

【自动驾驶】运动规划 | 论文分享 | Baidu Apollo EM Motion Planner

Haoyang Fan^{1,†}, Fan Zhu^{2,†}, Changchun Liu, Liangliang Zhang, Li Zhuang,
Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, Qi Kong^{3,*}

Abstract—In this manuscript, we introduce a real-time motion planning system based on the Baidu Apollo (open source) autonomous driving platform. The developed system aims to address the industrial level-4 motion planning problem while considering safety, comfort and scalability. The system covers multilane and single-lane autonomous driving in a hierarchical manner: (1) The top layer of the system is a multilane strategy that handles lane-change scenarios by comparing lane-level trajectories computed in parallel. (2) Inside the lane-level trajectory generator, it iteratively solves path and speed optimization based on a Frenet frame. (3) For path and speed optimization, a combination of dynamic programming and spline-based quadratic programming is proposed to construct a scalable and easy-to-tune framework to handle traffic rules, obstacle decisions and smoothness simultaneously. The planner is scalable to both highway and lower-speed city driving scenarios. We also demonstrate the algorithm through scenario illustrations and on-road test results.

The system described in this manuscript has been deployed to dozens of Baidu Apollo autonomous driving vehicles since Apollo v1.5 was announced in September 2017. As of May 16th, 2018, the system has been tested under 3,380 hours and approximately 68,000 kilometers (42,253 miles) of closed-loop autonomous driving under various urban scenarios.

The algorithm described in this manuscript is available at <https://github.com/ApolloAuto/apollo/tree/master/modules/planning>.

increase traffic efficiency and provide convenient travel. However, autonomous driving is a challenging task that requires accurately sensing the environment, a deep understanding of vehicle intentions and safe driving under different scenarios. To address these difficulties, we constructed an Apollo open source autonomous driving platform. The flexible modularized architecture of the developed platform supports fully autonomous driving deployment <https://github.com/ApolloAuto/apollo>.

In the figure, the HD map module provides a high-definition map that can be accessed by every on-line module. Perception and localization modules provide the necessary dynamic environment information, which can be further used to predict future environment status in the prediction module. The motion planning module considers all information to generate a safe and smooth trajectory to feed into the vehicle control module.

In motion planner, **safety** is always the top priority. We consider autonomous driving safety in, but not limited to, the following aspects: traffic regulations, range coverage, cycle time efficiency and emergency safety. All these aspects are critical. **Traffic regulations** are designed by governments for public transportation safety, and such regulations

百度Apollo 大佬 樊昊阳的运动规划论文

I. INTRODUCTION

图1中高清地图模块每个在线模块都可以访问，感知和定位模块提供了必要的动态环境信息，在预测模块中进一步利用这些信息预测未来的环境状态。运动规划模块考虑所有信息，生成安全、平滑的轨迹，并输入车辆控制模块。

Apollo Realtime Module Pipeline

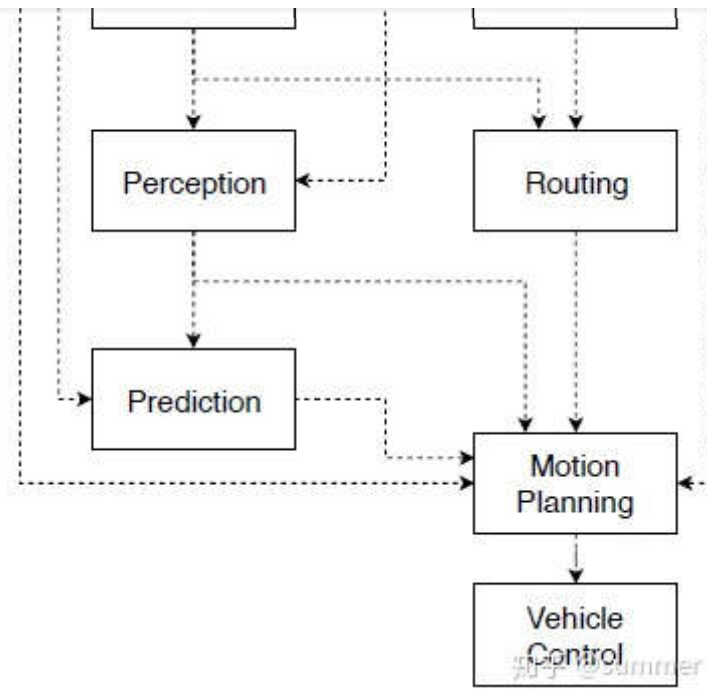


Fig. 1: On board modules of the Apollo open source autonomous driving platform

在运动规划中，安全总是最重要的。我们认为自动驾驶的安全性主要体现在以下几个方面:交通规则、里程覆盖、周期效率和应急安全。所有这些方面都至关重要。交通规则是政府为了公共交通安全而制定的，这些规则也适用于自动驾驶车辆。自动驾驶车辆应始终遵守交通规则。在覆盖范围内，我们的目标提供**至少具有8s或200m的运动规划轨迹**。这样做的原因是为了自动驾驶车辆能留下足够的空间来保证安全驾驶。运动规划算法的执行时间也很重要。在紧急情况下，系统可以在100毫秒内做出反应，而普通人的反应时间为300毫秒。安全应急模块是保护乘客安全的最后一道防护。对于L4级的运动规划，一旦上游模块不能正常运行，运动规划器内的安全模块将立即响应紧急行为并发送警告提醒驾驶员。此外，自动驾驶系统还应具备低层控制模块的应急反应能力。进一步的安全设计超出了本文范围。

除了安全之外，乘客的乘车体验也很重要。驾驶体验的测量包括但不限于场景覆盖、交通规则和舒适度。对于场景覆盖，运动规划不仅应该能够处理简单的驾驶场景(例如，停车，跟车、和超车)，还应该处理多车道驾驶、交通拥挤和其他复杂的道路驾驶场景。在交通规则范围内进行规划对驾驶体验也很重要。遵守交通规则除了是一项安全要求外，对于自动驾驶来说，遵守交通规则

提供的自主与规划层的工作又不重叠的。本文提出了Apollo EM Planner Planner，这是基于一个EM型迭代算法。本计划的目标是安全和乘车体验采用多车道、路径速度迭代、交通规则和决策的组合设计。

II. EM PLANNER FRAMEWORK WITH MULTILANE STRATEGY

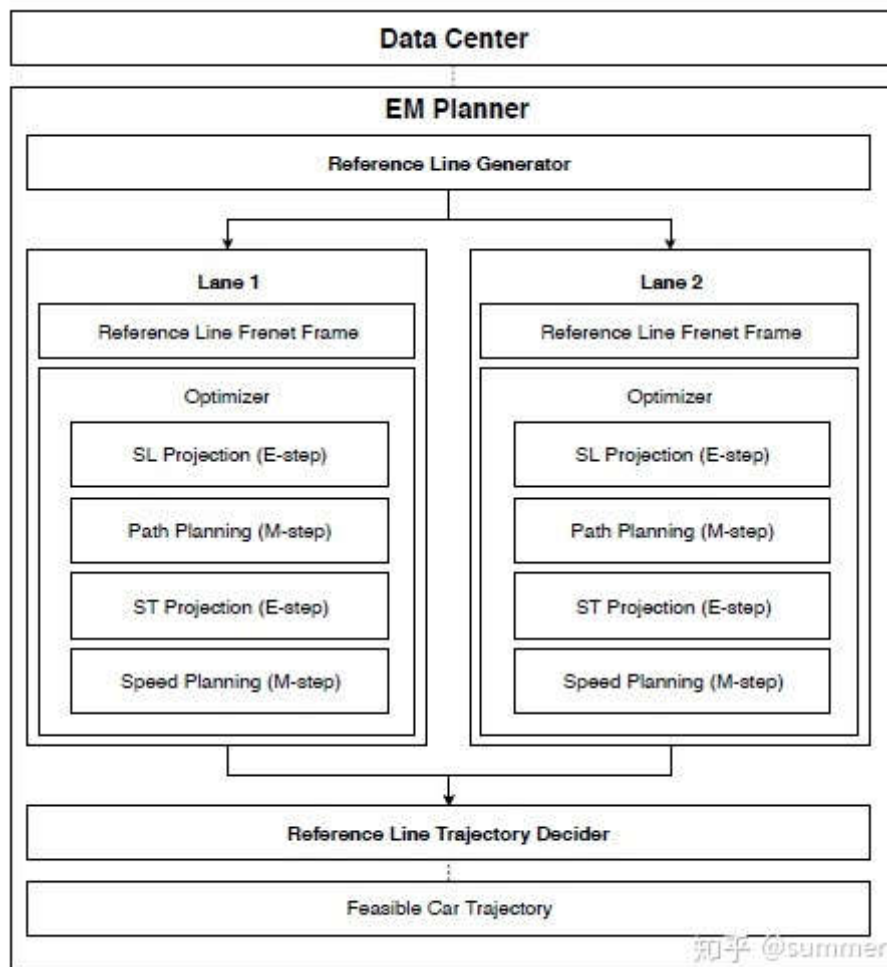


Fig. 2: EM Framework

III. EM PLANNER AT LANE LEVEL

图3为车道级别的速度EM算法。在一个规划周期中包括两个E-step和两个M-step。

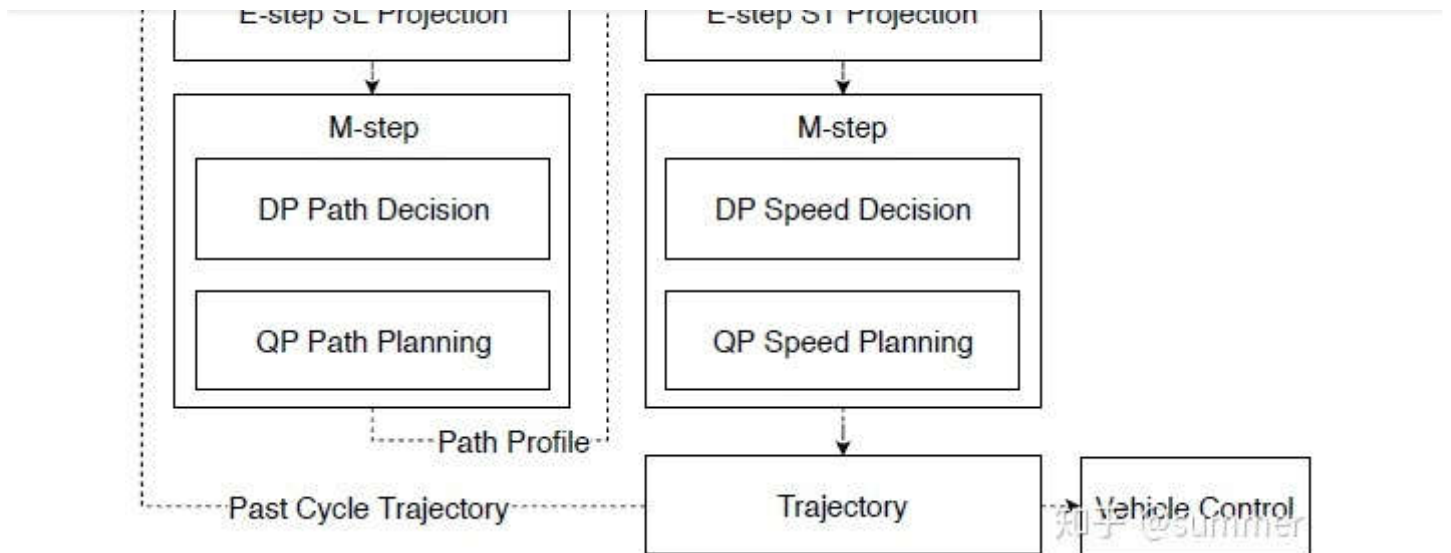


Fig. 3: EM Iteration

在第一个E-step中，将障碍物投影Lane Frenet上。该投影包括静态障碍物投影和动态障碍物投影。静态障碍物将直接基于笛卡尔-Frenet坐标系变换进行投影。在Apollo 架构中，动态障碍物的意图是通过障碍物运动轨迹来描述的。考虑到上一个周期规划轨迹，评估动态障碍物和本车在每个时间点的位置。然后，将动态障碍物与本车在每个时间点的重叠映射到Frenet坐标系中。此外，路径优化过程中出现的动态障碍最终会导致轻推。因此，出于安全考虑，动态障碍物的SL投影只考虑低速的障碍物。对于高速的障碍物，EM Planner的平行变道策略将涵盖这种情况。在第二步的E-step中，基于生成的路径曲线，对高速、低速的障碍物进行评估。如果障碍物轨迹与规划路径有重叠，则生成相应占位区域。

采用动态规划和二次规划相结合的方法，在两个E-step内生成路径和速度曲线。虽然我们在SL和ST上投影障碍物，但最优路径和速度的解仍然是非凸空间问题。因此，我们使用动态规划首先得到一个粗糙解；同时，该解决方案还可以提供跟随、避让和超车等决策。

A. SL and ST Mapping (E-step)

SL投影是基于G2(连续曲率导数)光滑参考线。在笛卡尔空间中，障碍物和本车的状态用位置和航向来描述(x;y), 航向角以及曲率和曲率的导数。然后，这

的导致。由于静态障碍物的位置是固定的，因此障碍物是且仅向运动初始速度方向运动的。对于动态障碍物，我们利用本车的最后一个周期轨迹来绘制障碍物。将上一周期的运动轨迹投影到Frenet坐标上，提取出方向速度，并估计本车在特定时间与动态障碍物的关系。一旦本车的坐标与障碍物轨迹点同时存在交集，SL空间上的阴影区域将被标记为与动态障碍物的碰撞区域。这里，交集被定义为本车与障碍物边界的重叠。

例如，如图4所示，用红色标记预测模块估计的迎面而来的动态障碍物及其轨迹，本车是用蓝色标记的。首先将迎面运动障碍的轨迹离散为几个随时间变化的轨迹点，然后将这些轨迹点投影到Frenet坐标系中。一旦我们发现本车的坐标与投射的障碍物点有交集，重叠区域(图中紫色部分)将会在Frenet坐标系中被标记出来，ST投影帮助我们评估本车的速度曲线。当路径优化模块在Frenet坐标系中生成一个平滑的路径曲线后，如果有任何交集，静态障碍物和动态障碍物轨迹都会被投影到路径上。

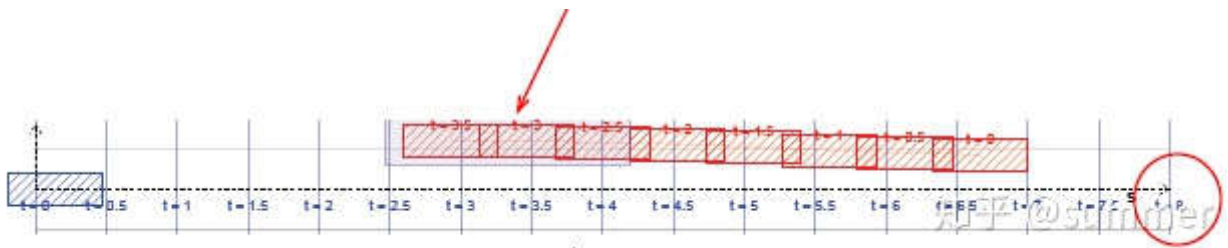


Fig. 4: SL projection with oncoming traffic example

在图5中，前方2秒40米处有一个障碍物切入ego driving path，用红色标记，后面有一个障碍物用绿色标记，剩余区域为速度可行区域。速度优化M-step试图在该区域找到可行的最优解。

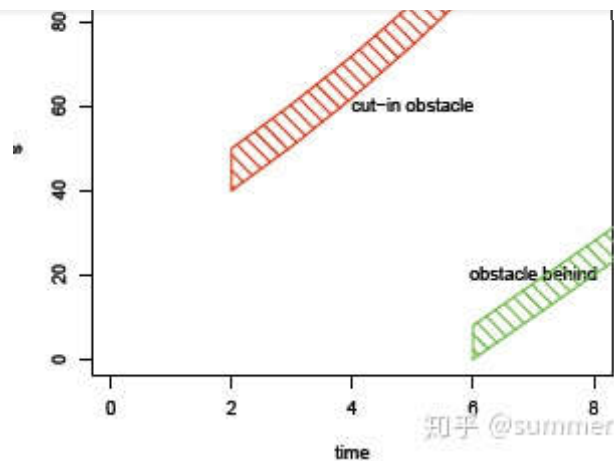


Fig. 5: ST projection with cut-in obstacle and obstacle behind ego car

B. M-Step DP Path

M-step路径优化模块优化了Frenet坐标中的路径配置文件，这表示为在非凸SL空间中寻找横向坐标 $l = f(s)$ 的一个最优函数解(例如，nudging from left and right可能是两个局部最优解)。因此，路径优化模块包含2个步骤：基于动态规划的路径决策和基于样条曲线的路径规划。动态规划路径步骤给出了一个粗糙的路径轮廓线，并给出了可行的路径。

如图6所示，该步骤包括Lattice采样器、Cost代价函数和DP动态规划搜索算法。该Lattice采样器是基于Frenet坐标的。

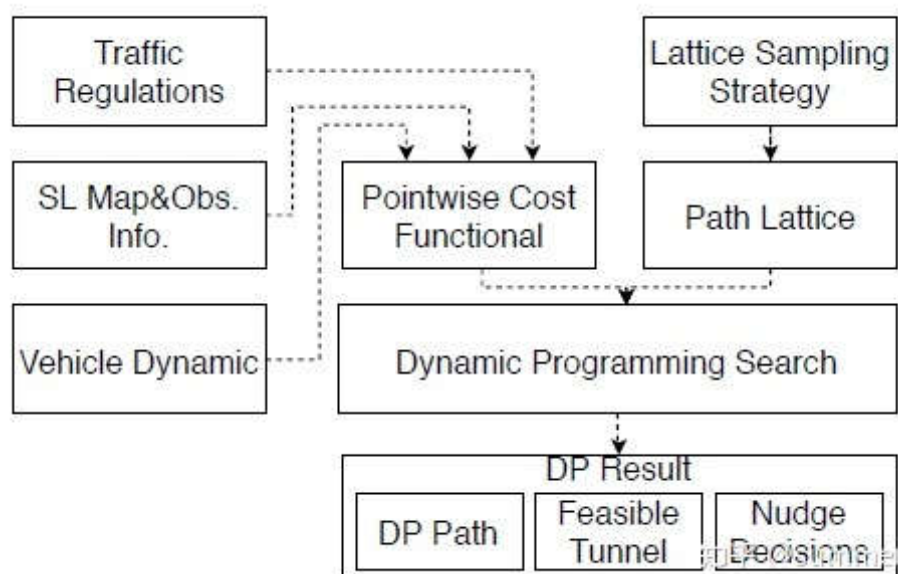


Fig. 6: Dynamic programming structure



用建模。采样之间的时间间隔取决于速度、道路几何、又遵守。根据应用的不同，场景定制采样策略。例如，变道可能需要比当前车道驾驶更长的采样间隔。此外，lattice将覆盖至少8秒或200米的安全因素。在构造lattice之后，每条图的边都由代价泛函的总和来计算。我们使用来自SL投影、交通规则和车辆动力学的信息来构造函数。总的代价函数是平滑性、避障性和车道代价函数的线性组合。

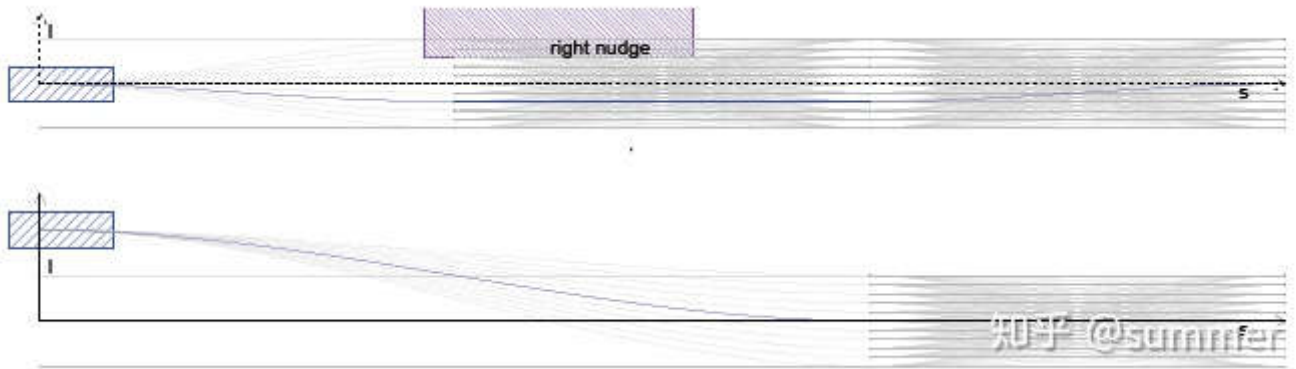


Fig. 7: Dynamic programming path optimizer sampling for default lane and change lane

$$C_{total}(f(s)) = C_{smooth}(f) + C_{obs}(f) + C_{guidance}(f)$$

1、对于给定路径的平滑函数：

$$C_{smooth}(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds + w_3 \int (f'''(s))^2 ds.$$

平滑度代价函数中， $f'(s)$ 表示车道与ego car的航向差， $f''(s)$ 与路径曲率有关， $f'''(s)$ 与ego car曲率的导数有关。

2、障碍物成本函数：

以障碍物与本车之间的边界框距离为基础，距离为d：



$$C_{obs}(d) = \begin{cases} C_{nudge}(d - d_c), & d_c \leq d \leq d_n, \\ C_{collision}, & d \leq d_{collision} \end{cases}$$

单调递减函数

3、车道保持函数：

$$C_{guidance}(f) = \int (f(s) - g(s))^2 ds$$

如果道路外的路径点将会受到很高的惩罚。

最终的路径成本是所有这些的组合平滑、障碍物和车道偏移的成本。然后，通过动态规划搜索选择一个最低的候选路径成本。

C. M-Step Spline QP Path

在动态规划路径中，根据所选路径生成可行区间。然后，基于样条的QP步骤将在该可行区间内生成平滑的路径，如图8所示。

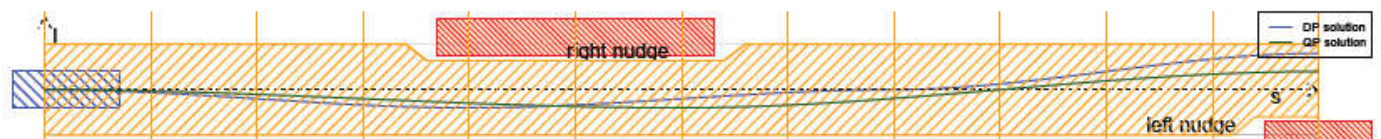
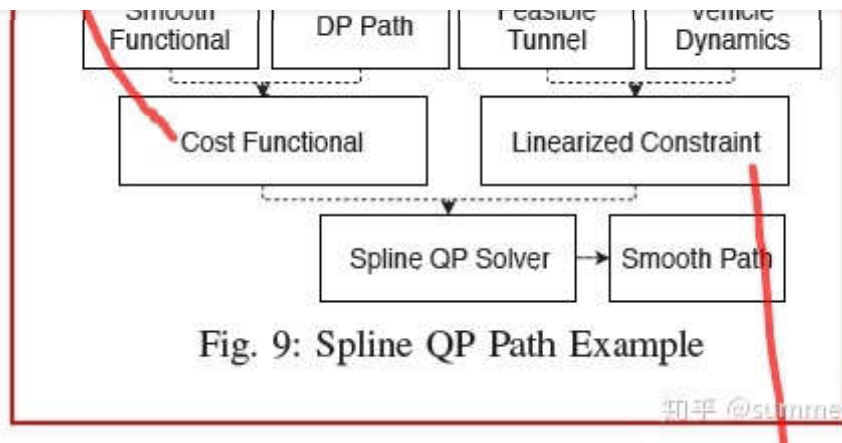


Fig. 8: Spline QP Path Example

通过QP样条求解器优化线性化约束的目标函数，生成样条QP路径。图9显示了QP路径步骤。



QP路径的目标函数是平滑性成本和指引线成本的线性组合。这一步中的指引线是DP路径。数学上，QP Path步骤优化了以下函数：

$$C_s(f) = w_1 \int (f'(s))^2 ds + w_2 \int (f''(s))^2 ds + w_3 \int (f'''(s))^2 ds + w_4 \int (f(s) - g(s))^2 ds.$$

目标函数中， $f'(s)$ 表示车道与ego car的航向差， $f''(s)$ 与路径曲率有关， $f'''(s)$ 与ego car曲率的导数有关。而 $f(s)-g(s)$ 表示与指引线之间的距离偏差。

【百度自动驾驶技术系列课程】自动驾驶核心模块（运动规划） 精讲

<https://www.bilibili.com/video/BV1p54y1S7Gb?p=12>

www.bilibili.com



<https://www.bilibili.com/video/BV1p54y1S7Gb?p=14>

www.bilibili.com



【百度自动驾驶技术系列课程】自动驾驶核心模块精讲_哔哩哔哩 (゜-゜)...

www.bilibili.com



<https://www.bilibili.com/video/BV1p54y1S7Gb?p=16>

www.bilibili.com



【百度自动驾驶技术系列课程】自动驾驶核心模块精讲_哔哩哔哩 (゜-゜)つロ 干...

www.bilibili.com



【百度自动驾驶技术系列课程】自动驾驶核心模块精讲_哔哩哔哩 (゜-゜)つロ 干...

www.bilibili.com

