

时空联合规划

丁文超





大纲



- 🔷 时空联合规划的概念
- 🔷 基于搜索的规划方法
- 🔷 基于迭代计算的规划方法
- 🔷 基于时空走廊的规划方法
- 🔷 应用案例展示



时空联合规划的概念



□ 思考

怎样解决窄道会车问题？

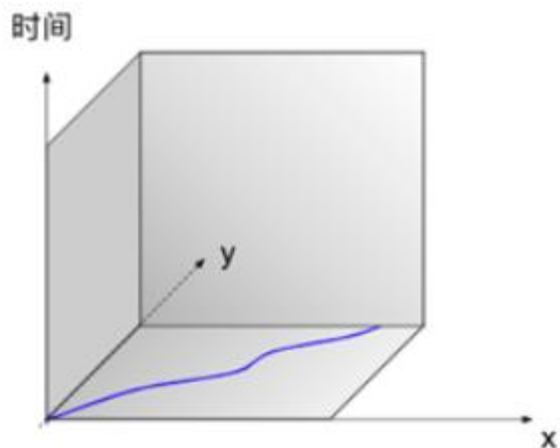




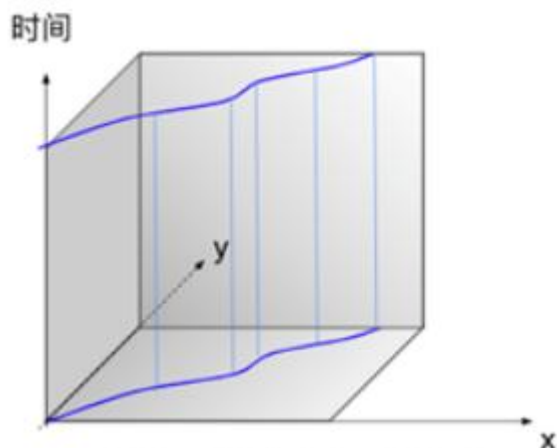
时空联合规划的概念



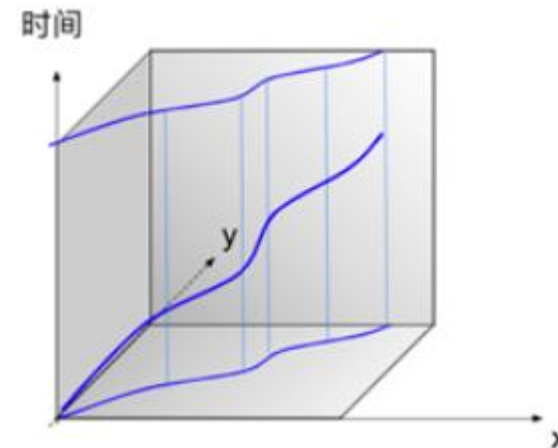
□ 时空分离算法



第一步：先在x-y平面求解最佳行车路线



第二步：根据路径计算行车速度的曲面



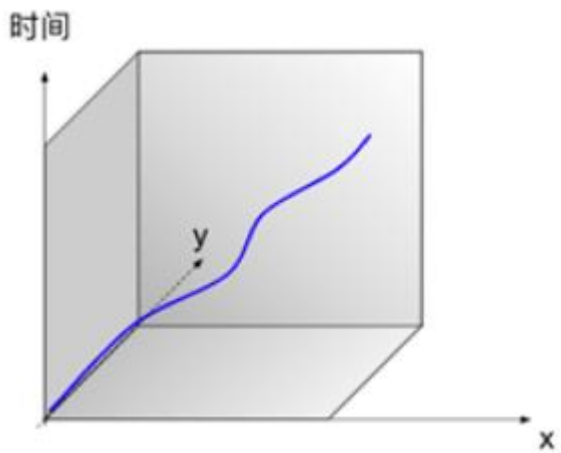
第三步：计算表面上的最佳速度，获得轨迹



时空联合规划的概念



□ 时空联合算法



时空联合：在三维空间内直接
计算最佳轨迹

时空分离算法	时空联合规划
分两步依次计算两个二维空间的解， 每一步损失一个维度的解空间	直接在三维空间内计算最优解， 求解空间大一个维度
求解速度相对更快，算力要求低	充分考虑动态障碍物信息， 路径规划更合理
复杂动态场景下容易陷入轨迹次优问题	符合人类驾驶习惯， 适合直接学习人类司机行为



大纲

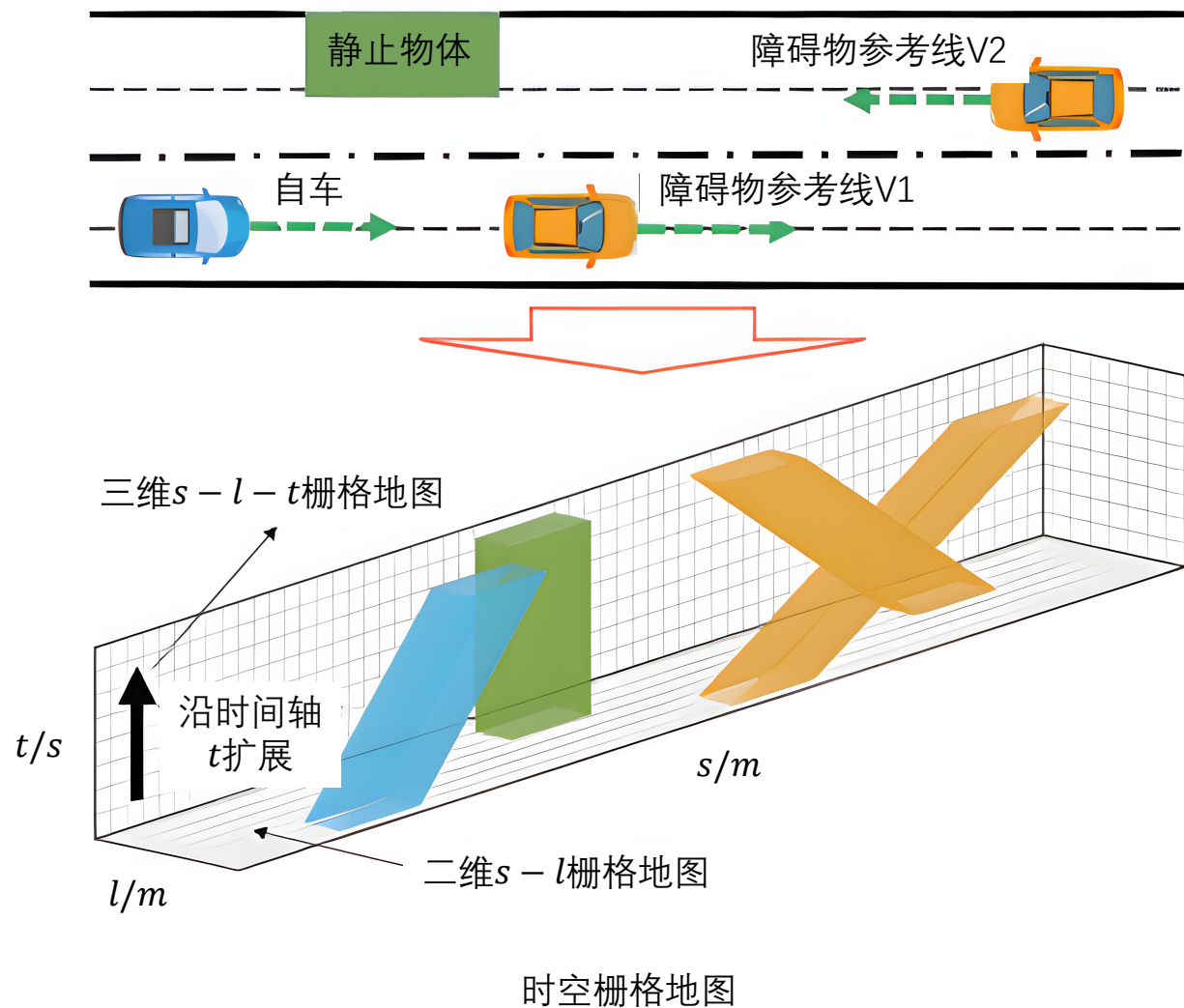


- ◈ 时空联合规划的概念
- ◈ 基于搜索的规划方法
- ◈ 基于迭代计算的规划方法
- ◈ 基于时空走廊的规划方法
- ◈ 应用案例展示



基于搜索的路径规划

基于Hybrid A*的时空联合规划





基于搜索的路径规划

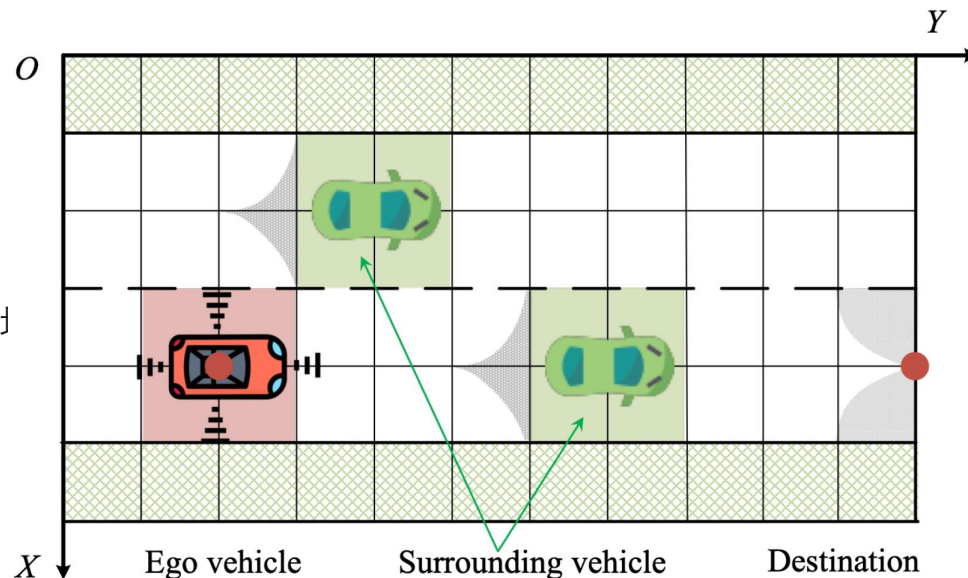


构建三维时空联合规划地图

二维x-y栅格地图只具有几何属性，无法直接搜索带有时间属性的可行驶轨迹。

- 建立2D占据栅格地图；
- 为了提高搜索效率，消除一部分无用栅格块。

(灰色圆形区域部分：在最大转向角下也会发生碰撞 或 不可能到达目的)
半径大小：最小转弯半径；方向：车速方向相切)



2D spatial grid Map for a two-lane road



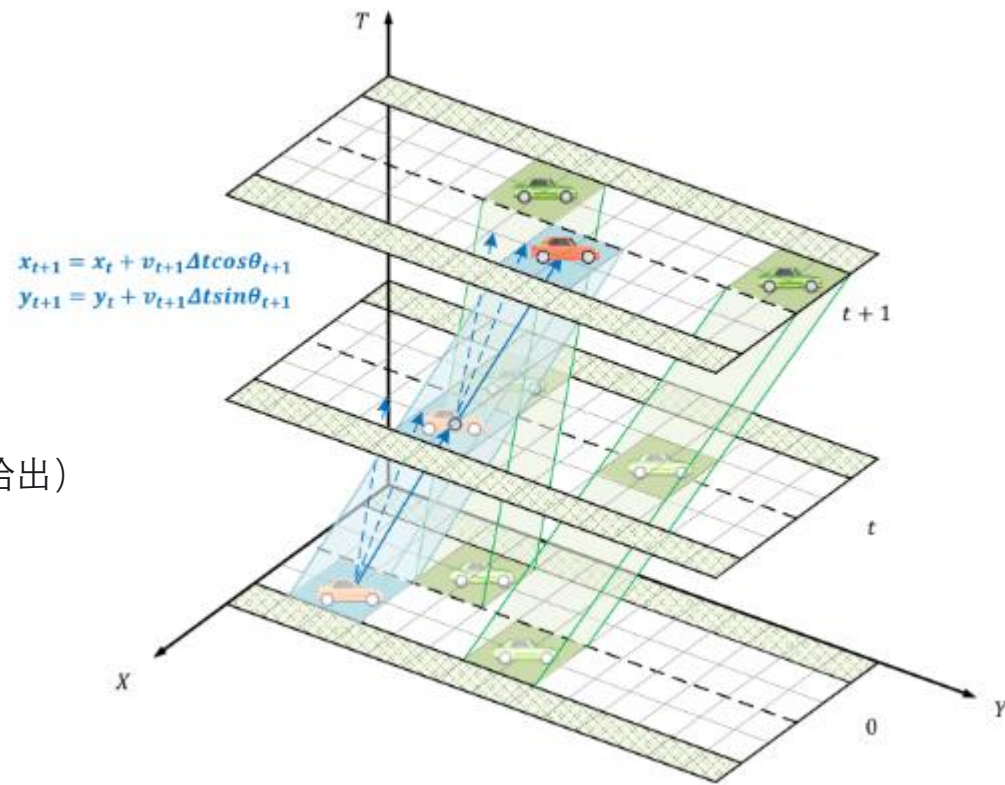
基于搜索的路径规划



构建三维时空联合规划地图

二维栅格地图沿时间轴扩展生成三维时空地图

- 多个地图层相互平行
- 相邻图层中的两个状态根据时间步长由有向边相连
- 蓝色有向边连接的点序列：以 Δt 为时间步长离散化的时空轨迹
- 浅绿色部分：动态障碍物在 Δt 为时间步长离散化的位置（由轨迹预测模块给出）



3D spatio-temporal Map constructed by DAG



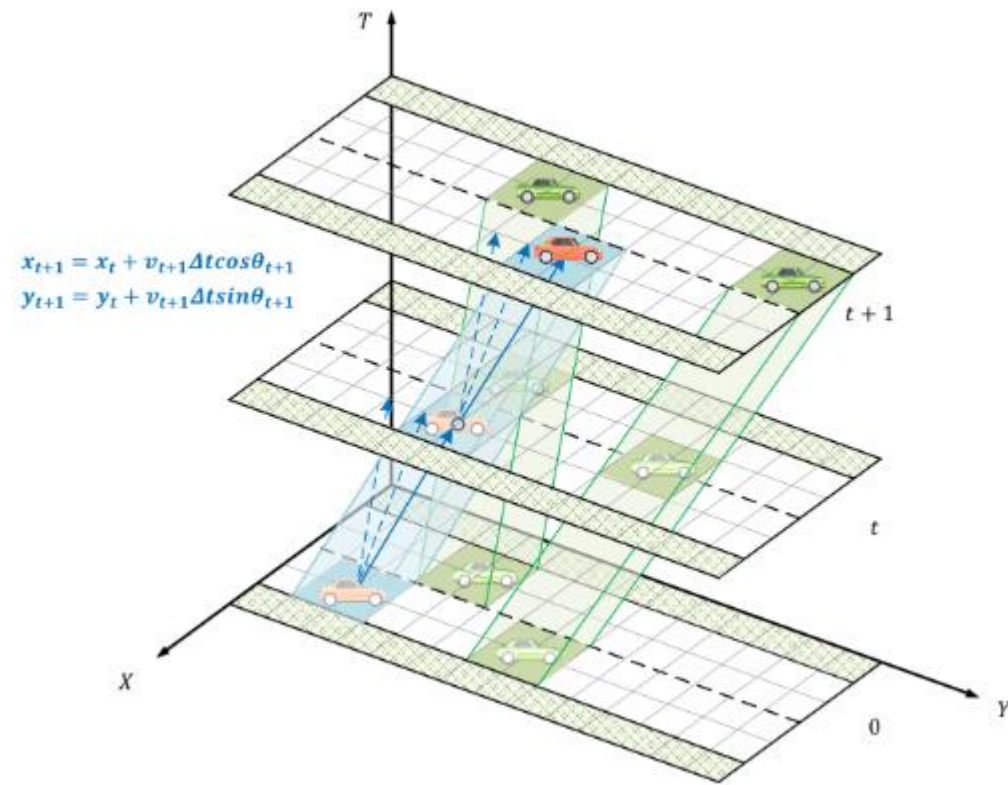
基于搜索的路径规划



构建三维时空联合规划地图

三维时空地图包含的信息

- 车辆位置状态信息
- 车辆运动学信息：
 - 蓝色有向线段在 $X-O-T$ 平面的投影斜率为横向速度
 - 蓝色有向线段在 $Y-O-T$ 平面的投影斜率为纵向速度
 - 蓝色有向线段在 $X-O-Y$ 平面的投影倾斜角为偏航角
- 根据相邻状态可以计算动作空间
- 根据动作空间扩展时空地图（图中由蓝色实线、虚线所示）



3D spatio-temporal Map constructed by DAG



基于搜索的时空联合规划



基于Hybrid A*的时空节点扩展

- 离散化前轮转角集合 Δ :

$$\Delta = \left\{ \delta_i \mid -\delta_{max} + (i-1) \cdot \frac{\delta_{max} - (-\delta_{max})}{N-1} \right\}$$

δ_i 为离散后的前轮转角

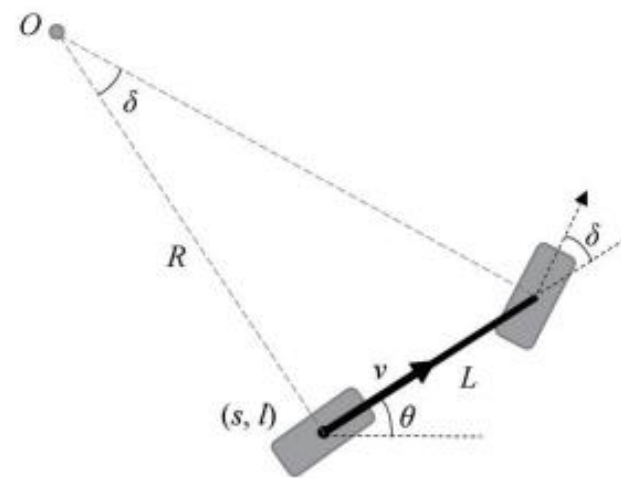
- 离散加速度集合A:

$$A = \{-a_{max}, \dots, 0, \dots, a_{max}\}$$

- 状态更新方程:

$$\begin{cases} s_i = s_p + v_p \cdot dt \cdot \cos \theta_i \\ l_i = l_p + v_p \cdot dt \cdot \sin \theta_i \\ \theta_i = \theta_p + v_i \cdot dt \cdot \tan \delta_i / L \\ v_i = v_p + a_i \cdot dt \\ t_i = t_p + dt \end{cases}$$

dt 为相邻节点时间步长



车辆运动学模型



基于搜索的时空联合规划



□ Hybrid A* :时空节点启发式函数设计

$$H_i = w_{H1} \cdot \sqrt{(s_{\text{goal}} - s_i)^2 + (l_{\text{goal}} - l_i)^2} + w_{H2} \cdot (t_{\text{goal}} - t_i)$$

欧氏距离启发项

时间距离启发项

s_{goal} 、 l_{goal} 分别是目标节点的纵向、横向位置；

t_{goal} 为目标节点对应的相对时间，避免陷入时间停滞，加快搜索过程；

w_{H1} 、 w_{H2} 分别为对应项的权重。



基于搜索的时空联合规划



□ Hybrid A* :时空节点成本函数设计

$$G_i = G_p + w_{g1} \cdot E_i + w_{g2} \cdot C_i + w_{g3} \cdot S_i$$

$$C_i = |l_i - l_{centerline}|$$

$$E_i = |v_i - v_{expected}|$$

$$S_i = R(s, l) \cdot \exp(\cos \theta_r)$$

$$R(s, l) = \exp \left[-\frac{1}{2} \times \left(\frac{(s - u_s)^2}{\sigma_{sg}^2} + \frac{(l - u_l)^2}{\sigma_{lg}^2} \right) \right]$$

$$r = (s - u_s, l - u_l)$$

G_p 为父节点成本

E_i 为子节点目标代价

C_i 为子节点舒适性代价

S_i 为子节点安全性代价

w_{g1} 、 w_{g2} 、 w_{g3} 为权重系数

(s, l) 、 (u_l, u_s) 分别表示节点和障碍物的位置;

σ_{sg} 、 σ_{lg} 分别表示障碍物在 s 向、 l 向的权重系数;

r 为节点相对于障碍物的位置向量;

θ_r 为向量 r 与障碍物速度 v 之间的夹角。

通过上述成本函数的设计可以提高轨迹的拟人性



基于搜索的时空联合规划

□ Hybrid A*生成参考轨迹

- 以当前自车所在位置($s_1, l_1, \theta_1, v_1, t_1$)为起点进行时空节点扩展;
- 扩展过程中考虑节点无碰撞和道路边界约束问题进行检查;
- 通过节点代价评估寻找最优扩展节点, 直至达到目标节点;
- 通过各个最优节点得到初始粗轨迹。



Algorithm 1: Hybrid A* Algorithm for Spatio-Temporal Trajectory Search

Input: A spatio-temporal DAG map

Output: The optimal spatio-temporal trajectory from initial position to destination

```
1 Initialize the trajectory set with the initial point;
2 Initialize the open set with the initial point;
3 Construct the closedi set at different time step according
  to the dynamic obstacles;
4 while the open set is not empty do
5   Estimate the Cost of each node in the open set with
     (2);
6   Take the node current with the lowest Cost as the
     vehicle states  $s_t$ ;
7   Add the states  $s_t$  to the trajectory set;
8   if The selected node current is the destination then
9     The trajectory is found successfully;
10  else
11    Remove the current node from the open set;
12    Generate the neighbouring nodes in the next
       map layer with (1);
13    for each neighbouring node do
14      if it is in the closedt+1 then
15        ignore it
16      else
17        if it is not in the open set then
18          Add it to the open set
19        end
20      end
21    end
22  end
23   $t=t+1$ ;
24 end
25 return
```



大纲



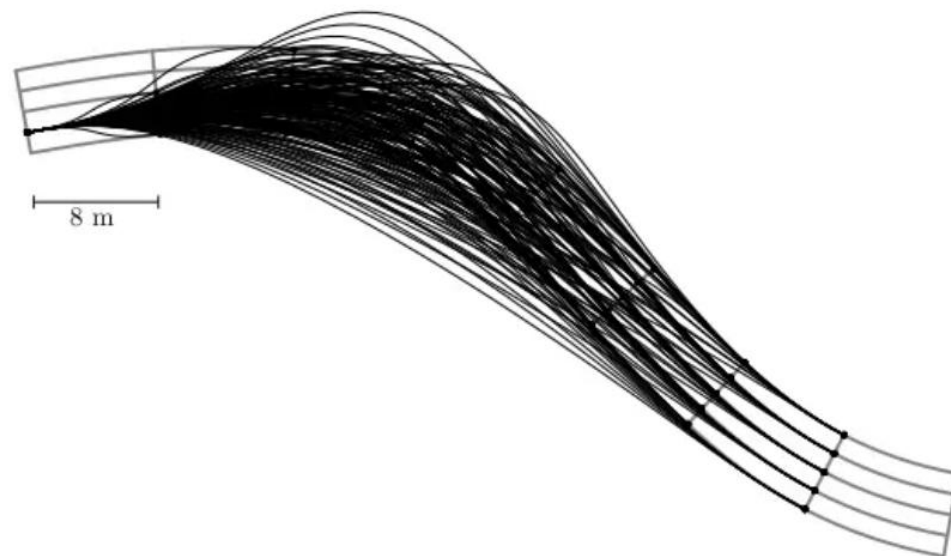
- ⬢ 时空联合规划的概念
- ⬢ 基于搜索的规划方法
- ⬢ 基于迭代计算的规划方法**
- ⬢ 基于时空走廊的规划方法
- ⬢ 应用案例展示



基于迭代计算的时空规划方法

□ 暴力搜索方法潜在问题

- 时空联合搜索的轨迹空间过于庞大；
- 减少时间和空间上的搜索复杂度，使规划聚焦于最优解可能出现的区域？



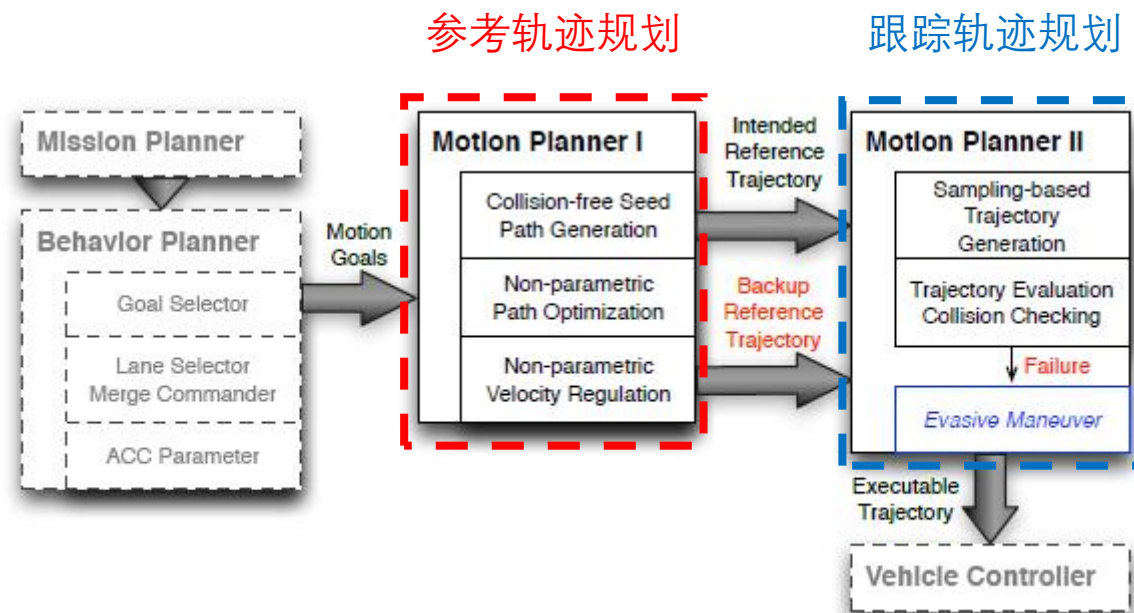


基于迭代计算的时空规划方法



□ 迭代计算方式求解流程

- Step 1 参考轨迹规划
考虑道路集合形状、障碍物，生成易于调整的类人参考时空轨迹
- Step 2 跟踪轨迹规划
选择进一步满足运动学和动力学约束的参数化轨迹





基于迭代计算的时空规划方法



种子路径生成

- 利用动态规划（DP）求解以下问题
生成无碰的最短路径

$$\min_{\{e_{n_k \rightarrow n_{k+1}}\}} \sum_{k=0}^{K-1} C(e_{n_k \rightarrow n_{k+1}})$$

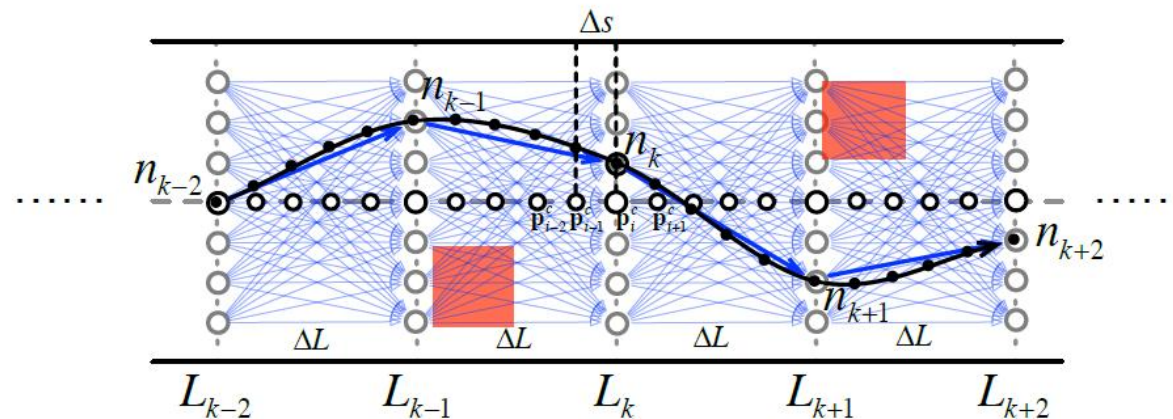
$$C(e) = w_d \cdot d(e) + (1 - w_d) \cdot o(e) + \delta_{obstacle}(e)$$

参考轨迹的规划

(考虑道路形状、障碍物)

跟踪轨迹的规划

(考虑运动学、动力学约束)





基于迭代计算的时空规划方法



非参数化路径优化

- 路径表示为一系列路径点

$$P = \{\mathbf{p}_i \mid i = 0 \dots N - 1\}$$

- 每个路径点的位置

$$\mathbf{p}_i = \mathbf{p}_i^c + (o_i + o_i^{seed}) \vec{\mathbf{n}}_i^c$$

\mathbf{p}_i^c 中心线的采样点位置, o_i 待优化的偏移量

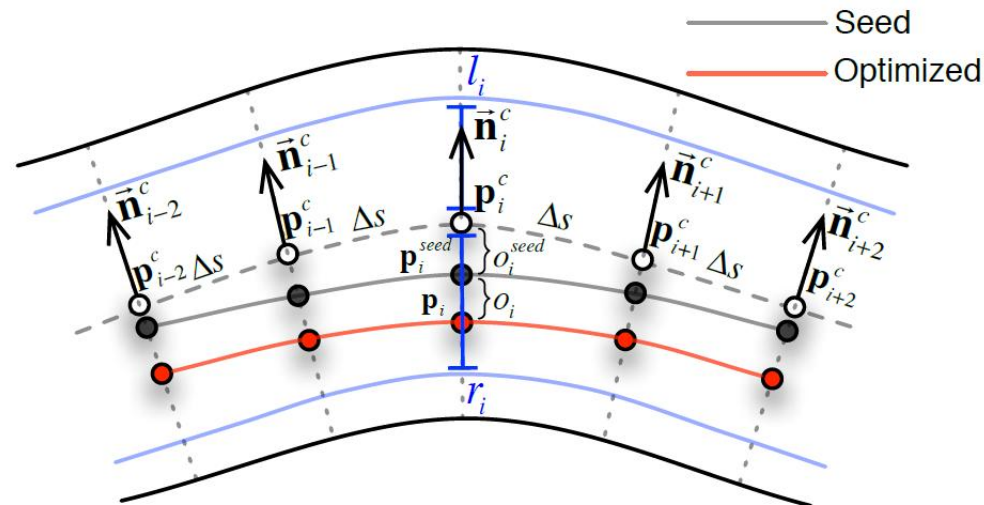
o_i^{seed} 种子路径得到的偏移量, $\vec{\mathbf{n}}_i^c$ 径向单位向量

参考轨迹的规划

(考虑道路形状、障碍物)

跟踪轨迹的规划

(考虑运动学、动力学约束)





基于迭代计算的时空规划方法



非参数化路径优化

- 每个路径点的位置

$$\mathbf{p}_i = \mathbf{p}_i^c + (o_i + o_i^{seed}) \cdot \vec{\mathbf{n}}_i^c$$

- 路径的曲率近似表示为

$$\mathbf{u}_i = \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}$$

- 优化路径

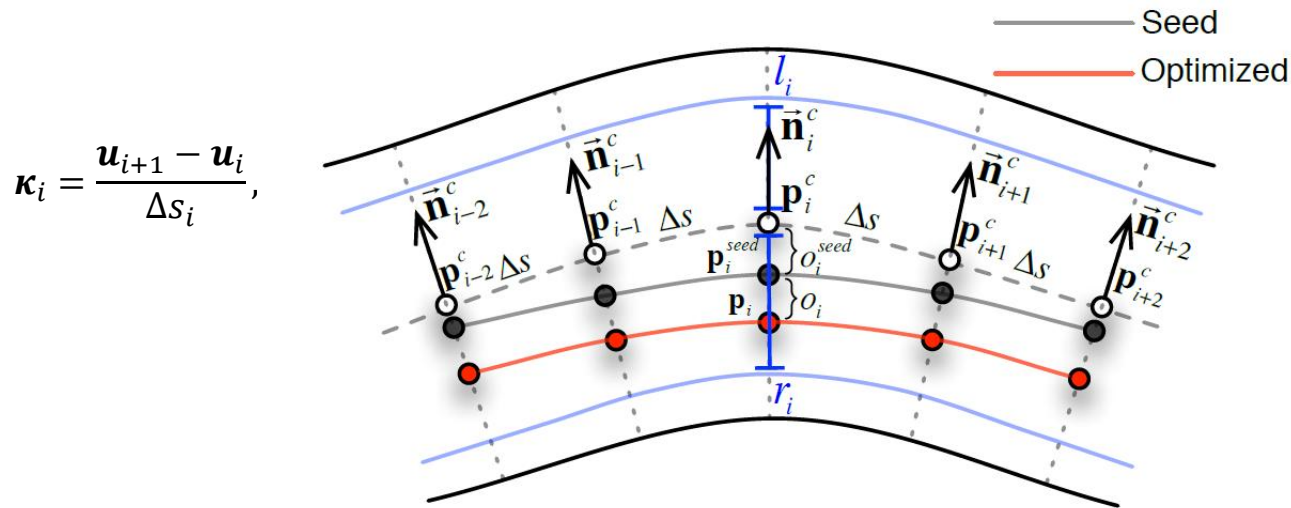
$$\min_{\{o_i\}} \left(\sum_{i=0}^{N-1} \omega_{\kappa} \cdot \|\kappa_i\| + (1 - \omega_{\kappa}) \cdot |o_i| \right)$$

参考轨迹的规划

(考虑道路形状、障碍物)

跟踪轨迹的规划

(考虑运动学、动力学约束)





基于迭代计算的时空规划方法



速度调整

- 对每个路径点赋值相应的速度 v_i^{reg} 并做出以下约束:

道路速度规定

$$v_i^{reg} \leq V_{lon}^{Limit}$$

最大向心加速度

$$|\kappa_i| \cdot (v_i^{reg})^2 \leq A_{lat}^{Limit}$$

最大纵向加速度/减速度

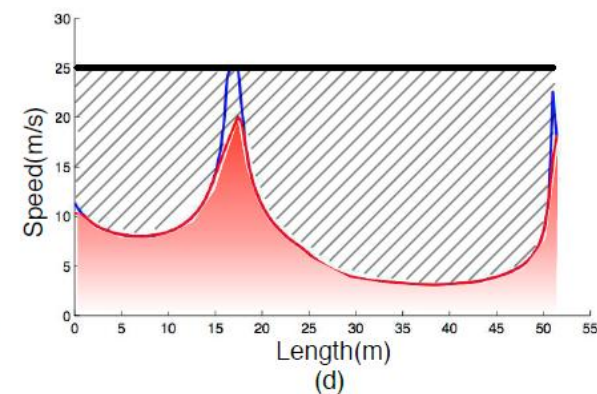
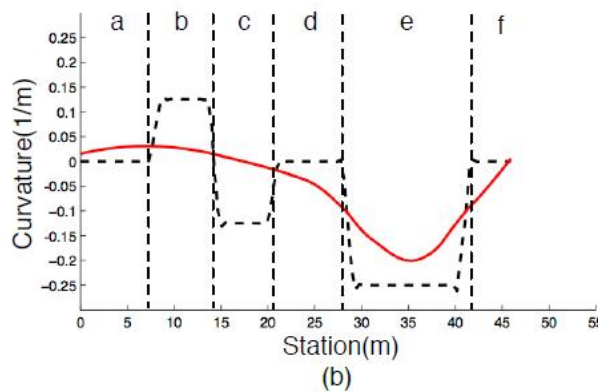
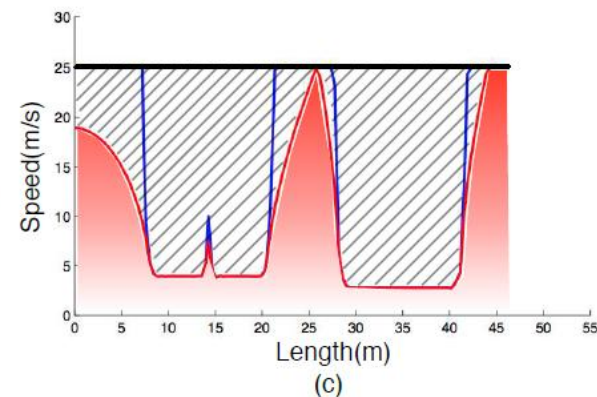
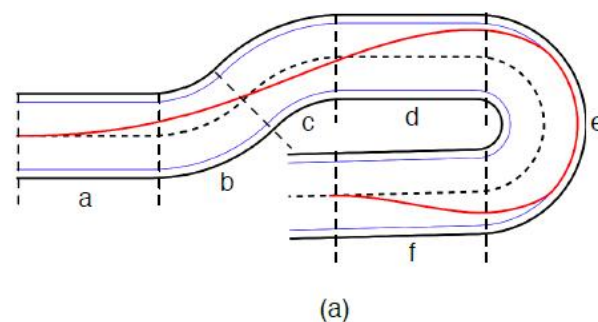
$$-D_{lon}^{Limit} \leq \dot{v}_i^{reg} \leq A_{lon}^{Limit}$$

参考轨迹的规划

(考虑道路形状、障碍物)

跟踪轨迹的规划

(考虑运动学、动力学约束)





基于迭代计算的时空规划方法



□ 参数化路径生成

- 以多项式曲率的螺旋线作为路径参数

参考轨迹的规划
(考虑道路形状、障碍物)

跟踪轨迹的规划
(考虑运动学、动力学约束)

$$\begin{aligned}x(s) &= \int_0^s \cos(\theta(s)) \cdot ds \\y(s) &= \int_0^s \sin(\theta(s)) \cdot ds \\ \theta(s) &= \int_0^s \kappa(s) \cdot ds \\ \kappa(s) &= p_0 + p_1 \cdot s + p_2 \cdot s^2 + p_3 \cdot s^3\end{aligned}$$

- 且满足边界条件 $X_0 = [x_0, y_0, \theta_0, \kappa_0]^T$, $X_f = [x_f, y_f, \theta_f, \kappa_f]^T$
- 利用打靶法^[1]求解上述边界问题



基于迭代计算的时空规划方法



□ 聚焦式轨迹参数化速度生成

- 指定速度为时间的三次多项式

$$v(t) = q_0 + q_1 t + q_2 t^2 + q_3 t^3$$

- 速度满足以下等式，已知初末状态 v_0, a_0, v_f, a_f, s_f

$$v(0) = v_0 = q_0$$

$$a(0) = a_0 = q_1$$

$$v(t_f) = v_f = q_0 + q_1 t_f + q_2 t_f^2 + q_3 t_f^3$$

$$a(t_f) = a_f = q_1 + 2q_2 t_f + 3q_3 t_f^2$$

$$s(t_f) = s_f = q_0 t_f + \frac{1}{2} q_1 t_f^2 + \frac{1}{3} q_2 t_f^3 + \frac{1}{4} q_3 t_f^4$$

参考轨迹的规划

(考虑道路形状、障碍物)

跟踪轨迹的规划

(考虑运动学、动力学约束)



基于迭代计算的时空规划方法

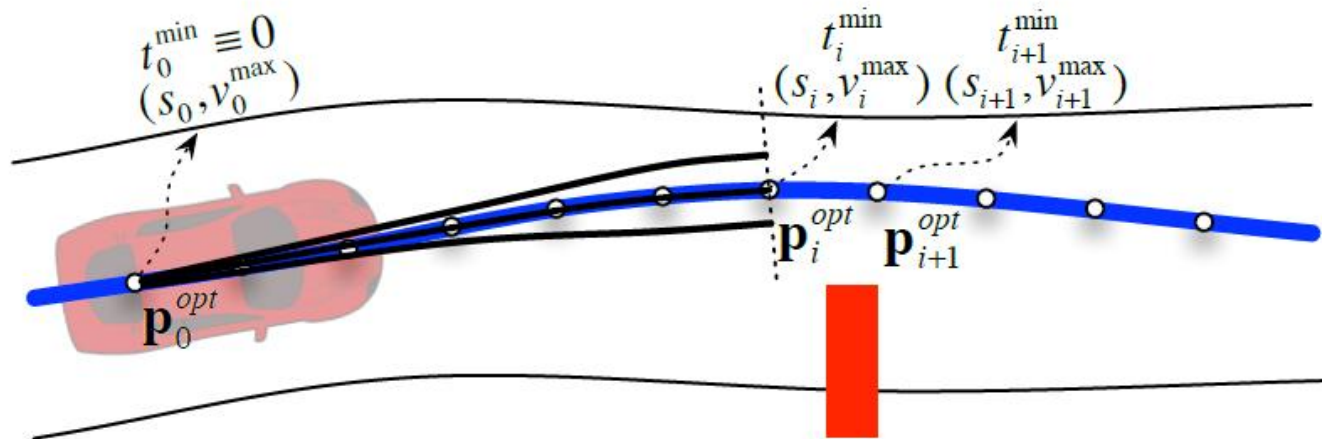


□ 轨迹采样与评估

- 根据到达时间、轨迹末位置和速度在参考轨迹采样

参考轨迹的规划
(考虑道路形状、障碍物)

跟踪轨迹的规划
(考虑运动学、动力学约束)



- 轨迹评估：相对参考轨迹的累积横向偏移距离和时间偏移

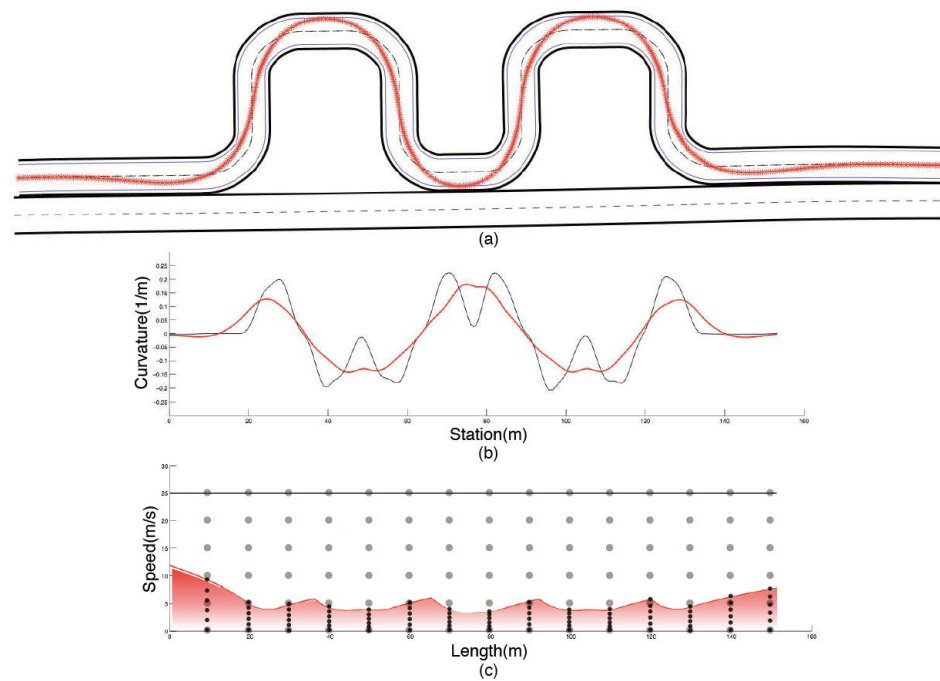
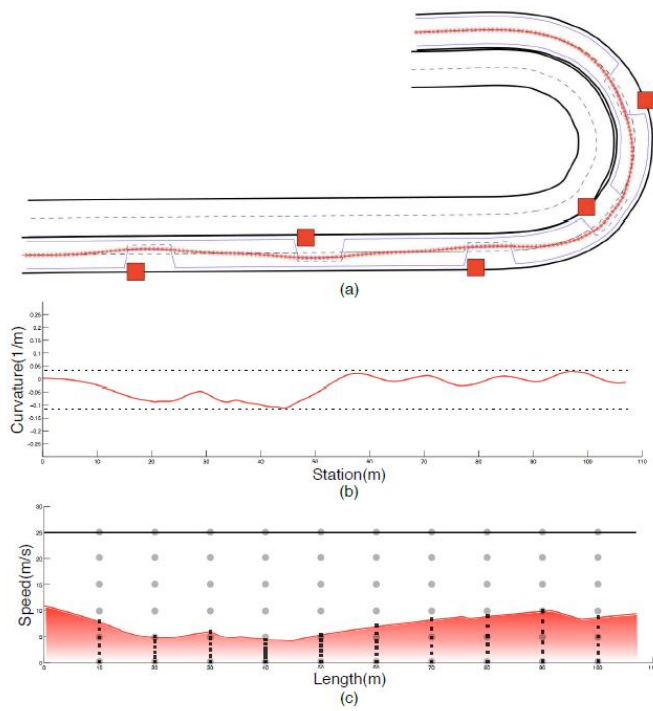
$$cost = \omega_{spatial} \cdot C_{spatial} + \omega_{temporal} \cdot C_{temporal}$$



基于迭代计算的时空规划方法

□ 如何加速迭代计算方法规划效率？

- 利用优化粗略生成初始的非参数化轨迹
- 在参考的轨迹附近聚焦搜索以减少计算





大纲



- ⬢ 时空联合规划的概念
- ⬢ 基于搜索的规划方法
- ⬢ 基于迭代计算的规划方法
- ⬢ 基于时空走廊的规划方法**
- ⬢ 应用案例展示

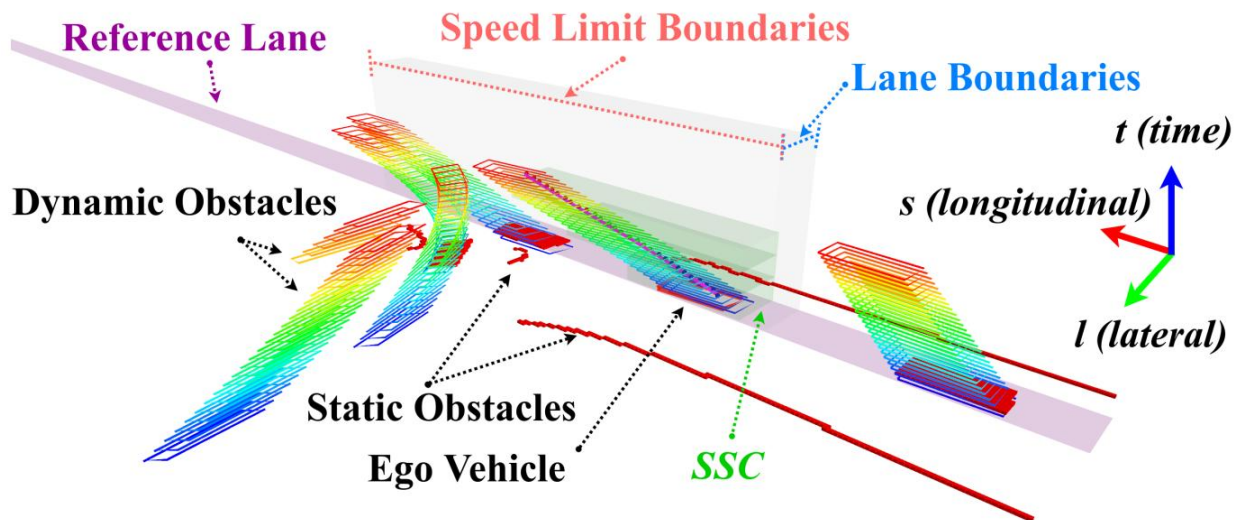
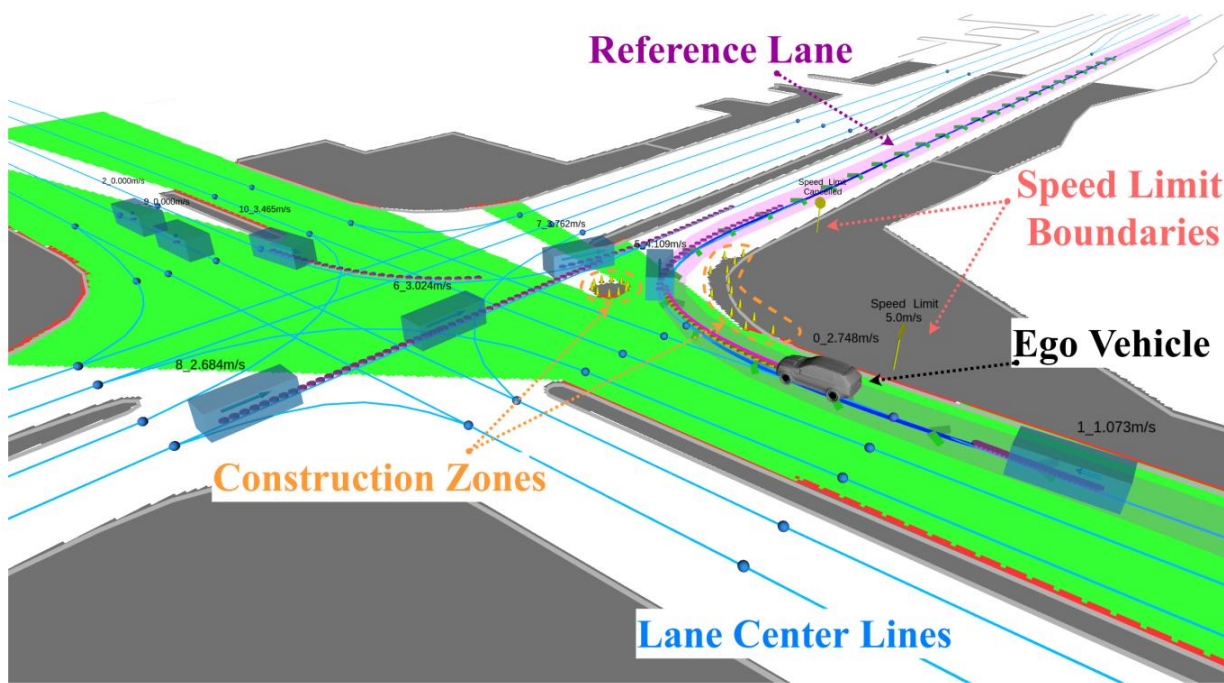


基于时空走廊的规划方法



□ 语义时空走廊的概念

- 规划面临的问题：自动驾驶的语义要素具有不同形式的数学表征；保证规划轨迹的安全性和可行性。
- 语义时空走廊（Semantic Spatio-Temporal Semantic Corridor）一系列被语义要素约束的相互连接的无碰撞立方体。



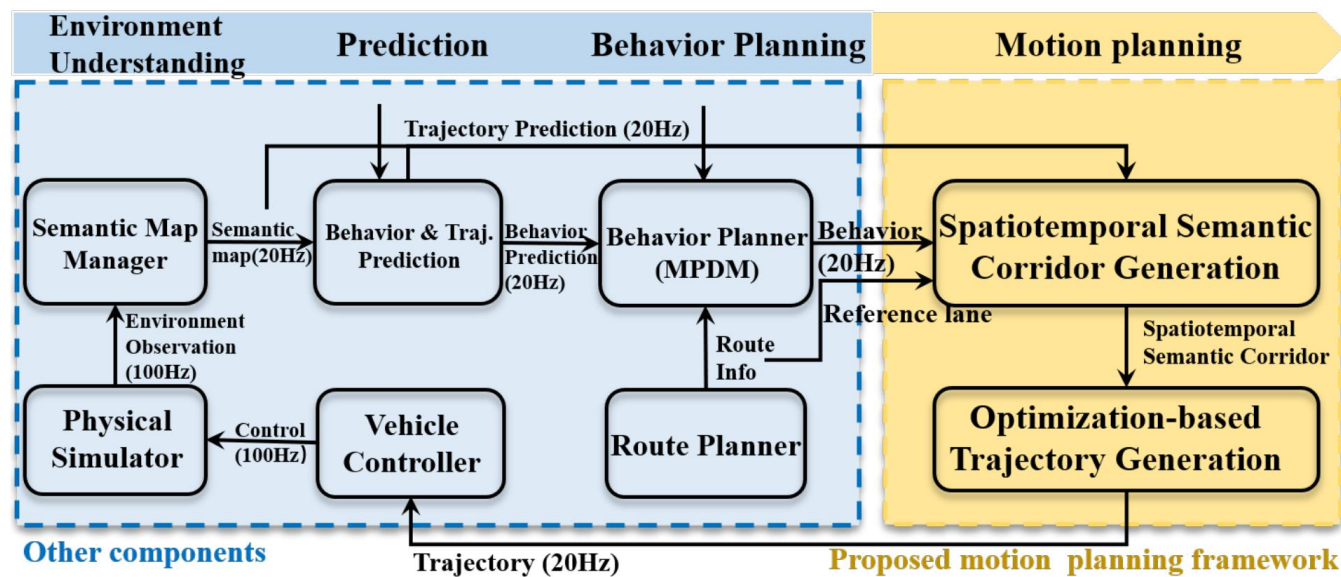


基于时空走廊的规划方法



□ 轨迹生成框架

- 环境理解：处理规划所需要的语义要素（占据栅格地图、动态物体、车道线、交通规则）；
- 预测：提供移动物体的未来轨迹；
- 行为规划MPDM[2]:
 - 假设所有车辆执行有限的闭环策略（车道保持，左变道，右变道）；
 - 使用简化模型模拟所有车辆的未来状态；
 - 根据代价函数评估未来的局面，选择最优的行为；
- 运动规划：时空语义走廊生成/轨迹优化。



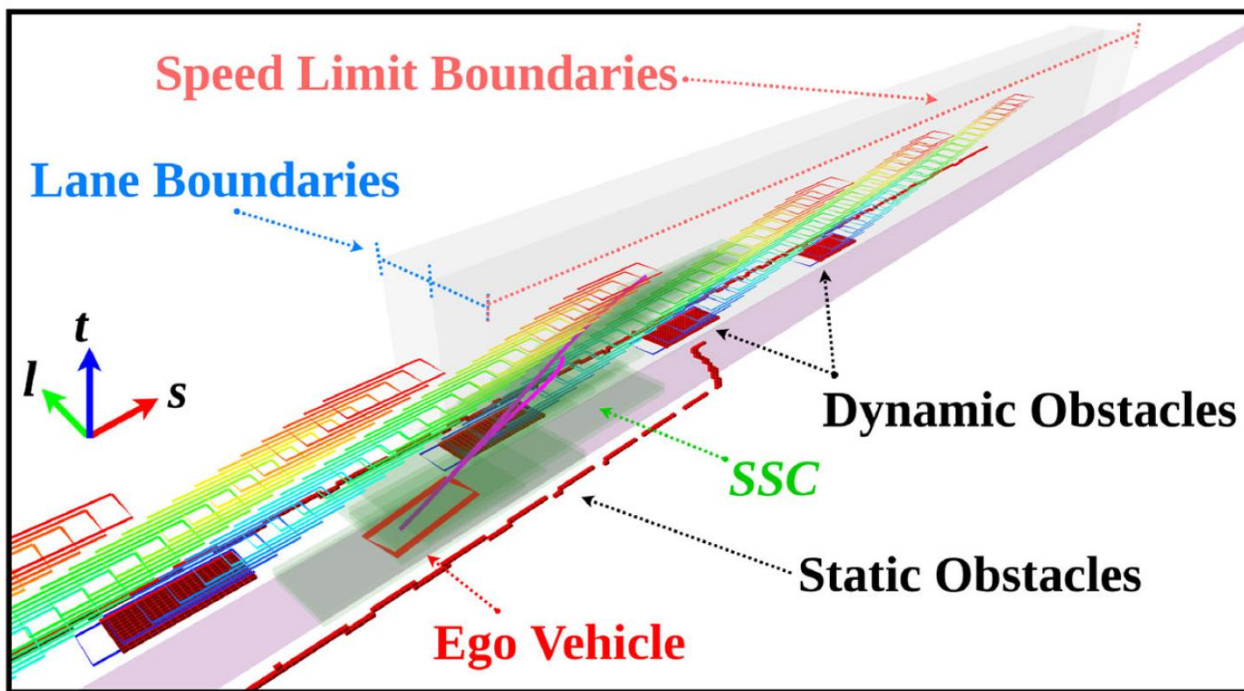
[2] Galceran E, Cunningham A G, et al. Multipolicy Decision-Making for Autonomous Driving via Changepoint-based Behavior Prediction[C]//Robotics: Science and Systems. 2015, 1(2):



基于时空走廊的规划方法

□ 语义时空走廊的障碍物要素

- 坐标系： $s-l-t$ 3D构型空间（Recall：用于搜索的 $s-l-t$ 栅格地图）
- 静止障碍物：横跨整个时间轴的障碍物
- 动态障碍物：根据预测轨迹在时间轴上存在的一系列障碍物
- 红灯：占据特定纵向空间和时间的障碍物



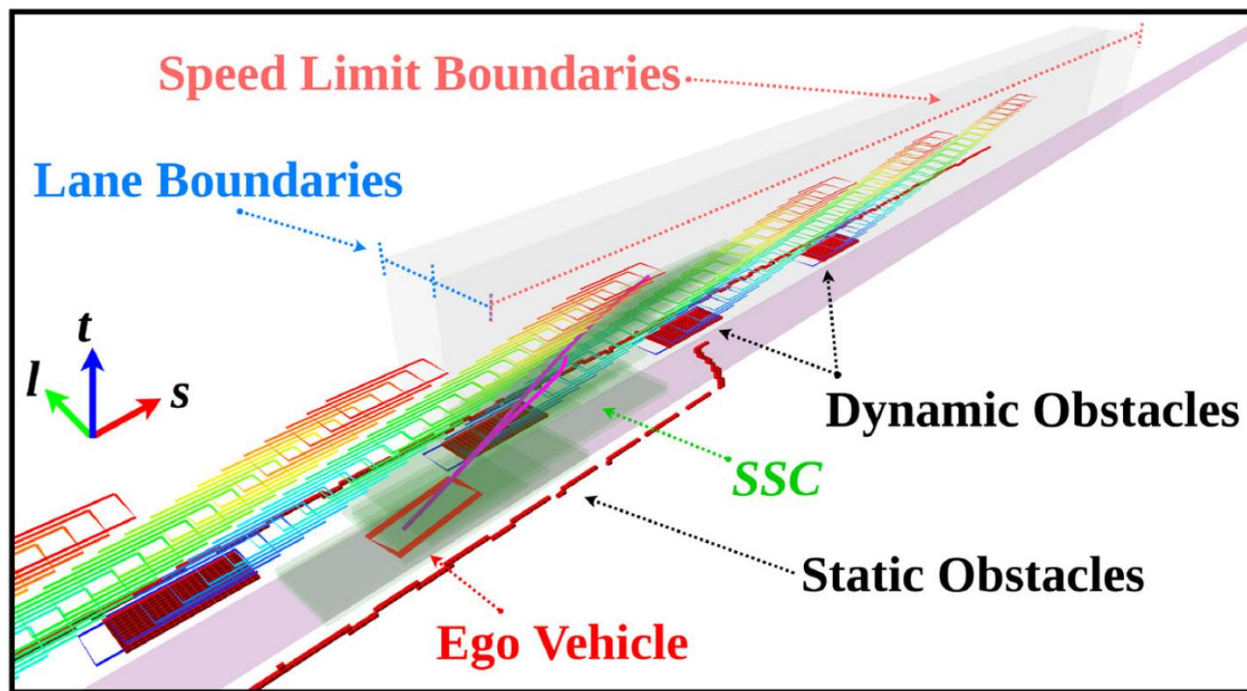


基于时空走廊的规划方法



□ 语义时空走廊的约束要素

- 速度约束：道路限速，停止标志
- 时间约束：变道时长
- 软约束与硬约束
- 语义边界的概念：特定语义约束开始和停止生效的位置，例如速度约束的纵向范围 $[s_{begin}, s_{end}]$ ，变道时长约束的横向范围 $[d_{begin}, d_{end}]$





基于时空走廊的规划方法



□ 语义时空走廊构建：输入数据

- 其他车辆的未来状态（MPDM前向模拟）
- 自车初始状态
- 语义边界
- $s-l-t$ 构型空间

Algorithm 1: Semantic Corridor Generation.

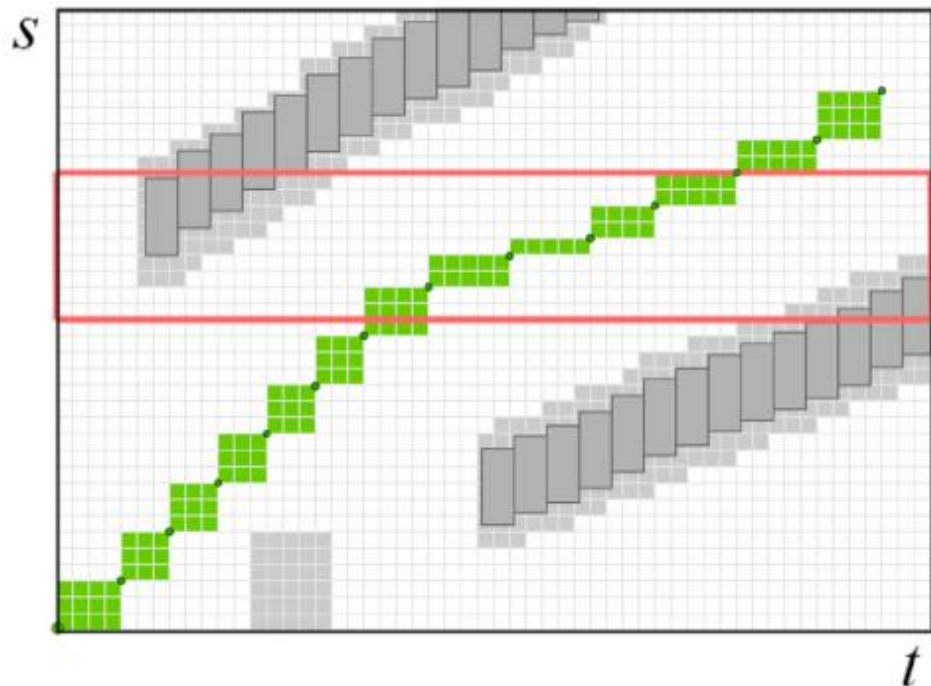
- 1: Inputs: forward simulated states $\{x_0, x_1, \dots, x_t\}$, initial state x_{des} , semantic boundaries \mathcal{B} , slt configuration space \mathcal{E} ;
 - 2: Initializes: seeds $\mathcal{S}^{\text{seed}} = \emptyset$;
 - 3: $\mathcal{S}^{\text{seed}} \leftarrow \text{SeedGeneration}(\{x_0, x_1, \dots, x_t\}, x_{\text{des}})$;
 - 4: $\mathcal{C}^{\text{infl}} \leftarrow \text{CubeInflation}(\mathcal{S}^{\text{seed}}, \mathcal{B}, \mathcal{E})$;
 - 5: $\mathcal{C}^{\text{infl}} \leftarrow \text{ConstraintAssociation}(\mathcal{C}^{\text{infl}}, \mathcal{B})$;
 - 6: $\mathcal{C}^{\text{final}} \leftarrow \text{CubeRelaxation}(\mathcal{C}^{\text{infl}}, \mathcal{E})$;
-



基于时空走廊的规划方法

语义时空走廊构建：生成种子

- 将前向模拟状态投影至 $s-l-t$ 坐标系
- 初始立方体由连续两个种子作为顶点构成
- 初始立方体需保证无碰撞
- 保证时空拓扑的一致



- 移动障碍物
- 膨胀障碍物
- 前向轨迹种子
- 初始立方体
- 速度约束边界

Algorithm 1: Semantic Corridor Generation.

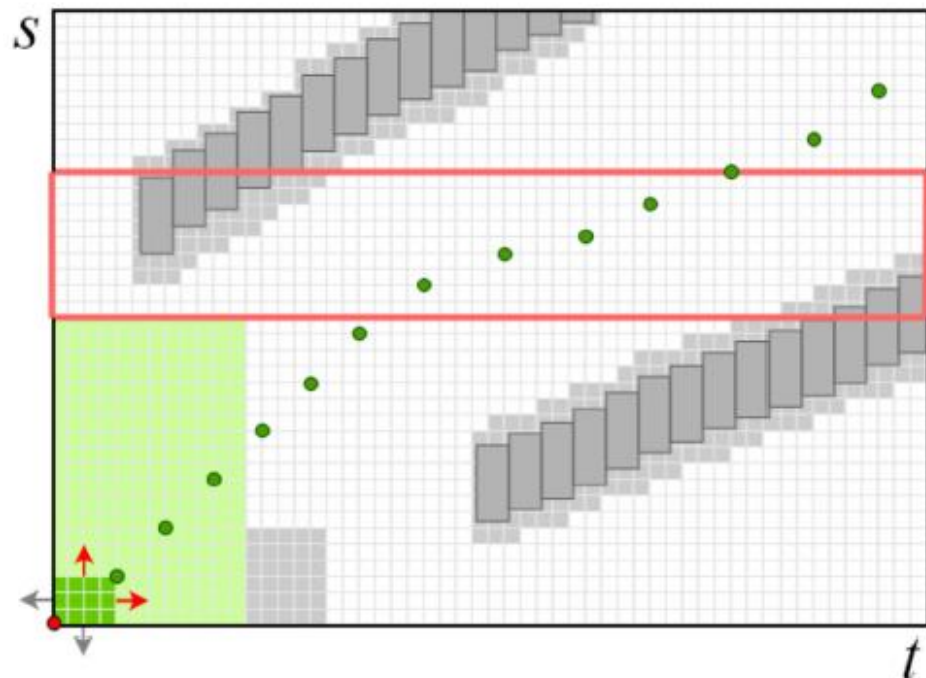
- 1: Inputs: forward simulated states $\{x_0, x_1, \dots, x_t\}$, initial state x_{des} , semantic boundaries \mathcal{B} , slt configuration space \mathcal{E} ;
- 2: Initializes: seeds $\mathcal{S}^{\text{seed}} = \emptyset$;
- 3: $\mathcal{S}^{\text{seed}} \leftarrow \text{SeedGeneration}(\{x_0, x_1, \dots, x_t\}, x_{\text{des}})$;
- 4: $\mathcal{C}^{\text{infl}} \leftarrow \text{CubeInflation}(\mathcal{S}^{\text{seed}}, \mathcal{B}, \mathcal{E})$;
- 5: $\mathcal{C}^{\text{infl}} \leftarrow \text{ConstraintAssociation}(\mathcal{C}^{\text{infl}}, \mathcal{B})$;
- 6: $\mathcal{C}^{\text{final}} \leftarrow \text{CubeRelaxation}(\mathcal{C}^{\text{infl}}, \mathcal{E})$;



基于时空走廊的规划方法

语义时空走廊构建：膨胀立方体

- 将第一个立方体膨胀至语义边界和障碍物



- 可膨胀方向
- 不可膨胀方向
- 前向轨迹种子
- 膨胀后立方体
- 速度约束边界

Algorithm 1: Semantic Corridor Generation.

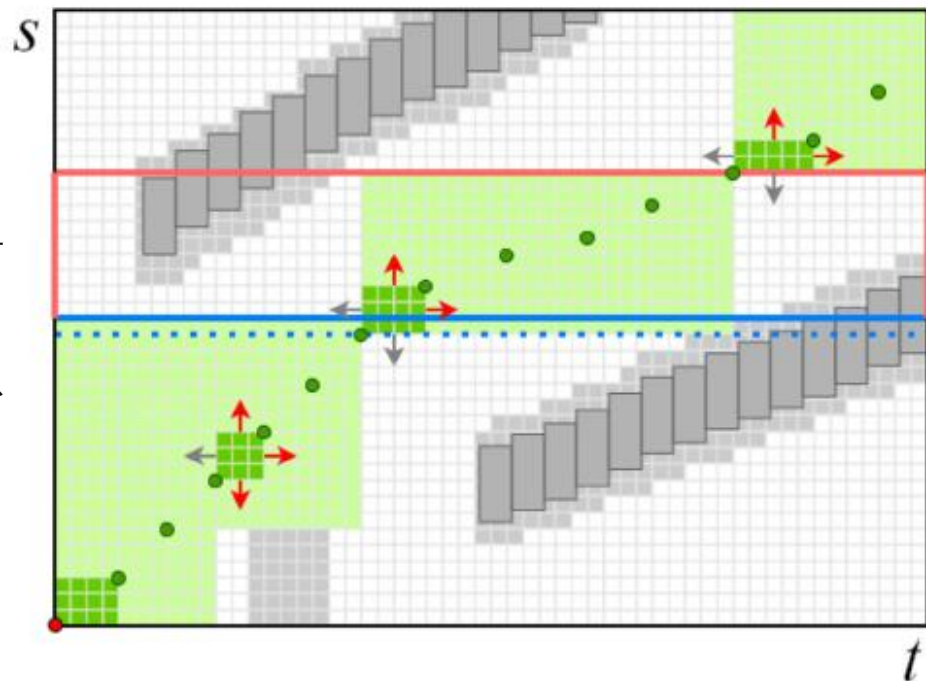
- 1: Inputs: forward simulated states $\{x_0, x_1, \dots, x_t\}$, initial state x_{des} , semantic boundaries \mathcal{B} , slt configuration space \mathcal{E} ;
- 2: Initializes: seeds $\mathcal{S}^{\text{seed}} = \emptyset$;
- 3: $\mathcal{S}^{\text{seed}} \leftarrow \text{SeedGeneration}(\{x_0, x_1, \dots, x_t\}, x_{\text{des}})$;
- 4: $\mathcal{C}^{\text{infl}} \leftarrow \text{CubeInflation}(\mathcal{S}^{\text{seed}}, \mathcal{B}, \mathcal{E})$;
- 5: $\mathcal{C}^{\text{infl}} \leftarrow \text{ConstraintAssociation}(\mathcal{C}^{\text{infl}}, \mathcal{B})$;
- 6: $\mathcal{C}^{\text{final}} \leftarrow \text{CubeRelaxation}(\mathcal{C}^{\text{infl}}, \mathcal{E})$;



基于时空走廊的规划方法

语义时空走廊构建：关联约束

- 第二个立方体：以第一个立方体的最后一个种子和立方体外的第一个种子构成初始立方体
- 第三个立方体：初始立方体和语义边界相交，只允许单向膨胀（s轴正方向）



- 可膨胀方向
- 不可膨胀方向
- 初始立方体
- 膨胀后立方体
- 相交的语义边界

Algorithm 1: Semantic Corridor Generation.

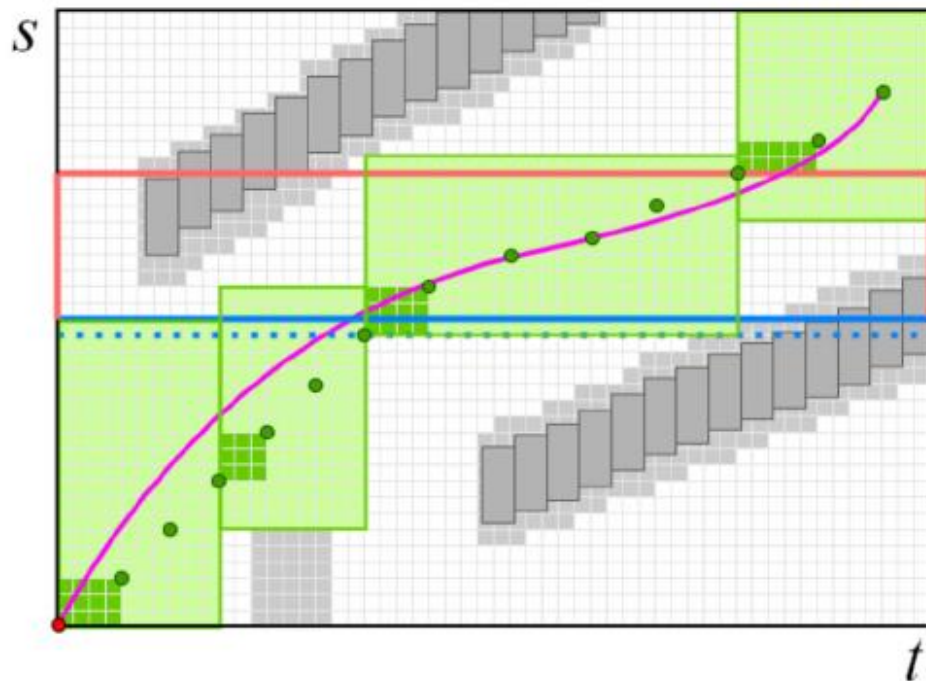
- 1: Inputs: forward simulated states $\{x_0, x_1, \dots, x_t\}$, initial state x_{des} , semantic boundaries \mathcal{B} , slt configuration space \mathcal{E} ;
- 2: Initializes: seeds $\mathcal{S}^{\text{seed}} = \emptyset$;
- 3: $\mathcal{S}^{\text{seed}} \leftarrow \text{SeedGeneration}(\{x_0, x_1, \dots, x_t\}, x_{\text{des}})$;
- 4: $\mathcal{C}^{\text{infl}} \leftarrow \text{CubeInflation}(\mathcal{S}^{\text{seed}}, \mathcal{B}, \mathcal{E})$;
- 5: $\mathcal{C}^{\text{infl}} \leftarrow \text{ConstraintAssociation}(\mathcal{C}^{\text{infl}}, \mathcal{B})$;
- 6: $\mathcal{C}^{\text{final}} \leftarrow \text{CubeRelaxation}(\mathcal{C}^{\text{infl}}, \mathcal{E})$;



基于时空走廊的规划方法

语义时空走廊构建：放宽边界

- 考虑软约束，留出额外的空间
- 保留硬约束和避障



- 可膨胀方向
- 不可膨胀方向
- 初始立方体
- 时空语义走廊
- 优化后轨迹

Algorithm 1: Semantic Corridor Generation.

- 1: Inputs: forward simulated states $\{x_0, x_1, \dots, x_t\}$, initial state x_{des} , semantic boundaries \mathcal{B} , slt configuration space \mathcal{E} ;
- 2: Initializes: seeds $\mathcal{S}^{\text{seed}} = \emptyset$;
- 3: $\mathcal{S}^{\text{seed}} \leftarrow \text{SeedGeneration}(\{x_0, x_1, \dots, x_t\}, x_{\text{des}})$;
- 4: $\mathcal{C}^{\text{infl}} \leftarrow \text{CubeInflation}(\mathcal{S}^{\text{seed}}, \mathcal{B}, \mathcal{E})$;
- 5: $\mathcal{C}^{\text{infl}} \leftarrow \text{ConstraintAssociation}(\mathcal{C}^{\text{infl}}, \mathcal{B})$;
- 6: $\mathcal{C}^{\text{final}} \leftarrow \text{CubeRelaxation}(\mathcal{C}^{\text{infl}}, \mathcal{E})$;

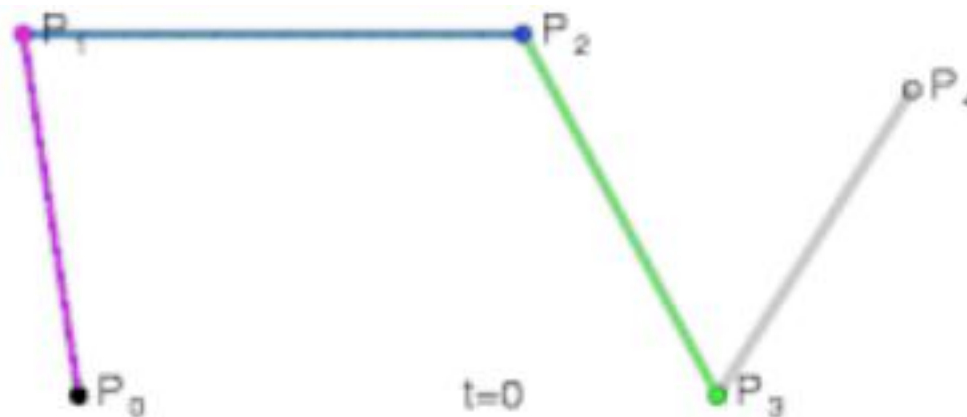
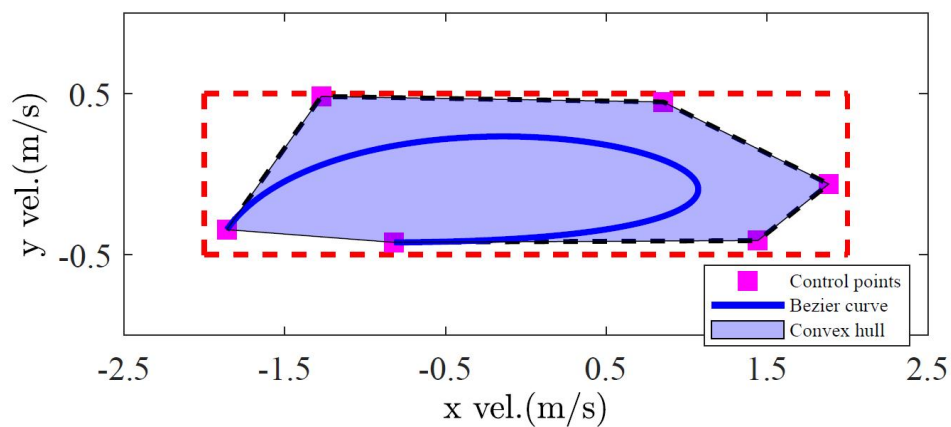


基于时空走廊的规划方法



□ 保证安全性和可行性的轨迹生成

- 贝塞尔曲线 $f(t) = p_0 b_m^0(t) + p_1 b_m^1(t) + \dots + p_m b_m^m(t) = \sum_{i=0}^m p_i \cdot b_m^i(t), t \in [0, 1]$
- Bernstein多项式 $b_m^i(t) = \binom{m}{i} t^i \cdot (1-t)^{m-i}$
- 控制点 $p = [p_0, p_1, \dots, p_m]$
- 凸包性质: $f(t)$ 能保证被完全限制在控制点 p 组成的凸包内
- Hodograph性质: 贝塞尔曲线的导数 $\frac{df(t)}{dt}$ 是由控制点 $p_i^{(1)} = m \cdot (p_{i+1} - p_i)$ 组成的另一条贝塞尔曲线





基于时空走廊的规划方法



□ 轨迹表示

- 每个时空走廊中的立方体对应着一段贝塞尔曲线
- 每个维度 $\sigma \in \{s, l\}$ 各用一条 n 段的贝塞尔轨迹表示
- p_i^j 对应着第 j 段轨迹的第 i 个控制点, t_0, t_1, \dots, t_n 是每段轨迹的初始和末尾时刻
- 使用尺度因子 α_j 保证轨迹定义在 $[0, 1]$ 的区间上

$$f^\sigma(t) = \begin{cases} \alpha_1 \cdot \sum_{i=0}^m p_i^1 \cdot b_m^i \left(\frac{t-t_0}{\alpha_1} \right), t \in [t_0, t_1] \\ \alpha_2 \cdot \sum_{i=0}^m p_i^2 \cdot b_m^i \left(\frac{t-t_1}{\alpha_2} \right), t \in [t_1, t_2] \\ \vdots \\ \alpha_n \cdot \sum_{i=0}^m p_i^n \cdot b_m^i \left(\frac{t-t_{n-1}}{\alpha_n} \right), t \in [t_{n-1}, t_n] \end{cases}$$



基于时空走廊的规划方法



□ 轨迹优化问题（二次规划QP）

- 最小化jerk

$$J_i = \omega_s \int_{t_{j-1}}^{t_j} \left(\frac{d^3 f^s(t)}{dt^3} \right)^2 dt + \omega_l \int_{t_{j-1}}^{t_j} \left(\frac{d^3 f^l(t)}{dt^3} \right)^2 dt$$

- 利用凸包性质得到的安全和动力学约束, $\beta_{j,-}^{\sigma,(k)}$ 和 $\beta_{j,+}^{\sigma,(k)}$ 表示第 j 段轨迹 k 阶导数的上下界

$$\beta_{j,-}^{\sigma,(k)} \leq \frac{d^k f_j^\sigma(t)}{dt^k} \leq \beta_{j,+}^{\sigma,(k)}, \sigma \in \{s, l\}, k = 0, 1, 2$$

- 首末状态约束

$$\frac{d^k f_0^\sigma(t_0)}{dt^k} = \sigma_{t_0}^{(k)}, \frac{d^k f_n^\sigma(t_n)}{dt^k} = \sigma_{t_n}^{(k)}$$

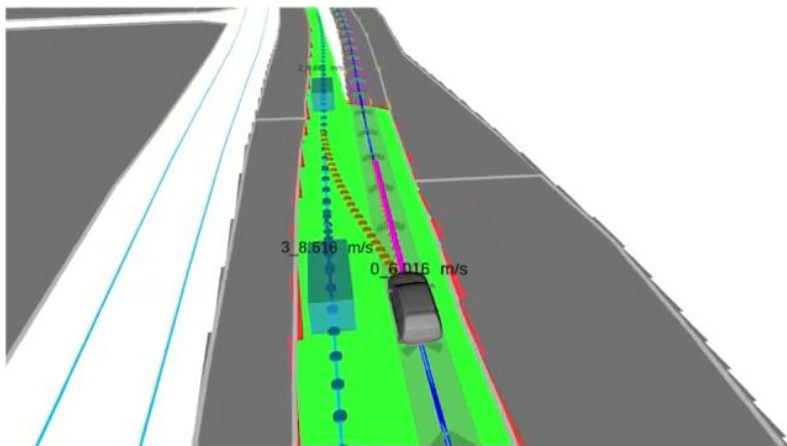
- 连续性约束

$$\frac{d^k f_j^\sigma(t_j)}{dt^k} = \frac{d^k f_{j+1}^\sigma(t_j)}{dt^k}$$

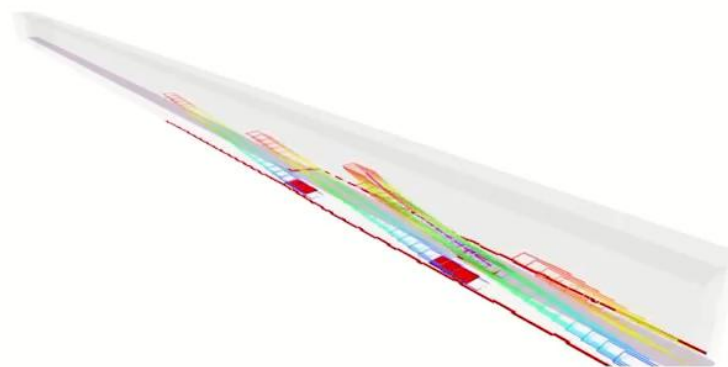


基于时空走廊的规划方法

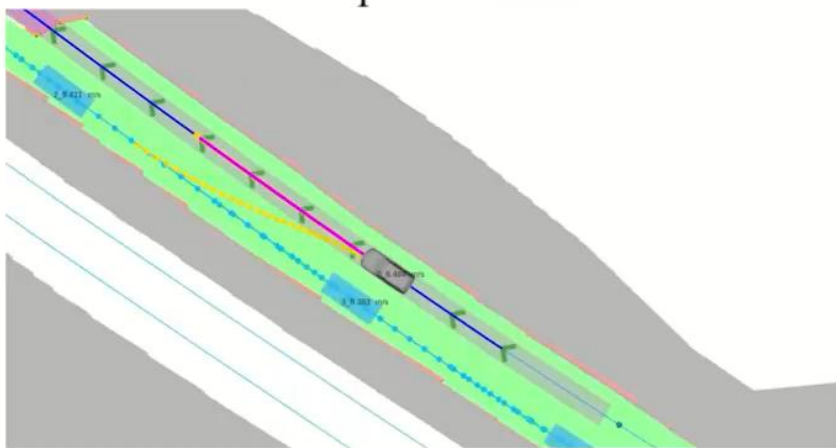
Merging into congested traffic due to road construction



Third-person view



slt space



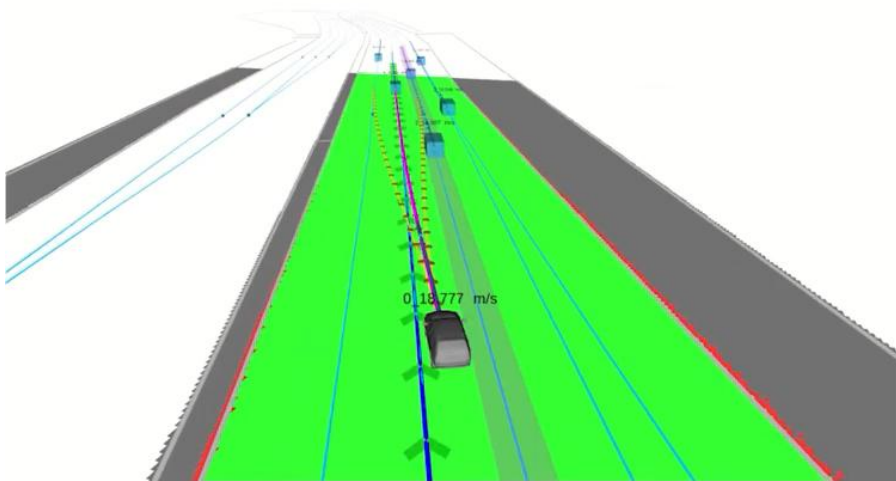
Bird view

因道路施工而汇入拥堵交通

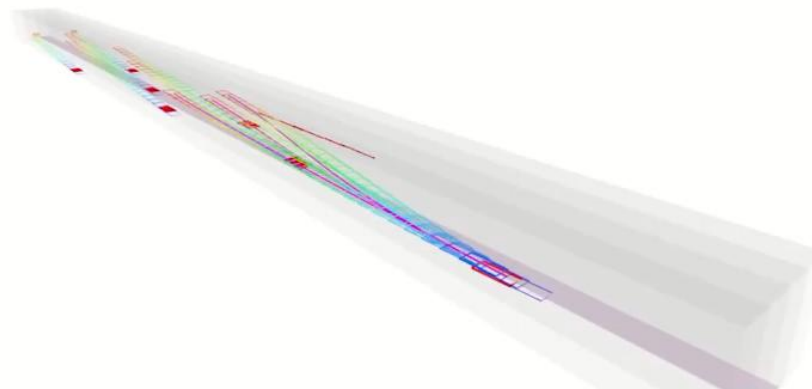


基于时空走廊的规划方法

Overtaking on an urban expressway with dense traffic

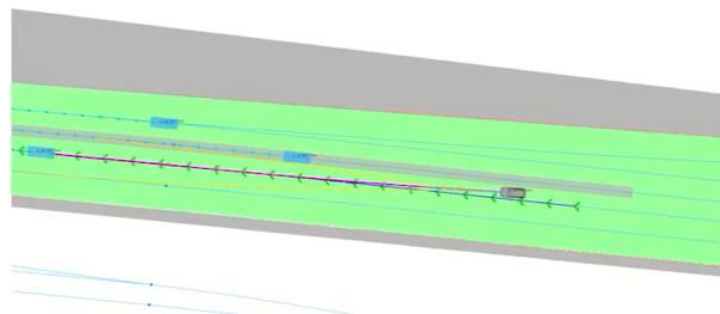


Third-person view



slt space

Bird view



在交通密集的城市快速路超车



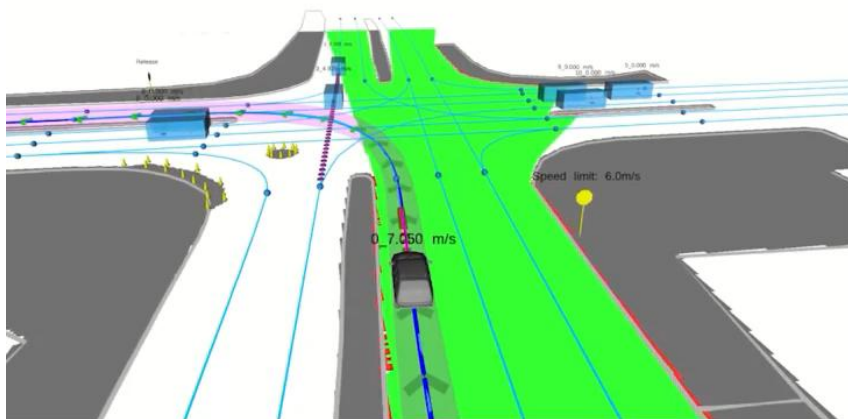
基于时空走廊的规划方法



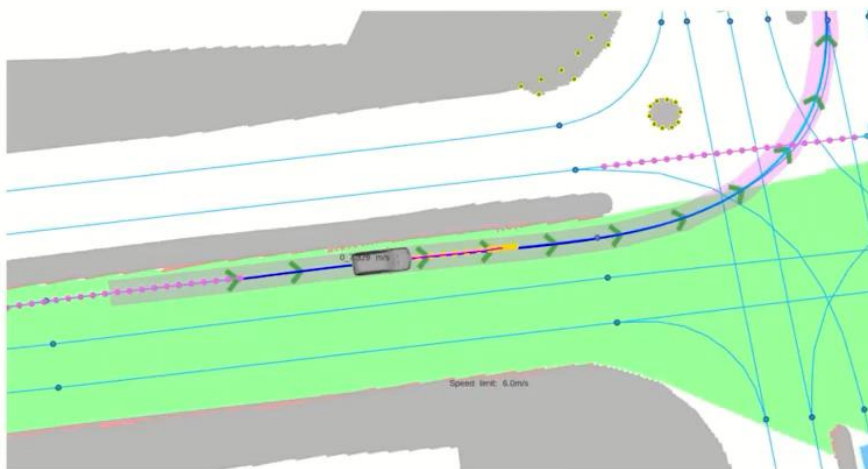
An unprotected left turn at an intersection

Round 1

x2



Third-person view



Bird view

在路口的无保护左转



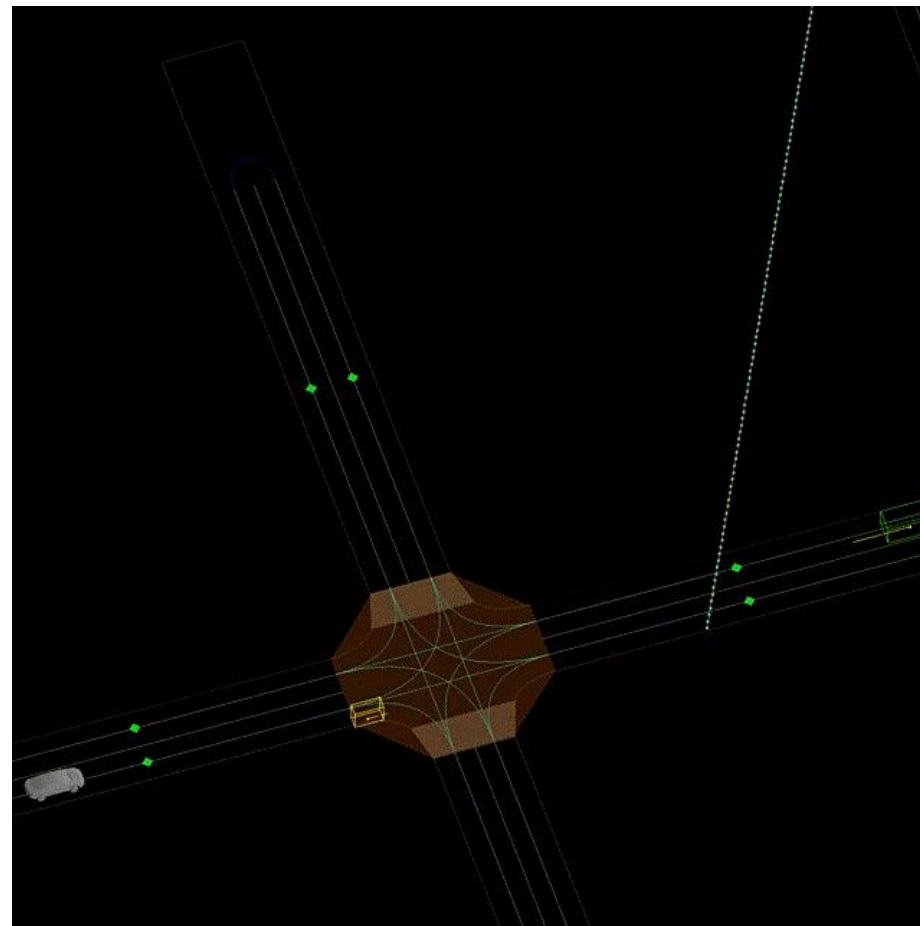
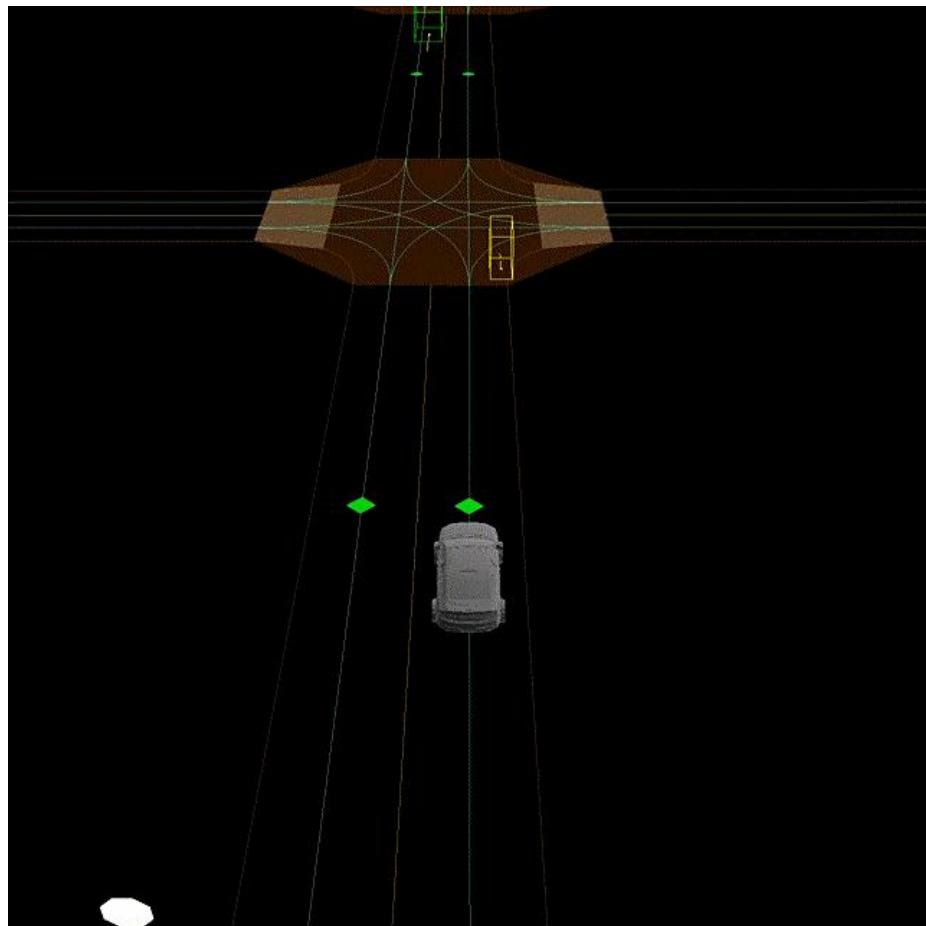
大纲



- ⬢ 时空联合规划的概念
- ⬢ 基于搜索的规划方法
- ⬢ 基于迭代计算的规划方法
- ⬢ 基于时空走廊的规划方法
- ⬢ 应用案例展示



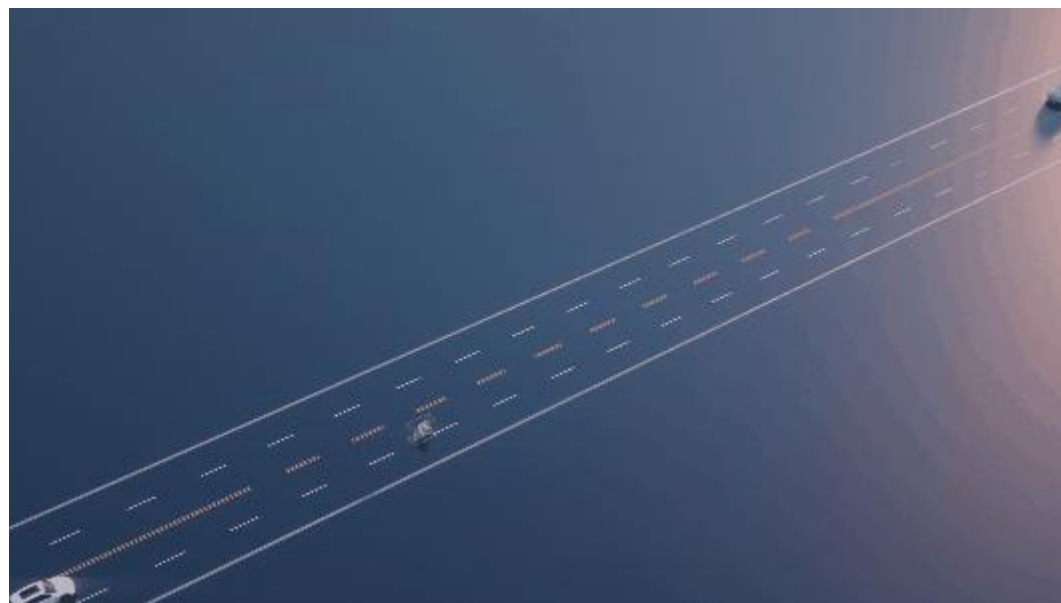
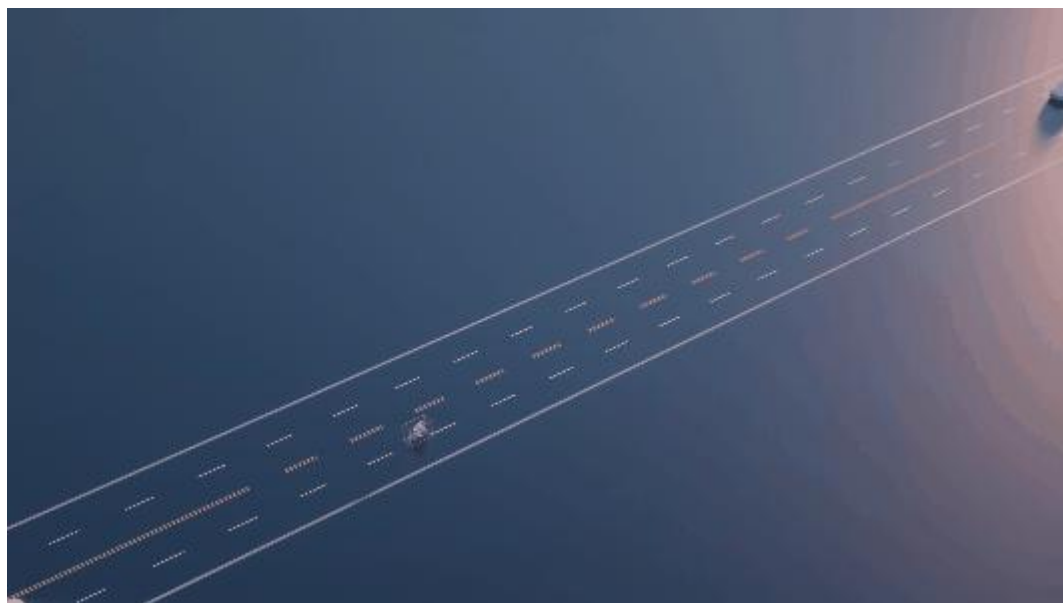
应用案例展示



The animation is from QCrafts



应用案例展示



The animation is from QCrafts

感谢聆听!

Thanks for Listening

