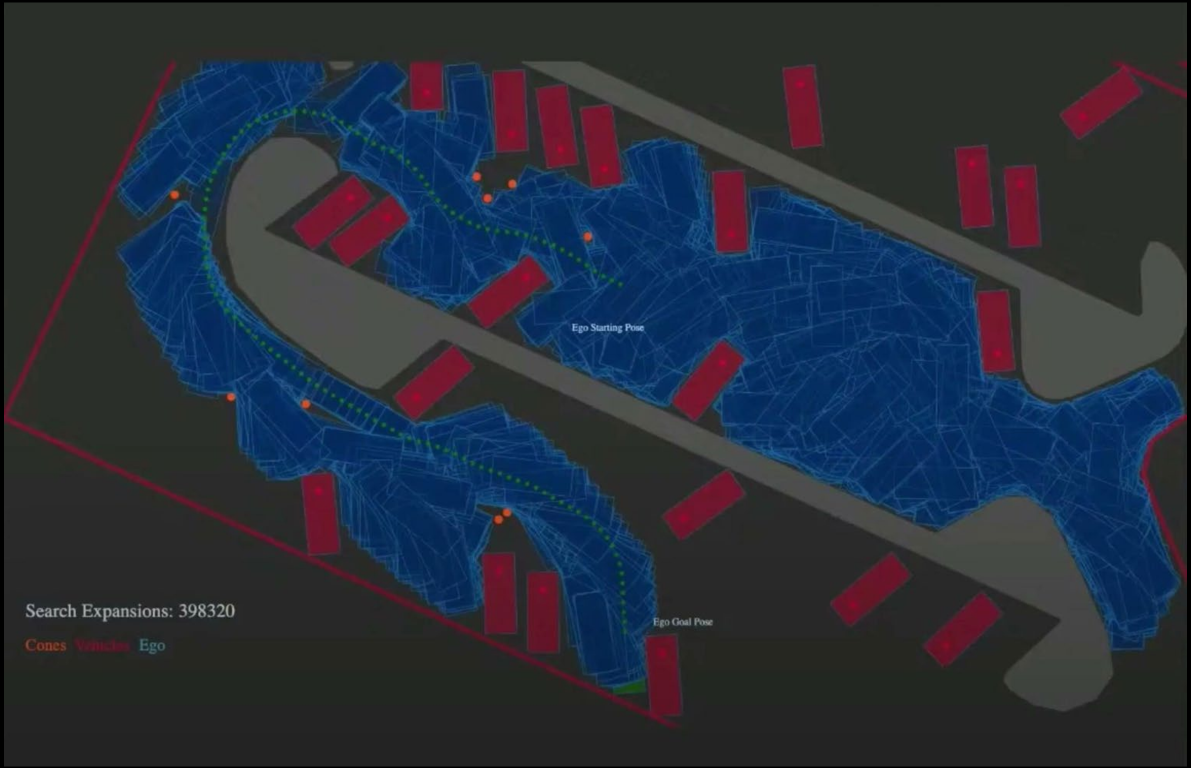


Fig. 2: EM Framework

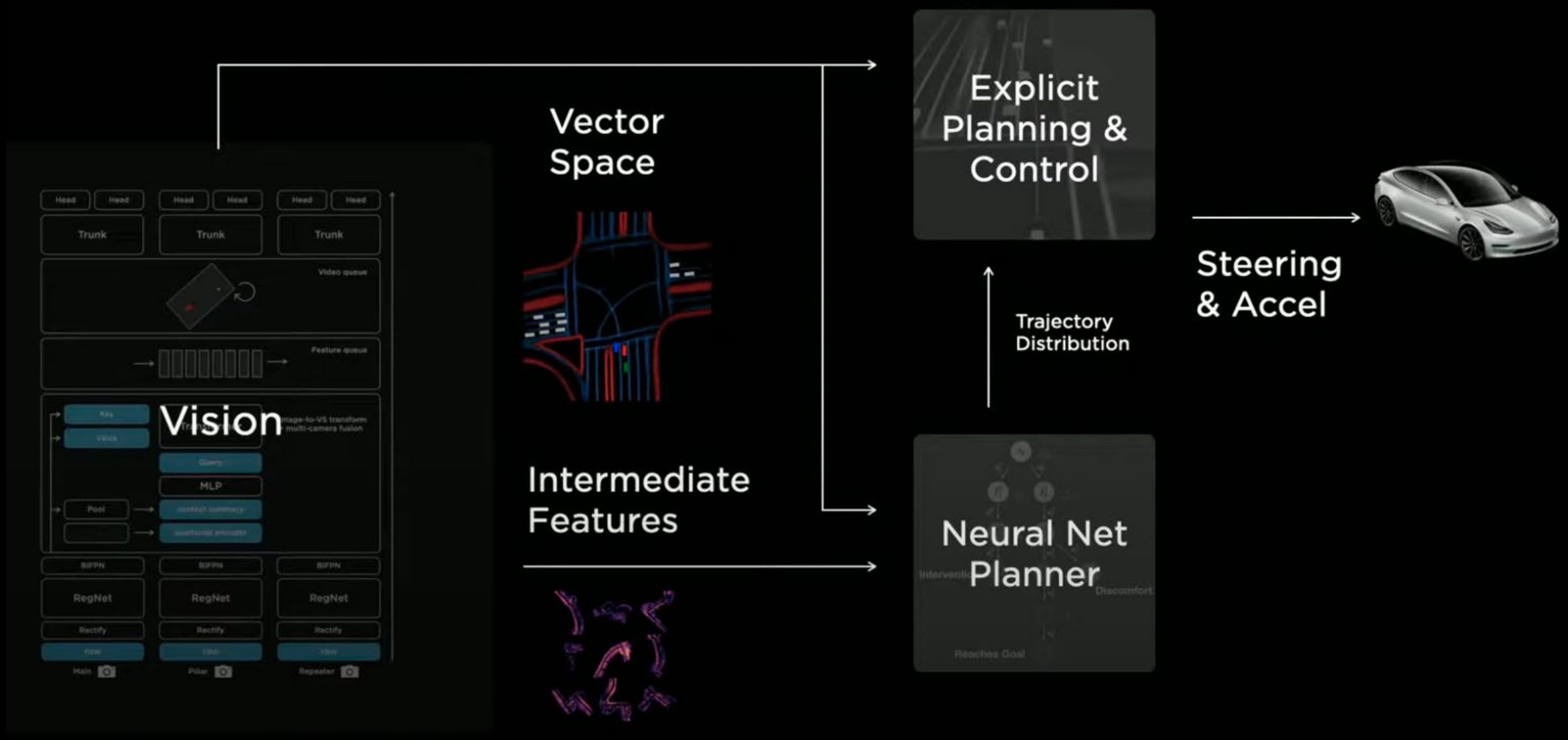
Fan, Haoyang, et al. "Baidu apollo em motion planner." arXiv preprint arXiv:1807.08048 (2018).

AVP Route Searching: a comparison

Search Space: 5 Constant Curvature Arcs			
Algorithm	A*	A*	MCTS Argmax Sampling
Search Heuristic	Euclidean Distance to Goal	Euclidean + Navigation	Neural Network Policy & Value Function
Number of Expansions	398,320	22,224	288



Final architecture



Overview

- Tesla vision
- Planning and Control
- **Manual/Auto Labelling**
- Simulation
- Hardware Integration/Infrastructure
- Dojo

数据很重要

To get any neural network signal to work need:

1. **Large** (millions of videos)
2. **Clean** (labeled data, here: depth, velocity, acceleration)
3. **Diverse** (a lot of edge cases, not just nominal/"boring" scenarios)

dataset.

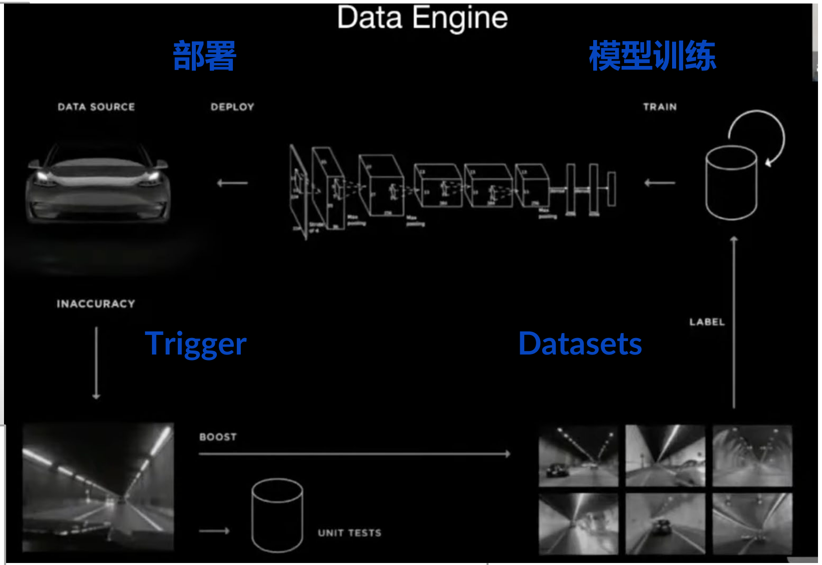
And train a large enough neural network on it.

7
rounds of shadow mode

1 million
8-camera 36fps 10-second videos
(of highly diverse scenarios)

6 billion
object labels,
with accurate depth/velocity

1.5 petabytes



Triggers

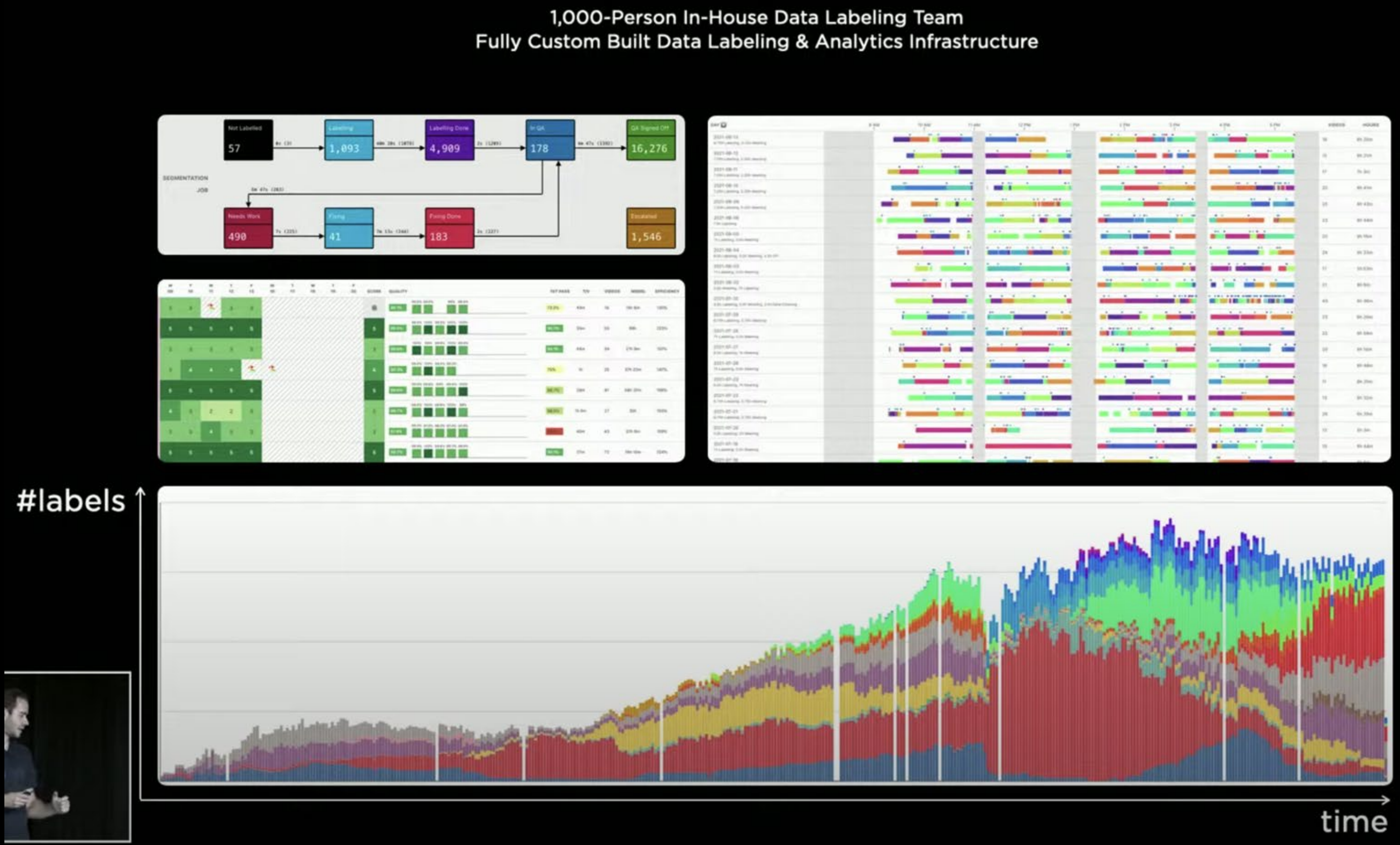
Developed and maintained 221 triggers. E.g.:

- radar vision mismatch
- bounding box jitter
- detection flicker
- detection in Main camera but not Narrow camera
- driver didn't break but tracker thinks CIPV is rapidly decelerating
- break lights are detected as on but acceleration is positive
- rarely high/low velocity or acceleration
- CIPV cuts in / cuts out
- CIPV has high lateral velocity
- bounding-box derived depth disagrees with network-predicted depth
- rarely sloping road surface (hillcrest or dip)
- rarely sharp turning road surface
- driver breaks sharply on the highway
- stop an go traffic
- Main or Narrow or both cameras appear to be blinded
- driver enters/exits tunnel
- objects on the roof (e.g. canoes)
- driver breaks harshly and there is a VRU cloys to us but there is no intersection
- motorcycle on the highway at night

For details, see CVPR workshop video.

Data Labelling Growth:

bring the labelling **in house** instead of third-party



Data Labeling:

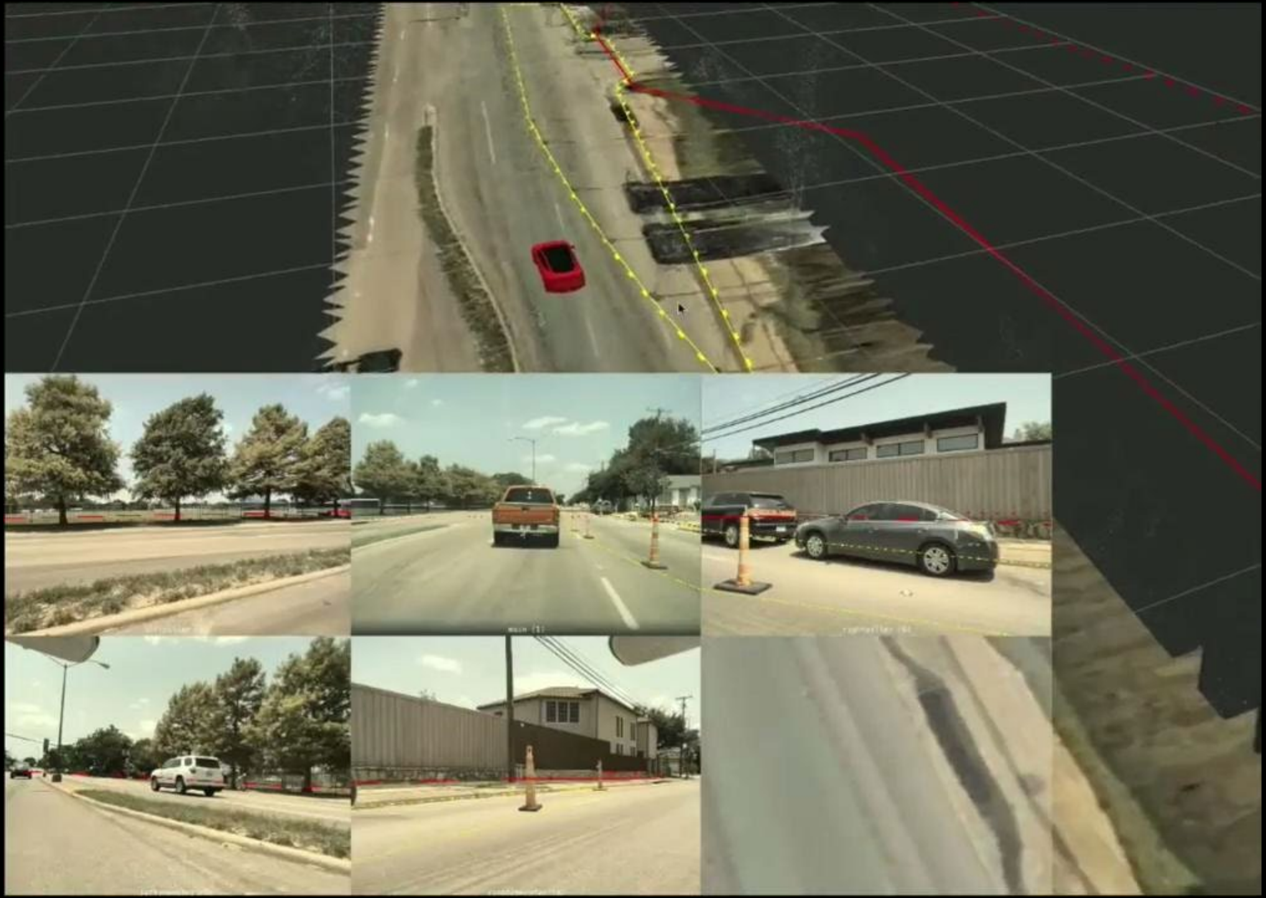
- infrastructure
- labelers

2D image labelling

2D annotation is not sufficient

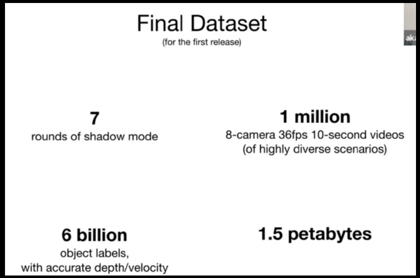
not gonna cut it!

4D Space + Time Labelling



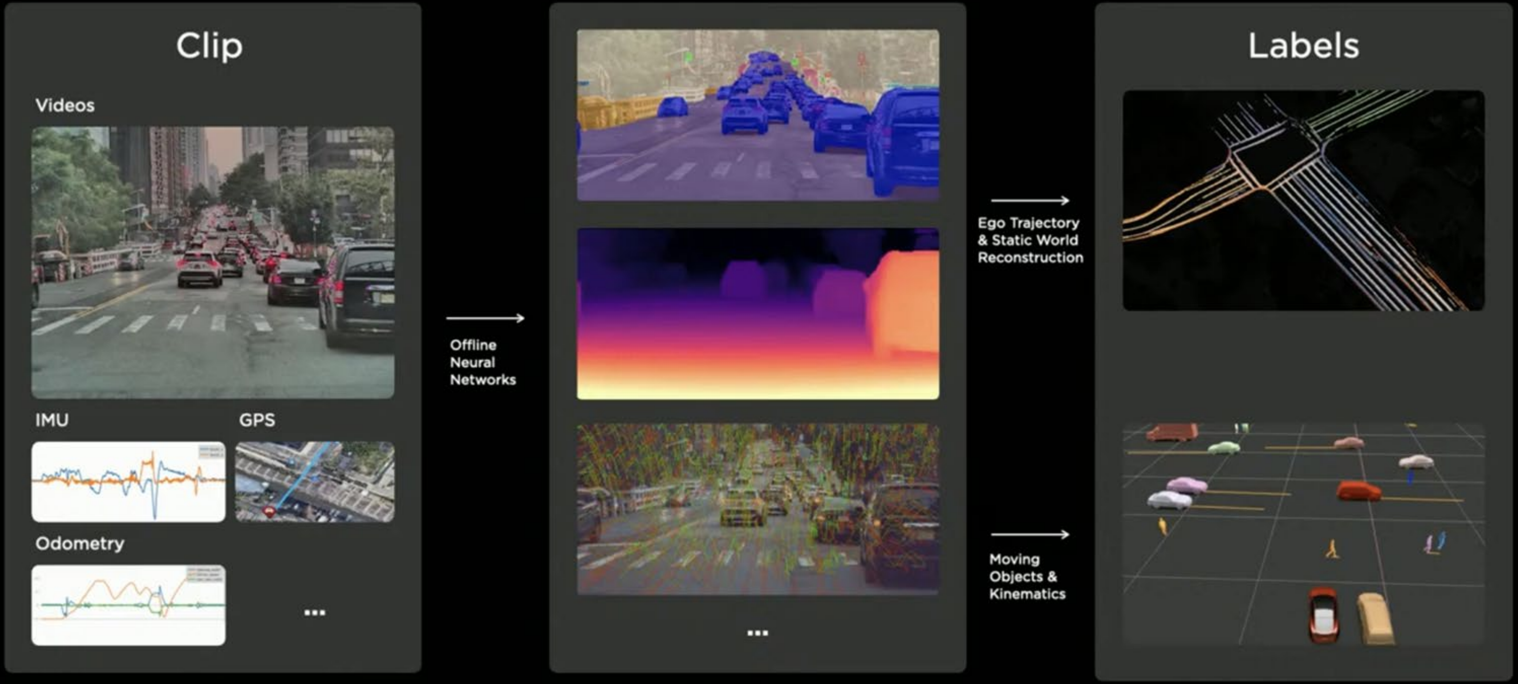
Label in vector space. Project to different cameras

thus obtain 8x data!



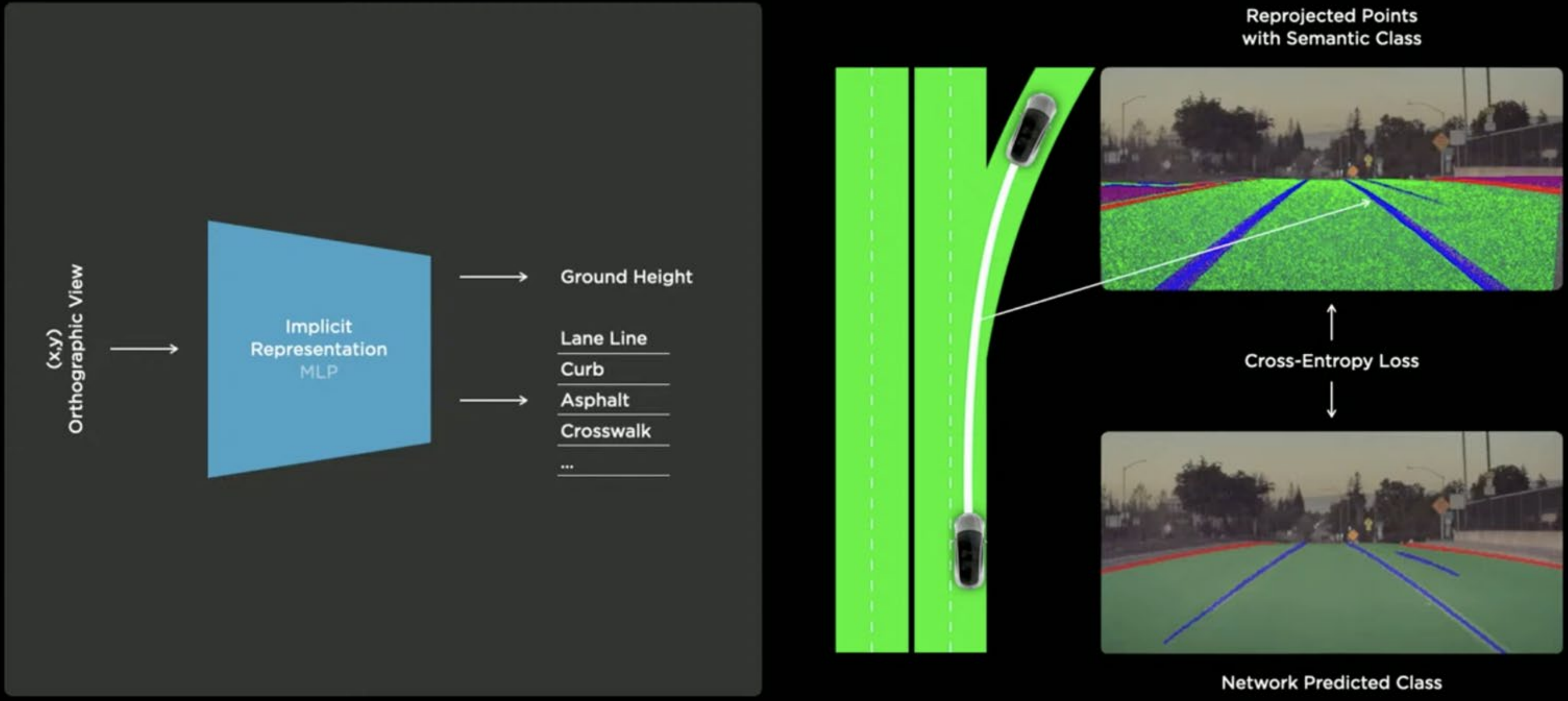
Auto Labelling

Life of a Clip

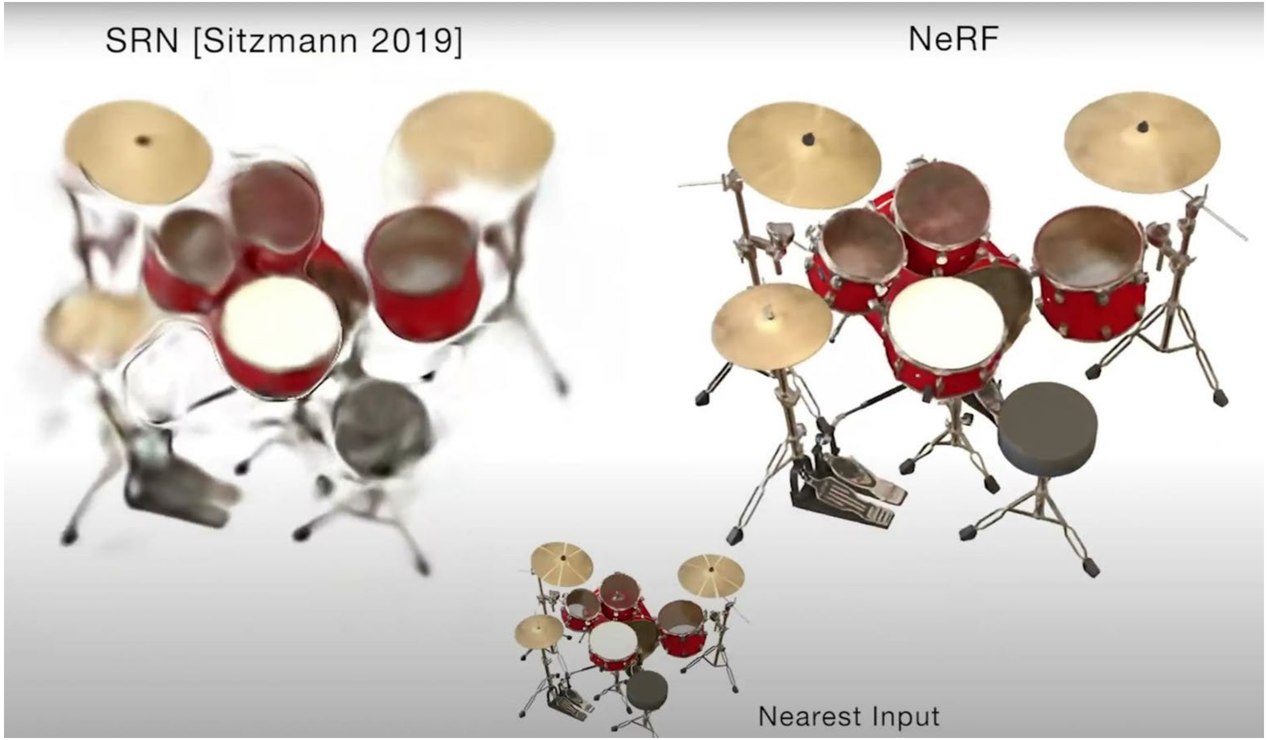


Restructuring the Road - Idea

CORE

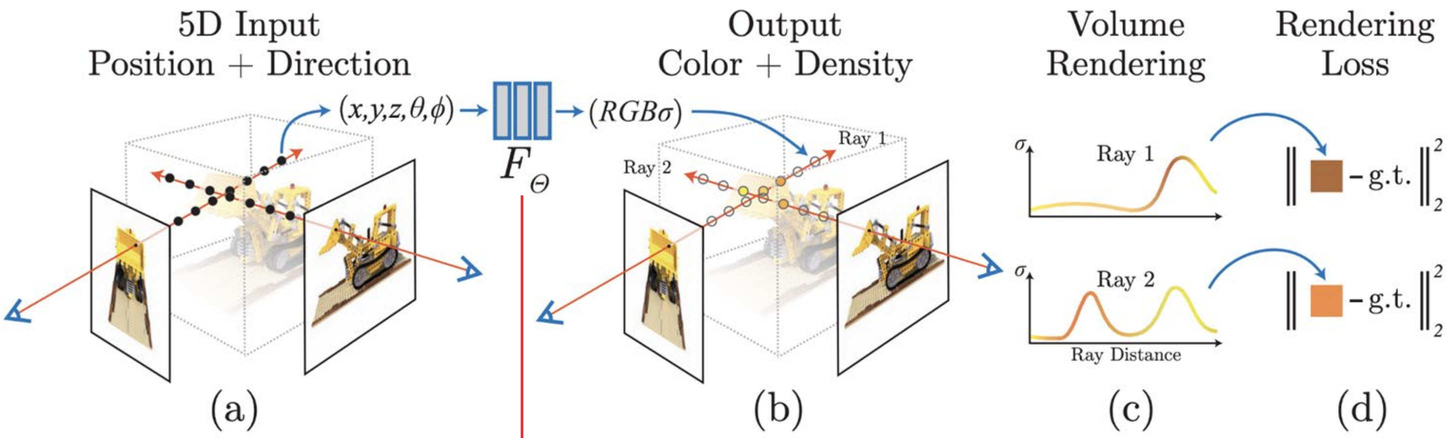


Prerequisite the Paper: NeRF (ECCV 2020)



Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *ECCV*, 2020.

Prerequisite the Paper: NeRF (ECCV 2020)

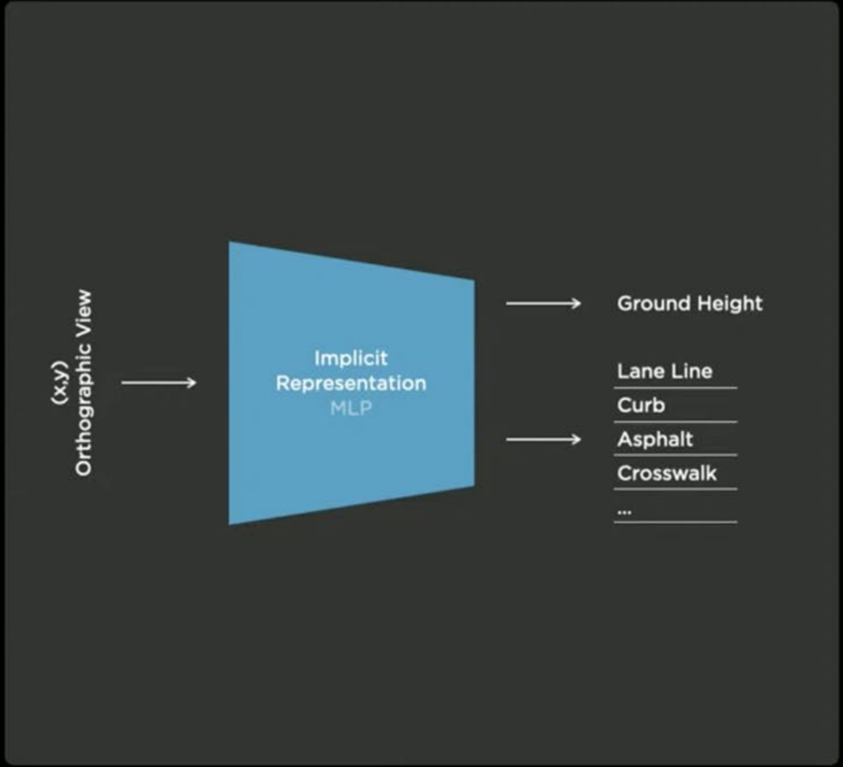


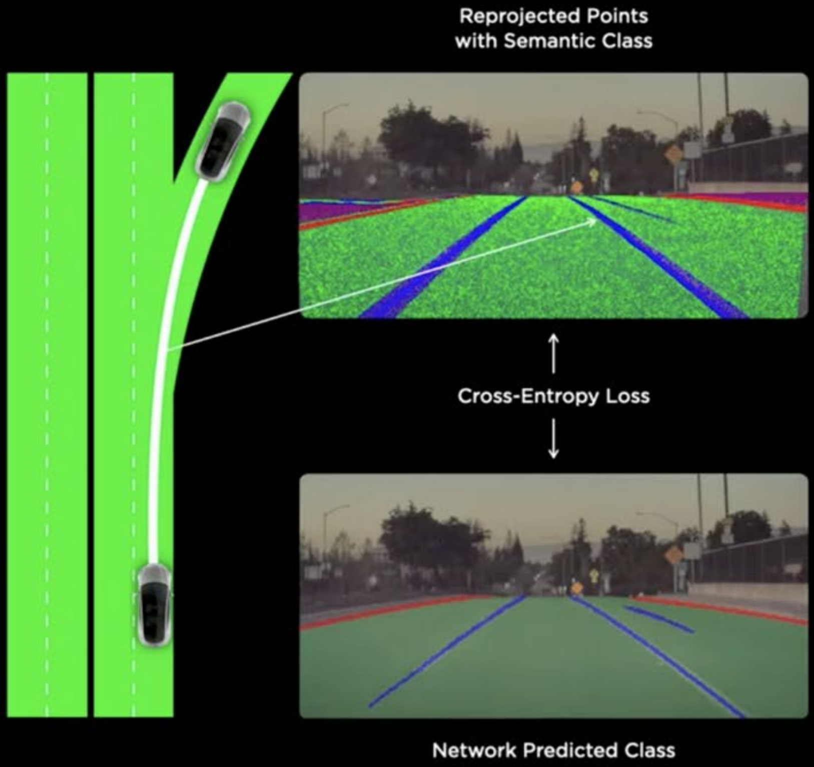
MLP implicitly represent scene.

Whole pipeline is differentiable. So the MLP can trained using backpropagation

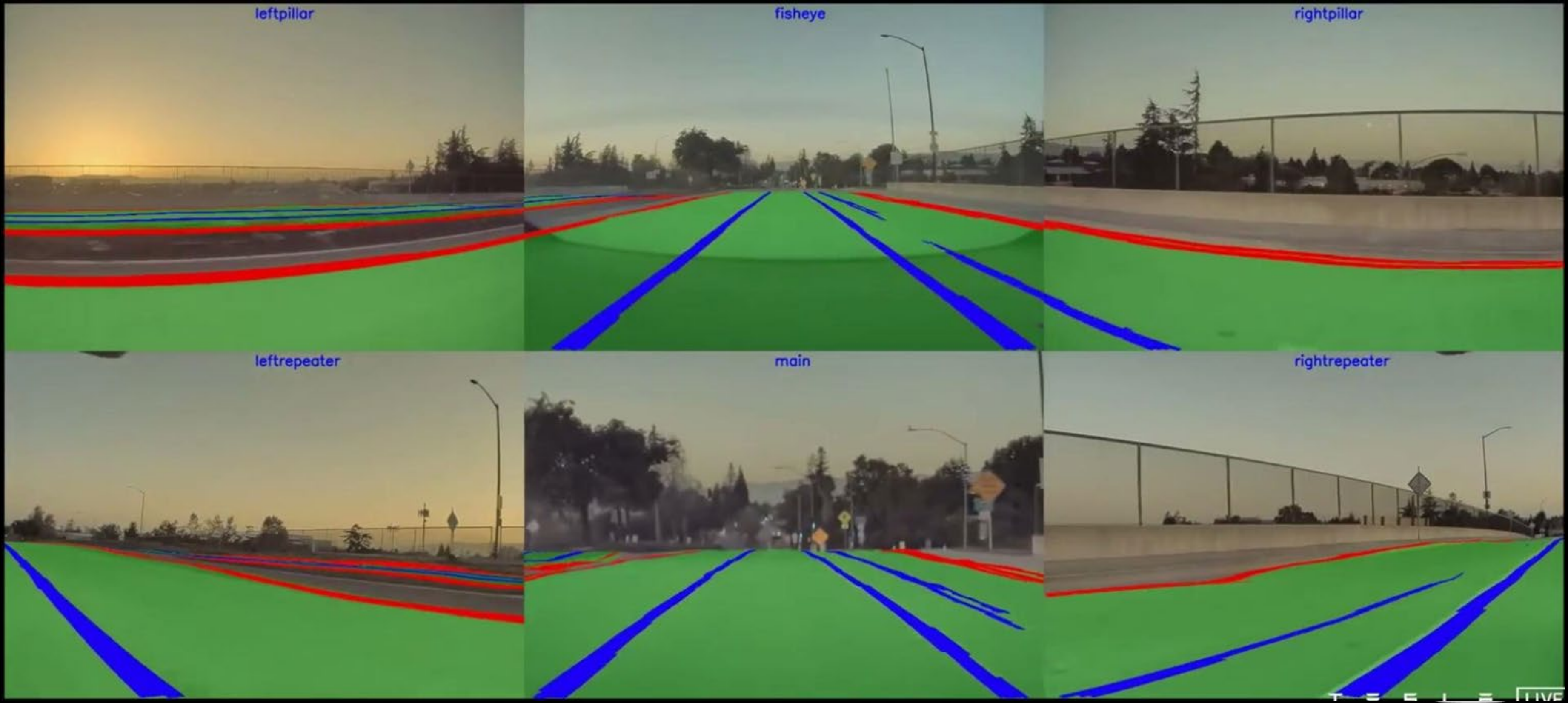
Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *ECCV*, 2020.

Restructuring the Road





Restructuring the Road - Results

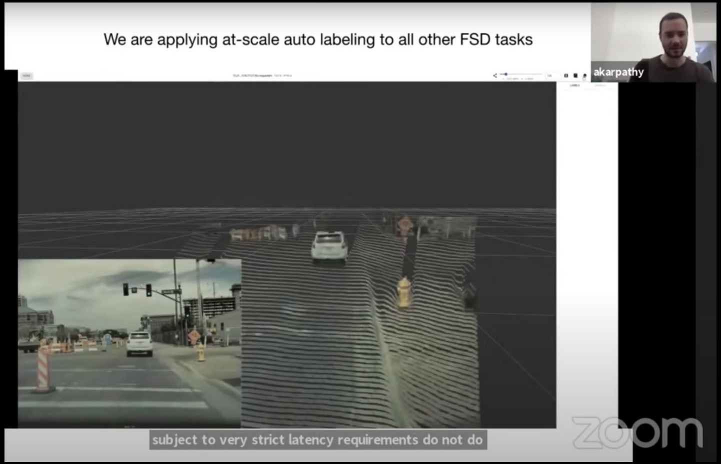


Static Objects

Walls, Barriers & Everything Else

Dynamic Objects

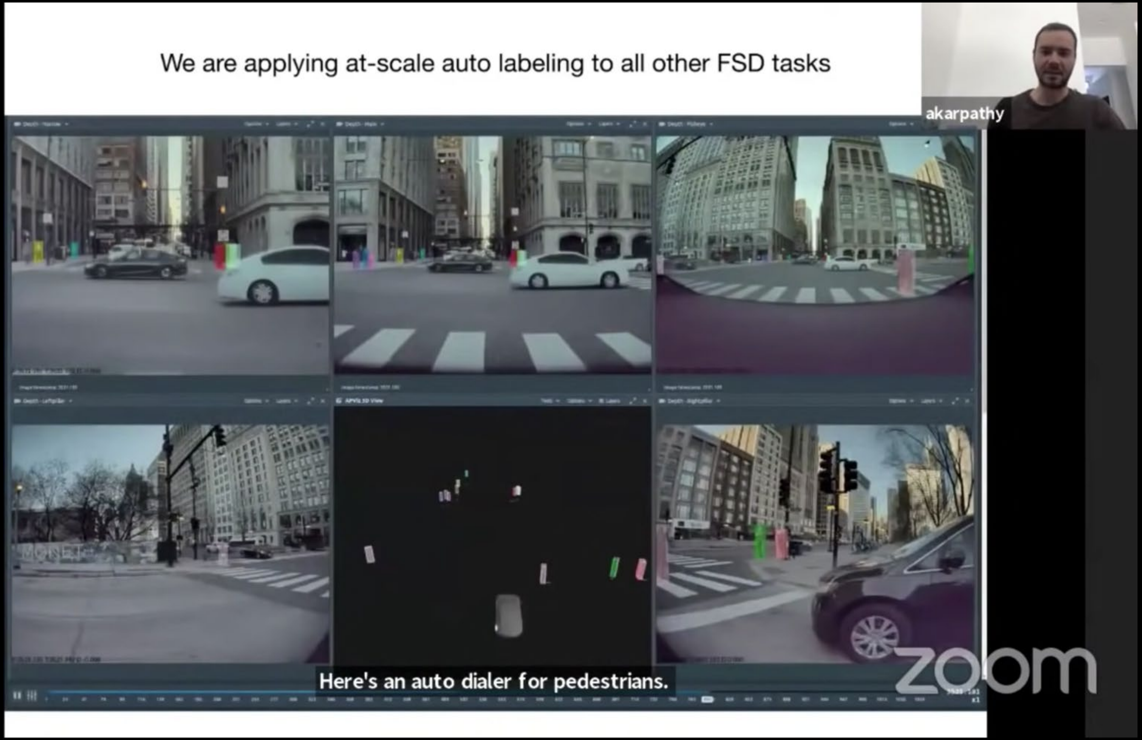
Pseudo Lidar



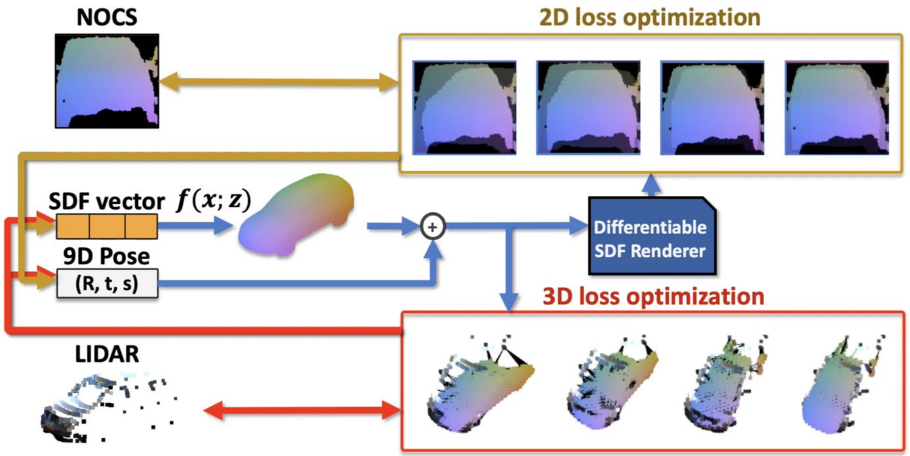
Use extra sensor (Radar in this case)

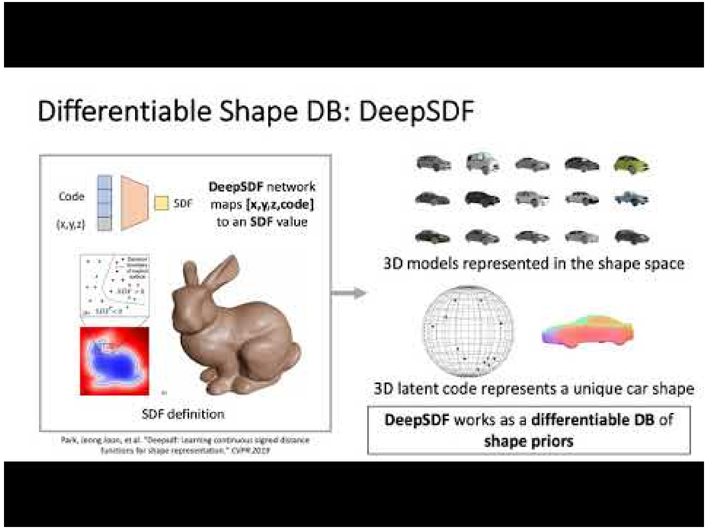
Dynamic Objects

Pedestrian Labeling



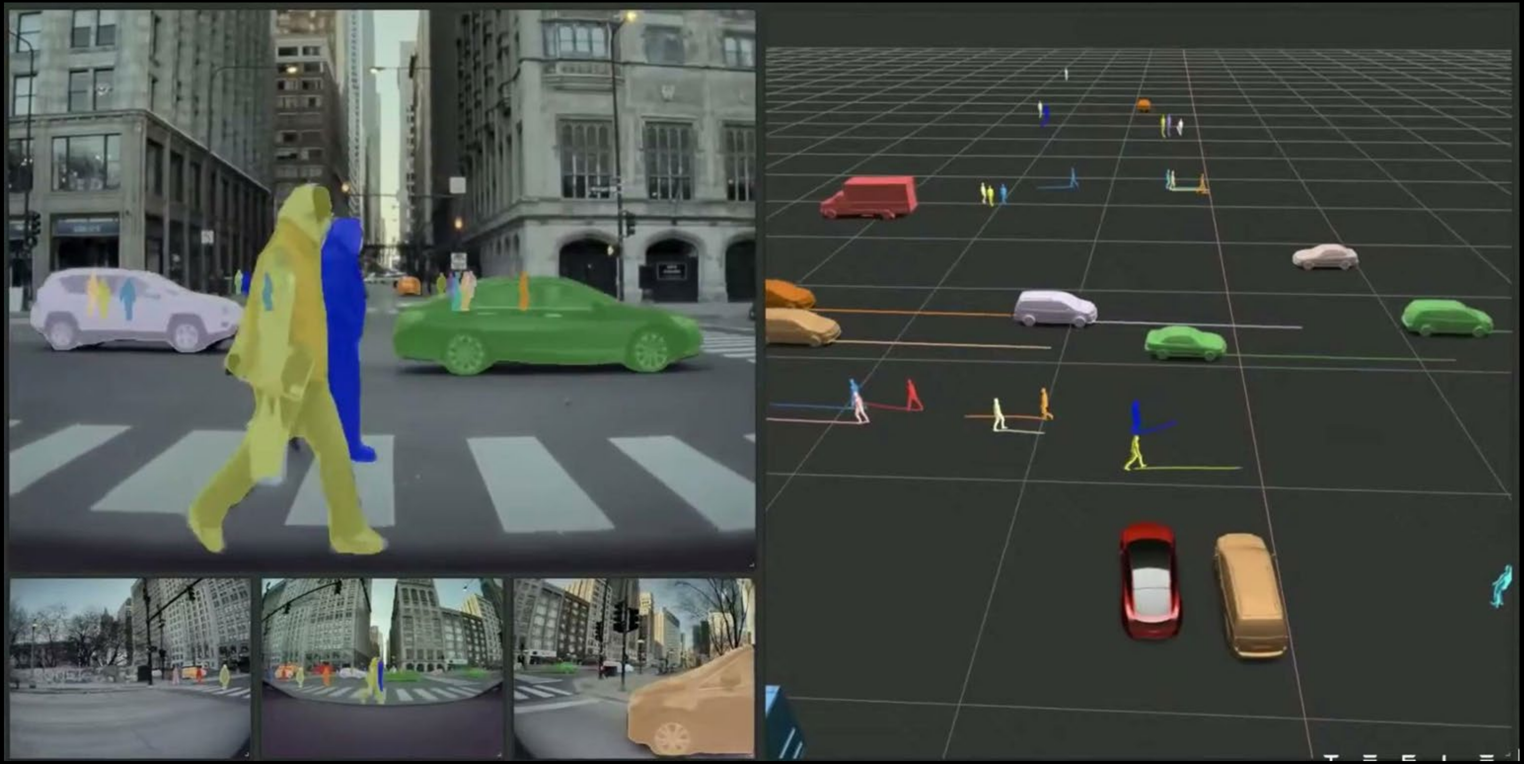
Dynamic Objects



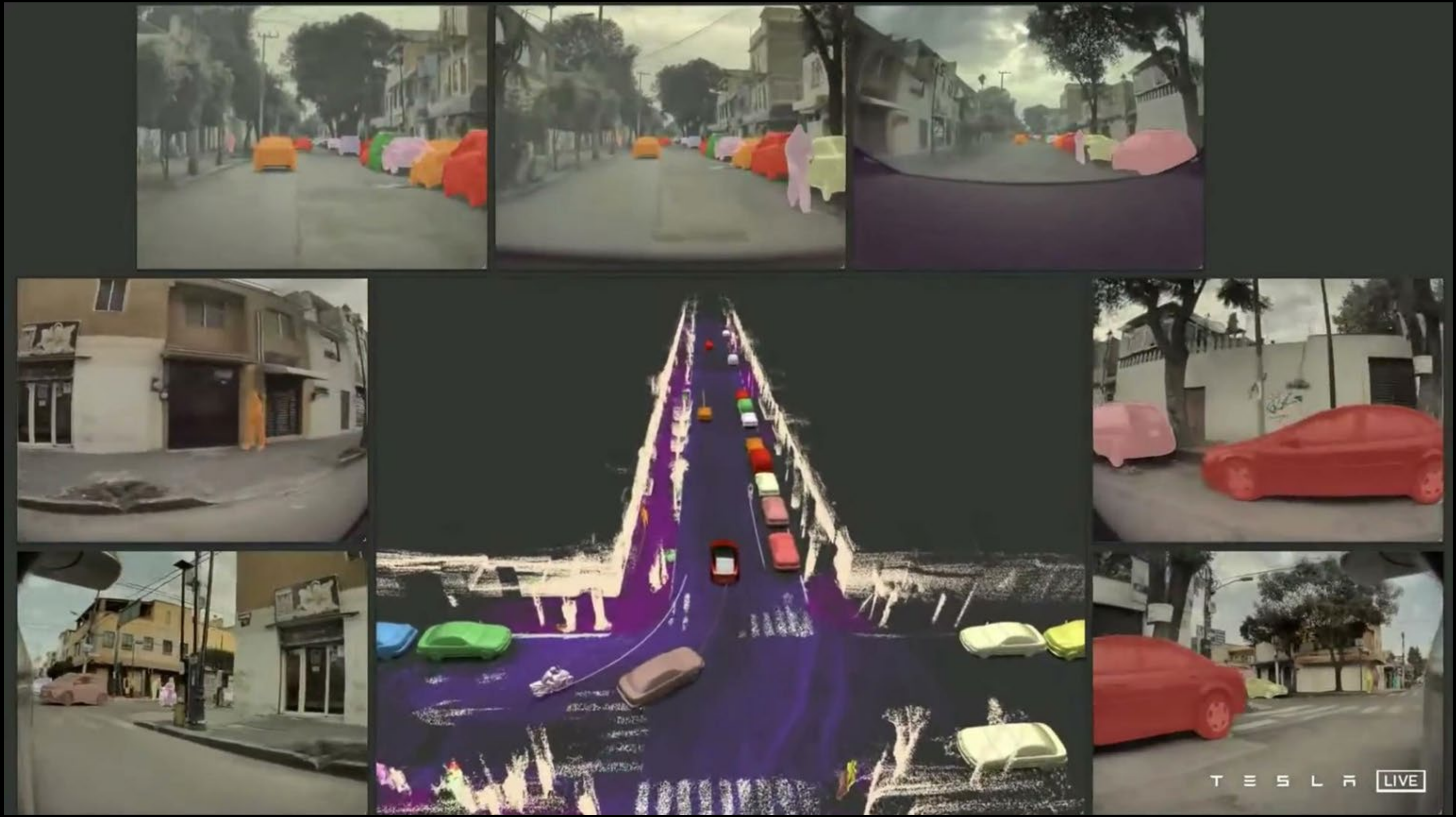


Zakharov, Sergey, et al. "Autolabeling 3d objects with differentiable rendering of sdf shape priors." *CVPR* 2020

Dynamic Objects



Auto Label Dataset



Removing Radar

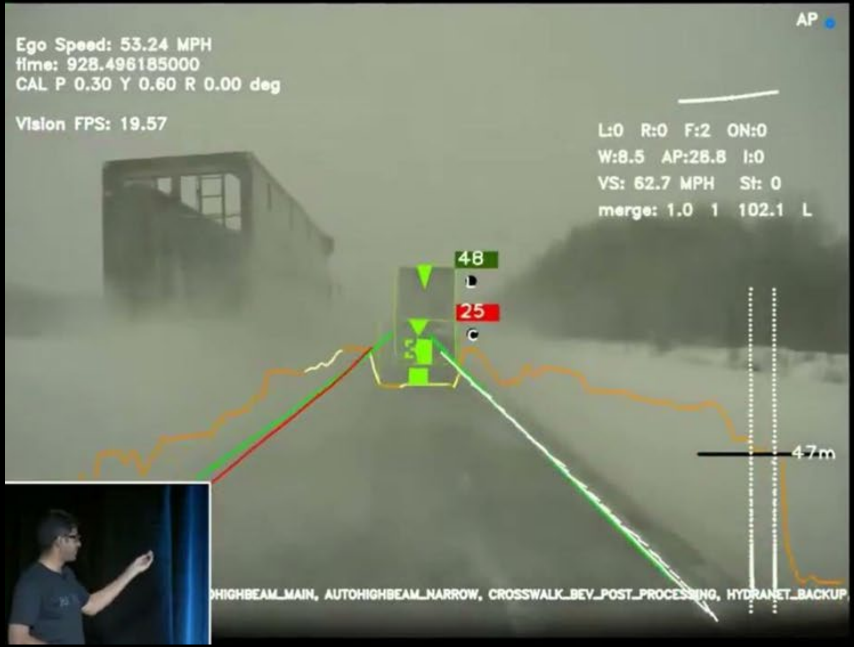
- **Why**
  - Fail in some cases(front car brake harshly).
  - Mismatching(scene below, mismatch static bridge and car)
  - Time Latency

- **How**
  - Data Driven

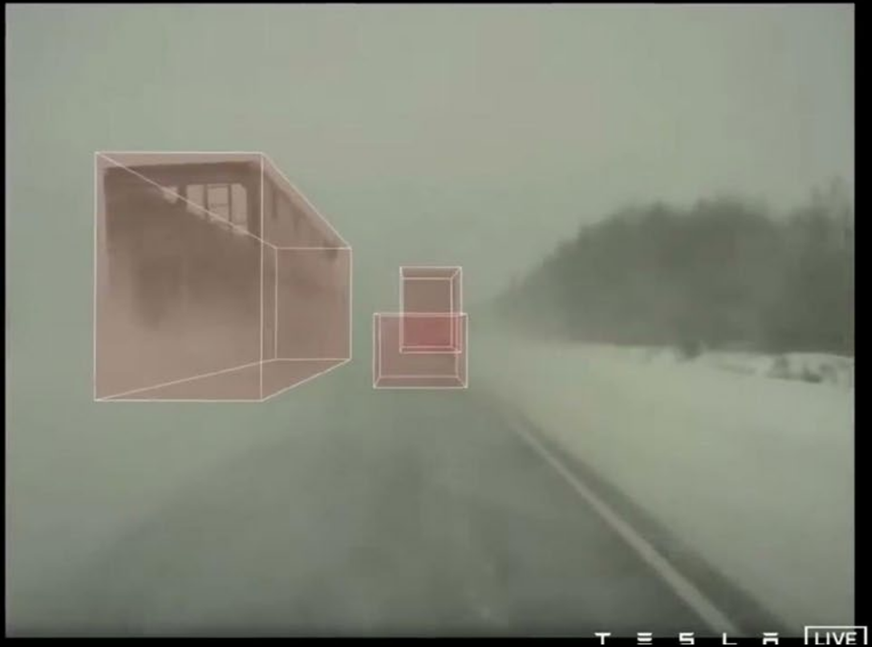
More analysis in CVPR workshop video

Removing Radar -Results

Before



After



Sum-up: data labelling at Tesla

Labeling Techniques and Requirements

- Label in vector space
  - Label infrastructure
- Auto labeling
  - offline neural network servers
  - big model for distill
  - data collection
- Pure vision based labeling
  - 3D reconstruction
  - Visual SLAM

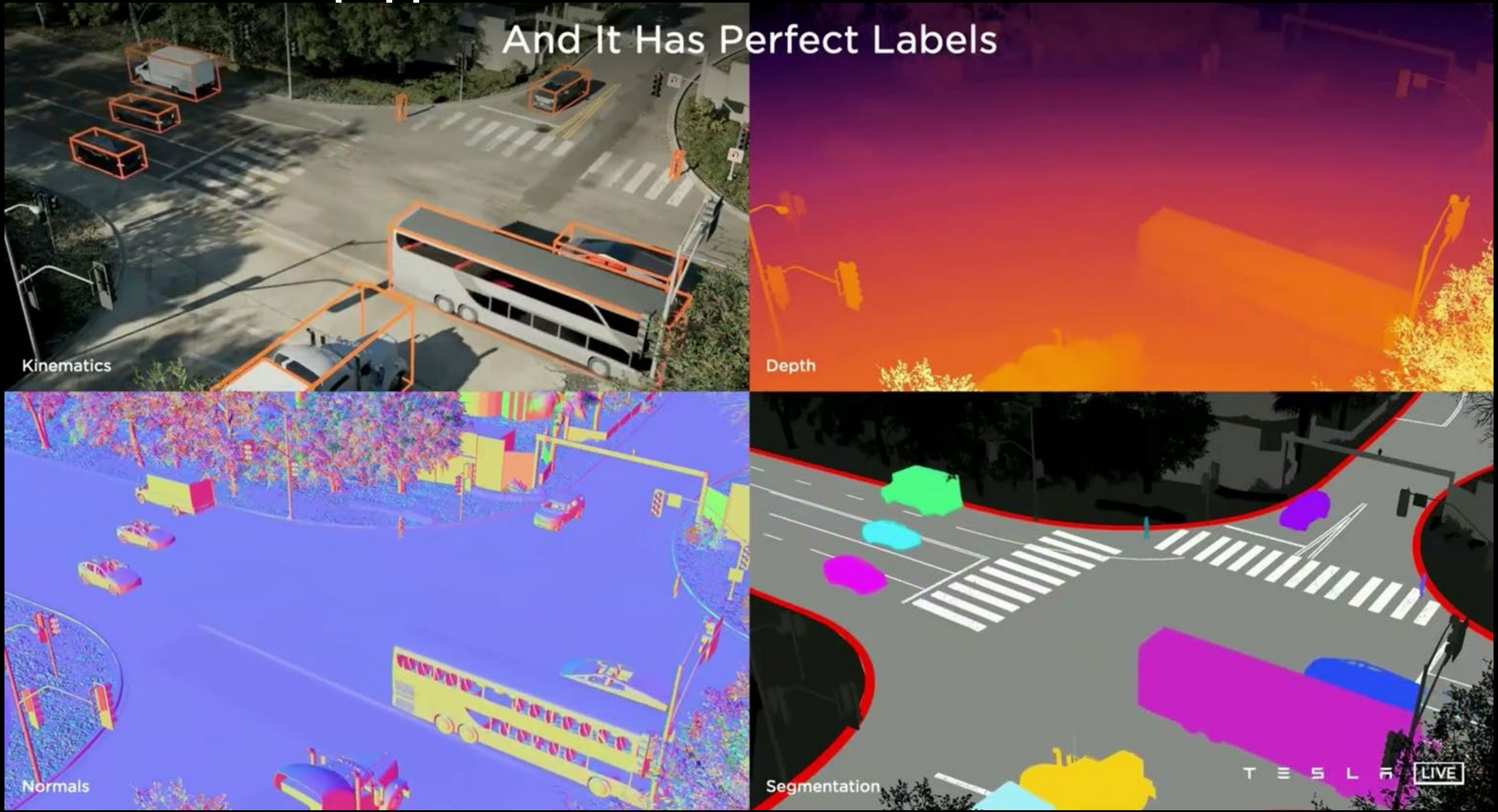
enable →

- label **massive** data in short time.
- label sufficient data for **complex** neural network training.
- collect specific scene data in **short** time, then label it and fix corner cases.

Overview

- Tesla vision
- Planning and Control
- Manual/Auto Labelling
- **Simulation**
- Hardware Integration/Infrastructure
- Dojo

Simulation is equipped at Tesla

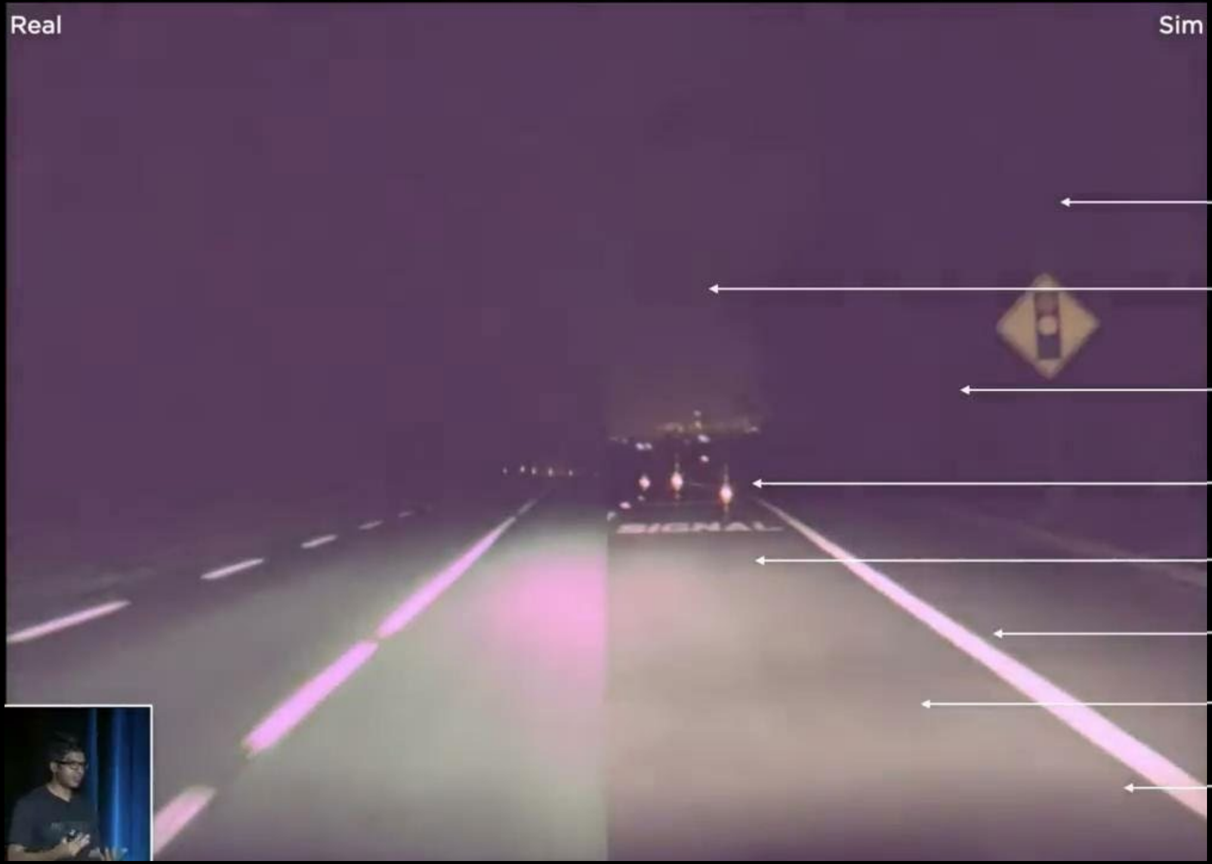


Simulation helps when data

What’s needed to make this happen?

Necessity 1: Accurate sensor simulation

On Semi
AR0136AT
1.2MP
1/3"
Global Shutter



Sensor Noise/Integration Time
传感器噪声/集成时间

Photometrically
Calibrated Sun & Sky
光度法校准的太阳和天空辐射

FSD Computer Exposure in Loop
FSD电脑在环的曝光模拟

Light Diffraction From Heater Grid
挡风玻璃等的光衍射效应

IES Profiles & Physically-Based
Light Transmission
IES配光曲线/基于物理特性的光传输特性

Retro Reflection
逆反射

Motion Blur
运动模糊

Optical Distortion & MTF
光学畸变&MTF调制传递函
数

Necessity 2: Photorealistic rendering（光保真渲染)

Goal of being visually indistinguishable from reality with a hybrid real-time *raytracing* and *neural rendering stack*

Necessity 3: Diverse actors and locations

Thousands of Unique Vehicles,
Pedestrians, & Props

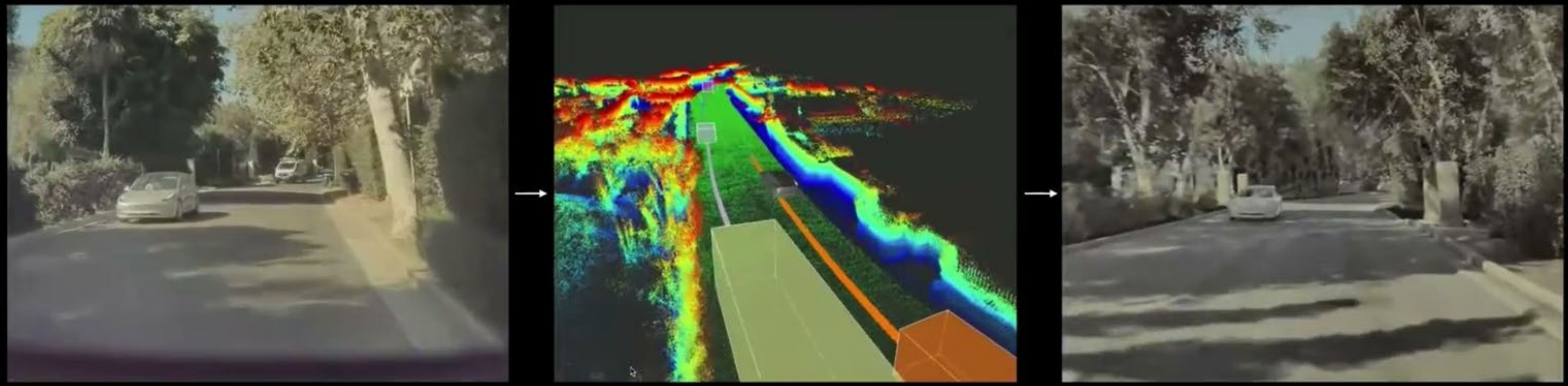
2000+ Miles of Hand-Built Roads
Using In-House Pipeline

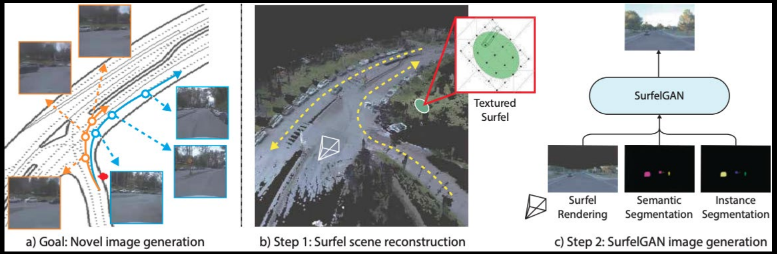
Necessity 4: Scalable scenario generation

Tip of the iceberg

Necessity 5: Scenario reconstruction

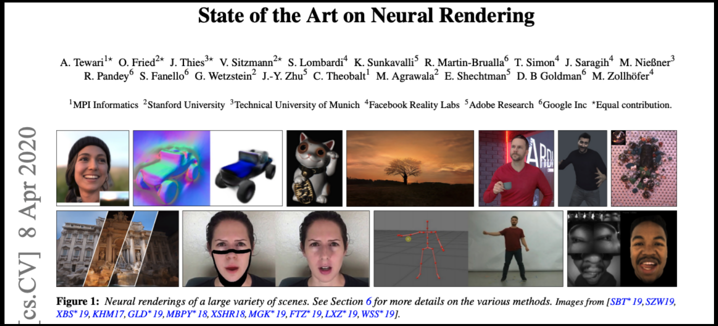
Real World Clip | Auto-Labeled Reconstruction | Recreated Synthetic World





Yang, Zhenpei, et al. "SurfelGAN: Synthesizing realistic sensor data for autonomous driving." CVPR 2020.

Neural rendering


State of the Art on Neural Rendering

A. Tewari[1*] O. Fried[2*] J. Thies[3*] V. Sitzmann[2*] S. Lombardi[4] K. Sunkavalli[5] R. Martin-Brualla[6] T. Simon[4] J. Saragih[4] M. Nießner[3] R. Pandey[6] S. Fanello[6] G. Wetzstein[2] J.-Y. Zhu[5] C. Theobalt[1] M. Agrawala[2] E. Shechtman[5] D. B Goldman[6] M. Zollhöfer[4]

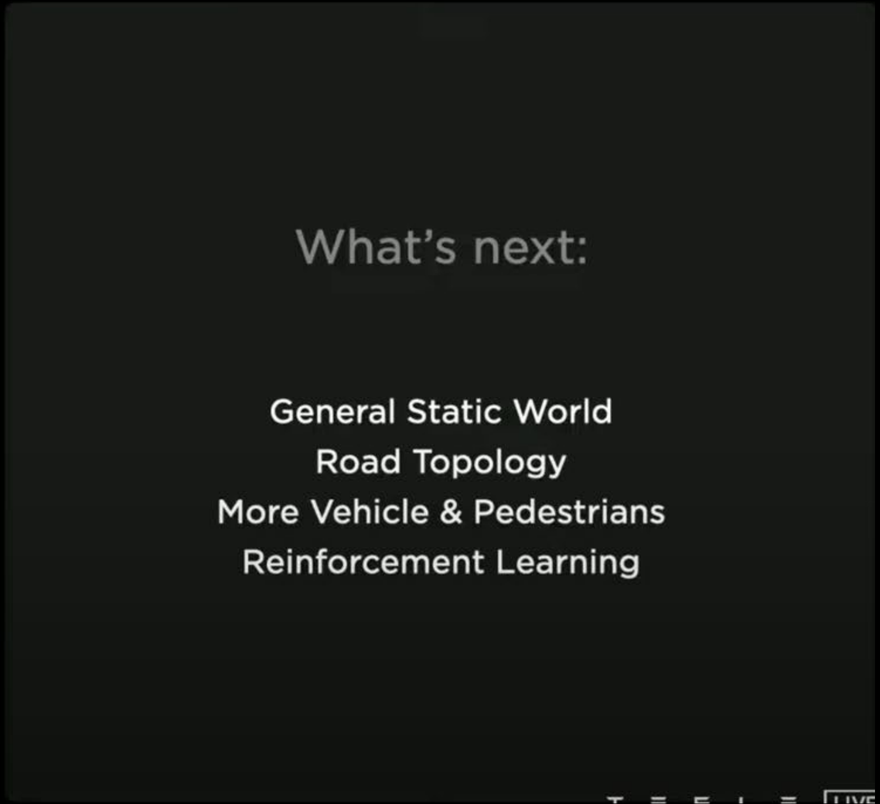
[1]MPI Informatics [2]Stanford University [3]Technical University of Munich [4]Facebook Reality Labs [5]Adobe Research [6]Google Inc *Equal contribution.

[cs.CV] 8 Apr 2020

Figure 1: *Neural renderings of a large variety of scenes. See Section 6 for more details on the various methods. Images from [SBT* 19, SZW19, XBS* 19, KHM17, GLD* 19, MBPY* 18, XSHR18, MGK* 19, FTZ* 19, LXZ* 19, WSS* 19].*

Tewari, Ayush, et al. "State of the art on neural rendering." Computer Graphics Forum. Vol. 39. No. 2. 2020.

Simulation works! and has improved so far:



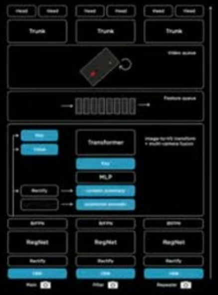
Overview

- Tesla vision
- Planning and Control
- Manual/Auto Labelling
- Simulation
- **Hardware Integration/Infrastructure**
- Dojo

- 10 billion **(100亿)** labels on 250w video clips
- 2000 CPU cores
- multiple GPUs

AI Compiler & Scheduling

Hardware integration



Neural Networks

Algorithms

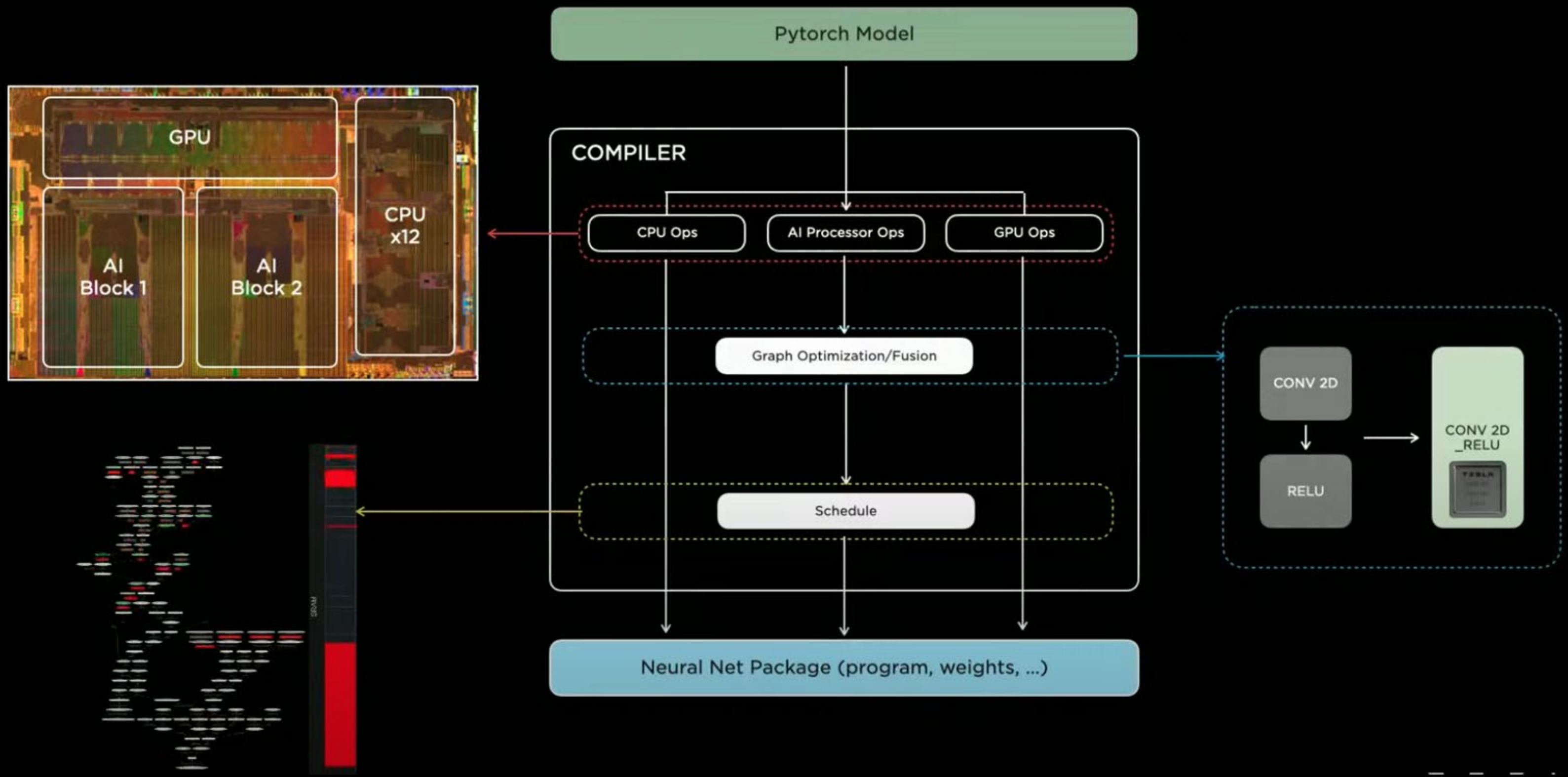
FSD Chip: 72 TOPS

FSD Computer

Minimize Latency
&
Maximize Frame Rate

AI Compiler & Scheduling

Neural Net Compiler

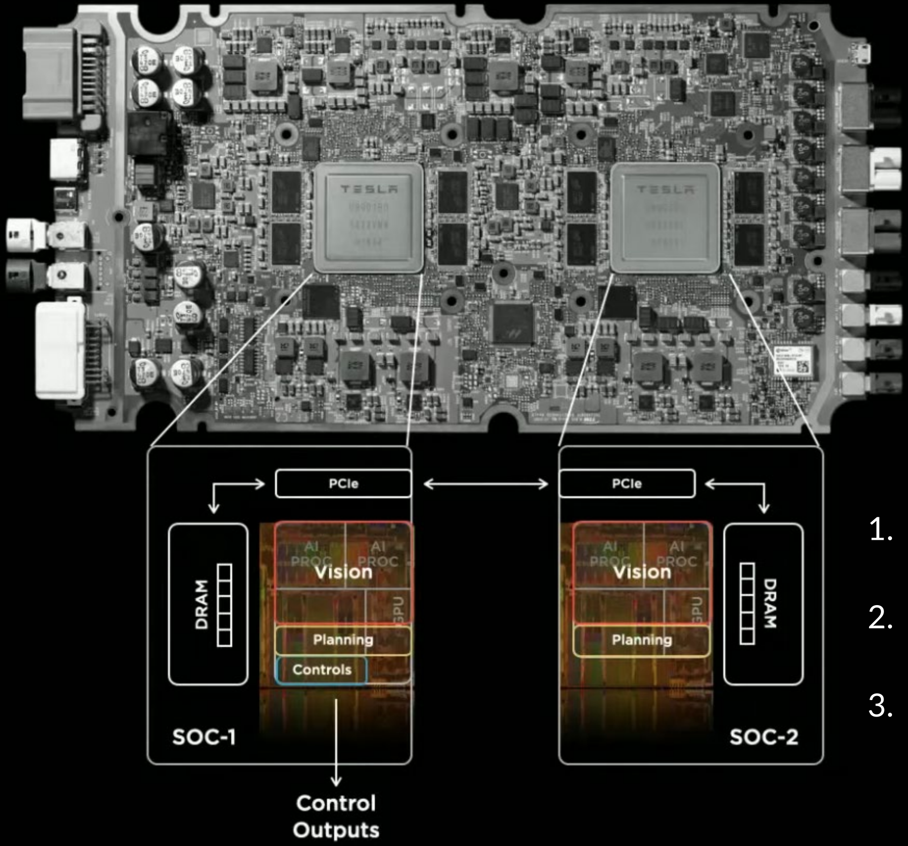


AI Compiler & Scheduling

Dual SoC

Dual FSD Chip:

144 TOPS



1. Only one of the two engines outputs the control comments
2. The other one serve as an extension of compute
3. Roles are interchangeable

AI Evaluation Infrastructure

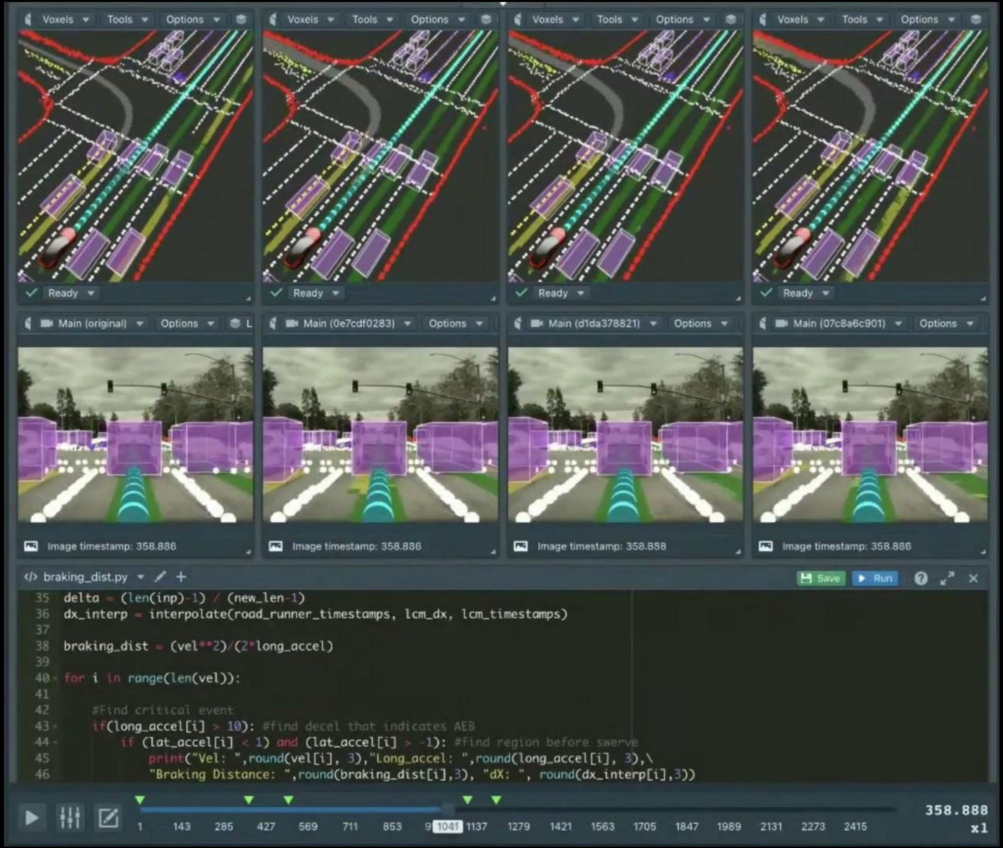
Road to develop the AI Cycle

1M+ runs/week

- 3 datacenters + cloud
- 3,000+ Autopilot FSD computers
- Bit-perfect evals on real FSD AI Chip hardware
- Custom job scheduling & device management software

Debugging Tools

Helping the development and iteration of Neural Network

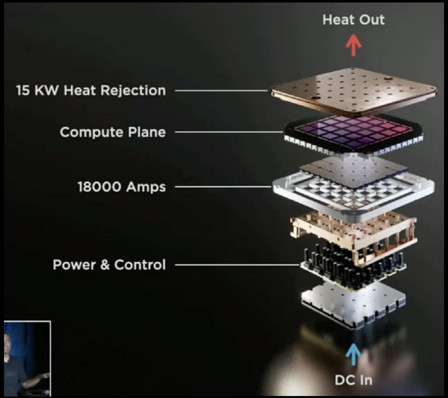


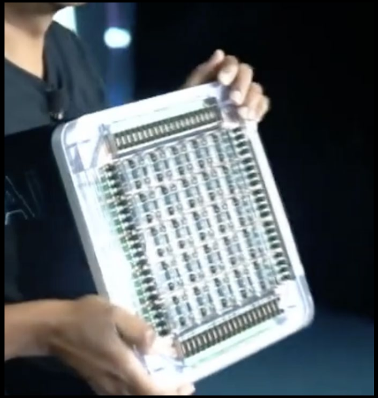
- A visualization platform
- Comparing different NN
- Increase efficiency

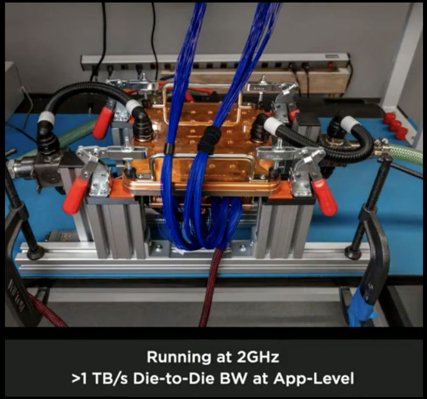
Overview

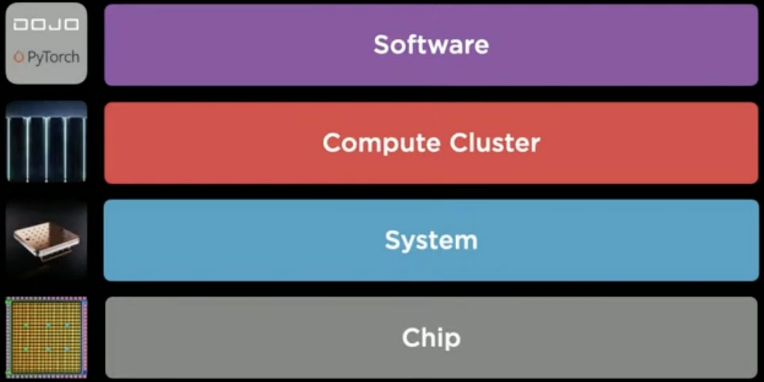
- Tesla vision
- Planning and Control
- Manual/Auto Labelling
- Simulation
- Hardware Integration/Infrastructure
- **Dojo**
  Dojo芯片深度解析
  https://mp.weixin.qq.com/s/4OMM6rTFmSuJhvwTOkcNRg

Dojo at a Glance

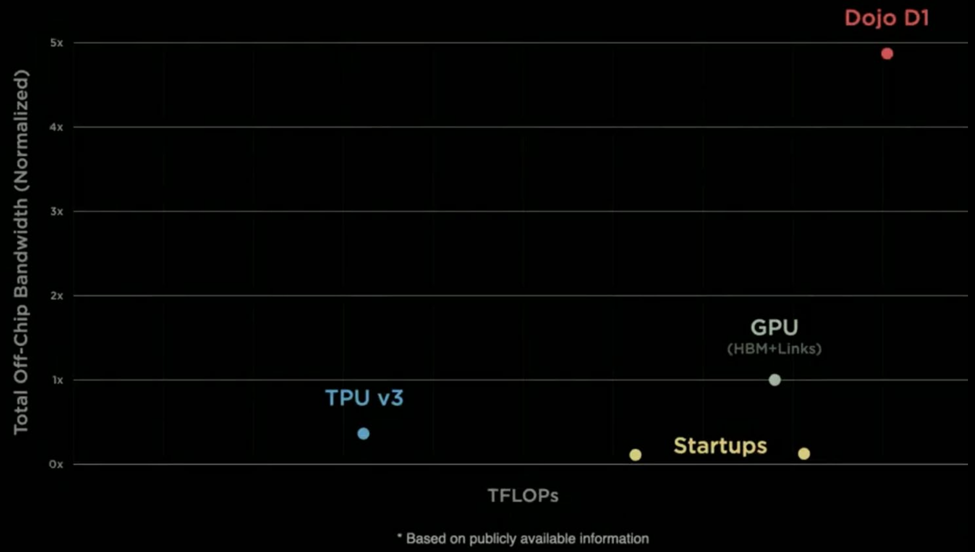








Compute Scaling for Training



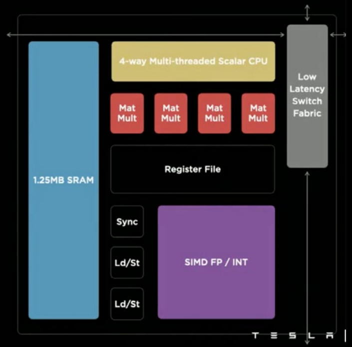
Training Node

512 GB/s in each cardinal direction, each node

354 Training Nodes

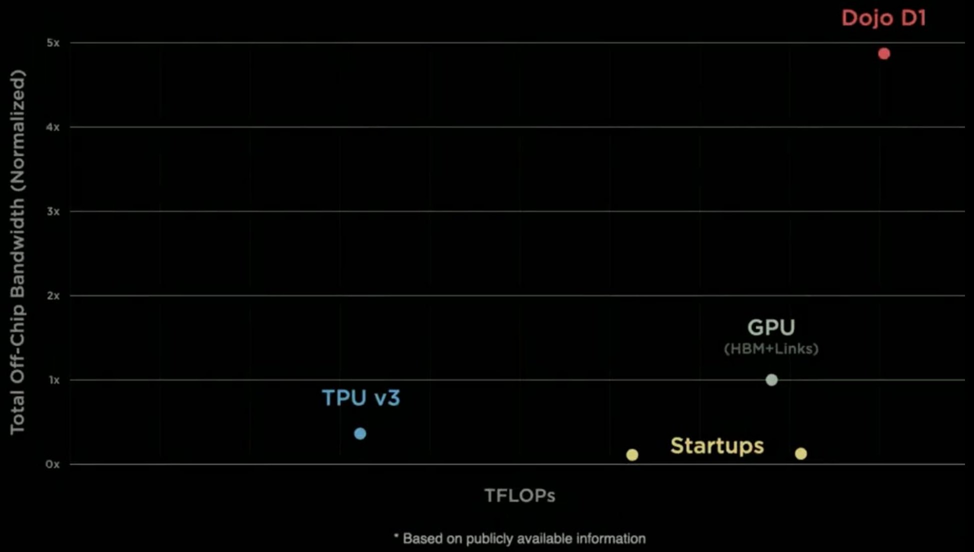
362 TFLOPS BF16/CFP8

22.6 TFLOPS FP32



Compute Scaling for Training

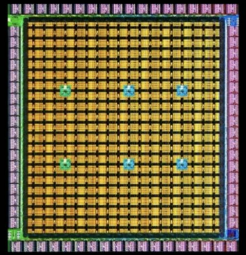
D1 chip



362 TFLOPs BF16/CFP8
22.6 TFLOPs FP32

10TBps/dir. On-Chip Bandwidth
4TBps/edge. Off-Chip Bandwidth

400W TDP

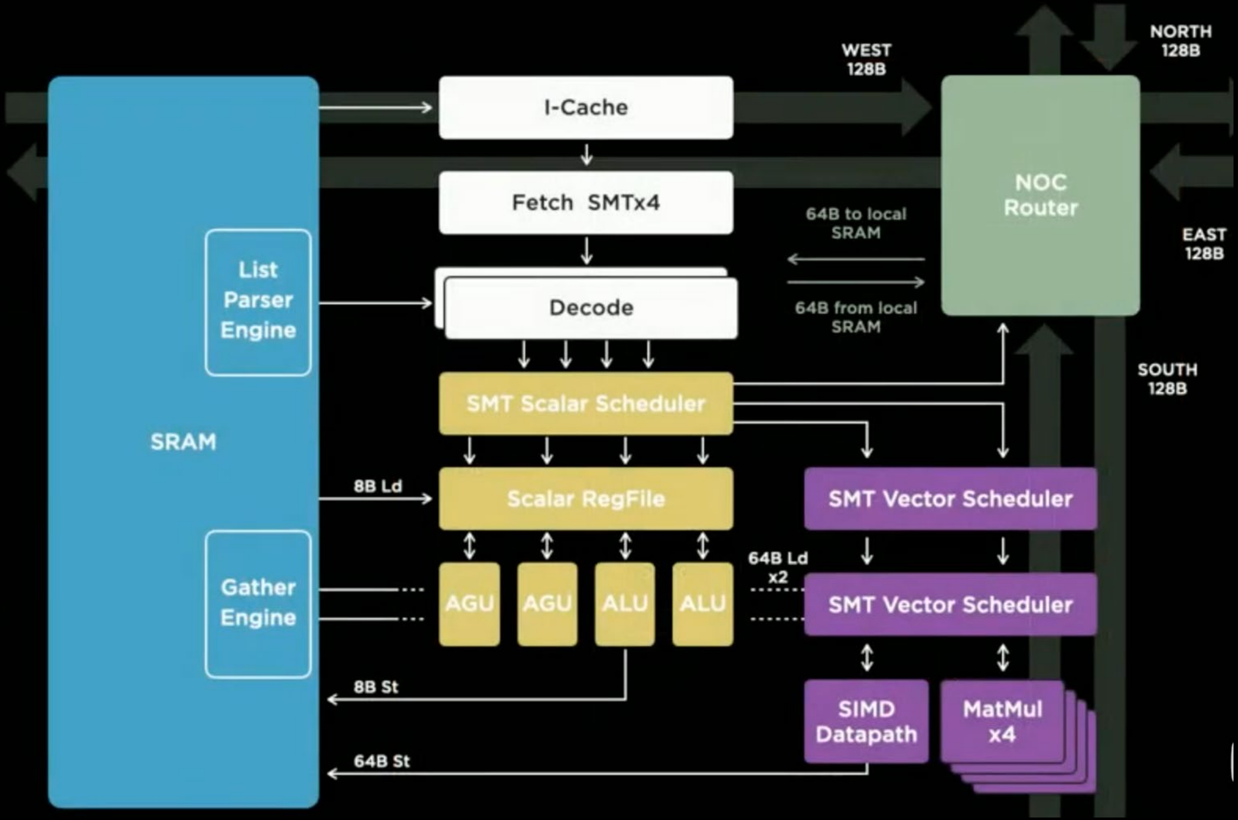


645mm²
7nm Technolo

50 Billion
Transistors

11+ Miles
Of Wires

Distributed Compute Architecture



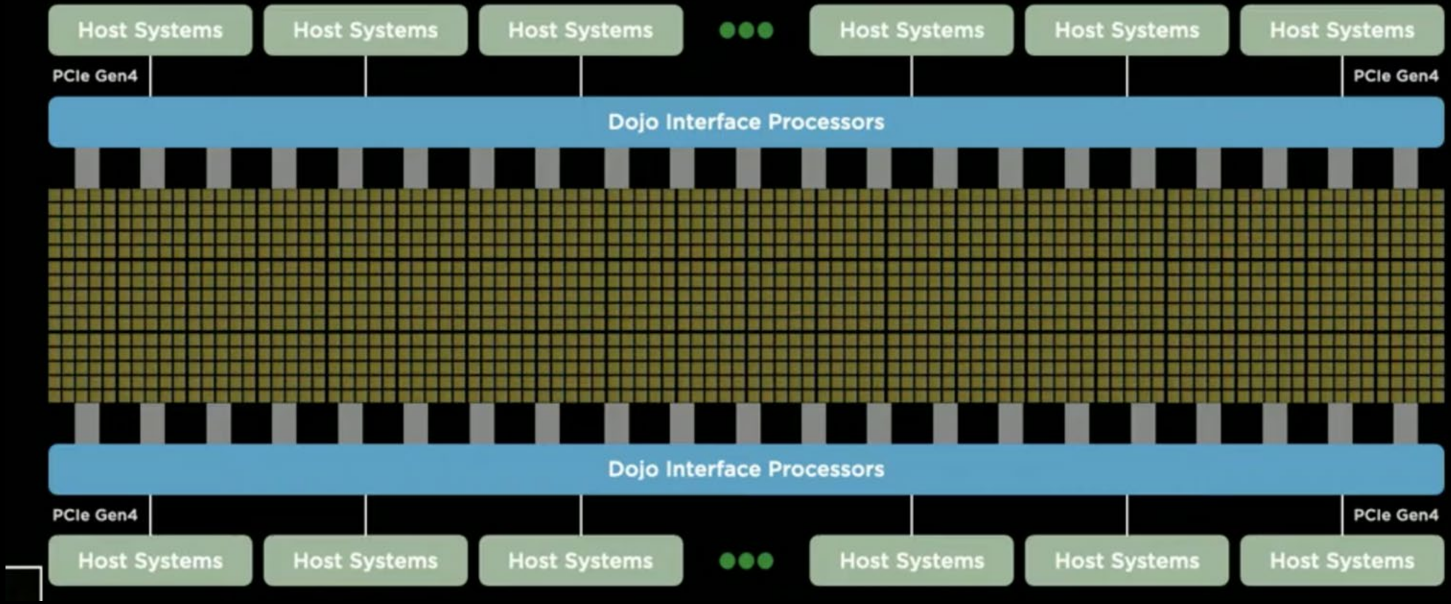
Superscalar in-order CPU

4 wide scalar + 2 wide Vector Pipes

4-way Multithreaded

Custom ISA Optimized for ML Kernels

Dojo Architecture



Dojo Architecture

Logical view of the System

DPU - Dojo Processing Unit

Virtual device that can be sized
based on Application Needs

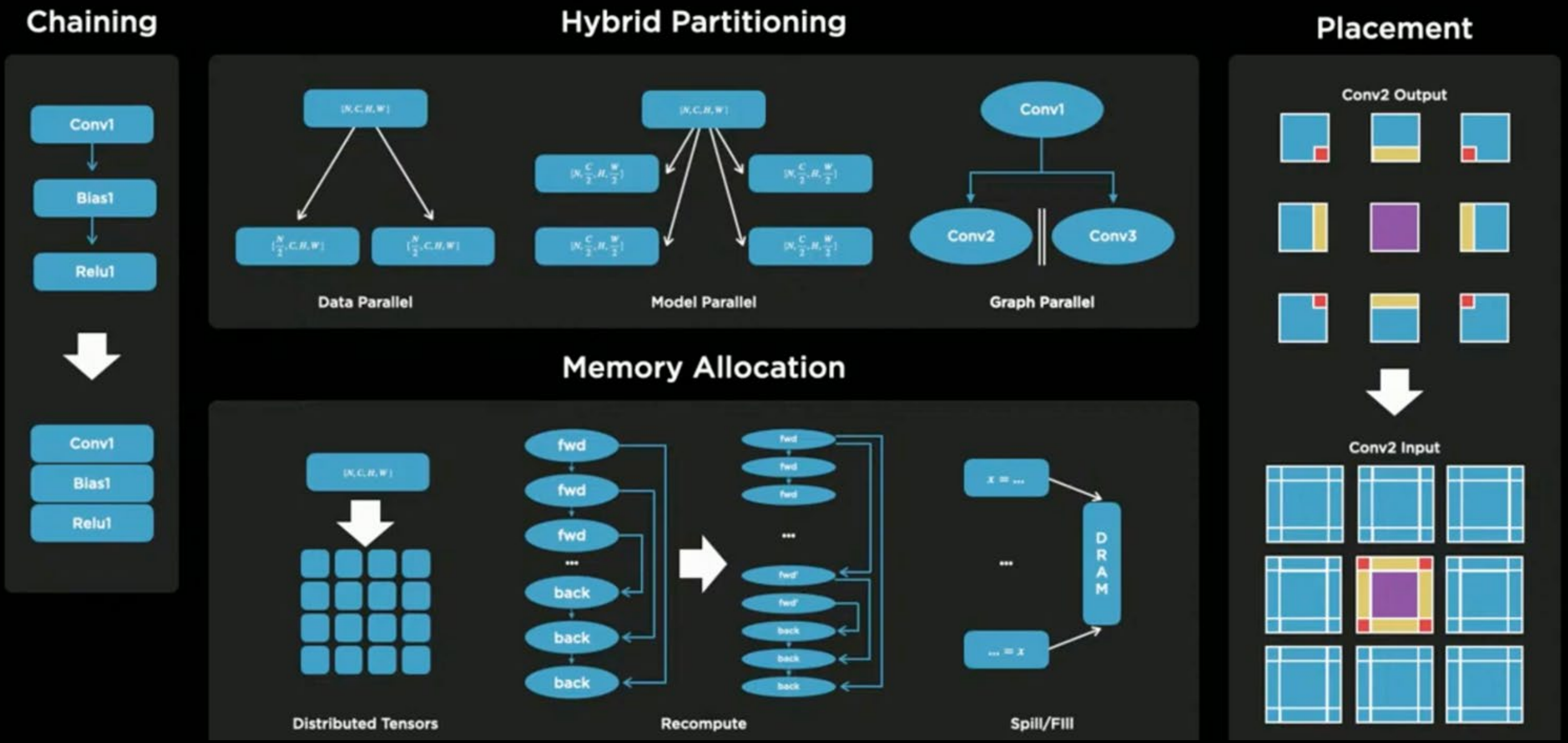
D1 Accelerator Chips (Compute + Local Memory)
Dojo Interface Processors (Ingest + Shared Memory)

User view of the System

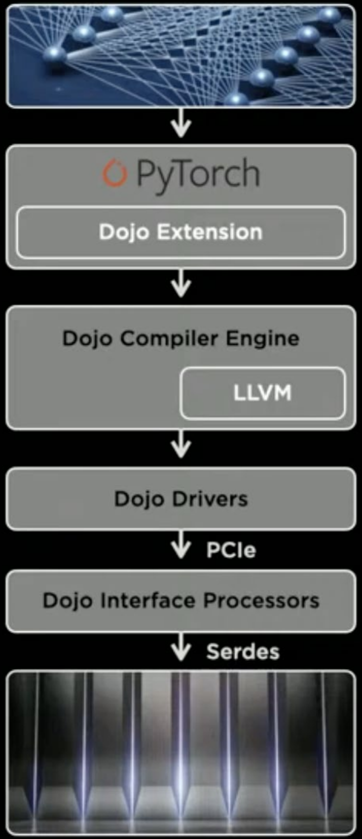
```
device = torch.device(αcuda:0δ)
                 ↓
device = torch.device("dojo")
```

Compiler performs mapping onto DPU (virtual device)
automatically without user involvement

Dojo Architecture Compiler Engine



Dojo Architecture Software Stack



Neural Net Models

PyTorch Extension

JIT NN Compiler

LLVM Backend

Multi-host, Multi Partition Management

Ingest & Shared Mem

ExaPOD

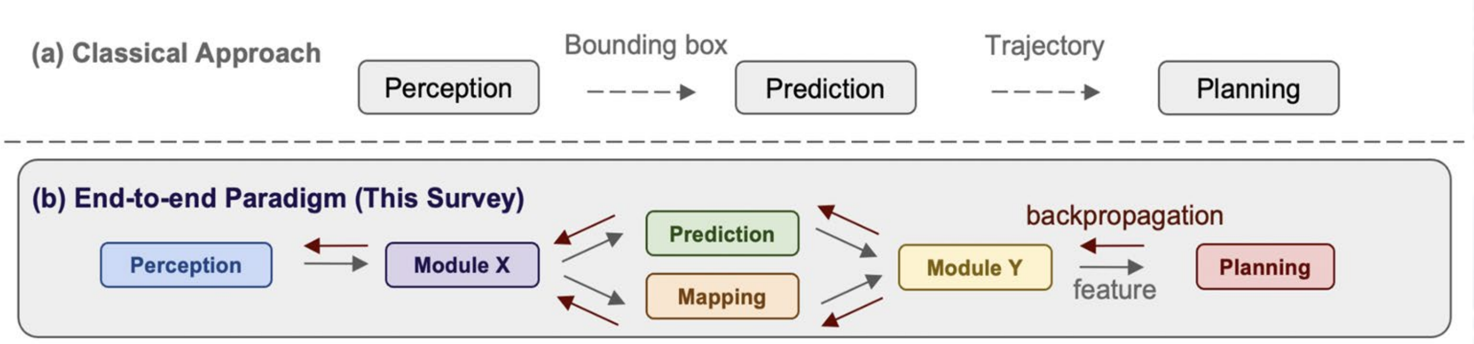
OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

End-to-end Autonomous Driving

An Introduction

回顾：Why end to end?



https://github.com/OpenDriveLab/End-to-end-Autonomous-Driving

端到端自动驾驶系统：

- 将原始传感器数据作为输入
- 输出轨迹规划，或低级别的控制信号

回顾：Why end to end?

优势

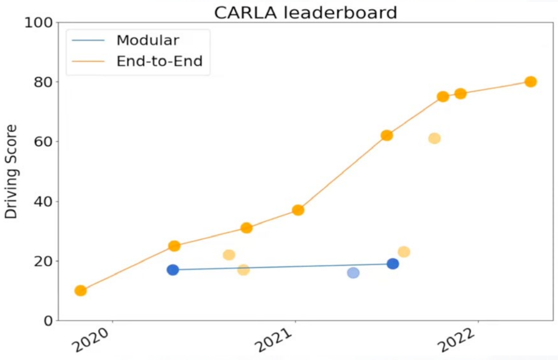
+ 将所有模块合并为一个可**联合训练**的单一模型带来的**便利性**
+ 避免模块化设计带来的级联错误
+ 直接**针对最终任务进行优化（规划/轨迹预测）**
+ 计算效率高 (共享 backbone), 对最终产品友好

回顾：Why end to end?

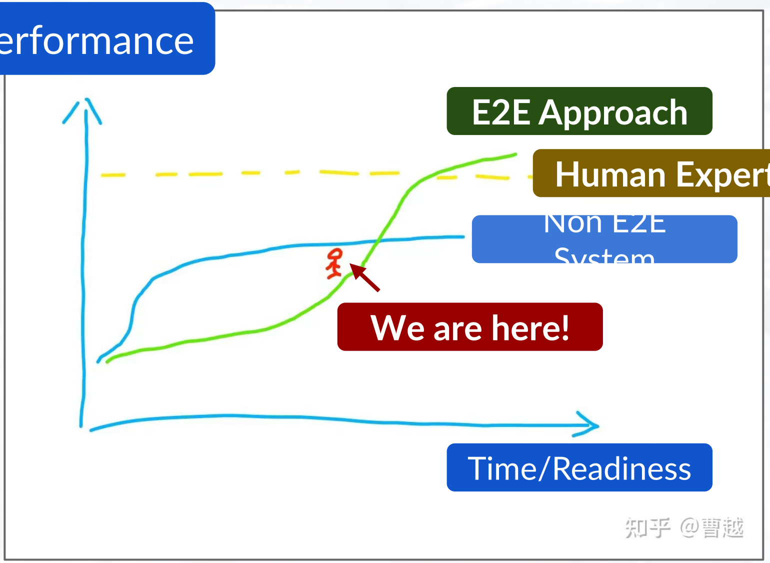
劣势

- 只能在模拟器和机载测试中进行闭环评测(Closed-loop evaluation)
- 缺少真实世界数据
- 可解释性差

Credit to Andreas Geiger @ CVPR Workshop 2023

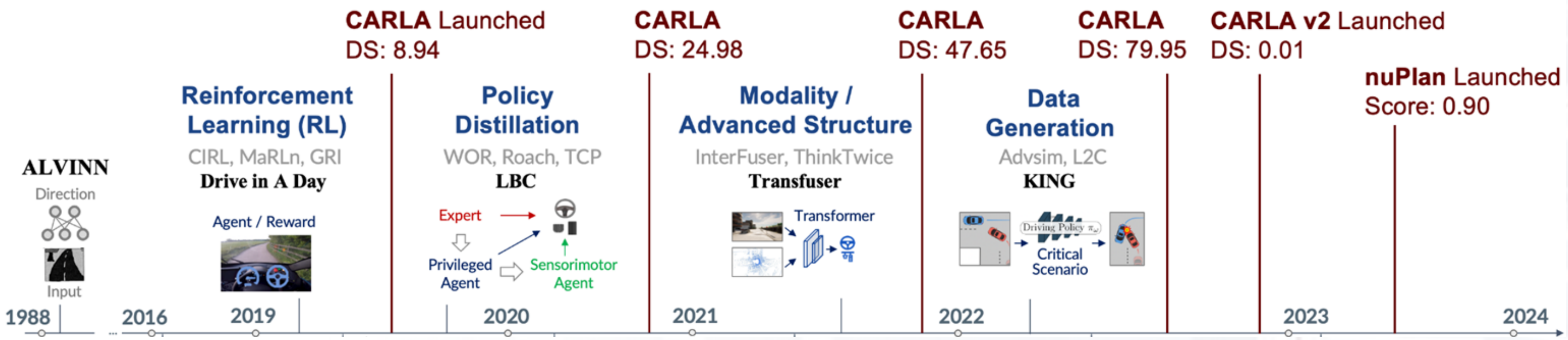


E2E vs Non-E2E



Credit to Dr. Yue Cao @ Zhihu

Roadmap | End-to-end Autonomous Driving



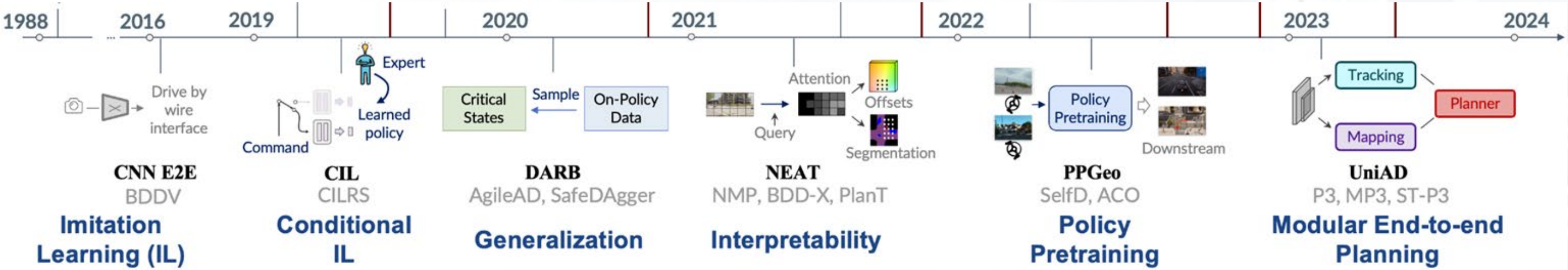
Summary (1/2)

- Carla leaderboard gets much improved over the years. With new mapping / routes (Carla v2) and nuPlan benchmark, this field got so much to do.
- RL method is prevalent in the beginning (since it's natural)
- Input modality and more advanced structure boosts the performance

Roadmap | End-to-end Autonomous Driving

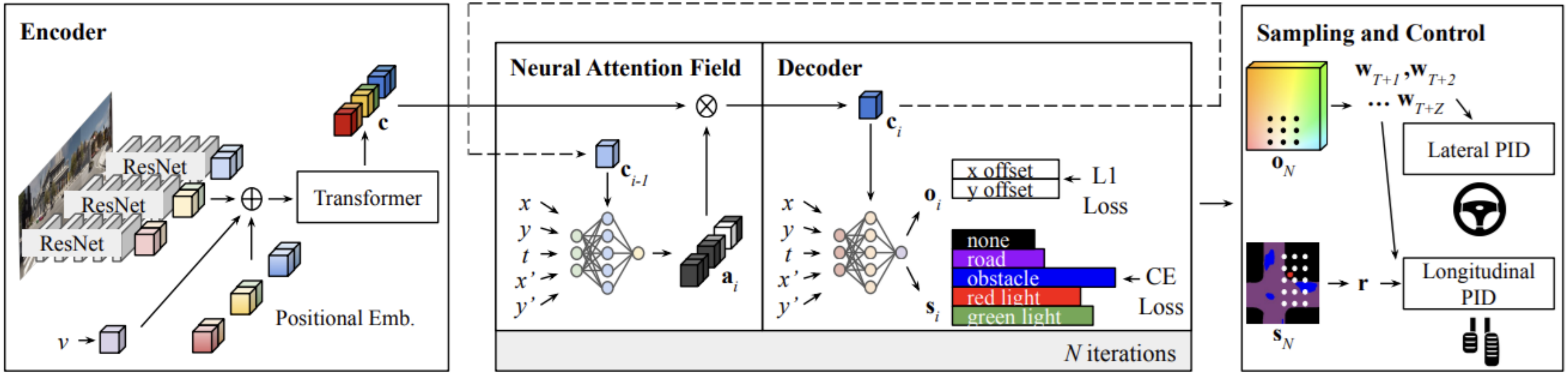
Summary (2/2)

- The First Neural Net based method dates back to 2016 using Imitation Learning
- Learned policy from Experts (IL), with data augmentation, could prevail in performance
- Interpretability, with explicit design in the network stands out recently
- End-to-end design comes to obsess many merits in previous attempt



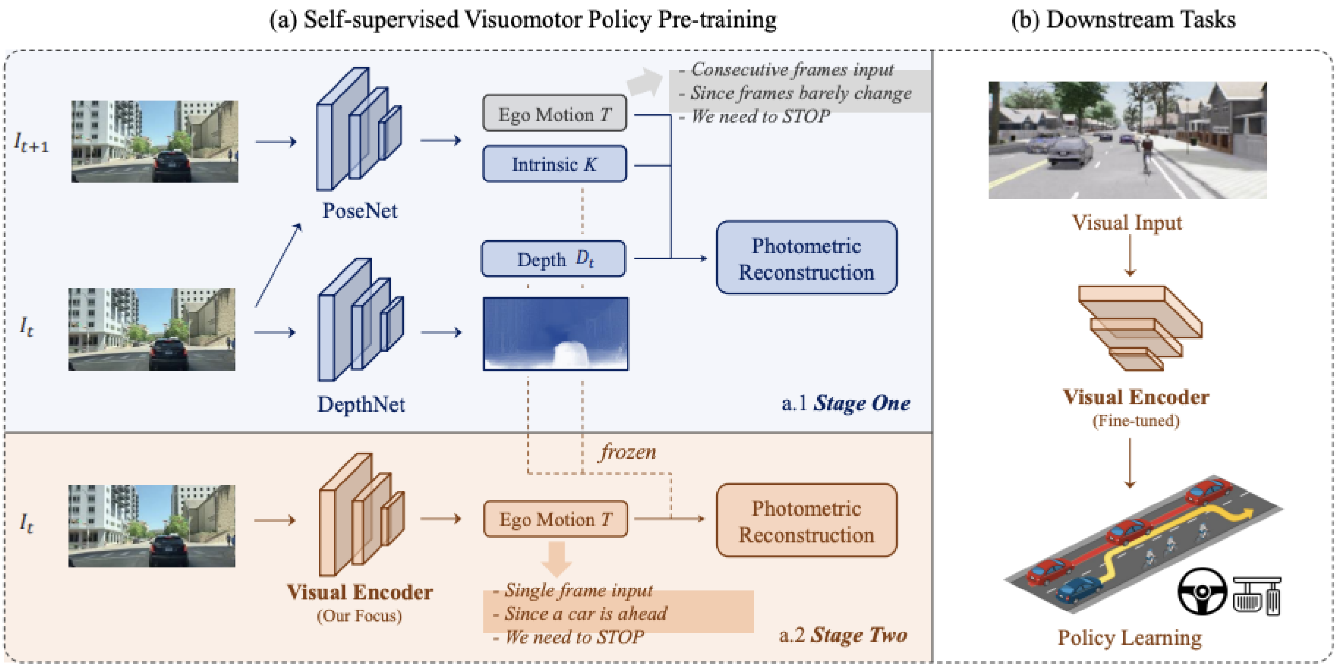
主流方法：NEAT

https://arxiv.org/pdf/2109.04456.pdf



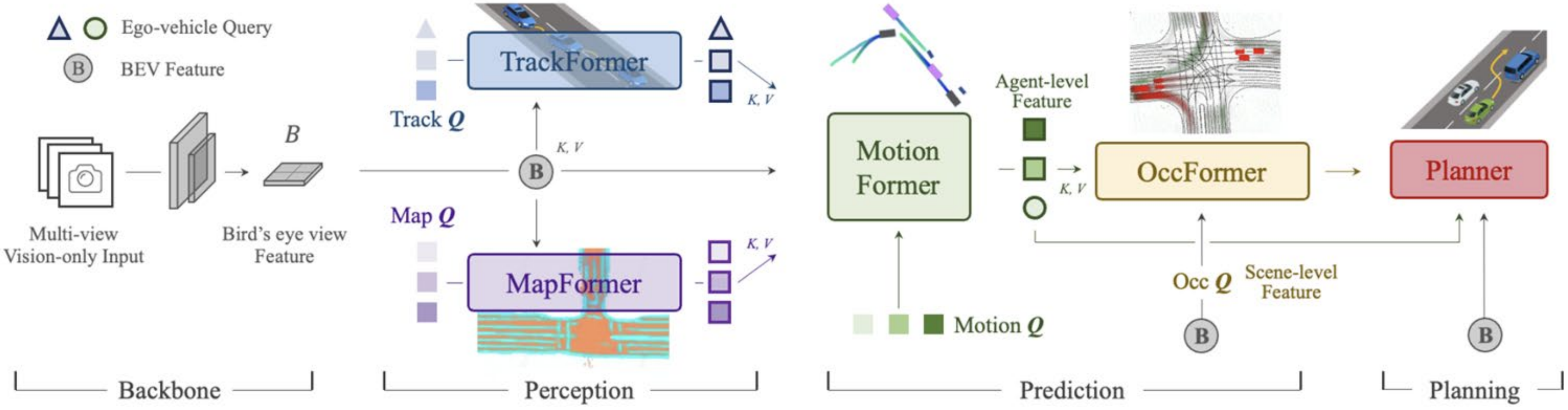
主流方法：PPGeo

https://arxiv.org/pdf/2301.01006.pdf



主流方法：UniAD

https://arxiv.org/pdf/2212.10156.pdf



OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

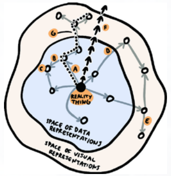
End-to-end Autonomous Driving

Key Challenges

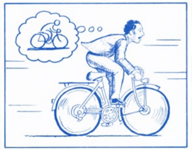
Challenges in End-to-end Autonomous Driving

An Overview

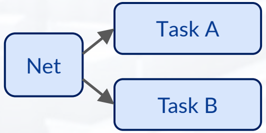
Input Modality



Visual Abstraction

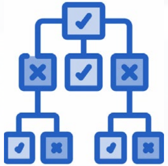


World Model



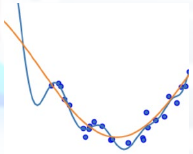
Multi-task Learning

Policy Distillation



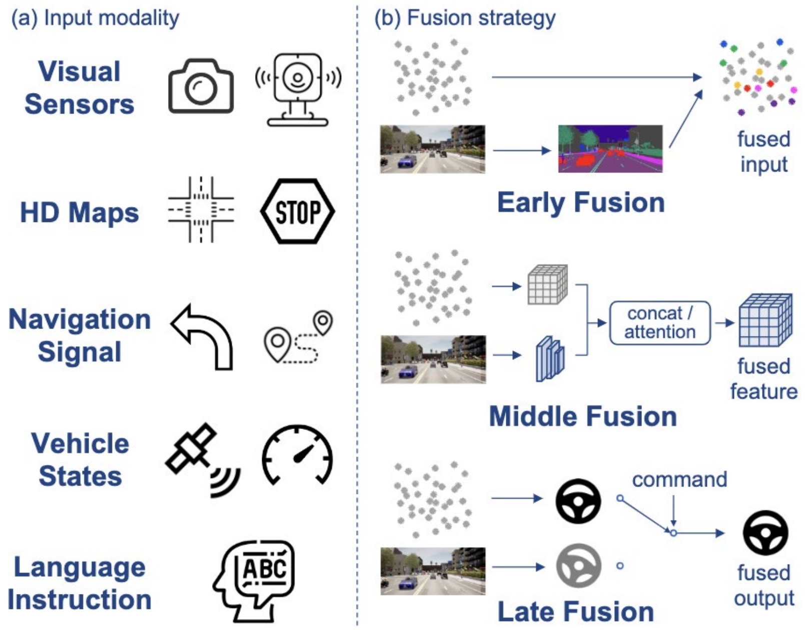
Interpretability

Causal Confusion



Robustness and Generalization

挑战（1/8） - Input Modality



- **Early Fusion:** Combine sensory information before feeding it into the feature extractor
- **Middle Fusion:** Separately encode inputs and then combining them at the feature level
- **Late Fusion:** Combine multiple results from multi-modalities **(Worst Performance)**

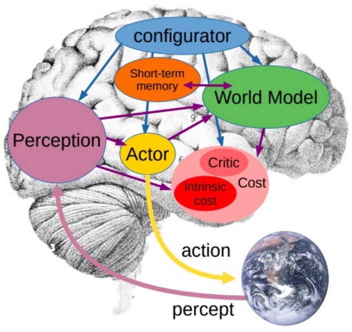
挑战（2/8）- Visual Abstraction

Current methods first pre-train the visual encoder of the network using **proxy pre-training tasks.**

⇩

There inevitably exist possible **information bottlenecks** in the learned representation, and redundant information unrelated to driving decisions may be included.

挑战（3/8）- World Model



	States	Cost / Reward
RL Gyms	- Ego agent - Other objects (**static**) - Background environment	- Success/Fail - Intermediate Reward
Autonomous Driving	- Ego-vehicle - Other vehicles, pedestrians, cyclists, etc (**moving**) - Background environment	- Collision - Comfort - Forward - etc

Complicated!

Hard to define!

A video predictor?

挑战（4/8） - Multi-task Learning

Multi-task learning (MTL)： Jointly perform several related tasks based on a shared representation through separate branches/heads.

Pros

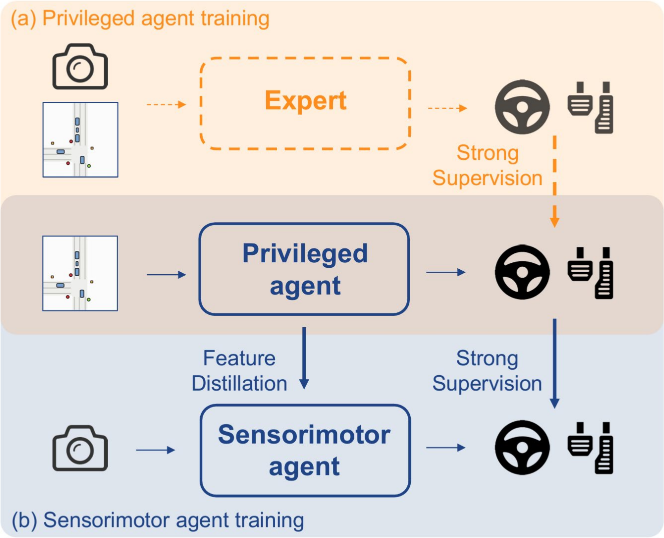
- Significant computational cost reduction
- Related domain knowledge is shared within the shared model

Challenges

- The optimal combination of auxiliary tasks and the appropriate weighting of their losses
- Construct large-scale datasets with multiple types of aligned and high-quality annotations

挑战（5/8）- Policy Distillation

The popular “Teacher-Student” IL Paradigm



- Expert: Ground Truth (GT) to action

  Gap

  Student: Image to action

- **Expert** (by RL/IL/hand-rule, gt input)
  - Not/Can’t perfect, even for a certain benchmark

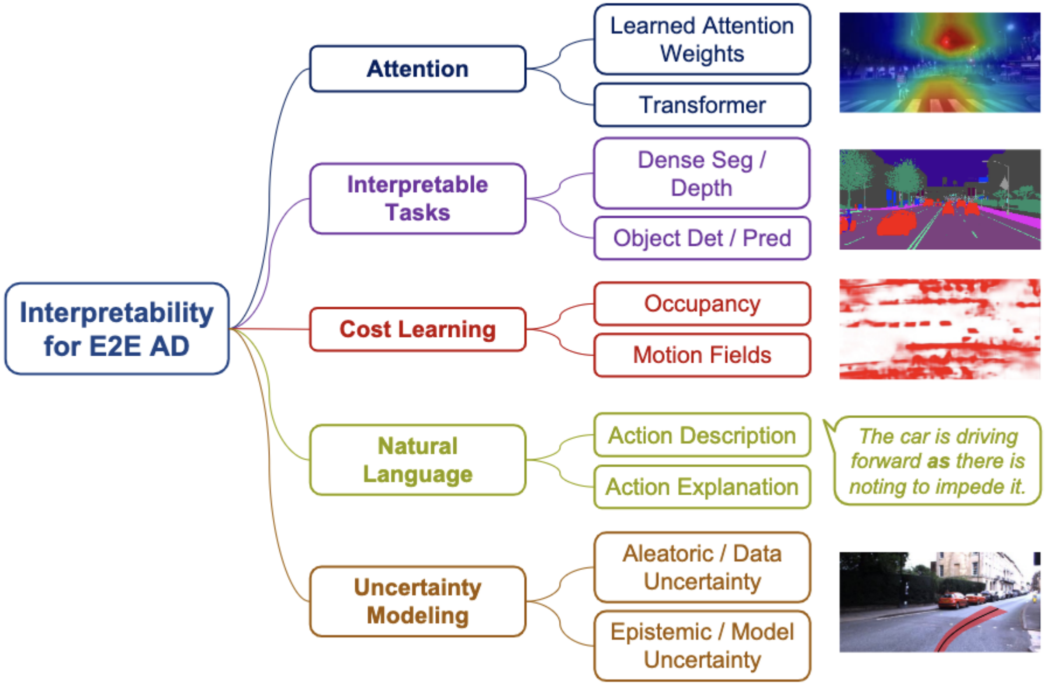
Method	Input	Driving Score ↑
Transfuser [39, 8]	Camera + LiDAR	31.0
LAV [3]	Camera + LiDAR	46.5
Student Model + Frozen Roach	Camera + LiDAR	8.9
Roach [55]	Privileged Info.	74.2
Roach + Rule [50]	Privileged Info.	**87.0**

From DriveAdapter work, ICCV 2023

- What for or How to **Distillation**
  - **Critical** features
  - Input gap - Casual confusion

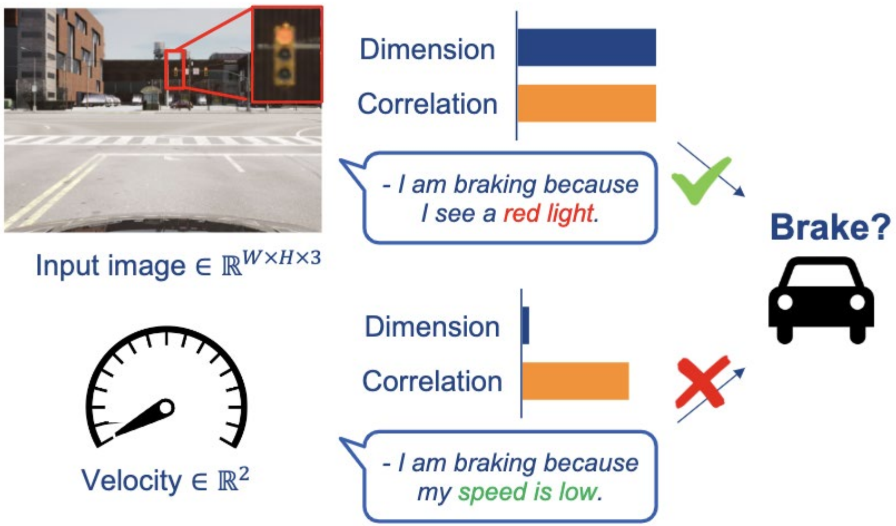
挑战（6/8）- Interpretability

Summary of the different forms of interpretability



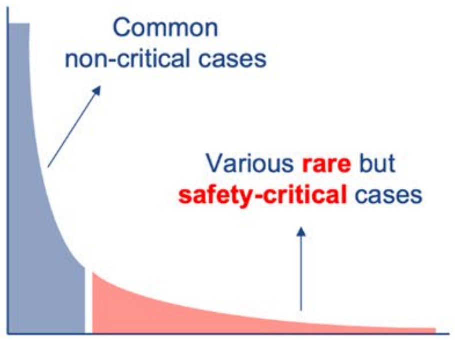
They aid in human comprehension of the **decision-making processes** of end-to-end models, **perception failures**, and the **reliability of the outputs**.

挑战（7/8） - Causal Confusion

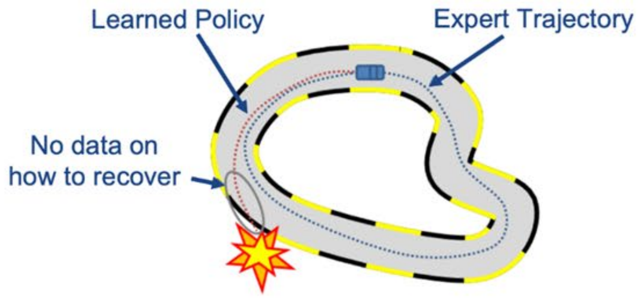


- Driving is a task that exhibits **temporal smoothness**, which makes past motion a reliable predictor of the next action.
- However, methods trained with **multiple frames** can become overly reliant on this shortcut. This is referred to as the **copycat problem** and is a manifestation of **causal confusion.**

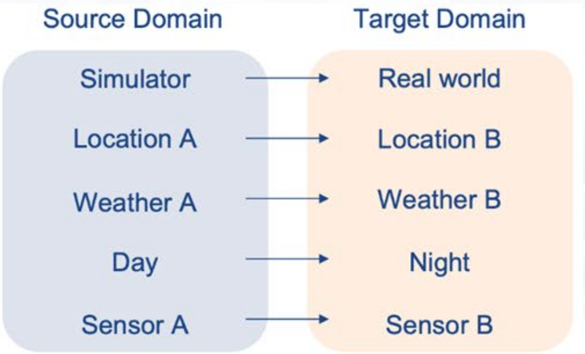
挑战（8/8） - Robustness and Generalization



(a) Long-tailed Distribution



(b) Covariate Shift



(c) Domain Adaptation

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

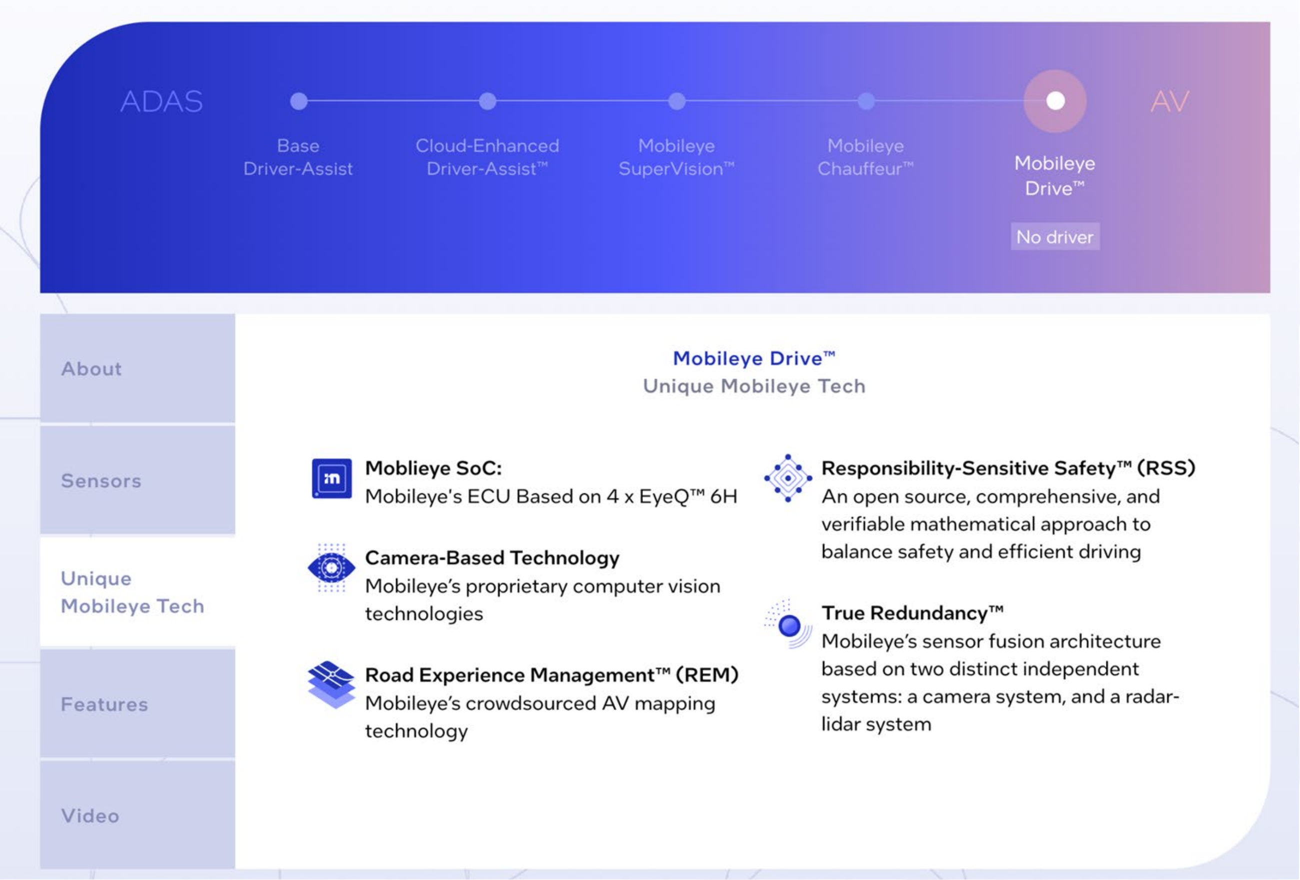
Mobileye

https://www.mobileye.com/ces-2024/

Mobileye 技术路线

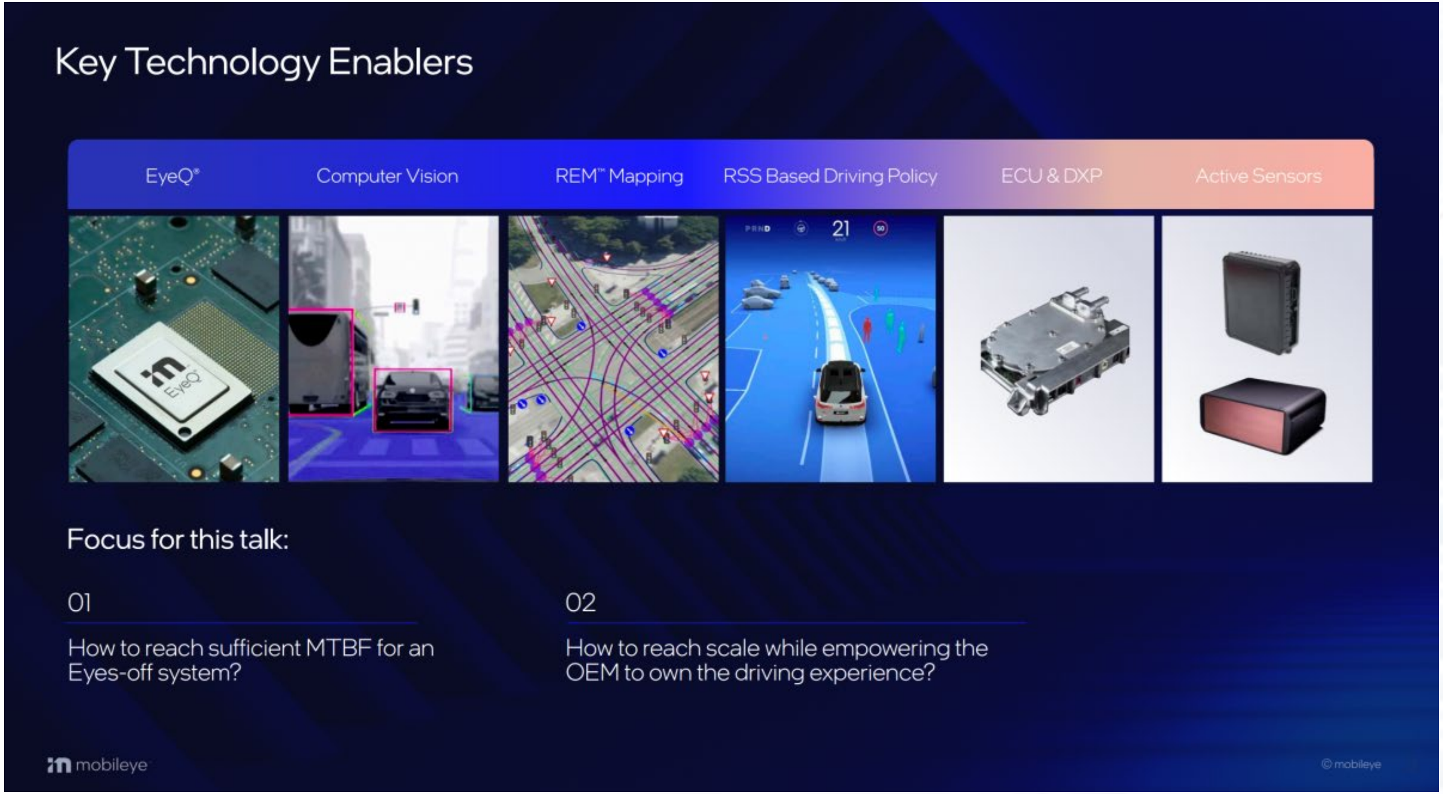
Mobileye 技术路线

https://www.mobileye.com/ces-2024/



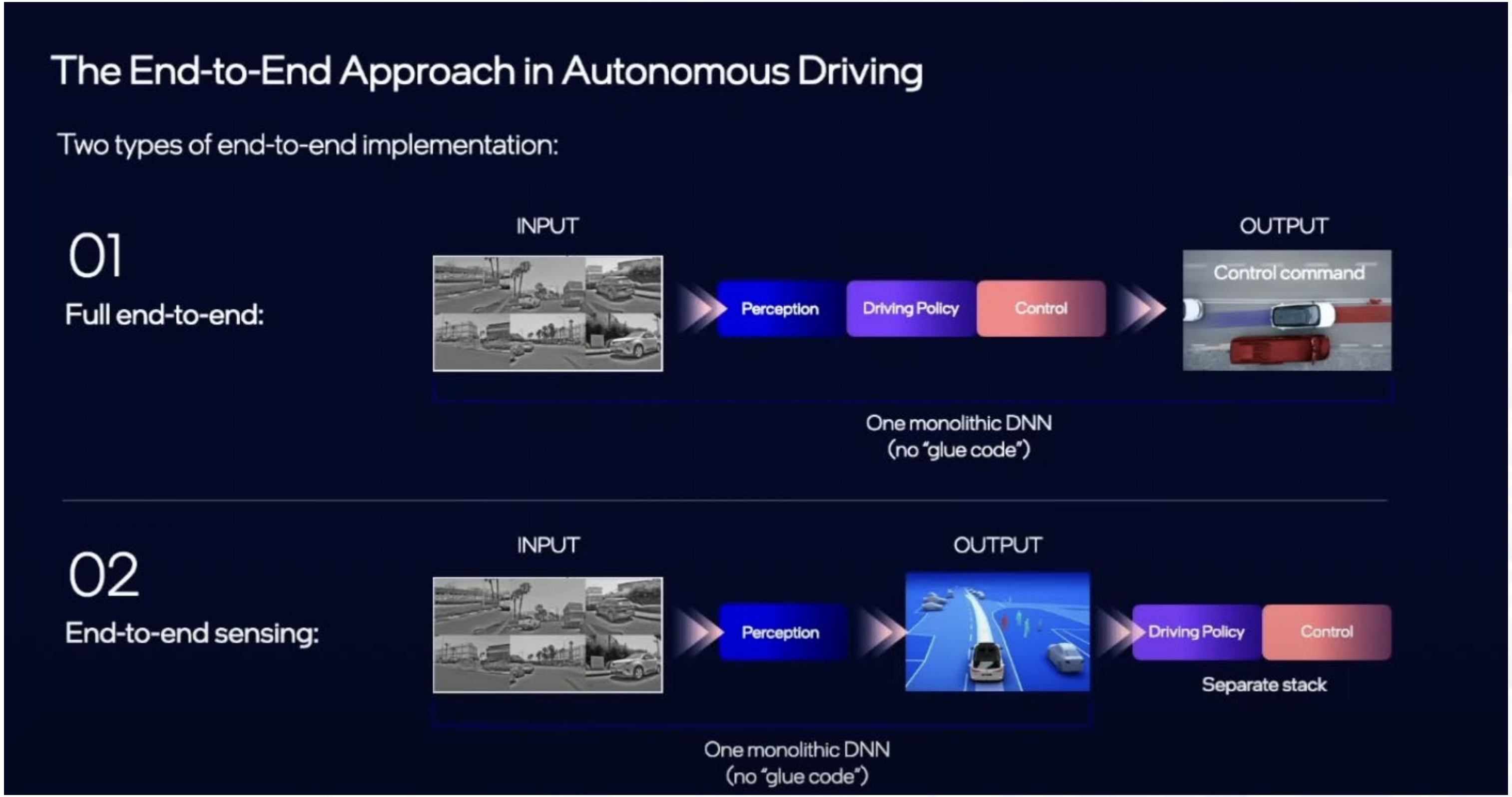
https://www.mobileye.com/ces-2024/

Mobileye 技术路线



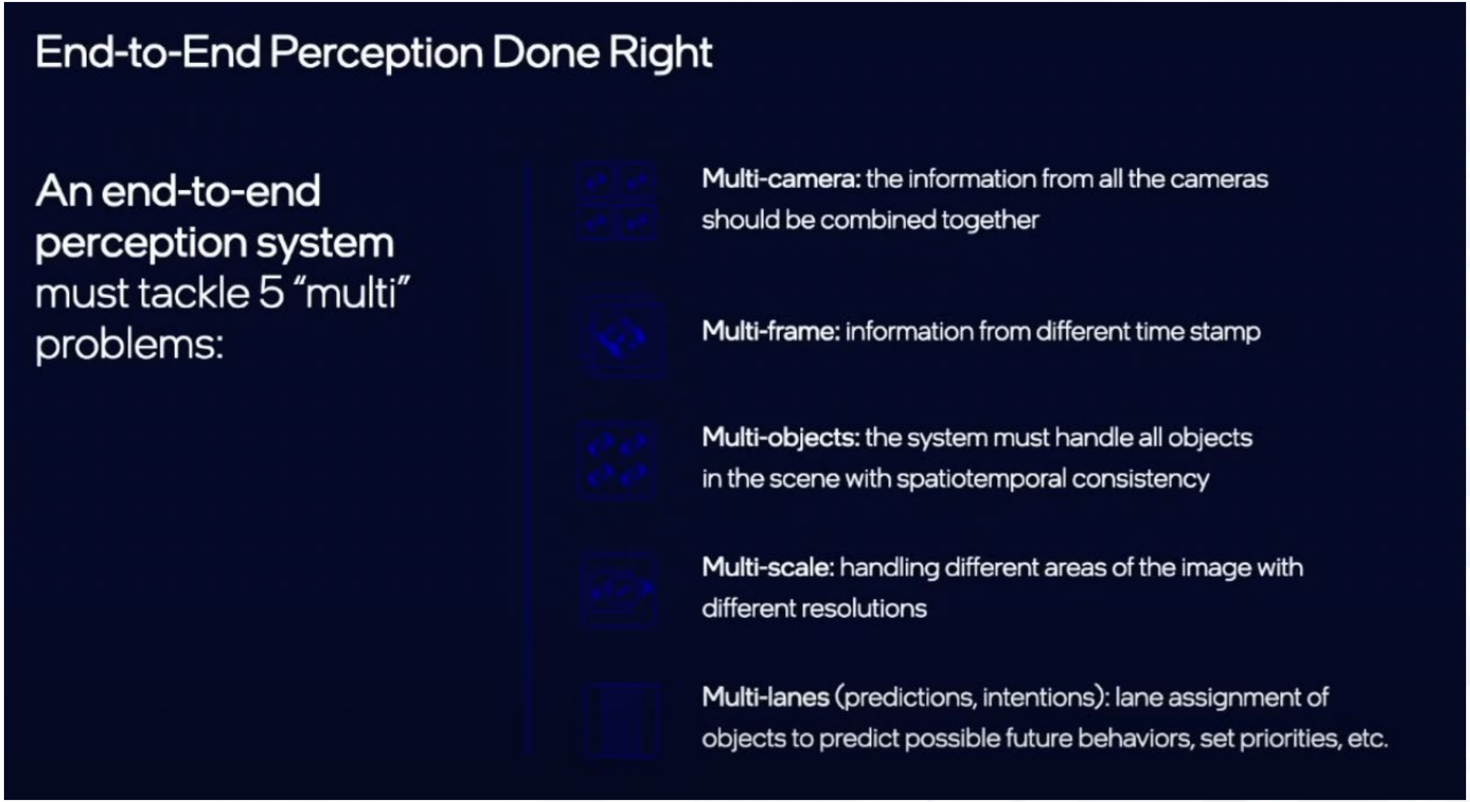
https://www.mobileye.com/ces-2024/

Mobileye 技术路线

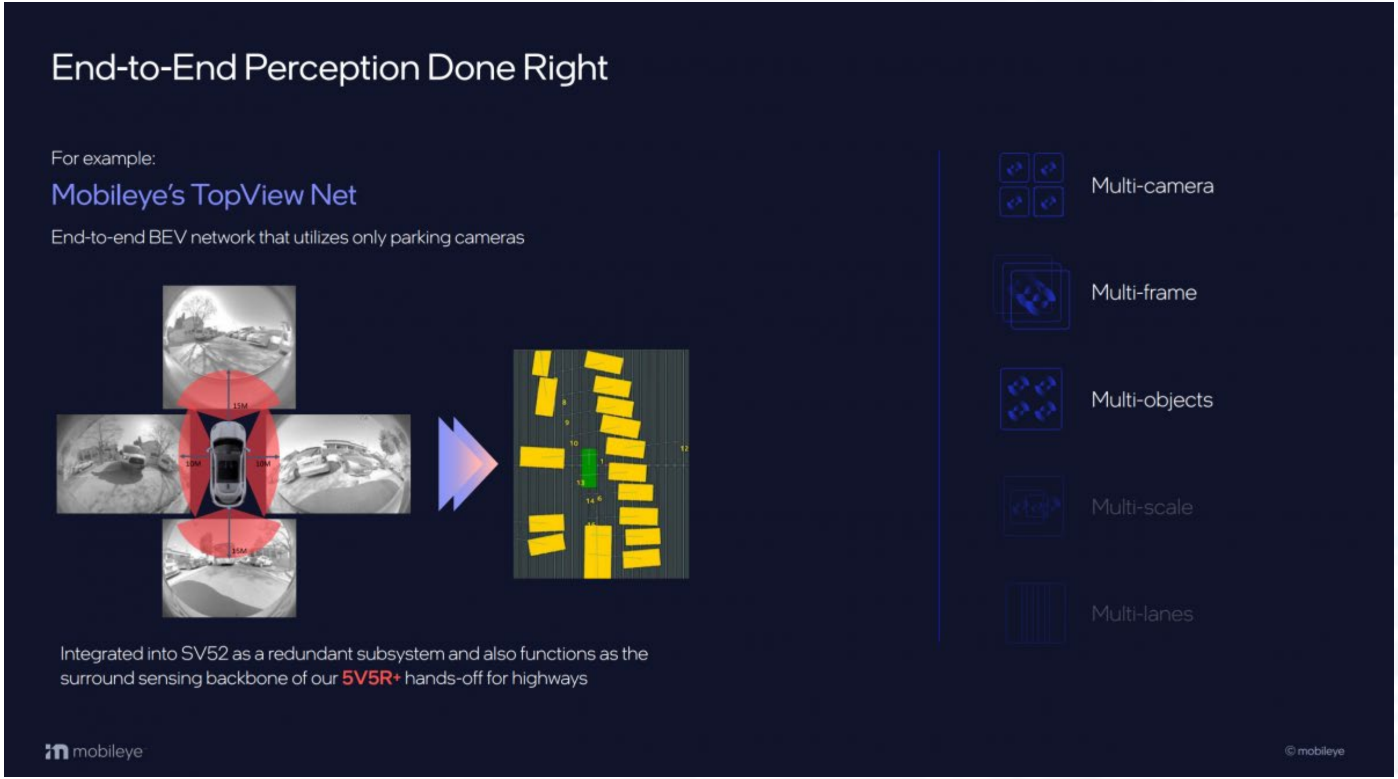


https://www.mobileye.com/ces-2024/

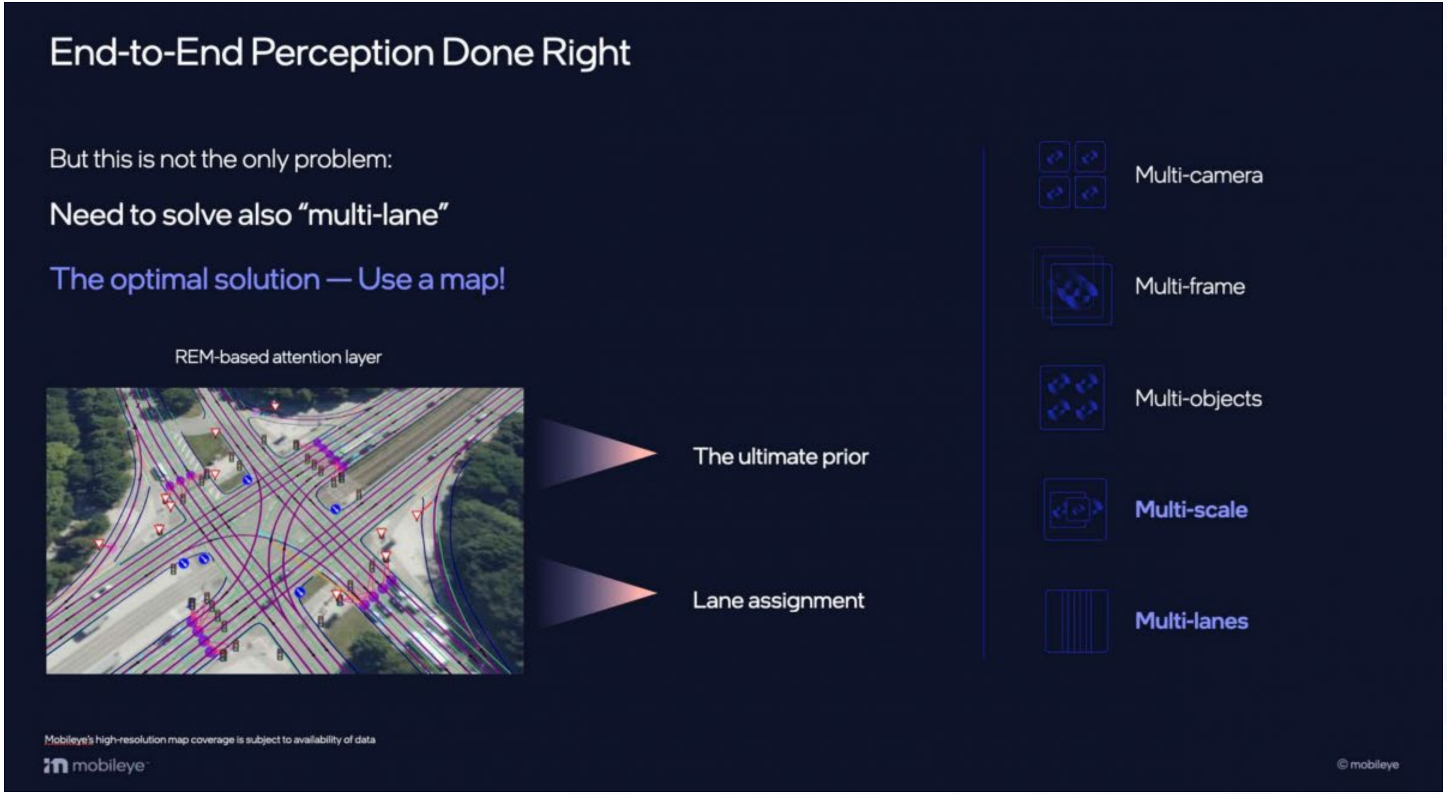
Mobileye - End-to-end Perception Done Right



Mobileye - End-to-end Perception Done Right



Mobileye - End-to-end Perception Done Right



Exercise and Hands-on Project

思考题

在近年Tesla AI day中，Tesla FSD在感知部分采用的算法从BEV Layer发展到了Occupancy Network，试比较下两者的关系，有何共通点。

参考文献：

[1] Li, Zhiqi and Wang, Wenhai and Li, Hongyang, et al. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. arXiv:2203.17270

[2] Chen Min and Liang Xiao and Dawei Zhao and Yiming Nie and Bin Dai. UniScene: Multi-Camera Unified Pre-training via 3D Scene Reconstruction. arXiv: 2305.18829