Artificial Intelligence in Autonomous Driving
自动驾驶中的人工智能

李力耘

JDX Silicon Valley

京东硅谷X研发中心

Artificial Intelligence

主办

O'REILLY intel AI

Topic

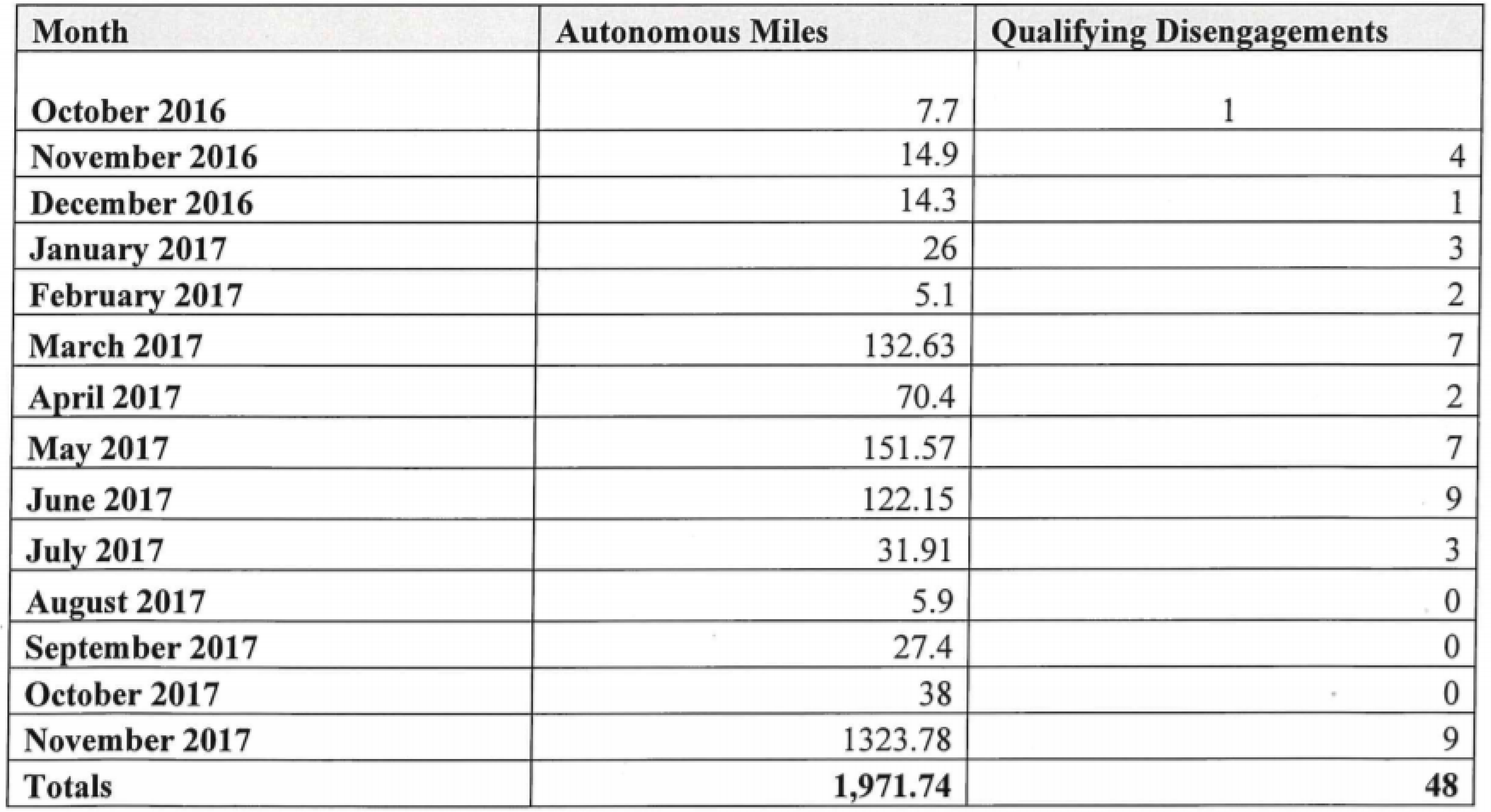
- **Autonomous Driving Software System Modules**

- **Deep Learning in Autonomous Driving System Modules**
  - **Perception**
  - **Prediction**
  - **Decision & Planning**

- **Future Directions for AI in Autonomous Driving**

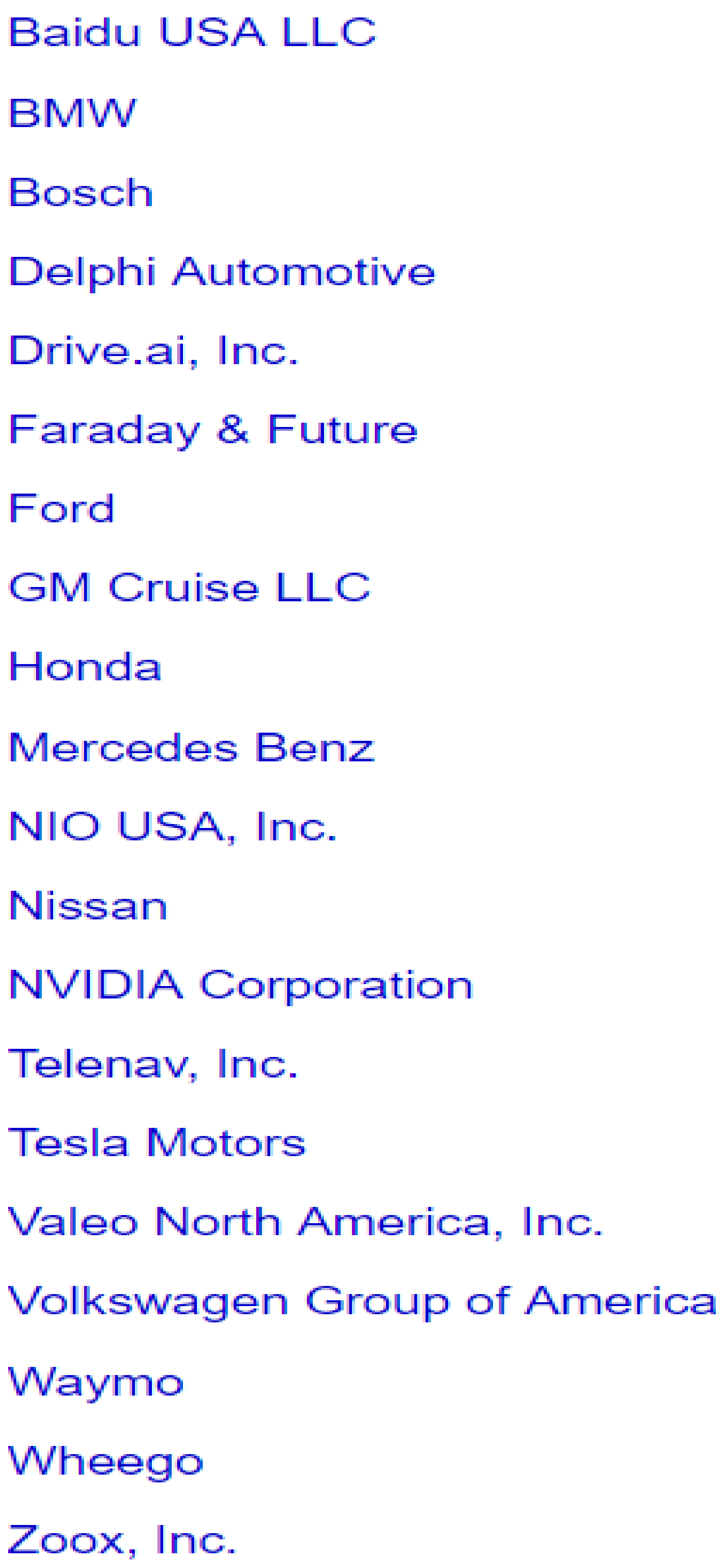
Apollo

- An open source platform for autonomous driving
- 2017 California DMV Autonomous Driving Report
- Current MPI is at around 147 Miles/Per Interruption



Month	Autonomous Miles	Qualifying Disengagements
October 2016	7.7	1
November 2016	14.9	4
December 2016	14.3	1
January 2017	26	3
February 2017	5.1	2
March 2017	132.63	7
April 2017	70.4	2
May 2017	151.57	7
June 2017	122.15	9
July 2017	31.91	3
August 2017	5.9	0
September 2017	27.4	0
October 2017	38	0
November 2017	1323.78	9
Totals	**1,971.74**	**48**

Baidu's 2017 DMV Report



Baidu USA LLC
BMW
Bosch
Delphi Automotive
Drive.ai, Inc.
Faraday & Future
Ford
GM Cruise LLC
Honda
Mercedes Benz
NIO USA, Inc.
Nissan
NVIDIA Corporation
Telenav, Inc.
Tesla Motors
Valeo North America, Inc.
Volkswagen Group of America
Waymo
Wheego
Zoox, Inc.

Companies Testing Autonomous Driving in California

自动驾驶软件系统

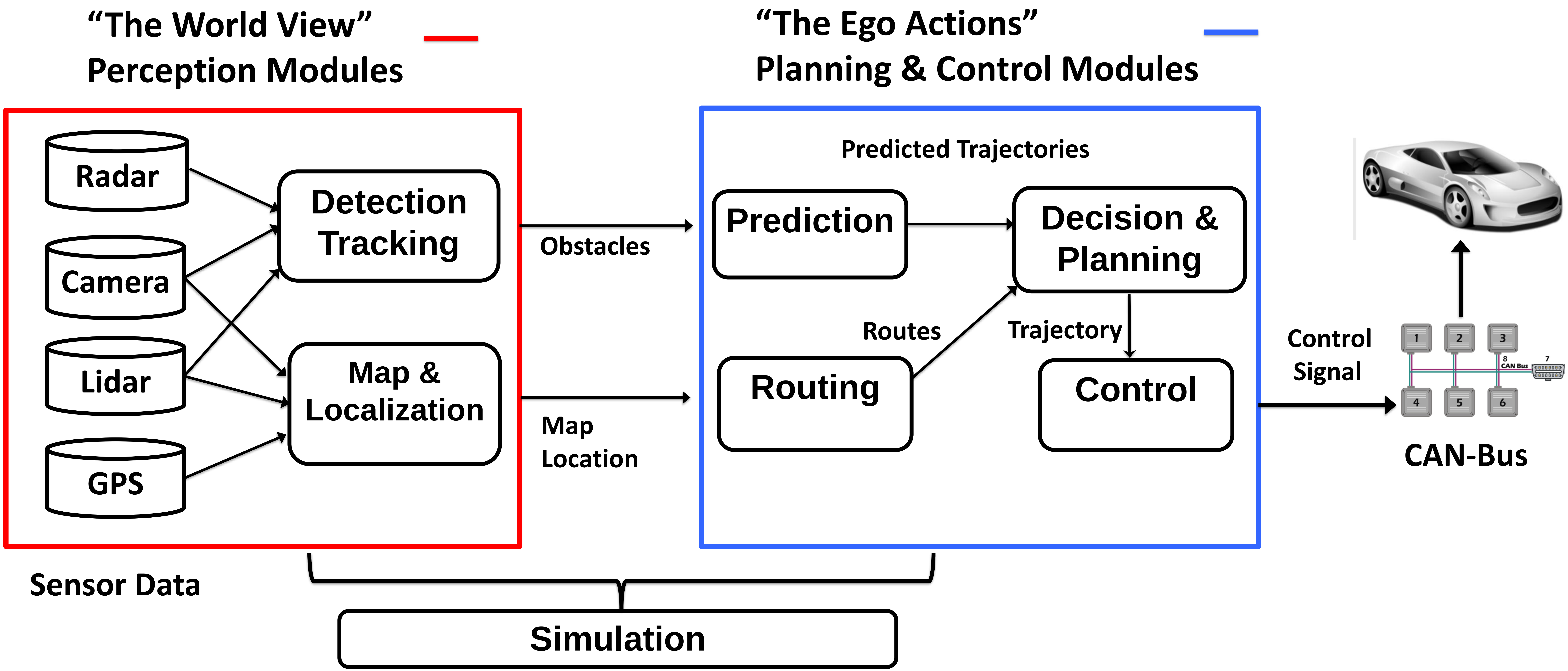
- 客观世界 **The “World”: Perceive the objective environment**
  - 感知外界的所有一切客观存在
  - 包括了地图**(HD-Map)**，定位**(Localization)**，物体检测**(Object Detection)**，物体跟踪**(Object Tracking)**

- 主观行动 **The “Ego”: How the autonomous vehicle behaves**
  - 自动驾驶车辆“自我”的主观世界和行动
  - 包括：路由导航**(Routing Navigation),** 决策**(Decision),** 规划**(Planning),** 以及控制**(Control)**

- 模拟器 **Simulation : An Integration Platform where all the above modules are tested**

- 线控接口 **Can-Bus: Provide an accurate drive-by-wire interface**
  - 提供**drive-by-wire**的接口
  - 保证对车辆精确控制的可行性

A Typical Framework of Modules in Autonomous Driving System



"The World View"
Perception Modules

"The Ego Actions"
Planning & Control Modules

Radar

Camera

Lidar

GPS

Detection Tracking

Map & Localization

Obstacles

Map
Location

Predicted Trajectories

Prediction

Decision & Planning

Routes

Trajectory

Routing

Control

Control
Signal

CAN-Bus

Sensor Data

Simulation

Topic

- Autonomous Driving Software System Modules

- Deep Learning in Autonomous Driving System Modules
  - Perception
  - Prediction
  - Decision & Planning

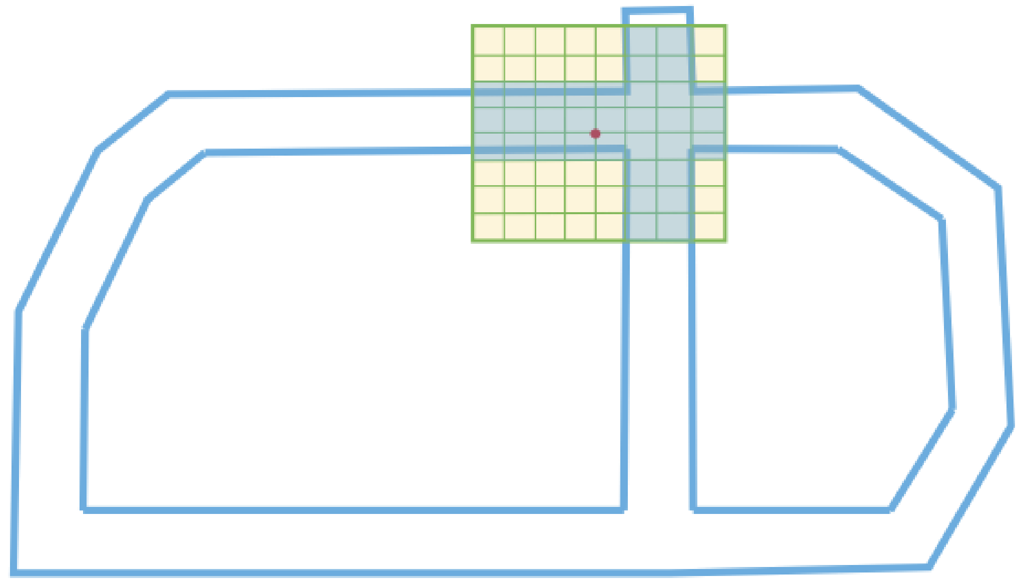
- Future Directions for AI in Autonomous Driving

Apollo系统中的物体检测

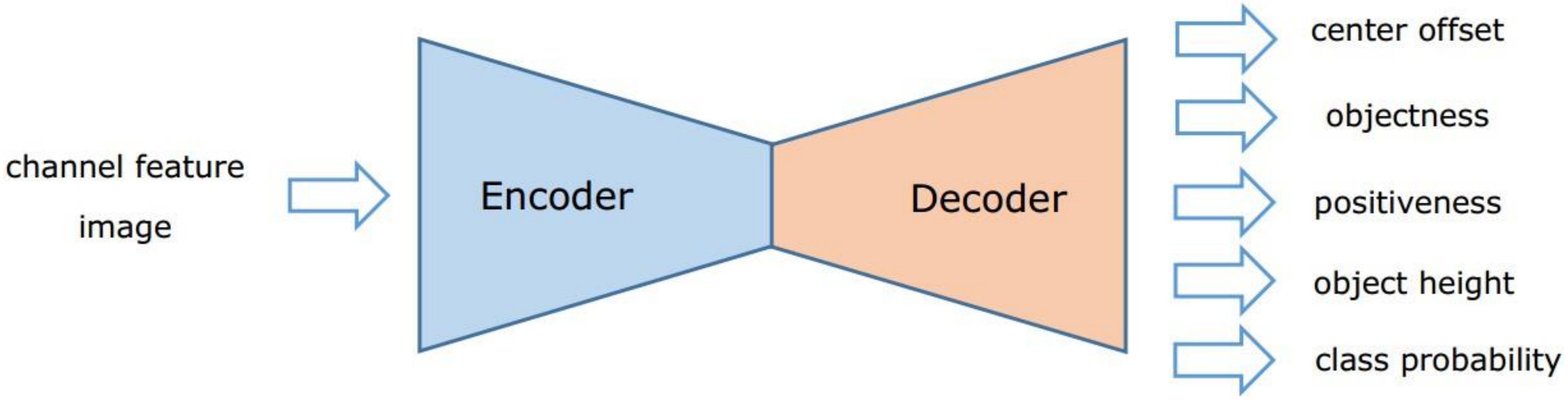
- Region of Interest (ROI)区域的构建
- Bird Eye View (BEV) “将四维(X,Y,Z,I)的点云数据投影到二维的XY平面上
- 基于卷积神经网络CNN的物体检测
  - Network: FCNN
  - Features: 8 Channel Features
  - Output : 5 Attributes
- 物体的再次聚类Obstacle Clustering
- 构建3D Bounding Box和物体跟踪 (Tracking)

优点：实现简单，物体识别的准确率高

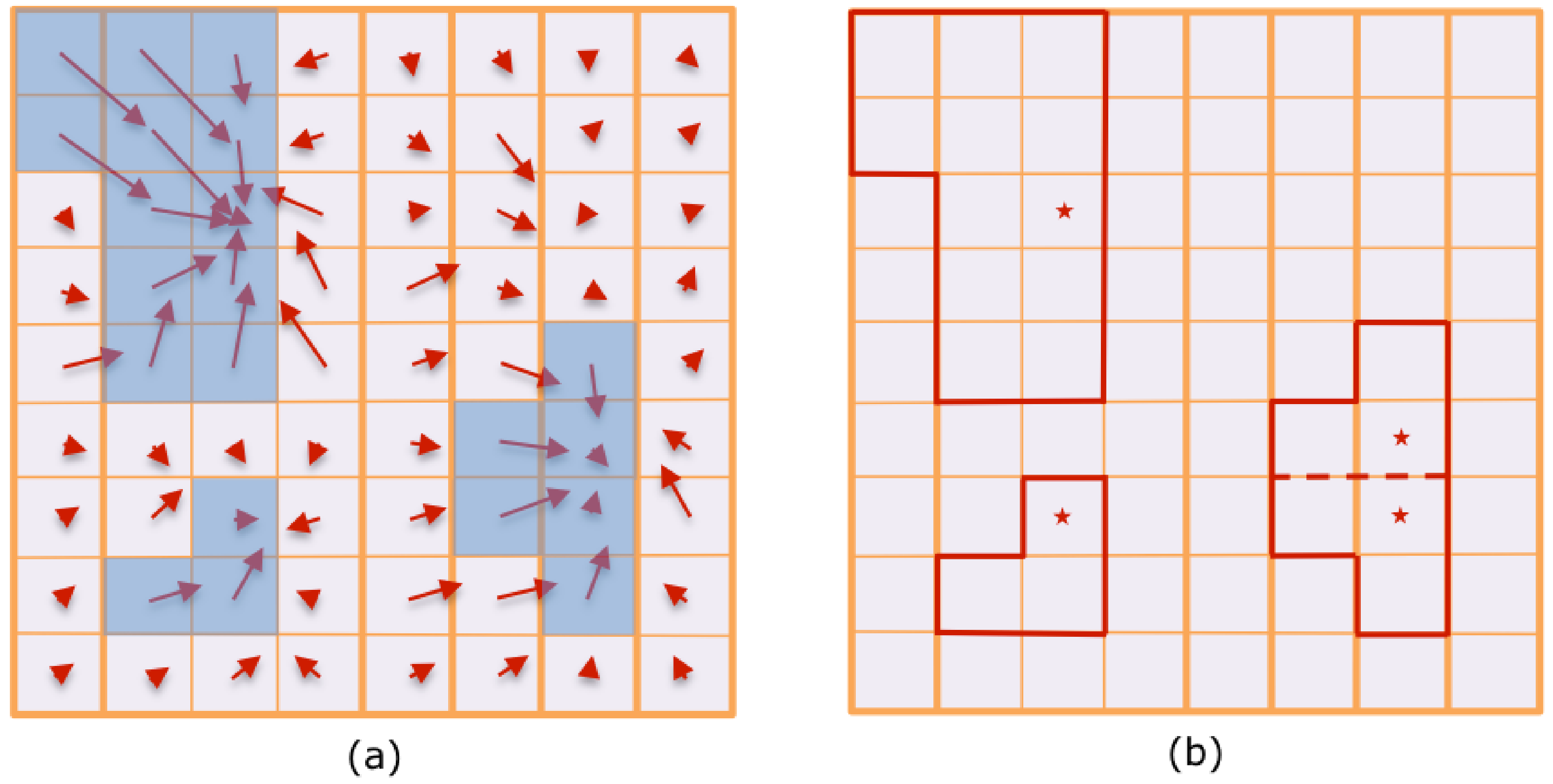
不足：不稳定的物体分割比如物体聚合分裂） 以及对ROI区域的依赖



ROI

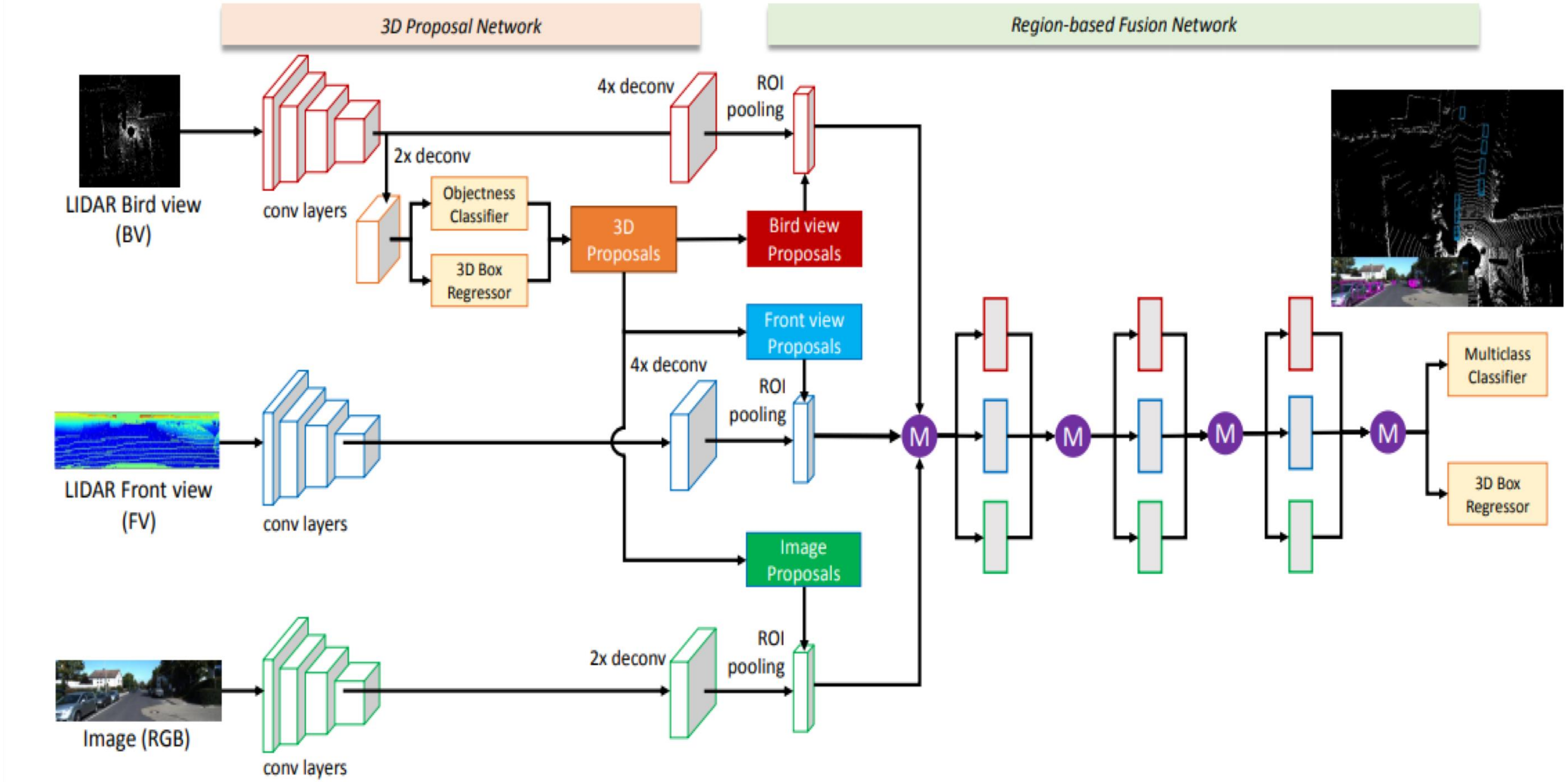


FCNN



Clustering

Obstacle Perception : Fusing Multi-View Lidar and Camera Input



Multi-View 3D Object Detection Network for Autonomous Driving
X. Chen, H, Ma, J. Wan, B. Li, T. X 2017

- 准确率高
- 融合图像Proposal后的分割结果更加稳定
  Stable segmentation with camera based fusion
- 对计算资源要求高
  0.36s inference time on a GPU engine

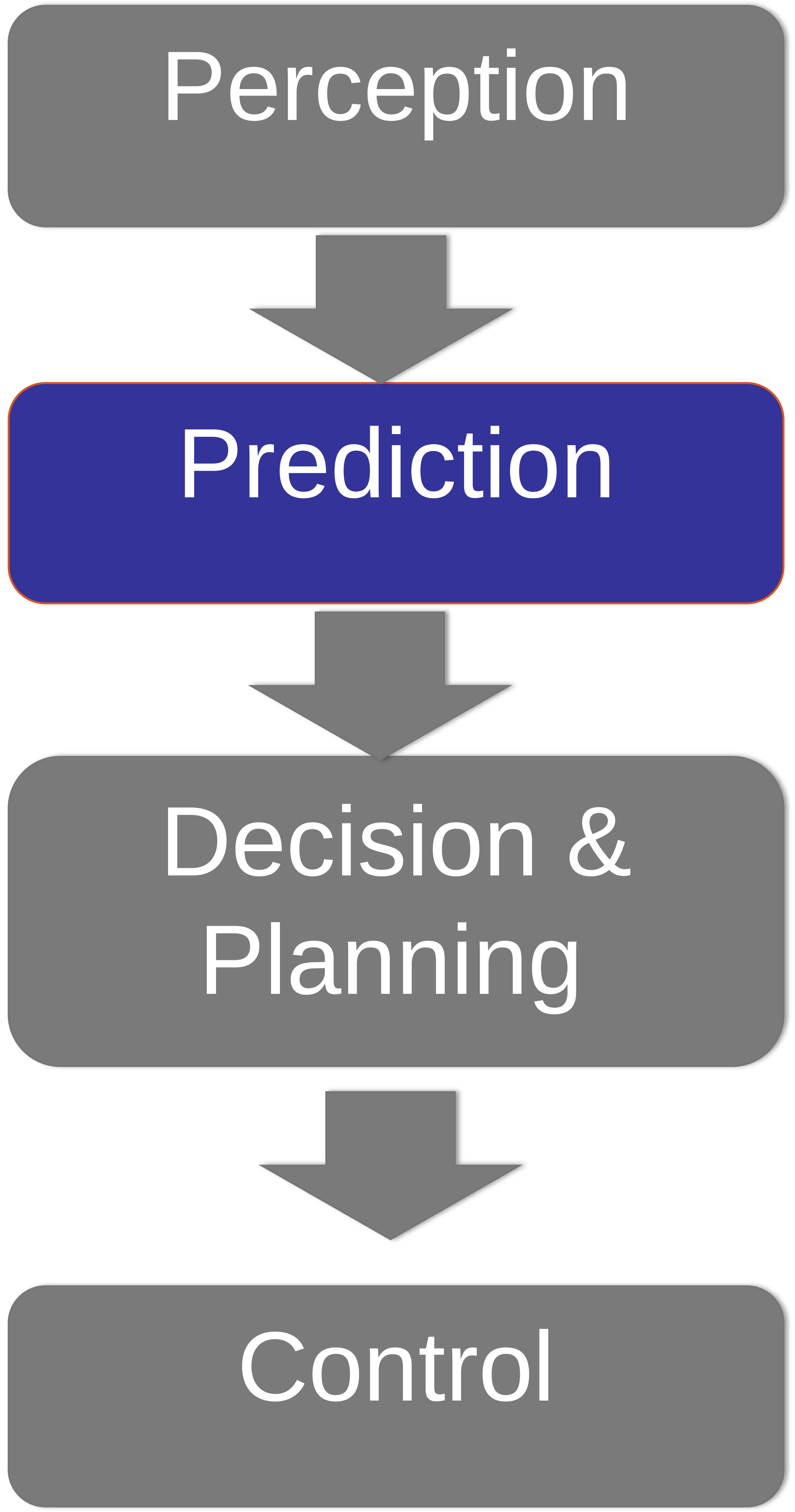
Topic

- Autonomous Driving Software System Modules

- Deep Learning in Autonomous Driving System Modules
  - Perception
  - Prediction
  - Decision & Planning

- Future Directions for AI in Autonomous Driving

交通预测在无人驾驶中的作用


Perception

Prediction

Decision & Planning

Control

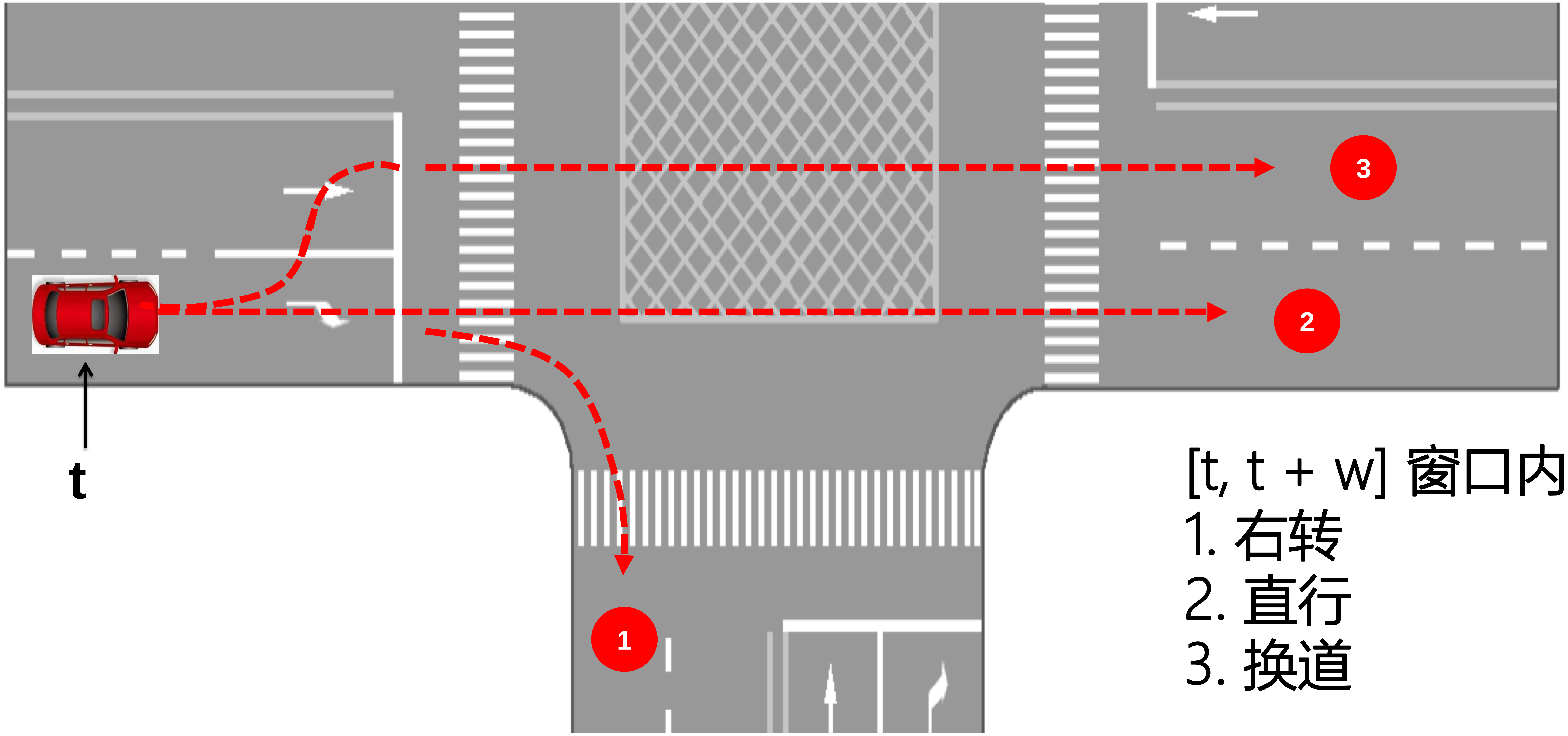
- 车上的最小可行模块(Minimum Viable Product Modules)之一，连接感知和规划的桥梁

- 预测轨迹的精确性直接影响下游决策规划的轨迹形状和速度规划；没有预测的轨迹，无人驾驶车辆就无法安全平稳的运行

- 简单的基于跟踪历史或者运动学模型的线性外推无法满足Planning的需要

核心问题：如何有效对预测车辆未来运行轨迹这一问题进行建模?

问题：如何建立准确高效的模型



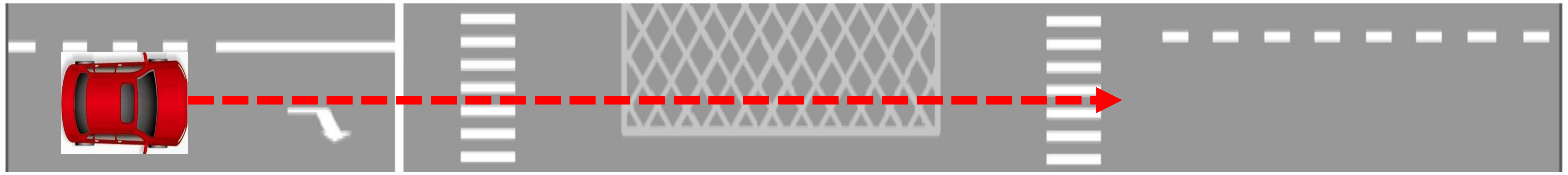
[t, t + w] 窗口内的目标车辆可能:
1. 右转
2. 直行
3. 换道

- 如何对“预测不同可能的时空轨迹”

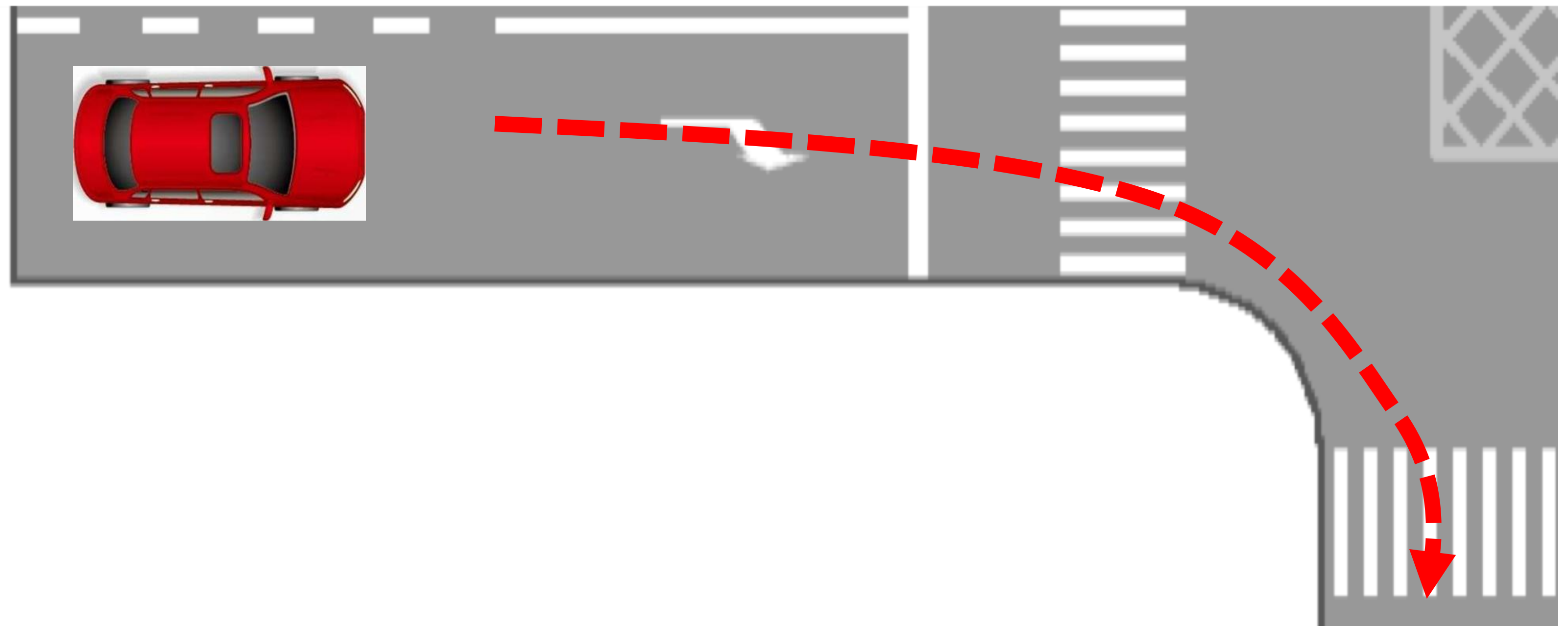
这样一个复杂问题建立统一有效的模型？

考虑：针对不同的行为建模

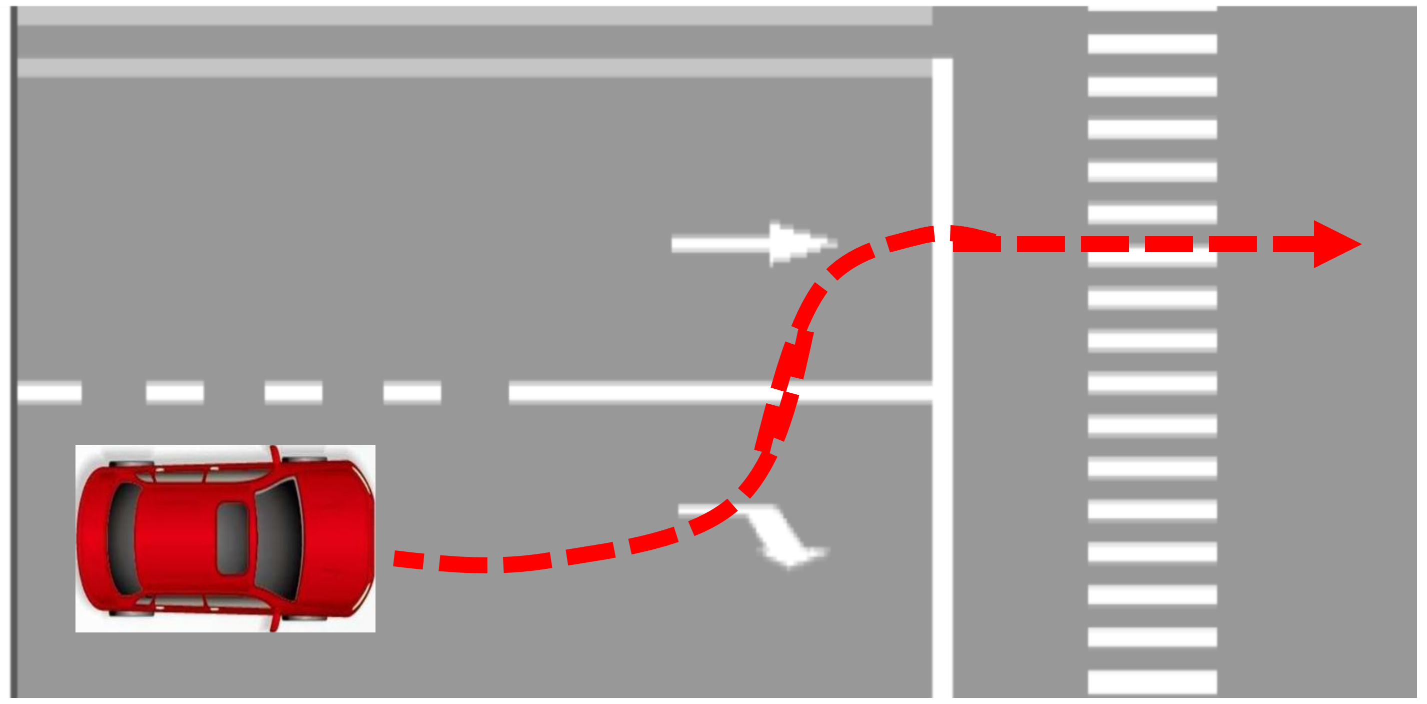
- 直行模型



- (右)拐弯模型

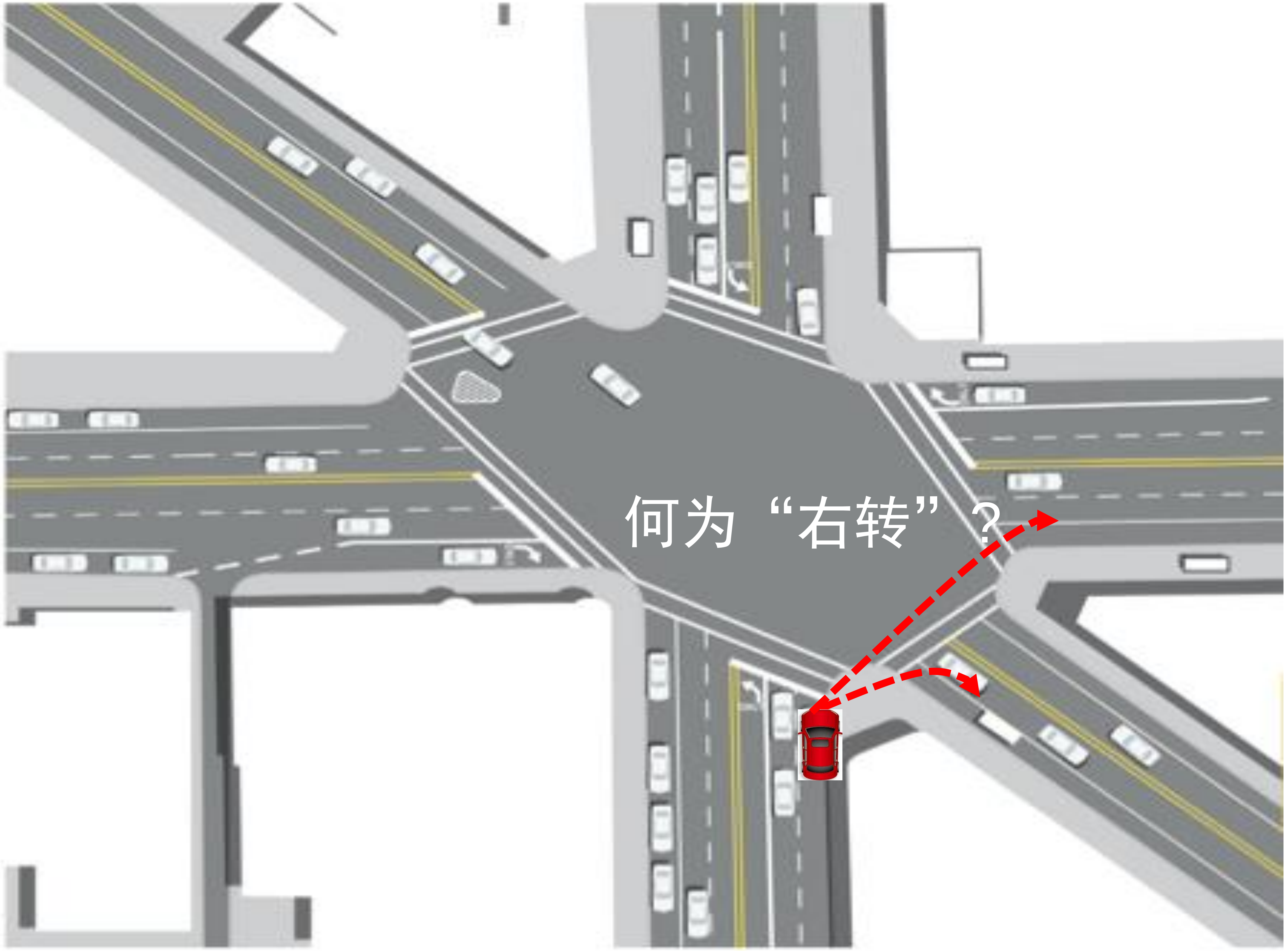


- 换道模型



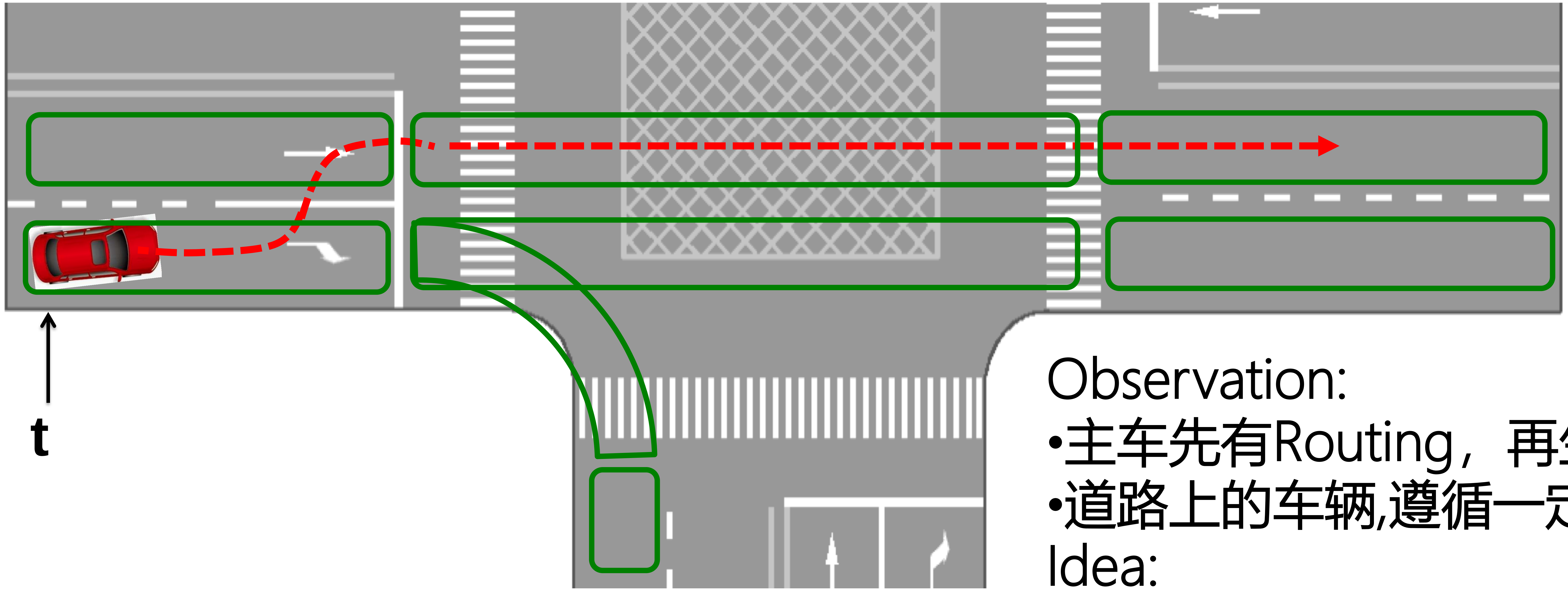
This does NOT work!

分别建模的问题



- 判断适用模型的逻辑会复杂模糊
- 数据分散
- 多个模型同时使用，调参和训练效率低

从主车的动作规划来启发的建模



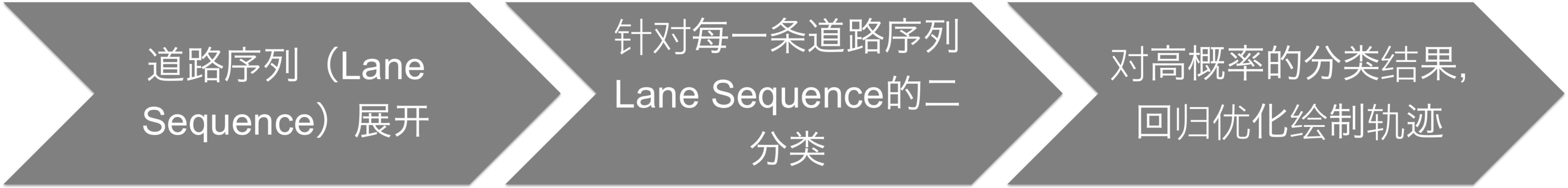
Observation:

•主车先有Routing，再生成轨迹

•道路上的车辆,遵循一定的道路逻辑来进行行驶

Idea:

猜测目标车的“Routing”



道路序列（Lane Sequence）展开

针对每一条道路序列Lane Sequence的二分类

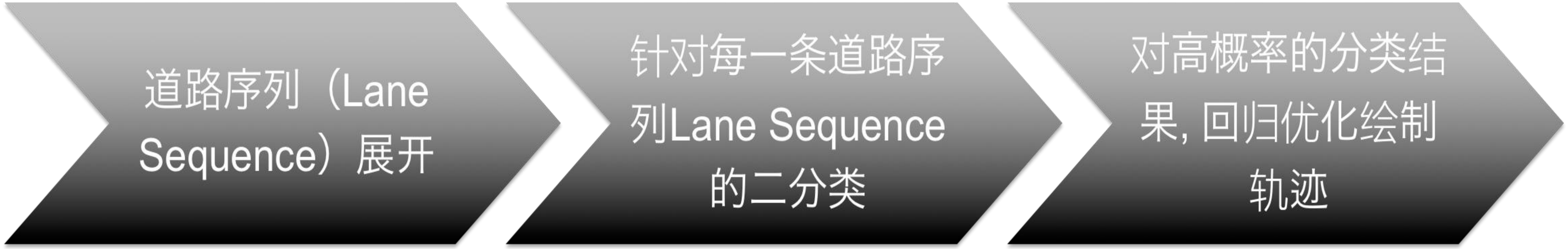
对高概率的分类结果,回归优化绘制轨迹

Prediction as Classification + Regression

- 预测时空轨迹的问题分解成：

（1）二分类问题: 对于每一条可能的道路序列，无人车会不会按照这个道路序列行驶

（2）回归优化问题: 给定一个分类选中的道路序列, 优化轨迹形状和速度特



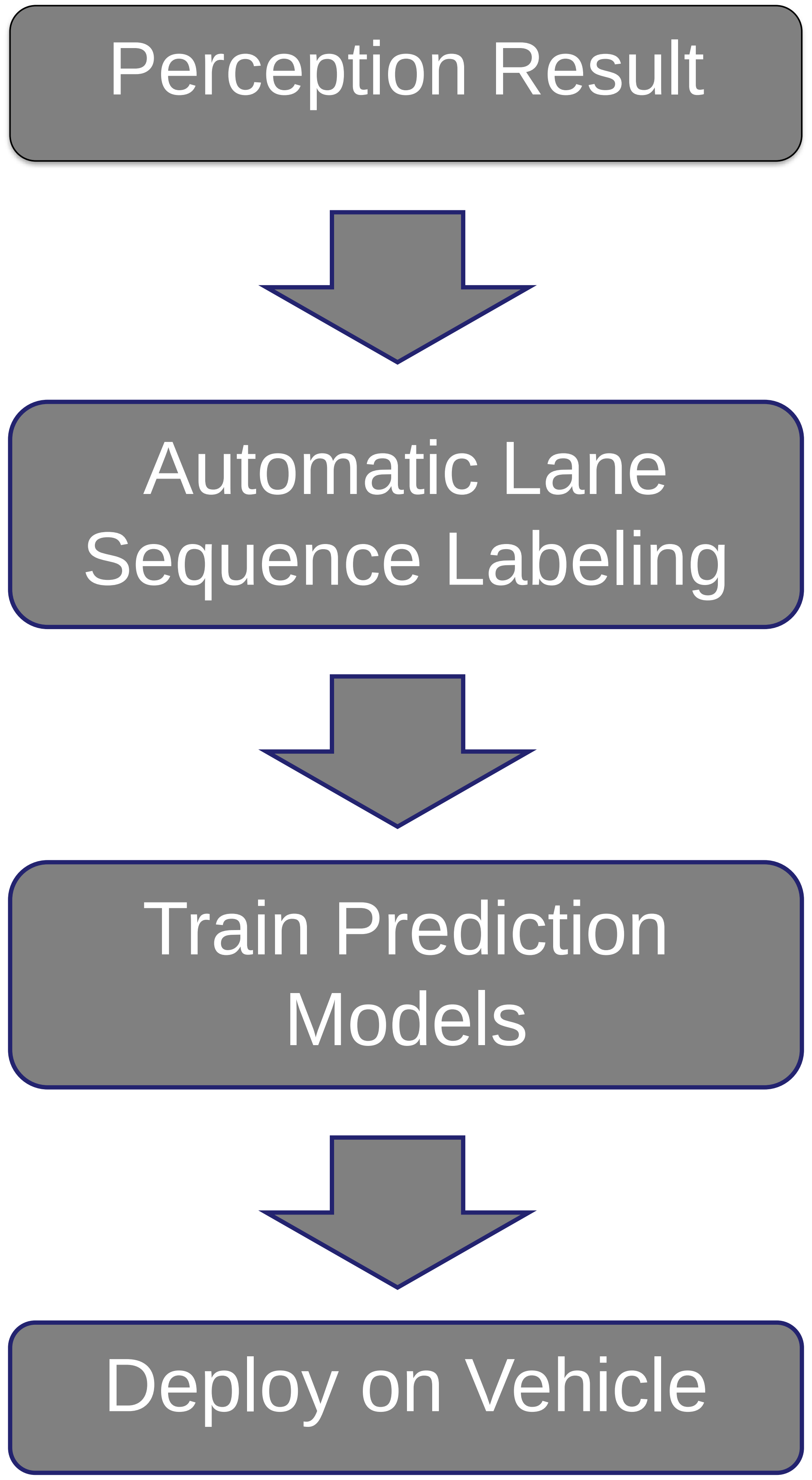
道路序列（Lane Sequence）展开

针对每一条道路序列Lane Sequence的二分类

对高概率的分类结果, 回归优化绘制轨迹

- 将复杂轨迹的预测问题，映射到离散有限的道路序列空间中
- 统一的道路序列模型，无需适用模型场景选择
- 无需人工标注，极大提高了感知数据的利用率，样本数累积迅速
- 有效利用了大数据和机器学习技术，进行特征抽取和模型训练
- 便于结合Planning轨迹优化的方法，提高了轨迹的优化效果

Data Pipeline



Perception Result

Automatic Lane Sequence Labeling

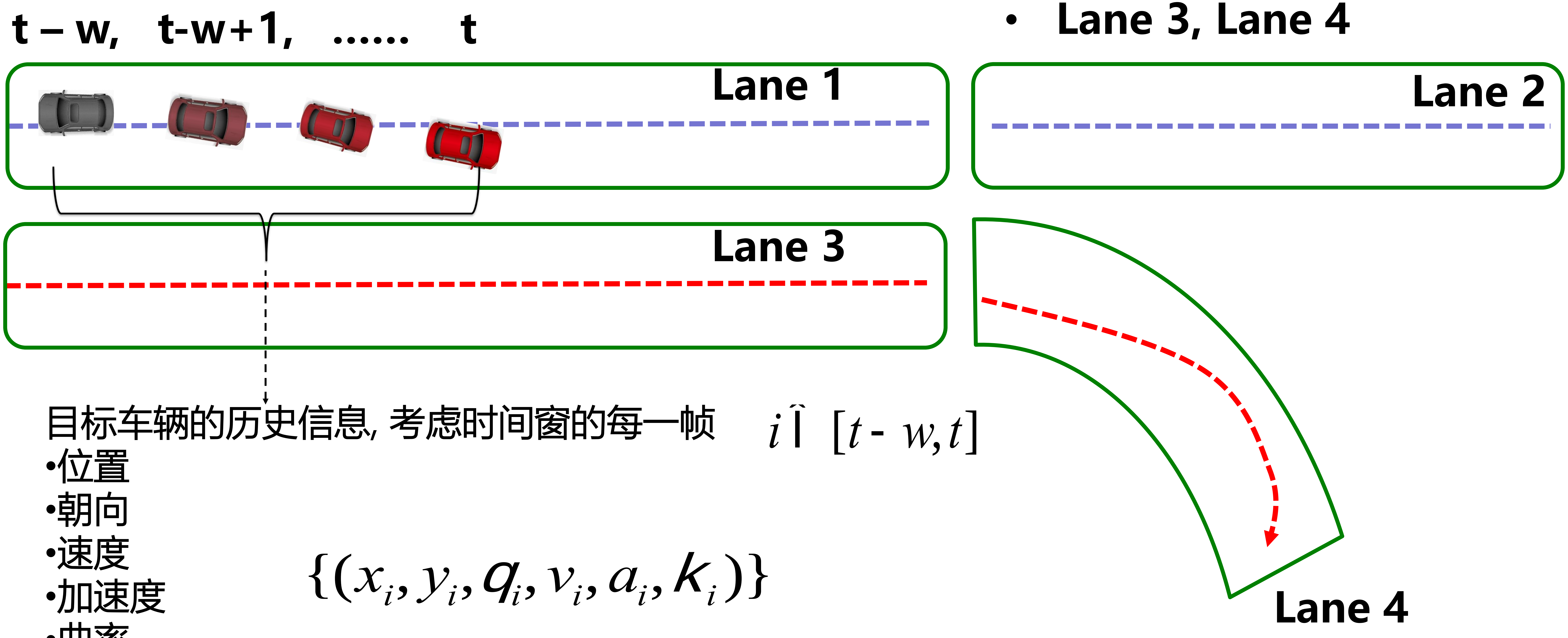
Train Prediction Models

Deploy on Vehicle

Feature：车辆历史信息

考虑目标道路序列:

- Lane 3, Lane 4



目标车辆的历史信息, 考虑时间窗的每一帧 $i \in [t-w,t]$

•位置

•朝向

•速度

•加速度

•曲率

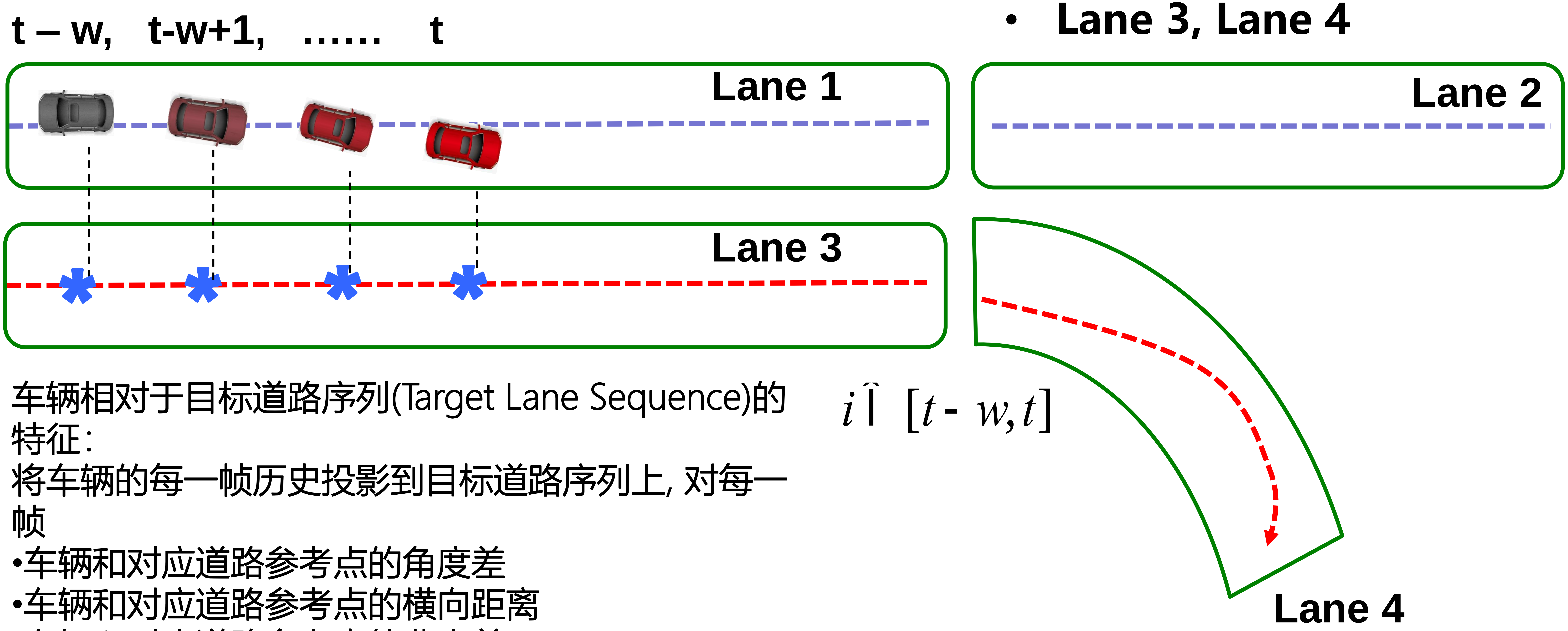
$$\{(x_i, y_i, \theta_i, v_i, a_i, \kappa_i)\}$$

以及由这些历史特征组成数组的统计特性

Feature：车辆结合道路

考虑目标道路序列:

- Lane 3, Lane 4



i Î $[t - w, t]$

车辆相对于目标道路序列(Target Lane Sequence)的特征：

将车辆的每一帧历史投影到目标道路序列上, 对每一帧

•车辆和对应道路参考点的角度差

•车辆和对应道路参考点的横向距离

•车辆和对应道路参考点的曲率差

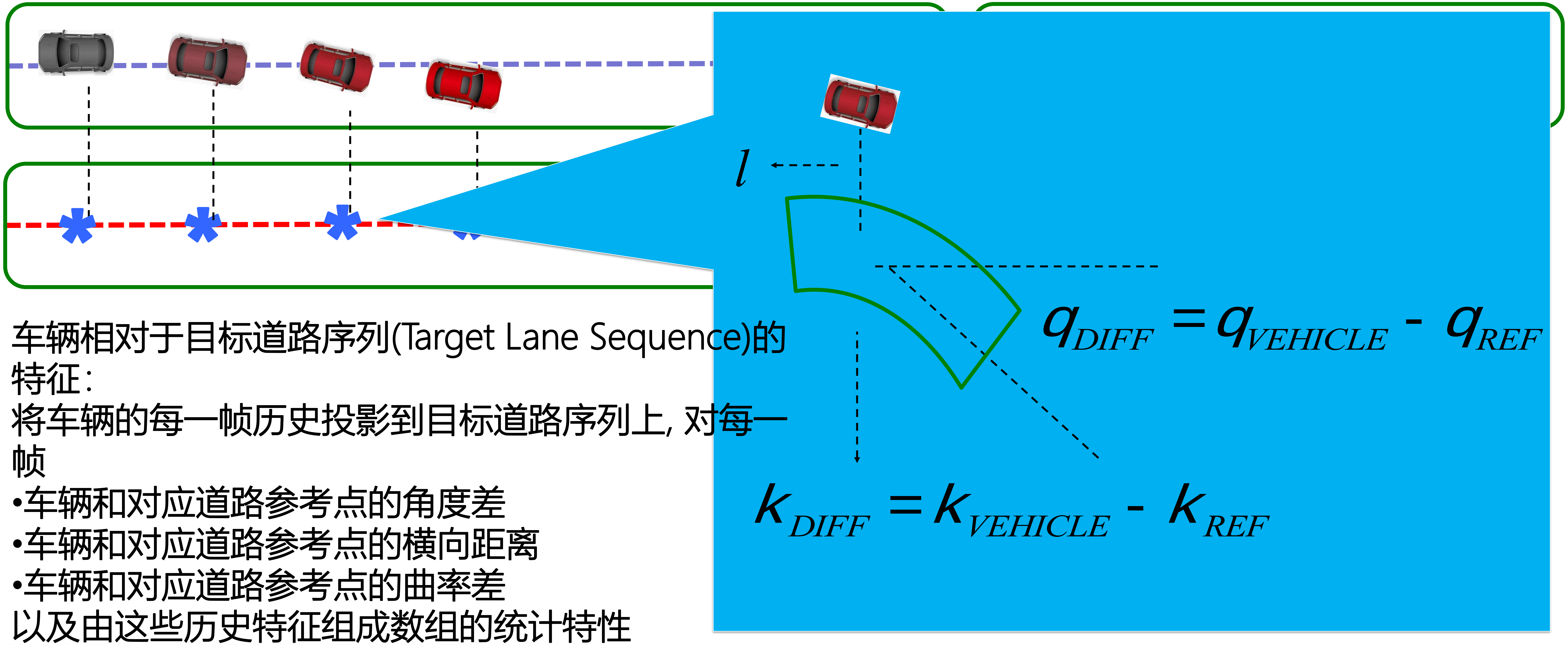
以及由这些历史特征组成数组的统计特性

Feature：车辆结合道路

t – w, t-w+1, …… t

考虑目标道路序列:

- Lane 3, Lane 4



$$q_{DIFF} = q_{VEHICLE} - q_{REF}$$

$$k_{DIFF} = k_{VEHICLE} - k_{REF}$$

车辆相对于目标道路序列(Target Lane Sequence)的特征：

将车辆的每一帧历史投影到目标道路序列上, 对每一帧

•车辆和对应道路参考点的角度差

•车辆和对应道路参考点的横向距离

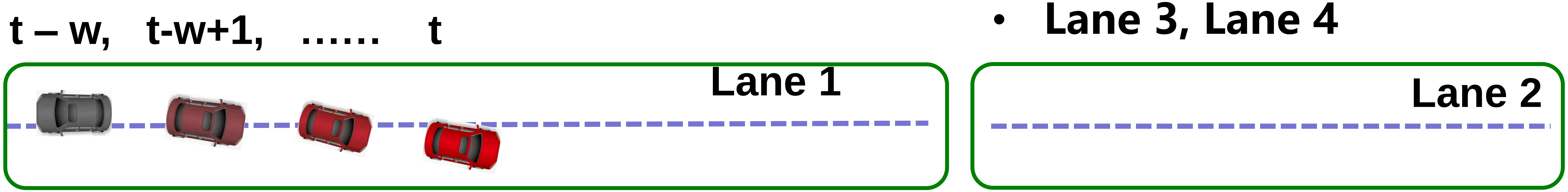
•车辆和对应道路参考点的曲率差

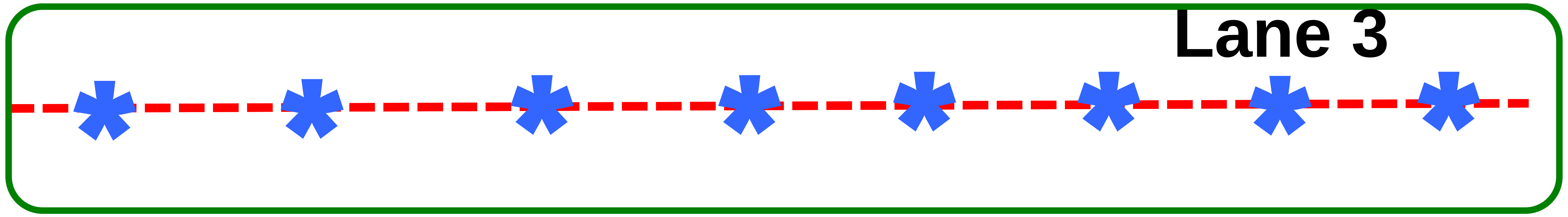
以及由这些历史特征组成数组的统计特性

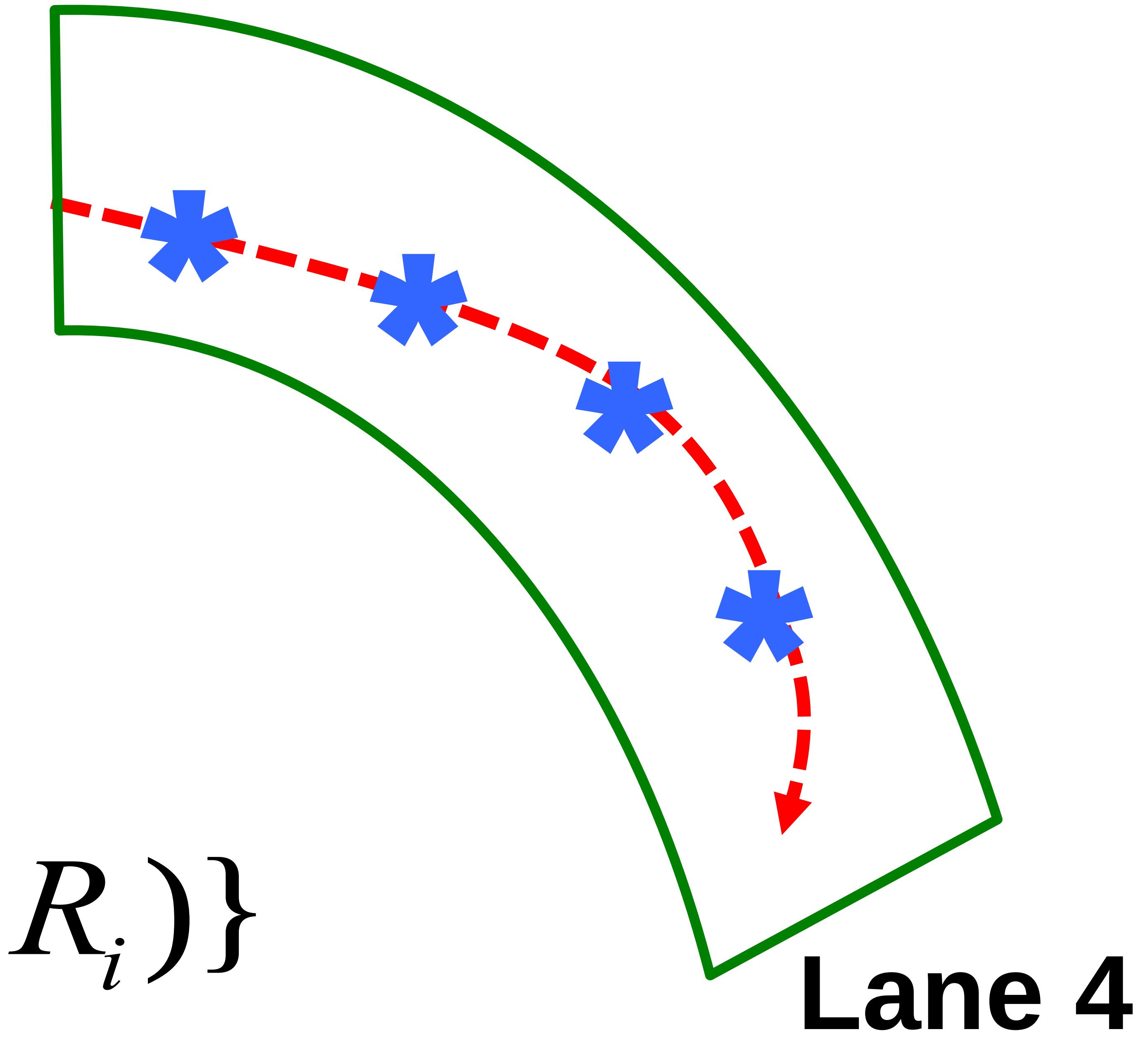
Feature：道路形状

考虑目标道路序列:

- Lane 3, Lane 4





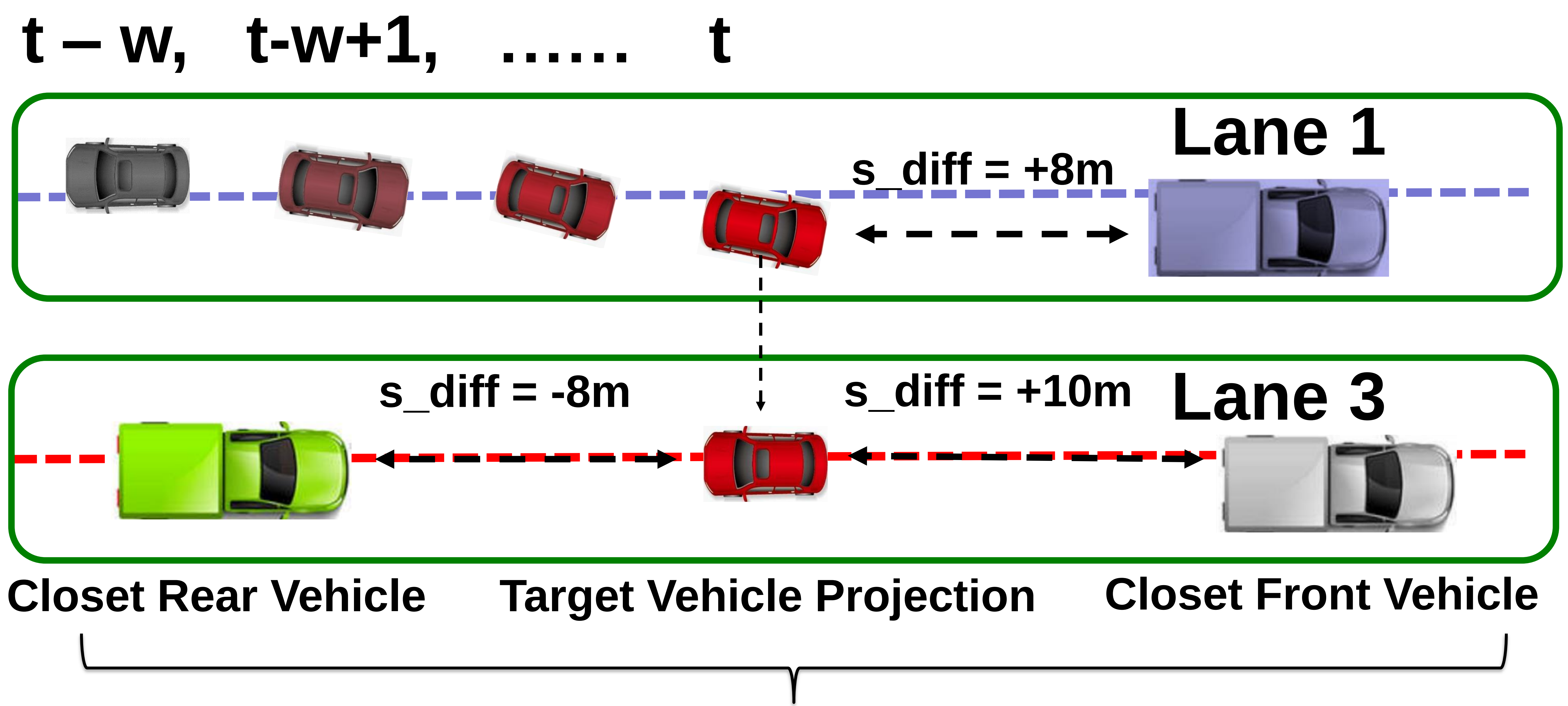


道路序列(Lane Sequence)未来展开的形状特征：
沿着道路序列的行进方向撒抽样点

- 位置
- 朝向
- 曲率
- 宽度

$$\{(x_i, y_i, q_i, k_i, width\_L_i, width\_R_i)\}$$

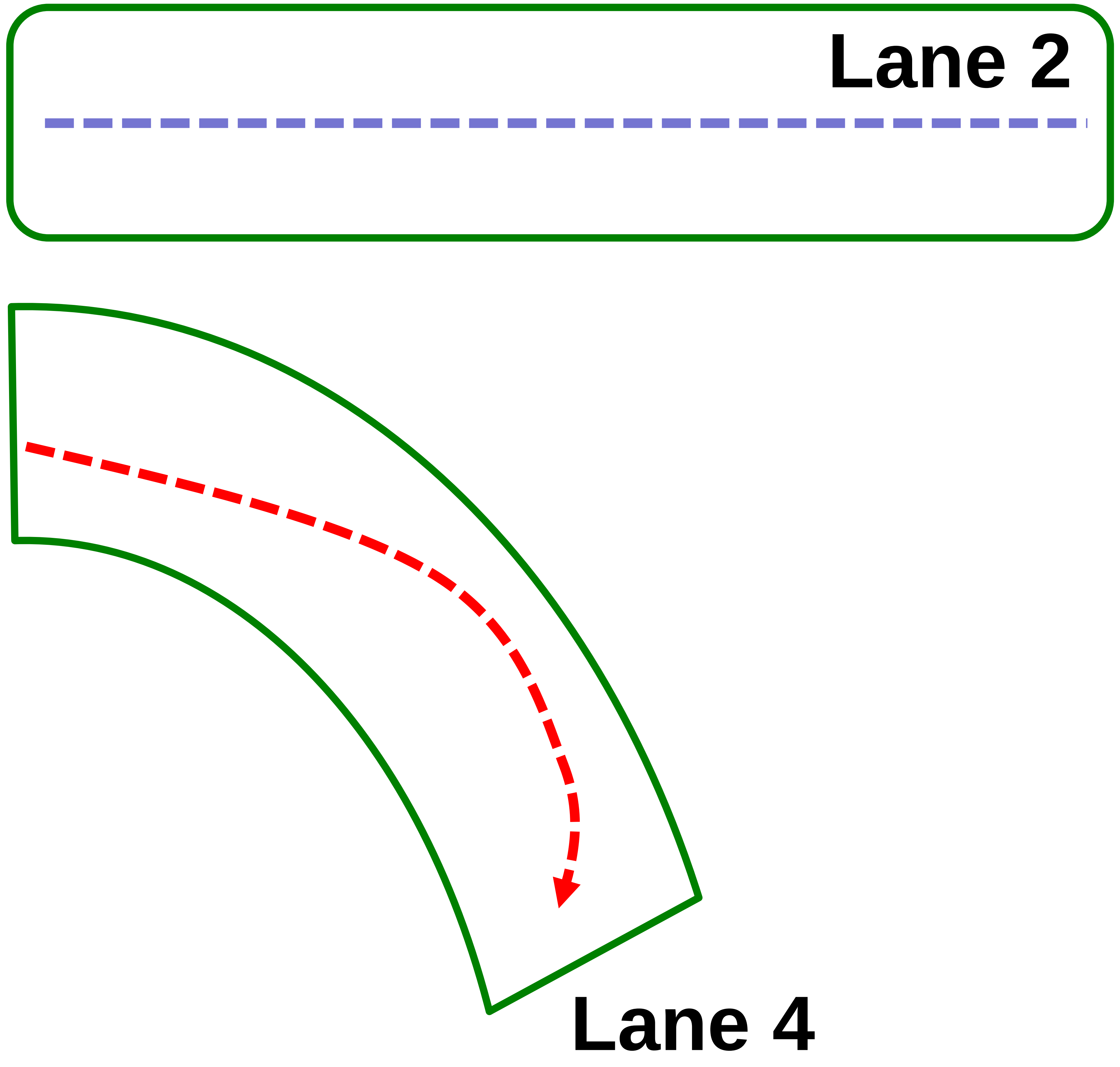
Feature：周边车辆信息



周边车辆的信息

考虑目标道路序列:

- Lane 3, Lane 4



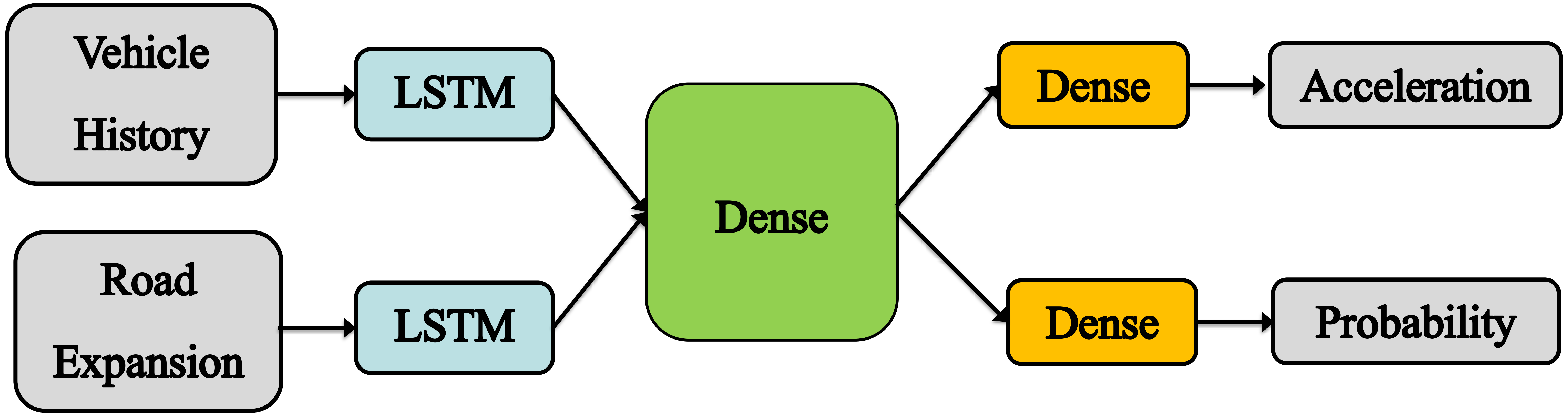
对于一个给定的道路序列(reference lane sequence)，
将主车投影到该Reference Lane Sequence上：

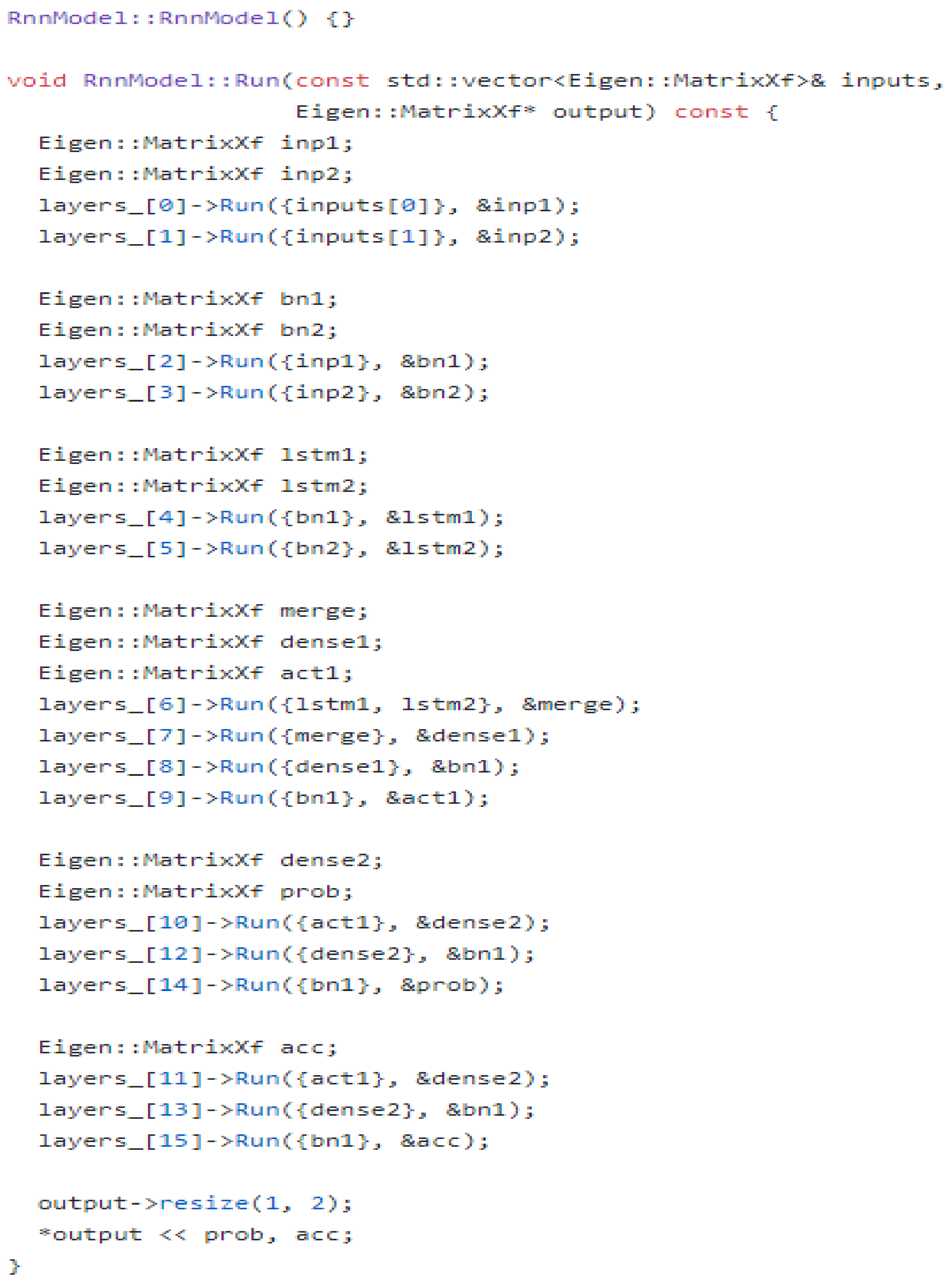
- •Closest rear vehicle s difference
- •Closest front vehicle s difference

同时考虑在主车当前Lane上的Closest front & rear vehicle s difference

RNN Model in Apollo

- One network modeling the historical information about obstacle
- One network modeling the future road path characteristics
- Two output for any legal output lane sequence
  - Probability
  - Accuracy





```
RnnModel::RnnModel() {}

void RnnModel::Run(const std::vector<Eigen::MatrixXf>& inputs,
                   Eigen::MatrixXf* output) const {
  Eigen::MatrixXf inp1;
  Eigen::MatrixXf inp2;
  layers_[0]->Run({inputs[0]}, &inp1);
  layers_[1]->Run({inputs[1]}, &inp2);

  Eigen::MatrixXf bn1;
  Eigen::MatrixXf bn2;
  layers_[2]->Run({inp1}, &bn1);
  layers_[3]->Run({inp2}, &bn2);

  Eigen::MatrixXf lstm1;
  Eigen::MatrixXf lstm2;
  layers_[4]->Run({bn1}, &lstm1);
  layers_[5]->Run({bn2}, &lstm2);

  Eigen::MatrixXf merge;
  Eigen::MatrixXf dense1;
  Eigen::MatrixXf act1;
  layers_[6]->Run({lstm1, lstm2}, &merge);
  layers_[7]->Run({merge}, &dense1);
  layers_[8]->Run({dense1}, &bn1);
  layers_[9]->Run({bn1}, &act1);

  Eigen::MatrixXf dense2;
  Eigen::MatrixXf prob;
  layers_[10]->Run({act1}, &dense2);
  layers_[12]->Run({dense2}, &bn1);
  layers_[14]->Run({bn1}, &prob);

  Eigen::MatrixXf acc;
  layers_[11]->Run({act1}, &dense2);
  layers_[13]->Run({dense2}, &bn1);
  layers_[15]->Run({bn1}, &acc);

  output->resize(1, 2);
  *output << prob, acc;
}
```

Prediction Result Demo Video

- Demo Video

Topic

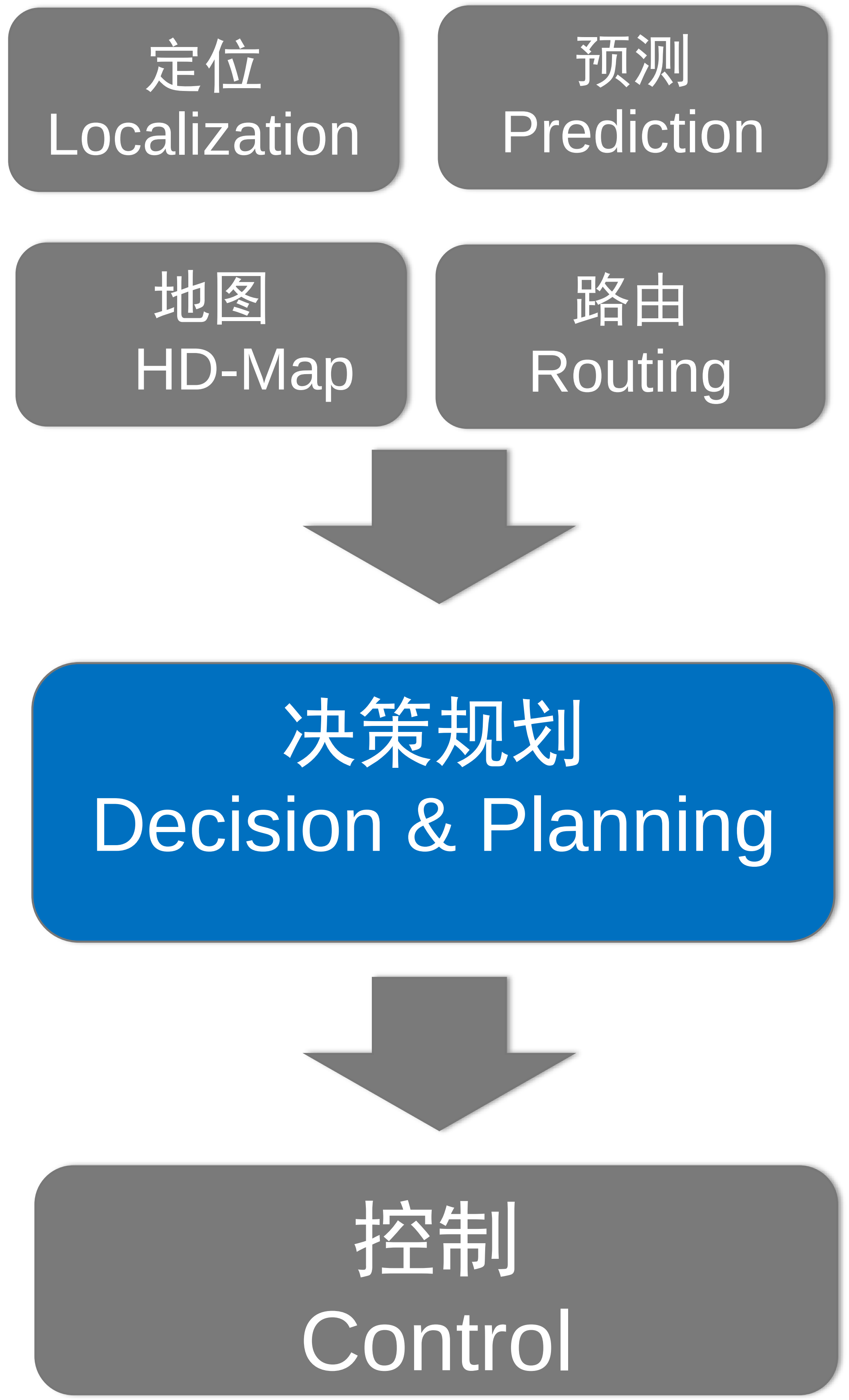
- Autonomous Driving Software System Modules

- Deep Learning in Autonomous Driving System Modules
  - Perception
  - Prediction
  - Decision & Planning

- Future Directions for AI in Autonomous Driving

规划问题(Planning)的重要术语(Terminology)

- *Planning :* 在一个较短的时间内，产生从一个点到另一个点的运动轨迹
  - 上游：定位，感知，预测，地图
  - 下游：控制
- *Path*: 路径 只描述位置形状(only the shape)
- *Trajectory*: 轨迹 不仅包含位置信息Path，还包含了沿着Path的时间和速度信息(describing both the shape and the movements/speeds along the Path)
- *Reference Line (指引线/基准线)*
  - 基于导航路由，一条足够平滑并且能够保证控制模块可以执行的，指引无人车前往目的地的曲线
- *Frenet Frame*: *(s,l)* system **given a reference line**
  - **S方向：车辆沿着指引线的径向方向**
  - **L方向：车辆垂直于指引线的横向方向**
- *ST Graph*: *描述车辆在(s,l)方向运动行为的图*



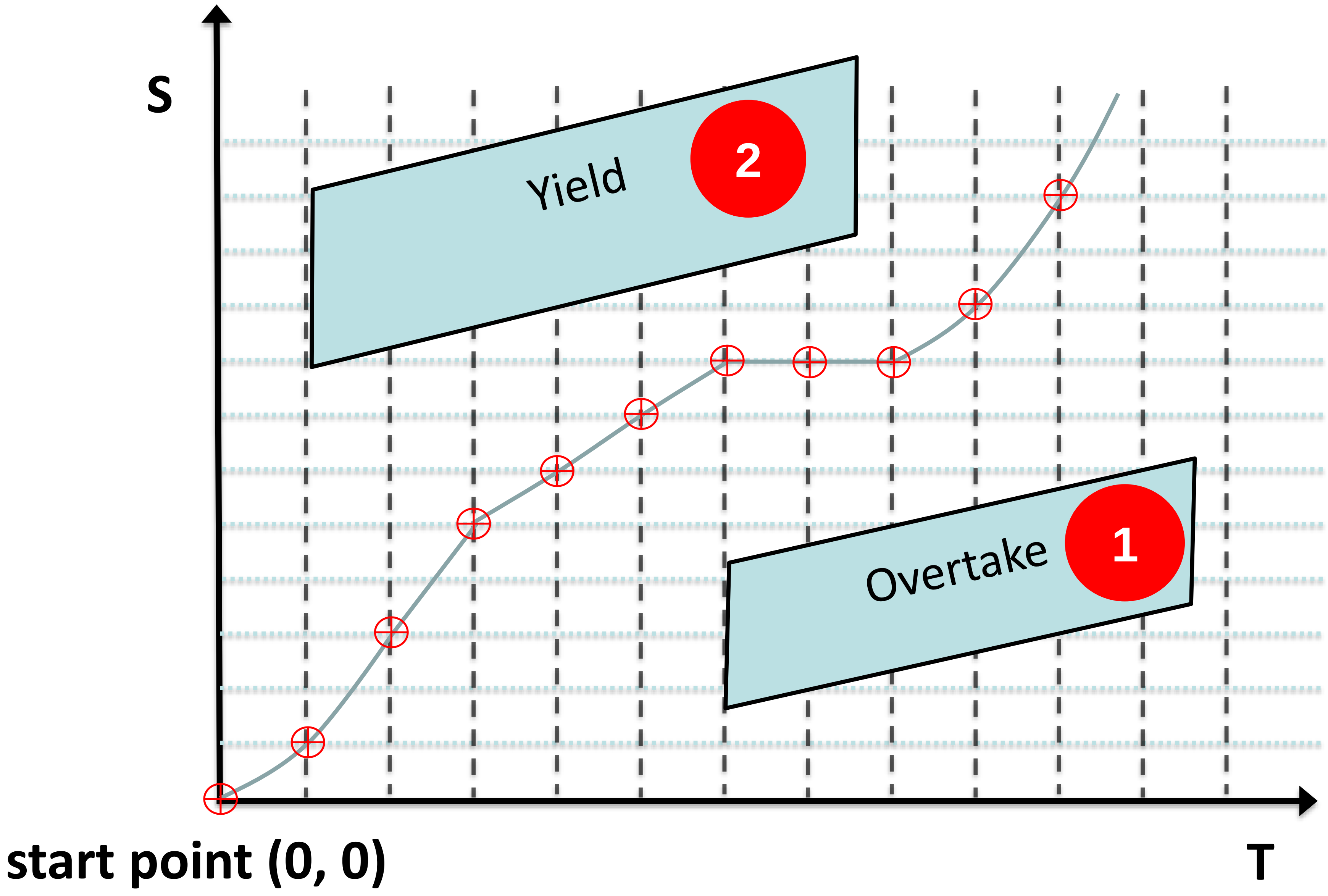
OReillyData

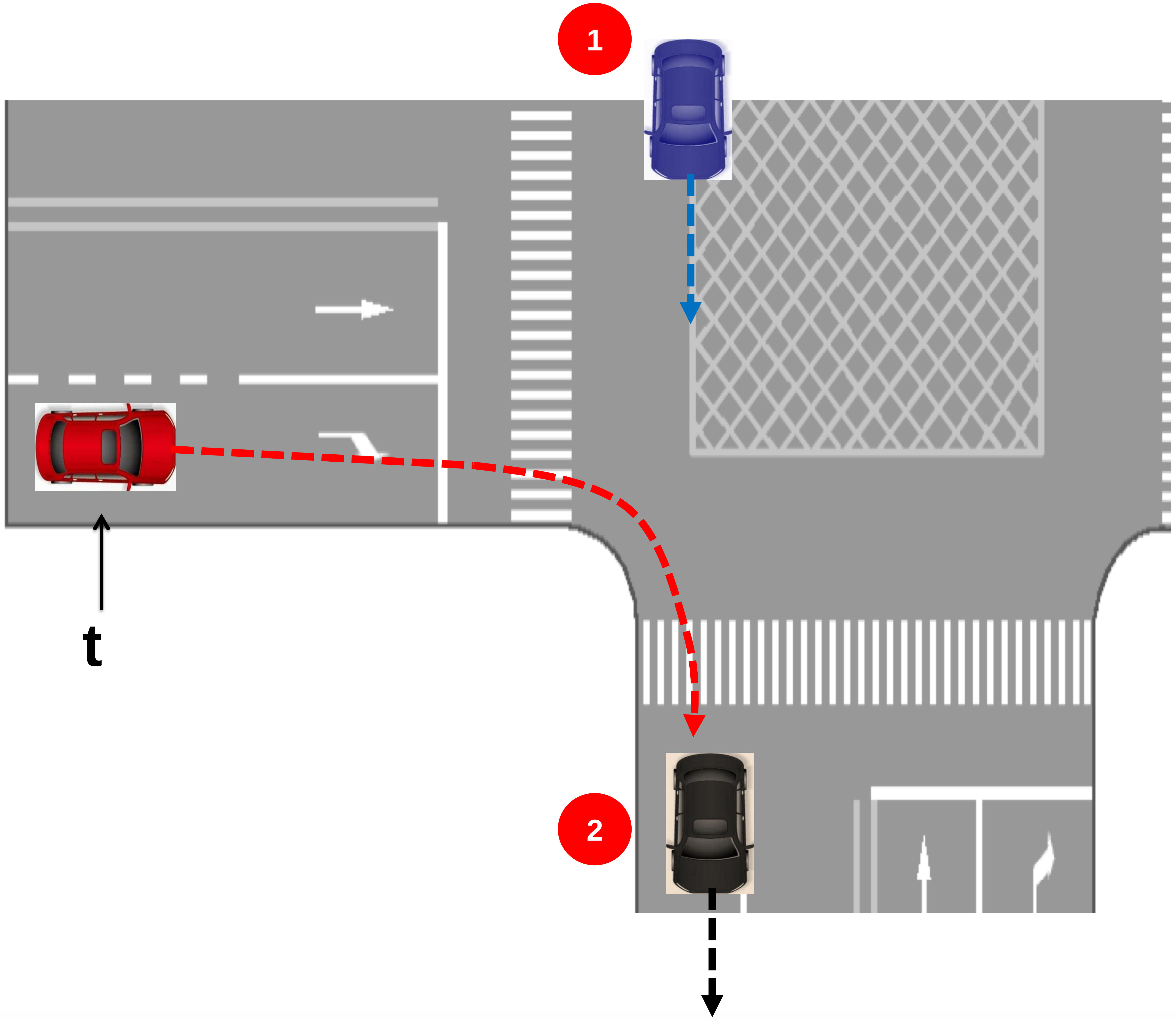
Artificial Intelligence

Some More About ST-Graph : An Example

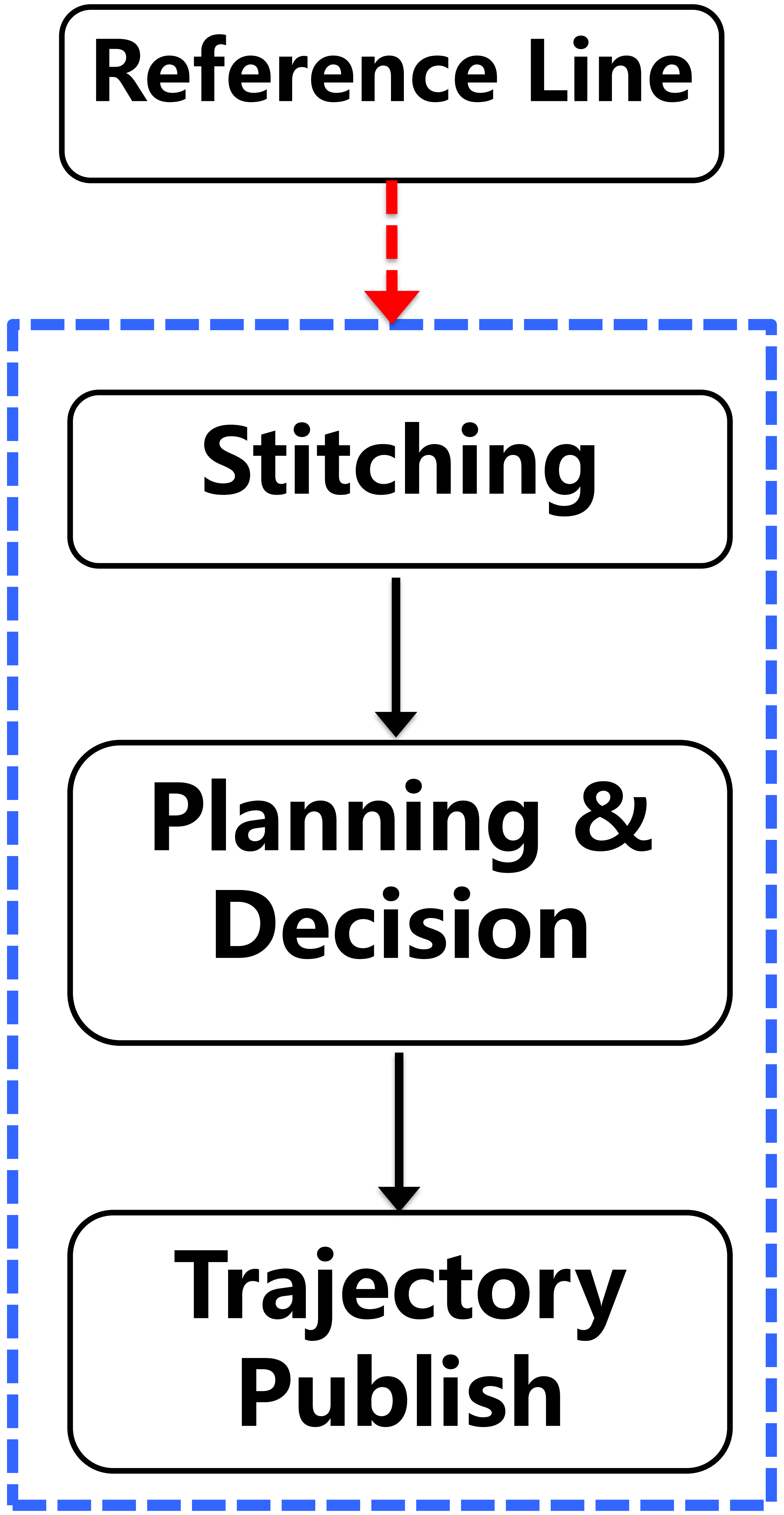
ST Graph:

- In planning, the XY coordinate system does not represent any semantics
- Planning happens on the SL coordinate system
- ST-Graph models the semantics along a certain path (s direction)





Apollo Planning的结构设计



1.计算指引线

2.产生规划起点

3.轨迹生成和优化

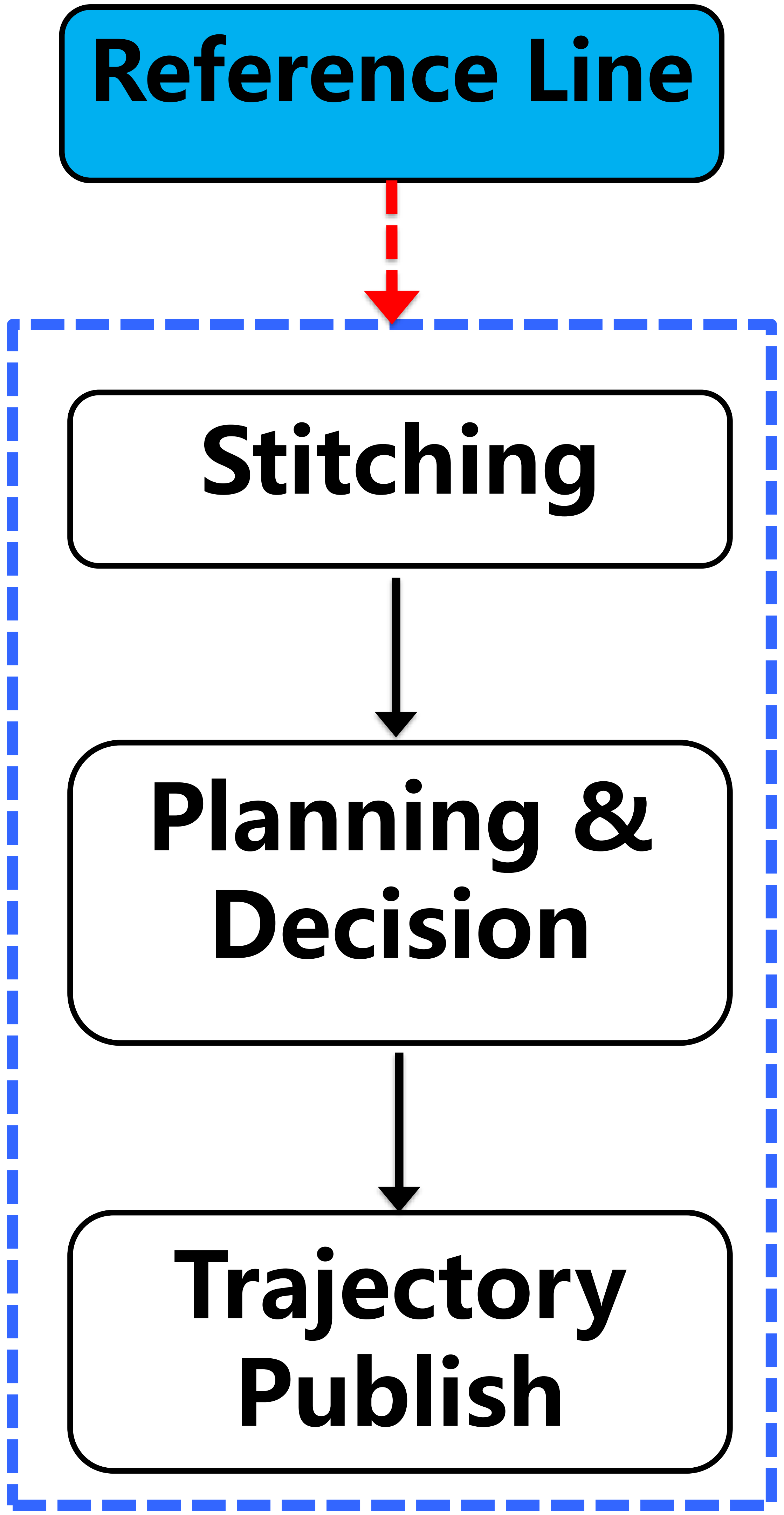
4.轨迹发布给下游

- 指引线既可以实时拼接生成，也可以离线载入. Apollo 2.0中采用实时拼接生成

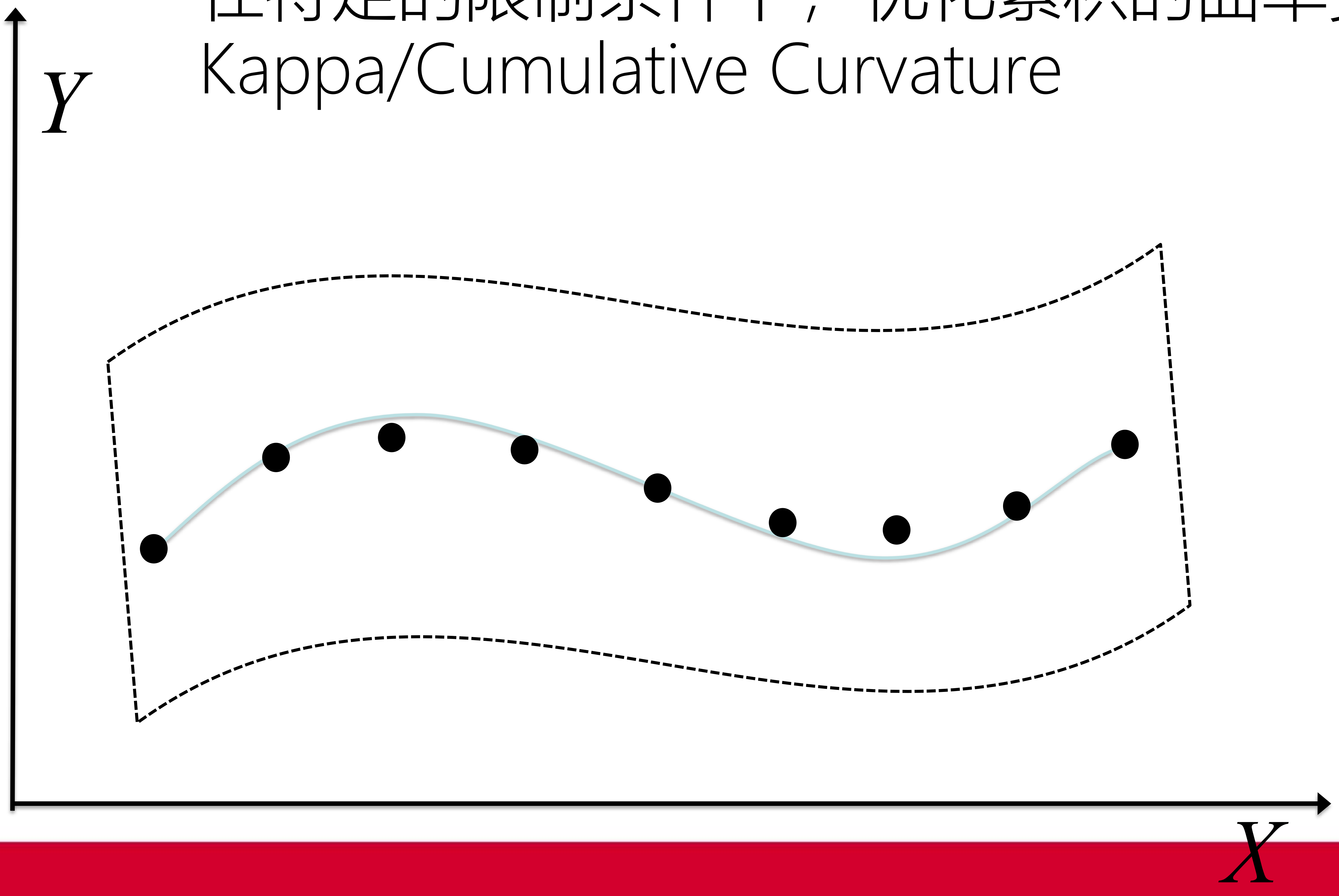
- 指引线生成和整体的决策规划模块可以独立分开异步运行

- 主Planning模块设计为时钟驱动Timer Based, 频率10Hz

Apollo Planning中的指引线优化算法



- 平滑的指引线是规划的基础
- 既可以离线载入，也可以异步运行
- 指引线优化问题的描述：
  - 在特定的限制条件下，优化累积的曲率变化率 Kappa/Cumulative Curvature



Apollo中的指引线优化算法

- The Problem Formalization: given a n sample points (usually along lane)

$$\begin{array}{ccccccccccc}
\theta_0 & & \theta_1 & & \theta_2 & \cdots & \theta_{n-2} & & \theta_{n-1} \\
\dot{\theta}_0 & & \dot{\theta}_1 & & \dot{\theta}_2 & \cdots & \dot{\theta}_{n-2} & & \dot{\theta}_{n-1} \\
\ddot{\theta}_0 & & \ddot{\theta}_1 & & \ddot{\theta}_2 & \cdots & \ddot{\theta}_{n-2} & & \ddot{\theta}_{n-1} \\
x_0 & & x_1 & & x_2 & \cdots & x_{n-2} & & x_{n-1} \\
y_0 & & y_1 & & y_2 & \cdots & y_{n-2} & & y_{n-1} \\
& \Delta s_0 & & \Delta s_1 & & \cdots & & \Delta s_{n-2} &
\end{array}$$

6n-1 variables

- Connect them piece-wisely with polynomial spirals. Formulated as :

$$\boldsymbol{\theta}_i(s) = a * s^5 + b * s^4 + c * s^3 + d * s^2 + e * s + f$$

, where

$$\begin{array}{ll}
\boldsymbol{\theta}_i(0) = \theta_i & \boldsymbol{\theta}_i(\Delta s) = \theta_{i+1} \\
\dot{\boldsymbol{\theta}}_i(0) = \dot{\theta}_i & \dot{\boldsymbol{\theta}}_i(\Delta s) = \dot{\theta}_{i+1} \\
\ddot{\boldsymbol{\theta}}_i(0) = \ddot{\theta}_i & \ddot{\boldsymbol{\theta}}_i(\Delta s) = \ddot{\theta}_{i+1}
\end{array}$$

Apollo中的指引线优化算法

- Optimization Goal:

$$\sum_{i=0}^{n-2}\sum_{j=0}^{m-1}\ddot{\boldsymbol{\theta}}_i(s_j)^2$$

- Constraints:

-Equality of continuity : (2n-2) equality constraints

$$x = x_i + \int_0^s \boldsymbol{cos}(\boldsymbol{\theta_i}(s))ds$$

$$y = y_i + \int_0^s \boldsymbol{sin}(\boldsymbol{\theta_i}(s))ds$$

→

Gauss-Lengedre Approximation

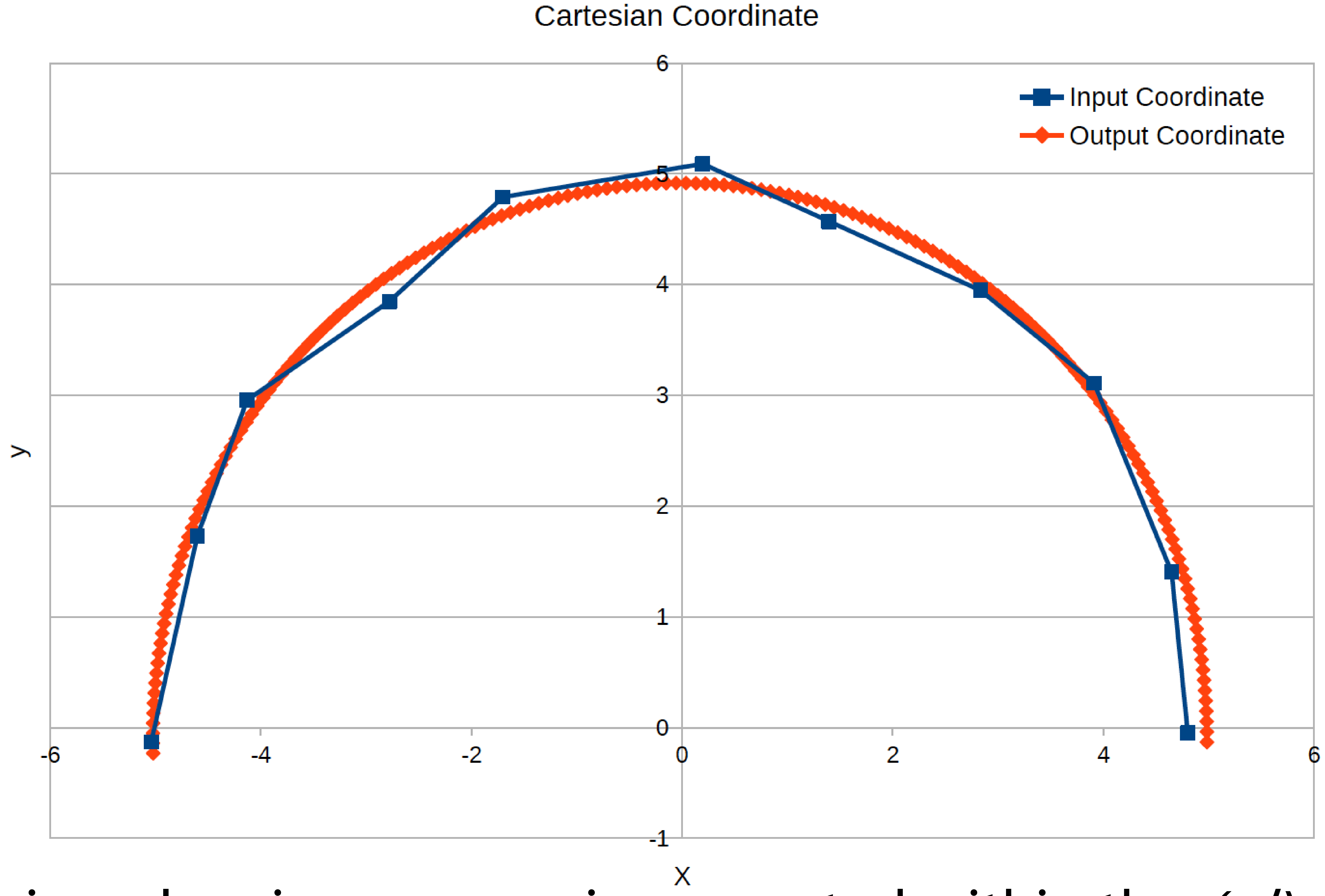
$$x_{i+1} - x_i - \frac{1}{2}\Delta s_i * \sum_{i=0}^{N-1} w_i * \boldsymbol{cos}(\boldsymbol{\theta}(\frac{1}{2}\Delta s_i * g_i + \frac{1}{2}\Delta s_i)) = 0$$

$$y_{i+1} - y_i - \frac{1}{2}\Delta s_i * \sum_{i=0}^{N-1} w_i * \boldsymbol{sin}(\boldsymbol{\theta}(\frac{1}{2}\Delta s_i * g_i + \frac{1}{2}\Delta s_i)) = 0$$

-Inequality of deviation radius $(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2 \leq r_i^2$

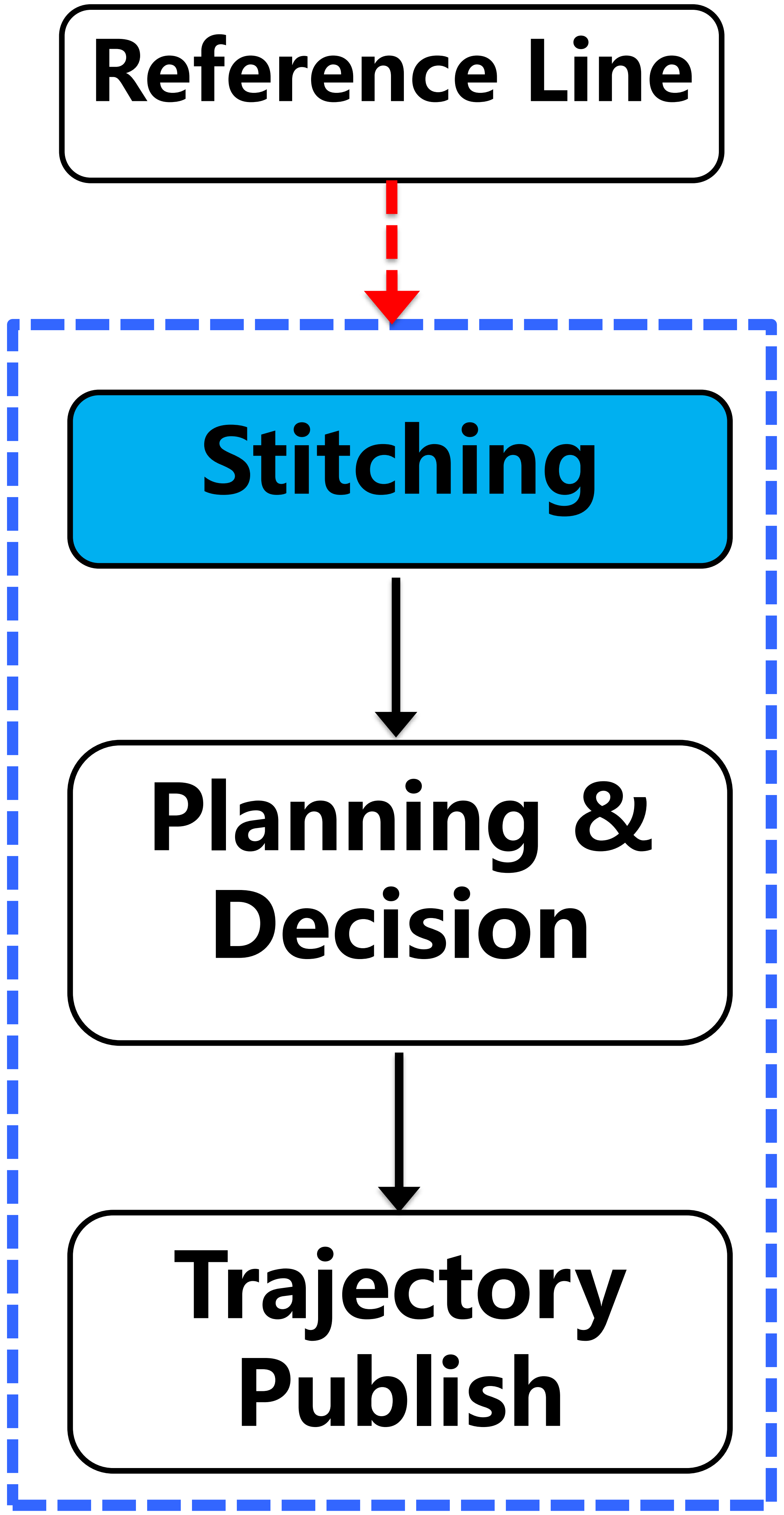
优化效果

- An example:

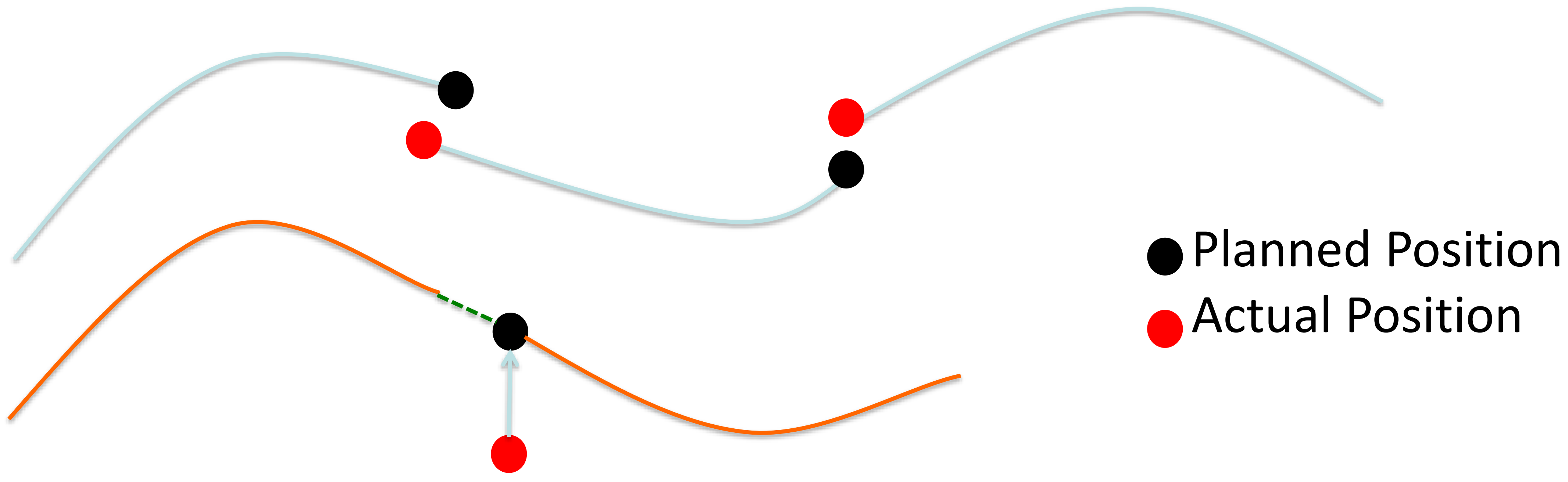


- All the remaining planning process is computed within the *(s,l)* coordinate system as per reference line

Apollo Planning中的Stitching

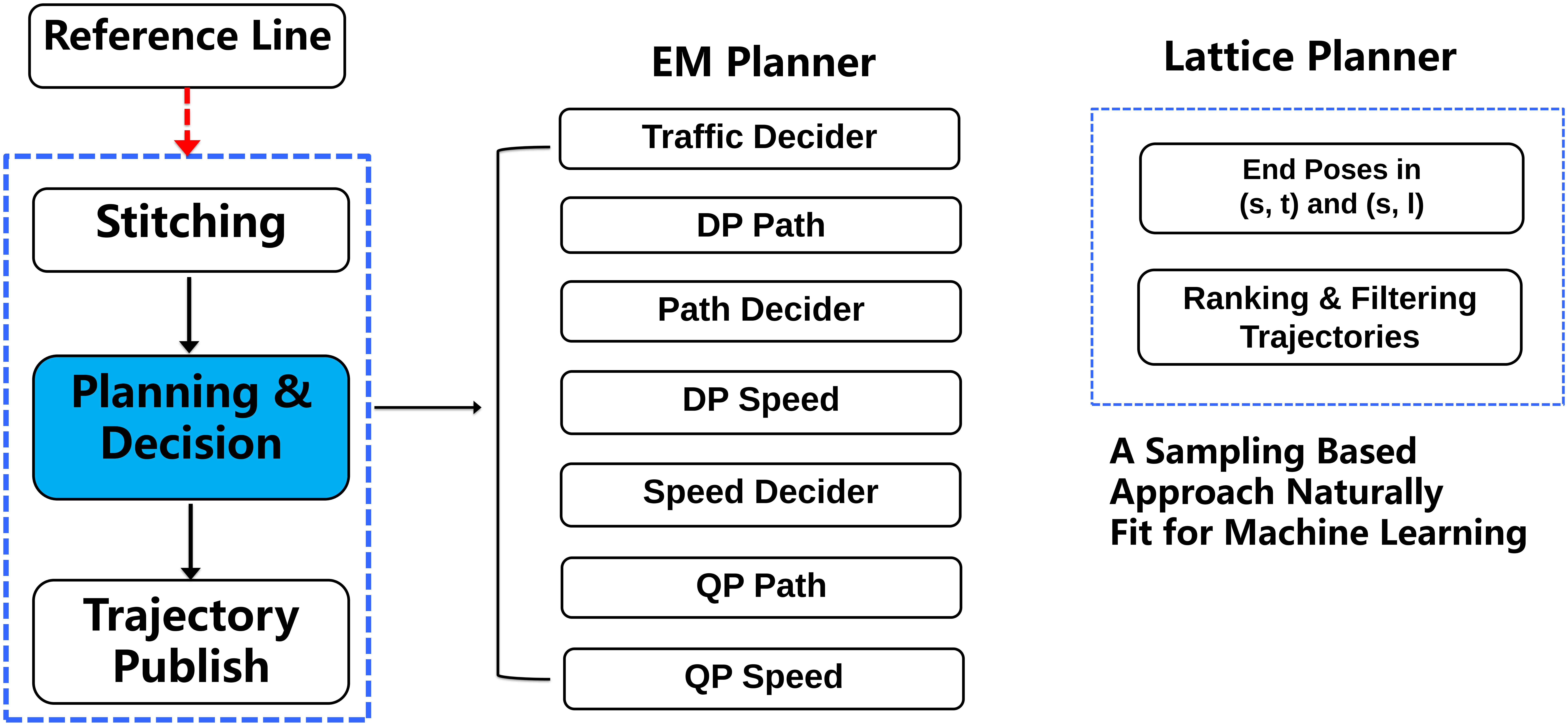


- 在每个规划周期(Frame/Cycle), 规划的起始点是哪里?
- 一种显而易见的方法是从主车当前位置作为起始点开始规划
- 但由于控制不能完美追踪上一个周期的轨迹，从主车当前位置开始规划会导致发给控制模块的轨迹不连续



- 解决方法: 将主车当前位置投影到上一个规划周期的轨迹中，将该投影点作为当前周期的规划起始位置 “Stitch” the current planning start point to previous planning trajectory

Apollo系统中的Planning设计



Reference Line

Stitching

Planning & Decision

Trajectory Publish

EM Planner

Traffic Decider

DP Path

Path Decider

DP Speed

Speed Decider

QP Path

QP Speed

Lattice Planner

End Poses in (s, t) and (s, l)

Ranking & Filtering Trajectories

A Sampling Based Approach Naturally Fit for Machine Learning

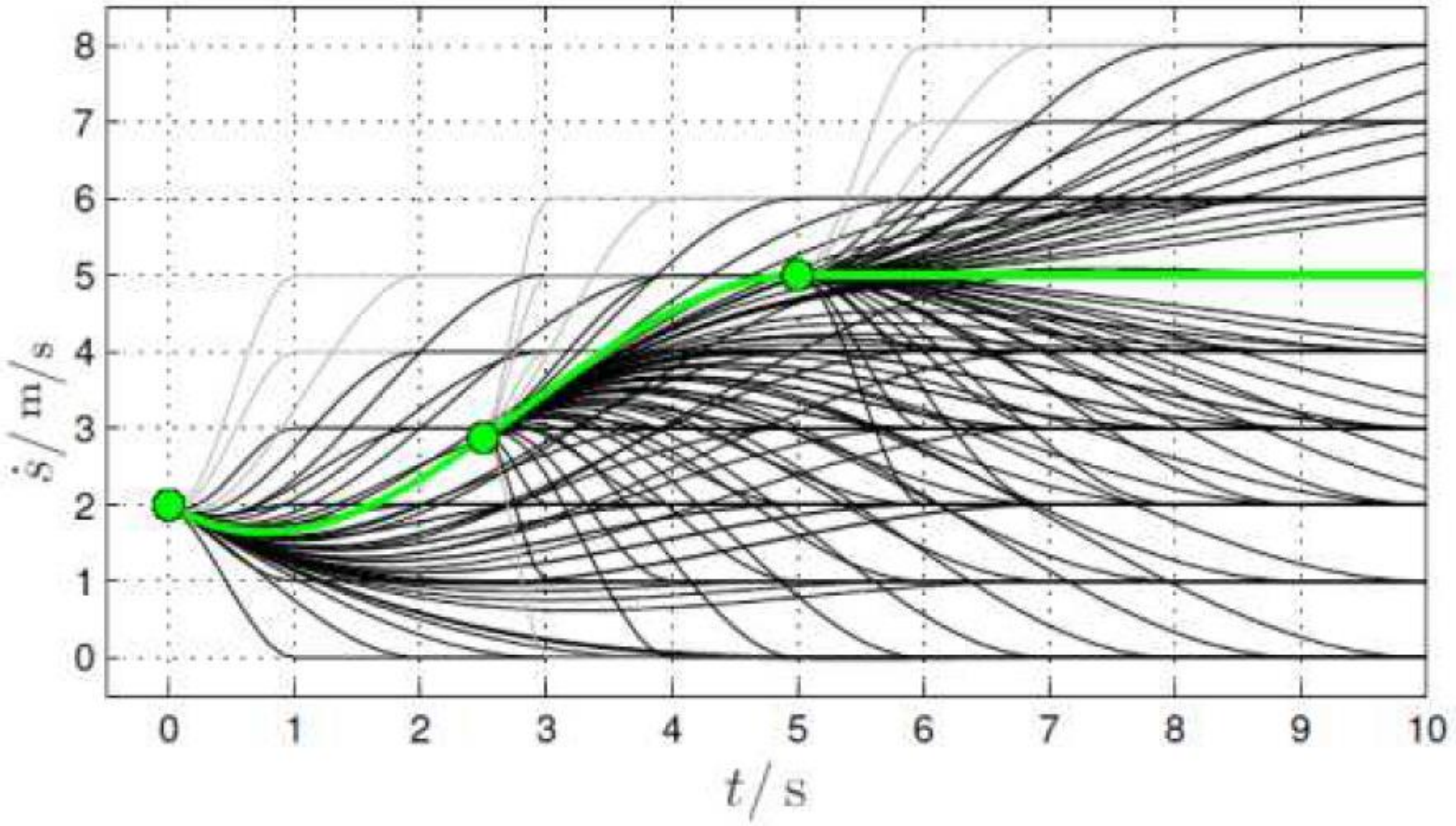
Apollo中的Lattice Planner

- **目标**: 规划出一条符合交规，安全，舒适，可执行的动作规矩
- **参考文献** Stanford’s method on DARPA Grand/Urban Challenge.

  Optimal trajectory generation for dynamic street scenarios in a Frenet Frame

  ICRA 2010 M.Werling, J.Ziegler, S.Kammel, S. Thrun
- **为什么称为 “Lattice Planner” ?** Use different Decisions to sample fixed-patterned end conditions (time, position, velocity, acceleration)

Stanford Junior



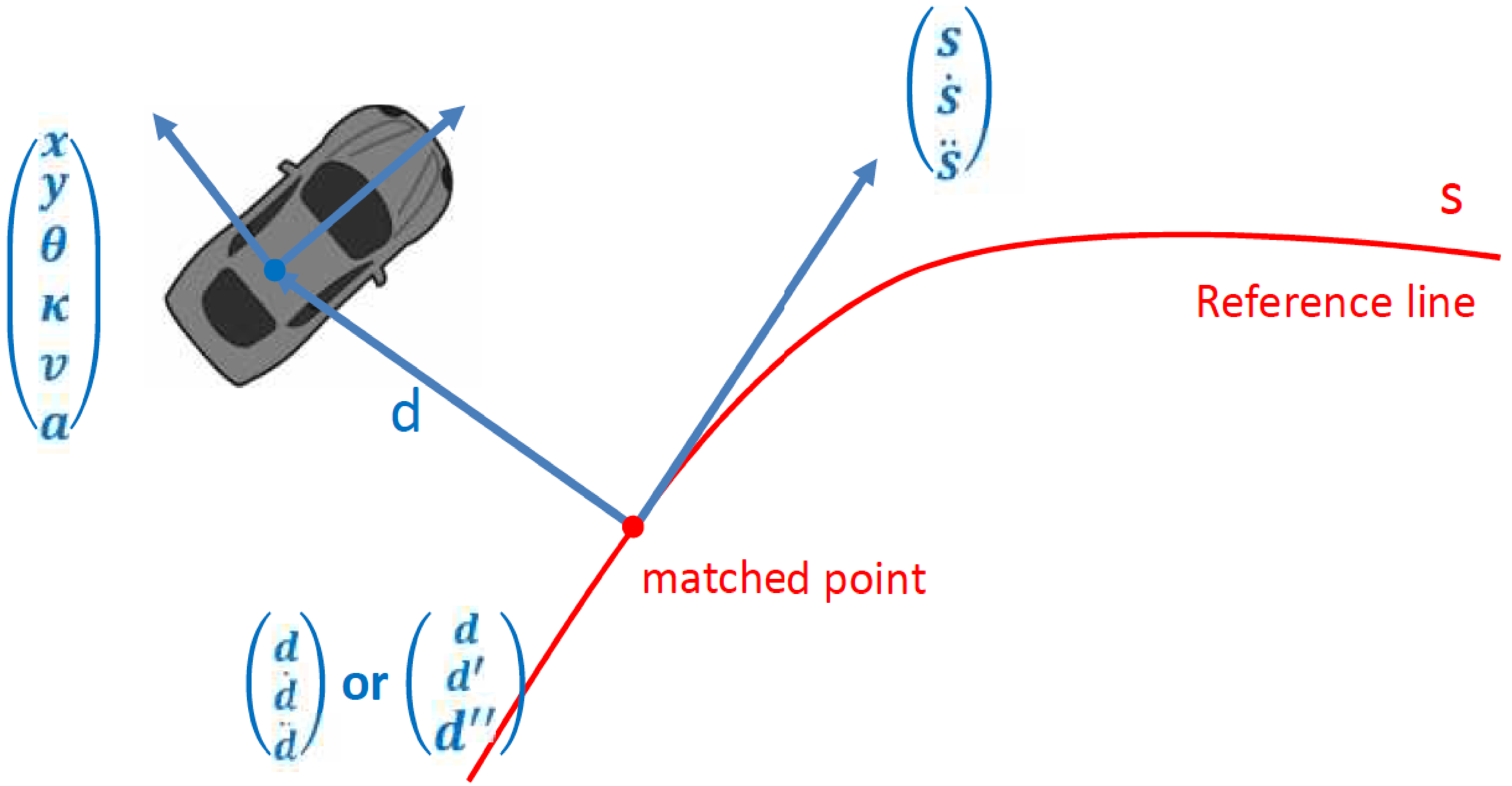
End conditions for cruising

Apollo中的Lattice Planner

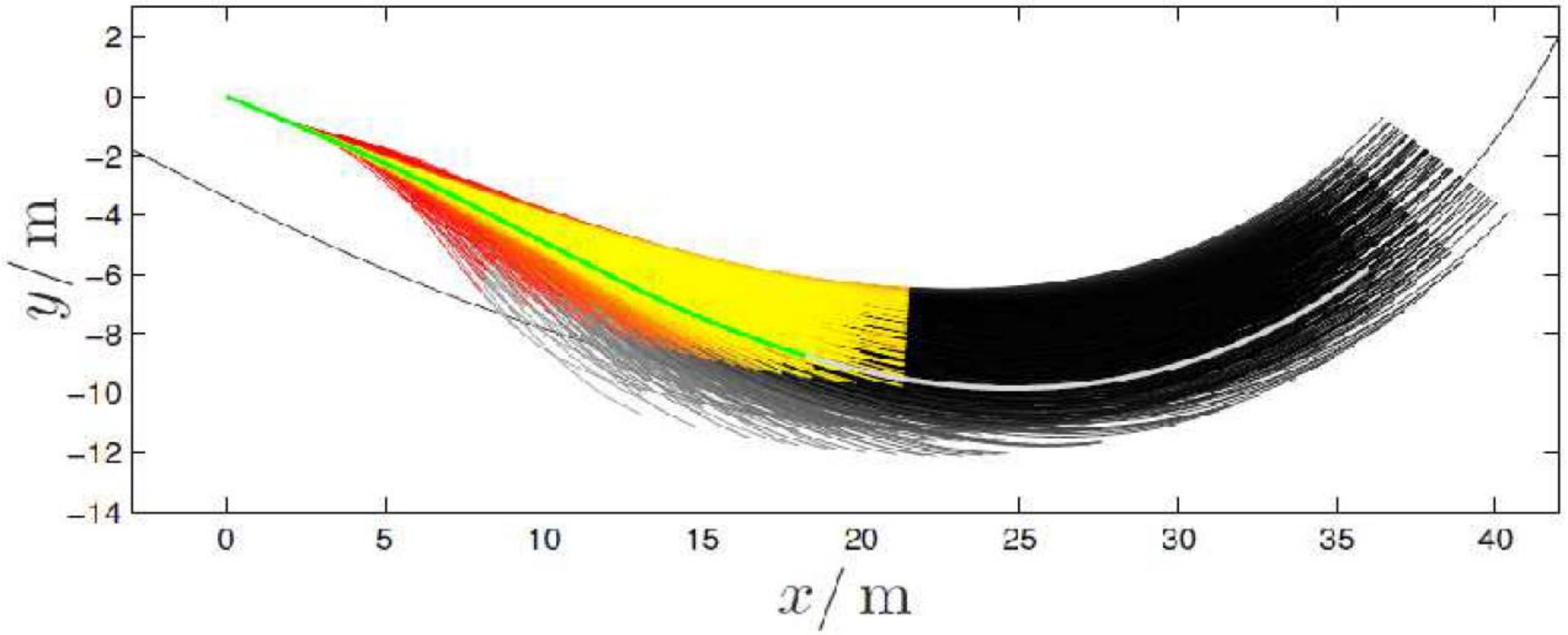
- 基于运动学的规划方法, 同时产生轨迹形状和速度
- 在Frenet Frame中按照指引线(s,l)方向进行规划
  - 纵向和横向分开独立规划
  - 形状和速度在一维空间联合优化，保证控制模块能够执行
  - 城市道路环境和高速都

算法步骤：

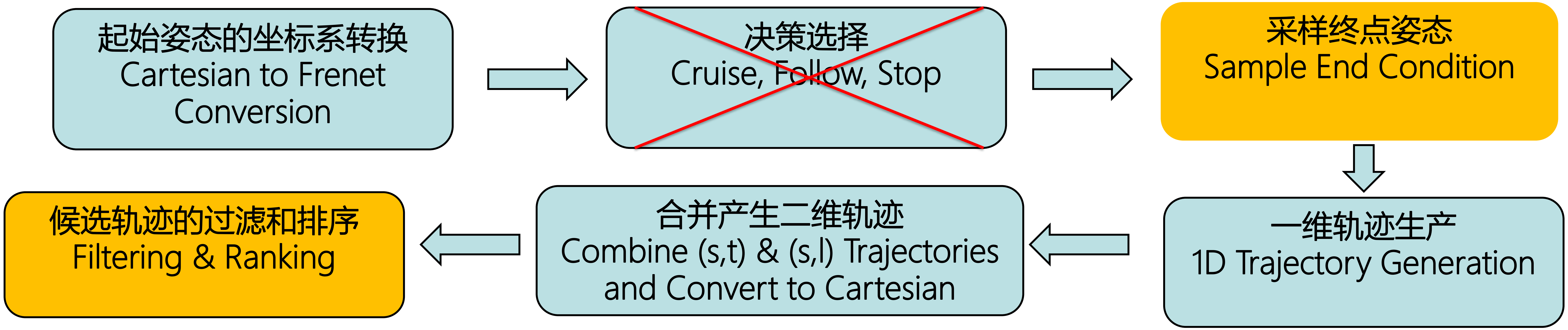
- 首先分别独立在纵向和横向采样大量的终端姿态点
- 使用5阶多项式进行连接起始和终点姿态
- 在一维空间内简单过滤轨迹后，合并产生候选的时空轨迹
- 过滤+排序： 选出最优的轨迹



$\begin{pmatrix} x \\ y \\ \theta \\ \kappa \\ v \\ a \end{pmatrix}$
$\begin{pmatrix} s \\ \dot{s} \\ \ddot{s} \end{pmatrix}$
s
Reference line
d
matched point
$\begin{pmatrix} d \\ \dot{d} \\ \ddot{d} \end{pmatrix}$ or $\begin{pmatrix} d \\ d' \\ d'' \end{pmatrix}$



Apollo中的Lattice Planner



起始姿态的坐标系转换
Cartesian to Frenet Conversion

~~决策选择~~
~~Cruise, Follow, Stop~~

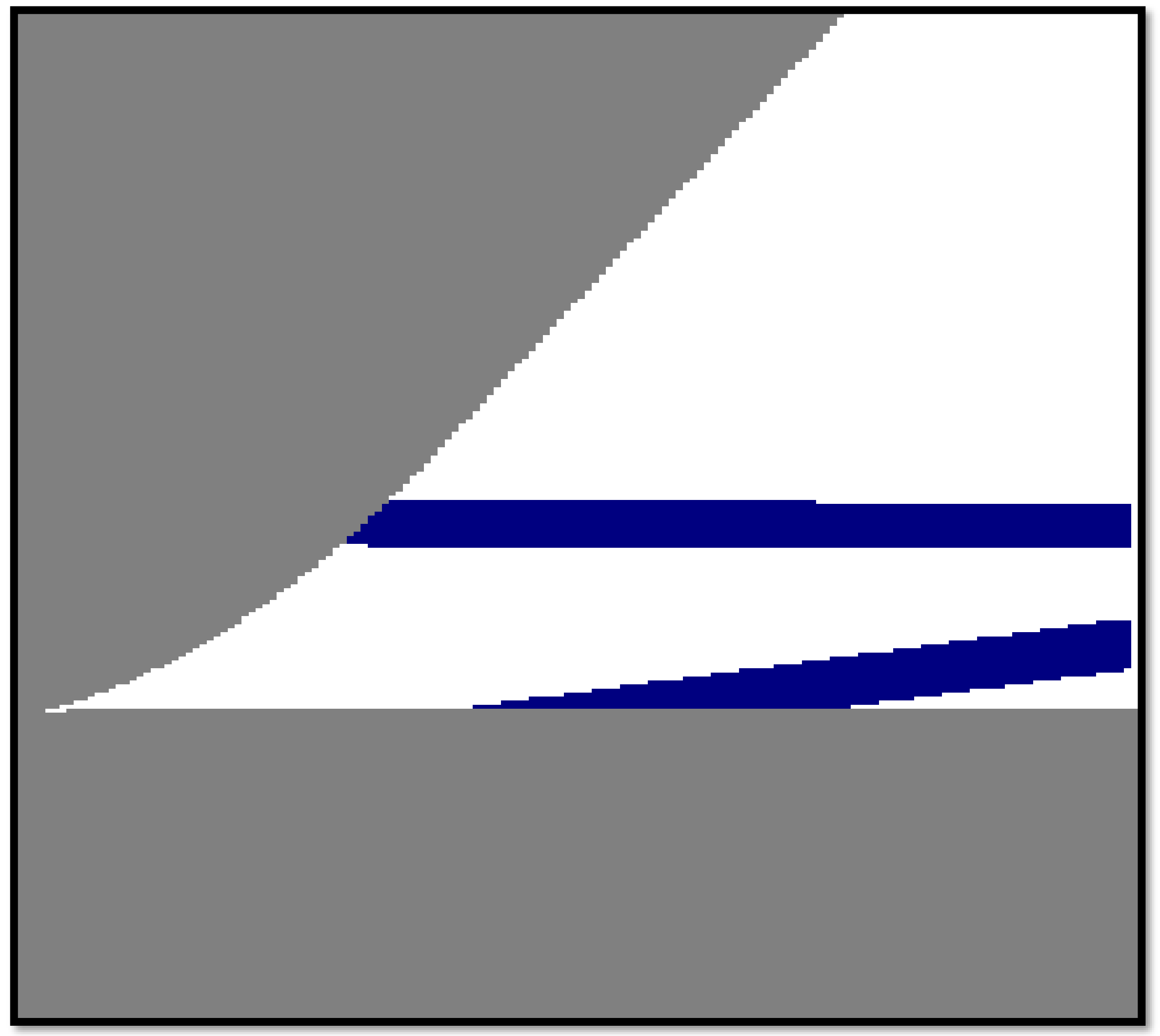
采样终点姿态
Sample End Condition

一维轨迹生产
1D Trajectory Generation

合并产生二维轨迹
Combine (s,t) & (s,l) Trajectories and Convert to Cartesian

候选轨迹的过滤和排序
Filtering & Ranking

ST Graph



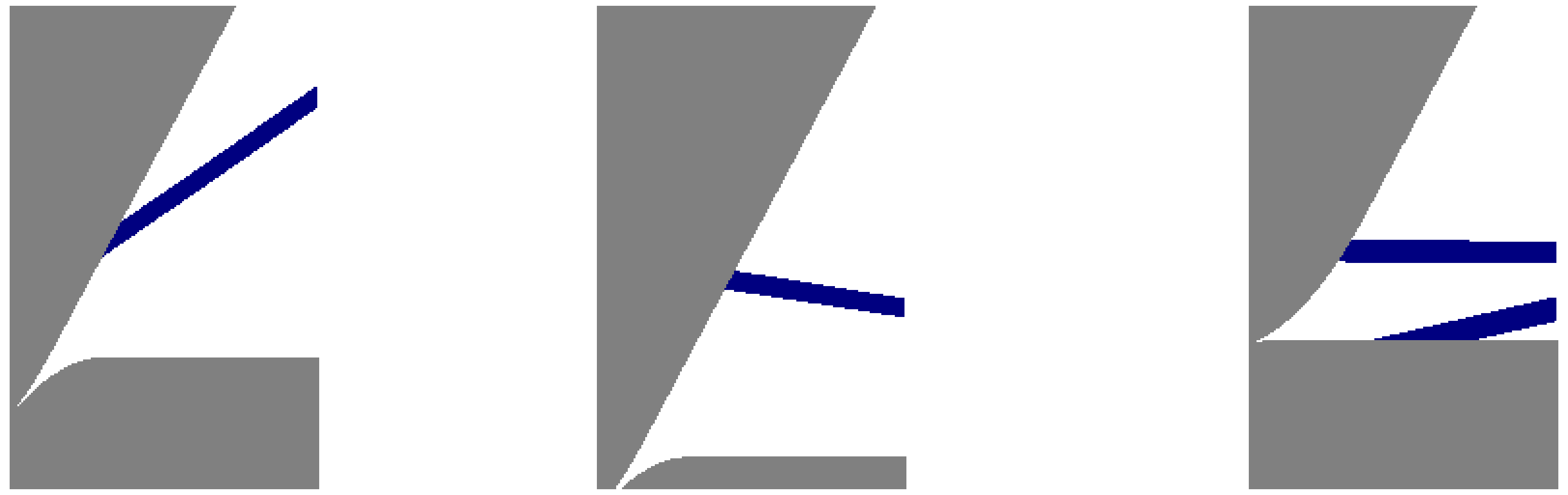
- 行为层面的“人”的决策被体现在
  - 终点姿态的采样策略上
  - 候选轨迹的Cost分量的不同权重选择上

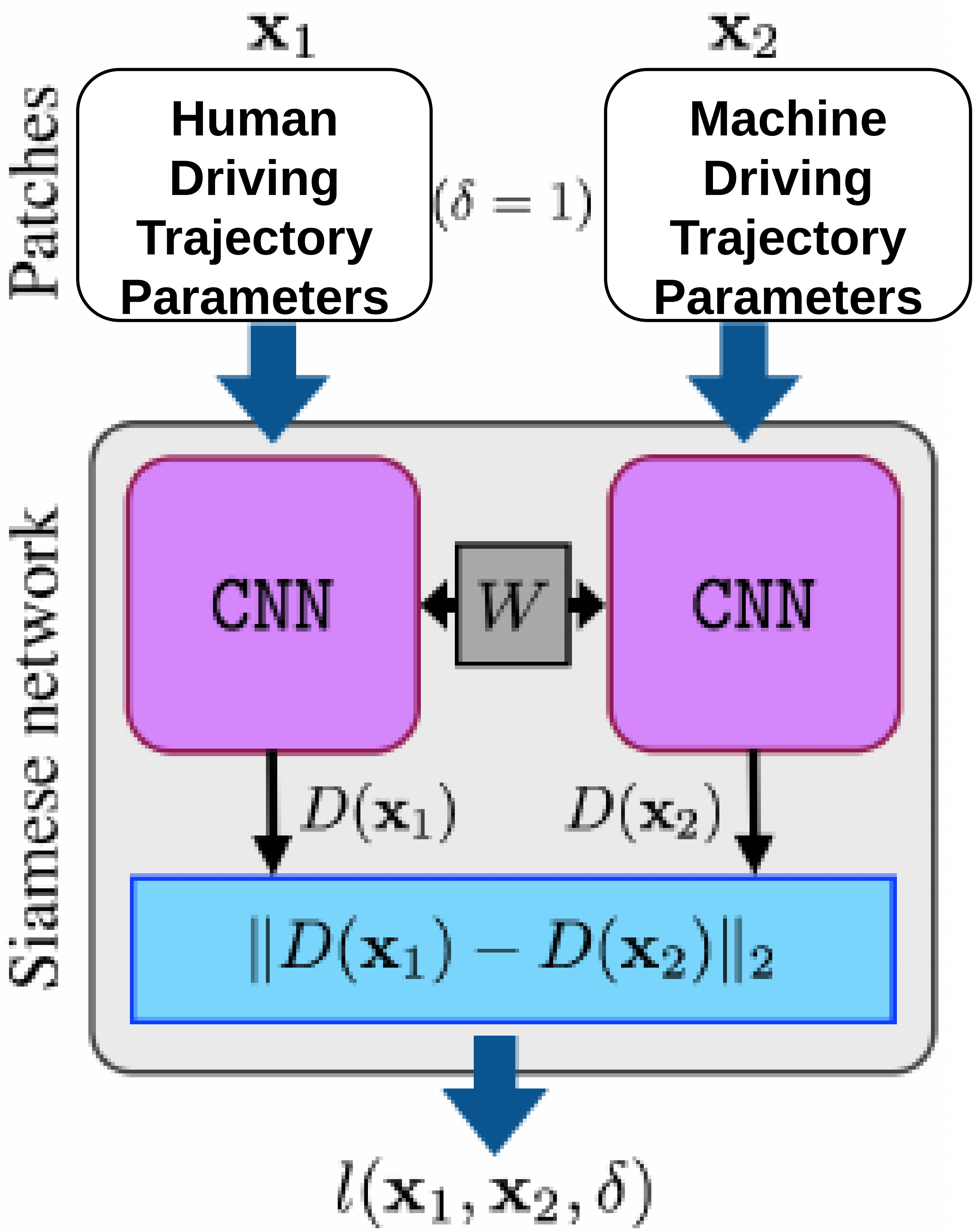
Here Machine Learning Kicks in => Automatically Tuning the Parameters To Be Like A Human Driving

- 利用人的驾驶数据来“教导”机器如何去
  - 选择终点姿态
  - 选择轨迹Cost分量的不同权重

Apollo中的Lattice Planner 自动调参Auto-Tuning

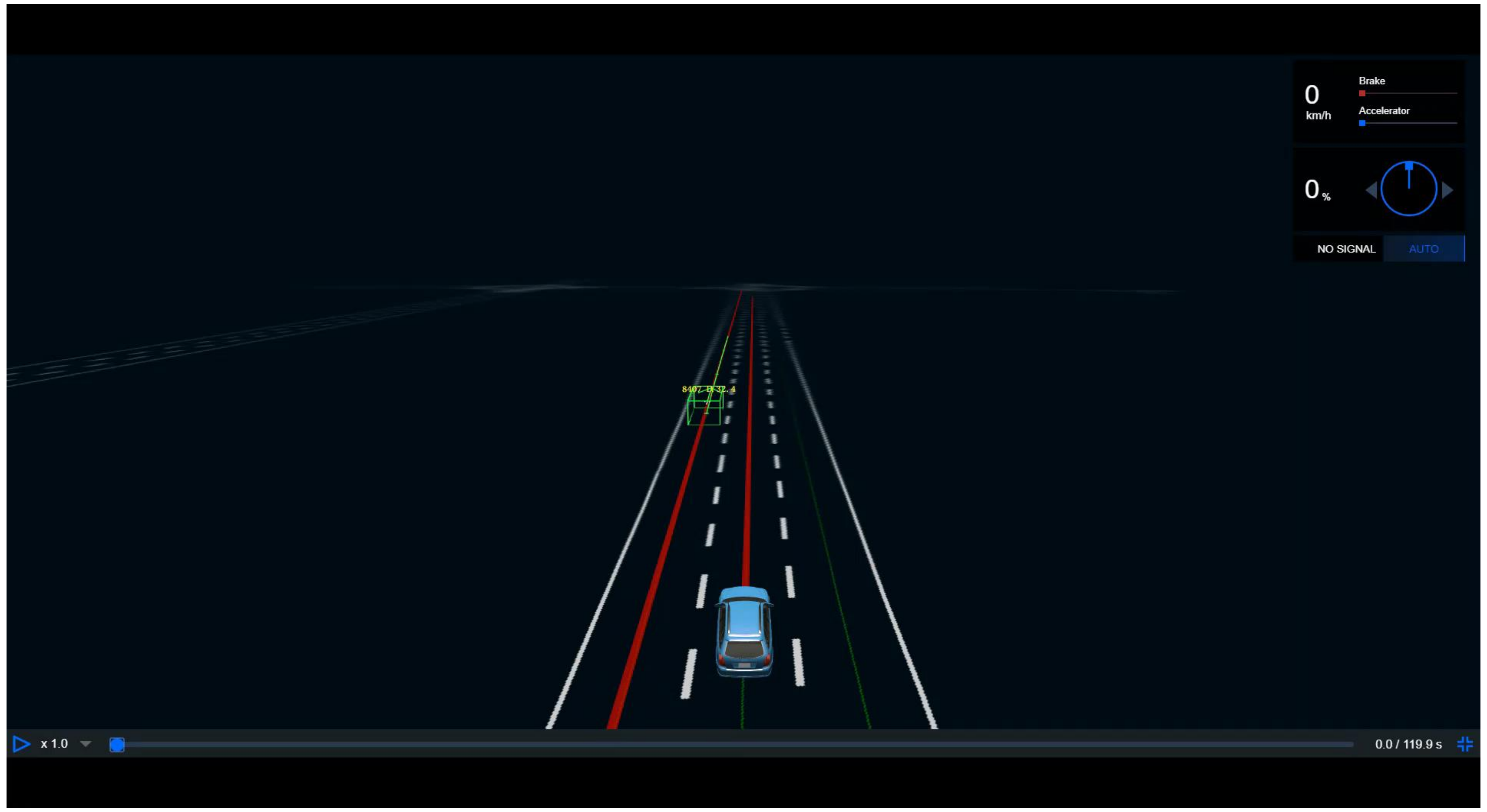
- 自动调参 Auto-Tuning
- 借鉴人脸识别的经验 Idea from Face Recognition
- 目标Goal: 使得无人驾驶车的决策和规划越来越趋近于人类的驾驶模式 To make the machine drives like a human
- Siamese 网络的两种输入模式:
  - 人和机器的不同Cost分量值 Human & Machine Driving Cost Components : $\text{cost}_{i_{Machine}}$ v.s. $\text{cost}_{i_{Human}}$
  - 人和机器的驾驶轨迹中提取的Feature: 例如轨迹点的形状和周边信息 : $Trajectory_{Machine}\ v.s.\ Trajectory_{Human}$

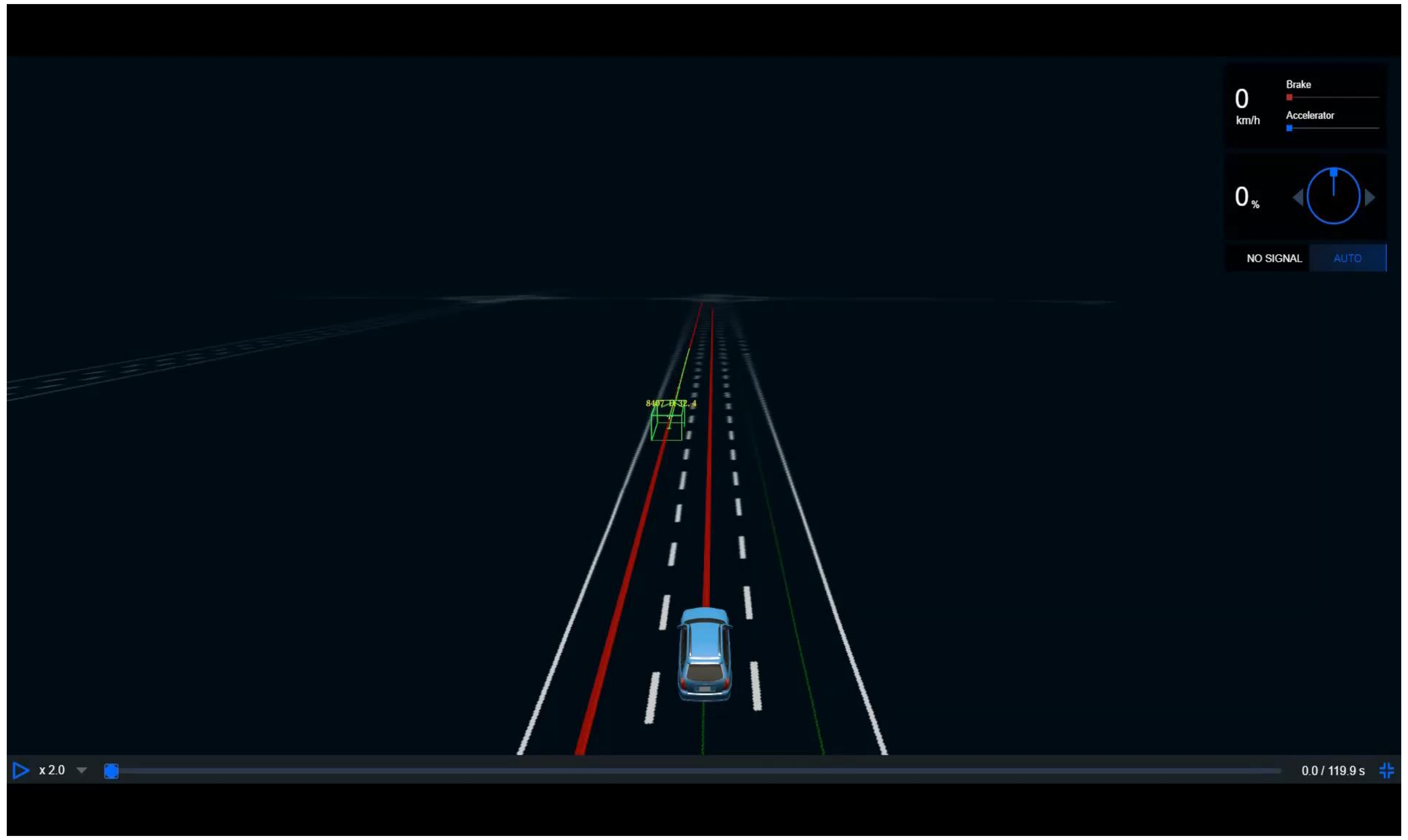



$\mathbf{x}_1$
$\mathbf{x}_2$
Patches
Human Driving Trajectory Parameters
$(\delta = 1)$
Machine Driving Trajectory Parameters
Siamese network
CNN
W
CNN
$D(\mathbf{x}_1)$
$D(\mathbf{x}_2)$
$\|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2$
$l(\mathbf{x}_1, \mathbf{x}_2, \delta)$

Simple Effect of Auto Tuning

- Before Auto Tuning the Cost Map for ST-Graph
- Vehicle follows on lead vehicle on desired lane. The lead vehicle stops and autonomous vehicle stops too
- After auto tuning with human drive data, the autonomous vehicle will overtake the stopped vehicle using the other lane





Topic

- Autonomous Driving Software System Modules

- Deep Learning in Autonomous Driving System Modules
  - Perception
  - Prediction
  - Decision & Planning

- **Future Directions for AI in Autonomous Driving**

Future Directions for AI in Autonomous Driving

- Deeping learning in perception is getting more and more mature.
  - Fusion based approach is becoming more and more mainstream.
  - Hardware based pre-fusion and joint training . Eg RoadStar.ai and Hesai's Pandora

- The importance of irregular behavior prediction is still a challenge

- Decision & Planning starts to be more and more "data-driven"
  - AI in planning will be more of a Bottom-Up process rather than top-down

Thank You

Q & A

Artificial
Intelligence

主办

O'REILLY

intel AI