HUAWEI

СИРИУС

ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР

Trajectory Prediction

19 июля-8 августа 2021

СОВРЕМЕННЫЕ МЕТОДЫ ТЕОРИИ ИНФОРМАЦИИ, ОПТИМИЗАЦИИ И УПРАВЛЕНИЯ

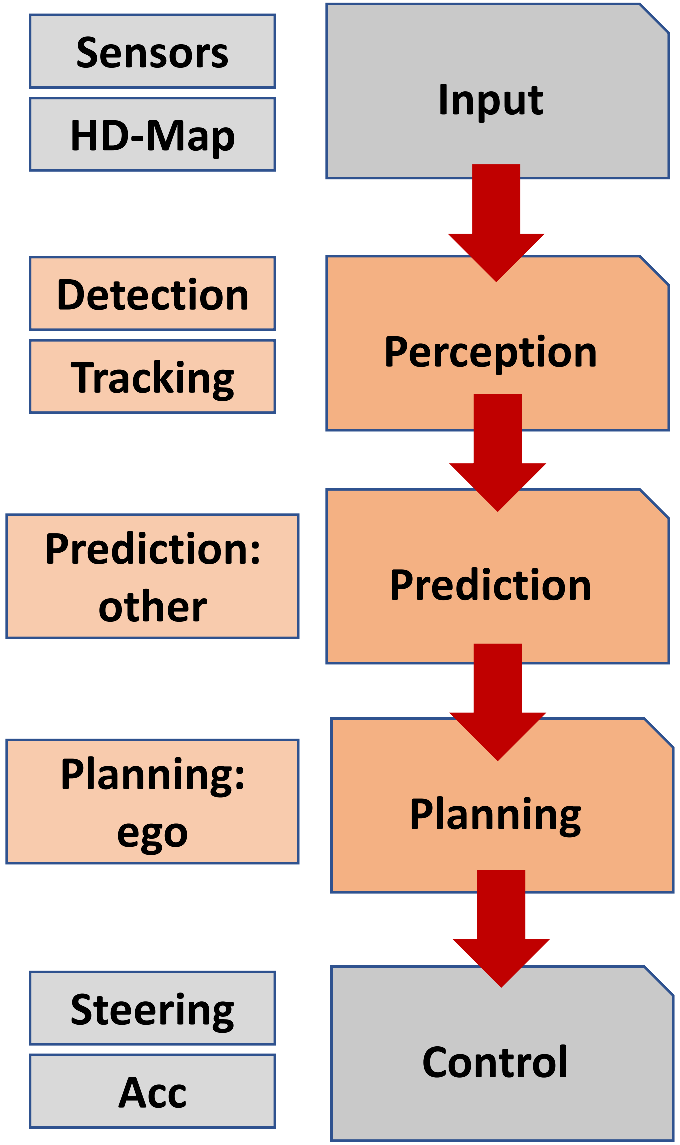
By *Petyushko Alexander, Komkov Stepan*
2021-07-23

SDV: Stack of technologies

- **SDV** = Self-Driving Vehicle
- SDV is one of the most complex tasks theoretically and practically
- Safety is crucial

SDV: Stack of technologies

- The main software parts are the so-called **P^3**:
  - Perception, Prediction and Planning
- Hardware parts:
  - Input: Sensors
  - Output: Control (steering, acceleration)
- High-Definition Map as the helper
  - **HD-Map** contains info about the road

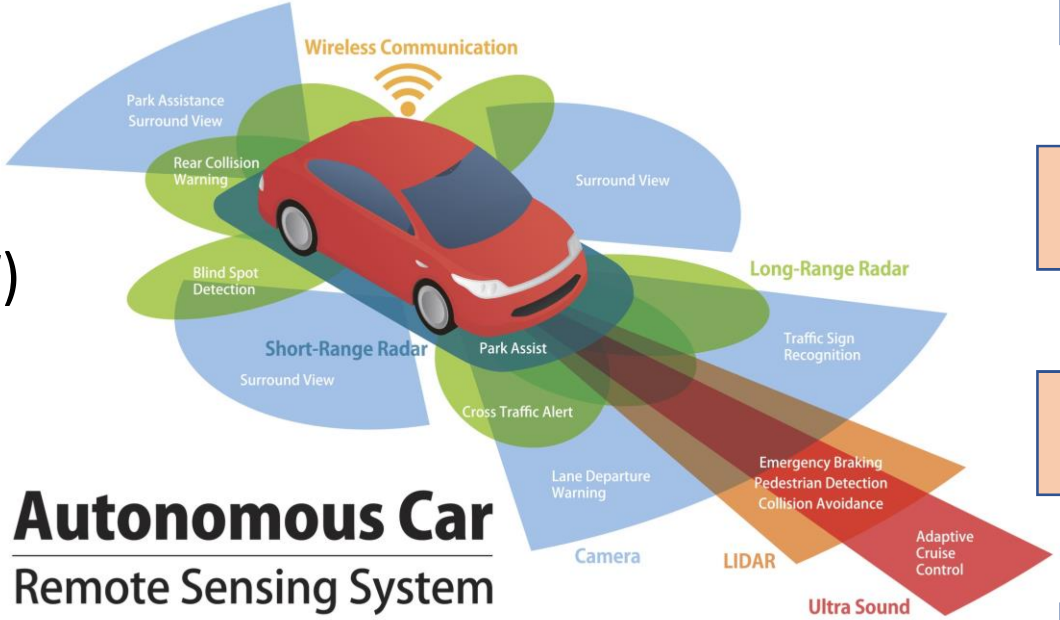


http://www.allaboutselfdriving.com

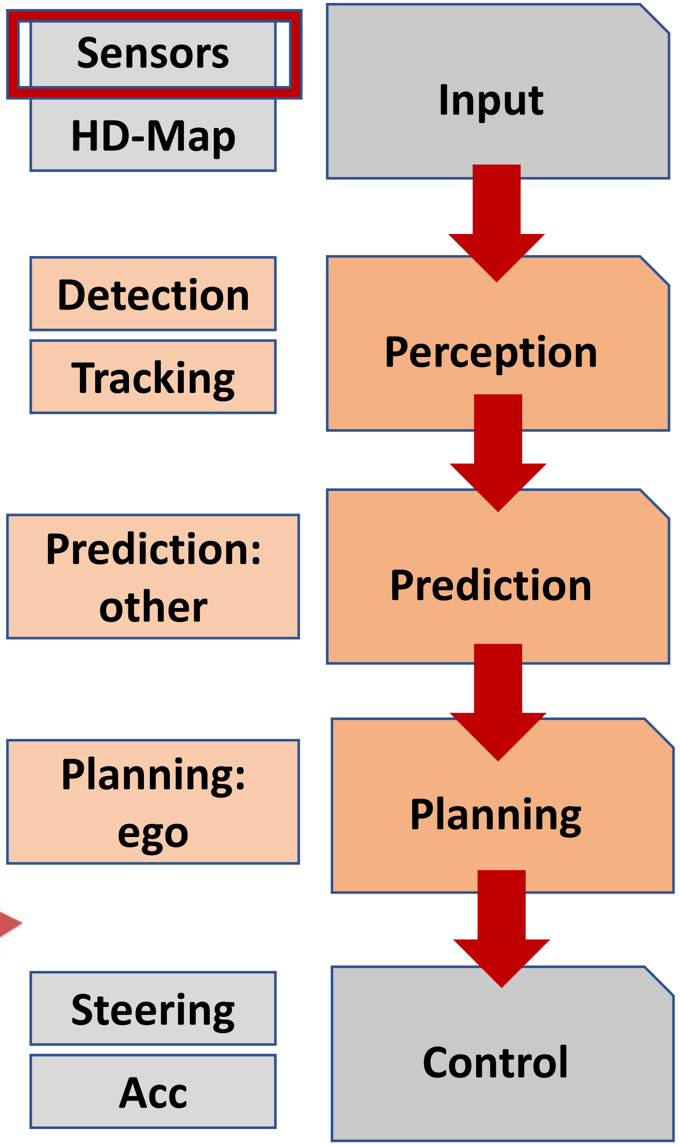
HUAWEI

SDV: Sensors

- Various sensors are used:
  - LIDAR
  - Radar
  - Ultra Sound
  - Cameras (*x N*)

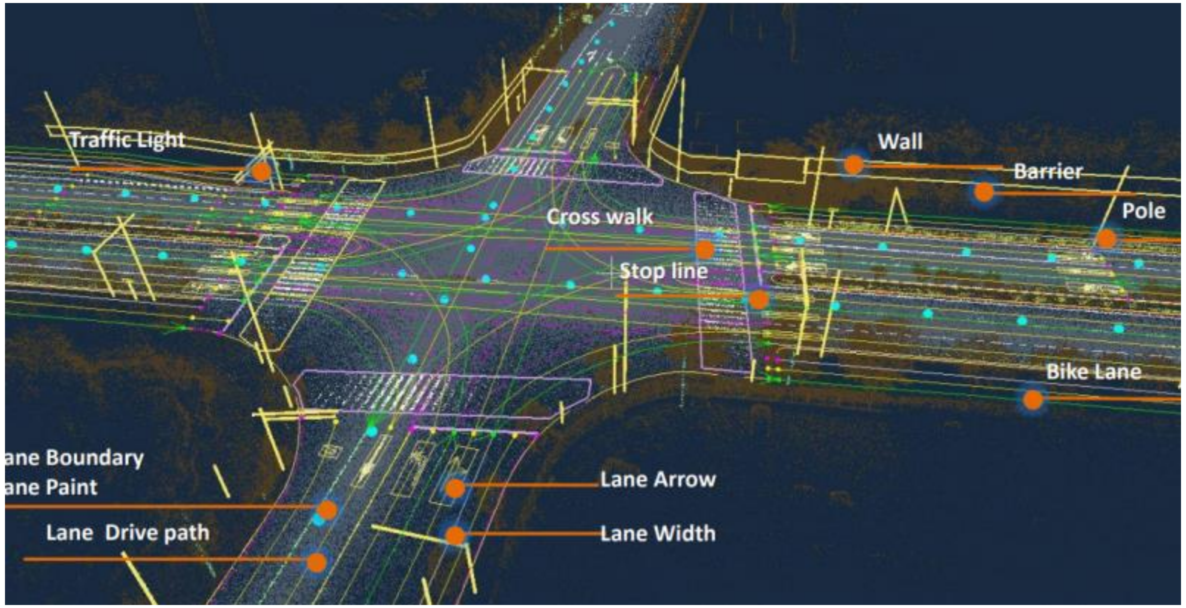


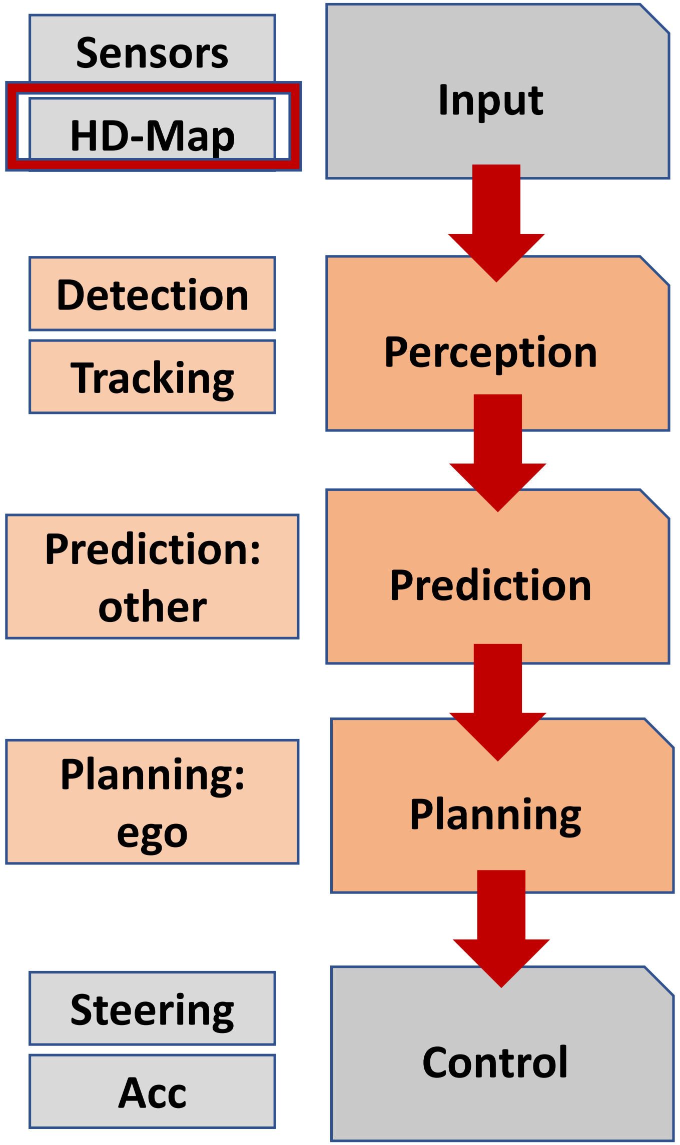
- Expensive
- Hard to synchronize



SDV: HD-Map

- Helpful for prediction and planning
  - Contains information about road:
    - Lanes, crosswalks, traffic lights, etc.
- Every company has its own format

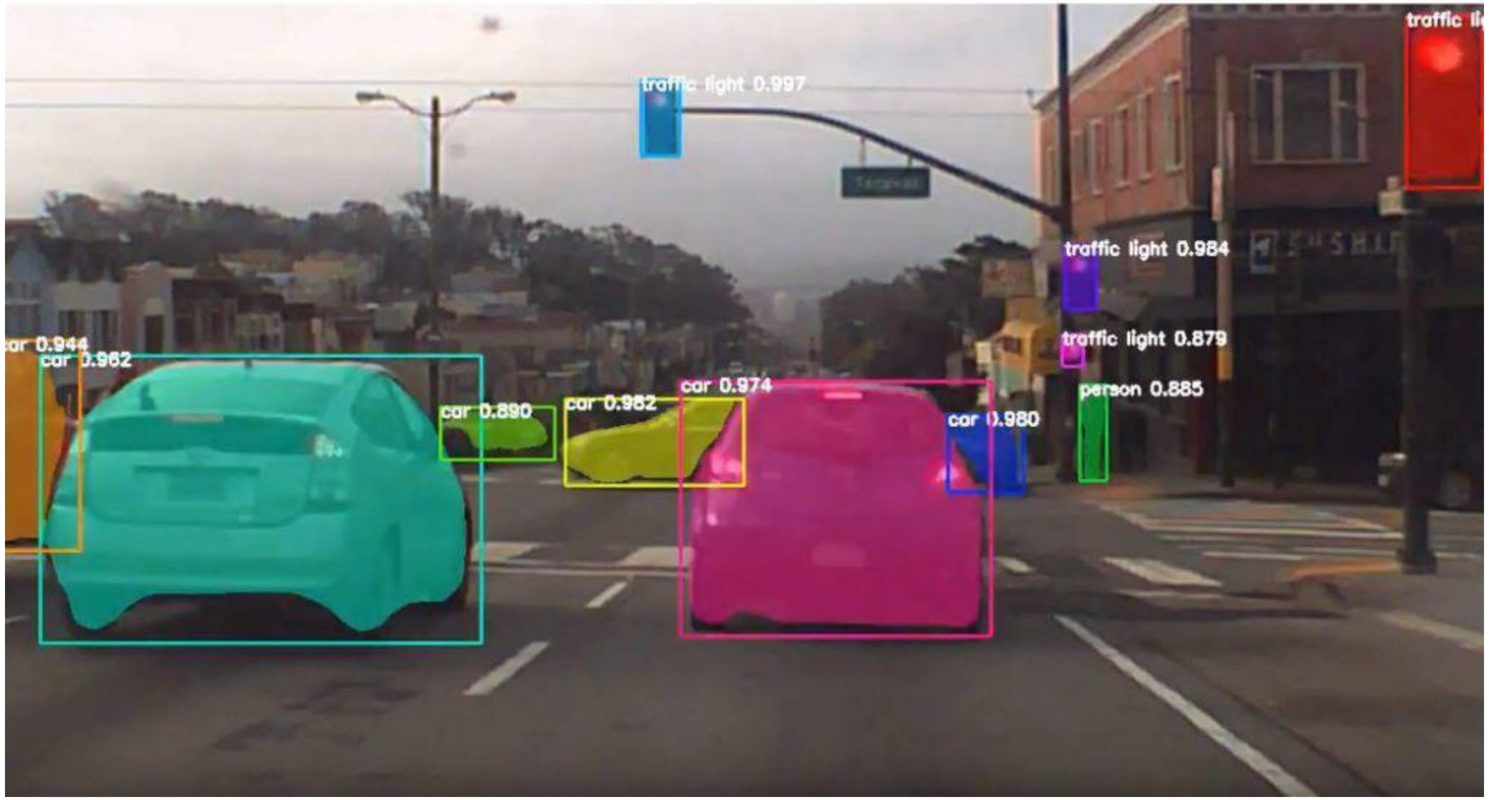


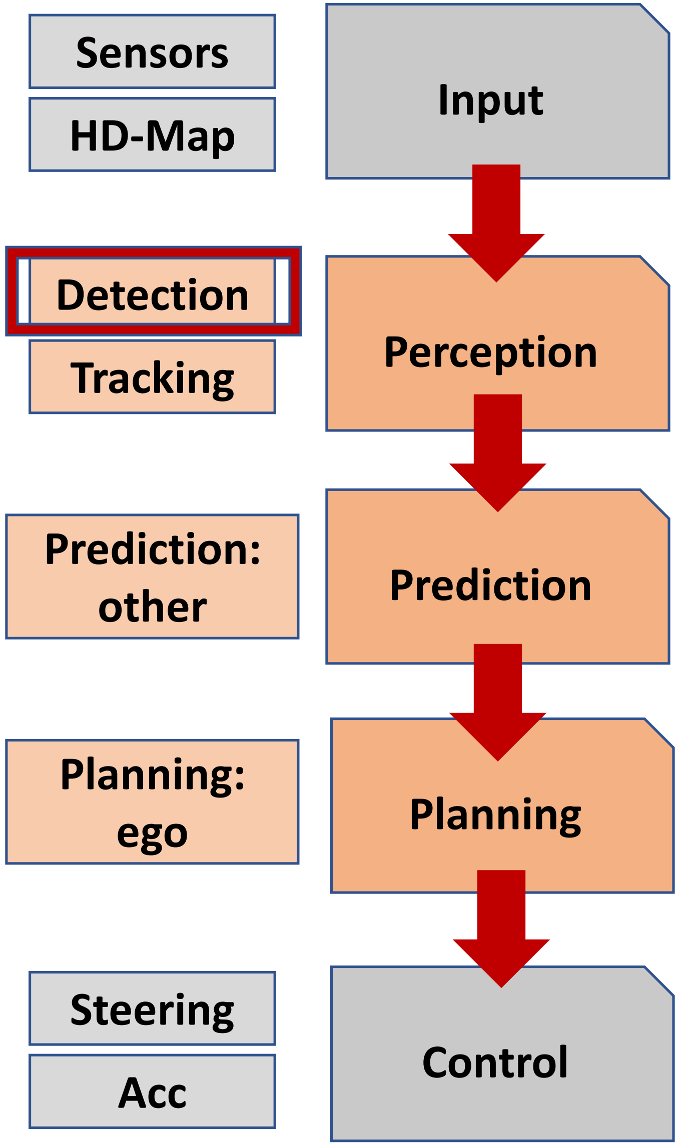


HUAWEI

SDV: Detection

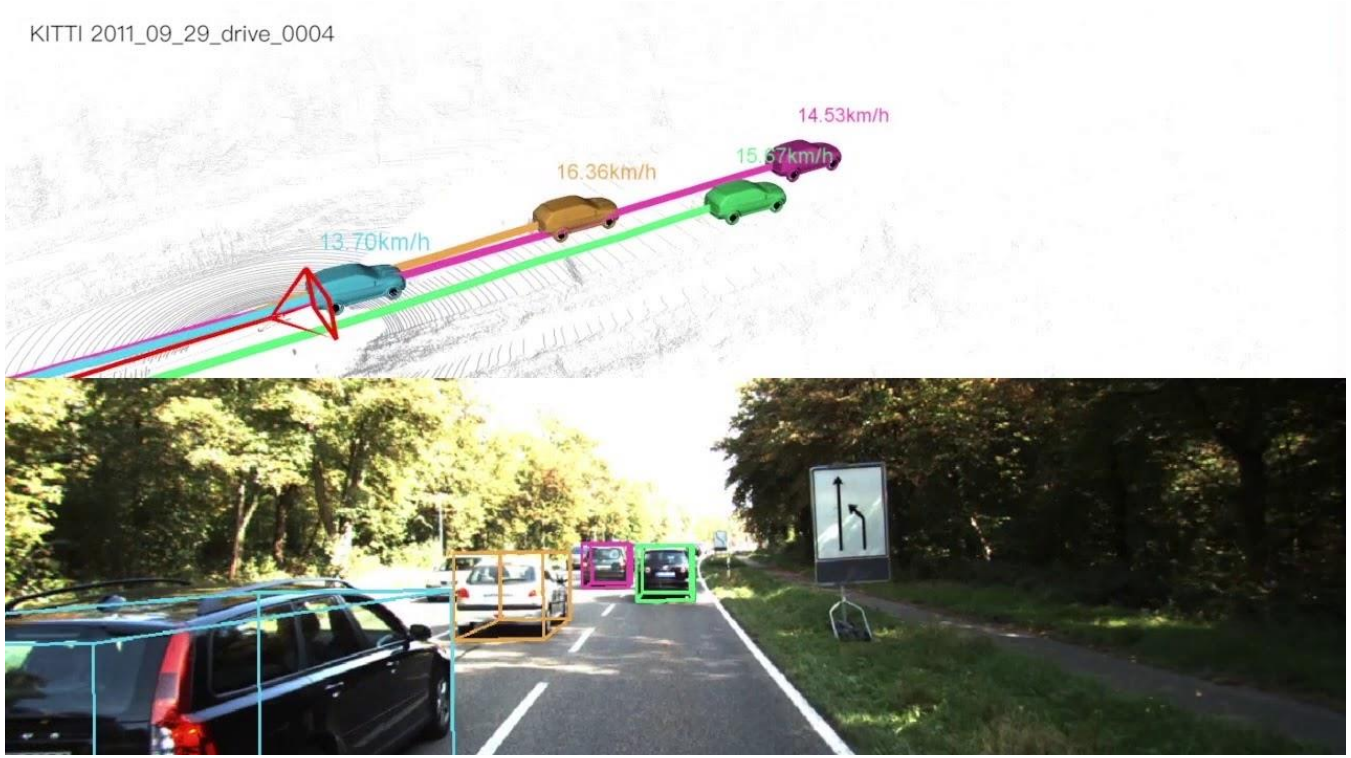
- The *first* step of the Perception part:
  - **Detection** (segmentation, depth-estimation, etc.) of the objects around

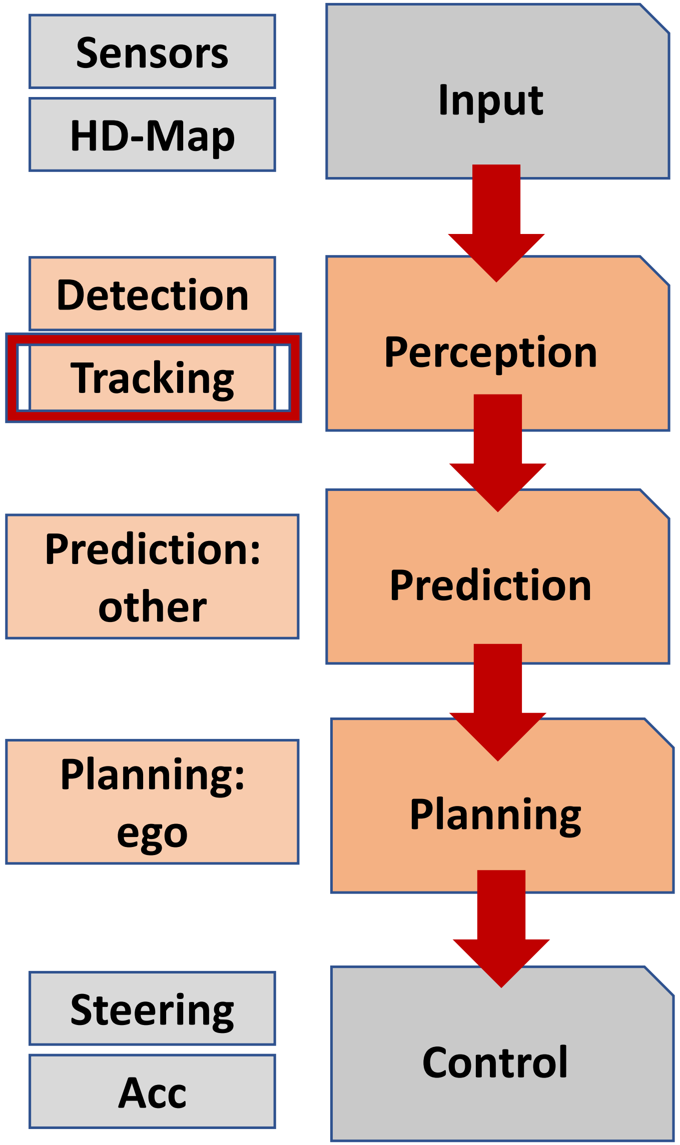




SDV: Tracking

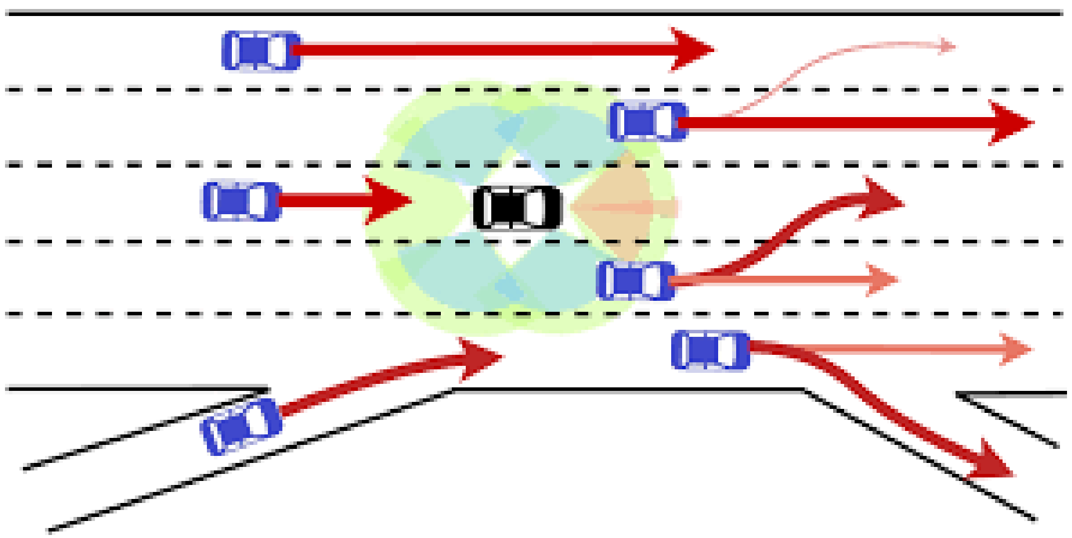
- The *second* step of the Perception part:
  - **Tracking** of the detected objects and estimation of their coordinates for the Prediction part

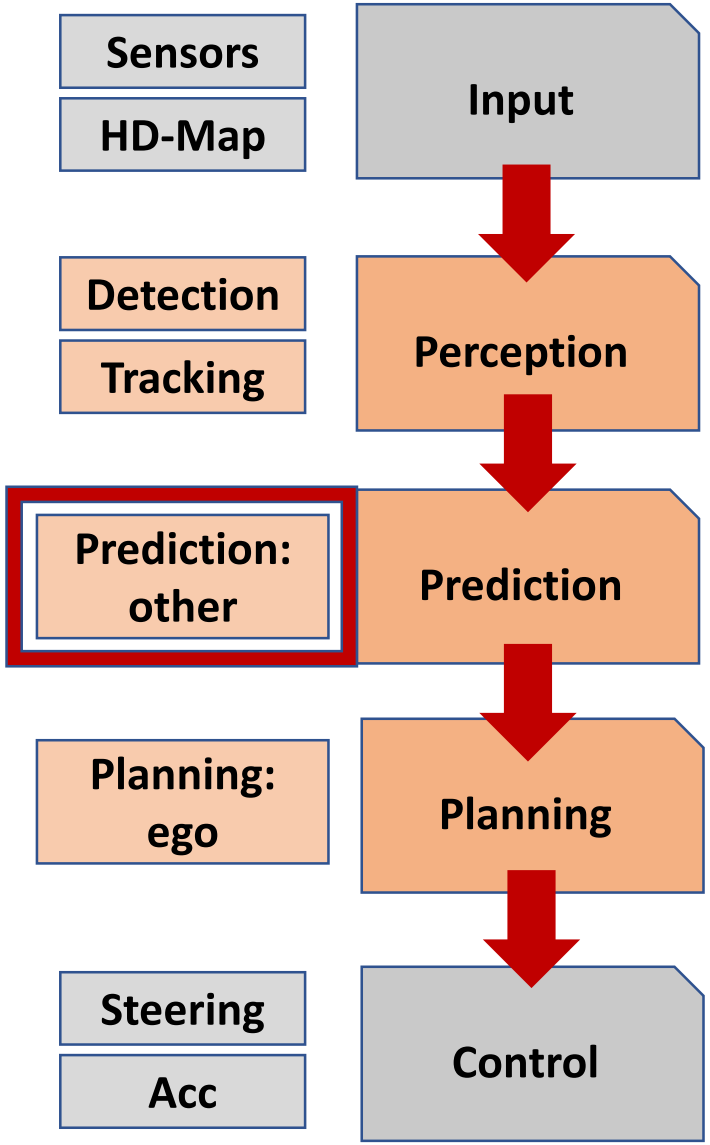




SDV: Prediction

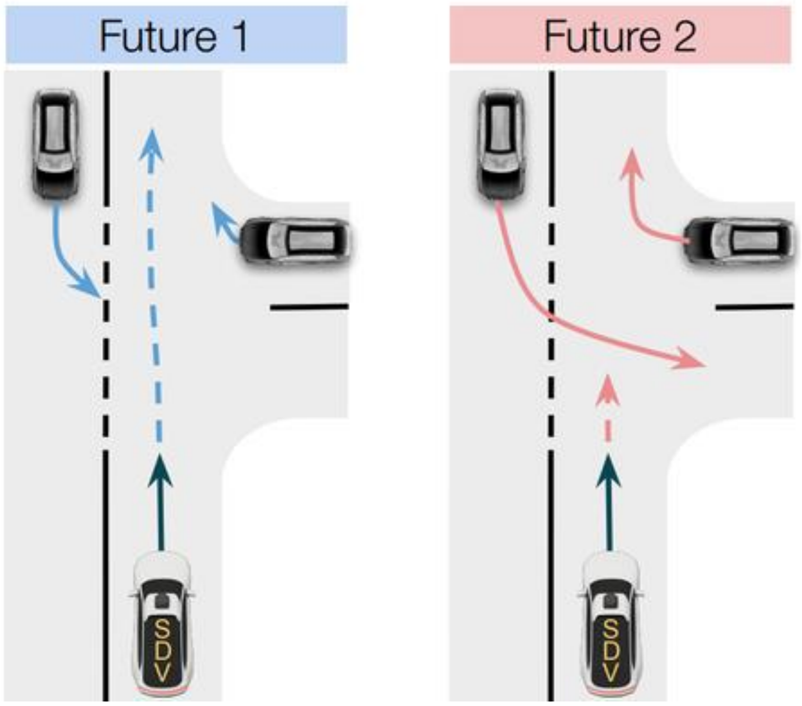
- Future trajectories **prediction** of all surrounding objects based on the *tracking history* and *HD-Map*
  - Usually, 1-10 second

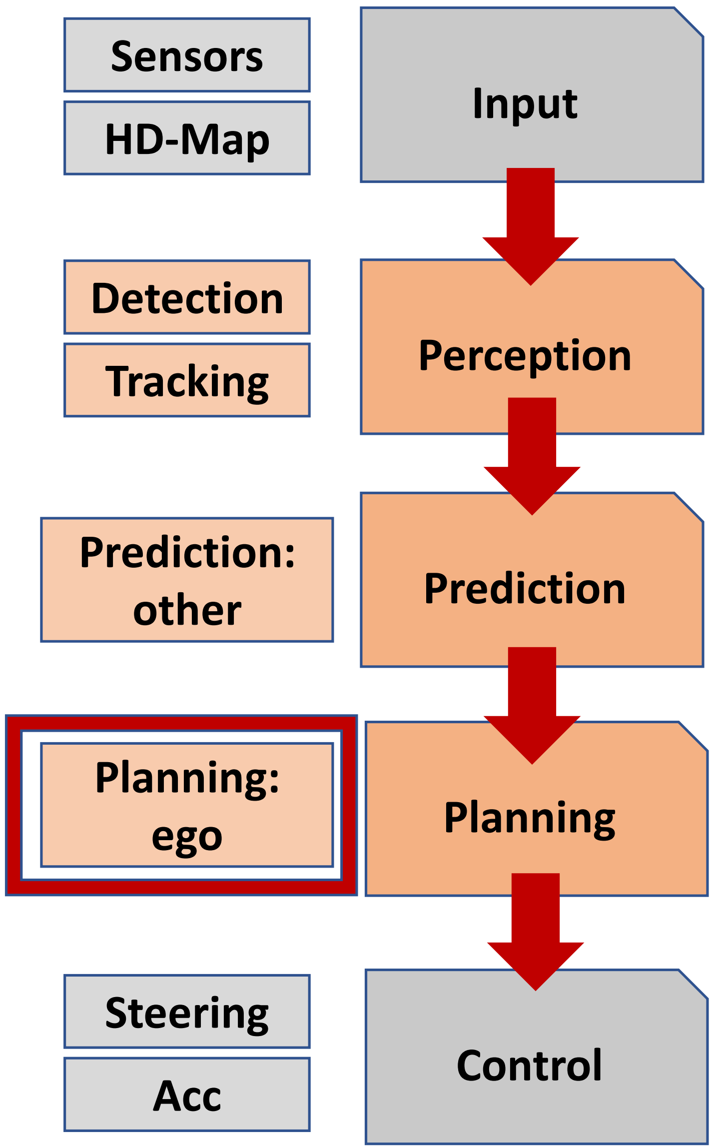




SDV: Planning

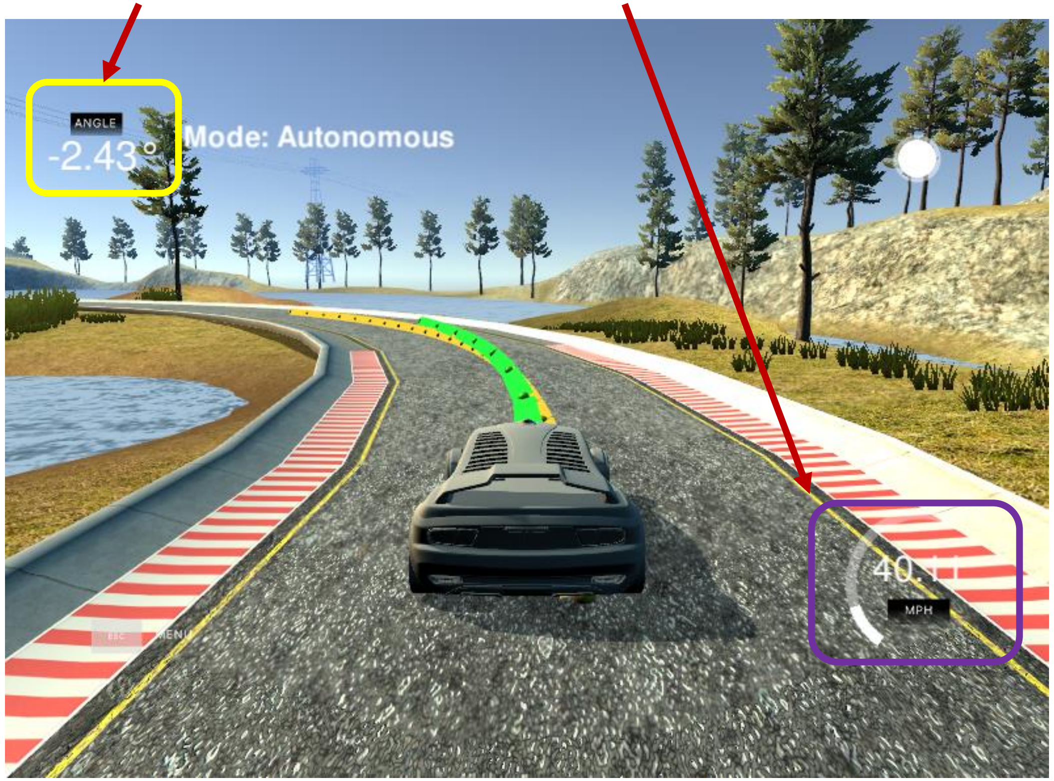
- **Planning** of SDV future actions based on the *predictions* and *HD-Map*

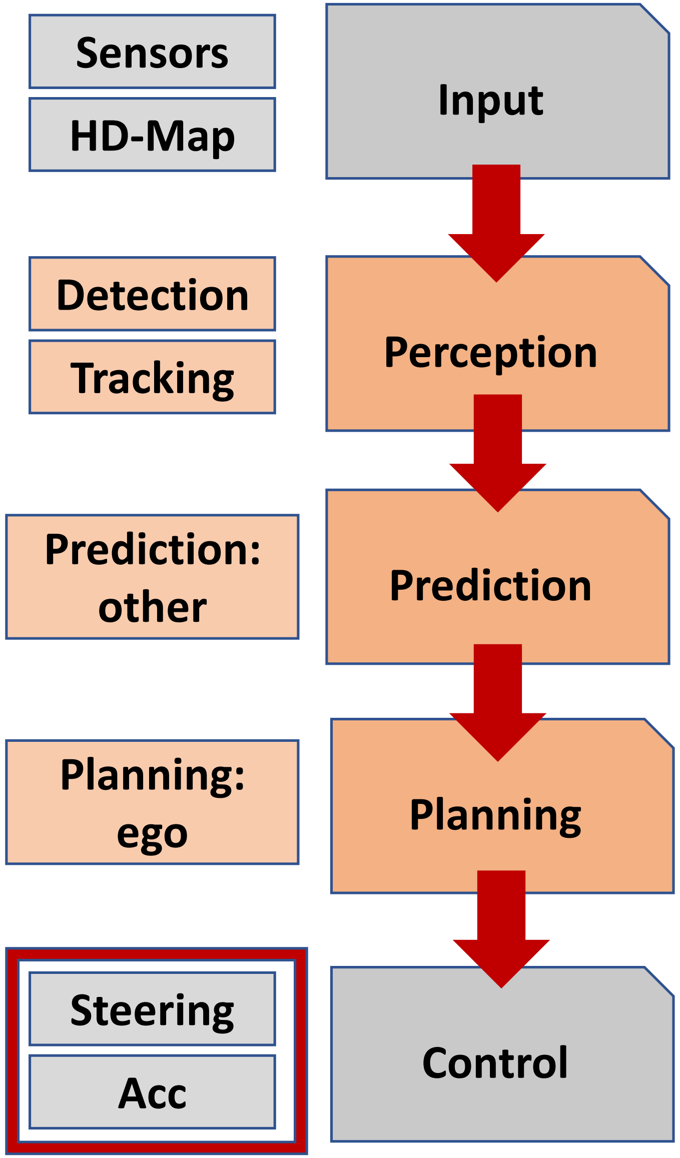




SDV: Control

- Realization and **control** of SDV actions based *motion plan*
  - Steering control, acceleration control, etc.

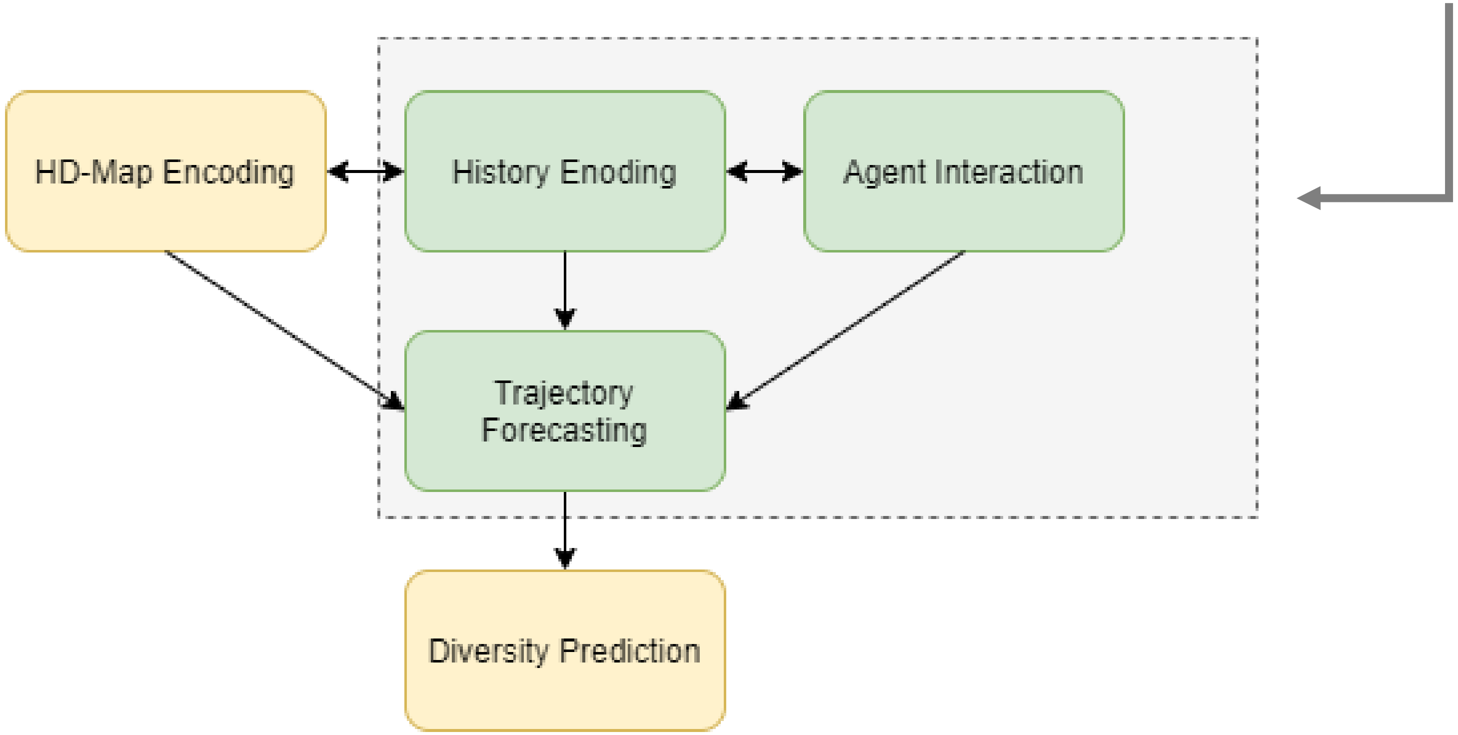


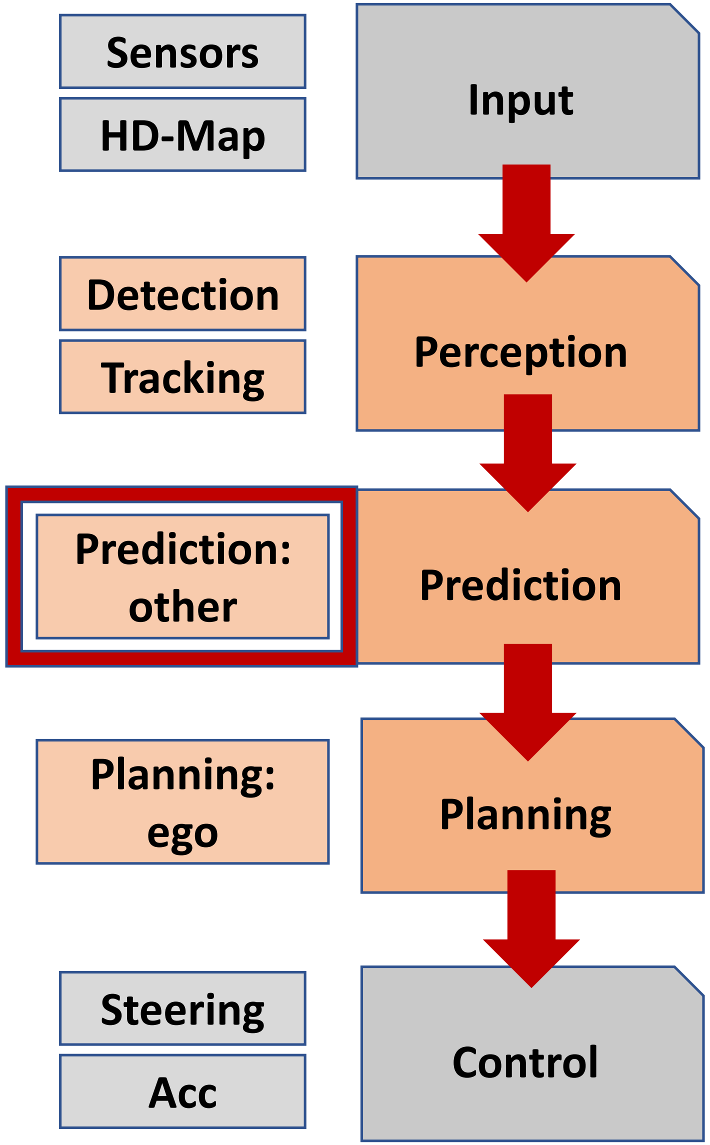


Task formulation

More about Prediction

- Usually, consists of 5 blocks
  - We are going to concentrate on 3 basic blocks





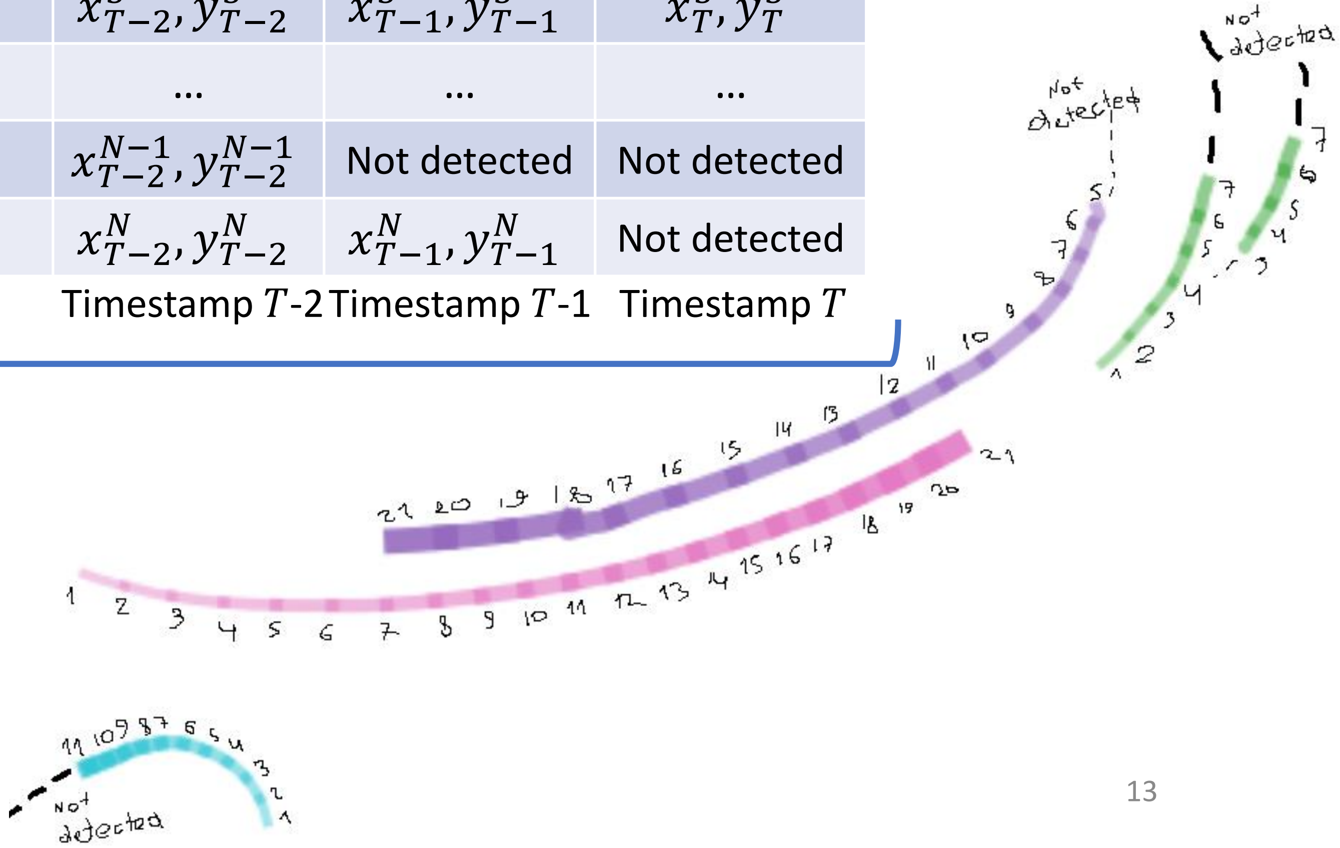
HUAWEI

What we have

- Set of observed coordinates for each agent in the scene for the last T timestamps

	Timestamp 1	Timestamp 2	Timestamp 3		Timestamp T-2	Timestamp T-1	Timestamp T
Agent 1	x_1^1, y_1^1	x_2^1, y_2^1	x_3^1, y_3^1	…	x_{T-2}^1, y_{T-2}^1	x_{T-1}^1, y_{T-1}^1	x_T^1, y_T^1
Agent 2	x_1^2, y_1^2	x_2^2, y_2^2	x_3^2, y_3^2	…	x_{T-2}^2, y_{T-2}^2	x_{T-1}^2, y_{T-1}^2	x_T^2, y_T^2
Agent 3	Not detected	Not detected	x_3^3, y_3^3	…	x_{T-2}^3, y_{T-2}^3	x_{T-1}^3, y_{T-1}^3	x_T^3, y_T^3
	…	…	…	…	…	…	…
Agent N-1	x_1^{N-1}, y_1^{N-1}	x_2^{N-1}, y_2^{N-1}	x_3^{N-1}, y_3^{N-1}	…	$x_{T-2}^{N-1}, y_{T-2}^{N-1}$	Not detected	Not detected
Agent N	Not detected	Not detected	x_3^N, y_3^N	…	x_{T-2}^N, y_{T-2}^N	x_{T-1}^N, y_{T-1}^N	Not detected

Available history of agents positions



HUAWEI

What we need

- We need to predict the next P positions for the predefined set of agents

x_1^1, y_1^1	x_2^1, y_2^1	x_3^1, y_3^1	…	x_{T-2}^1, y_{T-2}^1	x_{T-1}^1, y_{T-1}^1	x_T^1, y_T^1	$\hat{x}_{T+1}^1, \hat{y}_{T+1}^1$	…	$\hat{x}_{T+P}^1, \hat{y}_{T+P}^1$
x_1^2, y_1^2	x_2^2, y_2^2	x_3^2, y_3^2	…	x_{T-2}^2, y_{T-2}^2	x_{T-1}^2, y_{T-1}^2	x_T^2, y_T^2	$\hat{x}_{T+1}^2, \hat{y}_{T+1}^2$	…	$\hat{x}_{T+P}^2, \hat{y}_{T+P}^2$
Not detected	Not detected	x_3^3, y_3^3	…	x_{T-2}^3, y_{T-2}^3	x_{T-1}^3, y_{T-1}^3	x_T^3, y_T^3	$\hat{x}_{T+1}^3, \hat{y}_{T+1}^3$	…	$\hat{x}_{T+P}^3, \hat{y}_{T+P}^3$
…	…	…	…	…	…	…	…	…	…
x_1^{N-1}, y_1^{N-1}	x_2^{N-1}, y_2^{N-1}	x_3^{N-1}, y_3^{N-1}	…	$x_{T-2}^{N-1}, y_{T-2}^{N-1}$	Not detected	Not detected	Uninteresting	…	Uninteresting
Not detected	Not detected	x_3^N, y_3^N	…	x_{T-2}^N, y_{T-2}^N	x_{T-1}^N, y_{T-1}^N	Not detected	Uninteresting	…	Uninteresting

Available history of agents positions

Positions that are needed to be predicted

How to measure

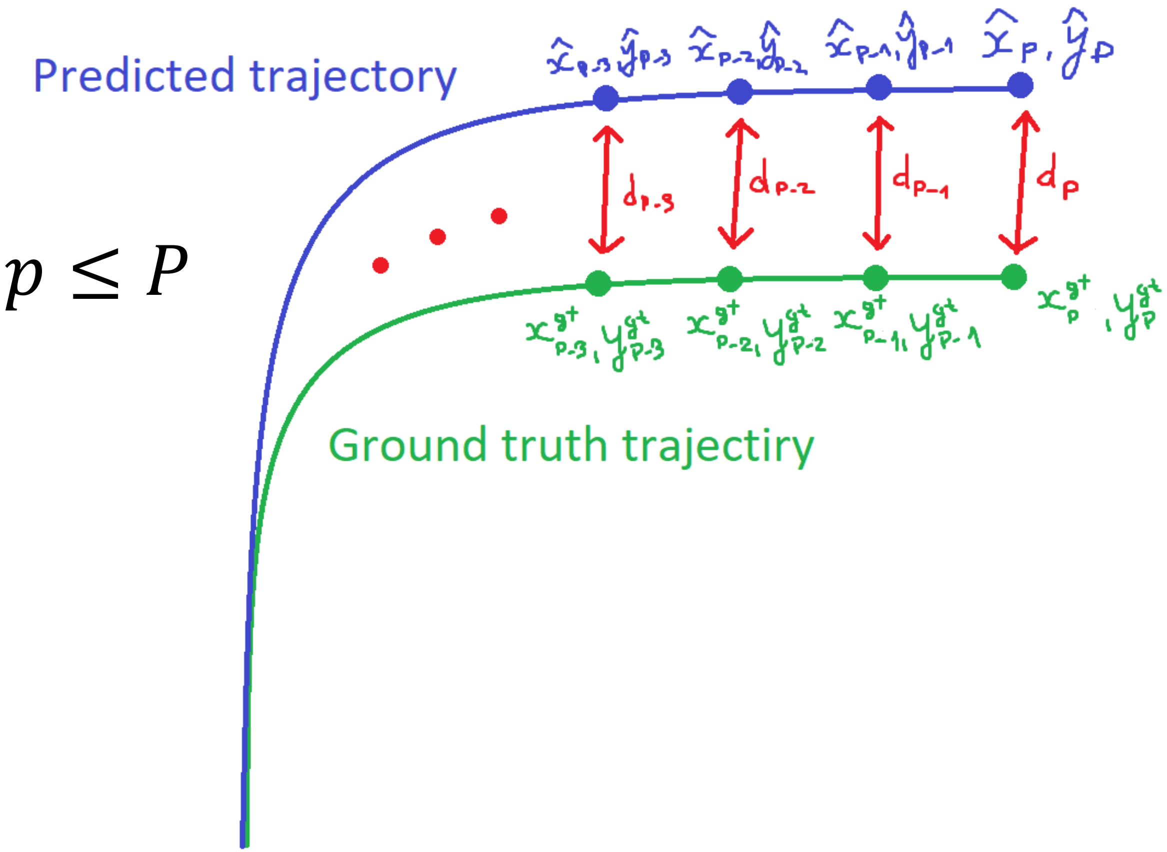
$$d_p = \sqrt{(x_p - \hat{x}_p)^2 + (y_p - \hat{y}_p)^2}, 1 \leq p \leq P$$

- **Average Displacement Error (ADE):**

$$ADE = \frac{d_1 + d_2 + \cdots + d_{P-1} + d_P}{P}$$

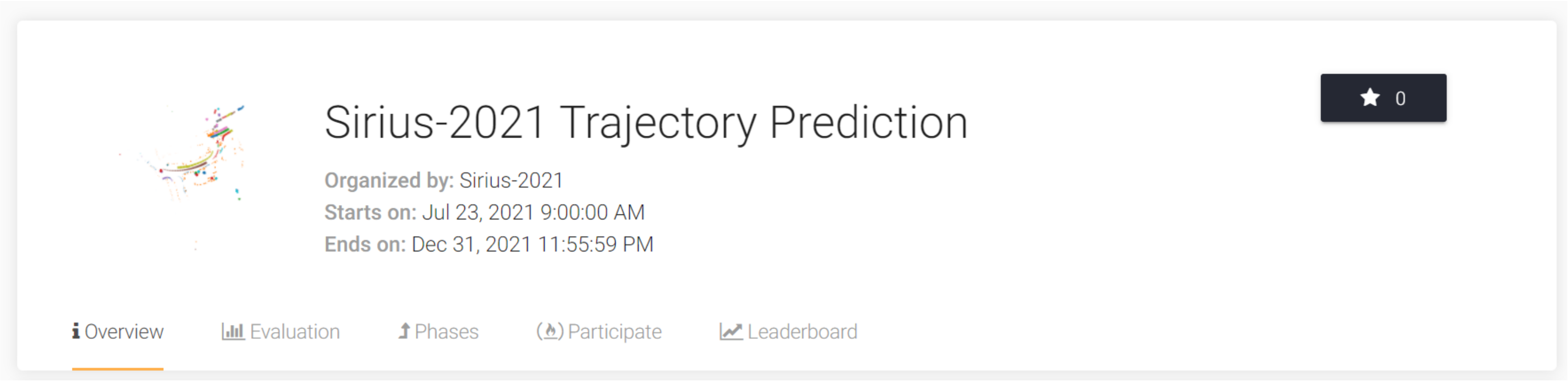
- **Final Displacement Error (FDE):**

$$FDE = d_P$$



How it will be

- Automatic system to evaluate your solutions on the subset of the test set
    - https://eval.ai/web/challenges/challenge-page/1194/phases
- The final results will be ranked based on the full test results



Task benefits

- Data is simple enough to process on an ordinary laptop
  - The whole dataset takes less than 7 Gbyte
  - GPU is helpful but not necessary
- Despite the small scale of data, the task formulation is close to the real product tasks
  - Findings for this challenge may be beneficial for the state-of-the-art solutions
- Trajectory Prediction solution consists of several blocks
  - Scene preprocessing, history encoder, agents interaction, trajectory forecaster, loss function, etc.
  - So, there are a wealth of ways to improve baseline solutions

Contacts

- Stepan Komkov
  - Until the July 31st
- Alexander Petyushko
  - From the July 29th
- Chat of the challenge
  - https://t.me/joinchat/Km7p_wYqHcU3MGFi
- News of the challenge
  - https://t.me/joinchat/GewH2zFxBpYyZGRi