Autonomous Driving

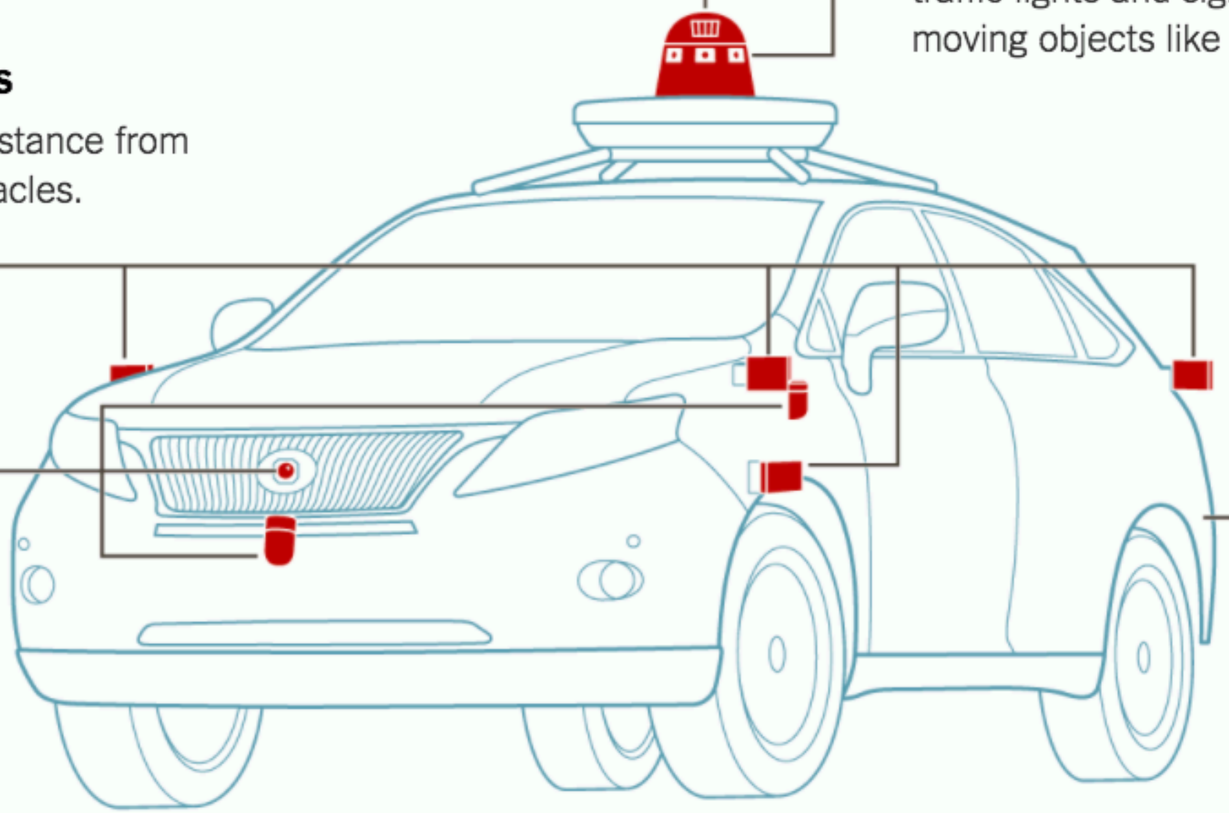
Jerry Lin

Why do we want autonomous driving?

Humans are not always good drivers

- ~3700 people lose their lives everyday on roads
- We want much better drivers than humans

SDVs have better perception



LIDAR UNIT

Constantly spinning, it uses laser beams to generate a 360-degree image of the car's surroundings.

RADAR SENSORS

Measure the distance from the car to obstacles.

ADDITIONAL LIDAR UNITS

CAMERAS

Uses parallax from multiple images to find the distance to various objects. Cameras also detect traffic lights and signs, and help recognize moving objects like pedestrians and bicyclists.

MAIN COMPUTER (LOCATED IN TRUNK)

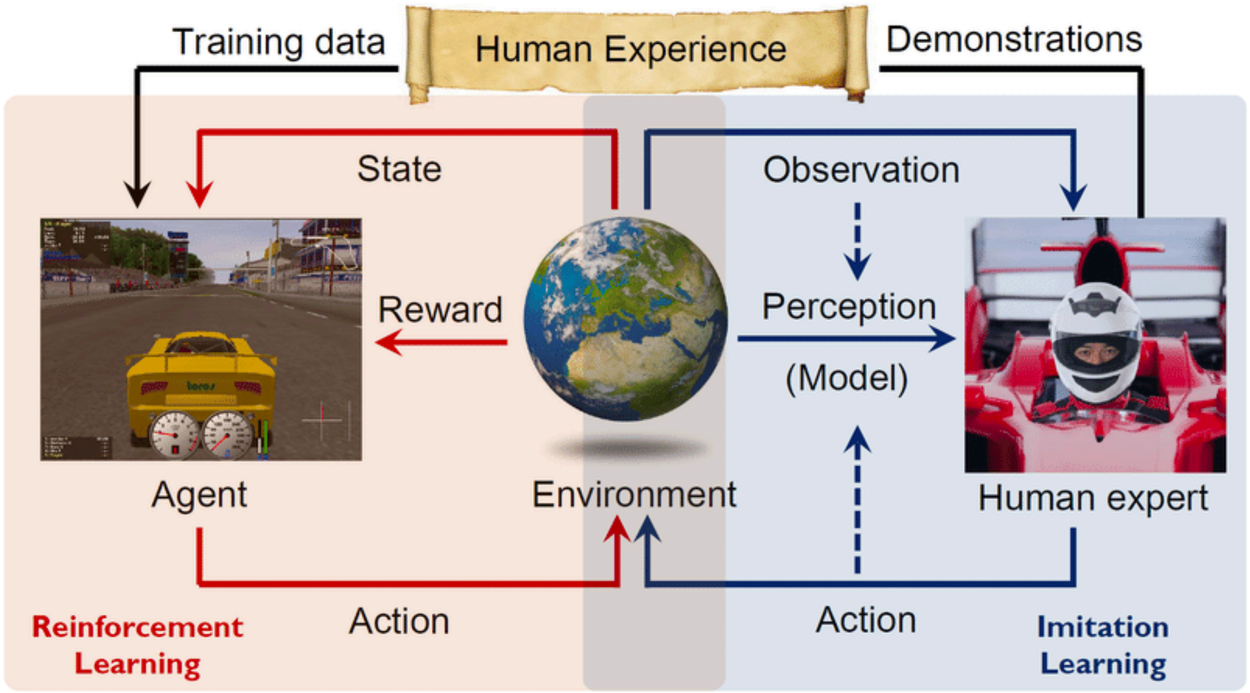
Analyzes data from the sensors, and compares its stored maps to assess current conditions.

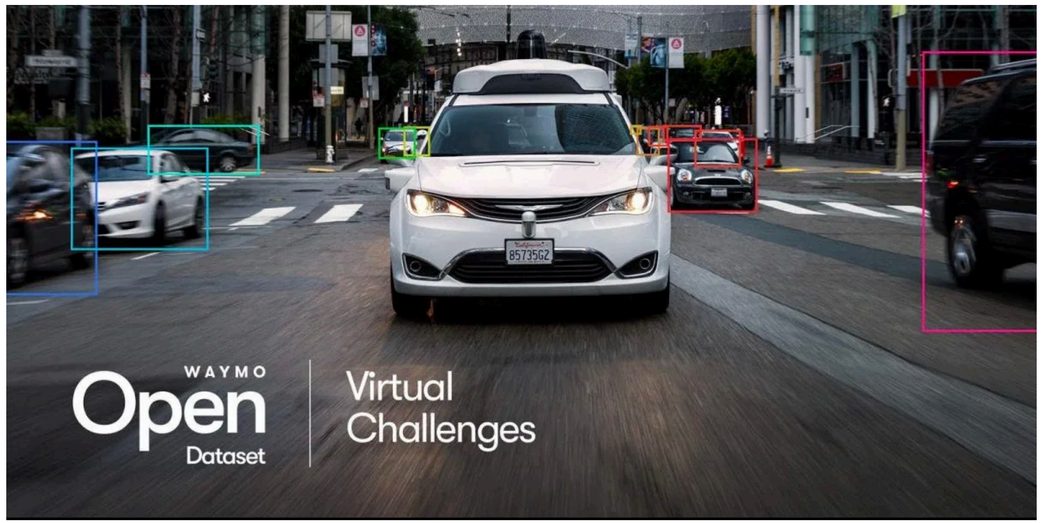
What does SDV ‘see’?

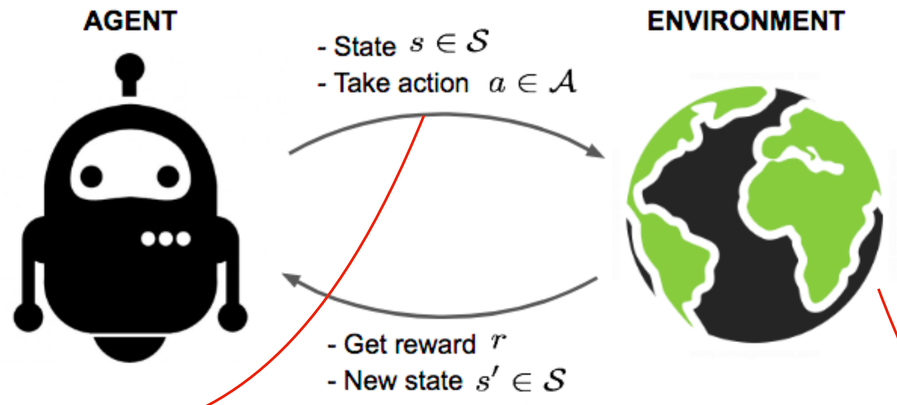
SDVs have better control

What do we have so far?

- Huge real-world dataset
- Algorithms







Learn a policy from data

- Imitation Learning (IL)
  - A detour into sensor fusion
- Reinforcement Learning (RL)
  - Model-free
  - Model-based

Learn to simulate from data

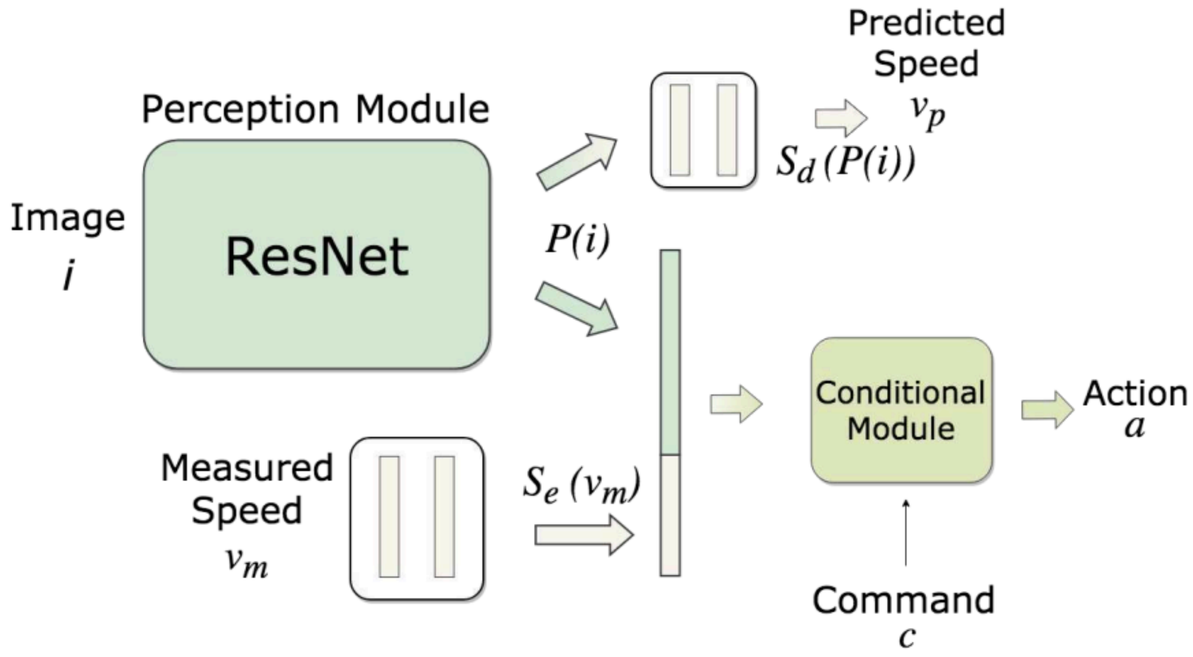
- Trajectory Forecasting
  - From LiDAR points
  - From cameras

State representation

- Rasterized map
- Occupancy field
- Vectorized map

Imitate expert behavior

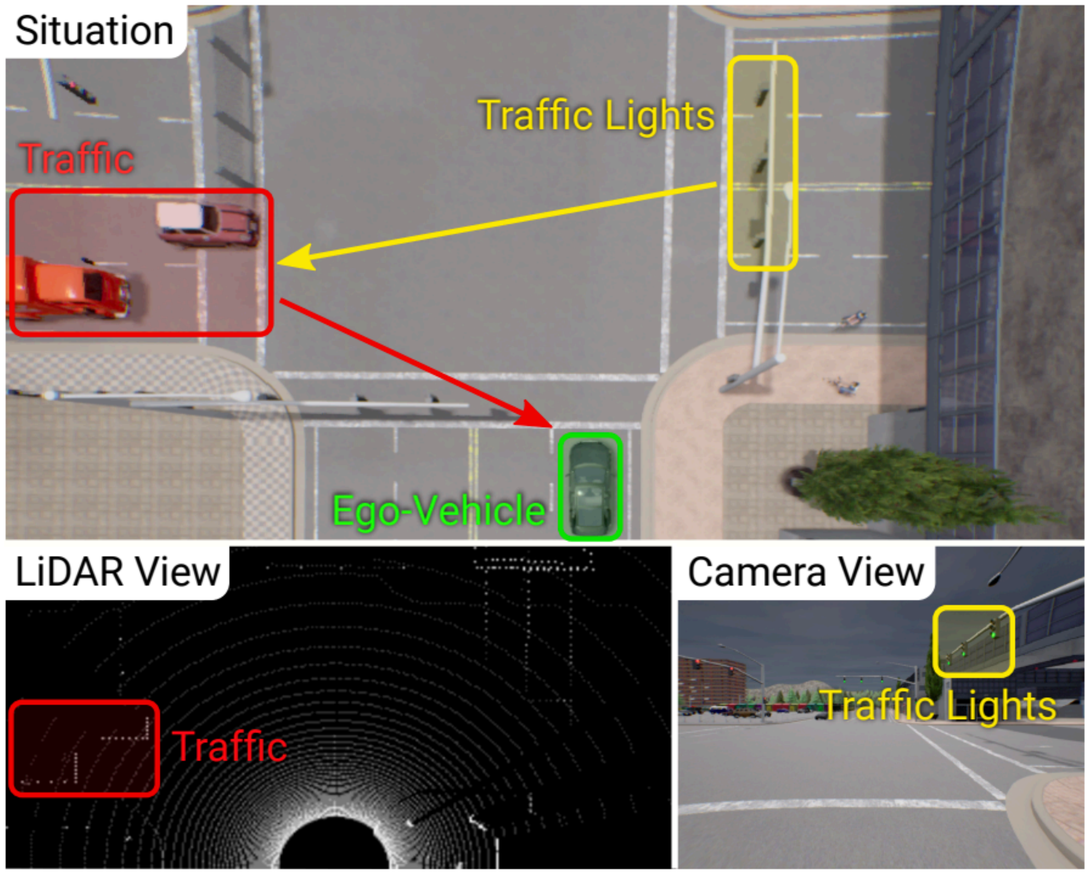
- CIRLS improves Conditional Imitation Learning (CIL)
  - Speed prediction
    - Prevent action predictor to overly rely on current speed
    - L1 loss (instead of MSE)
  - Better backbone
- Common tricks
  - Noise injection (DART)
  - 3-camera trick



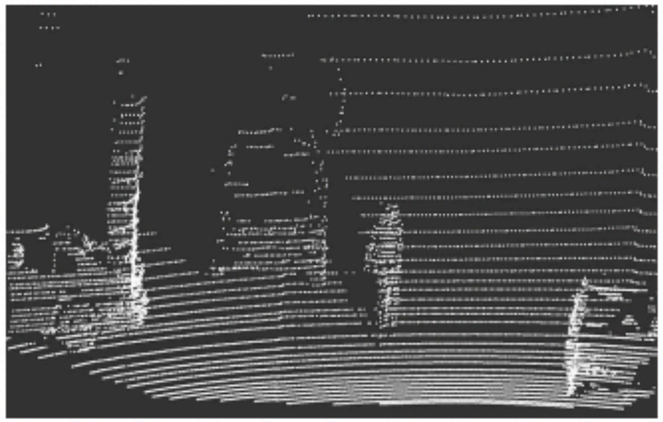
Codevilla et.al. Exploring the Limitations of Behavior Cloning for Autonomous Driving.

Go beyond cameras

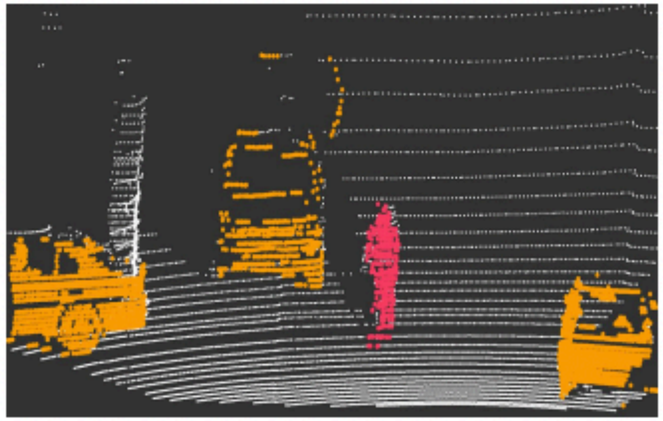
- Get traffic lights from camera
- Get vehicle detections from LiDAR



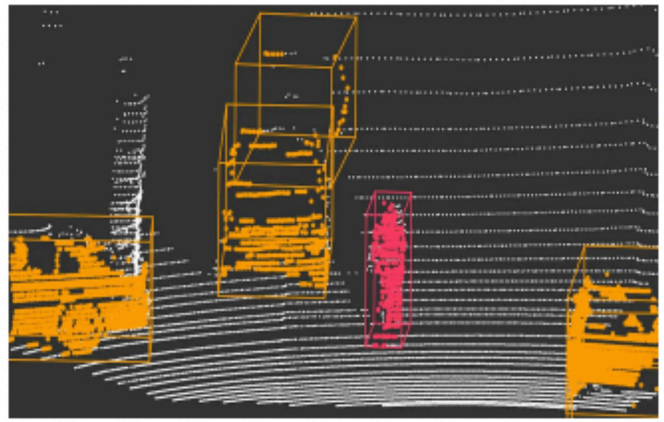
Sensor fusion: PointPainting



2
Point Painting



3
Lidar Detector
e.g.
Point-RCNN
PointPillars
etc

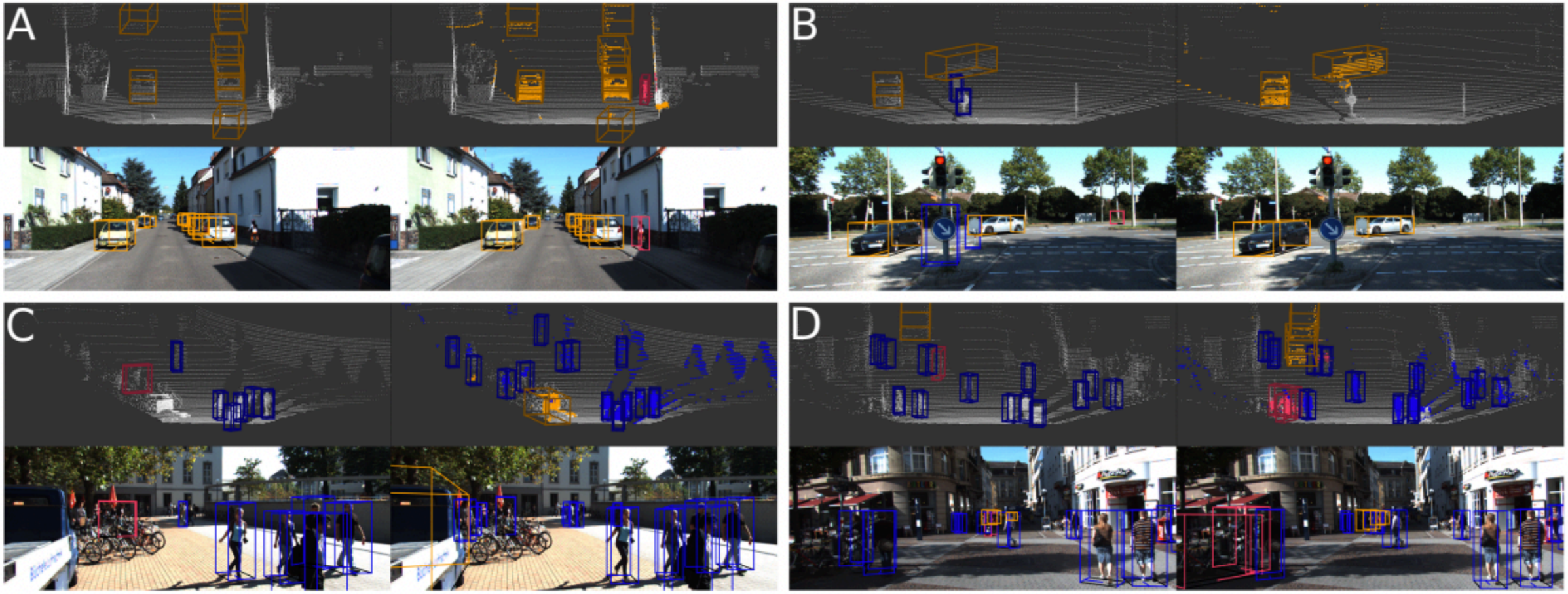


1
Sem. Seg

2 Point Painting

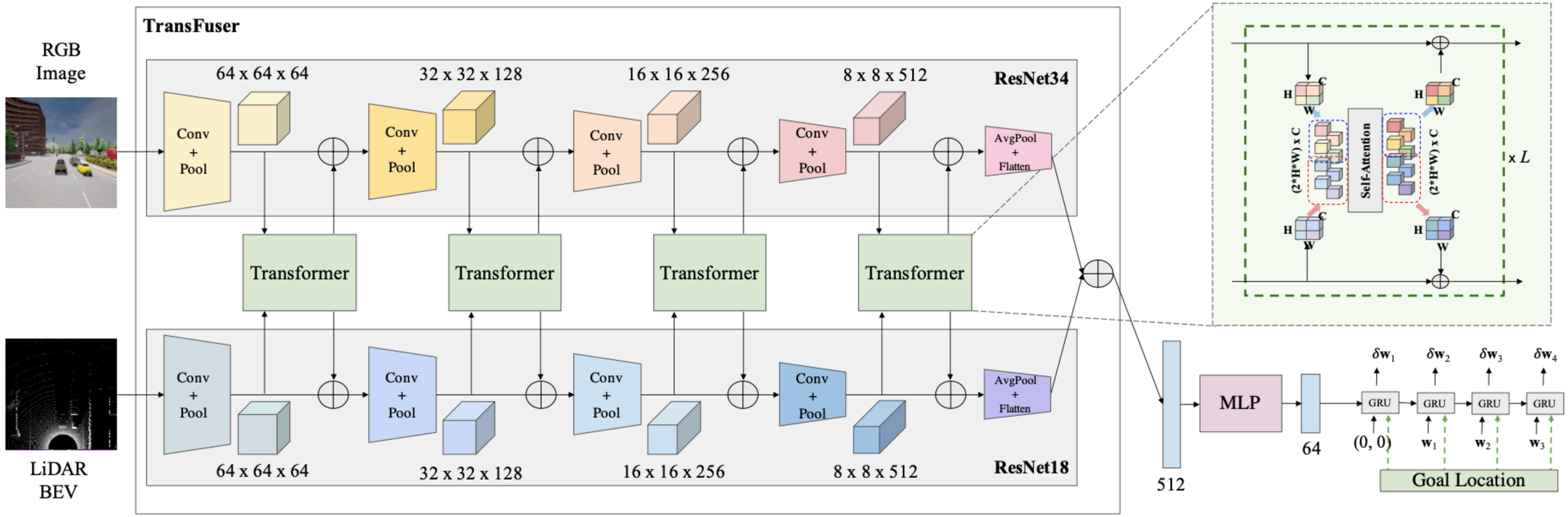
Vora et.al. PointPainting: Sequential Fusion for 3D Object Detection

Results



Sensor fusion: feature-level

- Use image and LiDAR data
- Use Transformer (self-attention) to fuse at multiple resolutions (S*H*W, C)
- Learnable positional embedding to infer spatial dependencies

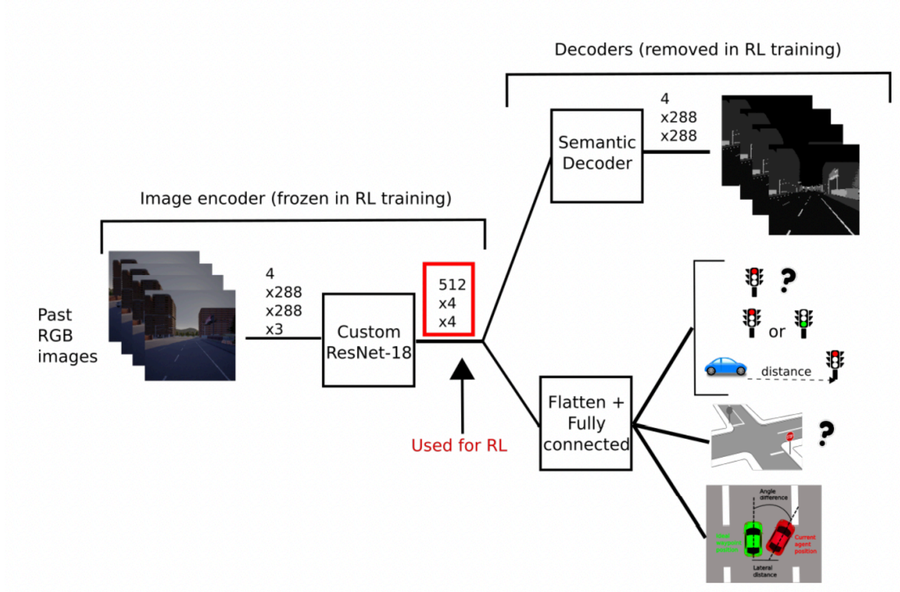


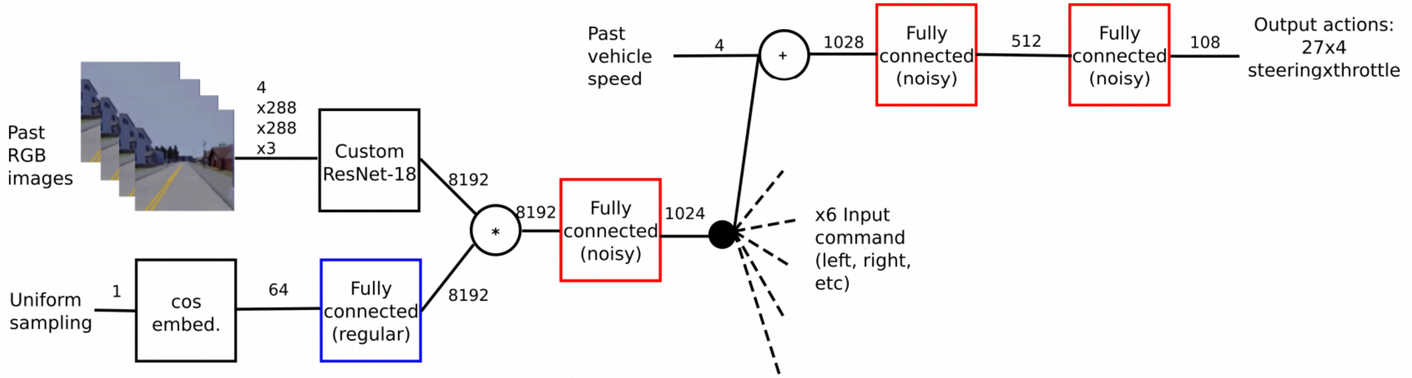
Prakash and Chitta et.al. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving.

Limitations of imitation learning

- Human drivers seldom encounter long-tail cases
- Doesn’t learn from mistakes

Model-free RL





- Use affordances to pre-train the encoder
- Use hidden states (512*4*4) in Rainbow-IQN (distributed deep Q-learning)
- Best model-free RL results so far

$$Q^{\pi}(s,a) = r + \gamma Q^{\pi}(s', \pi(s'))$$

$$\delta = Q(s,a) - (r + \gamma \max_{a} Q(s',a))$$

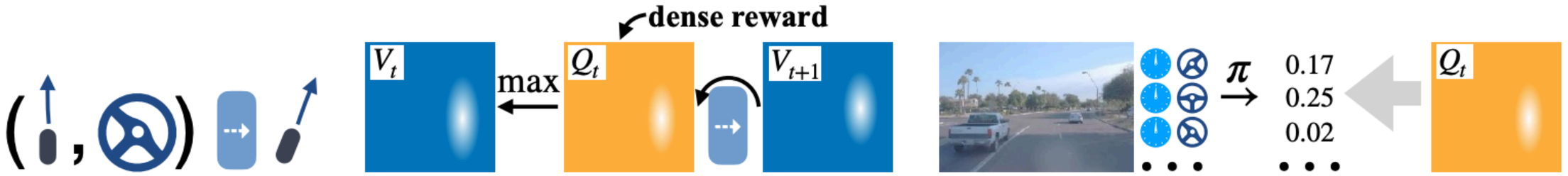
Toromanoff et.al. End-to-End Model-Free Reinforcement Learning for Urban Driving using Implicit Affordances.

Model-based RL

- Assume the world is independent of ego vehicle's behaviors
- Bellman update is simplified to ego model + log replay

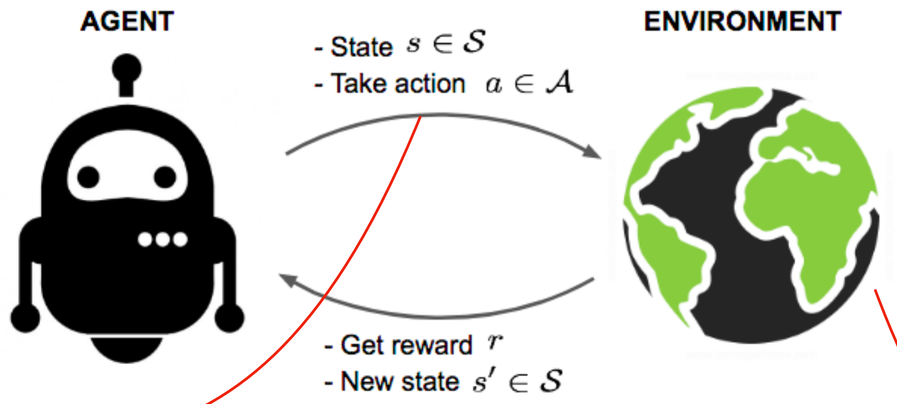
$$V(L_t^{ego}, \hat{L}_t^{world}) = \max_a Q(L_t^{ego}, \hat{L}_t^{world}, a)$$

$$Q(L_t^{ego}, \hat{L}_t^{world}, a_t) = r(L_t^{ego}, \hat{L}_t^{world}, a_t) + \gamma V(\mathcal{T}^{ego}(L_t^{ego}, \hat{L}_t^{world}, a), \hat{L}_{t+1}^{world}).$$



(a) Forward model (b) Bellman update (c) Distillation

Chen et.al. Learning to drive from a world on rails.



Learn a policy from data

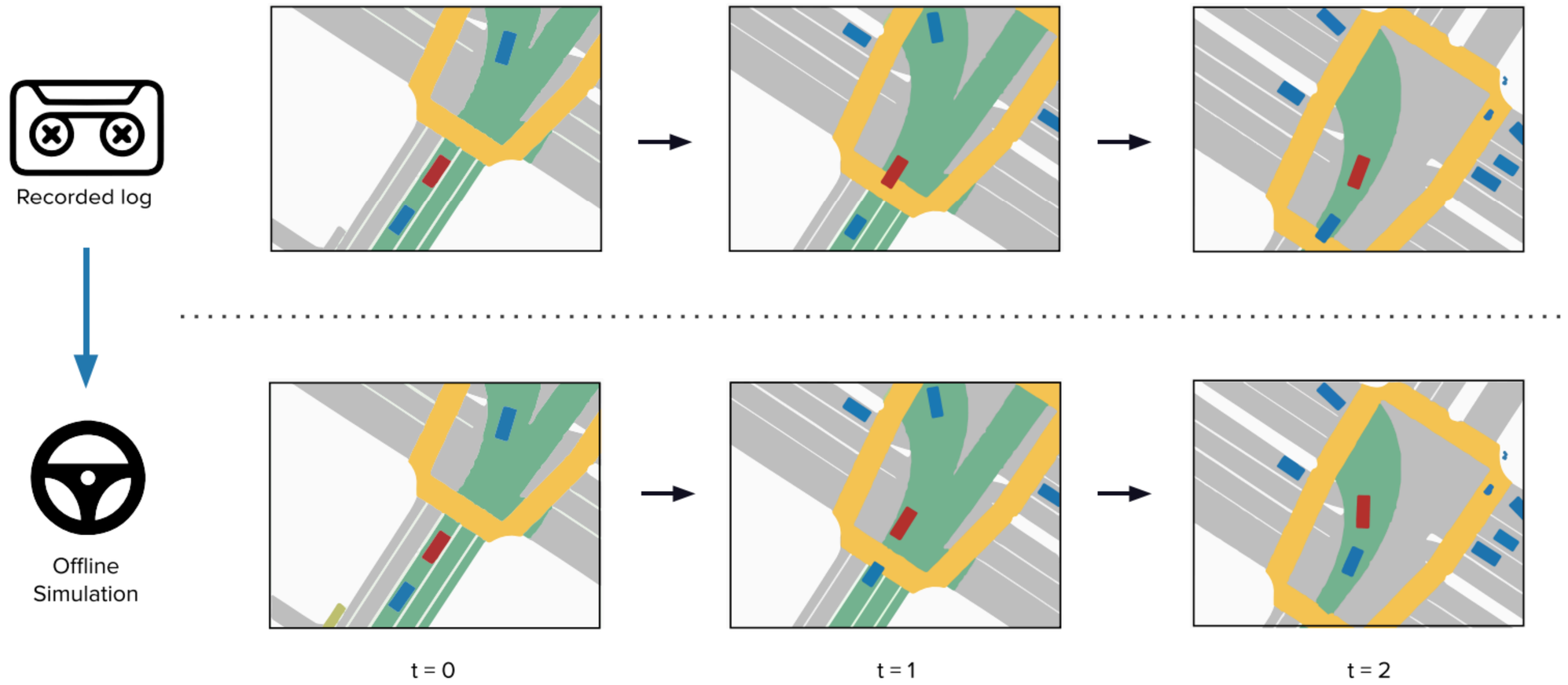
Learn to simulate from data

- Trajectory Forecasting
  - From LiDAR points
  - From cameras

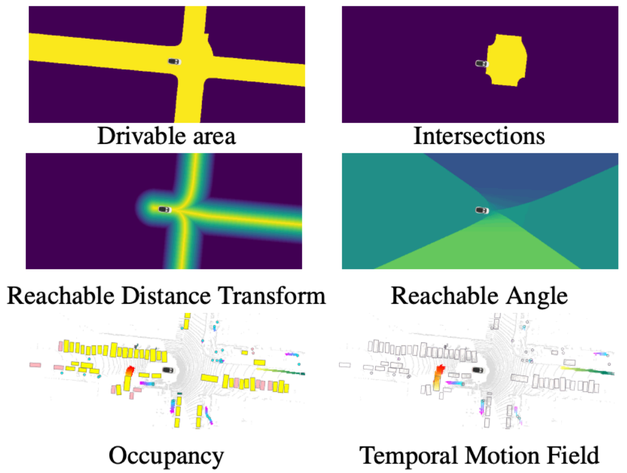
State representation

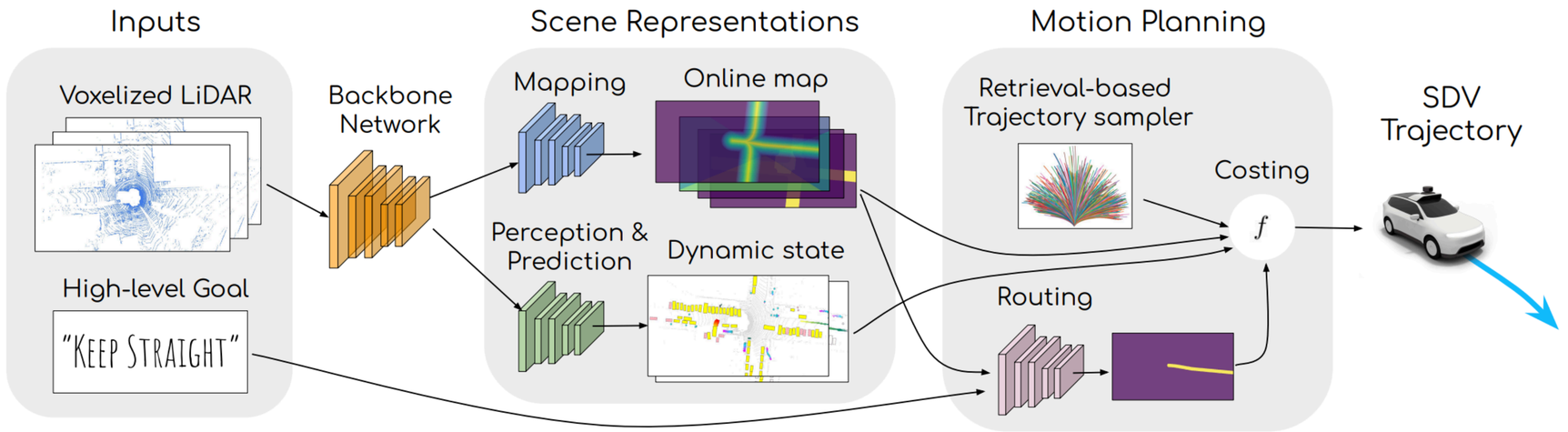
Data-driven closed-loop simulation

- synthesize diverse and realistic driving scenarios with high fidelity and reactivity



Simulation: Trajectory Forecasting from LiDAR





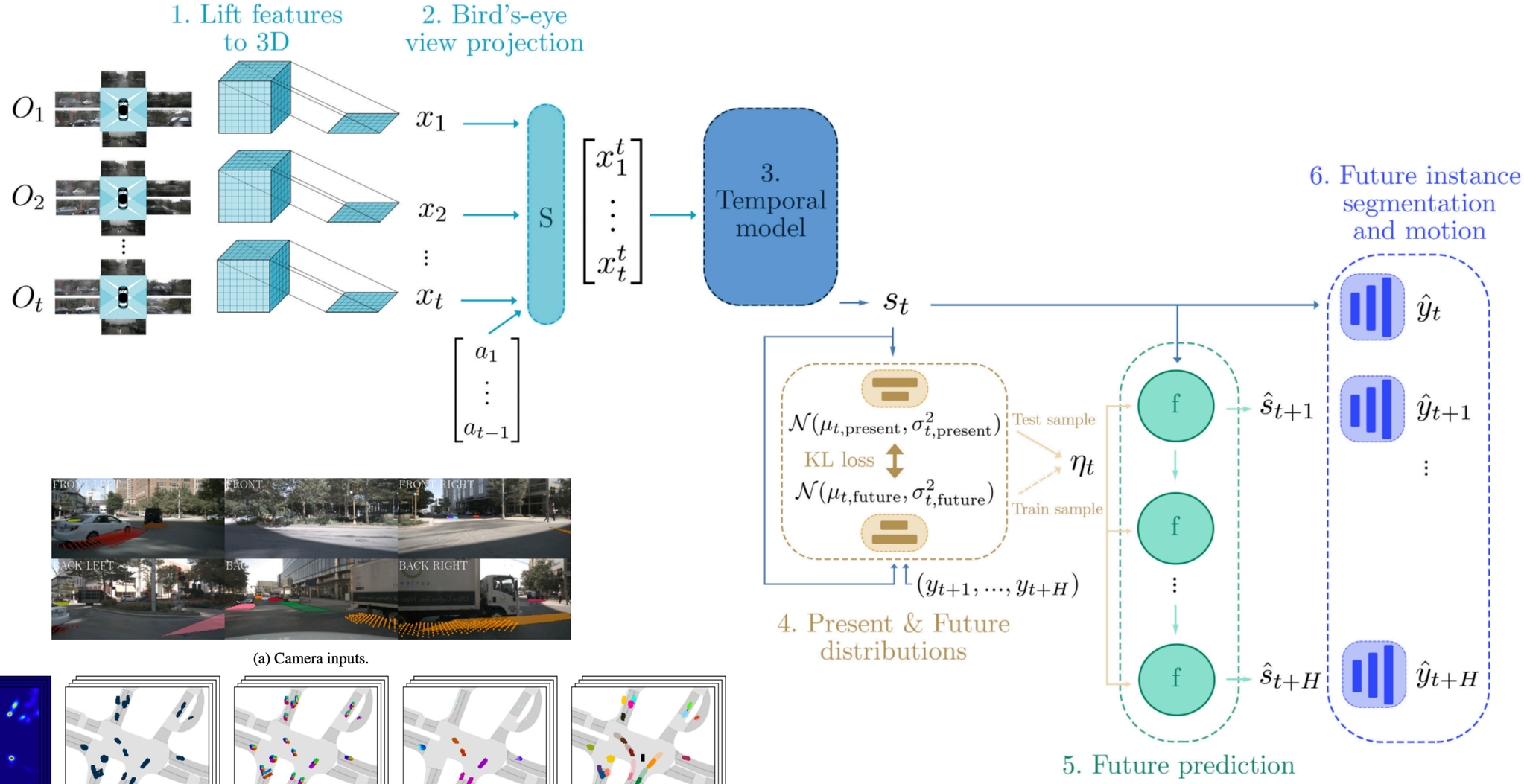
- Probabilistic model for state representation
- Motion representation
  - K motion vectors for each class and spatial temporal loc $\{\mathcal{V}^c_{t,i,k} : k \in 1 \dots K\}$

$$p(\mathcal{F}^c_{(t,i_1)\to(t+1,i_2)}) = \sum_k p(\mathcal{O}^c_{t,i_1}) p(\mathcal{K}^c_{t,i_1} = k) p(\mathcal{V}^c_{t,i_1,k} = i_2)$$

$$p(\mathcal{O}^c_{t+1,i}) = 1 - \prod_j \left(1 - p(\mathcal{F}^c_{(t,j)\to(t+1,i)})\right)$$

Cases & Sadat et.al. MP3: A unified model to Map, Perceive, Predict and Plan

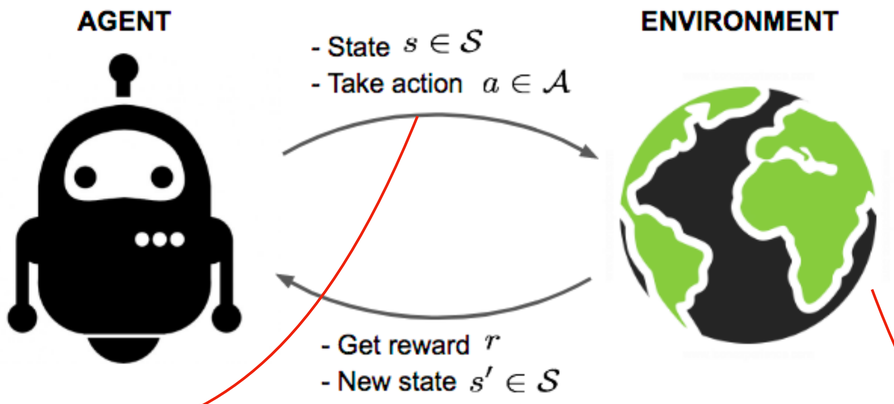
Simulation: Trajectory Forecasting from Images



(a) Camera inputs.

(b) Centerness. (c) Segmentation. (d) Offset. (e) Future flow. (f) Instance segmentation.

Hu et.al. FIERY



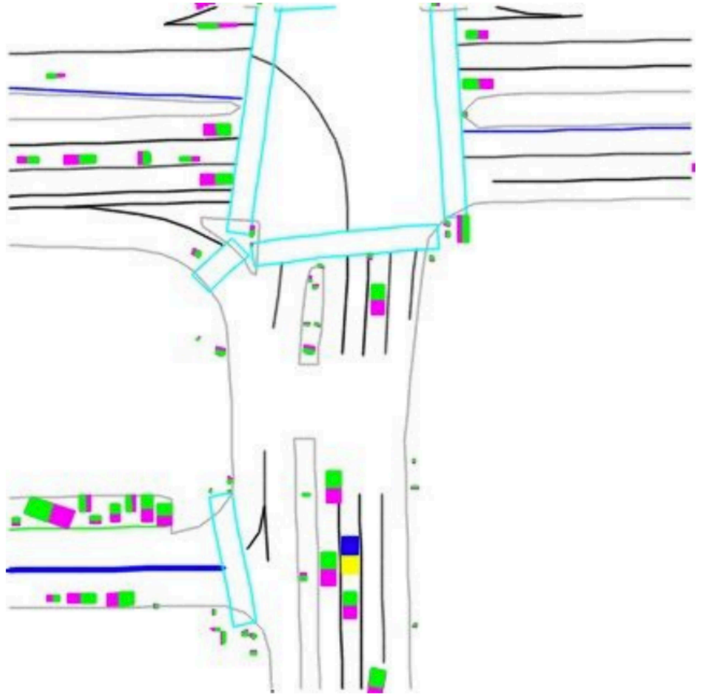
Learn a policy from data

Learn to simulate from data

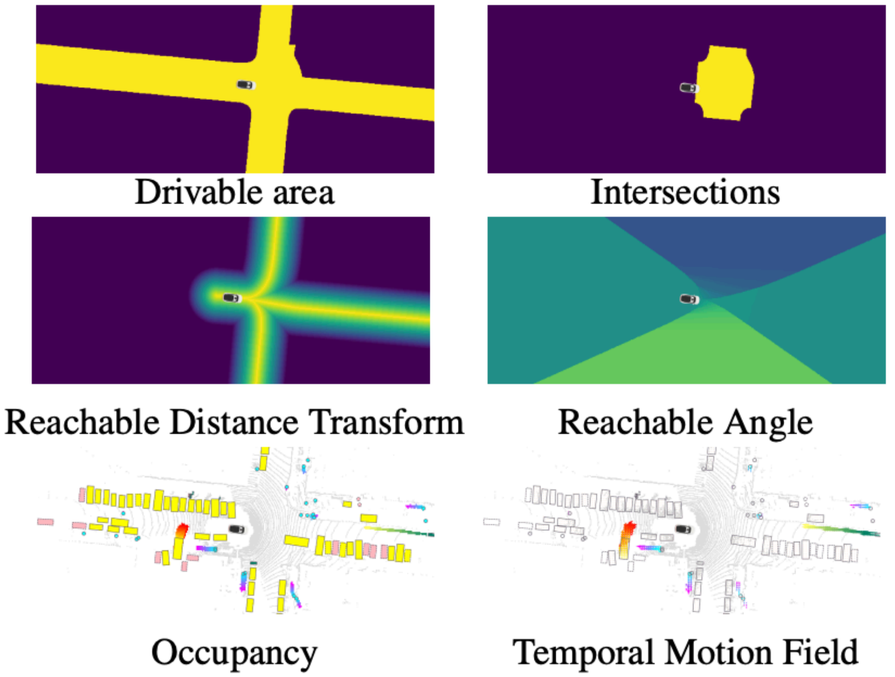
State representation

- Rasterized map
- Occupancy field
- Vectorized map

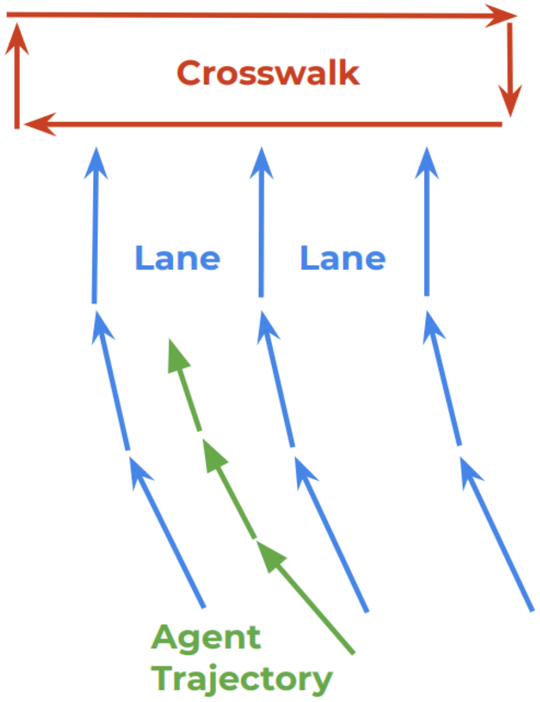
State representation



Rasterized Map



Online Map + Dynamic Occupancy Field



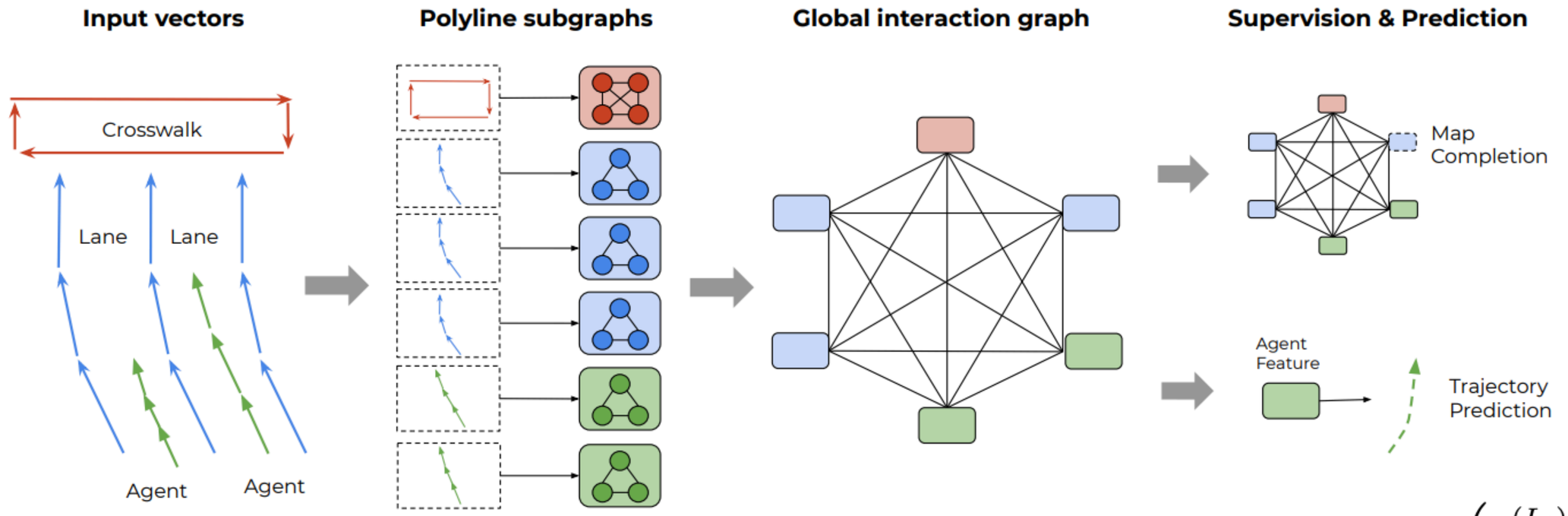
Vectors

Limitations of rasterized representation

- Need larger receptive field (larger resolution/kernel size/crop size)
- Need more computationally efficient representations

Resolution	Kernel	Crop	In-house dataset				Argoverse dataset			
			DE@1s	DE@2s	DE@3s	ADE	DE@1s	DE@2s	DE@3s	ADE
100×100	3×3	1×1	0.63	0.94	1.32	0.82	1.14	2.80	5.19	2.21
200×200	3×3	1×1	0.57	0.86	1.21	0.75	1.11	2.72	4.96	2.15
400×400	3×3	1×1	0.55	0.82	1.16	0.72	1.12	2.72	4.94	2.16
400×400	3×3	3×3	0.50	0.77	1.09	0.68	1.09	2.62	4.81	2.08
400×400	3×3	5×5	0.50	0.76	1.08	0.67	1.09	2.60	4.70	2.08
400×400	3×3	traj	0.47	0.71	1.00	0.63	1.05	2.48	4.49	1.96
400×400	5×5	1×1	0.54	0.81	1.16	0.72	1.10	2.63	4.75	2.13
400×400	7×7	1×1	0.53	0.81	1.16	0.72	1.10	2.63	4.74	2.13

Simulation w/ HD map: Vector representation



$$\mathbf{v}_i^{(l+1)} = \varphi_{\text{rel}}\left(g_{\text{enc}}(\mathbf{v}_i^{(l)}), \varphi_{\text{agg}}\left(\left\{g_{\text{enc}}(\mathbf{v}_j^{(l)})\right\}\right)\right)$$

$$\left\{\mathbf{p}_i^{(l+1)}\right\} = \text{GNN}\left(\left\{\mathbf{p}_i^{(l)}\right\}, \mathcal{A}\right)$$

$$\hat{\mathbf{p}}_i = \varphi_{\text{node}}\left(\mathbf{p}_i^{(L_t)}\right)$$

$$\mathbf{v}_i^{\text{future}} = \varphi_{\text{traj}}\left(\mathbf{p}_i^{(L_t)}\right)$$

Gao and Sun et.al. VectorNet.

Why don’t we see many SDVs on the road?

AUTOMATION LEVELS OF AUTONOMOUS CARS

LEVEL 0

There are no autonomous features.

LEVEL 1

These cars can handle one task at a time, like automatic braking.

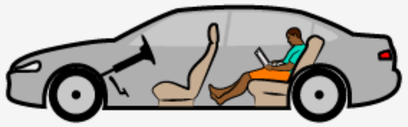
LEVEL 2

These cars would have at least two automated functions.

LEVEL 3

These cars handle “dynamic driving tasks” but might still need intervention.

LEVEL 4



These cars are officially driverless in certain environments.

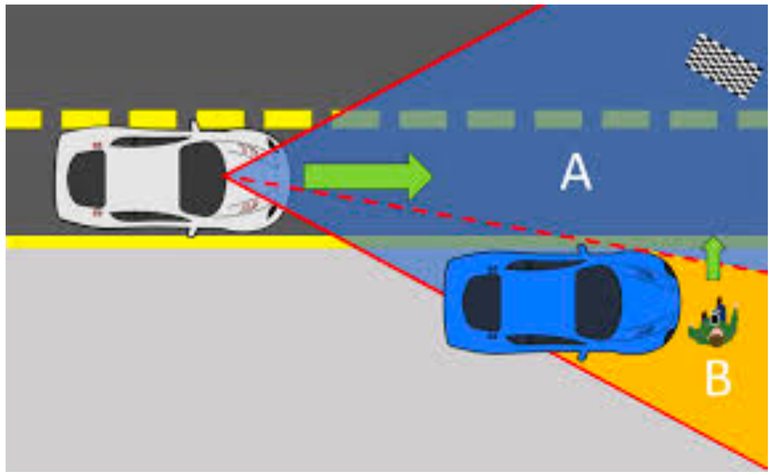
LEVEL 5

These cars can operate entirely on their own without any driver presence.

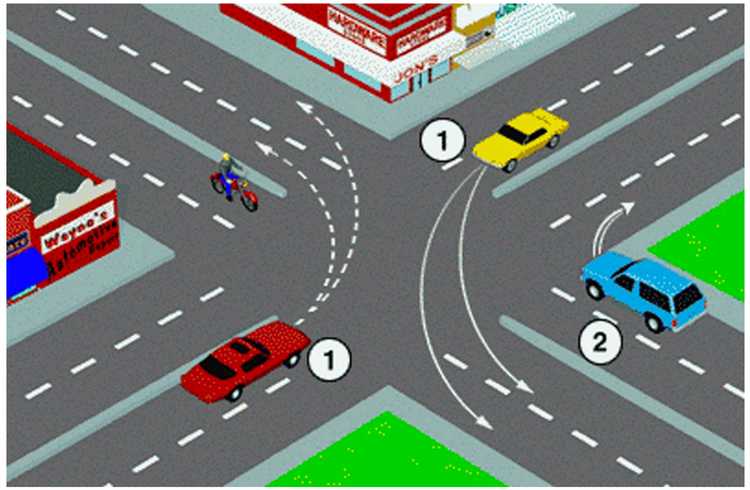
SOURCE: SAE International

BUSINESS INSIDER

Challenges

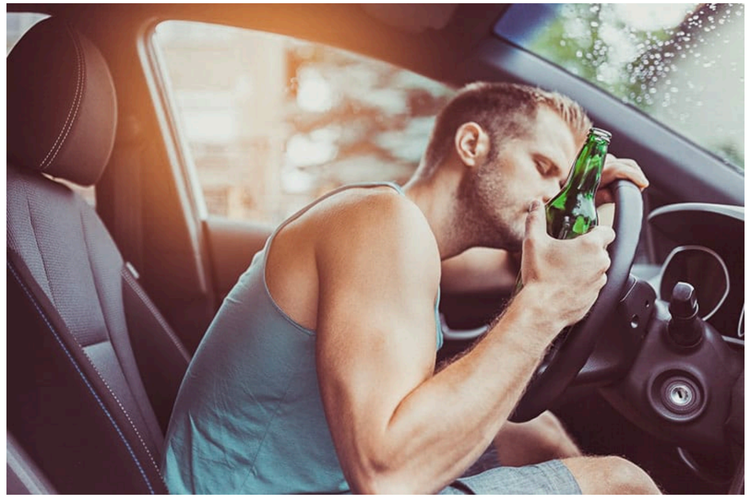


Occlusion



Theory of minds (Interactions)

Long-tail distribution



Multi-modal behaviors

Open problems

- What is the appropriate state representation for simulation and planning? How much should it be learned vs interpretable?
- How should we measure the probability of scenarios? How should we detect outlier, never-before-seen cases?
- What are the limits of what can be trained offline from human demonstrations vs need real-time reasoning using search?
- How much do we need to simulate? How should we measure the performance of the offline simulation itself?
- How much data do we need to train high-performing planning and simulation components? What sensors should we use for large-scale data collection?

Jain et.al. Autonomy 2.0: Why is self-driving always 5 years away?