

自动驾驶中的深度学习和强化学习技术综述

山东大学 数学与统计学院 数据科学与人工智能实验班

刘润泽 王晨晔 王正阳 虞若凡 韩泽宇

摘要

在过去十年中，自动驾驶车辆技术 (AV) 取得了越来越快的进步，这主要得益于智能交通系统 (ITS) 和人工智能 (AI) 领域的进步，由于可以显著地减少道路事故和人身伤害，自动驾驶汽车成为了一个非常热门的研究课题。本文的目的为调查用于自动驾驶的深度学习和强化学习技术的最新进展。我们首先介绍了基于人工智能的 CNN、LSTM 和 RL 范式，这些方法构成了场景感知、运动规划和行为控制算法的基础。我们从自动驾驶级别、硬件、安全性、训练数据集、算力平台与域控制器设计五方面介绍近年来自动驾驶领域的基本概念和发展，通过宽领域全方面来讲述该领域的成果发展。我们重点调查了近年来 DL 与 RL 在研究场景感知、运动规划和行为控制领域的算法发展和最新成就。最后，本调查从前文介绍的各个角度分析强调了自动驾驶领域的挑战，并指出了未来可能的研究方向。本调查有助于深入了解自动驾驶的深度学习和强化学习方法的优势和局限性，有助于从宽领域全方面了解自动驾驶的各个组成部分，并根据分析帮助读者进行选择和设计。

关键词： 自动驾驶 深度学习 强化学习 场景认识 运动规划 车辆控制

一、引言

[1] 一文中指出,自动驾驶汽车与道路旁基础设施和响应单元相互合作收集和共享信息,从而感知其环境,并在没有人类参与的情况下做出决策。AVs 可以在发生交通堵塞或事故时依靠车辆通信,从而交换安全信息、交通状况和警告信息。无人驾驶汽车主要依靠传感器、执行器、复杂算法、人工智能技术和强大的计算资源来运行程序。因此,AVs 理论上能够处理复杂的道路情况,从而显著提高用户的安全性、舒适性和便利性。

在过去十年中,深度学习和人工智能 *Deep Learning and Artificial Intelligence(AI)* 广泛应用于计算机视觉 *Computer Vision* [2]、机器人 *Robotics* [3] 和自然语言处理 *Natural Language Processing(NLP)* [4] 领域,并取得了许多突破,同时它们对当今学术界和工业界的自动驾驶革命也产生了重大影响。我们使用 Google Trends 研究了从 2004 年到 2022 年间来自世界各地的研究者对深度学习技术的关注程度,如图 1 2所示,纵坐标数字表示给定区域和时间相对于图表上最高点的搜索兴趣。值为 100 说明该术语达到了受欢迎程度峰值。值为 50 表示该术语的受欢迎程度只有原来的一半。0 分表示此期限的数据不足。强化学习作为机器学习技术,从 2004 年到 2013 年得到了超过其他三项的关注程度,但总的来说技术发展不温不火,研究者较少。在 2013 年,DeepMind 发表了利用强化学习玩 Atari 游戏的论文,至此强化学习开始了新的十年,得到了更为广泛的关注。而 2006 年,神经网络之父 Geoffrey Hinton 的团队在《科学》上发表的文章,提出了深层网络训练中梯度消失问题的解决方案,至此开启了深度学习在学术界和工业界的浪潮。自动驾驶技术作为目前最热门的方向之一,其关注程度与研究者对深度学习和强化学习的研究数量同步增长。

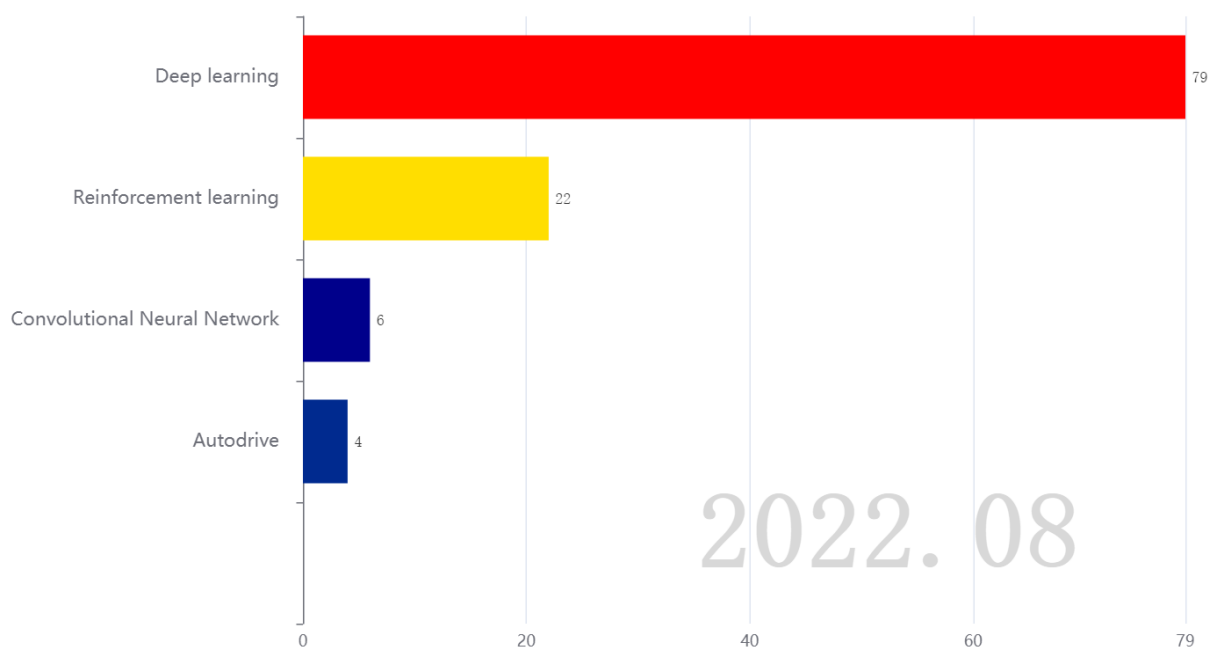


图 1 基于 Google Trends 的研究兴趣变化图 (2004-2022 年)

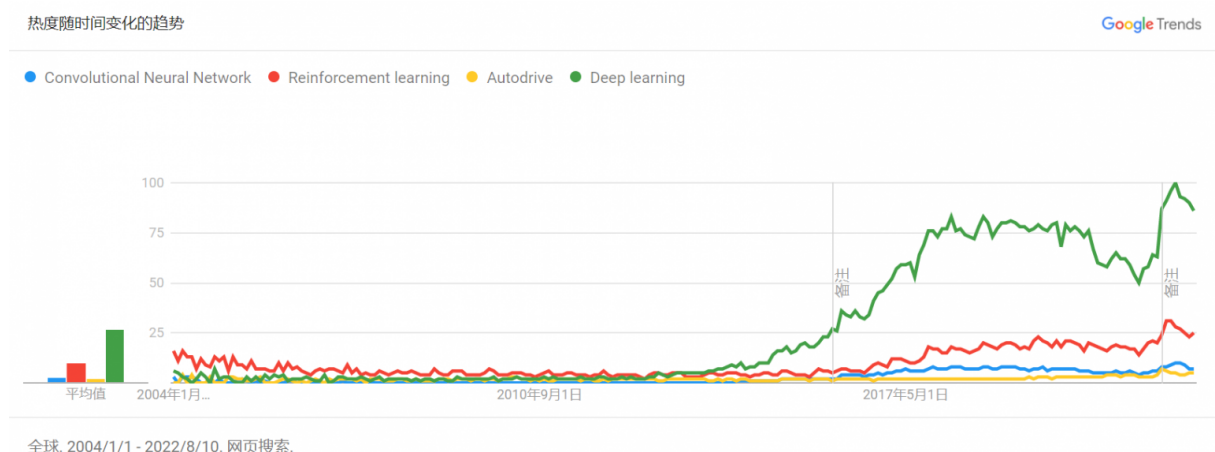


图2 基于 Google Trends 的研究兴趣折线图 (2004-2022 年)

近年来, 自动驾驶汽车开始从实验室开发与测试转向在公共道路上行驶。理论上讲, 完善的自动驾驶技术的部署可以减少道路事故和交通拥堵, 并改善我们在拥挤城市中的机动性。根据 *SAE J3016* 标准 [5] 引入自动驾驶从 1 到 5 共五个等级, 用于对车辆自动化程度进行分级。较低的 SAE 级别具有基本的驾驶员辅助功能, 而较高的 SAE 级别则更倾向无需任何人机交互, 级别 5 的汽车不需要人工输入, 通常甚至不会配备方向盘或脚踏板。

智能交通系统 (ITS)、计算系统和人工智能 (AI) 的最新进展为 AVs 的广泛引入铺平了道路。这为智能道路、智能交通安全和旅行者舒适度带来了新的机遇。研究人员估计到 2025 年将有 800 万辆自动驾驶汽车上路 [6]。然而, [7] 中指出由于过去发生的一些事件, AVs 的普及受到阻碍。虽然大多数驾驶场景可以用一些经典的感知、路径规划和运动控制方法相对简单地解决, 但剩余的未解决场景往往成为传统方法失败的案例, 这正是 [8] 中指出的黑天鹅 (Black Swans) 情形, 所以在自动驾驶领域迫切需要新的技术的革新。

20 世纪 80 年代, Ernst Dickmanns [9] 开发了首批自动驾驶汽车, 这为未来新的研究项目铺平了道路。1994 年, VaMP 无人驾驶汽车成功行驶了 1600 公里, 其中 95% 是通过自动驾驶技术实现的。1995 年, CMU NAVLAB 基于自动驾驶技术行驶了 6000 公里, 实现了 98% 的自动驾驶。自动驾驶的其他重要里程碑是 2004 年和 2005 年的 DARPA Grand 挑战, 以及 2007 年的 DARPA Urban 挑战, 这两项挑战的目标是让无人驾驶汽车在没有人为干预的情况下尽可能快地在越野道路上行驶。2004 年, 15 辆车没有一辆可以完成比赛。而到了 2005 年比赛中 Stanley 利用机器学习技术在非结构化环境中完成了比赛并获得了冠军。这是自动驾驶汽车发展的一个转折点, 其意义在于人们开始认同机器学习和人工智能是自动驾驶的核心组成部分。因此自动驾驶领域中的深度学习和强化学习算法大多在 2005 年之后开始进行的。

为了使 5 级车辆成为可能，就要要求车辆具有像人类驾驶员一样的“思考”、“感知”和“反应”能力。随着人工智能技术在不同领域有了越来越多的成就，特别是在图像分类、目标检测和语音识别方面，使得人工智能技术 (如深度学习 (DL) 和强化学习 (RL)) 越来越多地用于实现 5 级车辆。基于深度学习的方法使许多研究能够解决 AVs 中不同的挑战性问题，例如准确识别和定位道路上的障碍物、适当的车辆控制和运动规划。

本文将总结近年来用于处理 AVS 中主要功能的基于深度学习和强化学习的技术，即场景理解、运动规划、决策、车辆控制、社会行为和通信。我们关注范围限于基于 DL 和 RL 的方法，忽略了基于浅层机器学习 (ML) 的技术，传统的 ML 方法是一个过去已经广泛研究的主题，在实现 AVs 的主要功能 (例如场景理解和运动规划) 方面表现出了局限性 [10,11]。因此，我们将目光关注于更有价值的 DL 和 RL 方法，以挖掘其在实现这些功能方面的潜力与优势。

本文还将总结近年来自动驾驶领域的深度学习和强化学习算法概述的相关文章 [12–15]，并对该领域做出自己的总结和理解。

在本篇文献回顾中，我们回顾了自动驾驶中常使用的深度学习和强化学习算法，详细介绍自动驾驶领域的级别划分、硬件技术、安全性研究、常用的训练数据集以及算力平台与域控制器，并对应用于自动驾驶汽车的最先进的深度学习和强化学习方法进行了调查。最后对自动驾驶领域的当前的挑战进行了分析，对未来的发展进行了展望。下面的图 3 为该概述的整体架构图，具体可视化交互见网页。

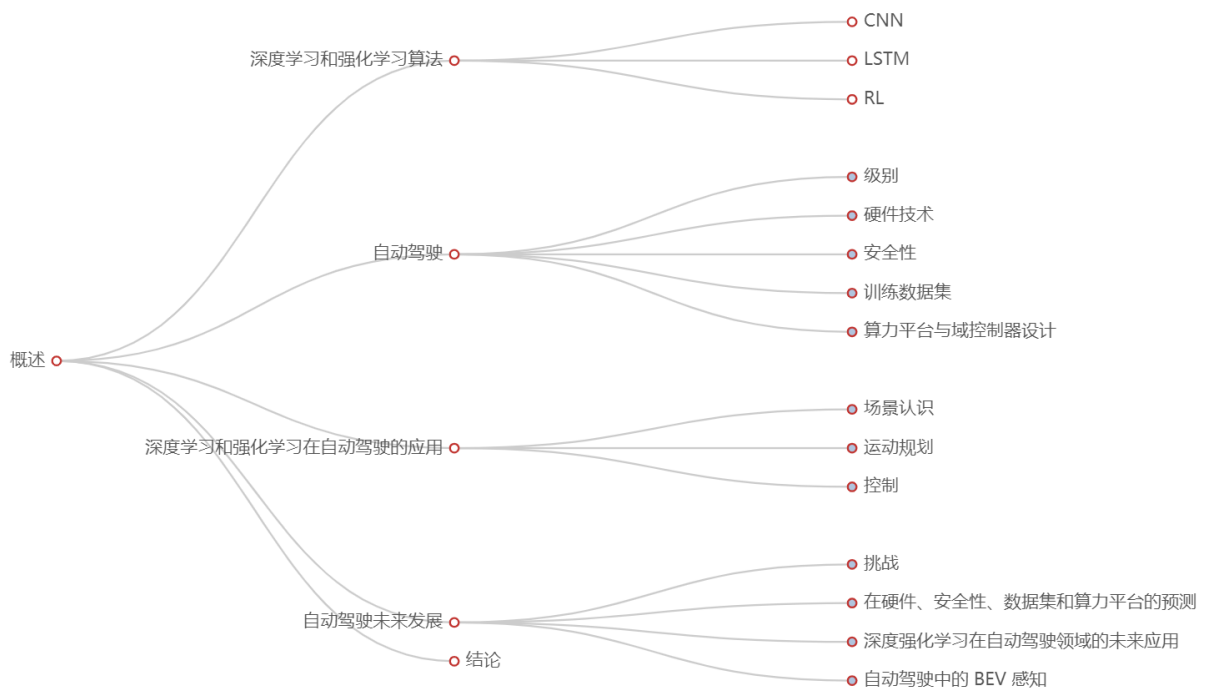


图 3 概述的整体文章架构

二、深度学习和强化学习：概述

在本节中，我们简要描述 AVs 中使用的深度学习技术的基本原理，并展示每种范式的能力。我们在 2.1 与 2.2 小节总结了两种应用于自动驾驶的最常见的深度学习方法：Convolutional Neural Networks(CNN) 和 Long Short-Term Memory(LSTM)，在 2.3 小节总结了强化学习方法。

在整个调查过程中，我们使用以下符号来描述与时间相关的序列。变量的值定义为单个离散时间步长 t ，写作上标 $\langle t \rangle$ ，或定义为时间间隔 $\langle t, t+k \rangle$ ，其中 k 表示序列的长度。例如，状态变量 z 的值在离散时间 t 处定义为 $\mathbf{z}^{\langle t \rangle}$ ，或在序列间隔内定义为 $\mathbf{z}^{\langle t, t+k \rangle}$ 。向量和矩阵用粗体符号表示。

2.1 Deep Convolutional Neural Networks (CNN)

2.1.1 简介

DL 模型已成功应用于各种自动驾驶应用领域，在过去几年中，它们已成为计算机视觉应用的关键组成部分。神经网络可以解决的关键问题之一是检测和定位图像与视频中的目标对象，使 AV 能够识别其环境。在目标检测中常使用的 DL 方法中有 CNN。它们为 AVs 提供可操作的信息，即检测和分类对象 (例如车道、交通灯、行人、交叉线和交通标志) [16]。CNN 在图像分类、目标检测和语义分割方面表现很好，其包含了三个基本组件，即卷积层、池化层和完全连接层。卷积层由滤波器组成，用于提取图像或视频中的主要视觉特征，通过卷积和池化操作将图像的特征转换为小阵列，这些阵列馈送到全连接层中输出预测的图像类别 [17]。

卷积神经网络 (CNN) 主要用于处理图像等空间信息，可以被用作图像特征提取器和通用的非线性函数近似器 [18,19]。在深度学习兴起之前，计算机视觉系统通常基于人工标记的特征来进行训练，例如 HAAR [20]、Local Bi-nary Patterns(LBP) [21] 或 Histograms of Oriented Gradients(HoG) [22]。与这些传统的手工特征相比，卷积神经网络能够在训练集编码的特征空间的表示下自主学习。

本文考察了最常用的十种 CNN 算法，表 1 列出了这十种算法的信息，图 4 为这十种算法架构的引用次数 (具体可视化交互见网页)。

10 种常用的 CNN 架构

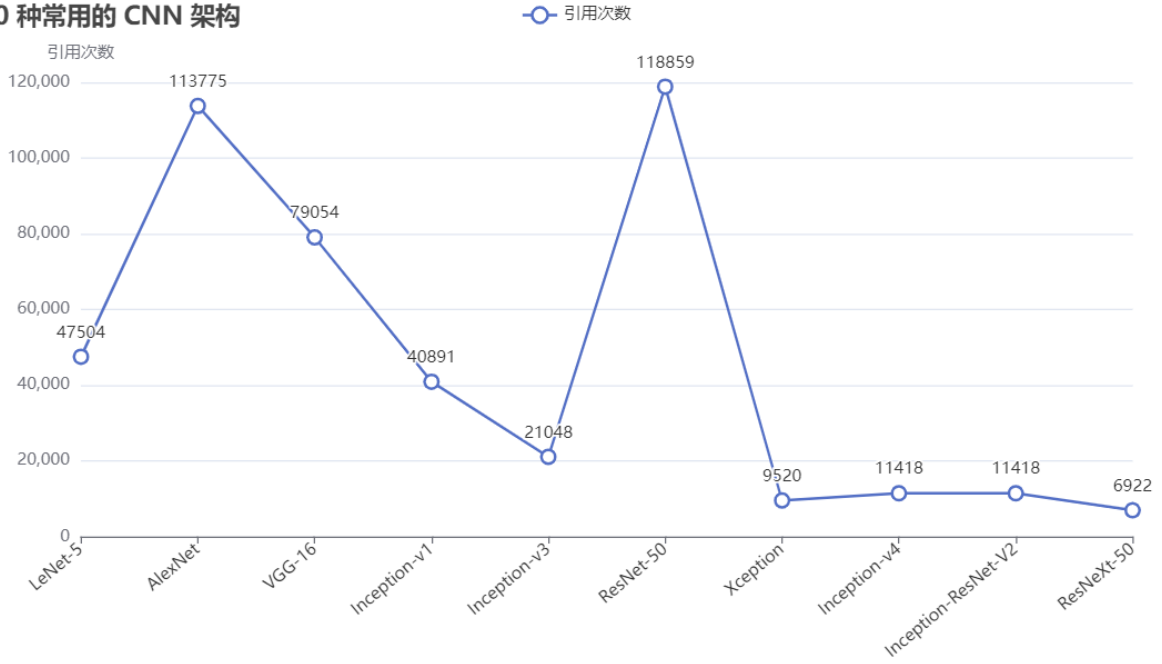


图 4 10 种常用的 CNN 架构引用次数

2.1.2 原理

CNN 首先参数化权重向量 $\theta = [\mathbf{W}, \mathbf{b}]$ ，其中 \mathbf{W} 是控制神经元连接的权重集， \mathbf{b} 是神经元偏差值集。根据权重集 \mathbf{W} 创建图像滤波器，并在训练过程中调整系数。CNN 内的卷积层利用图像像素的局部空间相关性来学习平移不变卷积滤波器，该卷积滤波器用于捕获判别图像特征。

考虑 k 层中的多通道信号表示 \mathbf{M}_k ，信号的通道集成表示为 $\mathbf{M}_{k,c}$ ，其中 $c \in N$ 、 $k+1$ 层生成信号表示为：

$$\mathbf{M}_{k+1,l} = \varphi(\mathbf{M}_k * \mathbf{w}_{k,l} + \mathbf{b}_{k,l}), \quad (2.1)$$

其中 $\mathbf{w}_{k,l} \in \mathbf{W}$ 是一个卷积滤波器，其通道数为 \mathbf{M}_k ， $\mathbf{b}_{k,l} \in \mathbf{b}$ 表示方差， l 是通道索引， $*$ 表示卷积运算， $\varphi(\cdot)$ 是应用于输入信号中每个像素的激活函数。通常，Rectified Linear Unit(ReLU) 是计算机视觉应用中最常用的激活函数。CNN 的最后一层通常是全连接层，它在对象的高级抽象表示上充当对象鉴别器。

在有监督方式下，可以使用训练数据库 $\mathcal{D} = [(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)]$ 训练 CNN 的响应函数 $R(\cdot; \theta)$ ，其中， x_i 是数据样本， y_i 是相应的标签， m 是训练示例数。可以使用最大似然估计 (MLE) 计算最优网络参数。为了解释清楚，我们以简单最小二乘误差函数为例，该函数可用于在训练回归估计器时驱动 MLE 过程：

表 1 10 种常用的 CNN 架构

架构名称	论文名称	发表时间	参数大小	架构特点
LeNet-5	Gradient-Based Learning Applied to Document Recognition [18]	1998	60,000	用激活函数和池化层堆叠卷积，并以一个或多个全连接层结束网络。
AlexNet	ImageNet Classification with Deep Convolutional Neural Networks [2]	2012	60M	第一个采用 ReLU 做为激活函数
VGG-16	Very Deep Convolutional Networks for Large-Scale Image Recognition [23]	2014	138M	通过堆叠均匀卷积设计了更深的网络 (大约是 AlexNet 的两倍)。
Inception-v1	Going Deeper with Convolutions [24]	2015	5M	使用密集模块或块构建网络。 (不是堆叠卷积层，而是堆叠模块或块)。
Inception-v3	GRethinking the Inception Architecture for Computer Vision [25]	2016	24M	最早使用批量归一化的算法之一。
ResNet-50	Deep Residual Learning for Image Recognition [26]	2016	26M	(1) 普及了跳过连接 (又名快捷连接，残差) (2) 在不影响模型泛化能力的情况下设计更深的 CNN(最多 152 层) (3) 最早使用批量标准化的算法之一。
Xception	Xception: Deep Learning with Depthwise Separable Convolutions [27]	2017	23M	引入了完全基于深度可分离卷积层的 CNN。
Inception-v4	Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [28]	2017	43M	对 Inception-v3 三点的改进： (1) 对 Stem 组和 Inception-C 模块做出细微变化； (2) 为每个 Inception 块做出了统一的选择”； (3) 提到” 剩余连接可以显著提高训练速度”。
Inception-ResNet-V2	Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [28]	2017	56M	与 Inception-v3 相比有三点改进： (1) 添加更多 Inception 模块。 (2) 将 Inception 模块转换为 Residual Inception 块。 (3) 在 Stem 模块后添加了新型的 Inception-A 模块。
ResNeXt-50	Aggregated Residual Transformations for Deep Neural Networks [29]	2017	25M	在每个模块中添加了平行的塔/分支/路径。

$$\hat{\theta} = \arg \max_{\theta} \mathcal{L}(\theta; \mathcal{D}) = \arg \min_{\theta} \sum_{i=1}^m (R(\mathbf{x}_i; \theta) - y_i)^2 \quad (2.2)$$

出于分类目的,最小二乘误差通常由交叉熵或负对数似然损失函数代替。公式 (2.2) 中的优化问题通常通过随机梯度下降 (SGD) 和梯度估计的反向传播算法来解决。在实践中,可以使用了不同的 SGD 变体,例如 Adam 或 AdaGrad。

2.2 Long Short-Term Memory (LSTM)

2.2.1 简介

另一个适合处理数据序列的 DL 模型是 LSTM；该网络是一种递归神经网络。LSTM 网络使用反馈连接进行序列和模式识别，并使用输入、输出和遗忘门。因此，它记住了从上一步时间步长计算的输出，并根据输入得到输出 [30]。该架构被证明能够基于 AVs 的过去动作预测当前动作 [31]。LSTM 网络已应用于不同的自动驾驶任务，如运动规划、决策和车辆控制。

长短时记忆 (LSTM) 网络 [32] 是非线性函数近似器，用于估计序列数据中的时间相关性。与传统的递归神经网络不同，LSTM 通过合并三个门来解决消失梯度问题，这三个门分别控制输入、输出和存储状态。

本文考察了常用的八种 LSTM 算法，表 2 列出了这八种算法的信息，图 5 列出了相关架构的引用信息，具体可视化交互见网页。可以看出不论是 CNN 架构还是 LSTM 架构，大多都是 2016 年前的引用数量较多，2016 之后的论文引用次数有限，但这并不代表算法不如过去，恰恰相反最近几年的算法在参数量、计算成本、计算速度、精准度等方面都有所提升，而引用量略低可以反映出虽然当前全新的算法性能有所提升，但仍未被普遍考察和使用，后续在人工智能领域的深度学习和强化学习算法仍有很大的发展潜力，同时也反映了当前深度学习和人工智能领域尤其需要新鲜血液来为行业助力，相信在自动驾驶领域中深度学习算法能依靠其优势不断向 L4 和 L5 级别迈步。

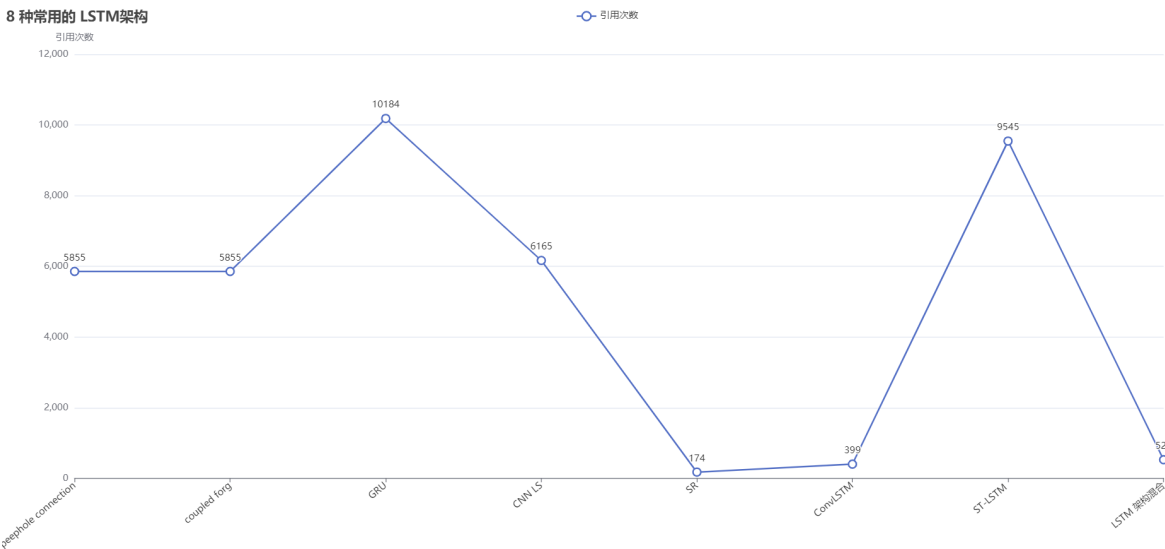


图 5 8 种常用的 LSTM 相关架构引用次数

表 2 8 种常用的 LSTM 架构

架构名称	论文名称	发表时间	架构特点
peephole connection	Learning to forget: continual prediction with LSTM [33]	2000	增加 “peephole connection”(窥探孔结构) 门层可以接受细胞状态的输入
coupled forgetting	Learning to forget: continual prediction with LSTM [33]	2000	使用 coupled 忘记机制 统一决定需要忘记和添加的信息。
GRU	Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling [30]	2014	应用最广泛的 LSTM 变体之一; 门控数量减少到两个: 复位门和更新门。
CNN LSTMs	Long-Term Recurrent Convolutional Networks for Visual Recognition and Description [34]	2015	使用 CNN 对输入数据做特征提取, 并结合 LSTM 来 支持序列预测。常用于活动识别、图像描述和视频描述。
SRU	Training RNNs as Fast as CNNs [35]	2017	实现多个时间步并行计算, 相比原始的 LSTM 训练速度快若干倍。
ConvLSTM	PredRNN: Recurrent Neural Networks for Predictive Learning Using Spatiotemporal LSTMs [36]	2017	将 LSTM 中的全连接权重改为卷积; 用 LSTM 单元中每个门的卷积运算代替矩阵乘法; 解决了传统 LSTM 单元不适用于 视频, 卫星, 雷达图像等空间序列数据集的问题。
ST-LSTM	Xception: Deep Learning with Depthwise Separable Convolutions [36]	2017	相较于 ConvLSTM 该网络实现了时空记忆流传递
LSTM 架构混合	Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features [37]	2017	使用 CNN 和 DB-LSTM 处理视频数据, 能够学习长期序列, 并且可以通过分析特定时间间隔的特征来处理冗长的视频。

2.2.2 原理

递归层利用序列数据的时间相关性来学习与时间相关的神经结构。考虑 LSTM 网络中的记忆状态 $\mathbf{c}^{<t-1>}$ 和输出状态 $\mathbf{h}^{<t-1>}$, 在时间 t 的输入数据 $\mathbf{s}^{<t>}$ 。门的开启或关闭由 sigmoid 函数控制, 具体公式如下:

$$\Gamma_u^{<t>} = \sigma(\mathbf{W}_u \mathbf{s}^{<t>} + \mathbf{U}_u \mathbf{h}^{<t-1>} + \mathbf{b}_u) \quad (2.3)$$

$$\Gamma_f^{<t>} = \sigma(\mathbf{W}_f \mathbf{s}^{<t>} + \mathbf{U}_f \mathbf{h}^{<t-1>} + \mathbf{b}_f) \quad (2.4)$$

$$\Gamma_o^{<t>} = \sigma(\mathbf{W}_o \mathbf{s}^{<t>} + \mathbf{U}_o \mathbf{h}^{<t-1>} + \mathbf{b}_o) \quad (2.5)$$

其中 $\Gamma_u^{<t>}$ 、 $\Gamma_f^{<t>}$ 、 $\Gamma_o^{<t>}$ 分别是输入门、遗忘门和输出门的门函数。给定当前观察结果, 记忆状态 $\mathbf{c}^{<t>}$ 将更新为:

$$\mathbf{c}^{<t>} = \Gamma_u^{<t>} * \tanh(\mathbf{W}_c \mathbf{s}^{<t>} + \mathbf{U}_c \mathbf{h}^{<t-1>} + \mathbf{b}_c) + \Gamma_f^{<t>} * \mathbf{c}^{<t-1>}, \quad (2.6)$$

新的网络输出 $\mathbf{h}^{<t>}$ 计算公式如下:

$$\mathbf{h}^{<t>} = \Gamma_o^{<t>} * \tanh(\mathbf{c}^{<t>}) \quad (2.7)$$

LSTM 网络 Q 由 $\theta = [\mathbf{W}_i, \mathbf{U}_i, \mathbf{b}_i]$ 参数化, 其中 \mathbf{W}_i 表示门和存储单元的权重乘以输入状态, \mathbf{U}_i 是控制激活的权重, \mathbf{b}_i 表示神经元偏差值集, $*$ 为按照元素乘法。

在监督学习中, 给定一组训练序列 $\mathcal{D} = \left[\left(\mathbf{s}_1^{<t-\tau_i, t>}, \mathbf{z}_1^{<t+1, t+\tau_o>} \right), \dots, \left(\mathbf{s}_q^{<t-\tau_i, t>}, \mathbf{z}_q^{<t+1, t+\tau_o>} \right) \right]$, 也就是说, 对于赋值为 $\mathbf{z}^{<t, t+\tau_o>}$ 的 q 独立观测序列对, 可以使用最大似然估计来训练 LSTM 网络 $Q(\cdot; \theta)$ 的响应:

$$\begin{aligned} \hat{\theta} &= \arg \max_{\theta} \mathcal{L}(\theta; \mathcal{D}) \\ &= \arg \min_{\theta} \sum_{i=1}^m l_i \left(Q \left(\mathbf{s}_i^{<t-\tau_i, t>} ; \theta \right), \mathbf{z}_i^{<t+1, t+\tau_o>} \right), \\ &= \arg \min_{\theta} \sum_{i=1}^m \sum_{t=1}^{\tau_o} l_i^{<t>} \left(Q^{<t>} \left(\mathbf{s}_i^{<t-\tau_i, t>} ; \theta \right), \mathbf{z}_i^{<t>} \right), \end{aligned} \quad (2.8)$$

其中, 观察值的输入序列 $\mathbf{s}^{<t-\tau_i, t>} = [\mathbf{s}^{<t-\tau_i>}, \dots, \mathbf{s}^{<t-1>}, \mathbf{s}^t]$ 由 τ_i 个连续数据样本组成, $l(\cdot, \cdot)$ 是 logistic 回归损失函数, t 表示时间指数。

2.3 Reinforcement Learning

RL 是 ML 的一个子领域, 其解决随着时间推移的自主学习和最优决策问题。DL 方法侧重于开发使用数据进行自主学习的计算机程序, 而 RL 方法允许人工智能从错误和经验中学习。RL 算法中人工智能通过在环境中行动获得奖励, 其目标是选择随着时间推移使得累积预期回报最大化的行动。因为训练数据集无法涵盖现实生活的各个方面, 所以在训练过程中迫切需要在环境中学习, 这正是 RL 的优势所在, 所以 RL 目前在自动驾驶领域越来越受到关注。

RL 代理可以建模为马尔可夫决策过程 (Markov Decision Process, MDP), 如下所示: 代理通过执行动作与环境交互、接受观察和奖励。如图 6 所示, 在每个时间步骤 t , 代理接收环境状态 $s_t \in S$ 。基于此状态, 代理选择一个动作 $a_t \in A$ 。选择任何操作都基于代理的行为, 这个过程也称为策略。策略告诉代理应该为每个可能的状态选择哪些操作。作为每个动作的结果, 代理收到奖励 $r_t \in R$ 并观察下一个状态 $s_{t+1} \in S$ [38]。我们可以将获得奖励的过程表示为任意函数 f 。在每个时间 t , 我们有: $f(s_t, a_t) = r_t$ 。

RL 的主要目标是找到使每个状态的策略最大化的操作。Q-learning [39] 通过学习每个状态-动作对的最优 Q 值来选择最优策略。Q-Learning 算法使用 Bellman 方程迭代更新每个状态-动作对的 Q 值, 直到 Q 函数收敛到最优 Q 函数 q_* :

$$q_*(s, a) = E \left[r_{t+1} + \gamma \max_{a'} q_*(s', a') \right] \quad (2.9)$$

在 Q-Learning 中, 动作的选择基于 ϵ 贪婪策略, 其中 ϵ 是选择随机动作的概率, $1 - \epsilon$

是选择具有高 Q 值的动作的概率。该 Q 值在每个选定动作中使用给定表达式迭代更新：

$$q(s, a) \leftarrow q(s, a) + \alpha \left[r + \gamma \max_{a'} q(s', a') - q(s, a) \right] \quad (2.10)$$

其中 γ 反映当前奖励相对于未来奖励重要性的贴现率， α 是学习速率， s' 是下一个状态。可以看出，Q-Learning 算法的关键是维护一个 Q 表，该表存储状态-动作对的 Q 值。

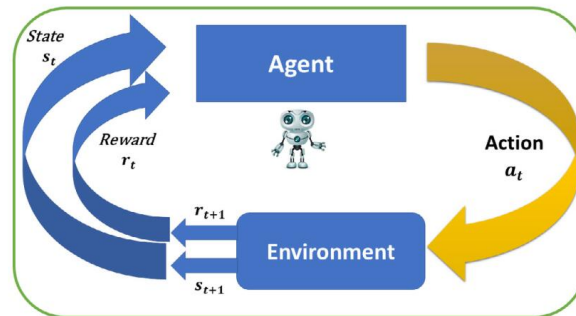


图 6 RL 的结构图

三、自动驾驶

3.1 自动驾驶级别

3.1.1 自动驾驶基本概念

- **DDT(Dynamic Driving Task): 动态驾驶任务**

包括车辆驾驶过程中所有的实时操作类和战术类功能，不包括战略类功能，包括但不限于：车辆的横纵向控制、目标和事件的探测和响应、行动规划和交互提醒。简单来说，从你坐到驾驶位置并启动车辆，所有的观察、思考和动作都属于 DDT，比如看到前方有个反光锥，你决定变道绕开，并打了一把方向盘，这就是一个 DDT 过程。

- **OEDR(Object and Event Detection and Response): 目标和事件的监测和响应**

OEDR 是 DDT 的一项子任务，包括了对驾驶环境、事件的监测和响应，比如看到前方路口红灯亮了 (驾驶环境监测)，决定减速并踩下了刹车 (响应)。

- **DDT Fallback: DDT 接管**

俗称备胎，即在某些情况下驾驶员 (L4/L5 级中没有这个概念) 重新操控车辆。

- **ODD(Operational Design Domain): 设计运行域**

自动驾驶的工作条件，包括但不限于天气、道路、速度等限制条件。

3.1.2 自动驾驶不同等级

- **L1 级-无自动驾驶**

没有自动化技术，车辆驾驶完全靠驾驶员手工操作，无任何自动驾驶功能。

- **L2 级-半自动驾驶**

目前市面上大多数新车所搭载的自动驾驶系统，甚至很多十万级别的自主品牌上都能够达到 L2 级别自动驾驶。L2 级自动驾驶具备许多自动驾驶车辆的雏形功能，例如全速段自动辅助驾驶、拥堵时自动辅助驾驶、自动危险预判刹车等功能，但更多时候，驾驶员仍需主导车辆行驶，只有在特定情况下，驾驶员双手才可以暂时的离开方向盘。

- **L3 级-有条件自动驾驶**

在 L3 级别下，车辆可以实现绝大部分路况的自动驾驶，接管汽车一大部分驾驶功能。但是在 L3 阶段，驾驶员仍需时刻保持注意力，以便在出现紧急情况时及时接管车辆。到了这一级别，驾驶员的重要性明显降低了，只需要像考官盯住学生不要作弊一样，随时关注路况信息，防止汽车行车系统因遇到无法识别的路况而不知如何处理即可。

- **L4 级-高度自动驾驶**

到了 L4 级自动驾驶，车内的自动化系统已经非常完善，车辆已经可以接替驾驶员的工作达到自动驾驶的水平，但如果驾驶员想亲自开车，仍可以接管车辆。

- **L5 级-完全自动驾驶**

L5 级别的自动驾驶，也是各个汽车工业设计师心中的理想状态，此级别的车辆，在任何天气、任何地域均可以实现完全自动驾驶，甚至该级别的汽车将由“座驾”转为“座舱”，让你可以完全忽略驾驶情况，专心的在车内休息即可。

3.2 硬件技术

3.2.1 定义

摄像头技术成熟且成本低，成为率先装车且用量最大的感知硬件。车载摄像头是 ADAS 系统的主要视觉传感器，也是最为成熟的车载传感器之一。然而由于摄像头与人眼一样，属于被动地接收可见光，因此在逆光或者光影复杂的情况下视觉效果较差，且易受恶劣天气影响。

毫米波雷达受天气环境的影响最小，全天候性能最佳。毫米波雷达与激光雷达工作原理相似，目前车载领域常用的毫米波雷达频段为 24GHz、77GHz 和 79GHz，分别对应短、长、中距离雷达。毫米波雷达由于波长够长，绕物能力好，受天气环境的影响最小，但同时由于波长过长，探测精度大大下降。

激光雷达精度最佳，满足 L3-L5 自动驾驶需求。激光雷达以激光作为载波，波长比毫米波更短，因此探测精度高、距离远。激光雷达还能通过回收不同方向激光尺的信息，以点成线，以线成面，形成障碍物 3D “点云” 图像。受限于技术难度大、成本高，目前还未实现大规模装车，随着未来产业链的日趋成熟，成本下调后，激光雷达产业或将迎来爆发。

3.2.2 性能对比

因为需要平衡从可靠性到成本的一系列因素，以便能够确定最佳点并选择最佳传感器组合，所以自动驾驶车辆选择合适的传感器组是一项微妙的任务。因此也可以从以下 10 个维度，分析相机、激光雷达和毫米波雷达的优缺点差异：

表 3 为常用的三个硬件的信息比较，图 7 为硬件信息的可视化雷达图，雷达图的具体可视化交互请见网页。

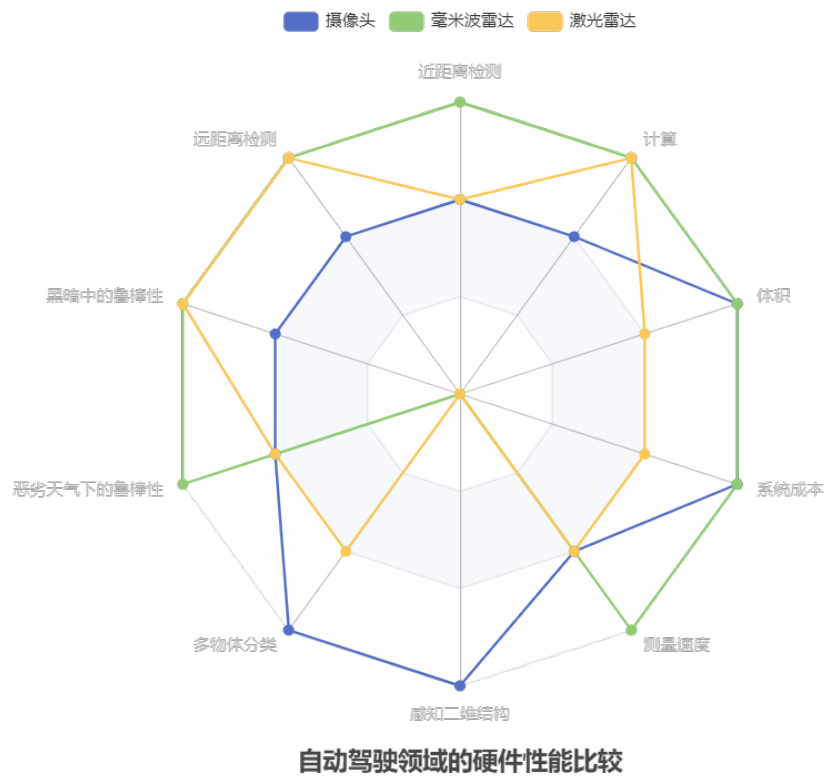


图 7 自动驾驶领域的硬件性能比较雷达图

• Range 范围

激光雷达和毫米波雷达系统可以探测几米到 200 米以上的物体。但是许多激光雷达系统很难探测非常近的物体，而毫米波雷达甚至可以探测到不到一米的物体，所以在近距离的探测上存在差异。雷达的探测距离往往取决于雷达类型 (长距离、中距离或短距离)。而单摄像头无法可靠地测量物体的距离，只能通过对现实情况的性质

表 3 三种常见自动驾驶传感器

分类	最远距离	工作原理	优势	劣势
摄像头	≥50m	通过摄像头采集外部信息并根据算法进行图像识别	唯一可读取“内容”的信息传感器	受光线干扰；算法要求高；丢失深度信息
毫米波雷达	≥250m	反射及接收毫米波，分析折返时间测算距离	具备绕物能力；受天气影响和环境最小；全天候全天时测距远	精确度下降，不能清晰呈现目标点云；数据稳定性差；对金属敏感
激光雷达	≥200m	反射及接收激光，分析折返时间测算距离	探测距离远，精度高；响应速度灵敏；形成三维还原	在不良天气下表现不佳；价格昂贵

(如平面路面)做出一些合理假设从而得到对应假设下的距离结果，这为判断障碍物的距离造成了些许困难。另一方面，立体摄像头最远只能测量到约 80 米的距离，大于这个距离，精度会显著下降，所以在远距离探测上摄像头的表现性能弱于激光雷达和毫米波雷达。

总结来看，摄像头需要特定的假设约束且在远距离的表现上欠佳；而激光雷达和毫米波雷达在远距离的表现较好，但在近距离上毫米波雷达的表现更加。

• Spatial resolution 空间分辨率

由于红外激光波长较短，激光雷达扫描的空间分辨率约为 0.1° ，所以这允许我们进行高分辨率 3D 扫描，从而对场景中的对象进行表征。另一方面，当随着距离的增加，毫米波雷达系统分辨小特征的性能明显下降。相机系统的空间分辨率由光学元件、图像上的像素大小及其信噪比决定。当小物体发出的光线扩散到图像传感器上的几个像素(模糊)时，小物体的细节就会丢失。此外，当几乎不存在环境光来照亮对象时，随着图像传感器的噪声增加，对象细节会被叠加，从而摄像头的空间分辨率会降低。

总结来看，摄像头受光线等因素产生的噪音的影响较大，受远距离的限制大；毫米波雷达随着距离增大精度会下降，受远距离的限制较大；激光雷达空间分辨率高，受远距离的影响较小。

- **Robustness in darkness 黑暗中的鲁棒性**

雷达和激光雷达在黑暗中都具有极好的鲁棒性，这是由于它们使用的都是主动传感器。虽然激光雷达系统的白天性能非常好，但它们在夜间的性能甚至更好，这是由于夜晚没有可能干扰红外激光反射探测的环境阳光。而与之不同的是，摄像头在夜间的探测能力非常低，这是由于它使用的是依赖环境光的被动传感器。尽管近几年随着图像传感器的提升，图像传感器的夜间性能有所提高，但在三种传感器类型中，它们的性能最低。

总结来看，在黑暗中摄像头的性能明显弱于毫米波雷达和激光雷达。

- **Robustness in rain, snow, fog 在雨、雪、雾中的鲁棒性**

毫米波雷达传感器的最大优点之一是在恶劣天气条件下的性能优异，它们不会受到雪、大雨或空气中任何其他障碍物 (如雾或沙粒) 的显著影响。作为一种光学系统，激光雷达和相机容易受到恶劣天气的影响，其性能通常会随着逆境程度的增加而显著下降。

总结来看，在恶劣天气下毫米波雷达的性能优异，而摄像头和激光雷达受天气和环境的影响较大。

- **Classification of objects 物体分类**

摄像机擅长对车辆、行人、速度标志等物体进行分类，这是摄像系统的主要优势之一，在人工智能的最新进展中更是强调了这一点。使用高密度 3D 点云的激光雷达扫描也可以实现一定程度的分类，但是在检测的对象多样性上不如摄像头。而毫米波雷达系统不允许进行多物体分类。

总结来看，在多物体分类领域，摄像头的性能最佳，激光雷达的性能较佳，而毫米波雷达不能进行多物体分类。

- **Perceiving 2D structures 感知二维结构**

摄像机系统是唯一能够解释二维信息的传感器，例如速度标志、车道标记或交通灯，因为它们能够测量颜色和光强度，这也是摄像头相对于其他传感器类型的主要优势。

- **Measure speed 测量速度**

雷达可以利用多普勒频移直接测量物体的速度，这是雷达传感器的主要优势之一。激光雷达只能通过连续的距离测量来近似速度，这使得它在这方面的精度较低。尽管相机无法测量距离，但可以通过观察图像平面上物体的位移来测量碰撞时间。

总结来说，毫米波雷达可以直接测量速度，精度最高；而激光雷达只能测出近似速度，精度较低；不同的是摄像头只能测量时间。

- **System cost 系统成本**

近年来，雷达系统已广泛应用于汽车行业，目前的系统高度紧凑且价格合理。单目相机也是如此，在大多数情况下价格远低于 100 美元。由于硬件成本的增加和市场上数量的显著减少，立体相机的价格更高。激光雷达在过去几年中越来越受欢迎，尤

其是在汽车行业。由于技术进步，其成本已从 75000 多美元降至 5000 美元以下，许多专家预测，未来几年，激光雷达模块的成本可能会降至 500 美元以下。

总结来看，成本上，摄像头和毫米波雷达的成本较低，而激光雷达目前成本较高，未来激光雷达的成本可能会大幅缩小。

- **Package size 包装尺寸**

由于毫米波雷达和单摄像头体积较小，它们都可以很好地集成到车辆中。因为立体摄像头在某些情况下体积庞大，它们有时可能会限制驾驶员的视野，这使得它们更难集成在挡风玻璃后面。激光雷达系统有各种尺寸。360° 扫描激光雷达通常安装在屋顶顶部，因此从外观看非常清晰可见。目前行业向更小的固态激光雷达系统的转变，将在不久的将来大幅缩小激光雷达传感器的系统尺寸。

总结来看，单摄像头和毫米波雷达的尺寸较小，方便安装；立体摄像机和激光雷达尺寸较大，未来雷达传感器的体积可能会大幅缩小。

- **Computational requirements 计算要求**

激光雷达和毫米波雷达几乎不需要后端计算处理。虽然摄像头是一种成本低且易于使用的传感器，但它们需要大量计算处理才能从图像中提取有用信息，这增加了系统的总体成本。

3.3 安全性

安全意味着不存在导致系统危险的条件，而运行深度学习技术的自动驾驶系统的安全性在很大程度上取决于技术类型和应用环境。因此，关于深度学习技术安全性的推理需要考虑以下四点：

- 了解可能发生的故障所造成的影响；
- 了解更广泛的系统的背景；
- 定义有关系统和环境的假设；
- 定义安全行为的含义 (包括非功能性约束)。

在该小节中我们通过大量查阅相关文献按照以下十个模块的行文逻辑进行介绍：

3.3.1 对“安全”的定义

在机器学习和深度学习的背景下，对于“安全”这一术语实际上没有普遍同意的定义。在 [40] 中，Varshney 从风险、认知不确定性和意外结果引起的伤害方面定义了安全性。然后，他分析了成本函数 (cost function) 的选择以及最小化经验平均训练成本 (empirical average training cost) 的适当性。

3.3.2 对“事故”的定义

[41] 考虑了机器学习系统中的事故问题。此类事故被定义为人工智能系统设计不佳可能产生的意外和有害行为。作者列出了五个与事故风险相关的实际研究问题，问题的分类依据为以下三点：(1) 是否源于错误的目标函数；(2) 目标函数计算成本高，无法经常评估；(3) 学习过程中的不良行为 (安全探索和分配转移)。

3.3.3 扩大“安全”，提出广泛的安全决策论

[42] 扩大了安全范围，并提出了适用于广泛领域和系统的安全决策论定义。他们将安全性定义为：减少或者最小化那些严重到足以被视为有害的意外结果相关的风险和认知不确定性。该定义中的关键点有以下两点：(1) 在某些人类意义上，意外后果的成本必须足够高，事件才会被认定为有害；(2) “安全”涉及降低预期伤害的可能性以及意外伤害的可能性。

3.3.4 为确保安全提出“标准”

由于深度学习算法的复杂性，其可能会以独特的方式失败。尽管有了上述关于“安全”以及“事故”等方面的定义，在安全系统中使用深度学习组件仍然是一个问题。例如，在深度强化学习系统中，奖励函数中的错误会对训练模型产生负面影响 [41]。ISO 26262(道路车辆功能安全标准) 为确保安全提供了一套全面的要求，但这一套要求仍然没有解决基于深度学习的软件的独特特性。

3.3.5 扩大“标准”

[43] 通过分析机器学习可能影响标准的地方，解决了标准的差距，并就如何适应这种影响提供了建议。这些建议的重点是识别危险、实施故障和失效情况的工具和机制，同时确保完整的训练数据集和设计多级架构。

3.3.6 定义“危险”并作出“评估”

标准 ISO 26262 建议使用 **Hazard Analysis and Risk Assessment(HARA)** 来识别系统中的危险事件，并指定减轻危险的安全目标。该标准分共为 10 个部分，这里重点介绍第 6 部分：软件层面的产品开发，该标准遵循著名的 *V model for engineering*。**Automotive Safety Integrity Level (ASIL)** 是指 ISO 26262 中为汽车系统中的项目 (如子系统) 定义的风险分类方案。**ASIL** 表示降低风险所需的严格程度，其中 **ASIL D** 表示最高风险，**ASIL A** 表示最低风险。如果将要素分配给 **QM**(质量管理)，则不需要安全管理。**ASIL** 首先分配解决危险的安全目标集，然后由该目标衍生的安全要求继承 [43]，从而给定危险评估。

根据 ISO 26226, 危险被定义为“故障行为造成的潜在伤害源, 其中伤害是指身体伤害或对人的健康的损害” [44]。然而, 自动驾驶的深度学习和强化学习算法可能会产生新类型的危险。例如在 [45] 文章中提及了这种危险的目前经常发生, 并且人们普遍认为使用辅助驾驶比完全自动驾驶更可靠一些。

3.3.7 定义“损失函数”与“经验风险”, 实现数值化、函数化

分析深度学习算法安全性的一个关键要求是检查结果的直接人为成本是否超过某些危害严重性阈值。从人类的角度来看, 意外的结果确实是有危害的, 其影响几乎是实时的, 这些结果可归类为安全问题。深度学习决策的成本与优化公式有关, 其中重要的部分之一就是损失函数 L 。损失函数 $L: X \times Y \times Y \mapsto R$ 被定义为通过将观察值 x 的标签预测为 $f(x)$ 而不是 y 而产生的误差的度量。统计学习将 f 的风险称为 P 下 f 损失的期望值:

$$R(f) = \int L(x, f(x), y) dP(x, y) \quad (3.1)$$

其中, $X \times Y$ 是观测值 X 和标签 Y 的随机变量空间, 满足概率分布函数 $P(X, Y)$ 。统计学习问题包括找到优化 (即最小化) 风险 R 的函数 f [46]。对于算法的假设 h 和损失函数 L , 训练集上的预期损失称为 h 的经验风险 (empirical risk):

$$\mathbf{R}_{emp}(h) = \frac{1}{m} \sum_{i=1}^m L(x^{(i)}, h(x)^{(i)}, y^{(i)}) \quad (3.2)$$

3.3.8 理想情况与现实情况在程序和数据集上的差异

然后, 机器学习算法根据风险显著降低的预期来优化经验风险。然而, 该标准公式未考虑与安全相关的不确定性相关的问题。训练样本 $(x_1, y_1), \dots, (x_m, y_m)$ 的分布是从 (X, Y) 的真实潜在概率分布中得出的, 情况可能并不总是这样, 通常概率分布未知, 因此无法使用域自适应技术 [46, 47]。因为在不同分布的数据集上进行训练可能会产生不同的偏见从而造成很大伤害, 这正是一种与安全相关的认知不确定性。

实际上, 机器学习系统训练测试阶段只会处理有限数量的样本, 即使 h 是风险最优的, 对于小基数测试集, 操作风险可能远大于实际风险。由测试集实例化引起的这种不确定性可能对单个测试样本有很大的安全影响 [48]。

编程组件的故障和失效与深度学习组件的故障和失效完全不同。深度学习组件的特定故障可能由不可靠或噪声信号 (恶劣天气引起的视频信号、雷达信号、GPS 数据等)、神经网络拓扑、学习算法、训练集或环境中的意外变化 (例如未知驾驶场景上的事故) 引起。例如特斯拉汽车的第一次自动驾驶事故, 由于物体分类错误, 自动驾驶功能将车辆撞上了卡车 [8]。尽管进行了 1.3 亿英里的测试和评估, 但鉴于卡车的高度、明亮天空下

的白色卡车表面以及车辆在道路的特殊位置，这起事故可以算是在极为罕见的情况下造成的，这也被称为 Black Swans。

训练集对深度学习组件的安全性起着至关重要的作用。ISO 26262 标准规定，应充分规定组件行为，并根据其规范验证法则进行逐次改进。在深度学习系统中，因为使用的是训练集而不是规范集，所以违反了上面的这一假设，但目前还尚不清楚如何确保始终缓解相应的危险。训练过程不是验证过程，因为在模型和学习算法 [43] 的限制下，通过构造训练集得到的训练模型肯定是正确的，但这不代表该训练集涵盖全部的道路情形，由训练集中不存在的数据引起的黑天鹅事件往往可能导致事故的发生 [49]。

3.3.9 针对现实的差异制定“要求”

应针对现实中的情形制定详细要求，例如指定如何获得训练、验证和测试集。随后，可以根据这个规范要求去验证收集到的数据。例如，一些规范要求车辆宽度不能超过 3 米，这种规范要求可用于拒绝误报检测。例如在 [50] 中在训练过程中直接使用一些规范要求，以提高模型的准确性。即使自动驾驶系统的完整安全保证仍然是一个仍未解决的问题，机器学习和深度学习技术也开始变得有效可靠，汽车行业的现行标准和法规无法完全映射到此类系统，因此需要制定新的科学全面的安全标准，以供模型的学习。

3.3.10 故障保护机制

自动驾驶车辆必须具有故障保护机制 (fail-safe mechanisms)，该机制通常以安全监控器 (Safety Monitors) 的形式出现。一旦检测到故障，必须停止自动控制软件 [51]。论文 [52]、[53] 和 [49] 中对神经网络的特定故障类型和一般故障进行了分类。这促进了特定的、有针对性的工具和技术的发展，从而更好地发现故障。[54] 描述了一种可以调试由于不良训练数据导致的错误分类的技术，而 [55] 中提出了一种可以排除由于连接各个机器学习组件之间复杂交互而导致的故障的方法。在 [56] 中，白盒 (white box) 技术用于通过断开神经元之间的链接或随机改变权重将故障注入神经网络。

3.4 自动驾驶中深度学习的数据源

使用真实数据是训练和测试自动驾驶组件的关键要求。在过去几年中，主要由于人们对自动驾驶汽车的研究兴趣越来越大，许多驾驶数据集被公开并记录在案。它们在尺寸、传感器设置和数据格式上各不相同，数据格式和传感器设置没有标准要求，每个数据集类型在很大程度上取决于算法需要的类型，研究人员只需确定最适合其问题空间的适当数据集使用即可。[57] 发表了一项关于广泛数据集的调查。这些数据集通常涉及计算机视觉领域，但适合自动驾驶主题的数据集很少。

[58] 对自动驾驶车辆算法的公开可用数据集进行了最全面的调查, 该论文介绍了 27 个可用数据集, 其中包含公共道路上记录的数据。从不同角度对数据集进行比较, 以便读者可以选择最适合其任务的数据集。

Scale[®] 和 nuTonomy[®] 公司开始创建了迄今为止市场上最大、最详细的自动驾驶数据集之一。其中包括 Berkeley DeepDrive [59], 这是一个由伯克利大学研究人员开发的数据集。

在 [60] 中, 作者提出了一项旨在收集和分析半自动驾驶的大规模自然数据的研究, 以便更好地描述当前技术的最新状态。这项研究涉及 99 名参与者, 29 辆汽车, 405807 英里, 大约 55 亿个视频帧, 不过本研究中收集的数据不可供公众使用。

在本节剩余部分中, 我们将提供并强调开源的与自动驾驶相关数据集的显著特征。表 4 为常用的十种数据集的具体信息, 图 8 为数据集信息的可视化桑基图, 桑基图的具体可视化交互请见网页。

KITTI Vision Benchmark dataset (KITTI) 数据集 [61] 由德国 KIT 学院提供, 适用于立体视觉、光流、三维跟踪、三维物体检测或 SLAM 算法的基准测试。它被称为自动驾驶汽车领域最著名的数据集。数据采集车配备了多个高分辨率彩色和灰度立体摄像机、Velodyne 3D 激光雷达和高精度 GPS/IMU 传感器。总的来说, 它提供了卡尔斯鲁厄周围农村和公路交通场景中收集的 6 小时驾驶数据。该数据集由 Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License 提供的。

NuScenes dataset 数据集 [62] 由 nuTonomy 构建, 包含 1000 个从波士顿和新加坡收集的驾驶场景, 这两个城市以其密集的交通和极具挑战性的驾驶环境而闻名。为了便于执行常见的计算机视觉任务 (例如目标检测和跟踪), 提供者在整个数据集上用 2Hz 的精确 3D 边界框注释了 25 个对象类。车辆数据收集仍在进行中, 最终数据集将包括大约 140 万个摄像机图像、40 万个激光雷达扫描、130 万个雷达扫描和 40,000 个关键帧中的 110 万个对象边界框。该数据集由 Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License 提供的。

Automotive multi-sensor dataset (AMUSE) 数据集 [63] 由瑞典 Linköping University 提供, 由配备全方位多摄像机、高度传感器、IMU、速度传感器和 GPS 的汽车在各种环境中记录的序列组成。向公众提供了用于读取这些数据集的 API, 以及以给定格式存储的长多传感器和多摄像机数据流的集合。该数据集是根据 Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unsupported License 提供的。

Ford campus vision and lidar dataset (Ford) 数据集 [64] 由 Michigan 提供, 使用配备专业 Applanix POS-LV) 和消费者 (Xsens MTi-G) 惯性测量单元 (IMU) 的福特 F250 皮卡, 一个 Velodyne 激光雷达扫描仪, 两个推扫视里格尔激光雷达和 Point Grey Ladybug3 全方位相机系统收集。2009 年, 在福特研究院和密歇根州迪尔伯恩市中心记录了 100 GB

的数据。该数据集非常适合测试各种自动驾驶和同步定位和地图绘制 (SLAM) 算法。

Udacity dataset 数据集 [65] 使用的车辆传感器设置包括单目彩色摄像机、GPS 和 IMU 传感器, 以及一个 Velodyne 3D 激光雷达。数据集的大小是 223GB。对数据进行标记, 并向用户提供相应的人类驾驶员在测试运行期间记录的转向角度。

Cityscapes dataset 数据集 [66] 由德国戴姆勒公司研发中心, 德国 Max Planck Institute for Informat-ICS(MPI-IS), 德国 TU Darmstadt Visual Inference Group 提供, Cityscapes 数据集侧重于城市街景的语义理解, 这是它只包含立体视觉彩色图像的原因。图像的多样性非常大:50 个城市, 不同的季节 (春、夏、秋), 不同的天气条件和不同的场景动态。详细注释的图片 5000 张, 粗略注释的图片 20000 张。《Pyramid Scene Parsing Network》和《SGN: Sequential Grouping Networks for Instance Segmentation》两个重要的挑战是使用该数据集对语义分割算法的发展进行基准测试和实例分割。

Oxford dataset 数据集 [67] 收集由英国牛津大学提供, 跨度超过 1 年, 得到了超过 1000 公里的驾驶记录, 从安装在车辆上的 6 个摄像头收集了近 2000 万张图像, 以及 LIDAR, GPS 和惯性导航系统的地面实况。数据收集在所有天气条件下, 包括大雨, 夜间, 阳光直射和降雪。该数据集的一个特点是, 车辆在一年的时间内经常行驶相同的路线, 以使研究人员能够在真实世界、动态的城市环境中研究自动车辆的长期定位和地图绘制。

Cambridge-driving Labeled Video Dataset 数据集 (*CamVid*) [68] 由英国剑桥大学提供, 是文献中引用最多的数据集之一, 也是第一个公开发布的数据集, 包含了一个带有对象类语义标签的视频集合, 以及元数据注释。数据库提供了 ground truth 标签, 将每个像素与 32 个语义类之一关联起来。传感器的设置仅基于安装在车辆仪表板上的一个单目摄像机。场景的复杂度较低, 车辆仅在交通相对较低、天气条件较好的城市地区行驶。

Daimler pedestrian benchmark dataset 数据集 [69] 由戴姆勒公司和阿姆斯特丹大学提供, 适合行人检测、分类、分割和路径预测等主题。行人数据仅从交通车辆上使用车载单声道和立体摄像机观察获得。它是第一个包含行人的数据集。最近, 使用相同设置捕获骑自行车者, 并制成视频样本, 从而实现了数据集的扩展。

Caltech pedestrian detection dataset (Caltech) 数据集 [70] 由美国加州理工学院提供, 包含丰富的注释视频, 从移动的车辆上进行录制, 得到了很多具有挑战性的低分辨率图像, 并且有些图像会有所遮挡, 这正好为真实的模拟创造了条件。其中大约有 10 小时的驾驶场景累计约 250000 帧, 共计 35 万个边界框, 2300 条行人专属注释, 这些注释包括边界框和详细的遮挡标签之间的时间对应。

鉴于现有数据集的多样性和复杂性, 选择一个或多个来开发和测试自动驾驶组件可能很困难。可以看出, 在所有可用的数据集中, 传感器的设置各不相同。对于定位和车辆运动, 激光雷达和 GPS/IMU 传感器是必要的, 最流行的激光雷达传感器是 Velodyne

和 Sick。从雷达传感器记录的数据只存在于 NuScenes 数据集中。雷达制造商采用非公开的专有数据格式。几乎所有可用的数据集都包括从摄像机捕获的图像，而单目和立体摄像机的平衡使用主要用于捕获灰度图像。AMUSE 和 Ford 数据库是唯一使用全方位摄像头的数据集。

除了原始记录数据外，数据集通常还包含注释、校准文件、标签等文件。为了处理这些文件，数据集提供者必须提供工具和软件，使用户能够读取和处理数据。数据集的拆分也是一个需要考虑的重要因素，因为一些数据集 (如 Caltech、Daimler、Cityscapes) 已经提供了预处理数据，这些数据被分类在不同的集合中：训练、测试和验证。这使得针对类似方法的所需算法的基准测试保持一致。

另一个需要考虑的方面是许可证类型。最常用的许可证是 Creative Commons attribute - noncommercial - sharealike 3.0。它允许用户以任何媒介或格式复制和重新发布，也允许重新混合、转换和构建材料。KITTI 和 NuScenes 数据集就是这种分发许可的例子。Oxford 数据集使用 Creative Commons Attribution-Nonbusiness 4.0。与第一种许可证类型相比，它不会强制用户在与数据集相同的许可证下发布其贡献。与此相反，AMUS 数据集是根据 Creative Commons Attribution-Nonbusiness-NoDerivs3.0 授权的，如果对资料进行了修改，数据集的发布就是非法的。

除了极少数例外，这些数据集是从以个城市收集的，通常位于欧洲、美国或亚洲的大学校园或公司所在地附近。德国是驾驶录音车辆最活跃的国家。不幸的是，所有可用的数据集加在一起只覆盖了世界地图的一小部分。其中一个原因是数据的存储大小直接关系到传感器的设置和质量。例如，Ford 的数据集每行驶一公里大约需要 30GB，这意味着覆盖整个城市将需要数百 tb 的驾驶数据。大多数可用的数据集考虑阳光、日光和城市条件，这些都是自动驾驶系统的理想运行条件。

3.5 自动驾驶算力平台与域控制器设计

自动驾驶域控制器，也有人称其为智能驾驶域控制器，是智能驾驶汽车控制的核心所在。域控制器连接摄像头、激光雷达等传感器，车联网 V2X、组合导航等网路通讯部件，以及车辆线控单元，主要负责对传感器感知到的环境信息进行融合、识别和分类处理，结合地图定位对车辆行驶进行路径规划和决策，从而实现对车辆的精确控制和自动驾驶。

下面将对五种典型的域控制器设计进行分析。

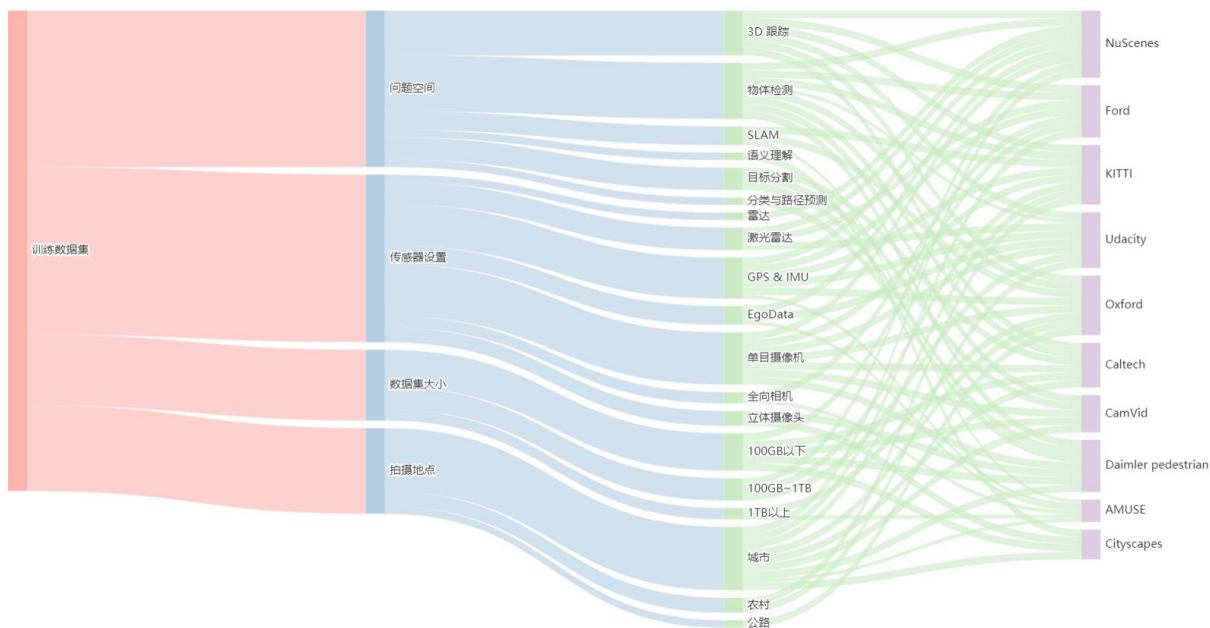


图8 自动驾驶系统训练数据集的可视化桑基图

表4 自动驾驶系统训练数据集摘要

数据集	问题空间	传感器设置	数据集大小	拍摄地点	交通情况	许可证
NuScenes	3D 跟踪,	雷达、激光雷达、摄像头,	345 GB	新加坡,	城市	CC BY -NC-SA 3.0
	3D 物体检测	GPS、IMU、EgoData				
AMUSE	SLAM	全向相机、IMU, EgoData、GPS	1 TB	洛杉矶	城市	CC BY -NC-SA 3.0
Ford	3D 跟踪,	全向相机、激光雷达,	100 GB	密歇根	城市	Not specified
	3D 物体检测	IMU、GPS				
KITTI	3D 跟踪、SLAM,	单目摄像头,	180 GB	卡尔斯鲁厄	城市、农村	CC BY -NC-SA 3.0
	3D 物体检测	IMU Lidar, GPS				
Udacity	3D 跟踪,	单目摄像机、激光雷达,	220 GB	山景	农村	MIT
	3D 物体检测	IMU、GPS、EgoData				
Cityscapes	语义理解	彩色立体摄像机	63 GB	达姆施塔特, 苏黎世、斯特拉斯堡	城市	CC BY -NC-SA 3.0
Oxford	3D 跟踪、SLAM	立体和单目摄像机,	23 TB	牛津	城市、公路	CC BY -NC-SA 3.0
	3D 物体检测	GPS Lidar, IMU				
CamVid	目标检测	单目彩色摄像机	8 GB	剑桥	城市	N/A
	目标分割					
Daimler pedestrian	行人检测、分类, 目标分割、路径预测	立体单目摄像机	91 GB	北京, 阿姆斯特丹	城市	N/A
Caltech	跟踪、分割, 目标检测	单目摄像机	11 GB	洛杉矶	城市	N/A

3.5.1 工控机

在前期算法预研阶段，推荐使用工控机 (Industrial PC, IPC) 作为最直接的控制解决方案。因为工控机相比于嵌入式设备更稳定、可靠，社区支持及配套的软件也更丰富。百度开源的 Apollo 推荐了一款包含 GPU 的工控机，型号为 Nuvo-5095GC(2016 年推出，只支持 75W nVidia® GPU 及后续 PASCAL GPU)，工控机具备特定硬件接口 (比如旧式接口)、具备较强抗压、抗震、防尘性能的计算机。当算法研究得较为成熟时，就可以将嵌入式系统作为控制器。

3.5.2 Tesla HW3.0

作为特斯拉自研专供域控制器，可供查询的资料不多，根据公开的资料分析其硬件构成：FSD 的 HW3.0 由两个相同的计算单元构成，每个计算单元上面有特斯拉自研的 2 块 FSD 计算芯片，每块算力为 36 Tops，设备总算力为 $4 \times 36 \text{ Tops} = 144 \text{ Tops}$ 。但是由于采用的是双机冗余热备的运行方式，实际可用的算力为 72 Top。

3.5.3 NVIDIA Driver AGX Xavier

提供每秒 30 万亿次运算 (TOPS)。其核心是 NVIDIA 首次生产的车规级 Xavier 系统级芯片，该芯片采用了六种不同类型的处理器，包括 CPU、GPU、深度学习加速器 (DLA)、可编程视觉加速器 (PVA)、图像信号处理器 (ISP) 和立体/光流加速器

AGX Pegasus™ 使用两块 Xavier 系统级芯片和两台 NVIDIA Turing™ GPU 的强大功能，实现了比以往更高的 320 TOPS 的超级计算能力。该平台专为 L4 级和 L5 级自主系统 (包括机器人出租车) 而设计和打造。

3.5.4 NVIDIA Jetson Xavier

非车规级 Xavier 系统级芯片，可以用来前期开发。芯片集成了 90 亿颗晶体管，CPU 采用 NVIDIA 自研 8 核 ARM64 架构 (代号 Carmel)，GPU 采用 512 颗 CUDA 的 Volta，支持 FP32/FP16/INT8，20W 功耗下单精度浮点性能 1.3TFLOPS，Tensor 核心性能 20TOPs，解锁到 30W 后可达 30TOPs。Xavier 内有六种不同的处理器：ValtaTensorCore GPU，八核 ARM64 CPU，双 NVDLA 深度学习加速器，图像处理器，视觉处理器和视频处理器。

3.5.5 华为 MDC

作为一个开放的平台，华为 MDC 具备组件服务化、接口标准化、开发工具化的特性，软件方面除了华为自有的鸿蒙平台之外，还兼容 Adaptive AUTOSAR 和 ROS，并且搭配配套的工具链，基于此平台，用户可快速开发、调试、运行自动驾驶算法与功能，

适配不同级别的智能驾驶应用。华为 MDC300 由华为 Ascend310 芯片、华为鲲鹏芯片和 Infineon 的 TC397 三部分构成，算力在 64Tops 左右，满足 L3 级自动驾驶算力需求。MDC600 基于 8 颗 Ascend310 AI 芯片，同时还整合了 CPU 和相应的 ISP 模块，算力高达 352TOPS。

介绍完五种典型的域控制器，下面我们来具体看一下域控制器的设计要求：

3.6 域控制器设计要求

域控制器作为一个智能硬件，为了完成复杂的 AI 计算和智能控制，硬件层面需要承担环境感知和深度学习等超大算力需求的 AI 处理芯片、负责控制决策和逻辑运算的 CPU、以及负责功能安全和车辆控制的 MCU；软件层面包括操作系统、中间件以及应用层 AI 算法等。

硬件层面的三部分芯片或者叫三颗重要芯片：

第一部分通常是 GPU 或 TPU，承担大规模浮点数并行计算需求，主要用于环境感知和信息融合，包括摄像头、激光雷达等传感器信息的识别、融合、分类等，如 Xavier 的 GPU 单元、Ascend310。

第二部分大多为 ARM 架构，类似于 CPU，主要负责逻辑运算和决策控制，处理高精度浮点数串行计算。

第三部分主要负责可靠性和车辆控制，也就是功能安全和冗余监控作用，不要求很高的算力，但是可靠性必须要有保障，ISO26262 等级要求达到 ASIL-D，目前用的较多的就是 Infineon 的 TC297 或者 TC397。

软件层面应用层 AI 算法一般由造车企业或主机厂自行研发，这是体现车辆性能和差性的所在；中间件一般由域控制器开发商提供，类似于 ROS，主要包含一些接口驱动和对于操作系统的优化部署。操作系统方面，华为采用自研鸿蒙系统，并且 MDC 平台支持兼容 Adaptive AUTOSAR；QNX 是目前车规级的最佳也是唯一选择，但是面临高昂的费用；Linux 和 Autoware 由于开源反而获取了最多的用户，尤其是 Linux，借助其强大的工程师基础和生态，加之 ROS 的多年应用基础，基于 Linux 内核的操作系统目前很受欢迎。

四、深度学习和强化学习在自动驾驶的应用

在这一部分，我们将自动驾驶的实现过程分成数个区块，分别概述各个区块内所使用的深度学习和强化学习方法。图9为本节介绍的自动驾驶领域常用的深度学习和强化学习方法关系图。

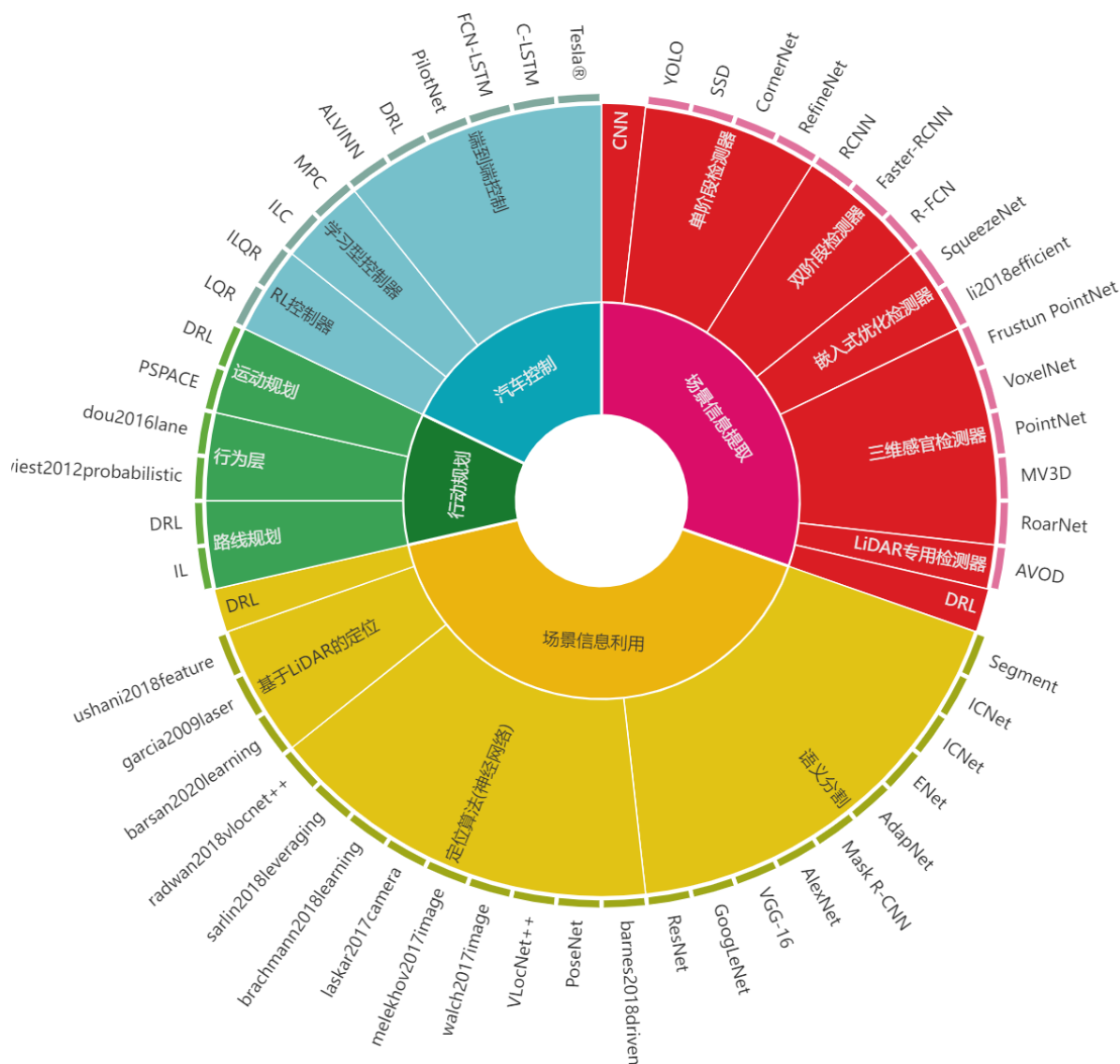


图9 一些深度学习和强化学习方法

4.1 场景认识

场景认识是一大关键模块。它将从感知模块获得的感知状态映射至高层的行动或决策模块。场景认识包括了三个任务：场景理解、决策和规划。场景认识的物质基础建立在探测或定位算法之上，旨在提供对周围场景更深入的理解。自动驾驶汽车常用的状态空间特征包括：车辆自身的位置、方向和速度以及传感器视野范围内的其他障碍物。融合异质传感器源能够使得在内容趋于抽象的情况下稳健地进行概括。在这样的信息融合之后，决策组件能够获得一般化、精简化的背景。融合过程最终会提供一个与传感器无关的环境表示，并对传感器的噪声和多种模式下的探测不确定性进行建模。其中常见的传感器包括激光雷达、照相机、雷达、超声。

4.1.1 场景信息提取

自动驾驶汽车必须能够检测交通参与者和可驾驶区域，特别是在城市地区——可能会出现各种各样的物体外观和遮挡物。基于深度学习的感知，特别是卷积神经网络 (CNN)，是物体检测和识别的砥柱技术。在 ImageNet 大规模视觉识别挑战赛等比赛中，CNN 获得了显著的成绩 [71]。不同的神经网络架构被用来检测作为二维兴趣区域的物体、图像中的像素分割区域、LiDAR 点云中的三维边界框，以及组合相机 LiDAR 数据中物体的三维表示。由于信息更丰富，图像数据更适合于物体识别任务。

用于图像中二维物体检测的最流行的架构是单、双阶段检测器。流行的单阶段检测器有“你只看一次”(Yolo) [72–74]、单次多箱检测器 (SSD) [75]、CornerNet [76] 和 RefineNet [77]。双阶段检测器，如 RCNN [78]、Faster-RCNN [79] 或 R-FCN [80]，将物体检测过程分成两部分：兴趣区域候选建议和边界盒分类。一般来说，单阶段检测器不能提供与双阶段检测器相同的性能，但速度明显更快。

如果车载计算资源匮乏，可以使用 SqueezeNet [81] 或 [82] 等检测器，它们被优化为在嵌入式硬件上运行。这些检测器通常有一个较小的神经网络结构，使其有可能使用少量的运行来检测物体，但要以检测精度为代价。

一些发表成果展示了原始三维感官数据的物体检测，以及视频和 LiDAR 信息的组合。PointNet [83] 和 VoxelNet [84] 被设计为仅从三维数据中检测物体，同时提供物体的三维位置。然而，单独的点云并不包含图像中的丰富视觉信息。为了克服这一点，相机-激光雷达的组合结构，如 Frustum PointNet [85]、多视角三维网络 (MV3D) [86] 或 RoarNet [87] 得到了使用。

在自动驾驶汽车的感官套件中使用激光雷达的主要缺点主要是其成本。这里的解决方案是使用神经网络架构，如 AVOD(聚合视图物体检测) [88]，它只利用 LiDAR 数据进行训练，而在训练和部署期间使用图像。在部署阶段，AVOD 能够仅从图像数据预测物体的三维边界框。在这样的系统中，LiDAR 传感器只需要用于训练数据的采集，就像今天用于收集导航地图道路数据的汽车一样。

在强化学习方面，深度强化学习 (DRL) 也作为一项重要的技术得到了使用。为了成功地将 DRL 应用于自动驾驶任务，设计适当的状态空间、行动空间和奖励函数是非常重要的。Leurent 等人 [89] 对自动驾驶研究中使用的不同状态和行动表示进行了全面的回顾。除去状态空间特征的识别外，为了避免状态空间维度的变化，经常使用围绕自动驾驶汽车的笛卡尔或极地占用网格。这进一步增加了车道信息，如车道号码 (自我车道或其他)、路径曲率、自我车辆过去和未来的轨迹、纵向信息，如碰撞时间 (TTC)，以及最后的场景信息，如交通法规和信号位置。

4.1.2 场景信息利用

驾驶场景的理解可以通过语义分割来实现，代表了图像中每个像素的分类标签。它是为完整的场景理解铺平道路的高级任务之一，被用于自动驾驶、室内导航或虚拟和增强现实等应用中。像 SegNet [90]、ICNet [91]、ENet [92]、AdapNet [93] 或 Mask R-CNN [94] 这样的语义分割网络大多拥有像素级分类层的编码器-解码器架构。这些都是基于一些常见网络拓扑结构的构件，如 AlexNet [2]、VGG-16 [23]、GoogLeNet [24] 或 ResNet [26]。

与边界盒检测器的情况一样，人们已经努力改善这些系统在嵌入式目标上的计算时间。在 [95] 和 ENet 中，作者提出了在自动驾驶的嵌入式设备上加快数据处理和推算的方法。这两个架构都是轻型网络，提供与 SegNet 类似的结果，但计算成本降低。在 AdapNet 中解决了语义分割的鲁棒性目标，进行了优化。该模型通过根据场景条件自适应地学习专家网络的特征，能够在各种环境中进行稳健的分割。使用 Mask R-CNN 等架构，可以得到一个结合的边界箱物体检测器和语义分割结果。该方法将 Faster-RCNN 的有效性扩展到实例分割上，在现有的边界盒识别分支的基础上增加了预测物体掩码的分支。

定位算法旨在计算自主车辆导航时的姿态 (位置和方向)。虽然这可以通过 GPS 等系统来实现，但也存在基于视觉定位 (VO) 的深度学习技术。深度学习可以通过直接影响关键点检测器的精度来提高 VO 的准确性。在 [96] 中，一个深度神经网络被训练用来学习单眼 VO 中的关键点分散器神经网络，如 PoseNet [97]、VLocNet++ [98]，或在 [99]、[100]、[101]、[102] 或 [103] 中介绍的方法，使用图像数据以 End2End 方式估计摄像机的三维姿态。场景语义可以和估计的姿态一起得出 [98]。

LiDAR 强度图也适合用于自动驾驶汽车的实时、校准无关的定位 [104]。该方法使用一个深度神经网络，从 LiDAR 扫描和强度图中建立一个驾驶场景的学习表示。车辆的定位是通过卷积匹配得到的。在 [105] 中，激光扫描和一个深度神经网络被用来学习城市和自然环境中的定位描述符。

为了安全地浏览驾驶场景，自动驾驶汽车应该能够估计周围环境的运动，也被称为场景流。以前基于 LiDAR 的场景流估算技术主要依赖于人工设计的特征。在最近的文章中，我们注意到一种趋势，即取代这些经典的方法用能够自动学习场景流程的深度学习架构。在 [106] 中，一个编码深度网络是在占用网格上训练的，目的是在连续的时间步长之间找到匹配或不匹配的位置。

尽管在基于深度学习的定位领域已经取得了很多进展，但 VO 技术仍然以经典的关键点匹配算法为主，并结合惯性传感器提供的加速度数据。这主要是由于关键点检测器的计算效率高，可以很容易地部署在嵌入式设备上。

一个车辆策略必须控制一些不同的执行器。车辆控制的连续值执行器包括转向角、油门和刹车。其他执行器，如换挡，是离散的。为了降低复杂性，并允许应用仅适用于离散动作空间的 DRL 算法 (例如 DQN)，可以通过将转向角、油门和刹车等连续执行器的范围划分为大小相等的仓来统一离散动作空间。也有人建议在对数空间中进行离散化，因为在实践中选择的许多转向角都接近于中心。然而，离散化确实有缺点；如果动作之间的步长值过大，它可能导致生硬或不稳定的轨迹。此外，当为一个执行器选择仓的数量时，有一个权衡，既要有足够的离散步数来实现平滑控制，又不能有太多的步数，以至于行动选择变得过于昂贵，无法评估。作为离散化的替代方案，执行器的连续值也可以由直接学习策略的 DRL 算法来处理 (例如 DDPG)。时间抽象选项框架) 也可以被用来简化选择行动的过程，代理选择选项而不是低级行动。这些选项代表了一个子政策，可以在多个时间步骤上扩展一个原始行动。

为自动驾驶的 DRL 代理设计奖励功能仍然是一个非常开放的问题。AD 任务的标准例子包括：向目的地行驶的距离，自我车辆的速度，保持自我车辆静止，与其他道路使用者或场景物体的碰撞，人行道上的违规行为，保持在车道上，以及在避免极端加速、制动或转向时保持舒适和稳定，和遵守交通规则。

4.2 运动规划

运动规划是确保目标点和目的地之间存在路径的任务。在通常由语义信息增强的先验地图上为车辆规划轨迹是一项不可或缺的任务。动态环境中的路径规划和变化的车辆动态是自动驾驶所需处理的关键问题。

DRL(DDPG) 在 AD 方面得到了应用，它使用一辆全尺寸的自主车辆。该系统首先在模拟中进行训练，然后使用车载计算机进行实时训练，并能够学会沿着车道行驶，成功地完成了在 250 米长的道路上进行的真实世界的试验。基于模型的深度 RL 算法已被提出，用于直接从原始像素输入中学习模型和策略。

在 [107] 中，深度神经网络被用来在模拟环境中生成数百个时间步长的预测结果。RL 也适用于控制。在 [108] 中，经典的最优控制方法如 LQR/iLQR 与 RL 方法进行了比较。经典的 RL 方法被用来在随机环境中进行最优控制，例如线性调节器 (LQR) 和非线性调节器的迭代 LQR(iLQR) 被利用。最近的一项研究 [109] 表明，对策略网络的参数进行随机搜索的结果可以和 LQR 一样好。

4.2.1 路线规划

自动驾驶汽车在两点之间，即起始位置和期望位置之间找到一条路线的能力代表了路径规划。路线规划根据路网地图定义行程的途经点，并且有可能采用实时交通数据作

为规划根据之一提出路线。

人们对使用深度学习技术进行路径规划和行为仲裁的兴趣越来越大。两个最具代表性的路径规划的深度学习范式为模仿学习 (IL) [110–112] 和基于深度强化学习 (DRL) 的规划 [113, 114]。

模仿学习的目标是通过记录驾驶经验来学习人类驾驶员的行为 [115]，采用 CNN 来从模仿中学习规划。例如，NeuroTrajectory [112] 是一个感知规划的深度神经网络，它在有限的预测范围内学习自我车辆的理想状态轨迹。模仿学习也可以被框定为逆向强化学习 (IRL) 问题，其目标是向人类驾驶员学习奖励函数 [116], [117]。这种方法使用真实的驾驶员行为来学习奖励函数，并产生类似人类的驾驶轨迹。

用于路径规划的 DRL 主要涉及在模拟器中学习驾驶轨迹 [113, 114, 118, 119]。基于转移模型，真实的环境模型被抽象化并转化为虚拟环境。在 [118] 中指出，目标函数不能确保功能安全而不引起严重的差异问题。针对这一问题提出的解决方案是构建一个由可学习和不可学习部分组成的策略函数。可学习的策略试图最大化一个奖励函数 (包括舒适性、安全性、超车机会等)。同时，不可学习的策略遵循功能安全的硬约束，同时保持一个可接受的舒适度水平。

用于路径规划的 IL 和 DRL 都有优势和劣势。IL 的优势在于它可以用从现实世界中收集的数据进行训练。然而，这种数据在拐弯处 (如驶出车道、车辆碰撞等) 是很稀缺的，这使得训练后的网络在面对未见过的数据时，反应不确定。另一方面，尽管 DRL 系统能够在模拟世界中探索不同的驾驶情况，但这些模型在被移植到现实世界时往往会有偏颇的行为。

4.2.2 行为层

行为层是自动驾驶的战略层面。考虑到当地的道路拓扑结构、交通规则和其他交通参与者的感知状态，代理人根据给定的路径点来决定短期政策。对于驾驶环境来说，有一套有限的可用行动，这一层的实现通常是一个有限的状态机，在其状态中具有基本的策略 (即汽车跟踪、改变车道等)，并根据环境的变化在它们之间有明确的过渡。然而，即使完全了解交通的当前状态，周围司机的未来意图也是未知的，这使得问题是部分可观察的 [120]。因此，未来的状态不仅取决于自我车辆的行为，而且还依赖于未知的过程；这个问题形成了一个部分可观察的马尔可夫决策过程 (POMDP)。存在不同的技术，通过预测其他道路使用者的可能轨迹来减轻这些影响，比如在 [121] 中，作者使用了高斯混合模型，或者在 [122] 中，根据记录的交通数据训练支持向量机和人工神经网络。由于有限行动 POMDPs 是对强化学习问题进行建模的自然方式，大量的研究论文涉及这一层面。

4.2.3 运动规划

为了执行行为层定义的策略，运动规划层需要设计一个可行的轨迹，包括车辆在短水平线上的预期速度、偏航和位置状态。当然，在这个层面上，必须考虑车辆的动力学，因此，经典的运动规划精确解决方案是不切实际的，因为它们通常是假设整体动力学的。众所周知，解决具有非整体动力学的运动规划问题的数值复杂性是多项式空间算法 (PSPACE) [123]，这意味着很难通过实时解决非线性编程问题来阐述一个整体解决方案 [124]。另一方面，该层的输出表示使其很难直接用“纯”强化学习来处理，只有少数论文只处理该层，而且它们通常使用 DRL 来定义花键作为训练的结果 [125], [126]。

经典的算法，如基于 Djisktra 算法的 A* 算法，在自动驾驶的非整体性情况下不起作用。快速探索随机树 (RRT) [127] 是一种非 holonomic 算法，通过随机采样和无障碍路径生成来探索配置空间。目前有各种版本的 RRT 用于自动驾驶管道的运动规划。

4.3 控制

一个控制器定义了从预先确定的地图 (如谷歌地图) 上获得的路径中的每一点所需的速度、转向角和制动动作，或专家在每个航点上记录的相同数值。与此相反，轨迹跟踪涉及到车辆动态的时间模型，在一段时间内按顺序查看航点。图10为端到端方法和传统学习型控制器方法的对比图。

4.3.1 学习型控制器

传统的控制器利用一个由固定参数组成的先验模型。当机器人或其他自主系统被用于复杂的环境，如驾驶，传统的控制器不能预见系统必须应对的每一种可能的情况。与固定参数的控制器不同，学习型控制器利用训练信息，随着时间的推移学习其模型。随着每一批训练数据的收集，真实系统模型的近似变得更加精确，从而实现了模型的灵活性、一致的不确定性估计以及对部署前无法建模的可重复影响和干扰的预测 [128]。在以前的工作中，学习型控制器是基于简单的函数近似器引入的，如高斯过程 (GP) 建模 [107, 108, 128, 129]，或支持向量回归 [109]。

学习技术通常用于学习动力学模型，从而改进迭代学习控制 (ILC) [130–133] 和模型预测控制 (MPC) [108, 128, 134–139] 中的先验系统模型。

迭代学习控制 (ILC) 是一种用于控制在重复模式下工作的系统的方法，如自动驾驶汽车的路径跟踪。它已被成功地应用于越野地形的导航 [130]、自主汽车停车 [131] 和自主赛车的转向动力学建模 [132]。它强调了多种好处，如使用简单和计算量小的反馈控制器，以及减少控制器的设计工作 (通过预测路径干扰和平台动态实现)。

模型预测控制 (MPC) [140] 是一种控制策略，它通过解决一个优化问题来计算控制

动作。在过去的二十年里，由于它能够处理具有状态和输入约束的复杂非线性系统，它受到了很多关注。MPC 的中心思想是在每个采样时间通过在短的时间范围内最小化成本函数来计算控制行动，同时考虑观察、输入-输出约束和由过程模型给出的系统动力学。对自主机器人的 MPC 技术的总体回顾见 [141]。

学习已经与 MPC 结合使用，以学习驾驶模型 [134,135]，在其操控极限下运行的赛车的驾驶动力学 [136,137]，以及提高路径跟踪的准确性。这些方法使用学习机制来识别非线性动力学，这些非线性动力学被用于 MPC 的轨迹成本函数优化。这使人们能够更好地预测干扰和车辆的行为，导致应用于控制输入的最佳舒适和安全约束。训练数据通常以过去的车辆状态和观察结果的形式出现。例如，CNN 可以用来在以机器人为中心的本地坐标系中计算密集的占用网格图。网格图被进一步传递给 MPC 的成本函数，以便在有限的预测范围内优化车辆的轨迹。

学习型控制器的一个主要优点是它们将传统的基于模型的控制理论与学习算法最佳地结合起来。这使得它仍然有可能使用既定的方法来进行控制器设计和稳定性分析，同时在系统识别和预测水平上应用一个强大的学习组件。

4.3.2 端到端控制

在自动驾驶的背景下，端对端学习控制被定义为从感觉数据到控制命令的直接映射。输入通常来自高维特征空间 (如图像或点云)。这与传统的处理管道相反，在传统的处理管道中，首先在输入图像中检测物体，然后规划路径，最后执行计算的控制值。

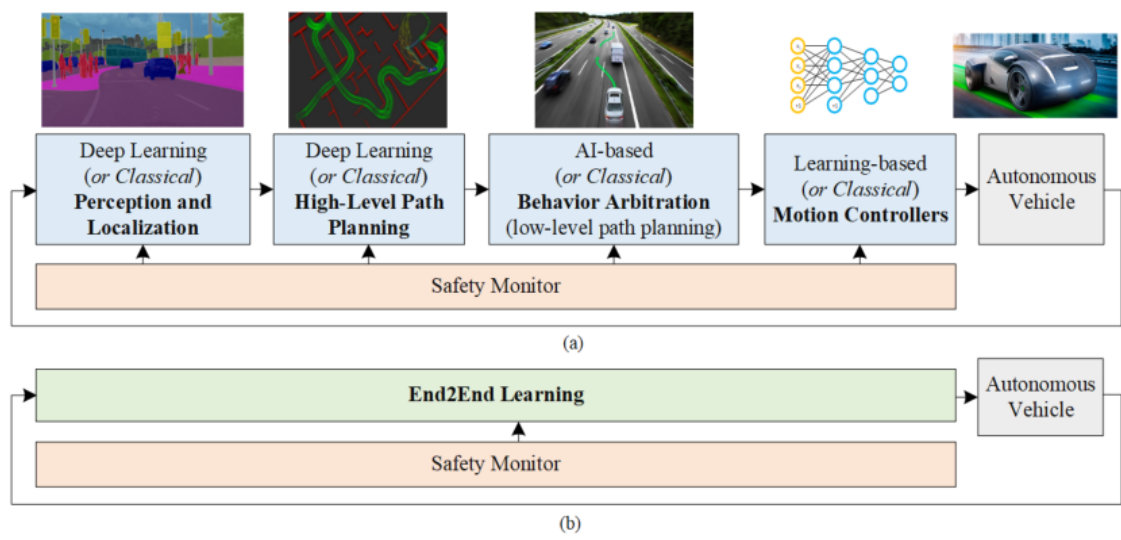


图 10 端到端方法和传统方法流程的对比

End2End 学习也可以被表述为一个扩大到复杂模型的反向传播算法。该范式在 20

世纪 90 年代首次被引入,当时建立了神经网络中的自主陆地车辆 (ALVINN) 系统 [142]。ALVINN 被设计为沿着预先定义的道路,根据观察到的道路曲率转向。End2End 驾驶的下一个里程碑被认为是在 2000 年中期,当 DAVE(Darpa Autonomous Vehicle) 成功地在充满障碍物的道路上行驶时,它已经在类似但不完全相同的驾驶场景中获得了数小时的人类驾驶训练 [143]。在过去的几年中,计算硬件的技术进步促进了 End2End 学习模型的使用。

End2End 控制论文主要采用在真实世界和/或合成数据上离线训练的深度神经网络 [60, 144–151],或在仿真中训练和评估的深度强化学习 (DRL) 系统 [152–154]。将模拟训练的 DRL 模型移植到真实世界驾驶的方法也有报道 [155],以及直接在真实世界图像数据上训练的 DRL 系统 [138, 139]。

End2End 方法在过去几年中被 NVIDIA® 推广,作为 PilotNet 架构的一部分。该方法是训练一个 CNN,将单个前置摄像头的原始像素直接映射为转向指令 [144]。训练数据是由在不同的照明和天气条件下以及在不同的道路类型上进行的驾驶场景中收集的图像和转向指令组成。在不同的道路类型上收集的图像和转向指令。在训练之前,使用增强技术来丰富数据,在原始数据中加入人工移位和旋转。

与 PilotNet 类似的 End2End 架构,将视觉数据映射到转向命令,已在 [148–150] 中报道。在 [145] 中,自动驾驶被表述为一个未来自我运动预测问题。引入的 FCN-LSTM(全卷积网络-长短期记忆)方法被设计为使用全卷积编码器联合训练像素级监督任务,同时通过时间编码器进行运动预测。在 [146] 中也考虑了输入数据的视觉时间依赖性之间的组合,其中 C-LSTM(卷积长短期记忆)网络被提出用于转向控制。在 [147] 中,环视摄像机被用于 End2End 学习。其主张是,人类驾驶员也使用后视镜和侧视镜进行驾驶,因此需要收集来自车辆周围的所有信息并将其整合到网络模型中,以便输出合适的控制命令。

为了开展对 Tesla® 自动驾驶系统的评估,[60] 提出了一个 End2End 卷积神经网络框架。它被设计用来确定自动驾驶和其自身输出之间的差异,并考虑到边缘情况。该网络使用真实数据进行训练,这些数据是从超过 420 小时的真实道路驾驶中收集的。特斯拉® 的自动驾驶仪和拟议的框架之间的比较是在特斯拉® 汽车上实时进行的。评估显示,在检测两个系统之间的差异以及将汽车的控制权转移给人类驾驶员方面,准确率达到 90.4

另一种设计端到端驾驶系统的方法是 DRL。这主要是在模拟中进行的,自主代理可以安全地探索不同的驾驶策略。在 [152] 中,一个 DRL End2End 系统被用来计算 TORCS 游戏仿真引擎中的转向指令。考虑到更复杂的虚拟环境,[153] 提出了一种异步优势演员批评法 (A3C) 方法来训练图像和车辆速度信息的 CNN。同样的想法在 [154] 中得到了加强,具有更快的收敛性和允许更多的泛化。这两篇文章都依赖于以下程序:接收游戏

的当前状态，决定下一个控制命令，然后在下一次迭代中获得奖励。实验设置得益于一个现实的汽车游戏，即世界拉力锦标赛 6，也得益于其他模拟环境，如 TORCS。

目前的车辆控制方法建立在经典的最优控制理论上，可以说是成本函数的最小化，定义在一组状态和控制动作上。控制输入通常在有限的时间范围内定义，并限制在可行的状态空间内 [156]。速度控制是基于经典的闭环控制方法，如 PID(比例积分衍生) 控制器，MPC(模型预测控制)。PID 的目的是最小化一个成本函数，该函数由三个项组成：比例项的当前误差，积分项的过去误差影响，以及导数项的未来误差影响。而 MPC 方法系列旨在稳定车辆的行为，同时跟踪指定的路径 [157]。最佳控制和强化学习是密切相关的，其中最佳控制可以被看作是一个基于模型的强化学习问题，车辆/环境的动力学被定义好的微分方程所模拟。强化学习方法被开发出来，以处理随机控制问题以及具有未知奖励和状态转换概率的不良问题。

在最底层，局部反馈控制负责最小化对规定路径或轨迹的偏离。尽管在这个层次上，行动空间变得连续，而经典的 RL 方法不能处理这个问题。因此，需要对控制输出进行离散化，或者—如一些论文中—使用 DRL 的连续变体。

基于 DRL 的控制的下一个趋势似乎是加入经典的基于模型的控制技术。经典的控制器提供了一个稳定的、确定的模型，在此基础上估计神经网络的策略。通过这种方式，建模系统的硬约束被转移到神经网络策略中 [158]。在 [138]，[139] 中提出了一个根据真实世界图像数据训练的 DRL 策略，用于激进性驾驶的任务。在这种情况下，一个被称为学习者的 CNN 在训练时由一个模型预测控制器提供的最佳轨迹实例进行训练。

五、 自动驾驶未来发展

在本节中基于前三节全方面宽领域介绍的各个部分，首先对当前自动驾驶领域所面临的挑战进行总结，其次就硬件、安全性、数据集以及算力平台进行合理预测，然后对未来自动驾驶领域的深度学习和强化学习算法做出展望，最后又详细地介绍一下当前自动驾驶领域热门的 BEV 领域，并根据分析得到未来 BEV 的发展趋势。图11为该节的各个模块的关系图，具体可视化交互见网页。

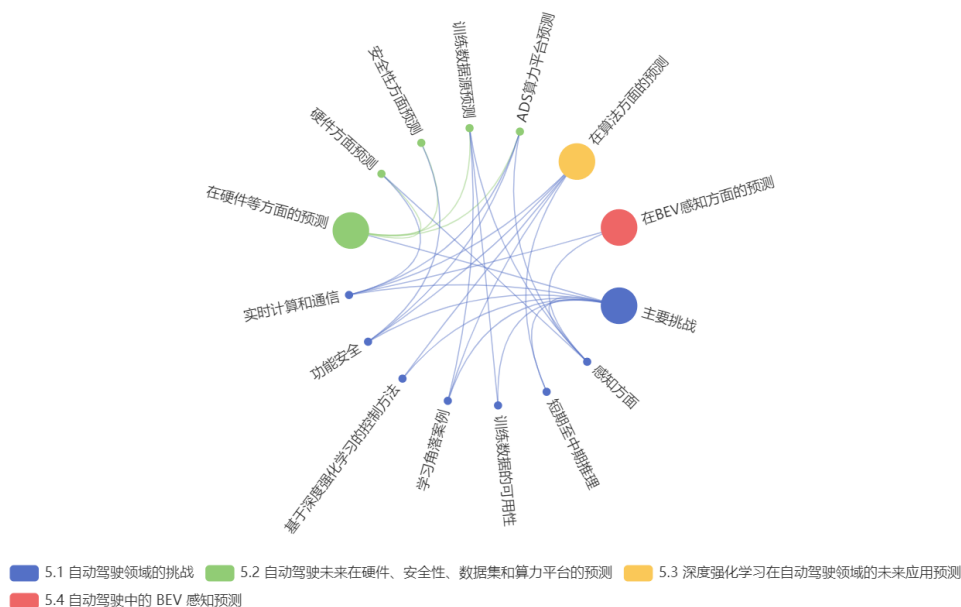


图 11 自动驾驶未来发展模块关系图

5.1 自动驾驶领域的挑战

我们对自动驾驶领域目前面临的挑战进行总结，归纳为七个主要领域。我们相信，深度学习和强化学习将在克服这些挑战方面上发挥至关重要的作用，下面具体介绍一下这七个挑战：

5.1.1 感知方面

为了让自动驾驶汽车安全地在驾驶场景中行驶，那么它必须能够感知理解周围环境。深度学习是大量感知系统背后的主要技术，尽管跟据报道 [159]：在物体检测和识别的准确性方面深度学习取得了很大进展，但当前的系统主要在设计得到用于计算物体类别的 2D 或 3D 边界框或可以提供驾驶环境的分割图像，这使得感知的细节水平较弱。未来的感知方法应侧重于提高识别细节的水平，使其能够实时感知和跟踪更多的对象。此外，还需要额外的工作来弥合基于图像和基于激光雷达的 3D 感知之间的差距 [160]，从而使计算机视觉界能够结束当前关于相机与激光雷达作为主要感知传感器的争论。

5.1.2 短期至中期推理

除了高鲁棒和准确率的感知系统外，自动驾驶车辆还应能够在短 (毫秒) 到中 (秒到分钟) 的时间范围内推理出驾驶行为 [161]。人工智能和深度学习是很有前途的工具，可用于驾驶场景所需的路径规划。目前，关于自动驾驶汽车深度学习的论文大部分主要集中在感知和末端学习 [158]。我们认为导航系统的长期推理问题已解决，所以在下一阶

段我们预计深度学习将在局部轨迹估计和规划领域发挥重要作用。

5.1.3 训练数据的可用性

“Data is the new oil”最近成为自动驾驶行业最流行的一句话。深度学习系统的有效性和准确性与训练数据集的可用性直接相关。目前的深度学习方法通常基于训练数据的质量进行评估 [57]。数据质量越好，算法的精度往往就越高。自动驾驶车辆记录的每日数据约为 PB 级别，这么大的数据量也给训练过程的并行化以及存储基础架构带来了挑战。在过去的几年中，模拟环境已经被用于弥补稀缺数据和深度学习的期望训练数据之间的差距，但在模拟世界和真实世界驾驶的准确性之间仍然存在着差距。

5.1.4 学习角落案例

大多数驾驶场景通常被认为可以用经典方法解决。然而，那些剩余的未解决方案被称为角落案例，到目前为止，这些案例需要人类驾驶员的主观推理。为了克服角落案例情况，应提高深度学习算法的泛化能力。深度学习中的泛化对于学习可能导致事故的危险情况具有特别重要的意义，特别是在该危险的角落案例被训练的很少的情况下也要尽可能做出正确的判断。

5.1.5 基于深度强化学习的控制方法

经典控制器使用由固定参数组成的先验模型，在复杂的情况下，如自动驾驶，这些控制器无法预测所有驾驶情况。深度学习组件基于过去经验进行调整的有效性也可用于学习汽车控制系统的参数，从而更好地逼近潜在的真实系统模型 [108]。但是如何设计深度学习算法和如何调参也是一项挑战。

5.1.6 功能安全

虽然目前的技术正在努力使计算智能和人类大脑更加接近，但是在自动驾驶这一安全关键系统中使用深度学习仍然是一个公开的辩论问题。当前的安全标准如 ISO 26262，不适用于机器学习软件 [43]。虽然已经提出了新的数据驱动设计方法，但深度神经网络的可解释性、稳定性或分类鲁棒性仍然存在开放性、政论性问题。

5.1.7 实时计算和通信

最后，必须满足实时要求，以处理从汽车传感器套件收集的大量数据，并通过高速通信线路更新深度学习系统的参数 [162]。这些实时限制可以通过专用于自动驾驶汽车的半导体芯片的进步以及 5G 通信网络的兴起来缓解。

5.2 自动驾驶未来在硬件、安全性、数据集和算力平台的预测

5.2.1 硬件方面

根据第三节的介绍，目前主流的驾驶传感器分为三类：摄像头、毫米波雷达和激光雷达，虽然综合性能上激光雷达的表现优于毫米波雷达和摄像头，但在受天气和环境影响的方面毫米波雷达表现更为优异，且在成本和体积方面摄像头和毫米波雷达的优势更加明显。

同时近几年尤其是激光雷达不断向着更低成本、更小体积、更高性能发展，所以未来可能激光雷达的使用率会不断增大，这必然会提升检测的性能。

所以如何合理地搭配这三种传感器、如何使得每种传感器增强优点降低缺点成为了未来的一个研究方向。

而根据第四节的介绍，不同的传感器存在相当的差异。因此先前的深度学习、强化学习研究进行了针对不同传感器的算法比较、适配算法设计等工作。今后的硬件发展会伴随着适配算法的同步跟进。未来的硬件配套算法将会更加具有针对性、高效性、泛用性。

5.2.2 安全性方面

根据第三节的介绍，因为人工智能和人类之间、在传统机器学习和深度强化学习之间存在区别，所以对于“规则”的制定需要非常严格的定义和进一步不断完善，随着对“安全”、“事故”、“标准”、“危险”、“评价指标”、“损失函数和经验风险”的不断完善，深度学习和强化学习自动驾驶领域有关“安全性”的理论基础更加合情合理，但是介于曾经“黑天鹅”理论的影响，人们仍需要不断对“安全性”的标准进行修改与补充，提出更多合理的假设以便应对日常生活中的各种突发情况，从而在保证安全性的同时努力提高自动驾驶的等级，尽快实现 5 级自动驾驶。

所以，在未来深度学习和强化学习的自动驾驶领域，不断针对现实生活中未曾考虑到的情形进行“规则”的合理补充和完善，同时完善故障保护机制算法，在遇到危险时尽可能降低损失，这都将成为了未来一项任务。

根据第四节的介绍，出于安全性考虑，深度学习、强化学习等算法必须在现有的基础上继续发展提升。未来的算法发展方向一是数学模型的优化，在现已成熟的模型上寻求提升的空间；二是运算效率的提升，在非硬件层面减少训练、识别、判断、规划所需的时间；三是训练过程的改良，通过调整训练数据集、优化训练过程以实现模拟环境的充分利用；四是实时控制的强化，准确性以及对突发情况的应对将是接下来算法的着力点之一，这一点上将会有应对多要素(其他行人、汽车等)的算法创新以及关于对抗性测试的研究。

5.2.3 数据源

根据第三节的介绍，虽然市面上的公开数据集有很多，但是鉴于数据集的多样性和复杂性，这就包含拍摄设备的不同、数据结构的不同、注释标准的不同、工具软件的不同、拍摄天气环境的不同以及许可证类型的不同，上面这些差异都会导致无法只通过一个或多个来开发测试所有的自动驾驶组件。目前各个项目组都是根据自己所需数据集的类型进行选择。另一方面，即便现在很多数据集规模很大，但仍然无法覆盖现实生活中的所有情形。同时不断增加现实情形时可能会使得自动驾驶的性能提高，但是随之带来的数据集规模的增大会使得算法计算成本增加、计算时间增长。

所以，在未来如何根据所需选择和拍摄采集数据集，如何尽可能在计算成本和计算时间可控的情况下选择和截取更加优异全面的数据集，这将成为未来的一项任务。

5.2.4 ADS(自动驾驶算力平台)

当前 ADS 自动驾驶采用决策层后融合的方式，其局限性主要表现在在极端恶劣气候与复杂遮挡等不确定性场景下分别进行单模结构化信息提取后再进行融合决策，每个通道信息会有不同层面丢失，很难能够进行多模有效互补与特征提取共享，算力内卷且性能远低于预期。

未来 ADS 算法会进入一个全新的 2.0 阶段，4D 空间下基于时空的多模感知与融合推理，也就是特征提取/统计推断/应急预测相结合，实现在动态复杂的有噪声干扰等场景下，全程安全无碰撞的高效行驶。挑战可以体现在，动态随机的人车物交互，多变天气路况，以及突发交通事件等。

大算力时代，ADS 系统首先是模仿人类的驾驶行为，通过注意力机制，期望在感知定位预测规划控制领域提供远超人类的决策能力。这需要在 AI 的三要素 (算法、算力、数据) 基础上添加第四要素，知识或者常识。而上述要素，均需要在充分理解算法的快速迭代的大趋势下，拥有充足的超大通用算力，ADS 系统在离线模仿学习人类驾驶先验知识经验与规则的基础上，能够提供在线自主学习能力，通过自学习激励与博弈共赢策略应对众多不确定性的人机交互的复杂环境，能够对决策不充分的场景下做到安全应对和提供可信解释能力。此外，大算力芯片也需要能够通过额外的算力，对芯片内百万级的并行计算单元提供故障检测与安全规避能力，对大量多模传感器的部分或者完全失效进行有效检测与应对决策处理，对驾驶环境针对传感器的主动攻击与外界目标非主动干扰进行有效检测与实时决策处理。

5.3 深度强化学习在自动驾驶领域的未来应用

5.3.1 概述

自动驾驶领域中可以由强化学习完成的任务包括但不限于以下方面：控制器优化、路径规划和轨迹优化、运动规划和动态路径规划、为复杂导航任务开发高级驾驶策略、高速公路、交叉路口、合并和拆分的基于场景的策略学习，预测行人、车辆等交通参与者的意图，并最终找到确保安全和执行风险估计的策略。

5.3.2 深度强化学习在自动驾驶仿真测试中的应用

目前强化学习的先进研究成果大多基于 simulation，end2end 的强化学习模型作为自动驾驶车辆的整车控制器还有很长的一段路要走。而在仿真测试环境中正确地评估自动驾驶性能、为自动驾驶算法的迭代建立具有动态性的学习环境，RL 算法则具有一些天然的优势。

在仿真环境中，由于真实的驾驶环境极其复杂，每一时刻、每一驾驶员都在根据当前路况做出连续的决策动作，基于 rule-based 设计的控制器难以表达如此繁复而深的决策分支。而 DRL 方法具有良好的对高维信息的提取能力，使得智能体可以从繁杂的环境中抽象出隐藏映射，并学习到最大化 reward 的策略。在实车测试中，我们更为关注驾驶场景中的 corner cases —— 类似道路突发事件或是不遵守交规的驾驶员，这些样本充斥在数据分布的长尾中，难以从真实交通环境中获取。例如在测试过程中，需要使用违规驾车辆用于评价待测车辆面对突发状况的处理能力，而真实数据中只有相当少比例的该类数据 —— 甚至会没有，因而导致了 sample efficiency 低下的问题。因此，无论是处于测试目的还是训练目的，具有动态性和针对性的仿真交通场景，能否被快速、准确地搭建，将决定自动驾驶算法的迭代的速度。

5.3.3 强化学习对自动驾驶研发的影响趋势

由于 low sample efficiency，RL safety，sim2real gap 等强化学习领域经典问题和挑战的存在，当前自动驾驶领域的强化学习研究大多还集中于 simulation 环境。即便如此，学术界和工业界依然看到了强化学习方法在自动驾驶领域巨大的应用潜力，如果强化学习搭配上高精传感器和高速响应的计算芯片，整体自动驾驶系统能力的“天花板”将是极高的。另一个趋势是和 DRL 相关的。经典的基于模型的控制技术提供稳定且确定的模型，建模系统的硬约束可以被转移到神经网络策略中。在这项技术下，得以使用 CNN 进行带有攻击性的对抗性测试。这样可以更大程度的适应驾驶的实际情况——包含了会违反规则的行人、车辆等的道路，以提升模型的实用能力。

5.4 自动驾驶中的 BEV 感知

5.4.1 概述

当前，工业界对相机 3D 感知有不同的探索路径，大体上可以分为 Image-view 和 BEV 方法两种。

Image-view 方案使用不同网络完成感知子任务，最后通过以规则为主的融合方法对不同网络的感知结果进行融合。

区别于 Image view 方案，BEV 方案通常采用 Transformer 将 Image feature 转换至 BEV 视角进行相关感知任务。

与此同时，学术界也有许多相关工作旨在利用纯视觉输入来完成自动驾驶感知任务，例如 3D 目标检测或者构建语义地图。但目前可参考的公开研究方法中，存在一个共性问题：现阶段基于视觉的 3D 目标检测方法并没有充分利用时序信息。

5.4.2 时空融合的必要性和可行性

时序信息对于自动驾驶感知任务十分重要。一方面，时序信息可以作为空间信息的补充，更好地检测当前时刻被遮挡的物体，为定位物体的位置提供更多参考信息。另一方面，时序信息能够帮助判断物体的运动状态，在缺少时序信息的条件下，基于纯视觉的方法几乎无法有效判断物体的运动速度。

从 HDMapNet，到 DETR3D 和 BEVDet，BEV 环视相机感知开始逐渐进入人们的视野。而近期 BEVFormer 和 PersFormer 分别在 3D 物体感知和单相机 3D 车道线感知任务中发力，BEVFusion 更是将触手伸向 Sensor Fusion 层面，也即在 BEV 视角下，如何做到多模态数据的融合感知。

BEV 环视感知的核心问题在于 View Transformation，也即如何从 Front View 转移到 BEV 时得到更有效的特征图。

这实际上是 2D 到 3D 空间的转换问题，在学术界目前主要有两种解决方法：3D-2D 转化，或 2D-3D 转化。如果从图像生成 BEV，再利用 BEV 进行检测，容易产生复合错误。因为在 3D 空间的信息投射到 2D 上，会聚集到一个点，无法分辨出这个点属于 3D 空间的哪一个部分，也即无法从 2D 图像上定位深度信息。对此学术界提出了对应的解决方案：

- (1) 3D to 2D：根据 3D 到 2D 的投影，对 **2D 局部特征进行采样**，最后得出 3D 特征。
- (2) 2D to 3D：**预测深度**，主流方法有 LLS 深度分布预测 (Lift, Splat, Shoot) 及 Psuedo Lidar Family 稠密深度预测。

5.4.3 BEVFormer

BEVFormer 是基于可变形注意力机制 (Deformable Attention) 实现的一种融合**多视角相机** (multi-camera) 和**时序特征**的**端到端框架**，适用于多种自动驾驶感知任务，检测算法具有鲁棒性。BEVFormer 由三个关键模块构成：

(1) BEV 询问向量 Q(BEV Queries Q)：查询得到的 BEV 特征图

使用一组预先设定的**可学习参数**——**询问向量 Q**，用来表征鸟瞰图特征，通过对空间信息和时序信息的轮番查询，能够将时空信息聚合在每个 BEV 询问向量的特征中。该 BEV 特征能够支持包括 3D 目标检测和地图语义分割在内的多种自动驾驶感知任务。

(2) Spatial Cross-Attention：融合多视角特征

基于稀疏交叉注意力模型，以投影点作为参考点，在周围进行特征采样，BEV 询问向量使用加权的采样特征进行更新，从而完成了空间的特征聚合。

(3) Temporal Self-Attention：融合时序 BEV 特征

每一时刻生成的 BEV 特征都从上一时刻的 BEV 特征获取所需的时序信息，这样可以保证获取的时序特征是动态更新的。

每个 BEV 询问向量既能通过 Spatial Cross-Attention 聚合空间特征，也能够通过 Temporal Self-Attention 聚合时序特征，这个过程会重复多次确保时空特征能够相互促进，进行更精准的特征融合。

5.4.4 BEVFormer++：Waymo 纯视觉 3D 检测挑战赛中斩获第一

Waymo Open Dataset 是 Waymo 推出的用于自动驾驶的公开数据集，是 CVPR 挑战赛之一，同时也是自动驾驶领域中算法研发方面最著名的国际性大赛，一直以来都以数据来源真实的驾驶场景、数据规模庞大、任务难度高而著称。

2022 Waymo 挑战赛聚焦于计算机视觉算法在解决自动驾驶运动和感知问题方面的进展。凭借对视觉感知技术多年的沉淀积累，**BEVFormer++ 在今年的纯视觉 3D 检测挑战赛中荣获第一**。

BEVFormer++ 结合了时序信息、多检测头集成，和 LET-IoU NMS 等有效模块，能够在自动驾驶场景下对目标进行更加精准的检测，同时能够有效去除假阳结果。

值得一提的是，BEVFormer++ 相比基线方法获得了**超过 60% 的性能提升**，最终取得了 **56.2 LET-mAPL** 的成绩，在最终榜单上大幅领先其他参赛队伍。

本次参与 Waymo 挑战赛上的 BevFormer++ 与 BEVFormer 相比有多种性能的提升，主要体现在以下五个方面：

(1) View Transformation 优化

采用更大的骨干网络，在同等训练设置下深度预训练优于 ImageNet 预训练。同时在图像特征中引入了 Coner Pooling 以增加特征感受野，从而提升 BEV 特征质量，并利用 Deformable View Encoder 增强了特征的多尺度融合。

(2) 时序特征优化

用 3x3 conv 代替时序注意力中的 Linear Offset 预测，提升了对于移动物体和相机参数不准等情况的鲁棒性。

(3) BEV 下多种检测器应用

不同检测器的结构设计对于不同的数据各有优劣，因此在 BEV 特征上采用了三种不同种类的典型检测器，并将所有模型用于 Ensemble 最终以提升性能。

(4) 多模型 Ensemble 策略能力提升

充分利用基于不同场景、不同类别数据下训练的 Expert Model，加上不同结构的模型和每个模型的 TTA (Test Time Augmentation)，利用遗传算法搜索得到最优 Ensemble 参数。

(5) 基于 LET IoU 的 Assignment & NMS

在 Assignment 和 NMS 中用 LET IoU 替代 LET-3D-AP，得到更偏向于径向分布的 BEV 特征点，更集中于图像上物体对应特征，从而使性能得到了提升，对于行人等小物体的去重效果也更好。

5.4.5 BEV 后续的发展

我们认为，未来 BEVFormer 还将持续以纯视觉为基础，持续提升物体环视检测性能，弥补纯视觉的 3D 检测与 Lidar 物体检测的性能差异。目前发现的潜力点集中在：提升 3D 检测性能的深度预训练、利用时序信息解决深度信息问题、以及能从根本上解决 BEV 检测性能的模型设计等。除了 BEV 感知本身，多模态信息融合也是一个关键。同时将 BEVFormer 部署到车端也是很有挑战性的工作。

六、结论

如今，自动驾驶技术不论是在硬件还是软件方面都有了极大的进步，使自动驾驶智能体未来享有与人类驾驶者相同的路权成为可能。

自动驾驶技术中的感知、决策、规划、控制系统，充分地识别道路中各类动态、静态目标、实现自主决策和车辆运行。并通过了长时间的虚拟仿真测试以及道路测试，极大程度上满足了安全性需求。而目前的自动驾驶技术等级还不够高，仍然停留在智能辅助驾驶上，已经开发应用的 AEB(自动紧急制动系统)，LKA(车道保持辅助系统)，BSD(盲区监视系统) 等等都已经在提升驾驶安全性上做出了卓越贡献。

自动驾驶技术同样在城市规划以及通勤效率方面能够起到举足轻重的作用。并且更加高效的通勤效率也有利于能源利用，对环保事业具有重要意义。

硬件方面，不仅有传统高科技企业入局进入自动驾驶领域参与硬件平台开发，例如英伟达以及华为，还有高科技自动驾驶新能源汽车品牌为自动驾驶开发整车平台并为其优化能耗以及布局。目前面对的共性问题依然是算力能效以及传感器融合在成本、算力以及极端场景下的融合方案。

而在软件方面，一方面在自动驾驶的深度学习领域对“安全性”等一系列的要求和标准仍没有完善统一的规定，另一方面在训练数据集上存在格式不统一、数据量大、角落案例多等情形。在算法方面，充分提取信息、提升计算效率、增加算法精确度、实现旧算法改进和新算法创新等工作都是当今和未来工作的关键点。所以目前面对的共性问题依然是假设的完善、数据集的制作和选择、计算效率的提升、信息的全面利用和学习模型的革新。

最后，我们指出了现有的主要研究要面临的挑战并确定了自动驾驶未来可能的发展方向。我们相信我们的调查研究能够帮助自动驾驶研究者提供一定程度的支持与帮助。

参考文献

- [1] Ayyoub Lamssaggad, Nabil Benamar, Abdelhakim Senhaji Hafid, and Mounira Msahli. A survey on the current security landscape of intelligent transportation systems. IEEE Access, 9:9180–9208, 2021.
- [2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.
- [3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. The International Journal of Robotics Research, 39(1):3–20, 2020.
- [4] Yoav Goldberg. Neural network methods for natural language processing. Synthesis lectures on human language technologies, 10(1):1–309, 2017.
- [5] J SAE. 3016 (2014). taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. Society of Automotive Engineers, 1, 2014.
- [6] O Bay. Abi research forecasts 8 million vehicles to ship with sae level 3, 4 and 5 autonomous technology in 2025, 2021.
- [7] Rasheed Hussain and Sherali Zeadally. Autonomous cars: Research results, issues, and future challenges. IEEE Communications Surveys & Tutorials, 21(2):1275–1313, 2018.
- [8] Sam Levin. Tesla fatal crash: ‘autopilot’ mode sped up car before driver killed, report finds. The Guardian, 8, 2018.
- [9] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. Machine vision and applications, 1(4):223–240, 1988.
- [10] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. IEEE Communications Surveys & Tutorials, 22(2):998–1026, 2020.
- [11] Zouhair Elamrani Abou El Assad, Hajar Mousannif, Hassan Al Moatassime, and Aimad Karkouch. The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review. Engineering Applications of Artificial Intelligence, 87:103312, 2020.

- [12] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 2021.
- [13] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. Journal of Field Robotics, 37(3):362–386, 2020.
- [14] Badr Ben Elallid, Nabil Benamar, Abdelhakim Senhaji Hafid, Tajjeeddine Rachidi, and Nabil Mrani. A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving. Journal of King Saud University-Computer and Information Sciences, 2022.
- [15] Szilárd Aradi. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 2020.
- [16] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. IEEE access, 7:53040–53065, 2019.
- [17] Abhishek Gupta, Alagan Anpalagan, Ling Guan, and Ahmed Shaharyar Khwaja. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. Array, 10:100057, 2021.
- [18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [19] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.
- [20] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, pages I–I. Ieee, 2001.
- [21] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. Pattern recognition, 29(1):51–59, 1996.
- [22] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05), volume 1, pages 886–893. Ieee, 2005.

- [23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.
- [25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [27] François Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017.
- [28] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-first AAAI conference on artificial intelligence, 2017.
- [29] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500, 2017.
- [30] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.
- [31] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. Neural computation, 31(7):1235–1270, 2019.
- [32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
- [33] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. Neural computation, 12(10):2451–2471, 2000.

- [34] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2625–2634, 2015.
- [35] Lei Tao, Zhang Yu, and A Yoav. Training rnns as fast as cnns. arXiv.org, 1709, 2017.
- [36] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. Advances in neural information processing systems, 30, 2017.
- [37] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. IEEE access, 6:1155–1166, 2017.
- [38] Phil Winder. Reinforcement learning. O’Reilly Media, 2020.
- [39] Zihan Ding. Algorithm table. In Deep Reinforcement Learning, pages 485–488. Springer, 2020.
- [40] Kush R Varshney. Engineering safety in machine learning. In 2016 Information Theory and Applications Workshop (ITA), pages 1–5. IEEE, 2016.
- [41] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. arXiv preprint arXiv:1606.06565, 2016.
- [42] Niklas Möller. The concepts of risk and safety. Handbook of risk theory: epistemology, decision theory, ethics, and social implications of risk, 1:55–85, 2012.
- [43] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An analysis of iso 26262: Using machine learning safely in automotive software. arXiv preprint arXiv:1709.02435, 2017.
- [44] Bernd Spanfelner, Detlev Richter, Susanne Ebel, Ulf Wilhelm, Wolfgang Branz, and Carsten Patz. Challenges in applying the iso 26262 for driver assistance systems. Tagung Fahrerassistenz, München, 15(16):2012, 2012.
- [45] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. Human factors, 39(2):230–253, 1997.
- [46] Luiz Eduardo G Martins and Tony Gorschek. Requirements engineering for safety-critical systems: An interview study with industry practitioners. IEEE Transactions on Software Engineering, 46(4):346–361, 2018.

- [47] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie El-hadad. Intelligent models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pages 1721–1730, 2015.
- [48] Kush R Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. Big data, 5(3):246–255, 2017.
- [49] Jim McPherson. How uber’ s self-driving technology could have failed in the fatal tempe crash. Forbes, March, 2018.
- [50] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In https://cn.overleaf.com/project/62cc3435682c12c8df6e5f31ernational conference on computer aided verification, pages 97–117. Springer, 2017.
- [51] Philip Koopman. Challenges in autonomous vehicle validation: Keynote presentation abstract. In Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles, pages 3–3, 2017.
- [52] Zeshan Kurd, Tim Kelly, and Jim Austin. Developing artificial neural networks for safety critical systems. Neural Computing and Applications, 16(1):11–19, 2007.
- [53] M Harris. Google reports self-driving car mistakes: 272 failures and 13 near misses. The Guardian, 2016.
- [54] Aleksandar Chakarov, Aditya Nori, Sriram Rajamani, Shayak Sen, and Deepak Vijaykeerthy. Debugging machine learning tasks. arXiv preprint arXiv:1603.07292, 2016.
- [55] Besmira Nushi, Ece Kamar, Eric Horvitz, and Donald Kossmann. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- [56] Itsuo Takanami, Masaru Sato, and Yun Ping Yang. A fault-value injection approach for multiple-weight-fault tolerance of mnns. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, volume 3, pages 515–520. IEEE, 2000.

- [57] J Janai, F Güney, A Behl, and A Geiger. Computer vision for autonomous vehicles: problems, datasets and state of the art. arxiv e-prints. [arXiv preprint arXiv:1704.05519](https://arxiv.org/abs/1704.05519), 2017.
- [58] Hang Yin and Christian Berger. When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets. In [2017 IEEE 20th International Conference on Intelligent Transportation Systems \(ITSC\)](#), pages 1–8. IEEE, 2017.
- [59] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. [arXiv preprint arXiv:1805.04687](#), 2(5):6, 2018.
- [60] Lex Fridman, Daniel E Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik, Jack Terwilliger, Julia Kindelsberger, Li Ding, Sean Seaman, et al. Mit autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation. [arXiv preprint arXiv:1711.06976](#), 1, 2017.
- [61] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. [The International Journal of Robotics Research](#), 32(11):1231–1237, 2013.
- [62] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In [Proceedings of the IEEE/CVF conference on computer vision and pattern recognition](#), pages 11621–11631, 2020.
- [63] P Koschorrek, T Piccini, P Öberg, M Felsberg, L Nielsen, and R Mester. A multi-sensor traffic scene dataset with omnidirectional video. ground truth—what is a good dataset. In [Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops \(CVPRW\)](#), Portland, OR, USA, pages 23–28, 2013.
- [64] Gaurav Pandey, James R McBride, and Ryan M Eustice. Ford campus vision and lidar data set. [The International Journal of Robotics Research](#), 30(13):1543–1552, 2011.
- [65] Udacity. Udacity data collection. <http://academictorrents.com/collection/self-driving-cars>. 2018.
- [66] Cityscapes. Cityscapes data collection. www.cityscapes-dataset.com/. 2018.

- [67] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research, 36(1):3–15, 2017.
- [68] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2):88–97, 2009.
- [69] F Flohr and D Gavrilu. Daimler pedestrian segmentation benchmark dataset. In Proc. of the British Machine Vision Conference, 2013.
- [70] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In 2009 IEEE conference on computer vision and pattern recognition, pages 304–311. IEEE, 2009.
- [71] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.
- [72] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
- [73] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.
- [74] Joseph Redmon and Ali Farhadi. Yolo3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
- [75] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.
- [76] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European conference on computer vision (ECCV), pages 734–750, 2018.
- [77] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4203–4212, 2018.

- [78] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.
- [79] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 2015.
- [80] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29, 2016.
- [81] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016.
- [82] Jianjun Li, Kangjian Peng, and Chin-Chen Chang. An efficient object detection algorithm based on compressed networks. Symmetry, 10(7):235, 2018.
- [83] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 652–660, 2017.
- [84] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4490–4499, 2018.
- [85] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum point-nets for 3d object detection from rgb-d data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 918–927, 2018.
- [86] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1907–1915, 2017.
- [87] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka. Roarnet: A robust 3d object detection based on region approximation refinement. In 2019 IEEE intelligent vehicles symposium (IV), pages 2510–2515. IEEE, 2019.
- [88] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In 2018 IEEE/RSJ

- International Conference on Intelligent Robots and Systems (IROS), pages 1–8. IEEE, 2018.
- [89] Edouard Leurent. A survey of state-action representations for autonomous driving. 2018.
 - [90] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12):2481–2495, 2017.
 - [91] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnets for real-time semantic segmentation on high-resolution images. In Proceedings of the European conference on computer vision (ECCV), pages 405–420, 2018.
 - [92] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147, 2016.
 - [93] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 4644–4651. IEEE, 2017.
 - [94] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.
 - [95] Michael Trembl, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, et al. Speeding up semantic segmentation for autonomous driving. 2016.
 - [96] Dan Barnes, Will Maddern, Geoffrey Pascoe, and Ingmar Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1894–1900. IEEE, 2018.
 - [97] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision, pages 2938–2946, 2015.
 - [98] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. IEEE Robotics and Automation Letters, 3(4):4407–4414, 2018.

- [99] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision, pages 627–637, 2017.
- [100] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In Proceedings of the IEEE international conference on computer vision workshops, pages 879–886, 2017.
- [101] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 929–938, 2017.
- [102] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4654–4662, 2018.
- [103] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In Conference on Robot Learning, pages 456–465. PMLR, 2018.
- [104] Ioan Andrei Barsan, Shenlong Wang, Andrei Pokrovsky, and Raquel Urtasun. Learning to localize using a lidar intensity map. arXiv preprint arXiv:2012.10902, 2020.
- [105] Olivier Garcia-Favrot and Michel Parent. Laser scanner based slam in real road and traffic environment. In IEEE International Conference Robotics and Automation (ICRA09). Workshop on Safe navigation in open and dynamic environments Application to autonomous vehicles, 2009.
- [106] Arash K Ushani and Ryan M Eustice. Feature learning for scene flow estimation from lidar. In Conference on Robot Learning, pages 283–292. PMLR, 2018.
- [107] Franziska Meier, Philipp Hennig, and Stefan Schaal. Efficient bayesian local model learning for control. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2244–2249. IEEE, 2014.
- [108] Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Robust constrained learning-based nmpe enabling reliable mobile robot path tracking. The International Journal of Robotics Research, 35(13):1547–1563, 2016.

- [109] Olivier Sigaud, Camille Salaün, and Vincent Padois. On-line regression algorithms for learning mechanical models of robots: a survey. Robotics and Autonomous Systems, 59(12):1115–1129, 2011.
- [110] Eike Rehder, Jannik Quehl, and Christoph Stiller. Driving like a human: Imitation learning for path planning using convolutional neural networks. In International Conference on Robotics and Automation Workshops, pages 1–5, 2017.
- [111] Liting Sun, Cheng Peng, Wei Zhan, and Masayoshi Tomizuka. A fast integrated planning and control framework for autonomous driving via imitation learning. In Dynamic Systems and Control Conference, volume 51913, page V003T37A012. American Society of Mechanical Engineers, 2018.
- [112] Sorin Mihai Grigorescu, Bogdan Trasnea, Liviu Marina, Andrei Vasilcoi, and Tiberiu Cocias. Neurotrajectory: A neuroevolutionary approach to local state trajectory learning for autonomous vehicles. IEEE Robotics and Automation Letters, 4(4):3441–3448, 2019.
- [113] Lingli Yu, Xuanya Shao, Yadong Wei, and Kaijun Zhou. Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning. Sensors, 18(9):2905, 2018.
- [114] Chris Paxton, Vasumathi Raman, Gregory D Hager, and Marin Kobilarov. Combining neural networks and tree search for task and motion planning in challenging environments. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6059–6066. IEEE, 2017.
- [115] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. Annual Review of Control, Robotics, and Autonomous Systems, 1(1):187–210, 2018.
- [116] Tianyu Gu, John M Dolan, and Jin-Woo Lee. Human-like planning of swerve maneuvers for autonomous vehicles. In 2016 IEEE Intelligent Vehicles Symposium (IV), pages 716–721. IEEE, 2016.
- [117] Markus Wulfmeier, Dominic Zeng Wang, and Ingmar Posner. Watch this: Scalable cost-function learning for path planning in urban environments. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2089–2095. IEEE, 2016.

- [118] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295, 2016.
- [119] Aleksandr I Panov, Konstantin S Yakovlev, and Roman Suvorov. Grid path planning with deep reinforcement learning: Preliminary results. Procedia computer science, 123:347–353, 2018.
- [120] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps. In 17th international IEEE conference on intelligent transportation systems (ITSC), pages 392–399. IEEE, 2014.
- [121] Jürgen Wiest, Matthias Höffken, Ulrich Kreßel, and Klaus Dietmayer. Probabilistic trajectory prediction with gaussian mixture models. In 2012 IEEE Intelligent Vehicles Symposium, pages 141–146. IEEE, 2012.
- [122] Yangliu Dou, Fengjun Yan, and Daiwei Feng. Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In 2016 IEEE international conference on advanced intelligent mechatronics (AIM), pages 901–906. IEEE, 2016.
- [123] John H Reif. Complexity of the mover’s problem and generalizations. In 20th Annual Symposium on Foundations of Computer Science (sfcs 1979), pages 421–427. IEEE Computer Society, 1979.
- [124] Ferenc Hegedüs, Tamás Bécsi, Szilárd Aradi, and György Gáldi. Hybrid trajectory planning for autonomous vehicles using neural networks. In 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), pages 000025–000030. IEEE, 2018.
- [125] Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. Driving in dense traffic with model-free reinforcement learning. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 5385–5392. IEEE, 2020.
- [126] Árpád Fehér, Szilárd Aradi, Ferenc Hegedüs, Tamás Bécsi, and Péter Gáspár. Hybrid ddpg approach for vehicle motion planning. 2019.
- [127] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

- [128] Chris J Ostafew, Angela P Schoellig, Timothy D Barfoot, and Jack Collier. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. Journal of Field Robotics, 33(1):133–152, 2016.
- [129] Duy Nguyen-Tuong, Jan Peters, and Matthias Seeger. Local gaussian process regression for real time online model learning. Advances in neural information processing systems, 21, 2008.
- [130] Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Visual teach and repeat, repeat, repeat: Iterative learning control to improve mobile robot path tracking in challenging outdoor environments. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 176–181. IEEE, 2013.
- [131] Benjamas Panomruttanarug. Application of iterative learning control in tracking a dubin’s path in parallel parking. International Journal of Automotive Technology, 18(6):1099–1107, 2017.
- [132] Nitin R Kapania and J Christian Gerdes. Path tracking of highly dynamic autonomous vehicle trajectories via iterative learning control. In 2015 American control conference (ACC), pages 2753–2758. IEEE, 2015.
- [133] Yang Zhao, Fengyu Zhou, Yan Li, and Yugang Wang. A novel iterative learning path-tracking control for nonholonomic mobile robots against initial shifts. International journal of advanced robotic systems, 14(3):1729881417710634, 2017.
- [134] Stéphanie Lefevre, Ashwin Carvalho, and Francesco Borrelli. A learning-based framework for velocity control in autonomous driving. IEEE Transactions on Automation Science and Engineering, 13(1):32–42, 2015.
- [135] Stéphanie Lefevre, Ashwin Carvalho, and Francesco Borrelli. Autonomous car following: A learning-based approach. In 2015 IEEE intelligent vehicles symposium (IV), pages 920–926. IEEE, 2015.
- [136] Paul Drews, Grady Williams, Brian Goldfain, Evangelos A Theodorou, and James M Rehg. Aggressive deep driving: Combining convolutional neural networks and model predictive control. In Conference on Robot Learning, pages 133–142. PMLR, 2017.
- [137] Ugo Rosolia, Ashwin Carvalho, and Francesco Borrelli. Autonomous racing using learning model predictive control. In 2017 American Control Conference (ACC), pages 5115–5120. IEEE, 2017.

- [138] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Learning deep neural network control policies for agile off-road autonomous driving. In The NIPS Deep Reinforcement Learning Symposium, 2017.
- [139] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile autonomous driving using end-to-end deep imitation learning. arXiv preprint arXiv:1709.07174, 2017.
- [140] B Rawlings JBJ and DQ Mayne. Model predictive control theory and design, 1999.
- [141] Mohamed A Kamel, Ahmed T Hafez, and Xiang Yu. A review on motion control of unmanned ground and aerial vehicles based on model predictive control techniques. Journal of Engineering Science and Military Technologies, 2(1):10–23, 2018.
- [142] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. Advances in neural information processing systems, 1, 1988.
- [143] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann Cun. Off-road obstacle avoidance through end-to-end learning. Advances in neural information processing systems, 18, 2005.
- [144] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.
- [145] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2174–2182, 2017.
- [146] Hesham M Eraqi, Mohamed N Moustafa, and Jens Honer. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. arXiv preprint arXiv:1710.03804, 2017.
- [147] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-end learning of driving models with surround-view cameras and route planners. In Proceedings of the european conference on computer vision (eccv), pages 435–453, 2018.
- [148] Viktor Rausch, Andreas Hansen, Eugen Solowjow, Chang Liu, Edwin Kreuzer, and J Karl Hedrick. Learning a deep neural net policy for end-to-end control of autonomous vehicles. In 2017 American Control Conference (ACC), pages 4914–4919. IEEE, 2017.

- [149] Michael G Bechtel, Elise McEllhiney, Minje Kim, and Heechul Yun. Deeppicar: A low-cost deep neural network-based autonomous car. In 2018 IEEE 24th international conference on embedded and real-time computing systems and applications (RTCSA), pages 11–21. IEEE, 2018.
- [150] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In Proceedings of the IEEE international conference on computer vision, pages 2722–2730, 2015.
- [151] Shun Yang, Wenshuo Wang, Chang Liu, Weiwen Deng, and J Karl Hedrick. Feature analysis and selection for training an end-to-end autonomous vehicle controller using deep learning approach. In 2017 IEEE Intelligent Vehicles Symposium (IV), pages 1033–1038. IEEE, 2017.
- [152] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19):70–76, 2017.
- [153] Etienne Perot, Maximilian Jaritz, Marin Toromanoff, and Raoul De Charette. End-to-end driving in a realistic racing game with deep reinforcement learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 3–4, 2017.
- [154] Maximilian Jaritz, Raoul De Charette, Marin Toromanoff, Etienne Perot, and Fawzi Nashashibi. End-to-end race driving with deep reinforcement learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 2070–2075. IEEE, 2018.
- [155] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In 2019 International Conference on Robotics and Automation (ICRA), pages 8248–8254. IEEE, 2019.
- [156] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI conference on artificial intelligence, volume 30, 2016.
- [157] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In International conference on machine learning, pages 1995–2003. PMLR, 2016.

- [158] Tianhao Zhang, Gregory Kahn, Sergey Levine, and Pieter Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In 2016 IEEE international conference on robotics and automation (ICRA), pages 528–535. IEEE, 2016.
- [159] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11):3212–3232, 2019.
- [160] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8445–8453, 2019.
- [161] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H Ang Jr. Perception, planning, control, and coordination for autonomous vehicles. Machines, 5(1):6, 2017.
- [162] Eriko Nurvitadhi, Ganesh Venkatesh, Jaewoong Sim, Debbie Marr, Randy Huang, Jason Ong Gee Hock, Yeong Tat Liew, Krishnan Srivatsan, Duncan Moss, Suchit Subhaschandra, et al. Can fpgas beat gpus in accelerating next-generation deep neural networks? In Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays, pages 5–14, 2017.