为自动驾驶车辆带来多样性：可解释的多车辆决策与规划框架

Licheng Wen、Pinlong Cai、Daocheng Fu、Song Mao 和 Yikang Li 上海
人工智能实验室，中国上海
{wenlicheng,caipinlong,fudaocheng,maosong,liyikang}@pjlab.org.cn

arXiv:2302.06803v1 [cs.RO] 14 Feb 2023

摘要
随着自动驾驶的发展，自动驾驶车辆（AV）和人类驾驶车辆（HV）在同一道路上行驶正变得越来越普遍。现有的车载单车规划算法难以应对现实世界中复杂的社会交互。这些方法做出的决策难以让人理解，从而增加了碰撞事故的风险，也使其不太可能在实践中得到应用。此外，开源交通仿真器生成的车辆流过于保守，且缺乏行为多样性。我们提出了一种分层多车决策与规划框架，具备多项优势。该框架能够对车流中的所有车辆进行联合决策，并通过高频规划模块及时响应动态环境。决策模块会生成可解释的动作序列，从而明确地向周围的人类驾驶车辆传达自身意图。我们还提出了合作因子和轨迹权重集，为交通中的自动驾驶车辆在社会层面和个体层面带来多样性。通过多个场景的实验验证了所提框架的优越性，并通过闭环仿真展示了生成的车辆轨迹中丰富多样的行为。

关键词
自动驾驶、驾驶行为、轨迹生成、车流

1 引言

在学术界和工业界的共同努力下，过去十年来，自动驾驶研究蓬勃发展。如今，越来越多的公司正在道路上测试其自动驾驶车辆（AV），使其成为人类驾驶车辆（HV）之外交通流中的新参与者。现实世界中的车流包含社会和个体层面的多种行为，分别称为社会行为和驾驶习惯。社会行为 [20, 27] 指车辆与其他车辆的互动方式。当另一辆车变道、超车或并入车流时，驾驶员会根据经验选择继续当前行驶方式还是让行。驾驶员在驾驶时也各有特点。因此，引入驾驶习惯 [17, 26, 28] 来描述个体差异，尤其是巡航舒适性和驾驶安全性方面的差异。

自动驾驶车辆的行为多样性主要体现在决策与规划模块中，这些模块是自动驾驶系统的核心组成部分。当前主流的单车决策与规划方法可分为两类：重决策框架和轻决策框架 [4, 11]。前一种方法通常将决策与运动规划分开，以降低计算负担和复杂度 [15, 24], ；而轻决策框架则通过弱化决策模块并增加规划器采样规模，提升算法在边缘场景中的适应性 [5, 7, 10]。然而，这些方法难以在现实世界中与其他交通参与者进行博弈，因此对社会交互的理解不足。

随着车联网（V2X）技术在自动驾驶车辆中的应用，多辆交通车辆的集中式决策和轨迹规划成为可能。集中式规划器能够有效处理车流中的复杂交互，包括竞争与协作。多车决策与规划方法有多种类型。基于优化的方法 [8, 9, 19] 能够很好地建模车辆之间的博弈论交互。基于蒙特卡洛树搜索的方法 [1, 12, 13] 采用树结构管理交互，并通过对车辆动作空间进行采样，大幅提升算法效率。然而，这些方法会直接输出长期控制变量；道路上存在人工驾驶车辆时，这可能带来风险。基于学习的方法 [2, 3, 16, 21] 从开放数据集中获取车辆的驾驶策略，并通过预测其他车辆的轨迹来避免碰撞。这类方法生成的轨迹无法明确表达驾驶意图，因此乘客或其他道路使用者难以理解。此外，这些方法还高度依赖数据集。

此外，流行的开源自动驾驶仿真器为生成背景交通提供了解决方案。SUMO [14]对每辆车应用跟驰模型和换道模型。该方法生成的轨迹能够避免碰撞，但缺乏车辆运动学约束。CARLA [6]采用交通管理模块，根据当前仿真状态为车辆生成指令。然而，这种方法采用僵化的行为模型，导致生成的轨迹过于保守且缺乏多样性。

我们认为，对行为多样性进行建模有助于更好地理解真实世界交通流，并使自动驾驶仿真更加可信。我们提出了一种分层多车决策与规划框架：高层为场景中的所有车辆做出联合决策，低层则为每辆受控自动驾驶车辆生成符合运动学约束的轨迹。该框架的主要特点可概括为**DII**：

- **多**样性：通过在决策模块中加入车辆协作因子，并在规划模块中引入轨迹权重集，我们的框架能够生成具有社会层面和个体层面多样性的解决方案。

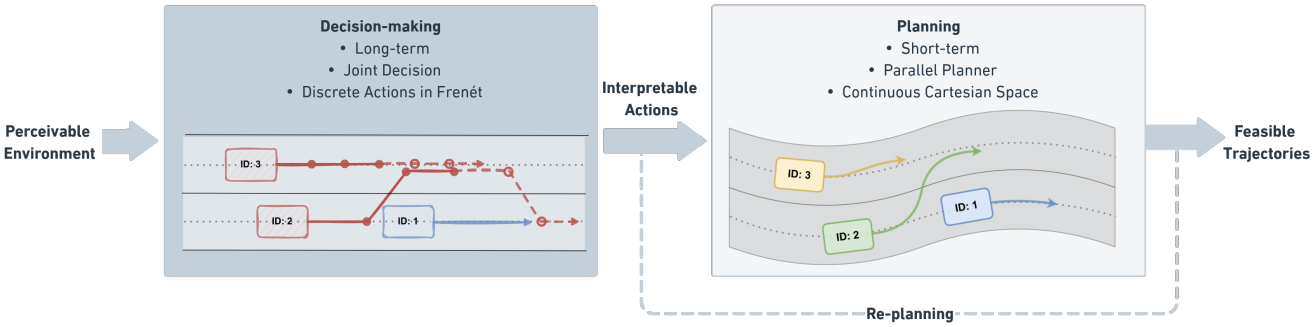

图1：所提出的决策与规划框架示意图

- **可**解释性：我们框架中的决策模块 - 为每辆自动驾驶车辆生成可解释的动作序列，让车内乘客了解当前情况，并通过刹车灯和转向灯向周围的人类驾驶车辆明确传达意图。
- **我**们将交通流中的所有车辆都纳入决策，因此能够探索车辆间复杂的相互作用。此外，该框架通过高频重规划，确保能够实时响应动态环境。

本文其余部分安排如下：第2节介绍所提出方法的框架。第3节详细介绍该框架，包括考虑社会行为的多车辆蒙特卡洛树搜索，以及结合驾驶习惯的并行轨迹规划器。第4节展示性能测试和闭环仿真，并讨论结果。最后，第5节总结全文并展望未来工作。

2 框架

考虑一个集合 $\mathrm{V} = \{V_1, V_2, \cdots V_N\}$，其中包含 N辆车辆，包括自动驾驶车辆和人工驾驶车辆。有 K辆自动驾驶车辆可供集中式规划使用，记为 $\mathrm{Vc} = \{V_1, \cdots V_k\} \subseteq \mathrm{V}$。其余自动驾驶车辆以及所有人工驾驶车辆均视为非受控车辆，它们独立进行决策和运动规划。

我们的任务是为每辆受控车辆 $V_i \in \mathrm{Vc}$ 生成具有行为多样性的可行轨迹，并与环境中的非受控车辆 $V_j \in \{\mathrm{V} \setminus \mathrm{Vc}\}$协同。我们提出了一个包含决策模块和轨迹规划模块的两阶段多车辆框架。该框架的输入是一个可感知环境，其中包含道路状况、路线信息、车辆状态以及对非受控车辆的预测。图 1 展示了我们所提出框架的示意图，其中 ID 为 2 和 3 的车辆是由该框架控制的自动驾驶车辆，而车辆 1 是一辆非受控车辆。下面将介绍所提出框架两个阶段的具体内容。

在第一阶段，我们提出了一个决策模块，用于处理车辆之间的社会交互。该模块为所有车辆联合且同时生成长期的、粗粒度决策。决策环境构建于Frenét坐标系 [18] 上，使决策模块能够专注于车辆间交互，而无需考虑道路约束。由于连续变道和超车等动作通常需要相对较长的时间才能完成，因此决策模块必须具备前瞻性并保持一致性。决策模块通过对车流进行集中式联合决策，解决车辆之间的交互博弈。通过在奖励函数中设置不同的合作因子，该模块能够生成车流中多样化的社会行为。决策模块为每辆受控车辆生成离散动作的时间序列。由于序列中的每个动作都具有明确的物理含义，例如变道、加速、保持速度等，因此我们的决策模块输出结果具有可解释性。

在第二阶段，规划模块接收来自第一阶段的离散动作序列，并为每辆受控车辆生成满足运动学约束的连续轨迹。得益于前端决策模块的前瞻性，轨迹规划模块只需在较短的规划时域内生成轨迹。我们的规划模块采用分布式并行架构，即为每辆车辆独立运行一个规划器，这更贴近现实世界中的人类驾驶方式。对于每个并行规划器，都会选取其接收到的动作序列的前段作为引导，生成在笛卡尔空间中符合车辆运动学约束的连续轨迹。该模块通过规划器针对不同轨迹的选择策略，体现不同驾驶员的驾驶习惯。值得强调的是，规划器会保持较高的重规划频率，并持续预测周围车辆的行为，以避免潜在碰撞。

我们提出的两阶段框架为每个模块提供了具有物理意义的可解释结果，使自动驾驶车辆能够告知乘客当前情况，并向周围的人工驾驶车辆明确传达自身意图。同时，该框架确保了底线安全。规划模块通过预测其他车辆来处理碰撞风险，并生成无碰撞轨迹，即使决策模块生成了一些不切实际的动作序列也是如此。

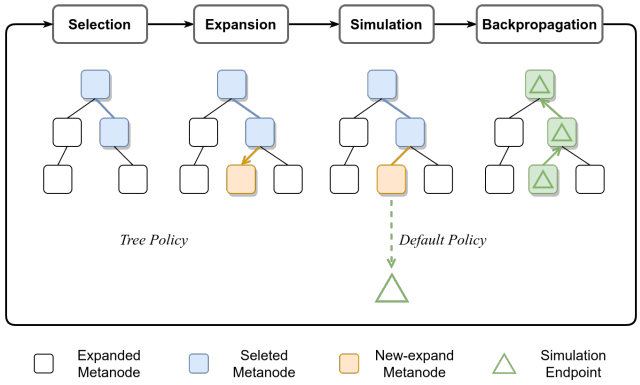

图 2：蒙特卡洛树搜索的流程

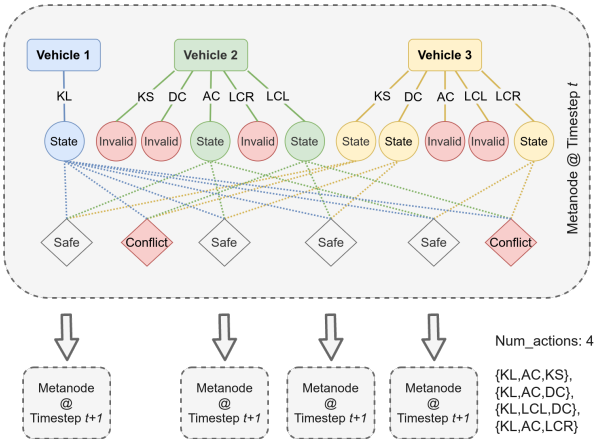

图3：蒙特卡洛树搜索中时间步 t 的一个元节点

3 方法

3.1 多车辆联合决策

3.1.1 基本蒙特卡洛树搜索算法。蒙特卡洛树搜索（MCTS）是一种用于决策过程的启发式搜索算法，它通过蒙特卡洛采样方法分析搜索空间，以找出最有希望的动作。通用 MCTS 算法 [1] 在一次搜索迭代中包含四个阶段：选择、扩展、仿真和反向传播，如图 2 所示。

选择阶段基于现有搜索树进行。算法将根据树策略递归地选择最有价值的子节点，直到到达尚未完全扩展的节点。然后，使用剩余可能动作中的一个扩展所选节点，以获得新的叶节点。算法将从该节点开始使用默认策略（也称为滚动模拟）进行仿真，直至到达终止节点。在反向传播阶段，算法将从仿真样本中提取奖励，并利用这些奖励更新所有遍历节点的平均奖励。达到预设的最大迭代次数后，即可通过在每一层贪心地选择奖励最大的子节点来获得解决方案。

为解决选择阶段的探索—利用困境，采用标准 UCT算法 [1] 。子节点 j 的 UCT 值可计算如下：

$$UCT = \overline{X}_j + 2C_p\sqrt{\frac{2\ln n}{n_j}} \quad (1)$$

其中，$\overline{X}_j$ 是节点 j 的平均奖励，n 是 j 的父节点被访问的次数，n_j 是子节点 j 被访问的次数，$C_p > 0$ 是用于平衡探索与利用的常数。每个节点的奖励分布应介于 0 和 1 之间。

原始的蒙特卡洛树搜索算法通常为单个智能体做出决策。然而，在我们的框架中，需要同时为交通流中的所有车辆做出决策。一种简单的方法是为每辆车分别执行一次蒙特卡洛树搜索。但是，分别为每辆车执行蒙特卡洛树搜索，需要在做出决策前准确预测其他车辆的行为，。这违反了Vapnik原则 [25]: “在解决一个问题时，不要把更难的问题作为中间步骤来解决。”因为对其他智能体的行为进行建模和预测是一个更难的问题。

3.1.2 蒙特卡洛树搜索中的元节点。目前已有多种基于蒙特卡洛树搜索的多车决策方法。Lenz 等人 [12] 提出了一种生成自动驾驶车辆预见性与协作行为的方法，但通过蒙特卡洛树搜索直接输出运动指令的方式，在动态环境中的鲁棒性较差，尤其是在道路上混有人工驾驶车辆时。另一种方法 [13] 为车辆设定优先级，并按照优先级从高到低依次作出决策，后续智能体必须遵循先前智能体已作出的决策。然而，这类方法无法体现交通流中不同的社会行为，因为低优先级车辆只能被动响应高优先级车辆作出的决策。此外，为交通流中的每一辆车分配优先级也绝非易事。

相反，我们的方法将交通流中的人类驾驶车辆纳入考虑，并且在决策过程中不设定车辆之间的优先级。我们用能够同时生成多车辆动作的元节点 替换通用蒙特卡洛树搜索方法中的节点。图3展示了元节点的扩展，其中车辆2和车辆3由我们的框架控制，而车辆1则不受控制。

对于扩展至时间步 t的元节点，它会接收当前时刻所有受控车辆的状态。具体而言，车辆 V_i在时间步 t的状态可表示为 $\mathbf{x}_i^t = [s, d, v]^T$、 s和 d分别是该车辆的Frenét坐标， v 是其纵向速度。与 [12], 类似，我们为每辆受控车辆定义了一组可在一个时间步内执行的可能动作。如表1所示，共有五种动作：保持速度（KS）、加速（AC）、减速（DC）、向左变道（LCL）和向右变道（LCR）。通过选择不同的动作，车辆便可计算出其在下一时间步对应的状态。对于交通流中的非受控车辆，我们假设它们执行车道保持（KL）动作。该动作会动态调整其速度，并在决策过程中始终与前车保持适当距离。

表1：车辆在一个时间步内可执行的动作

动作	状态变化	描述
KS	$[v\Delta t, 0, 0]$	保持当前速度
AC	$\left[v\Delta t + \frac{a_{acc}}{2}\Delta t^2, 0, a_{acc}\Delta t\right]$	提高当前速度 以固定加速度 a $_{acc}$
DC	$\left[v\Delta t - \frac{a_{dec}}{2}\Delta t^2, 0, -a_{dec}\Delta t\right]$	降低当前速度 以固定减速度 a_{dec}
LCL	$[v\Delta t, -\Delta d, 0]$	进行部分向左变道 宽度为 Δd
LCR	$[v\Delta t, \Delta d, 0]$	进行部分向右变道 宽度为 Δd
KL	-	保持当前车道 （仅适用于非受控车辆）

由于元节点需要同时为所有车辆做出决策，其输出会组合每辆车辆可能的行为。因此，随着蒙特卡洛树搜索逐步评估整个交互决策空间，搜索树会呈指数级增长。在实践中，找到计算上可行的最佳解变得不可能。因此，有必要对元节点的扩展进行剪枝。

3.1.3 蒙特卡洛树搜索中的剪枝。尽管每辆车辆有五种可选动作，但并非每种动作都可行，有些动作还可能导致潜在碰撞。显然，道路两侧的车辆不能继续向道路边界线外变道。根据 [22], 交通流中的车辆应与其前车保持最短安全距离 D_s。 D_s 由两部分组成：反应距离和最小车头时距，其定义如下：

$$D_s = v \cdot \tau + MTH \cdot \Delta v \tag{2}$$

其中， $\tau > 0$ 和 $MTH > 0$ 均为时间常数，分别表示反应时间和最小车头时距。 v 表示车辆纵向速度， $\Delta v = v - v_l$ 表示本车与前车之间的速度差。此外，交通流中任何单辆车辆都应满足以下速度限制：

$$v \in \left[v_f - \frac{\Delta s_f - \tau \cdot v_f}{MTH}, \min(\frac{MTH \cdot v_l + \Delta s_l}{\tau + MTH}, \frac{\Delta s_l}{\tau})\right] \tag{3}$$

其中， Δs_l和 Δs_f分别表示当前车辆与其前车/后车之间的间距。 v_l和v_f表示前车和后车的速度。值得注意的是，对于速度上限，即使车辆当前速度低于前车速度，也仍需保持最小反应距离。

在时间步 t 的元节点中，当车辆 i 执行表1中的一个动作a，且所得状态 $\mathbf{x}_i^{t+1}$不满足方程（3）时，该动作 a 将变为无效动作，相应的状态 $\mathbf{x}_i^{t+1}$也将被舍弃。此外，对于元节点中的所有有效状态组合，算法会检查每个组合中新状态之间是否发生碰撞，以及到达这些状态的过程中是否存在冲突（例如，两辆车辆驶入彼此的车道）。如果发生冲突，则不应在时间步 $t+1$扩展该动作组合。通过这些技术，决策空间得以显著缩小，同时保证了搜索效率。如图3所示，这三辆车辆最初生成了 $5^2 = 25$ 个可能的动作组合，剪枝后仅剩 4 个有效动作组合。在面对复杂交通状况时，该算法的效率提升更加显著。

3.1.4 具有社会行为的奖励函数。在蒙特卡洛树搜索的仿真阶段，算法会继续构建元节点，但每次仅选择一个有效的动作组合来扩展到下一个时间步，直到仿真终止。当所有车辆都在目标车道行驶了一定距离，或者仿真达到最大时间步而仍有车辆未能完成其意图时，仿真就会终止。随后，应根据仿真结果计算奖励。

我们首先分别计算交通流中每辆车辆的奖励 R_i i。奖励 R_i 由三个部分组成，分别是行驶在目标车道上、沿车道中心线行驶以及保持动作的一致性。此外，奖励的取值范围应为 0 到 1。

为了使车流展现出多样的社会行为，需要通过奖励来刻画车辆的合作倾向。与 [20], 类似，我们引入一种同时考虑自我奖励和他人奖励的行为奖励：

$$R = R_{\text{self}} + \gamma R_{\text{other}} \tag{4}$$

其中 $\gamma \in [0, 1]$ 为合作因子。 $\gamma = 0$ 表示车辆具有自利性，不考虑其他车辆的行为；而 $\gamma = 1$ 表示车辆在决策时将其他车辆视为与自身同等重要。交通流中的每辆车 i都有其独特的因子 γ_i。

由于搜索树中的每个元节点都处理整个车流的动作，因此仿真的奖励由所有车辆的行为奖励组合而成：

$$X_{\text{flow}} = \frac{1}{K} \sum_{1 \le i \le K} \frac{R_i + \gamma_i \sum_{j \neq i} R_j}{1 + (K-1)\gamma_i} \tag{5}$$

由于 $R_i \in [0, 1]$， $X_{\text{交通流}}$的分布也能保证保持在 $[0, 1]$内。最后，在反向传播阶段，使用奖励 $X_{\text{交通流}}$更新所有选定元节点的平均奖励。

3.2 并行车辆轨迹规划

决策模块为每辆车生成粗粒度动作序列，但这些动作没有考虑车辆运动学和道路曲率，因此车辆无法直接执行这些动作。同时，生成的离散动作仍可能使车辆面临潜在碰撞风险。我们在决策之后采用轨迹规划模块，生成连续且运动学可行的轨迹，并确保整个框架的安全底线。

3.2.1 笛卡尔坐标系与*Frenét*坐标系之间的转换。如前所述，为简化计算，我们在Frenét坐标系中进行决策。然而，规划模块生成的轨迹需要转换到全局笛卡尔坐标系，以验证车辆的运动学约束。下面简要介绍如何将Frenét坐标系转换为笛卡尔坐标。

通过由当前车道中心线生成的参考路径构建 Frenét坐标系，该坐标系由沿参考路径的弧长参数 s 和横向偏移 d 表示。 $\dot{s}$ 和 $\dot{d}$ 是对时间的导数。对于一辆车辆

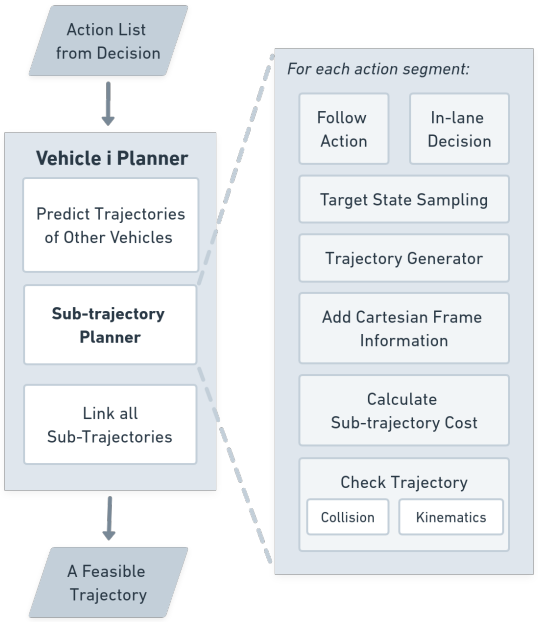

图4：单车轨迹规划器的流程

给定 Frenét坐标 $[s, \dot{s}, d, \dot{d}]$，我们可以根据 [29] 中提出的变换，计算其在笛卡尔坐标系中的位置（x， y）、速度 v 和方向 θ：

$$\begin{cases} x = & x_r - d\sin\theta_r \\ y = & y_r + d\cos\theta_r \\ v = & \sqrt{[(1-\kappa_r d)\,\dot{s}]^2 + \dot{d}^2} \\ \theta = & \arcsin\left(\frac{\dot{d}}{v}\right) + \theta_r \end{cases} \tag{6}$$

其中 (x_r, y_r)、 θ_r和 κ_r分别表示参考路径在长度为 s处的位置、方向和曲率。

3.2.2 可行轨迹规划器。如第2节所述，在规划模块中，我们采用并行架构对每辆车进行轨迹规划，这不仅提高了框架的运行效率，也模拟了人类在现实世界中的驾驶方式。单车轨迹规划器的流程如图4所示。

对于由我们控制的自动驾驶车辆 $V_i \in \mathrm{V}c$，它会从决策模块接收一个动作序列。随后，其中一个并行规划器接管规划任务。规划器首先预测交通流中其他车辆的未来轨迹，然后将序列中的不同动作拆分开来，并为每个动作片段规划一个子轨迹。最后，规划器将所有子轨迹连接起来，得到最终可行且连续的车辆轨迹。

我们的子轨迹规划器与 [29]，提出的方法类似，但流程有所不同。首先，算法根据当前动作片段确定子轨迹目标状态的范围。尽管如表1所示，决策时每个动作对应的状态变化都是固定的，但我们将其扩展为连续的子轨迹目标状态区域

规划。此外，对于执行车道保持动作的车辆，规划器会设置更大的目标状态区域，以便在车道内做出合理决策。随后，规划器在预定义区域内均匀采样若干目标状态。我们在 Frenét坐标系中使用五次多项式曲线，生成起始状态与子轨迹所有采样目标状态之间加加速度最优的连接。之后，根据公式6将每条子轨迹转换到笛卡尔坐标系中，包括其位置、速度和航向角。我们使用代价函数对所有备选子轨迹进行计算，具体内容详见第3.2.3节。我们检查每条子轨迹是否满足运动学约束，以及是否与交通流中的其他车辆无碰撞。满足所有约束且代价最小的子轨迹将被选中，其最终状态则设为下一条子轨迹的初始状态。

3.2.3 考虑驾驶习惯的代价函数。对于每条备选子轨迹，都应计算一个代价函数来评估其安全性和舒适性。具体而言，该函数包含若干代价项：

- 曲线能量代价：$J_{i,\ cur} = \sum \kappa_i^2$, κ_i表示轨迹点的曲率。此惩罚项用于惩罚带有急转弯的轨迹。
- 航向差成本：$J_{i,\ \phi} = \sum |\phi_i - \phi_r|^2$,$\phi_i$表示车辆的朝向角，$\phi_r$表示该点处的道路方向。
- 偏离车道成本：$J_{i,\ out} = \sum |d_i|^2$, d_i表示Frenét坐标系中的横向偏移。除变道动作外，此项仅对其他动作生效。
- 加速度成本：$J_{i,\ acc} = \sum a_i^2$, a_i表示车辆在轨迹点处的加速度。
- 加加速度成本：$J_{i,\ jerk} = \sum j_i^2$, j_i表示车辆在轨迹点处的加加速度（加速度的导数）。此项与加速度成本共同影响轨迹的舒适性。• 动态障碍物成本：$J_{i,\ obs}$。此项用于避免车辆与周围车辆距离过近，具体内容将在下文详细介绍。

动态障碍物成本考虑沿轨迹行驶的车辆与道路上的动态障碍物（即周围车辆）之间的距离。车辆 V_i 的成本定义如下：

$$J_{i,obs} = \sum_{t=0}^{T} \sum_{\forall j \neq i} J_{i,obs}(t, j) \tag{7}$$

其中，$J_{i,\ obs}(t,\ j)$表示时间步 t时车辆 V_j相对于当前车辆 V_i的位置。如图5所示，我们引入一个警戒区，其形状为环绕车辆 V_i车身的矩形。该区域在车辆前方的长度为公式2定义的最短安全距离 D_s，在车辆后方的长度为车身长度的1.5倍。警戒区的宽度为车身宽度的1.5倍。

车辆 V_j在交通流中相对于 V_i的位置可分为三种不同类型：位于警戒区外、位于警戒区内但未发生碰撞，以及与 V_i的当前位置发生碰撞。车辆 V_j在警戒区外是安全的，因此不会产生代价；而其与轨迹的直接碰撞是不可接受的。在警戒区内，车辆 V_j越接近 V_i,

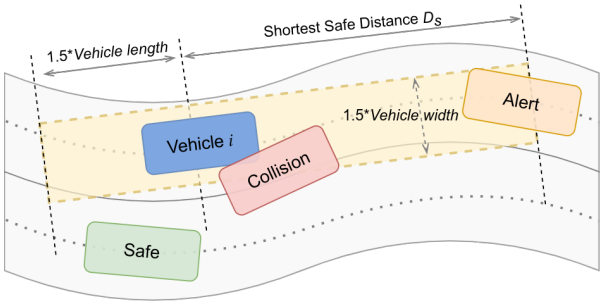

图5：车辆 V_i的警戒区

代价越高。具体而言，$J_{i,\ obs}(t,\ j)$定义如下：

$$J_{i,obs}(t,j) = \begin{cases} 0, & V_j \text{ outside alert zone} \\ C_z(2 - \frac{|D|_x}{D_s+1.5l} - \frac{|D|_y}{1.5d}), & V_j \text{ in alert zone} \\ \infty, & V_j \text{ has collision} \end{cases} \tag{8}$$

其中，$|D|_x$和$|D|_y$分别表示车辆V_j与车辆V_i之间的最短距离在警戒区长度和宽度方向上的分量。l和d表示车身的长度和宽度，$C_z > 0$为成本常数。

上述所有项相加构成代价函数：

$$J_i = \sum w_{(\cdot)} J_{i,(\cdot)} \tag{9}$$

其中$J_{i,\ (\cdot)} \in \{J_{i,\ cur},\ J_{i,\ \phi},\ J_{i,\ out},\ J_{i,\ acc},\ J_{i,\ jerk},\ J_{i,\ obs}\}$表示上述所有代价项，$w_{(\cdot)}$是与各代价项对应的权重参数。我们观察到，交通流中的不同驾驶员可能会根据各自的驾驶习惯选择不同的轨迹。换言之，驾驶员会以不同的方式权衡这些轨迹的代价项。例如，有些驾驶员可能更注重驾驶舒适性，而另一些则倾向于与其他车辆保持更长的安全距离。因此，我们根据驾驶员的驾驶习惯定义一个包含M个权重向量的轨迹权重集W：

$$\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots \mathbf{w}_M\} \tag{10}$$

每个权重向量$\mathbf{w}$都包含所有成本项的唯一权重组合，$\mathbf{w} = [w_{cur},\ w_{\phi},\ w_{out},\ w_{acc},\ w_{jerk},\ w_{obs}]$。交通流中的自动驾驶车辆会选取不同的权重向量来生成其轨迹，并结合决策模块中考虑社会行为的合作因子，从而为整个车流带来多样性。

4 实验

我们利用 CitySim 数据集中的真实驾驶记录 [30]生成实验环境，其中包含三种不同的场景，如图 6 所示：高速公路、匝道和环岛。高速公路场景最为简单，包含一条四车道高速公路。匝道场景包括一条三车道主路，以及一条汇入主路的入口匝道。环岛场景用于评估车辆流在无信号灯的双车道环岛入口处的表现。我们将数据集中的车辆信息作为车辆流的实验初始状态。即将变道（包括驶入和驶出匝道）的车辆，以及与其直接交互的车辆

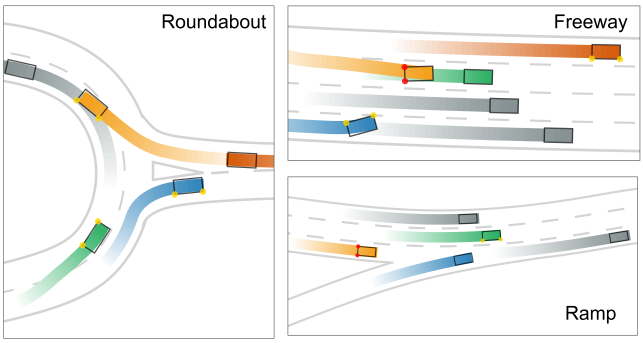

图6：不同场景中的车辆轨迹

表2：三种驾驶行为

Type	激进型	普通型	保守型
γ	0.1	0.5	0.9
$w_{i,obs}$	2.0	4.0	6.0
$w_{i,out}$	6.5	5.0	4.0
$w_{i,jerk}$	0.7	1.0	1.2

被设定为由集中式规划器控制的自动驾驶车辆。相比之下，其他不受影响的车辆则设定为不受控制的人工驾驶车辆。车辆的变道意图作为输入馈入该框架。所提出的框架完全使用 Python 实现，以便快速进行原型开发。

交通流中的车辆尺寸设为 5 × 2 C_p = 1 √2 1 τ = 0.5 m m。公式 . 中的常数 /。我们在公式2中设置反应时间 s 和最小车头时距 MTH = 3s。表1可选动作中的固定加速度a_{acc}和减速度a_{dec}设为 0.6m/s^2。决策模块中的时间步设为 1.5s，规划模块中的时间步设为 0.1s。为生成具有多样性的车流，定义了三种不同类型的驾驶行为：激进型、普通型和保守型。激进型车辆在决策时较少考虑周围车辆，因此其合作因子 γ较小，对避障和加速度变化的容忍度也更高。保守型车辆则相反。具体的风格相关参数定义见表2。

我们提出的框架在这三种场景中均能生成颇具前景的规划方案。图 6 展示了生成的交通轨迹，其中交通流中的受控自动驾驶车辆以彩色轨迹表示，未受控的人工驾驶车辆则以灰色轨迹表示。决策模块生成的可解释动作通过图中的刹车灯和转向灯显示。

在以下各节中，我们将进一步比较和分析所提出框架的性能，以及其生成方案的多样性。

4.1 框架性能

在本小节中，我们测试所提出框架的性能。为评估所提方法的有效性，我们首先实现了一种序贯蒙特卡洛树搜索方法作为对比，

表3：高速公路场景下的对比实验

	2辆受控自动驾驶车辆			3辆受控自动驾驶车辆			4辆受控自动驾驶车辆		
	序贯方法 MCTS	我们的方法（不含）规划	Ours	序贯方法 MCTS	我们的方法（不含）规划	Ours	序贯方法 MCTS	我们的方法（不含）规划	Ours
成功率↑	100%	100%	**100%**	100%	100%	**100%**	70%	90%	**100%**
平均扩展节点数↓	1304.1	1196.6	**1148.7**	1387.6	1127.8	**1037.6**	1706.6	1785.0	**1527.3**
平均完成时间（秒）↓	**3.57**	5.75	5.68	3.57	4.95	**3.50**	**4.58**	9.29	8.93
最小距离（米） ↑	4.34	8.41	**9.13**	4.74	7.02	**10.78**	5.02	7.54	**10.94**

该方法用于 [13]（标记为 序贯蒙特卡洛树搜索）。该方法为交通流中的每辆车分配通行优先级，并在每个决策步骤中按优先级顺序为车辆做出决策。在本实验中，纵向距离较大的车辆具有更高的决策优先级。由于前述研究未考虑场景中的非受控人工驾驶车辆，我们采用跟驰模型 [23] 预测所有人工驾驶车辆的轨迹，并将其作为动态障碍物输入序贯蒙特卡洛树搜索。此外，我们还在对比实验中纳入了一个仅进行决策的框架（标记为 我们的方法（不含）规划），以展示所提框架中规划模块的必要性和有效性。

根据 [23],，序贯蒙特卡洛树搜索中的纵向加速度选为 $[-1, 0, 1]$ 米/秒$_2$，横向速度选为 $[-1.2, 0, 1.2]$米/秒。我们将序贯蒙特卡洛树搜索和纯决策框架的决策时间步长设为1.0秒。三种方法的规划时域均为10秒，蒙特卡洛树搜索的最大迭代次数为3000次。

实验在四车道高速公路场景中进行，可控自动驾驶车辆的数量分别设为 2、3 和 4。每种方法均在相同的实验设置下重复十次。我们测量了每种方法的成功率、蒙特卡洛树搜索中的平均扩展节点数、完成车辆意图的平均时间，以及生成的解决方案中的最小车辆间距。结果如表3所示。

在包含两辆和三辆自动驾驶车辆的高速公路场景中，三种方法的成功率均达到 100%。然而，当受控自动驾驶车辆的数量增至四辆时，序贯蒙特卡洛树搜索的成功率仅为 70%。仅决策方法的成功率为 90%，而我们的框架仍通过了所有测试。由于序贯蒙特卡洛树搜索为交通流中的车辆设定了优先级，优先级较高的车辆在决策时不会向优先级较低的车辆让行。这导致高优先级车辆采取局部最优决策，使后续车辆无法完成其意图。

搜索树中扩展节点的数量用于衡量每种方法的搜索效率。正如我们在第3.1.3节中提到的，我们提出的框架为每辆车辆设置了更少的可选动作，并采用了各种剪枝策略。实验结果表明，与对比方法相比，我们的方法扩展的节点更少，这意味着我们的框架能够更快地搜索到最优解，而不会在不可行动作上浪费动作。

就解的质量而言，尽管序贯蒙特卡洛树搜索生成的解中，车辆完成意图的平均时间更短，但这些解的最小车辆间距也更短，因此风险更高。

序贯蒙特卡洛树搜索规划出的解中，车辆之间的最短距离通常小于 5 米（即车辆长度），很容易导致追尾碰撞。相比之下，我们的决策模块通过引入最短安全距离 D_s，将最小距离保持在七至八米。规划模块进一步优化轨迹，使整个方法的最小距离达到约 10 米，即车辆长度的两倍。此外，与仅决策方法相比，整个框架的平均意图完成时间更短；在 3辆车的情况下，表现甚至优于序贯蒙特卡洛树搜索。

4.2 闭环仿真

我们在匝道场景中进行了多次闭环仿真，以研究车辆轨迹在社会行为和驾驶习惯方面的多样性。我们还展示了所提出框架在面对人类驾驶员出乎意料的动作时的响应能力。

闭环仿真以10Hz运行。我们的决策模块时域为10.5秒，每个动作持续1.5秒。规划模块的时域为3秒，时间步为0.1秒，重规划周期为0.5秒。我们的测试场景如图8所示。编号为2、4和5的车辆是由我们的框架控制的自动驾驶车辆，而编号为1、3和6的车辆是环境中的人工驾驶车辆。车辆2和5在主路的相邻车道上并排行驶，而车辆4位于匝道上，即将汇入主路。此外，车辆2打算向右侧变道，从而形成一个复杂场景：交通流中的三辆车同时争夺最右侧车道的通行权。图7展示了由这一相同初始状态产生的三种不同情况。每种情况下受控自动驾驶车辆的速度曲线显示在图的右侧。

在案例 1 中，车辆 2 被设定为激进驾驶行为，车辆 4 采用保守驾驶行为，车辆 5 则被设定为正常行为。它们的风格相关参数根据表2进行设置。闭环实验结果表明，由于初始位置靠后，车辆 5 减速让行车辆 4；车辆 4 保持初始速度并汇入主路。在时刻 T_1，由于车辆 2 的驾驶行为更为激进，因此不太可能向车辆 4 让行。车辆 2 继续行驶在车辆 4 的左前方，并示意其变道意图。最后，车辆 2 减速，为车辆 4 留出空间，使其在 T_2完成变道。

我们改变了案例2中自动驾驶车辆的驾驶行为。车辆 4 的驾驶行为调整为激进型，车辆 2 则调整为

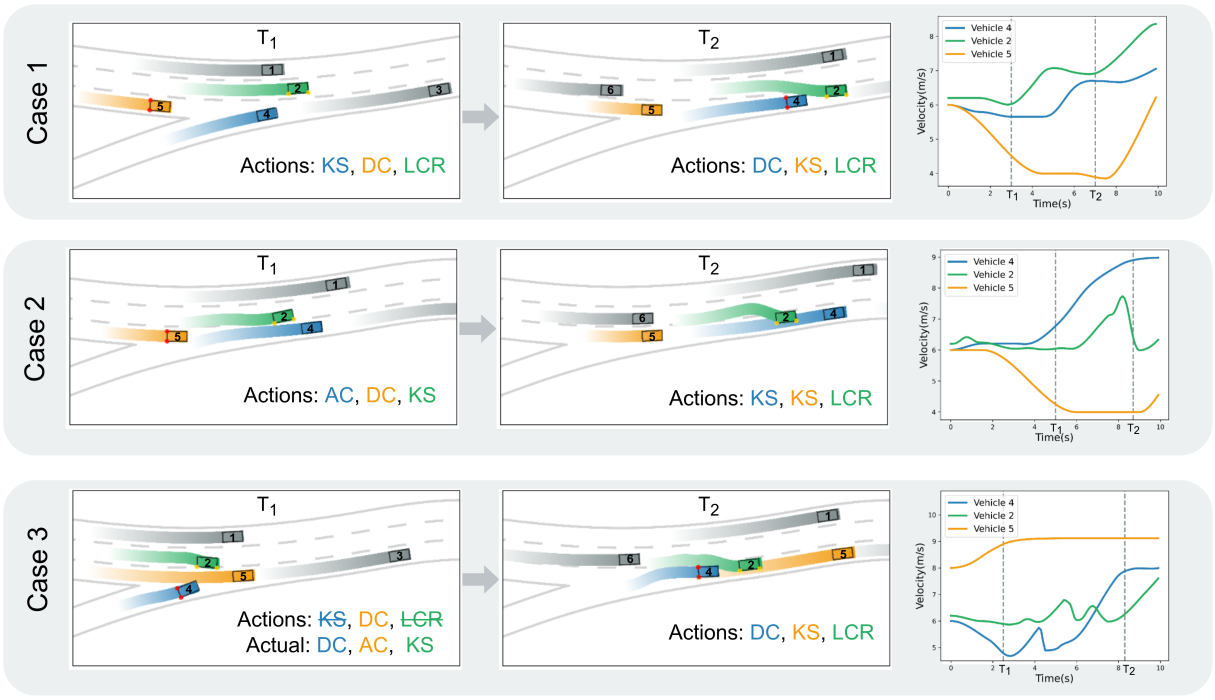

图 7：闭环仿真中的三个不同案例

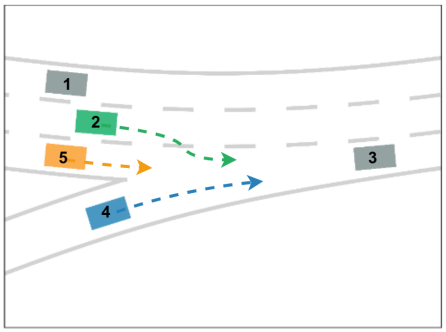

图 8：闭环仿真中的车辆初始状态

保守型。车辆 5 的驾驶行为保持不变。车辆 5 仍会减速，让车辆 4 在 T_1并入主路，但这次车辆 4 没有让行车辆 2 的变道，而是选择加速通过。车辆 2 注意到车辆 4 的加速度后，选择减速，并在车辆 4 之后完成变道动作，这与案例 1 截然不同。

在案例 3 中，我们探讨了当环境中一辆不可控的人工驾驶车辆不配合自动驾驶车辆的行为时会发生什么。我们将车辆 5 从自动驾驶车辆改为不受我们的规划器控制的人工驾驶车辆，并为其设置更高的初始速度。我们的框架原本预计它会减速让行车辆 2，但它却加速驶入匝道入口，造成了相当危险的局面。得益于规划模块中的高频重规划，当车辆 2 和车辆 4 发现车辆 5 正在加速时，它们便放弃了决策模块给出的原有动作，以避免潜在碰撞。车辆 4 进行了减速，车辆 2 则保持在原车道上。车辆 5 通过后，车辆 2 和车辆 4 随即安全地完成变道和汇入。这一案例展示了将规划模块集成到框架中的重要性：它能够迅速响应道路状况的变化，化解车辆之间的潜在碰撞，从而保障整个车流的安全。

5 结论

本文提出了一种新颖的分层多车决策与规划框架，可在自动驾驶车辆与人工驾驶车辆混合场景中生成兼具多样性、交互性和可解释性的轨迹。通过在决策模块中引入车辆的合作因子，并在规划模块中引入轨迹权重集，我们的框架生成的轨迹在社会交互和个体习惯两个层面上都具有多样性。我们的框架能够为交通流中的所有车辆联合制定决策，并通过高频重规划方法确保实时响应动态环境。此外，决策模块还会生成可解释的动作序列，用于向乘客告知当前情况，并向周围的人工驾驶车辆明确传达自身意图。

在后续工作中，我们的框架可用于在自动驾驶仿真器中生成自车周围的背景交通。它还可用于对现有交通数据集进行数据增强，以涵盖不同的驾驶风格。此外，我们的框架具有通用性，这意味着只要各模块的输入和输出符合要求，就可以替换其内部算法。

参考文献

[1] Cameron B Browne、Edward Powley、Daniel Whitehouse、Simon M Lucas、Peter I Cowling、Philipp Rohlfshagen、Stephen Tavener、Diego Perez、Spyridon Samoth- rakis 和 Simon Colton。2012年。《蒙特卡洛树搜索方法综述》。*IEEE*计算智能与游戏人工智能汇刊 4, 1（2012年3月），1 43 10 1109 2012 2186810 – 。https://doi.org/ 。/TCIAIG. 。

[2] Sergio Casas、Wenjie Luo 和 Raquel Urtasun。2021年。IntentNet：从原始传感器数据中学习预测意图。CoRL（2021年1月）。arXiv:2101.07907 http://arxiv.org/abs/2101 07907 。

[3] Dian Chen 和 Philipp Krähenbühl。2022年。从所有车辆中学习。载于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）论文集。17222–17231。

[4] Laurene Claussmann、Marc Revilloud、Dominique Gruyer 和 Sébastien Glaser。2019年。高速公路自动驾驶运动规划综述。*IEEE*智能交通系统汇刊 21, 5（2019年），1826–1848。

[5] Wenchao Ding、Lu Zhang、Jing Chen 和 Shaojie Shen。2019年。利用时空语义走廊为复杂城市环境生成安全轨迹。*IEEE*机器人与自动化快报 4, 3（2019年），2997–3004。https: //doi.org/10.1109/LRA.2019.2923954

[6] Alexey Dosovitskiy、German Ros、Felipe Codevilla、Antonio Lopez 和 Vladlen Koltun。2017年。CARLA：一款开放式城市驾驶模拟器。载于第一届机器人学习年会论文集。1–16。

[7] Haoyang Fan、Fan Zhu、Changchun Liu、Liangliang Zhang、Li Zhuang、Dong Li、Weicheng Zhu、Jiangtao Hu、Hongye Li 和 Qi Kong。2018年。百度Apollo EM运动规划器。（2018年）。arXiv:1807.08048

[8] Jaime F Fisac、Eli Bronstein、Elis Stefansson、Dorsa Sadigh、S Shankar Sastry 和 Anca D Dragan。2019年。面向自动驾驶车辆的分层博弈论规划。载于*2019*年国际机器人与自动化会议（*ICRA*）。IEEE，9590–9596。

[9] Peng Hang、Chao Huang、Zhongxu Hu 和 Chen Lv。2022年。无信号交叉路口自动驾驶车辆的驾驶冲突解决：一种微分博弈方法。*IEEE/ASME*机电一体化汇刊 27, 6（2022年），5136–5146。

[10] Thomas M. Howard 和 Alonzo Kelly。2007年。轮式移动机器人崎岖地形的最优轨迹生成。国际机器人研究杂志 26, 2 2007 141 166 10 1177 0278364906075328 ( )，– 。

[11] https://doi.org/ 。/ Christos Katrakazas、Mohammed Quddus、Wen-Hua Chen 和 Lipika Deka。2015年。自主道路驾驶的实时运动规划方法：最新进展与未来研究方向。《交通研究*C*辑：新兴技术》60 2015 416 442 10 1016 2015 09 011 ( )，– 。

[12] https://doi.org/ 。/j.trc. 。David Lenz、Tobias Kessler 和 Alois Knoll。 。采用蒙特卡洛树搜索的高速公路自动驾驶战术协同规划。*IEEE*智能车辆研讨会论文集 2016-Augus，IV（2016年），447–453。https://doi.org/ 10.1109/IVS.2016.7535424 2016

[13] Chenran Li、Tu Trinh、Letian Wang、Changliu Liu、Masayoshi Tomizuka 和 Wei Zhan。2022年。结合预测启发式的高效博弈论规划，实现符合社会规范的自动驾驶。*IEEE*机器人与自动化快报 7, 4 2022 10248 10255 10 1109 2022 3191241 ( )，– 。

[14] https://doi.org/ 。/LRA. 。Pablo Alvarez Lopez、Michael Behrisch、Laura Bieker-Walz、Jakob Erdmann、Yun-Pang Flötteröd、Robert Hilbrich、Leonhard Lücken、Johannes Rummel、Peter Wagner 和 Evamarie Wießner。2018年。使用SUMO进行微观交通仿真，载于第21届IEEE智能交通系统国际会议。*IEEE*智能交通系统会议（*ITSC*）。https://elib. dlr.de/124092/

[15] Michael Montemerlo、Jan Becker、Suhrid Bhat、Hendrik Dahlkamp、Dmitri Dolgov、Scott Ettinger、Dirk Haehnel、Tim Hilden、Gabe Hoffmann、Burkhard Huhnke 等。2008年。Junior：斯坦福大学参加城市挑战赛的参赛项目。《机器人领域期刊》25，9（2008年），569–597。

[16] Zhenghao Peng、Quanyi Li、Ka Ming Hui、Chunxiao Liu 和 Bolei Zhou。2021年。通过协同策略优化学习仿真自驱动粒子系统。神经信息处理系统进展34（2021年），10784–10797。

[17] Thomas A Ranney。1994年。驾驶行为模型：其演变综述。《事故分析与预防》26，6（1994年），733–750。

[18] Vladimir Y. Rovenski。2006年。曲线与曲面的微分几何：简明指南。Birkhäuser，波士顿，马萨诸塞州。https://doi.org/10 1007 137116 。/b Dorsa Sadigh、Shankar Sastry、Sanjit A Seshia 和 Anca D Dragan。2016年。规划能够利用对人类动作影响的自动驾驶汽车。载于机器人学：科学与系统，第2卷。美国密歇根州安娜堡，1–9。

[19] Wilko Schwarting、Alyssa Pierson、Javier Alonso-Mora、Sertac Karaman 和 Daniela Rus。2019年。自动驾驶车辆的社会行为。《美国国家科学院院刊》116，50（2019年），2492 24978 10 1073 1820676116 –。https://doi.org/。/pnas。

[21] Qiao Sun、Xin Huang、Junru Gu、Brian C Williams 和 Hang Zhao。2022年。M2I：从因子化边缘轨迹预测到交互式预测。载于*IEEE/CVF*计算机视觉与模式识别会议论文集。6543–6552。

[22] D. Swaroop 和 K. R. Rajagopal。2001年。自动车辆跟驰的恒定时距策略综述。载于*IEEE*智能交通系统会议论文集，*ITSC*。IEEE，奥克兰，加利福尼亚州，65–69。https://doi.org/10.1109/itsc.2001.948631

[23] Martin Treiber、Ansgar Hennecke 和 Dirk Helbing。2000年。实证观测与微观仿真中的拥堵交通状态。《物理评论*E*》62，2（2000年），1805。

[24] Chris Urmson、Joshua Anhalt、Drew Bagnell、Christopher Baker、Robert Bittner、MN Clark、John Dolan、Dave Duggins、Tugrul Galatali、Chris Geyer 等。2008年。城市环境中的自动驾驶：Boss与城市挑战赛。《机器人领域期刊》25，8（2008年），425–466。

[25] Vladimir Vapnik。2006年。《基于经验数据的依赖关系估计》。施普林格，纽约，纽约州。https://doi.org/10.1007/0-387-34239-7

[26] Junhua Wang、Wenxiang Xu、Ting Fu、Hongren Gong、Qiangqiang Shangguan 和 Anae Sobhani。2022年。基于图构建的激进驾驶行为建模。《交通研究*C*辑：新兴技术》138（2022年），103654。

[27] Wenshuo Wang、Letian Wang、Chengyuan Zhang、Changliu Liu 和 Lijun Sun。2022年。自动驾驶中的社会交互：综述与展望。《机器人学基础与趋势》10，3–4（2022年），198–376。https://doi.org/10.1561/2300000078

[28] Yunpeng Wang、Junjie Zhang 和 Guangquan Lu。2018年。驾驶行为对车辆跟驰稳定性的影响。*IEEE*智能交通系统汇刊20，3（2018年），1081–1098。

[29] Moritz Werling、Julius Ziegler、Sören Kammel 和 Sebastian Thrun。2010年。Frenét坐标系下动态道路场景的最优轨迹生成。载于*2010*年*IEEE*机器人与自动化国际会议。IEEE，美国阿拉斯加州安克雷奇，987 993 10 1109 2010 5509799 –。https://doi.org/。/ROBOT。。

[30] Ou Zheng、Mohamed Abdel-Aty、Lishengsa Yue、Amr Abdelraouf、Zijin Wang 和 Nada Mahmoud。2022年。CitySim：面向安全研究与数字孪生的基于无人机的车辆轨迹数据集。https://doi.org/10.48550/ARXIV.2208.11036