arXiv:2304.00365v1 [cs.RO] 1 Apr 2023

利用领域专家的关键状态进行自适应故障搜索

Peter Du 和 Katherine Driggs-Campbell

摘要—**发现潜在故障案例是验证自动驾驶汽车等安全关键系统的重要步骤。故障搜索可以通过在仿真环境或真实世界测试中记录大量车辆行驶里程来完成。由于故障事件十分稀少，朴素的随机搜索方法需要大量车辆运行时间才能发现潜在的系统弱点。因此，人们提出了自适应搜索技术，以便在仿真环境中高效探索并发现自动驾驶策略的失败轨迹。自适应压力测试（AST）就是此类方法之一，它将故障搜索问题表述为马尔可夫决策过程，并利用强化学习技术寻找高概率故障。然而，这种表述要求建立环境中所有智能体动作的概率模型。在环境动作是离散的且智能体之间存在依赖关系的系统中，完全刻画其分布或找到合适的代理指标可能并不可行。本文提出采用数据驱动方法，学习一个合适的分类器，尝试对人类识别关键状态的方式进行建模，并利用该分类器指导 AST 中的故障搜索。我们表明，在 AST 框架中引入关键状态后，可以在具有离散动作空间的自动驾驶策略中生成安全违规程度更高的故障场景。**

I. 引言

自主系统正迅速发展为能够部署于现实世界中的实用技术。在自动驾驶等安全关键型应用中，部署前必须进行大量测试，以确保使用该技术的人类以及环境中的弱势参与者的安全。由于自动驾驶汽车（AV）在公共道路上行驶时可能面临各种各样的场景，因此，验证自动驾驶汽车仍然是这些系统设计者面临的挑战。实体车辆测试是提高对自动驾驶汽车在现实世界中安全运行能力的置信度的一种常用方法 [1]；然而，现实世界车辆测试所需的资金和时间成本，使得全面验证自动驾驶汽车可能遇到的所有场景变得困难 [2]。仿真可用于补充实体车辆测试，让自动驾驶汽车设计者能够长时间运行一支规模庞大的车队 [3]。

然而，即便仿真提升了测试能力，危险场景十分罕见这一事实仍可能导致朴素的随机搜索技术无法充分覆盖故障空间。因此，人们探索了多种方法来生成可能导致自动驾驶汽车故障的关键场景。深度生成模型，例如 [4] 和 [5]，分别采用基于流的方法和变分自编码器模型，以生成富有信息量的驾驶场景。人们提出了自适应采样方法，以搜索可能导致自动驾驶汽车故障的关键场景 [6]–[9]。

P. Du 和 K. Driggs-Campbell 任职于伊利诺伊大学厄巴纳-香槟分校电气与计算机工程系。电子邮箱：{peterdu2,krdc}@illinois.edu

自适应压力测试（AST）是一种通过仿真来发现自主系统故障的技术 [10]。漏洞搜索被表述为寻找系统最有可能发生的故障，并建模为马尔可夫决策过程（MDP）。研究人员使用强化学习技术自适应地对环境进行采样，并求解该 MDP。在标准表述中，AST 的奖励函数取决于模拟器中状态转移的对数似然，并假设环境动作独立同分布，因此会优化策略以生成最有可能发生的故障。强化学习求解器从连续空间中采样动作，并且通常使用马氏距离作为动作概率的代理指标 [11]。然而，对于离散动作空间的高层决策系统验证而言，这种表述并不适用，因为这类系统通常无法直接获得类似的代理指标，或者可能需要手工设计该指标。在这些情况下，我们可能希望寻找一种能够生成最坏情况故障或风险敏感型故障的策略。为此，可以在奖励函数[12], 中使用训练启发式方法代替似然，但这会使得设计合适的启发式方法变得困难，同时还不能限制故障的范围。

我们试图通过不再采用人工设计的启发式方法，转而研究关键状态的特征，来解决这一问题。我们假设，故障场景可能是不安全 状态共同作用的结果；智能体表现出更多此类不安全行为的场景，有助于开展安全验证。已有多种度量指标可用于表征机器人状态的关键性，例如考察其风险度量或效用值的分布 [13], [14]。我们将关键状态的识别视为一项分类任务，并采用数据驱动方法训练一个合适的分类器，以预测系统状态是否危险。使用这一学习得到的模型，AST 能够为具有离散高层动作空间的黑盒系统高效生成不安全行为更多的失败轨迹。

我们提出以下贡献：1）讨论 AST 应用于离散动作空间系统时的局限性，并提出一种融入关键状态的框架，以更好地识别有意义的故障。2）我们考虑了关键状态的两种定义。第一种仅根据自主智能体的观测和策略来定义关键性。第二种

则从外部观察者的视角出发，使用对整个环境的观测。

3）我们将故障生成方法应用于一种经过训练、能够输出离散驾驶决策的自动驾驶车辆策略。我们展示了该方法在捕捉自动驾驶汽车行为不当和采取危险动作更多的场景方面的有效性。

本文组织如下：第 II 节概述 AST，并介绍强化学习策略中关键状态的特征。第 III 节描述如何将基于学习的关键状态分类器纳入自适应压力测试框架。第 IV 节和第 V 节介绍对 DQN 自动驾驶策略进行验证的实验及结果。我们在第 VI 节总结全文并讨论未来的研究方向。

II. 背景

本节概述自适应压力测试方法及关键状态的特征。

A. 自适应压力测试

自适应压力测试（AST）是一种将验证和故障搜索问题建模为马尔可夫决策过程，并运用强化学习方法生成策略以寻找被测系统（SUT）可能失效模式的技术。AST 的模块化特性使用户能够采用各种不同的求解器，并将该技术用于验证多种系统。在以往的研究中，AST 已应用于飞机防撞和自动驾驶策略[10], [15]。

AST 由三个主要组件模块组成，分别是模拟器 $\mathscr{S}$、奖励函数 R 和强化学习求解器 S。图 1 展示了该架构的示意图。模拟器 $\mathscr{S}$ 包含被测系统和其运行所在的环境。环境动作 $\mathcal{A}$ 用于推动模拟器随时间向前运行。以自动驾驶汽车应用为例，被测系统可以是在自动控制下运行的自车，而环境动作可能包括自动驾驶汽车上的传感器噪声、周围车辆的加速/制动、行人的运动等。模拟器定义了一组目标状态 E，用于表示被测系统发生的故障。在我们的自动驾驶汽车场景中，这包括自车与另一个智能体发生碰撞的所有状态。

强化学习求解器 S 生成环境动作，用于更新模拟器 $\mathscr{S}$ 的状态。AST 允许 $\mathscr{S}$ 采用黑盒或半黑盒形式，并不要求模拟器公开其完整的内部状态，只要其满足以下接口：

- Init($\mathscr{S}$)：将模拟器重置为初始状态。
- Step($\mathscr{S}, a$)：执行操作 a，并使模拟器向前推进一步。返回一个指示，用于说明模拟器的新状态是否属于目标状态集合 E。
- IsTerminal($\mathscr{S}$)：检查当前状态是否属于 E，或是否已达到时间范围。

进入新状态后，模拟器会将任何相关的状态/观测信息传递给 AST奖励函数 R，该函数会生成奖励信号。奖励会返回给求解器，并在生成下一个环境动作之前更新策略。奖励函数定义如下：

$$R(s) = \begin{cases} 0 & s \in E \\ -\alpha - \beta f(s) & s \notin E, t \geq T \\ -g(a) - \eta h(s) & s \notin E, t < T \end{cases}$$

其中：

- s是模拟器当前的状态/观测。
- E 是目标状态集合（系统故障）。
- T 是仿真的时间范围。
- α 是用于惩罚未以失败结束的轨迹的常数。
- $\beta f(s)$是一种可选的启发式方法，用于区分未以失败结束的轨迹。
- $g(a)$是当前动作的奖励（与对数概率成正比）。
- $\eta h(s)$是每个训练步骤的启发式奖励。

AST框架支持使用各种强化学习求解器。在本研究中，我们使用蒙特卡洛树搜索（MCTS）作为强化学习策略 [16]。在结合AST使用MCTS寻找失败轨迹方面，我们在以往的研究中取得了成功 [15], [17], [18]。

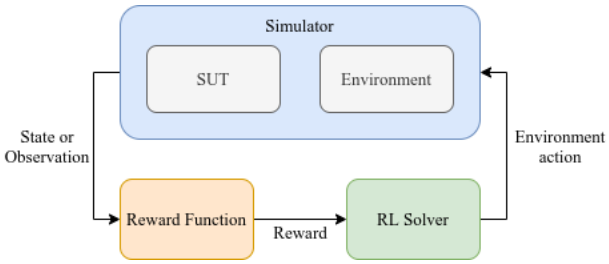


图 1：自适应压力测试架构。

B. 关键状态分类

基于学习的方法已被用于为自主系统生成复杂策略。例如，深度强化学习（DRL）可用于执行自动驾驶任务，也可以训练神经网络来执行视觉运动任务。 [19], [20]。在这些黑盒系统中，神经网络通常采用端到端方式训练，因此难以全面描述系统行为，因为不可能检查自主智能体在所有可能状态下的精确动作。为了解决这一问题，Huang 等人提出，我们可以检查一个规模较小的关键状态集合 [14]。关键状态是指智能体动作的选择对结果的期望效用值产生显著更大影响的状态。更具体地说，如果最大动作价值对（Q 值）比平均动作价值对高出某个阈值t，则可将相应状态界定为关键状态。直观而言，这一定义将关键状态视为这样的情景：智能体策略认为，随机行动的表现会远逊于最优行动。

如果被测系统由能够学习价值或动作价值对的策略构成，我们可以在 AST奖励函数中使用这种基于 Q 值的关键状态形式，取代人工设计的启发式方法。不过，我们认为，这种关键状态定义可能并不适用于失败轨迹搜索，因为并非所有结果都具有危险性。例如，在自动驾驶场景中，自车跟在一辆汽车后方，左右两侧还有其他汽车，此时最优动作（保持恒定速度）可能具有显著更高的动作价值，因为其他动作都会导致碰撞（向左或向右变道、加速等）。采用基于价值的关键状态界定方式，这种情景会被标记为关键状态。然而，鉴于跟车距离处于可接受范围，该状态可能并不危险。同样，在碰撞迫在眉睫的情景中，由于动作选择对结果影响甚微，该状态下所有动作的动作价值对可能都相近。在这种情况下，尽管情景很危险，且应在 AST奖励函数中得到适当权重，该状态却不会被归类为关键状态。

为了捕捉有助于失败场景识别的适当关键状态，我们认为，关键性不应仅从被测系统的角度定义，还应纳入环境其余部分的特征。为此，我们提出直接从人类观察中学习，以识别关键状态。

III. 方法

本节介绍关键状态识别任务及其与 AST奖励函数的集成。

A. 基于人类的关键状态（*HCS*）分类器

HCS分类器的训练数据可以通过随机仿真获得的轨迹，或通过采用纯启发式奖励函数的 AST 生成的样本轨迹来收集。使用带有启发式奖励的 AST 有助于提高数据收集效率，并生成更多失败/危险状态。数据收集完成后，将每条轨迹划分为一组状态，并根据人类响应进行标注。状态可以表示为包含模拟器内部状态（对于白盒/半黑盒模拟器）、自车的观测，或两者混合的向量。已标注数据被拆分为仅包含正样本和负样本的集合。随后随机打乱这两个集合并将其合并，以包含数量均衡的正负标签。

训练数据集 $D = \{(s_1, l_1), (s_2, l_2) ...(s_n, l_n)\}$由状态/标签对组成，其中：

- $s_i \in \mathbb{R}^d$ 是一个状态，包含环境动作（如第 II-A 节所定义）以及自车的观测/动作。
- $l_i \in \{0, 1\}$ 是一个人工标注指标，用于表示该场景是否危险。

给定数据集 D，我们学习一个映射 $\mathcal{F}: \mathbb{R}^d \to [0, 1]$，用于预测每个状态的分数。我们将其视为软分类任务，并使用贝叶斯神经网络进行拟合

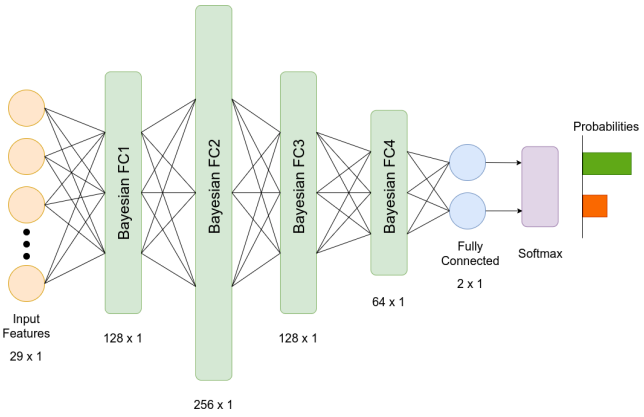


图 2：HCS分类器的网络架构。

在训练数据集上使用二元交叉熵损失进行训练。该网络接收状态 s 的特征向量，并输出一个 2x1 向量，其中各项分别对应于该状态属于危险状态或非危险状态的概率。采用软分类和贝叶斯网络，使我们能够刻画 HCS 分类器所给出预测的不确定性，并将其纳入 AST 奖励函数。

*B. AST*奖励函数

训练完成后，我们可以使用 HCS分类器推断给定状态被预测为危险状态的概率分布。在故障搜索过程中，与当前状态 s 对应的每个特征向量都会通过贝叶斯网络 n 次。设 μ 和 σ^2 分别为所获得分布的均值和方差。随后，时间 t 处状态 s的 AST奖励函数定义如下：

$$R_{hcs}(s) = \begin{cases} 0 & s \in E \\ -\infty & s \notin E, t \geq T \\ (1-\sigma^2)\beta\mu + \sigma^2 h(s) & s \notin E, t < T \end{cases}$$

此处，E 是用户定义的失效状态集合，T 是有限时间范围。$h(s)$ 是一个启发式度量，表示自车与其前方同一车道内最近车辆之间的纵向距离。β 是一个用户可调常数，用于调整关键状态所给奖励的权重，其数值应与启发式度量处于相近量级。

该奖励将预测分布的方差作为 HCS 分类器置信度的代理指标，以调整启发式方法与分类器所获奖励的贡献比例。这种混合形式的奖励函数使我们能够应对分类器对其预测有效性置信度较低的情况。在这些情况下，HCS 分类器的权重会降低，从而避免向强化学习求解器提供误导性的奖励信号。

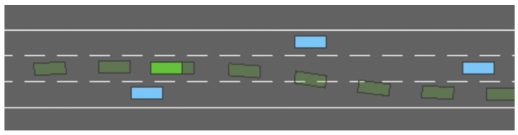


图3：自动驾驶汽车高速公路仿真器 [21]。

IV. 实验

本节介绍用于评估 AST 在 HCS 奖励下的性能以及各种基线方法的实验设置和场景。

用于测试自动驾驶汽车的场景由一条三车道直线高速公路组成，路上还有以不同速度行驶的其他车辆。我们使用 Highway-Env OpenAI Gym环境训练车辆控制器，其目标是在避免与周围车辆发生碰撞的同时达到较高速度 [21]。我们使用 Stable-Baselines库训练了一个深度Q网络（DQN）车辆控制器 [22]。图3展示了仿真器环境的示例。

我们围绕 Highway-Env Gym环境构建了一个 AST接口，并将所需组件暴露出来，如第 II-A 节所述。这里，被测系统是由 DQN控制器引导的自动驾驶汽车。车辆（包括自车和环境车辆）的动作空间由五种高级动作组成：左侧变道、右侧变道、怠速、加速和制动。在故障搜索期间，AST 可直接控制环境动作，这些动作决定了自车周围最近的八辆车的行为。每个环境动作都表示为一个 8 元组：

$$a_{env} = (a^{(1)}, a^{(2)}, a^{(3)}, ...a^{(8)})$$

其中，$a^{(i)}$ 是第 i近环境车辆的高级动作。

作为 HCS分类器输入的状态被定义为一个三元组：

$$s = (a_{ego}, a_{env}, o_{ego})$$

其中，a_{ego} 是自车的动作，a_{env} 是环境动作，o_{ego} 是自车的观测。该观测与 DQN控制器使用的观测相同，由自车周围五辆最近车辆的 x 位置、y 位置、x 速度和 y 速度组成。

我们使用 AST，通过一个初始化了 40 辆车的模拟器和一个 MCTS 求解器，寻找 DQN 驾驶策略的故障场景。我们考虑了两种基线方法。第一种使用依赖于启发式度量 $h(s)$（如第三节 B 部分所定义）的奖励函数来训练求解器。奖励定义如下：

$$R_{heur}(s) = \begin{cases} 0 & s \in E \\ -\infty & s \notin E, t \geq T \\ h(s) & s \notin E, t < T \end{cases}$$

第二种使用 Q 值关键状态判据，如第二节 B 部分所述。奖励函数定义如下：

$$R_{qcs}(s) = \begin{cases} 0 & s \in E \\ -\infty & s \notin E, t \geq T \\ \max\limits_{a} Q(s,a) - \frac{1}{|\mathcal{A}|}\sum\limits_{a} Q(s,a) & s \notin E, t < T \end{cases}$$

我们将这些基线方法与第 III-B 节定义的 HCS奖励进行比较，该奖励使用经过训练的分类器。

V. 结果

本节展示使用 AST 对第 IV 节所示自动驾驶智能体进行验证的结果。我们使用责任敏感安全（RSS）评估结果的质量：这是一组用于驾驶安全保障的可解释数学模型 [23]。对于每种奖励，我们都会分析发现的失败轨迹，以确定 RSS 判定为不安全的状态所占比例。我们还会对结果进行定性评估。

A. Q值关键状态奖励

采用基于Q值的关键状态（QCS）奖励的AST，为高速公路驾驶场景生成故障案例。图4显示了根据RSS，在使用基于Q值的关键状态判据（R_{qcs}）找到的失败轨迹中不安全状态所占的比例。这些轨迹与使用启发式奖励（R_{heur}）获得的失败轨迹进行了比较。可以看出，这一定义导致存在RSS违规的轨迹比例略有增加；然而，总体分布仍然具有可比性，在相近位置出现了两个明显的峰值。

使用 Q 值方差对关键状态进行分类的方法，并未显著提升通过 AST 找到的失败轨迹的质量。我们观察到

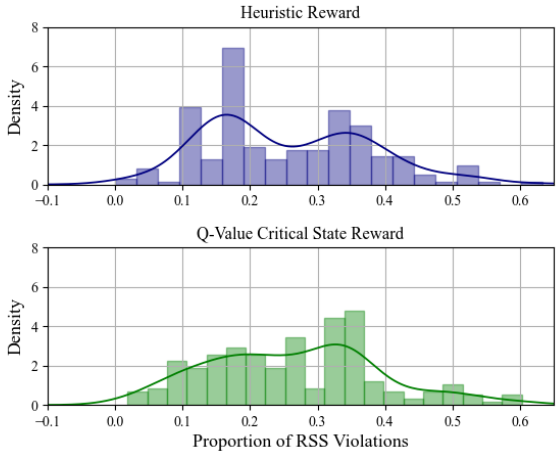


图4：使用Q值关键状态指标找到的失败轨迹中RSS违规比例的归一化直方图。

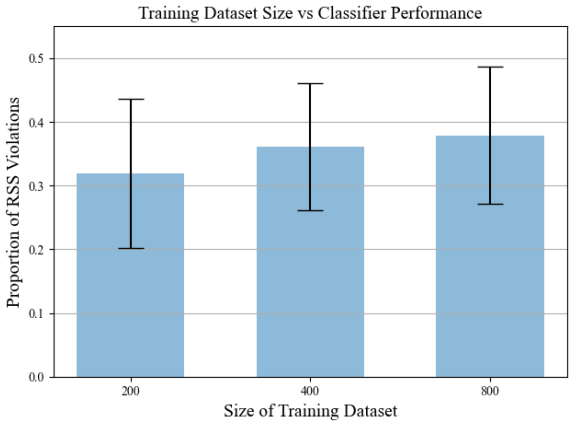


图 5：采用不同数据集规模训练的分类器在使用 HCS奖励时的 AST 性能。

，在失败轨迹搜索的背景下，仅根据自车控制器的动作价值对刻画奖励，不足以识别出更多会导致危险行为的动作（与 R_{heur} 相比）。在许多情况下，控制器可能只有一小部分明显更优的动作，但这些状态无法涵盖所有表现出危险行为的场景。

B. 人工关键状态奖励

我们在相同的高速公路驾驶场景和 MCTS 求解器中，使用 HCS奖励运行 AST 故障搜索。图 5 对比了使用不断增大的数据集规模训练的分类器所对应的 AST 性能。通过随机仿真以及采用启发式奖励的 AST 收集分类器的训练数据，我们发现，只需数量相当可控的人工标注样本，分类器的性能便会趋于稳定。

与之前一样，我们可以通过考察被判定为不安全的状态比例，对比使用 HCS奖励和基线启发式奖励找到的失败轨迹结果

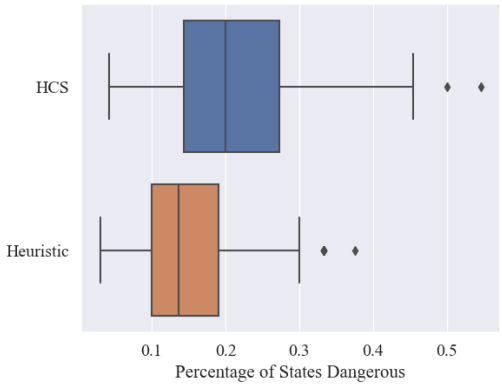
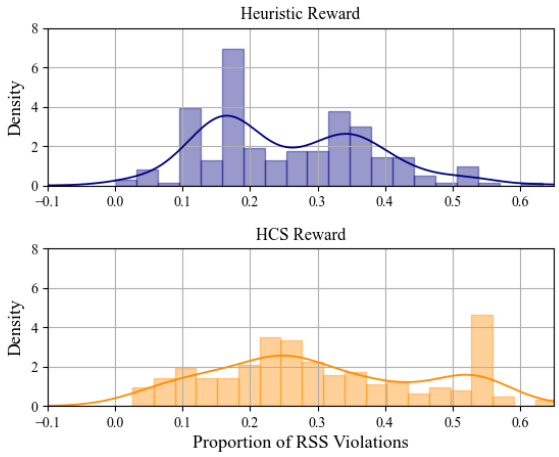
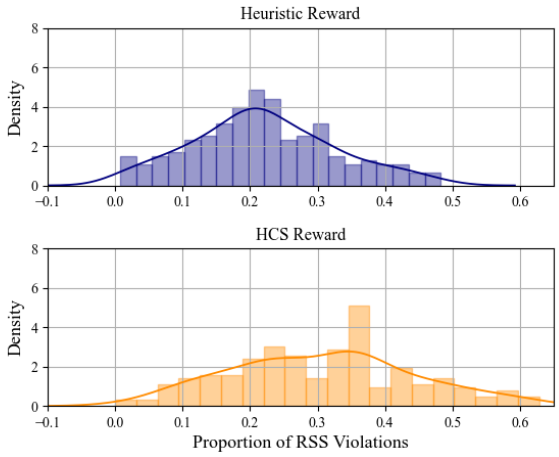


图 6：箱线图展示了使用 HCS奖励和启发式奖励找到的失败轨迹中，被人工标注为关键状态的状态比例分布。



(a)



(b)

图 7：使用和未使用 HCS分类器找到的失败轨迹中，RSS 违规比例的归一化直方图。(a) 和 (b) 中的图分别代表两种不同的初始模拟器配置。

由 RSS 计算。启发式方法与 HCS奖励使用相同的初始条件和模拟器配置。图 7 展示了两种不同初始仿真器状态下 RSS违规的分布。我们注意到，基于自车与最近环境车辆之间纵向间距的启发式方法，应该很适合用于评估 RSS 标准，因为该标准考察的是纵向不安全场景。尽管如此，我们发现使用 HCS奖励时，RSS违规的比例明显增加。例如，在第二种场景中，不当响应比例的中位数增加了 $\sim 50\%$，从 21% 升至 32%（图 7b）。

在图 7a 中，失败轨迹显示了一个初始模拟器配置，该配置会导致 RSS违规呈多峰分布；而图 7b 则显示了一个会导致单峰分布的配置。在这两种情况下，使用 HCS奖励都会使分布的众数向右移动

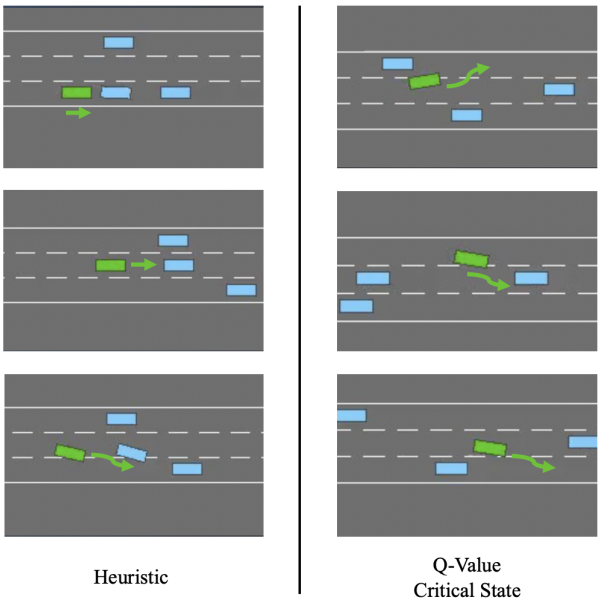


图 8：从启发式奖励（左）和 Q值奖励（右）的 RSS违规分布峰值中采样得到的轨迹的关键场景。

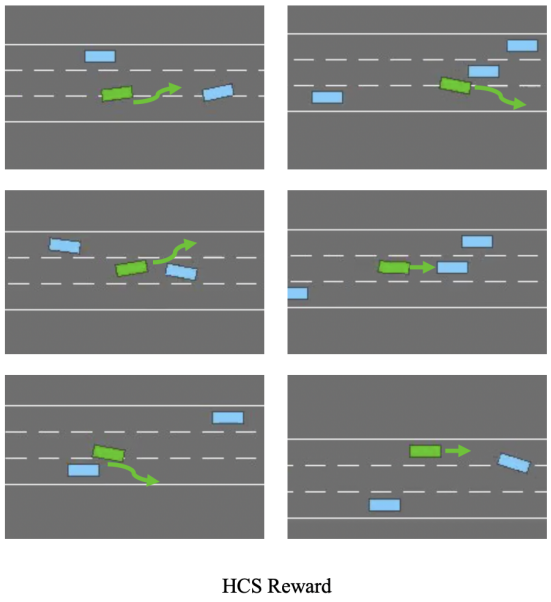


图 9：从 HCS奖励的 RSS违规分布峰值中采样得到的轨迹的关键场景。

，并使 AST 能够识别出失败，此时 RSS 判定场景为不安全的时间步占比更高。在使用 HCS奖励的两项实验中，状态处于不安全状态的最大占比也更高。

对失败轨迹进行人工标注，以将每个状态分类为安全或不安全。图 6 展示了这些分布的箱线图。在 HCS奖励的情况下，我们直接进行优化，使轨迹倾向于包含更多被人类判定为关键状态的状态，这一点也反映在结果中。使用 HCS奖励会生成更多样化的失败轨迹，并使表现出危险行为的状态百分比具有更大的分布范围。

C. 定性比较

我们对不同奖励公式生成的结果进行了定性比较。图 8 和图 9 分别展示了使用启发式奖励、QCS奖励和 HCS奖励从失败轨迹中找到的关键场景的所有实例。启发式奖励和HCS奖励的轨迹取样自图 7b 所示 RSS违规分布的峰值。QCS奖励的轨迹取样自图4中的分布峰值。启发式奖励的示例（图 8 左）包含两类危险行为：1）自车以较高速度行驶，未能与前方车辆保持安全距离。2）自车进行变道，导致其与前方车辆之间失去安全间隔。使用QCS奖励找到的关键场景（图 8 右）显示出类似结果：自车在三个不同实例中进行了不安全变道（其中一次是失去安全间隔，另外两次是强制切入，导致另一辆车被迫急刹车）。与前两种奖励相比，使用HCS奖励取样的失败轨迹（图 9）包含更多关键场景并展示了启发式奖励和QCS奖励中分别出现的不安全跟车距离和变道强制切入行为。从定性角度看，HCS奖励表现出更高程度的自车激进行为，其最小安全间隔更小，车速更高。

VI. 结论

本文提出，在使用自适应压力测试训练用于生成自主智能体故障场景的策略时，以关键状态替代奖励中的概率代理指标。我们介绍了在被测系统和环境具有表示高级指令的离散动作空间，且环境动作的底层分布难以直接获得的情况下，传统自适应压力测试奖励形式用于自适应故障搜索的局限性。我们考虑了关键状态的两种刻画方式。第一种从智能体的视角定义关键状态，并使用智能体策略生成的Q值方差对其进行刻画。我们证明，尽管这种形式的表现优于纯启发式奖励，但由于其区分危险自车行为的能力有限，因此并不适合故障搜索任务。这促使我们将关键状态刻画为一个分类问题，将环境观测映射为关键性。我们训练了一个贝叶斯神经网络来完成这一任务，并证明，在与自适应压力测试结合使用时，该方法能够有效生成失败轨迹，同时增加自车自动驾驶汽车表现出的危险行为。未来，我们希望研究用于表示系统状态的更通用特征编码，以便在关键状态分类时更好地纳入时空关系，例如使用环境的原始图像。

参考文献

[1] W. Huang、K. Wang、Y. Lv 和 F. Zhu，《自动驾驶汽车测试方法综述》，载于《IEEE智能交通系统会议》（ITSC），2016年。[2] N. Kalra 和 S. M. Paddock，《驶向安全：需要行驶多少英里才能证明自动驾驶汽车的可靠性？》兰德公司，2016年。[3] P. Koopman 和 M. Wagner，《自动驾驶汽车测试与验证中的挑战》，载于 SAE世界大会，2016年。[4] W. Ding、B. Chen、B. Li、K. J. Eun 和 D. Zhao，《用于决策算法评估的多模态安全关键场景生成》，2020年。[5] W. Ding、W. Wang 和 D. Zhao，《用于模拟车辆间交互的新型多车轨迹生成器》，载于 IEEE国际机器人与自动化会议（ICRA），2019年。[6] G. E. Mullins、P. G. Stankiewicz、R. C. Hawthorne 和 S. K. Gupta，《用于自动驾驶汽车测试与评估的挑战性场景自适应生成》，载于《系统与软件杂志》，第137卷，2018年。[7] M. O'Kelly、A. Sinha、H. Namkoong、J. C. Duchi 和 R. Tedrake，《通过稀有事件仿现实现可扩展的端到端自动驾驶汽车测试》，载于神经信息处理系统会议（NeurIPS），2018年。[8] C. E. Tuncali、G. Fainekos、H. Ito 和 J. Kapinski，《使用机器学习组件对自动驾驶汽车进行基于仿真的对抗性测试生成》，载于 IEEE智能车辆研讨会（IV），2018年。[9] L. Li、W. Huang、Y. Liu、N. Zheng 和 F. Wang，《自动驾驶汽车智能测试：一种新方法》，《IEEE智能车辆汇刊》，第1卷，第2期，第158–166页，2016年。[10] R. Lee、M. J. Kochenderfer、O. J. Mengshoel、G. P. Brat 和 M. P. Owen，《机载防撞系统的自适应压力测试》，载于 IEEE数字航空电子系统会议（DASC），2015年。[11] P. Mahalanobis，《统计学中的广义距离》，《印度国家科学院学报》，第2卷，第49–55页，1936年。[12] M. Koren、A. Corso 和 M. J. Kochenderfer，《自适应压力测试形式化》，载于安全自主系统研讨会（RSS），2019年。[13] A. Majumdar 和 M. Pavone，《机器人应如何评估风险？迈向机器人学中的风险公理化理论》，载于国际机器人研究研讨会（ISRR），2017年。[14] S. H. Huang、K. Bhatia、P. Abbeel 和 A. D. Dragan，《通过关键状态建立适当信任》，载于 IEEE智能机器人与系统国际会议（IROS），2018年。[15] M. Koren、S. Alsaif、R. Lee 和 M. J. Kochenderfer，《自动驾驶汽车自适应压力测试》，载于 IEEE智能车辆研讨会（IV），2018年。[16] C. B. Browne、E. Powley、D. Whitehouse、S. M. Lucas、P. I. Cowling、P. Rohlfshagen、S. Tavener、D. Perez、S. Samothrakis 和 S. Colton，《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》，第4卷，第1期，第1–43页，2012年。[17] R. Lee、O. Mengshoel、A. Saksena、R. Gardner、D. Genin、J. Brush 和 M. J. Kochenderfer，《防撞系统的差分自适应压力测试》，载于 AIAA建模与仿真会议，2018年。[18] A. Corso、P. Du、K. Driggs-Campbell 和 M. J. Kochenderfer，《通过奖励增强进行自动驾驶汽车验证的自适应压力测试》，载于 IEEE智能交通系统会议（ITSC），2019年。[19] B. R. Kiran、I. Sobh、V. Talpaert、P. Mannion、A. A. A. Sallab、S. Yo- gamani 和 P. P´erez，《自动驾驶的深度强化学习：综述》，2020年。[20] S. Levine、C. Finn、T. Darrell 和 P. Abbeel，《深度视觉运动策略的端到端训练》，载于《机器学习研究杂志》（JMLR），2016年。[21] E. Leurent，《用于自动驾驶决策的环境》，https://github.com/eleurent/highway-env，2018年。[22] A. Hill、A. Raffin、M. Ernestus、A. Gleave、A. Kanervisto、R. Traore、P. Dhariwal、C. Hesse、O. Klimov、A. Nichol、M. Plappert、A. Radford、J. Schulman、S. Sidor 和 Y. Wu，《Stable Baselines》，https://github. com/hill-a/stable-baselines，2018年。[23] S. Shalev-Shwartz、S. Shammah 和 A. Shashua，《安全且可扩展的自动驾驶汽车形式化模型》，arXiv，第1708.06374号，2017年。