

# 中山大学硕士学位论文

针对自主泊车问题的一种基于深度学习的运动规划方  
法

Deep Learning-based Motion Planning Method for  
Autonomous Parking

学 位 申 请 人	李樊
指 导 老 师	黄凯 教授
专 业 名 称	工程(软件工程)

答辩委员会主席(签名): \_\_\_\_\_

答辩委员会委员(签名): \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

二〇二〇年五月二十六日



## 论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名: \_\_\_\_\_

日期:       年   月   日

## 学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版；有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅；有权将学位论文的内容编入有关数据库进行检索；可以采用复印、缩印或其他方法保存学位论文；可以为建立了馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

保密论文保密期满后，适用本声明。

学位论文作者签名:                      导师签名:

日期:    年   月   日    日期:    年   月   日



论文题目：针对自主泊车问题的一种基于深度学习的运动规划方法

专业：工程(软件工程)

硕士生：李樊

指导老师：黄凯教授

## 摘要

近年来，自动驾驶领域得到越来越多的关注，要实现高自主性的自动驾驶，研究针对自主泊车的高适应性运动规划算法十分重要。对于现有的泊车运动规划方法，启发式的设计非常关键，但由于复杂多变的泊车场景间缺乏普遍和共同的特征，因此为规划方法手动设计通用的启发式相当困难。随着深度学习研究的发展，许多基于深度学习的算法开始受到关注，目前的研究大都基于全卷积神经网络从驾驶环境中提取可行区域作为启发式，以提高规划算法的场景适应性。但这种方式仍存在启发采样效率不高以及规划首条可行路径速度不够快的问题。因此，本文提出了一种基于深度学习的二阶段泊车运动规划方法，在第一阶段，利用深度神经网络为自动驾驶车辆实时规划初始泊车路径，充分利用神经网络并行运算快和场景适应性强的特点，提高规划首条可行路径的速度和稳定性。在第二阶段，考虑到神经网络存在泛化误差，利用基于采样的规划算法优化神经网络的预测以确保方法输出无碰撞和高质量的泊车路径。同时在优化过程中，利用神经网络规划的初始路径指导采样规划算法在最优采样点附近采样，以提升算法的采样效率和路径优化能力。此外，在方法中，还对神经网络的泛化误差进行最佳高斯拟合，利用拟合分布指导采样优化算法的偏置采样过程，进一步提高算法优化初始路径的能力。实验表明，与主流的两种先进的启发式方法相比，本文提出的二阶段泊车规划方法具有更优异的规划性能，在满足平均帧率 10 FPS 的实时性要求的同时规划成功率可达到 99 % 以上。

**关键词：**自主泊车，运动规划，路径规划，深度学习。



Title: Deep Learning-based Motion Planning Method for Autonomous Parking  
Major: Engineering (Software Engineering)  
Name: Fan Li  
Supervisor: Prof. Kai Huang

## ABSTRACT

Recently, more attention has been paid to autonomous driving. For autonomous driving, motion planning algorithms for autonomous parking is necessary. Using good heuristics is the key to successful motion planning. However, since the parking scenarios lack significant features in common, creating useful heuristics manually for planning methods is a non-trivial task. As the boom of deep learning, many learning-based motion planning methods accrue. Most of them focus on using Fully-convolutional Neural Networks to extract feasible regions for guiding the sampling of sampling-based methods, but such region-based heuristics is not enough efficient. Therefore, in this paper, a two-stage scheme is proposed. In the first stage, it makes use of a deep convolutional neural network to plan an initial parking path, in order to speed up planning for the first feasible path by taking advantage of the parallel computation and the stability over various scenes, of the neural network. In the second stage, since the neural network has the generalization error, it makes use of a sampling-based algorithm to perform optimization on the initial path to make sure that the output path is collision-free and quality. Moreover, in the proposed scheme, the generalization error is modeled as a Gaussian model to guide the sampling of the sampling-based algorithm and improve its ability to optimize the initial path. By experiments, comparing with the other two state-of-the-art heuristic planning methods, the proposed two-stage scheme is outperformed, and it can meet 10 FPS real-time requirement with over 99 % successful rate.

**Keywords:** Autonomous Parking, Motion Planning, Path Planning, Deep Learning.



# 目 录

第 1 章 绪论 ······	1
1.1 研究背景及意义 ······	1
1.2 研究现状及相关工作 ······	2
1.3 本文工作及创新点 ······	5
1.4 论文结构 ······	6
第 2 章 相关理论和技术原理 ······	9
2.1 运动规划的几何表示 ······	9
2.2 基于采样的运动规划算法 ······	12
2.3 工作空间表示和积分曲线求解 ······	18
2.4 基于深度神经网络的目标检测 ······	20
2.5 本章总结 ······	23
第 3 章 二阶段自主泊车规划系统 ······	25
3.1 系统框架 ······	25
3.2 第一阶段：YIPS 路径规划网络 ······	26
3.3 第二阶段：SO-RRT* 路径优化算法 ······	35
3.4 泊车规划数据集制作 ······	40
3.5 本章总结 ······	45
第 4 章 神经网络评估及验证 ······	47
4.1 实验平台参数 ······	47
4.2 网络预测性能评估 ······	47
4.3 网络模型设计验证 ······	56
4.4 本章总结 ······	67
第 5 章 泊车规划效果验证与评估 ······	69
5.1 实验平台参数 ······	69
5.2 泊车运动规划方法对比 ······	69
5.3 YIPS 网络初始路径规划评估 ······	72
5.4 SO-RRT* 算法初始路径优化验证 ······	76

5.5 本章总结 .....	84
第 6 章 总结与展望 .....	85
6.1 总结 .....	85
6.2 展望 .....	86
参考文献 .....	87
攻读硕士学位期间科研成果 .....	91
致 谢 .....	92

# 第1章      绪论

## 1.1 研究背景及意义

近年来，随着人工智能和机器人技术的迅速发展，自动驾驶领域得到越来越多的关注。自动驾驶汽车的功能性架构可以划分为四个模块 [1][2]：感知、决策、规划和控制。规划模块主要涉及运动规划(又称路径规划)，在自动驾驶系统中起着重要作用 [3][4][5]。概括来说，运动规划是指，在遵守控制物体内部约束(如车辆运动学约束)和外部约束(避免与障碍物碰撞)的条件下，规划出一条从初始状态到目标状态的可行轨迹。运动规划问题已经被证明是NP-完全问题 [6][7][8]，因此，实用的运动规划算法更多关注在提高计算次优解的速度和成功率，而非最优解。对自动驾驶的运动规划也同样如此，而且由于更严格的实时性要求，其对规划算法的规划速度更为看重。

就无人驾驶车辆而言，驾驶场景可以大致分为两类：结构化和非结构化。泊车场景是最主要也是最典型的非结构化场景，要实现高自主性的自动驾驶，研究针对自主泊车的高场景适应性的运动规划算法将十分重要。在结构化场景中，由于道路形态是人为规范且有规律的，因此可以借助道路中线或其他车道标记来有效的启发运动规划算法，从而实现实时高成功率的规划。但是，在非结构化路面上缺少类似的道路标识的指引，算法往往只有在简单规整的泊车场景中才能达到实时规划的要求。此外，与其他非结构化场景下的驾驶行为相比，泊车的运动规划需要综合考虑车辆向前和向后两种运动可能，搜索空间增加了一倍，在一些比较拥挤的泊车环境，由于可行的运动路径稀少，很容易导致规划无法在限制时间内完成而失效。另外，不同的泊车场景的相似性不明显，如垂向入库和侧方停车的路径相似性较低，更不要说一些非规则的乡间土路泊车场景。因此设计并实现场景适应性强同时规划速度快的自主泊车规划算法十分具有挑战性，实时性泊车运动规划算法的研究即是自动驾驶研究的重点也是研究难点之一。

## 1.2 研究现状及相关工作

为了在复杂多变的泊车场景下实现高效的运动规划，一个通用且合理的选择是利用高效的启发式去指导规划算法的规划过程，从而提升规划速度并降低失败概率。人工设计基于场景特征的启发式是十分普遍和实用的，因为在简单和结构化的路面上具有比较好的效果，文献 [9] 基于预定义的规则，根据不同场景调整高斯采样的参数，提高了 CL-RRT 算法在城市道路场景下的规划效率。文献 [10] 利用潜在人工场来启发 RRT\*[11] 方法的采样过程，一定程度提高了收敛速度。虽然前述方法证明了人工设计的启发式能提高规划效果，但是在缺乏明显共通特征的非结构化场景中进行规划时，依赖于共有或给定环境特征的启发式方法往往规划效果不佳，同时，此类算法通常依赖于算法参数对场景的针对性调整，但要自动识别不同场景的特征并自适应调整算法参数，其难度不亚于设计运动规划算法本身。

文献 [12][13] 提出并完善了朝向感知空间探索 [12](Orientation-aware Space Exploration, 简称 OSE) 启发方法，基于 A\*[14]，利用圆形晶格对无碰撞的空间进行探索，在探索中有效地考虑了车辆的朝向信息，使得启发式对车辆的运动规划具有了更好的启发性。但是圆形晶格的尺寸参数设计需要针对具有不同特征的场景进行定制，过分依赖于专家的经验，很难在不同场景下维持稳定的规划速度和质量。

鉴于人工设计启发式存在的问题，近年来，随着深度学习研究的发展，许多基于深度学习的算法开始受到关注。受益于深度学习模型成熟的并行计算框架，神经网络的运算速度基本可以满足实时性要求。文献 [15] 利用基于全卷积神经网络 [16](Fully Convolutional Networks, 简称 FCNs) 设计的网络模型，以激光雷达传感器数据为输入，为结构化道路上的自动驾驶车辆预测可行路径，其预测帧率可达 30 FPS。同时深度学习方法也可以应用到不同的控制对象上，文献 [17] 利用深度卷积神经网络，以 2D 激光雷达传感器数据为输入，为室内机器人直接输出当前的运动控制指令，实现了端到端的控制效果。文献 [18][19] 利用 3D 卷积神经网络，以占用栅格图为输入，为 CL-RRT 算法的状态采样提供启发，提高了 CL-RRT 算法的场景适应性。

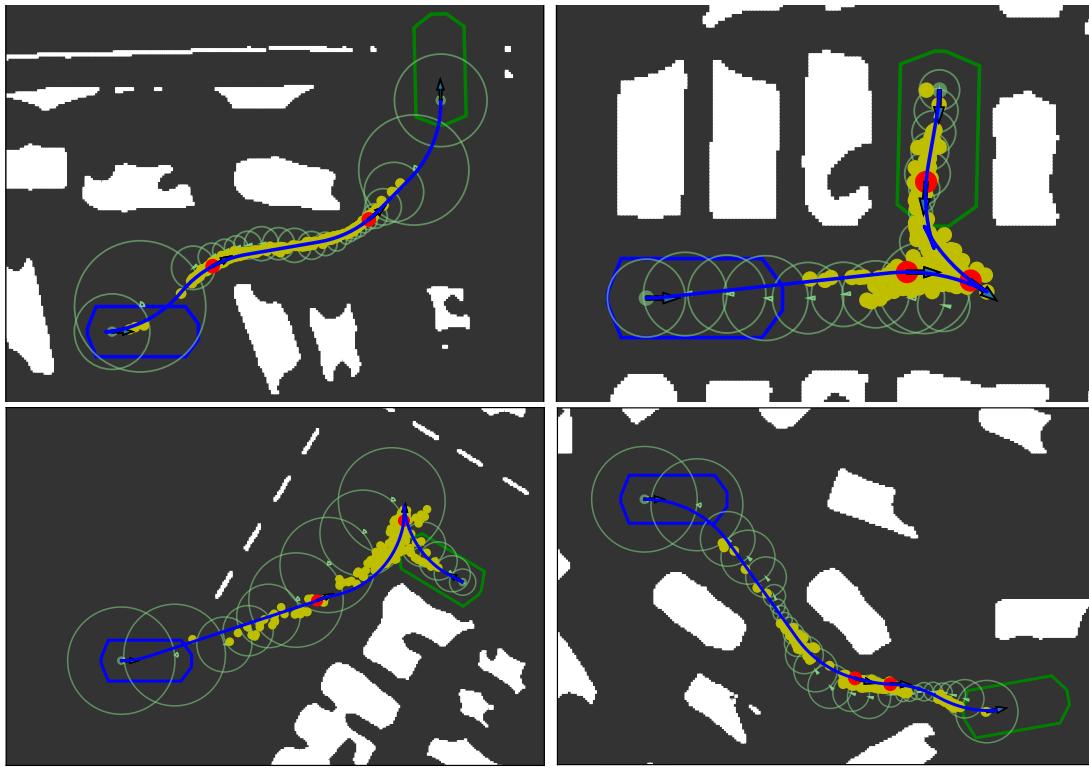


图 1-1 OSE 启发的 Bi-RRT\* 算法在四种不同的规划场景中的路径规划结果，对每个场景，Bi-RRT\* 运行 100 次，每次运行采样 1000 次后输出规划结果。其中蓝色、绿色多边形分别代表起始和目标配置下的车辆姿态；绿色圆圈代表 OSE 算法的探索结果；蓝色曲线表示 Bi-RRT\*100 次运行中获取的最优路径，红色圆点表示构造该最优路径的采样点，黄色圆点为构造其余 99 个次优路径的采样点；白色色块表示车辆周围的障碍物。

因为深度学习方法在图像领域的成熟，很多基于深度学习的规划方法都将输入处理成图片的形式，以使网络输入更加规整，更有利于网络训练，同时也便于利用成熟的神经网络结构如 FCNs 和 ResNet[20] 等进行语义分割和特征提取，提高方法的鲁棒性，也使方法更具有可解释性。文献 [21] 和文献 [22] 都基于 FCNs 设计神经网络，将占用栅格图以及起始和目标配置编码成图片后输入网络，通过神经网络从驾驶环境中分割出可行区域，进而利用可行区域指导 Bi-RRT\*[23] 算法的采样过程，实现了更稳定和快速的路径规划，不同的是前者应用于室内机器人的适应性避障，后者则用于自动驾驶车辆的自主泊车。文献 [24] 以已有的特征提取网络为主干设计神经网络，以激光雷达和路网图为输入，为结构化道路上的自动驾驶车辆输出用于路径选择算法的启发区域，并同时输出激光点云的语义分割结果，说明了神经网络在多输出任务上的应用潜力。

深度学习方法在自主泊车运动规划问题上的应用研究还较少，大部分集中在基于 FCNs 网络从驾驶环境中分割出可行区域这一范式上，这与文献 [13] 提出的 OSE 启发方法的思路类似，不同在于，OSE 方法基于 A\* 实现可行区域的提取而非神经网络。这种范式存在两个主要缺陷。首先，通过包含可行解的区域来启发基于采样的规划算法，虽然可以提高方法的场景适应性，但是在区域中求解可行路径完全由基于采样的规划算法来完成，而由于基于采样的算法具有随机性且其规划速度受启发区域的形态影响较大，因此方法规划首条可行路径的速度不稳定且往往不如启发的生成速度快，不能充分发挥神经网络或 OSE 方法速度快和场景适应性强的优势，同时，由于基于采样的算法需要一定的采样时间来获取首条可行路径，因此可用于优化路径的采样时间相应减少，无法对路径进行充分优化。其次，可行区域的启发效率还有提升空间，图1-1展示了利用 OSE 启发 Bi-RRT\* 的规划方法在四个典型场景中的结果，由图可见，要获取每个场景下的“最优”(近似最优，非理论上最优)规划路径(即图1-1中蓝色曲线)，实际上只需要获取最优的采样点(即图1-1中的红色点)，最优采样点和最优路径是等价的。而所有有利于求解可行路径的采样点包括“最优”采样点和次优采样点(即图1-1中的黄色点)并非在可行区域中均匀分布，相反，这些采样点大都聚集在“最优”采样点附近，呈现聚集性分布。因此，直接从可行区域中均匀的采样会造成很大程度的浪费，很多采样并不有益于规划结果更靠近最优解，也由此，利用类似 OSE 启发式的可行区域启发范式，在采样效率上还有很大的改进空间。

自动驾驶的仿真技术对自动驾驶运动规划的研究而言也是必不可少的，仿真环境可以加速算法的验证，同时，基于深度学习的方法也需要利用仿真环境快速定制丰富的场景，以生成训练神经网络所需的数据集。CARLA[25] 是一款较为成熟的自动驾驶模拟器，其基于虚幻游戏引擎搭建，在基于视觉的自动驾驶强化学习领域已有应用 [26]。其系统较好的仿真了车辆动力学特性以及激光雷达、超声波雷达和摄像头等多款传感器，同时场景也较为丰富。Airsim[27] 是另一款较为成功的自动驾驶模拟器，同样基于虚幻游戏引擎搭建，但在场景丰富度上较 CARLA 有所欠缺。这些自动驾驶模拟器都提供了软件接口，以配置车辆动力学特性、传感器以及控制场景中人车等障碍物的分布和运动，这使得根据不同需求设计不同场景成为可能，也为本论文的研究提供了基础。

### 1.3 本文工作及创新点

综上所述，在针对自主泊车的运动规划算法中，使用人工设计的启发式虽然可以在结构化场景下实现高效规划，但在复杂多变的非结构化环境中难以实现稳定快速的规划。因此应用深度学习方法来提高规划算法对场景的适应性和规划速度十分具有意义。当前在自主泊车运动规划问题上应用深度学习方法的主要思路是基于 FCNs 网络从驾驶环境中分割出可行区域，再利用可行区域启发 Bi-RRT\* 算法的采样。根据上述分析，这种范式存在求解首条可行路径速度不够快且场景适应性不强，求解速度不稳定的问题，同时，其启发采样效率有待提高，因为有益于规划成功和优化规划结果的采样区域并非均匀分布在可行区域中，而是聚集在特定区域，即“最优”采样点附近。

针对上述问题，本文提出了一种基于深度学习的二阶段泊车规划方法，在方法的第一阶段，利用深度神经网络为泊车车辆规划初始泊车路径，利用神经网络并行运算快和场景适应性强的优势，提高规划首条可行路径的速度和稳定性。在第二阶段，考虑到神经网络存在泛化误差，利用基于采样的规划算法优化神经网络的预测以确保方法输出无碰撞和高质量的泊车路径。同时在优化过程中，利用神经网络规划的初始路径指导采样规划算法在“最优”采样点附近采样，以提高算法采样效率，进而提升算法的路径优化能力。此外，在方法中，还采用了高斯分布对神经网络的泛化误差进行建模，以指导采样规划算法的偏置采样过程，进一步提高算法优化初始路径的能力。为了设计实现并对比验证本文提出的泊车规划算法，本文具体做了以下工作：

- 1 利用 CARLA 自动驾驶模拟器，设计制作了细节丰富的泊车场景，并用两种采样规划算法为泊车场景生成标签路径，最后形成用于神经网络训练和泊车规划方法验证的由 10962 个数据样本构成的泊车规划场景数据集。
- 2 设计用于方法第一阶段的初始路径规划神经网络，并对网络的规划性能进行实验评估。同时，探究了两种不同的损失函数、5 种不同类型的主干网络和 7 种不同的输入编码方式对网络学习能力的影响，并验证了用 Fine-tuning 方法进行网络训练的有效性。
- 3 设计用于方法第二阶段的初始路径优化算法并对方法的优化性能进行了

实验评估。同时，评估了两种不同的父顶点搜索方式对算法路径优化性能的影响。

- 4 对神经网络的泛化误差进行了分析并建立了误差的最佳拟合高斯模型，然后将拟合模型与3种不同的高斯模型进行了对比，验证了利用拟合模型指导路径优化算法的偏置采样过程的有效性和必要性。
- 5 将本文提出的二阶段泊车规划方法与两种基于先进启发式的规划方法进行对比，对方法的有效性和先进性进行了实验验证。同时，将目标检测神经网络和本文提出的路径规划网络进行了主干网络融合，验证了将二阶段规划方法所针对的泊车规划任务与目标检测任务进行任务融合的可行性。

其中，本文的创新点在于：

- 1 提出了一种基于深度学习的针对自主泊车场景的运动规划算法，该算法在满足平均帧率10 FPS 的实时性要求的同时规划成功率可以达到99%以上。同时，算法在规划成功率、规划首条路径的速度以及规划路径的质量三个方面都优于用于对比的两种启发式规划算法。
- 2 设计了一个轻量级和端到端的路径规划神经网络，该网络最快可以50 FPS 的帧率为泊车车辆规划初始泊车路径，且95%以上的初始路径碰撞面积少于车辆占用区域面积的一半。同时该网络的输出可用于启发基于采样的运动规划算法，并能实现比基于可行区域的启发式更好的启发效果。
- 3 设计了一种基于采样的路径优化算法，并利用神经网络泛化误差的最佳拟合高斯模型指导优化算法的采样过程，提高了算法的优化性能，使得算法在0.1s的优化时间内就可将神经网络规划的初始路径的可行率提高到90%以上，且优化后路径平均长度与最优值的差距少于1 m。

## 1.4 论文结构

本论文剩余部分将按如下组织：

在第二章中，对本文工作涉及的与运动规划算法和深度神经网络设计相关的概念和技术进行概括阐述。

在第三章中，首先对本文提出的路径规划神经网络的输入输出形式、网络架构、训练方法和损失函数设计等进行了详细介绍，然后介绍了本文提出的路径优

化算法的设计以及对神经网络预测误差进行最佳拟合高斯建模的方法。最后阐述了泊车规划场景数据集的制作过程。

在第四章中，首先对本文提出的路径规划神经网络的预测性能进行了验证和评估。然后对神经网络的模型设计进行验证，包括网络输入编码方式设计、损失函数设计、训练方法设计、主干网络选择以及直接任务融合可行性等。

在第五章中，首先将本文提出的二阶段泊车规划方法与两种基于先进启发式的泊车规划算法进行了对比以验证方法的先进性。然后对本文提出的路径规划神经网络的初始路径规划性能进行了评估，最后对本文提出的路径优化算法的优化性能进行验证，同时还验证了用神经网络泛化误差的最佳高斯拟合模型指导其优化采样过程的必要性和有效性。

在第六章中，对本文工作进行了总结，并对本文工作中尚存在的不足以及值得进一步研究的问题进行分析与展望。



## 第2章 相关理论和技术原理

在本章中，对与本文工作相关的理论和技术原理进行了介绍，包括利用几何代数方法规范化自主泊车运动规划问题、适用于泊车规划的基于采样的运动规划算法及先进启发式、运动规划的重要子问题：工作空间表示和积分曲线求解、以及基于深度卷积神经的目标检测算法。

### 2.1 运动规划的几何表示

#### 2.1.1 汽车的配置空间

假定一个系统的状态可以由  $n$  个独立变量完全定义，那么系统所有可能的状态将形成一个  $n$  维的空间，这个空间被称为配置空间 [28]，一般用符号  $C$  表示。系统实际的工作环境称为工作空间  $\mathcal{W}$ ，工作空间包含了系统运行中可能遇到的各种对象如工具、障碍物等。 $\mathcal{W}$  体现了在系统的实际工作过程中存在的各种外部约束（如不能与其他对象碰撞）。理论上  $\mathcal{W}$  中的障碍物都可以通过映射  $f : \mathcal{W} \rightarrow C$  在配置空间  $C$  中显示表示。所有通过映射  $f$  与障碍物对应的配置构成的子空间，称为障碍配置空间，用符号  $C_{obs}$  表示，其对应的补集称为自由配置空间  $C_{free}$ 。因为障碍物的存在只会约束系统配置空间中与位置和姿态相关的维度，所以  $C_{obs}$  空间和其补集  $C_{free}$  的往往是配置空间  $C$  的低维子空间。同时需要指出的是，映射  $f$  虽然理论上存在，但是在实践中要实现障碍物在配置空间的显式表示是相当困难的，尤其是对于在障碍物较多的工作空间中的高维系统，往往计算复杂度非常大。

$$\theta = \arcsin(l/r) \quad (2-1)$$

$$\kappa = 1/r \quad (2-2)$$

对于自动驾驶汽车（特指四轮汽车），确定其配置空间需要考虑其运动学模型，而全面完整地考虑一辆四轮汽车的运动学模型是十分复杂的，为了方便研究，一般将车辆简化成如图2-1所示的二轮自行车模型。基于图示的车辆运动学模型，直观上，车辆的配置可由  $(x, y, \phi, v, \theta)$  等 5 个变量定义， $(x, y)$  表示汽车后

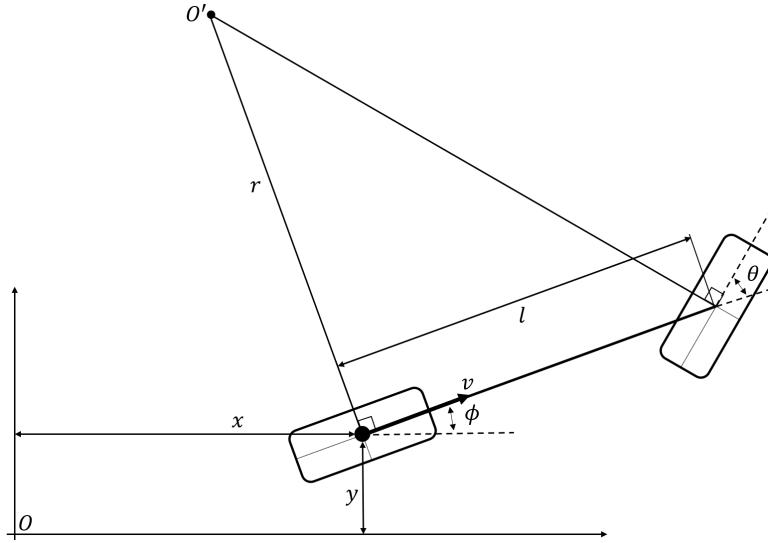


图 2-1 汽车的二轮自行车运动学模型。 $(x, y)$  表示汽车后轴中心的坐标;  $v$  表示后轴中心的标量速度;  $\phi$  代表汽车的朝向角;  $\theta$  代表汽车前轮的转向角, 而汽车的转弯半径由  $r$  表示;  $l$  代表汽车的轴距;  $O$  和  $O'$  分别表示坐标原点和转向中心点

轴中心的坐标,  $v$  表示后轴中心的速度, 为标量,  $\phi$  代表汽车的朝向角,  $\theta$  代表汽车前轮的转向角, 而汽车的转弯半径由  $r$  表示, 轴距由  $l$  表示。考虑到前轮转角  $\theta$  与转弯半径  $r$  之间存在公式2-1描述的等量关系, 按照公式2-2定义转向曲率  $\kappa$ , 则车辆状态也可以由变量  $(x, y, \phi, v, \kappa)$  等效定义, 两种定义下的配置空间是同胚的。

### 2.1.2 汽车的向量场

对于汽车而言, 通常只能通过方向盘和油门进行控制, 转向盘和油门分别独立影响参数  $\kappa$  和  $v$ , 故车辆配置的 5 个变量中只有  $\kappa$  和  $v$  是可直接控制的变量, 假定控制变量  $\kappa$  和  $v$  是线性变化的, 则汽车配置空间的所有线性独立的向量场可以由式2-3至2-5定义, 向量场体现了汽车运动学的特性, 约束了汽车的运动方式。由此也可推出, 汽车配置空间的维度是大于其线性独立向量场个数的, 因此汽车的运动规划问题是一个非完整约束问题。

$$\text{Steer}(\kappa) = \lambda \frac{\partial}{\partial \kappa} \quad (2-3)$$

$$\text{Drive}(v) = \gamma \frac{\partial}{\partial v} \quad (2-4)$$

$$\text{Move}(x, y, \phi, \kappa, v) = v \cos \phi \frac{\partial}{\partial x} + v \sin \phi \frac{\partial}{\partial y} + v \kappa \frac{\partial}{\partial \phi} \quad (2-5)$$

其中,  $\lambda$  和  $\gamma$  为任意给定常数。

三个向量场的积分曲线分别如式2-6、2-7、2-8所示。

$$\kappa(s) = \lambda s + \kappa_0 \quad (2-6)$$

$$v(s) = \gamma s + v_0 \quad (2-7)$$

$$\begin{aligned} \phi(s) &= \phi_0 + \int_0^s v(\tau) \kappa(\tau) d\tau \\ x(s) &= x_0 + \int_0^s v(\tau) \cos[\phi(\tau)] d\tau \\ y(s) &= y_0 + \int_0^s v(\tau) \sin[\phi(\tau)] d\tau \end{aligned} \quad (2-8)$$

### 2.1.3 泊车规划问题定义

基于上述对汽车配置空间及其向量场的描述，汽车运动规划问题可以一般地表述为：对于配置空间为  $C$  的车辆，给定初始配置  $X_{start}$  和目标配置  $X_{goal}$ ，求解一个（积分）曲线序列  $\mathcal{P} = (P_n)_{n=0}^N$ ，满足  $P_0(0) = X_{start}$ ,  $P_N(s) = X_{goal}$ ，且  $\forall P \in \mathcal{P}, P \in C_{free}$ （“ $P \in C_{free}$ ”表示该曲线经过的所有配置都属于  $C_{free}$ ），同时，尽可能最小化路径距离  $D(\mathcal{P})$ 。

但是要在 5 维高空间中求解非完整约束问题相当复杂，仅是求解式2-6至2-8定义的积分曲线就相当困难，涉及到大量常微分方程的求解。因此有必要针对泊车问题的特点进行合理的降维，以降低求解难度。

考虑  $C_{free}$  空间的维度，对于一个给定配置，其是否属于  $C_{free}$  是与车辆前轮的转向角无关的，而是取决于车辆的外轮廓的位置和朝向，因此  $C_{free}$  是与前轮转角  $\theta$  无关的空间，即与转向曲率  $\kappa$  无关。同时，考虑到泊车场景是一个相对静止的场景，即周围障碍物的移动都比较缓慢，因此可以认为障碍物在一定时间内是保持静止的，也就是  $C_{free}$  空间是时间无关的，因此可以合理认为  $C_{free}$  也是与速度  $v$  无关的。综上，对于泊车规划问题，其  $C_{free}$  空间可以看作是只与  $x, y, \phi$  有关的三维空间。

考虑配置空间的向量场，由于在泊车场景下，车辆往往需要原地调整前轮转向，故前轮转角的改变与车辆的位移和朝向的改变是相对独立的，因此可以合理将  $\kappa$  与  $(x, y, \phi)$  解耦，认为  $\kappa$  是向量场 Move 的超参数而非变量。同时，泊车场景下，车辆往往需要进行前后档切换，车辆移动速度较小，因此可以合理认为汽车在泊车过程中是保持匀速的，由此可将  $v$  同样视为向量场 Move 的常量而非变

量。综上，向量场  $\text{Move}$  可看作是以  $\phi$  为超参数、 $v$  为常量、 $(x, y, \phi)$  为变量的三维场，如式2-9。

$$\text{Move}^\kappa(x, y, \phi) = \cos \phi \frac{\partial}{\partial x} + \sin \phi \frac{\partial}{\partial y} + \kappa \frac{\partial}{\partial \phi} \quad (2-9)$$

当  $\kappa = 0$  时，向量场  $\text{Move}^\kappa$  的积分曲线如式2-10，曲线是斜率为  $\phi_0$  的直线；当  $\kappa \neq 0$  时，积分曲线如式2-11，曲线是半径为  $1/\kappa$ ，以  $(x_0 - \sin \phi_0 / \kappa, y_0 + \cos \phi_0 / \kappa)$  为圆心的圆弧。即沿着向量场  $\text{Move}$  移动的汽车，只有两种运动轨迹：半径为  $1/\kappa$  的圆弧和斜率为  $\phi_0$  的直线。

$$\begin{aligned} \phi(s) &= \phi_0 & \phi(s) &= \phi_0 + \kappa s \\ x(s) &= x_0 + \sin \phi_0 s & x(s) &= x_0 + \frac{1}{\kappa} [\sin(\phi_0 + \kappa s) - \sin \phi_0] \\ y(s) &= y_0 + \cos \phi_0 s & y(s) &= y_0 + \frac{1}{\kappa} [-\cos(\phi_0 + \kappa s) + \cos \phi_0] \end{aligned} \quad (2-10) \quad (2-11)$$

因此，综上所述，对于泊车规划问题，可以在合理简化  $C_{free}$  空间和向量场之后，将 5 维配置空间  $C(x, y, \phi, \kappa, v)$  简化为以  $\kappa$  为超参数的 3 维  $(x, y, \phi)$ -空间，记为  $C^\kappa(x, y, \phi)$ ，对应的向量场  $\text{Move}^\kappa$  如式2-9所示。从而将泊车规划问题简化为在三维  $(x, y, \phi)$ -配置空间的运动规划问题。由此，可将自主泊车运动规划问题依2.1定义，进一步，依此定义，一个泊车规划问题可由一个四元组  $(X_{start}, X_{goal}, C_{free}, \text{Move}^\kappa)$  概括表示，由于  $C_{free}$  是由工作空间  $\mathcal{W}$  决定的，因此问题也可以等效的由四元组  $(X_{start}, X_{goal}, \mathcal{W}, \text{Move}^\kappa)$  表示。

定义 2.1 泊车运动规划是指，在配置空间  $C^\kappa(x, y, \phi)$  中，给定起始和目标配置  $X_{start}$  和  $X_{goal}$ ，求解积分曲线序列  $\mathcal{P} = (P_i)_{i=0}^n$ ，满足：(1)  $P_0(0) = X_{start}$ ,  $P_n(s) = X_{goal}$ 。(2)  $\forall P \in \mathcal{P}, P \in C_{free}$ 。(3)  $\forall P \in \mathcal{P}, P' = \text{Move}^\kappa(P)$ 。同时，尽可能最小化  $D(\mathcal{P})$ 。

## 2.2 基于采样的运动规划算法

### 2.2.1 应用基于采样的算法的原因

运动规划问题已经被证明是一个 NP-完全问题 [6][7]，甚至一个基本的广义钢琴移动问题（在完全已知的环境中移动钢琴从起始点到目标点）也是 PSPACE-困难的 [29]。这意味着很难在合理的时间内用完备甚至最优的规划算法来解决运动规划问题 [8][11]。

应用基于采样的运动规划算法求解运动规划问题的一个原因在于其本质上是一类蒙特卡洛方法。而蒙特卡洛技术已经成功地应用于求解 NP-困难问题，其有效性基于这样的事实：NP-困难问题虽然有很大搜索空间，但是与之关联的是，往往存在一个很大的“可以接受”的次优解组成的解集。类似的，基于采样的运动规划算法之所以高效，也是因为尽管运动规划问题搜索空间很大，但是往往都有很多“可以接受”的次优解，因此，只要算法获取足够的信息或者指引，就可以在随机搜索过程中很快找到一个“可以接受”的次优解 [30]。另外一个应用基于采样的运动规划算法的原因在于其可以避免障碍物表示，即将障碍物显式地在配置空间中表示。这种表示有助于显式地划分出  $C_{free}$  空间，但是需要耗费大量计算的，尤其是对高维系统和存在大量障碍物的场景 [11][31]。

### 2.2.2 常用的基于采样的算法

基于采样的算法有两种基本的类型：多查询和单查询。对于运动规划问题，主要应用单查询方法，代表算法主要有 RRT、RRT\*。如表2-1所示，与 RRT 相比，RRT\* 是渐进最优和单调收敛的，这意味着 RRT\* 规划的路径往往比 RRT 更优，也使得 RRT\* 成为应用更广泛的运动规划算法。

表 2-1 RRT 和 RRT\* 的算法特性和复杂度对比 [11]。

算法	概率完备	渐进最优	单调收敛	时间复杂度 (构建树 + 查询路径)	空间复杂度
RRT*	是	是	是	$O(n \log n) + O(n)$	$O(n)$
RRT	是	否	否	$O(n \log n) + O(n)^{[1]}$	$O(n)$

[1] 变量  $n$  代表从自由配置空间采样的次数。

基于 RRT\* 存在很多的变种算法，针对泊车问题，Bi-RRT\*[23] 是其中最被广泛应用的一种。Bi-RRT\* 算法的伪代码如算法1所示。Bi-RRT\* 保留了 RRT\* 概率完备、渐进最优和单调收敛的特性，不同之处主要在于 Bi-RRT\* 同时构建两棵树  $T_s$  和  $T_g$ ， $T_s$  从起始配置往目标配置发展，而  $T_g$  则从目标配置往起始配置发展。Bi-RRT\* 中每棵树的构建过程与 RRT\* 中的基本都相同，但考虑到需要在合适顶点间连接两棵树，并且树数量的增加会间接增加顶点数量从而提高计算复杂度，因此 Bi-RRT\* 对算法进行了针对性的扩展，主要的扩展有三处，分别是算法1中的

行11、行14和行15。在行11，语句  $\text{Cost}(\mathcal{X}_{\text{nearest}}^*) + \mathcal{D}(\mathcal{X}_{\text{nearest}}^*, \mathcal{X}_{\text{new}}) + \text{CostToGo}(\mathcal{X}_{\text{new}})$  等效于  $\text{Cost}(\mathcal{X}_{\text{new}}) + \text{CostToGo}(\mathcal{X}_{\text{new}})$ ，代表从树根节点经过顶点  $\mathcal{X}_{\text{new}}$  到达目标（树发展的目标，而非规划目标）的最短路径长度，以  $p_{\mathcal{X}_{\text{new}}}$  表示，其中函数 **CostToGo** 返回输入顶点到目标的距离。 $c_{\text{best}}$  则代表当前规划得到的最短路径的长度，如果当前不存在路径则  $c_{\text{best}} = \infty$ 。通过判断  $p_{\mathcal{X}_{\text{new}}}$  是否比  $c_{\text{best}}$  更小，就可以判断将顶点  $\mathcal{X}_{\text{new}}$  加入树是否可能获得比当前路径更优的路径，从而确保每次加入树的顶点都是有用的，减少树顶点的冗余；在行14的函数 **ConnectGraphs** 会检查通过新加入的顶点  $\mathcal{X}_{\text{new}}$  能否连接两棵树并产生更优的路径，其伪代码如算法2所示。函数首先利用子函数 **Nearest** 从树  $T_b$  的顶点集  $\mathcal{V}_b$  中查询离  $\mathcal{X}_{\text{new}}$  欧拉距离最近的点  $\mathcal{X}_{\text{nearest}}$ ，然后利用函数 **Near** 在  $\mathcal{V}_b$  中查询所有离  $\mathcal{X}_{\text{nearest}}$  欧拉距离小于  $r_{\text{RRT}*}$  的所有顶点，组成待检查点集  $\{\mathcal{X}\}_{\text{near}}$ ，最后从集合  $\{\mathcal{X}\}_{\text{near}} \cup \{\mathcal{X}_{\text{nearest}}\}$  中查询能无碰撞地从  $\mathcal{X}_{\text{new}}$  到达且能产生长度比  $c_{\text{best}}$  更短的路径的顶点，如果这种顶点存在则更新  $c_{\text{best}}$  和最优路径  $\mathcal{P}_{\text{best}}$ 。行15的函数 **BranchAndBound** 作用是为了修剪树中的随着规划路径不断被优化而产生的冗余顶点，函数遍历两棵树中的所有顶点，并删除属于集合  $\{\mathcal{X} | \text{Cost}(\mathcal{X}) + \text{CostToGo}(\mathcal{X}) > c_{\text{best}}\}$  的顶点，从而确保树中的所有顶点都有可能在接下来的树发展过程中产生更优的路径。

Bi-RRT\* 的输出的路径为树的顶点序列  $\mathcal{P} = (\mathcal{X}_i)_{i=0}^n$ ，其中  $\mathcal{X}_0 = \mathcal{X}_{\text{start}}$ ， $\mathcal{X}_n = \mathcal{X}_{\text{goal}}$ 。另外路径  $\mathcal{P}$  也可以用边序列（即积分曲线序列）的形式等效表示，即  $\mathcal{P} = (\mathcal{E}_i)_{i=0}^{n-1}$ 。因此路径的顶点序列表示与定义2.1中的积分曲线序列表示是等效的。本文中将路径  $\mathcal{P} = (\mathcal{X}_i)_{i=0}^n$  中除去起始和目标配置的其他配置称为路径的采样点，依定义2.2定义。

定义 2.2 对于基于采样的规划算法，其输出路径为  $\mathcal{P} = (\mathcal{X}_i)_{i=0}^n$ ，称  $\mathcal{P}$  的子序列  $\mathcal{P}_s = (\mathcal{X}_i)_{i=1}^{n-1}$  为  $\mathcal{P}$  的采样点序列， $\mathcal{P}_s$  中的配置称为  $\mathcal{P}$  的采样点。全局最优路径的采样点序列为最优采样点序列，其采样点称为最优采样点。。

### 2.2.3 常用采样启发式

本小节主要介绍其中两种被广泛应用的先进启发式：高斯偏置采样（Gaussian-biased Sampling, GBS）[9] 和朝向感知空间探索（Orientation-aware Space Exploration, OSE）[12]。

**算法 1 Bi-RRT\*( $\mathcal{X}_{start}, \mathcal{X}_{goal}, \mathcal{W}, \mathcal{F}$ )**


---

```

1:  $T_s = (\mathcal{V}_s = \{\mathcal{X}_{start}\}, \mathcal{E}_s = \emptyset), T_g = (\{\mathcal{V}_g = \mathcal{X}_{goal}\}, \mathcal{E}_g = \emptyset)$ 
2:  $T_a = T_s, T_b = T_g$ 
3:  $c_{best} = \infty, \mathcal{P}_{best} = \emptyset$ 
4: for  $i$  in  $1 \dots, N$  do
5:    $\mathcal{X}_{rand} = \text{SampleFree}(\mathcal{W})$ 
6:    $\mathcal{X}_{nearest} = \text{Nearest}(\mathcal{V}_a, \mathcal{X}_{rand})$ 
7:    $\mathcal{X}_{new} = \text{Steer}(\mathcal{X}_{nearest}, \mathcal{X}_{rand}, \mathcal{F})$ 
8:   if CollisionFree( $\mathcal{X}_{nearest}, \mathcal{X}_{new}, \mathcal{W}$ ) then
9:      $\{\mathcal{X}\}_{near} = \text{Near}(\mathcal{V}_a, \mathcal{X}_{new}, r_{RRT*})$ 
10:     $\mathcal{X}_{nearest}^* = \text{Nearest}^*(\{\mathcal{X}\}_{near}, \mathcal{X}_{nearest}, \mathcal{X}_{new}, \mathcal{F}, \mathcal{W})$ 
11:    if Cost( $\mathcal{X}_{nearest}^*$ ) +  $\mathcal{D}(\mathcal{X}_{nearest}^*, \mathcal{X}_{new}) + \text{CostToGo}(\mathcal{X}_{new}) < c_{best}$  then
12:       $\mathcal{V}_a = \mathcal{V}_a \cup \{\mathcal{X}_{new}\}, \mathcal{E}_a = \mathcal{E}_a \cup \{(\mathcal{X}_{nearest}^*, \mathcal{X}_{new})\}$ 
13:      Rewire( $\{\mathcal{X}\}_{near}, \mathcal{X}_{new}, \mathcal{E}_a, \mathcal{F}, \mathcal{W}$ )
14:      ConnectGraphs( $c_{best}, \mathcal{P}_{best}, \mathcal{V}_b, \mathcal{X}_{new}, \mathcal{F}, \mathcal{W}$ )
15:      BranchAndBound( $T_a, T_b$ )
16:    end if
17:  end if
18:  SwapTrees( $T_a, T_b$ )
19: end for
20: return  $\mathcal{P}_{best}$ 

```

---

GBS 启发式是一种简单有效的采样启发式，对于给定参考配置，其通过设定的高斯参数对配置的每个维度进行偏置以实现偏置采样。对于配置空间为  $C^k(x, y, \phi)$  的汽车，GBS 启发式的伪代码如算法3所示，对于给定参考配置  $\mathcal{X}_{ref}$  和各维度的高斯偏置模型参数(均值  $\mu$  和方差  $\sigma$ )，在  $x$  和  $y$  维度，算法首先通过高斯随机方法计算配置  $\mathcal{X}_{ref}$  在  $(x, y)$ -平面的极坐标系内的极径的随机偏置  $r$  和极角的随机偏置  $\alpha$ ，然后将极坐标的偏置转换到平面直角坐标系上作为在  $x$  和  $y$  维度上的偏置，在维度  $\phi$ ，算法直接计算高斯随机数作为偏置。高斯偏置采样启发简单有效，且可以集成到许多不同的算法中，因此得到广泛应用。但是由于不同的

---

算法2 ConnectGraphs( $c_{best}, \mathcal{P}_{best}, \mathcal{V}_b, \mathcal{X}_{new}, \mathcal{F}, \mathcal{W}$ )

---

```

1:  $\mathcal{X}_{nearest} = \text{Nearest}(\mathcal{V}_b, \mathcal{X}_{new})$ 
2:  $\{\mathcal{X}\}_{near} = \text{Near}(\mathcal{V}_b, \mathcal{X}_{nearest}, r_{RRT*})$ 
3: for  $\mathcal{X}$  in  $\{\mathcal{X}\}_{near} \cup \{\mathcal{X}_{nearest}\}$  do
4:   if  $\text{Cost}(\mathcal{X}_{new}) + \mathcal{D}(\mathcal{X}_{new}, \mathcal{X}) + \text{Cost}(\mathcal{X}) < c_{best}$  then
5:     if  $\text{CollisionFree}(\mathcal{X}_{new}, \mathcal{X}, \mathcal{W})$  then
6:        $c_{best} = \text{Cost}(\mathcal{X}_{new}) + \mathcal{D}(\mathcal{X}_{new}, \mathcal{X}) + \text{Cost}(\mathcal{X})$ 
7:        $\mathcal{P}_{best} = \text{GeneratePath}(\mathcal{X}_{new}, \mathcal{X})$ 
8:     end if
9:   end if
10: end for

```

---



---

算法3 GaussianBiasSampling( $\mathcal{X}_{ref}, (\mu_r, \sigma_r), (\mu_\alpha, \sigma_\alpha), (\mu_\phi, \sigma_\phi)$ )

---

```

1:  $r \leftarrow \mathcal{N}(\mu_r, \sigma_r), \alpha \leftarrow \mathcal{N}(\mu_\alpha, \sigma_\alpha), \phi \leftarrow \mathcal{N}(\mu_\phi, \sigma_\phi)$ 
2:  $\Delta x \leftarrow r * \cos \alpha, \Delta y \leftarrow r * \sin \alpha, \Delta \phi \leftarrow \phi$ 
3:  $\mathcal{X}_{rand} \leftarrow \mathcal{X}_{ref} + (\Delta x, \Delta y, \Delta \theta)$ 
4: return  $\mathcal{X}_{rand}$ 

```

---

高斯偏置参数对算法的启发效果影响显著，需要对不同应用场景针对性调整，所以高斯偏置采样启发式具有场景适应性不强的缺点。

OSE 启发式是另一种场景适应性更强的先进启发式。OSE 启发基于 A\* 算法设计，利用不同半径大小的圆对工作空间进行探索，以获取从初始配置到目标配置的一条由圆形节点组成的路径。OSE 求解的圆形路径实际上在自由配置空间中划出了一片从起始配置到目标配置的无碰撞区域，因此利用 OSE 作为启发式，可以指导算法在隐含可行路径解的无碰撞区域中采样，从而减少算法规划首条可行路径所需的采样数和提高收敛到最优解的速度。对于配置空间为  $C^*(x, y, \phi)$  的汽车，OSE 启发式的伪代码如算法4所示，其中符号  $c$  代表一个圆形节点，每个圆形节点的状态由四元组  $(x, y, \phi, r)$  决定， $r$  代表该圆形节点的半径，代表在  $xy$ -平面内该圆形节点的圆心  $(x, y)$  到离其最近的障碍的距离，因此，一个圆形节点不止表示了系统的一个配置，也表示了以该配置为中心的最大无碰撞圆形区域。除此之外一个圆形节点也包含了一般 A\* 算法中的节点应有的属性：父节点

---

**算法4 OSE( $c_{start}$ ,  $c_{goal}$ ,  $\mathcal{W}$ )**

---

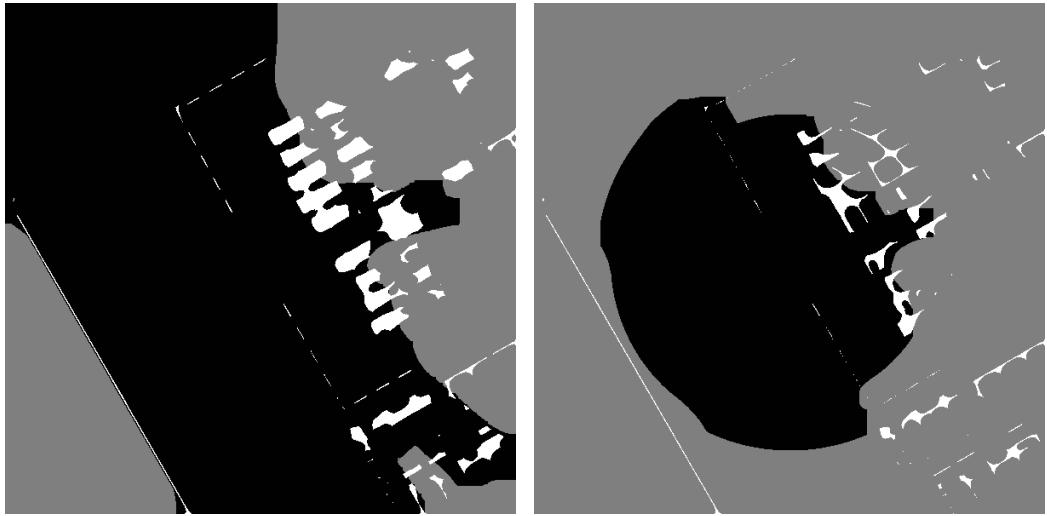
```

1:  $S_{close} \leftarrow \emptyset$ ,  $S_{open} \leftarrow \{c_{start}\}$ 
2: while  $S_{open} \neq \emptyset$  do
3:    $c_{top} \leftarrow \text{PopTop}(S_{open})$ 
4:   if  $f[c_{goal}] < f[c_{top}]$  then
5:     return  $\text{GenerateCirclePath}(c_{goal})$ 
6:   end if
7:   if  $\text{!Exist}(X, S_{close})$  then
8:      $S_{open} \leftarrow \text{Expand}(c_{top}, \mathcal{W}) \cup S_{open}$ 
9:     if  $\text{Overlap}(c_{top}, S_{goal})$  then
10:      if  $f[c_{top}] < f[c_{goal}]$  then
11:         $f[c_{goal}] = f[c_{top}]$ 
12:         $\text{parent}[c_{goal}] = c_{top}$ 
13:      end if
14:    end if
15:     $S_{close} \leftarrow \{c_{top}\} \cup S_{close}$ 
16:  end if
17: end while
18: return  $\emptyset$ 

```

---

$parent$ 、根节点到该点的距离  $g$ 、以及该点到目标的启发距离  $h$ ,  $h$  和  $g$  之和用  $f$  表示。给定起始节点  $c_{start}$  和目标节点  $c_{goal}$ , OSE 算法在探索过程中维护两个集合  $S_{close}$  和  $S_{open}$ , 分别表示已访问的节点和待访问节点, 算法会循环地从  $S_{open}$  取出并访问节点直到集合为空。在每次循环, 函数 **PopTop** 从  $S_{open}$  中取出  $f$  最小的节点  $c_{top}$ , 如果  $f[c_{goal}] < f[c_{top}]$ , 则表示  $S_{open}$  中不存在可以产生更优路径的节点, 算法结束, 通过函数 **GeneratePath** 生成并返回路径。否则继续通过函数 **Exist** 判断  $c_{top}$  是否属于重复节点, 该判断需要计算两个节点间的距离, 在 OSE 中, 两个节点的距离由公式2-12给出, 其中  $\kappa_{max}$  代表最大的转向曲率, 该距离的计算考虑了节点的朝向角, 是 OSE 算法相较其他类似算法的最大不同, 如果  $c_{top}$  与  $S_{close}$  中任意一节点  $c$  的距离小于  $c$  的半径  $r[c]$  则  $c_{top}$  为重复节点, 反之为不重复节点。



(a) 基于深度相机云生成的占用栅格图 (b) 基于 32 线激光雷达生成的占用栅格图

图 2-2 自主泊车工作空间的两种占用栅格图表示。黑色、灰色和白色区域分别表示可行区域、未知区域和障碍区域。图2-2a为基于深度相机生成的占用栅格图，图2-2b为基于 32 线激光雷达生成的占用栅格图，前者显然比后者获取了更大的可行区域，也更好的保留了障碍物轮廓等信息。

若  $c_{top}$  不重复，则利用函数 **Expand** 计算  $c_{top}$  的所有子节点并加入  $S_{open}$ ，然后通过函数 **Overlap** 判断  $c_{top}$  与  $c_{goal}$  间的距离是否小于给定阈值，如果是则说明  $c_{top}$  可以到达目标，如果进一步满足  $f[c_{top}] < f[c_{goal}]$ ，即经过  $c_{top}$  到达目标是更好的选择，则更新  $c_{goal}$  的属性，将  $c_{goal}$  的父节点设为  $c_{top}$ ，最后将  $c_{top}$  加入  $S_{close}$ 。

$$d(i, j) = \max(\|(x_i - x_j, y_i - y_j)\|, |\phi_i - \phi_j|/\kappa_{max}) \quad (2-12)$$

## 2.3 工作空间表示和积分曲线求解

### 2.3.1 工作空间的表示

工作空间的表示形式对运动规划问题的求解至关重要，对于规划算法，简洁有效的表示工作空间有助于提高对给定配置和路径进行碰撞检查的速度。占用栅格图 (Occupancy Grid Map)[32] 是广泛应用于平面机器人运动规划的工作空间表示方法。占用栅格图将平面空间以一定的分辨率离散化为一个个格子，对于每一格，如果该区域存在障碍物，值置为 1，如果无障碍物，则值为 0，如果无法判断就置为 0.5。由于占用栅格图不需要对障碍物形状特征和运动状态进行提取和计算，计算复杂度主要取决于离散化的分辨率，因此栅格图的生成速度很快，能

满足实时性要求。考虑到，泊车场景往往都是平面的，因此应用占用栅格图来表示自主泊车的工作空间是合理的选择。图2-2展示了基于深度相机和32线激光雷达(Light Detection And Ranging, LiDAR)传感器制作的泊车场景的占用栅格图，可见占用栅格图对工作空间的表示效果显著地受到传感器性能的影响，深度相机由于能生成更稠密的点云，其对障碍物的细节的表示更丰富。因此要获取细节更丰富的工作空间表示，提高规划算法进行碰撞检查时的准确性，合理选择和组合不同的传感器是十分重要的。

### 2.3.2 积分曲线求解

对于任意给定的两个配置 $X_0$ 和 $X_s$ ，求解连接两配置的积分曲线或积分曲线序列(以下称曲线)是运动规划问题的重要子问题之一。通过求解配置间的曲线来判断两配置在向量场中是否可连通，并进一步通过插值获取曲线上的配置来检查曲线是否无碰撞，是包括基于采样的规划算法在内的众多规划算法的重要步骤之一。

对于自主泊车问题，根据定义2.1，积分曲线求解问题是指在式2-10或2-11定义的曲线束中，求解连接两给定配置的曲线。因为2-10定义的曲线为直线，2-11定义的曲线为圆弧，所以问题可以表述为用不同的直线段和圆弧段组成的序列连接给定两配置。同时，需要注意，直线和圆弧是有方向性的， $s > 0$ 代表车辆往前行驶，为正向， $s < 0$ 代表车辆往后倒车为反向，另外圆弧还有左右之分， $\kappa > 0$ 代表车辆往左转， $\kappa < 0$ 代表车辆往右转。

文献[33]给出了只考虑正向曲线段的求解方法，所得曲线称为Dubins曲线，Dubins曲线是最优的，即给定两配置间，只考虑正向曲线的条件下，Dubins曲线的长度是所有可能的曲线中最短的。Dubins曲线将向量场Move $^k$ (式2-9)的超参数 $\kappa$ 的可能取值限制为三个 $\{\kappa_0, 0, -\kappa\}$ ， $\kappa_0$ 代表车辆最大转向曲率，即最小转向半径的倒数。通过限制 $\kappa$ 的取值虽然可以缩小求解空间，但是也导致Dubins曲线只能是 $G^1$ 连续的。文献[34]扩展了Dubins曲线，提出了Reeds-Shepp曲线。Reeds-Shepp曲线进一步考虑了反向曲线段，实现了问题的完全求解，同时保持了曲线长度的最优性，因此Reed-Sheep曲线也被视为配置空间中沿着向量场的路径长度测量空间中的测地线，在需要考虑反向运动的泊车运动规划中得到广泛应用。

对于要求  $\kappa$  线性变化，即曲线  $G^2$  连续的情况，式2-11定义的曲线转化为欧拉螺线。要求解欧拉螺线需要大量积分计算，十分耗时，文献 [35] 提出了一种插值方法以实现高效求解，但求解速度仍不理想。文献 [36] 基于欧拉螺线进一步提出了  $G^2$  连续(即  $\kappa$  线性变化)的 Reeds-Shepp 曲线的变种，虽然提高了曲线的连续性，但是无法保证曲线长度的最优性。

## 2.4 基于深度神经网络的目标检测

在本文的定义中，路径预测问题是指从图片形式表示的规划场景中提取场景的特征并预测采样点序列(依定义2.2)的长度和位置，这与机器视觉领域的目标检测问题有相近之处，研究目标检测领域先进的解决方法可以为路径预测问题的解决提供重要的启示。因此，本小节将介绍和对比目标检测领域的一些主流算法及其原理。

### 2.4.1 目标检测的主要方法

目标检测是指从给定图片中识别和定位特定类别的物体并用合适尺寸的矩形检测框将其框出，目前先进的目标检测算法都是基于深度学习的，主要可分为两类：一阶段方法和二阶段方法。

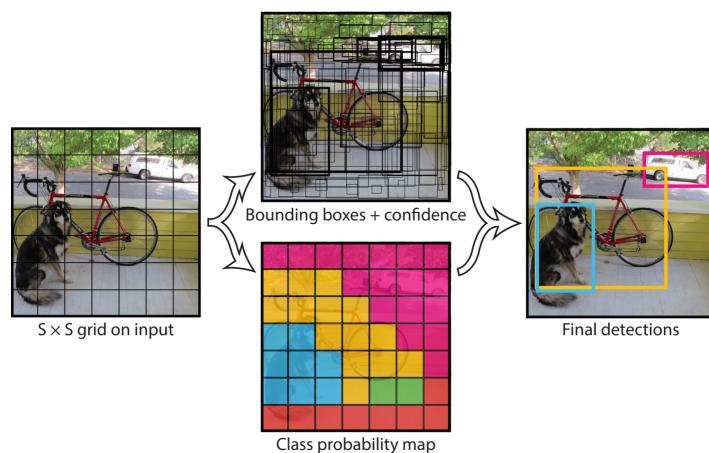


图 2-3 YOLO 的系统框图 [37]。

一阶段的算法主要有 YOLO[37]、SSD[38]、和 RetinaNet[39] 等。其中最具代表性的是 YOLO，其系统框图如图2-3所示，对于给定的输入图片，YOLO 首先将图片划分为尺寸  $S \times S$  的栅格。然后，对每个栅格，YOLO 会利用深度卷积

网络直接为预设的  $B$  个检测框预测其尺寸和置信度，置信度代表检测框是否框出了需要检测的对象，同时 YOLO 也会为每个栅格预测一个类别量，表示该栅格所预测的检测框所框出的对象的类别。最后根据检测框的置信度挑选检测框，并通过非极大值抑制 (Non-Maximum Suppression, NMS) 剔除重复的检测框，然后输出检测结果。

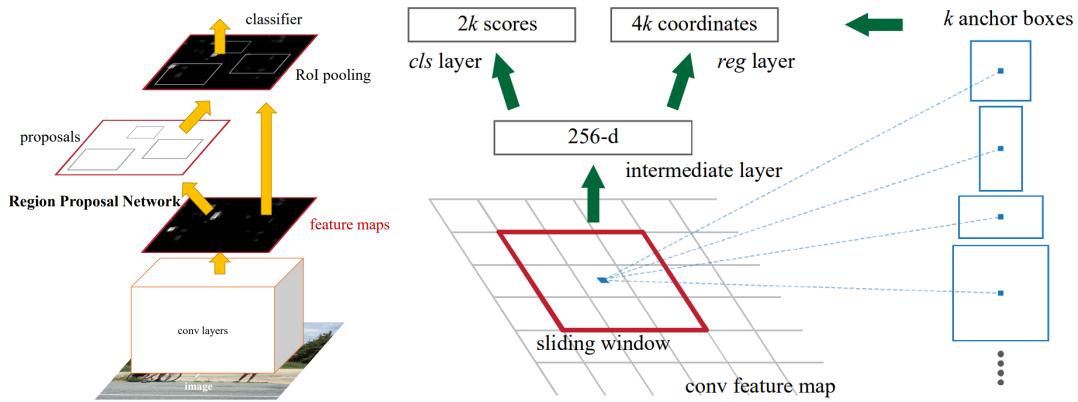


图 2-4 Faster-RCNN 的系统框图(左)和 RPN(Region Proposal Network) 的原理图(右)[40]。

属于二阶段的算法主要有 Faster-RCNN[40]、R-FCN[41] 和 Mask-RCNN[42] 等。其中最具代表性的是 Faster-RCNN，其系统原理图如图2-4左所示，对于给定输入图片，在第一阶段，Faster-RCNN 首先利用特征提取网络 (Features Extraction Networks, FEN) 提取输入图片的特征图，FEN 可以理论上可以是任意深度卷积神经网络，如 ResNet[20]、VGG[43] 等。然后利用区域提案网络 (Region Proposal Network, RPN)，以提取出的特征图为输入，输出为目标对象绘制的检测框的位置和尺寸。RPN 为自定义的轻型卷积神经网络，其原理图如图2-4右所示，RPN 会在特征图的每个像素位置上预锚  $k$  个尺寸不同的检测框，对每个检测框，RPN 会预测一个类别量，代表检测框所框出的内容是否包含需要检测的对象，四个坐标量，代表检测框的位置和大小。然后根据检测框的类别量，RPN 将不包含对象的检测框剔除，同时根据坐标量剔除超出图片范围的检测框，最后输出包含对象且位置符合要求的检测框。在第二阶段，对 RPN 输出的每个检测框，Faster-RCNN 会通过区域池化 (ROI Pooling) 将检测框所框出的内容的尺寸统一，然后用轻型卷积神经网络识别检测框所框出对象的类别并修正检测框的位置，最终输出检测结果。

### 2.4.2 目标检测方法的对比

一阶段和二阶段方法虽然在方法步骤上有所不同，但都需要利用特征提取网络从输入图片中提取特征图。因为特征提取网络都是比较深的大型卷积神经网络，所以需要利用大量数据来训练网络以提高网络的特征提取和泛化能力，但要制作大型的目标检测数据集是十分困难的，考虑到目标检测任务和图片分类任务的相似性，特征提取网络会先在包含 140 万组训练样本的大型图片分类数据库 Imagenet[45] 上针对图片分类任务进行预训练，使网络具有成熟的特征提取能力，再应用到目标检测任务中，减低网络训练难度的同时通过特征提取能力的迁移提高检测性能。

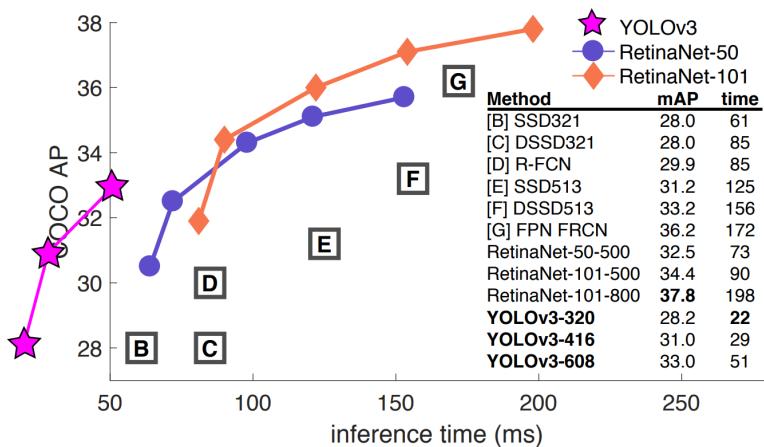


图 2-5 代表性的一阶段和二阶段目标检测算法的比较图。图中横坐标为推断时间，纵坐标为算法在数据集 COCO[44] 上的准确度。图中所有方法均运行在英伟达性能相同的两款显卡 Titan X 和 M40 上。[37][39]。

一阶段和二阶段的代表性算法的准确度 AP[44] 和推断时间的对比如图2-5所示。二阶段方法因为在第一阶段利用 RPN 子网络对的检测框进行分类，剔除了不包含检测对象，因此与一阶段方法相比，其检测误差一般会更小，但通过引入 Focal Loss[39]，一阶段方法也可以实现与二阶段方法相当的精度。另外，由于二阶段方法需要对检测框进行两次分类和位置校准，因此在运行速度上远远不如一阶段方法，难以实时推断，如图2-5中所示的二阶段方法中最快的 R-FCN，其推断时间是相近准确度的一阶段方法 YOLOv3-416 的三倍，推断速度相比而言还是很慢。而在一阶段方法中，以基于 YOLOv3[46] 的方法推断速度最快，在英

伟达 Titan X 显卡上可以实现 20 FPS 以上的帧率，远远快于同类的其他方法，同时，基于 YOLOv3 的方法也是在速度和准确度的综合性能上表现最好的。因此，对于本文提出的用于自主泊车实时运动规划的算法，借鉴基于 YOLOv3 的一阶段目标检测算法的设计和实现思路是更加合理的选择。

## 2.5 本章总结

在本章中，首先利用几何代数方法对本文所要研究的自主泊车规划问题进行了分析，根据泊车场景的特点，将 5 维配置空间的规划问题降为 3 维并进行了规范化。然后介绍了在泊车规划中应用基于采样的算法的原因、泊车规划常用的采样规划算法 Bi-RRT\*、以及两种先进的启发式 GBS 启发和 OSE 启发。同时，本章还介绍了工作空间表示和积分曲线求解两个泊车规划的重要子问题，针对泊车规划的特点本文使用占用栅格图表示工作空间，使用 Reed-Shepp 曲线求解积分曲线。最后，基于本文所提出的方法的特点，介绍了目标检测领域可供借鉴的基于深度卷积神经网络的一阶段和二阶段两类方法，基于自主泊车规划实时性的要求，选择基于 YOLO 的一阶段方法作为本文所提出的深度卷积神经网络的设计和实现参考。



## 第3章 二阶段自主泊车规划系统

在本章中将详细介绍本文提出的基于深度学习的二阶段自主泊车运动规划方法，包括第一个阶段中用于规划初始泊车路径的 YIPS 路径规划网络的设计和实现，以及第二阶段中用于优化初始泊车路径的采样优化算法 SO-RRT\* 的设计。此外，还将介绍方法中对 YIPS 网络的泛化误差进行最佳高斯拟合的方法和过程，以及泊车规划场景数据集的制作方法和流程。

### 3.1 系统框架

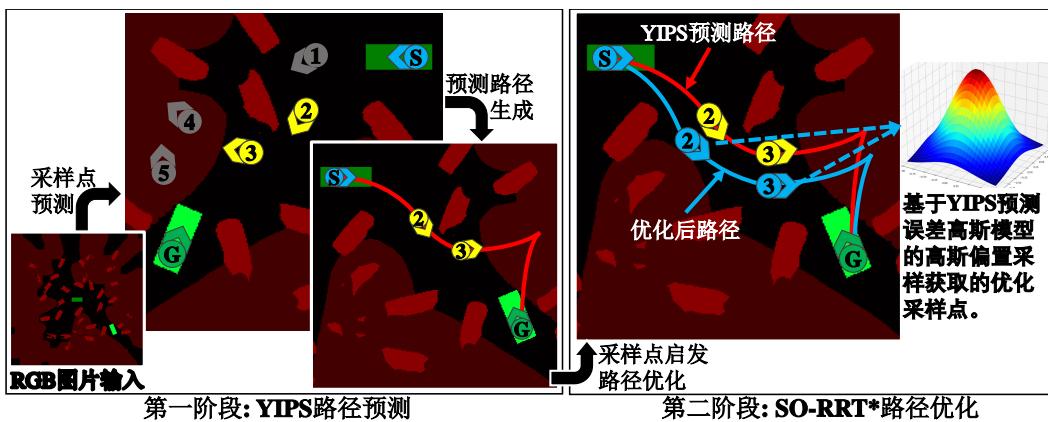


图 3-1 本文提出的基于深度学习的二阶段自主泊车运动规划方法的系统框图。图中由圆形和箭头组合而成的图形表示一个车辆配置，箭头指示车辆朝向，圆形指示车辆位置。图左框中灰色的 1、4、5 号配置表示被 YIPS 路径规划网络识别为无效的预设采样点，而左框中黄色的 2、3 号配置则是被识别为有效的预设采样点。图中蓝色 S 号和绿色 G 号配置分别表示起始和目标配置。红色曲线表 YIPS 预测的初始泊车路径。蓝色曲线表示 SO-RRT\* 路径优化算法对初始泊车路径的优化结果。

本文提出的基于深度学习的自主泊车运动规划方法的系统框架如图3-1所示，该方法分为两个阶段，在第一阶段，利用本文提出的 YIPS (You Infer Paths Straightly) 路径规划神经网络，以编码了起始配置、目标配置和占用栅格图的规划场景图片为输入，为泊车车辆规划初始泊车路径。利用神经网络并行运算快和场景适应性强的优势，提高方法规划首条可行路径的速度和稳定性。在第二阶段，考虑到 YIPS 网络存在泛化误差，初始路径仍有优化空间，因此利用本文提

出的基于采样的路径优化算法 SO-RRT\* 对初始路径进行优化以确保方法输出无碰撞和高质量的泊车路径。SO-RRT\* 对初始路径的优化过程即是利用初始路径指导 SO-RRT\* 在最优采样点附近采样的启发规划过程。通过初始路径的启发加速 SO-RRT\* 的收敛速度，提升方法整体的规划速度和质量。此外，在 SO-RRT\* 的采样阶段，利用 YIPS 网络泛化误差的最佳拟合高斯模型指导 SO-RRT\* 的偏置采样过程，以提高算法优化初始路径的能力。

## 3.2 第一阶段：YIPS 路径规划网络

### 3.2.1 输入输出设计

根据定义2.1，一个规划场景可以由四元组  $\mathcal{S} = (\mathcal{X}_{start}, \mathcal{X}_{goal}, \mathcal{W}, \text{Move}^\kappa)$  表示，又根据定义2.2，对采样规划算法而言，规划最优路径的实质是寻找最优的采样点序列  $\mathcal{P}_s = (\mathcal{X}_i)_{i=1}^{n-1}$ ，于是基于这两个定义，设计泊车路径规划算法相当于设计一个从  $\mathcal{S}$  到  $\mathcal{P}_s$  的映射  $H : \mathcal{S} \rightarrow \mathcal{P}_s$ ，而利用神经网络进行路径预测，就相当于通过训练神经网络来拟合映射  $H$ 。因此 YIPS 路径规划神经网络的学习任务就是拟合函数  $H$ ，实现以泊车规划场景  $\mathcal{S}$  为输入，推断出对应的泊车路径即采样点序列  $\mathcal{P}_s$ 。而对于给定车辆的泊车任务， $\text{Move}^\kappa$  不随规划场景变化， $\mathcal{S}$  可以进一步简化为  $\mathcal{S}^\kappa = (\mathcal{X}_{start}, \mathcal{X}_{goal}, \mathcal{W})$ 。因此针对特定车辆训练的 YIPS 网络可以定义为映射  $\text{YIPS} : \mathcal{S}^\kappa \rightarrow \mathcal{P}_s$ 。

而映射 YIPS 可以通过学习进行拟合的原因在于，采样点序列  $\mathcal{P}_s$  与规划场景  $\mathcal{S}^\kappa$  间存在隐式对应关系。图3-2展示了四类典型泊车场景中的采样点分布，由图可见，对于规则停车场的垂直和水平泊车场景(图3-2左两列)，最优采样点均分布在目标泊车位的入口处，最优采样点数量则与场景中可行区域的面积大小有关，泊车场景越拥挤，则最优采样点数量越多。而对于半规则和非规则停车场(图3-2右两列)，最优采样点则分布在重要的绕障点附近，起始和目标配置间的可通行道路越曲折和狭窄，则最优采样点数量越多。同时由图可见，四种场景下，最优采样点序列  $\mathcal{P}_s$  的附近均存在大量的次优采样点，因此尽管神经网络的拟合存在泛化误差，但通过对网络预测的次优路径进行优化也可以使路径逼近甚至收敛到最优路径。

作为 YIPS 神经网络输入  $\mathcal{S}^\kappa$  的组成部分， $\mathcal{X}_{start}$ 、 $\mathcal{X}_{goal}$  和  $\mathcal{W}$  具有不同的表示

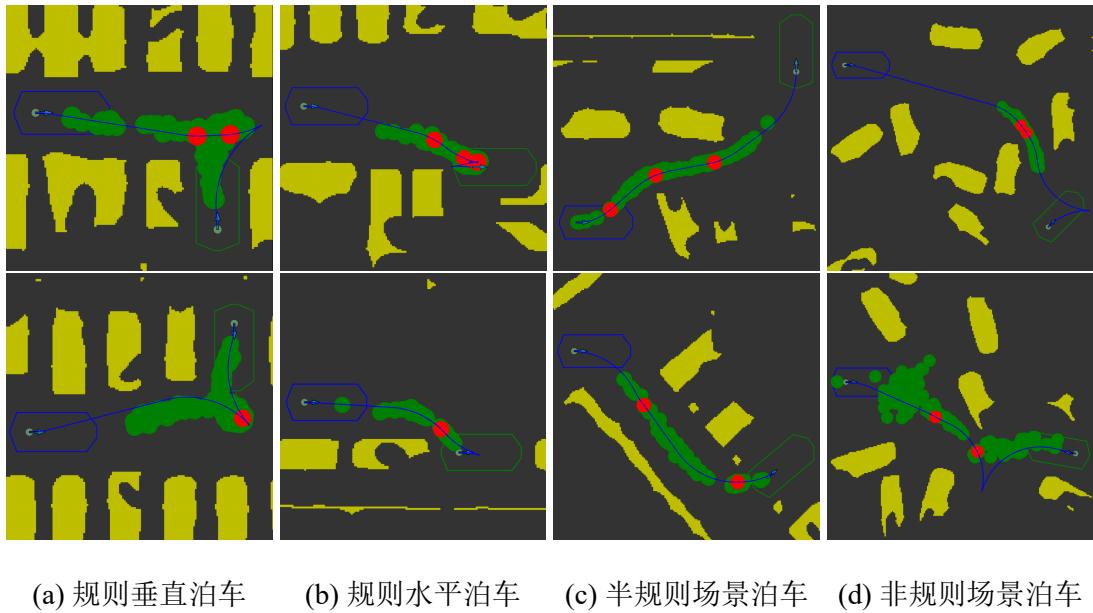


图 3-2 四类典型泊车场景的最优及次优采样点(依定义2.2)分布图。左两列依次为规则停车场的垂直和水平泊车场景,右两列依次是半规则和非规则泊车场景。对每一场景,使用基于 OSE 方法启发的 Bi-RRT\* 算法规划 100 次,每次规划采样 2000 次。将每次规划输出的采样点作为次优采样点,以绿色圆点表示其位置,取 100 次规划输出的所有路径中长度最小的一条作为近似最优路径,其采样点作为最优采样点,以红色圆点表示。此外,黄色像素表示障碍物,蓝色曲线为最优泊车路径,蓝色和绿色多边形分别表示起始和目标配置。

形式,为了便于网络学习和训练,有必要对  $X_{start}$ 、 $X_{goal}$  和  $\mathcal{W}$  的表示进行规整。存在两种可能的规整方法,一种是将  $\mathcal{W}$  规整到  $X_{start}$  和  $X_{goal}$  所在的配置空间  $C(x, y, \phi)$ ,用  $C(x, y, \phi)$  中的  $C_{obs}$  子空间间接表示  $\mathcal{W}$ ,最终以三维空间  $C(x, y, \phi)$  作为网络输入。另一种是将  $X_{start}$  和  $X_{goal}$  表示到  $\mathcal{W}$  中,用相当于灰度图的占用栅格图表示  $\mathcal{W}$ ,再通过平面几何图形间接表示  $X_{start}$  和  $X_{goal}$ ,最后将平面几何图形与占用栅格图整合为 RGB 图片作为网络输入。两种方法在同一场景下的编码结果对比图如图3-3所示,对于第一种编码方式,首先,用  $C_{obs}$  间接表示  $\mathcal{W}$  的计算复杂度是相当高的,其次,要较好的保留  $\mathcal{W}$  中的障碍物特征,需要以较高的分辨率表示  $C(x, y, \phi)$ ,这会产生上千万的数据点,如图3-3右侧对  $C(x, y, \phi)$  的表示需要 2700 万个配置点,从而使网络输入尺寸变得相当大。相比而言,第二种编码方式可以避免对  $C(x, y, \phi)$  的表示且通过传感器制作占用栅格图的复杂度相对更低,输入制作速度更快,用相对更少的数据点就能很好的保留障碍物的特征,如图3-3左侧对  $\mathcal{W}$  的表示只需 108 万个像素点。因此相对而言第二种 RGB

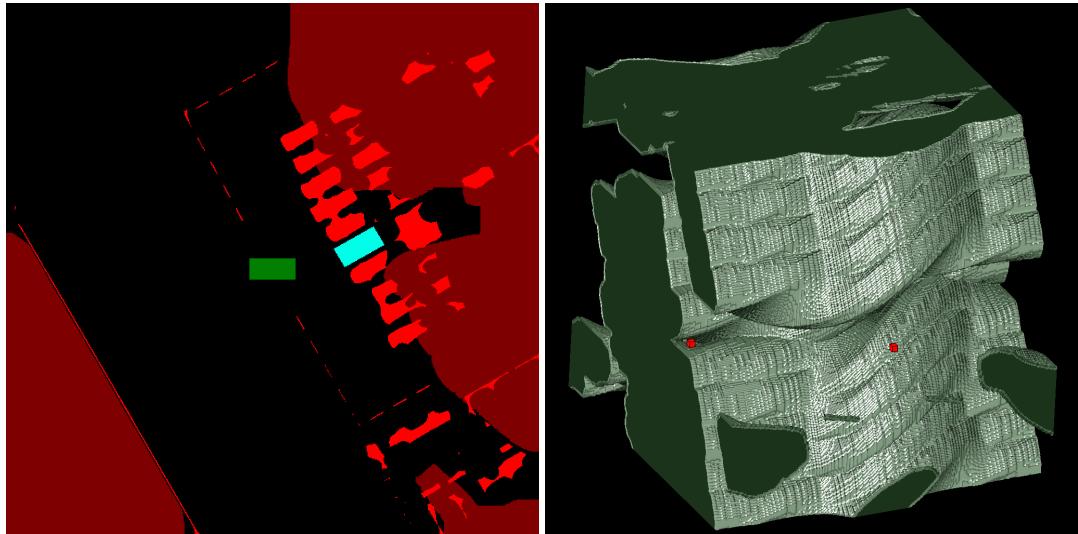


图 3-3 YIPS 网络的输入  $\mathcal{S}^k$  的不同编码方式对比图。左图为将起始和目标配置编码到工作空间的占用栅格图上的编码结果，以分辨率  $600 \times 600$  的 RGB 图片显示，其中绿色和青蓝色矩形表示起始和目标配置，红色、深红色和黑色像素区域代表工作空间中的障碍、未知和可行区域。右侧为将工作空间编码为配置空间中的障碍物子空间的编码结果，截取配置空间中大小为  $30 \text{ m} \times 30 \text{ m} \times 2\pi \text{ rad}$  的子空间，以  $300 \times 300 \times 300$  分辨率显示，其中左侧红点和右侧红点分别表示起始和目标配置（已放大 5 倍以可视化），绿色三维流形代表障碍物子空间。

图片的编码方法是一种更加合理的选择。同时，RGB 图片的编码方法也有利于使用基于图片输入的各种成熟卷积方法和网络设计技巧，与第一种编码方式需要的三维点云卷积操作相比，其成熟度更高，运算速度也更快。

此外，虽然存在多种将起始配置、目标配置和占用栅格图编码成 RGB 图片的方式，但本文选择将起始和目标配置以占用栅格图的分辨率编码为 RGB 图片中长宽为车辆实际长宽值的矩形图像，并在 RGB 图片中编码包括障碍、未知和可行区域在内的完整占用栅格图。实验结果表明这种编码形式在保证数据易读性的同时是各类编码形式中最利于 YIPS 网络学习的。

YIPS 网络的输出是采样点序列  $\mathcal{P}_s$ ，因此 YIPS 需要同时推断  $\mathcal{P}_s$  中的采样点数量以及采样点配置，前者可以理解为类别量，因为采样点数量是个离散值，后者可以理解为回归量，因为配置中的  $(x, y, \phi)$  都是连续量。因此路径规划问题是一个分类和回归任务并存的多任务学习问题，与目标检测问题相似度很高，根据节 2.4 的介绍，在目标检测中需要预测检测框的类别和尺寸位置，前者是分类任

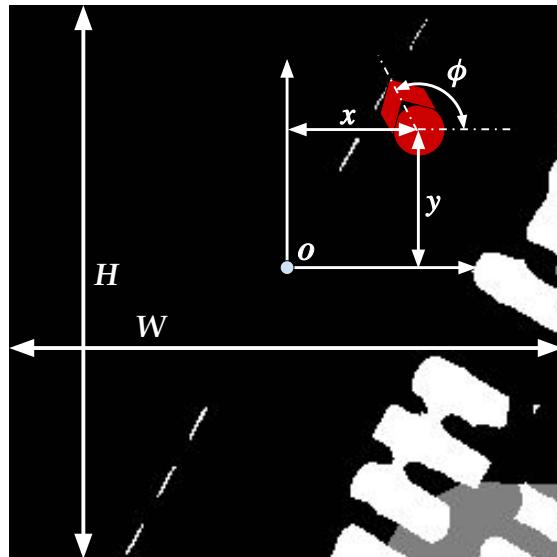


图 3-4 占用栅格图上的一个采样点。采样点以红色箭头和圆的组合来表示， $H$  和  $W$  分别表示占用栅格图的高和宽， $(x, y)$  和  $\phi$  表示采样点在占用栅格图的局部坐标系中的坐标值和朝向角。 $O$  为占用栅格图的局部坐标系的原点，也是占用栅格图的几何中心点。

务，后者是回归任务，因此在网络输出和架构设计上参考目标检测领域已有的成熟设计方法和技巧是合理且必要的。

参考目标检测网络 YOLO 的输出设计，YIPS 预设输出  $\mathcal{P}_s$  为由  $B$  个采样点组成的序列， $B$  与路径复杂度相关， $B$  越大则 YIPS 能规划的路径就越复杂，对于本文的泊车规划任务，选取  $B = 5$ 。对于  $\mathcal{P}_s$  中的每一采样点，YIPS 会为其预测  $A = 4$  个量：采样点的 3 个配置  $(x, y, \phi)$  和 1 个类别量  $p$ 。需要指出的是，预测量的数目  $A$  可随不同系统进行调整，系统配置空间  $C$  的维度越高则  $A$  越大， $A = \dim(C) + 1$ 。对于类别量  $p$ ，当  $p = 1$  时指示该预设采样点为必要类采样点，即属于目标输出序列  $\hat{\mathcal{P}}_s$  的采样点， $p = 0$  时则指示采样点不属于  $\hat{\mathcal{P}}_s$ ，为非必要类采样点。 $p$  也可以理解为一个概率值，即预设采样点属于  $\hat{\mathcal{P}}_s$  的概率。而对于配置量  $x$ 、 $y$  和  $\phi$ ，因其具有不同的取值范围，有必要对其取值空间进行归一化，让 YIPS 网络预测取值空间相同的配置量，将更有利子网络训练和学习。考虑在尺寸为  $W \text{ m} \times H \text{ m}$  的占用栅格图上的一个采样点，如图3-4所示，在占用栅格图的局部坐标系中，采样点的位置坐标不应超出栅格图的范围，同时，车辆的朝向

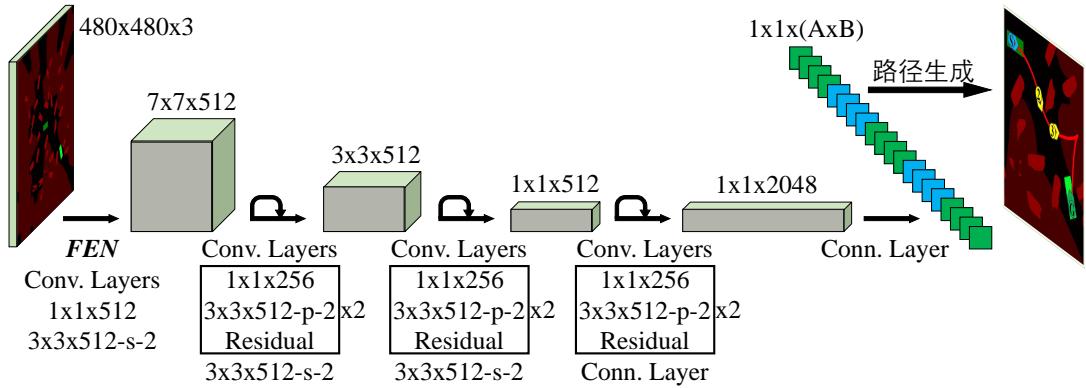


图 3-5 本文提出的 YIPS 路径规划神经网络的网络架构。其中 **FEN** 表示特征提取网络，虽然理论上所有成熟的大型卷积神经网络如 VGG、ResNet、Xception 和 DarkNet 等都能作为 YIPS 的主干，但是本文的实验结果表明，VGG 网络作为主干时 YIPS 更易于训练且训练效果是其中最好的。“Residual”代表残差连接结构 [20]。卷积尺寸中，“-s-2”和“-p-2”表示卷积的 strike 为 2 和 padding 为 2。网络的输出尺寸中。“路径生成”步骤中，利用 Reeds-Shepp 曲线连接采样点序列进而生成一条路径作为 YIPS 的规划路径。此外，在每一次卷积后都会进行批规范化 (Batch Normalization) [47] 以降低网络的训练难度和提高网络的鲁棒性。除了输出层外，所有神经元的激活函数均为 Leaky ReLU [48]。

角  $\phi$  的取值范围可以规整为  $[0, 2\pi]$ ，因此三个配置量可用公式3-1归一化。

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\phi} \end{bmatrix} = \left( \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} + \begin{bmatrix} \frac{W}{2} \\ \frac{H}{2} \\ \pi \end{bmatrix} \right) \circ \begin{bmatrix} \frac{1}{W} \\ \frac{1}{H} \\ \frac{1}{2\pi} \end{bmatrix} \quad (3-1)$$

其中， $(\tilde{x}, \tilde{y}, \tilde{\phi})$  代表归一化后的配置量， $\circ$  表示矩阵的哈达玛乘积。

### 3.2.2 网络架构设计

根据上述对 YIPS 网络输入输出形式的分析，并参考 YOLO 等一阶段目标检测网络的设计思想，本文将 YIPS 网络设计为单输入多输出的端到端深度卷积神经网络，其架构如图3-5所示。YIPS 网络主要有两个组成部分：特征提取主干和路径规划头部。前者负责从输入中提取特征图，后者负责根据特征图进路径规划。主干网络主要由大型的特征提取网络如 VGG、Resnet 组成，其参数量为千万级。对每一帧的规划场景图输入，主干网络首先将其尺寸规范为  $480 \times 480 \times 3$ ，然后提取输入的特征图，再用尺寸为  $1 \times 1$  的卷积核将特征图尺寸统一为  $15 \times 15 \times 512$ ，最后通过  $3 \times 3$  的卷积核 (strike 为 2) 将特征图尺寸卷成  $7 \times 7 \times 512$  后输出到路

径规划头部。需要指出的是，虽然理论上所有成熟的大型卷积神经网络如 VGG、ResNet、Xception 和 DarkNet 等都能作为 YIPS 的主干，但是本文的实验结果表明，VGG 网络作为主干时 YIPS 更易于训练且训练效果是其中最好的。

路径规划头部主要由三组残差结构 [20] 的卷积层组成，以第一组将  $7 \times 7 \times 512$  输入卷积成  $3 \times 3 \times 512$  输出的卷积层为例，卷积层会首先进行两次这样的残差操作：用尺寸为  $1 \times 1$  的卷积核将输入的通道数压缩为 256，再用  $3 \times 3$  的卷积核 (padding 为 2) 对压缩后的输入进行卷积后与原输入叠加。然后对两次残差操作的输出进行卷积核为  $3 \times 3$  的卷积操作 (strike 为 2)，将输入尺寸减半为  $3 \times 3 \times 512$ 。这种的残差结构的卷积操作可以在加深网络的同时更好的保留高维输入的特征，因而在深度神经网络设计中被广泛应用。第二组残差卷积层的结构和第一组完全相同，第三组与第一组的不同在于两次残差操作后接的是尺寸为 2048 的全连接层，将输入最终处理成目标输出尺寸  $1 \times 1 \times (A \times B)$ 。最后，根据类别预测挑选出所有属于必要类的采样点来构造采样点序列，再利用 Reed-Shepp 曲线连接序列的采样点最终生成 YIPS 所规划的路径。

基于 YIPS 网络主干 + 头部的架构特点，本文采用 Fine-tuning 方法对 YIPS 网络进行训练，即先在 Imagenet 数据集上预训练主干网络再训练整个 YIPS 网络。Fine-tuning 方法是进行迁移学习 [49] 的一种常用方法，不仅在目标检测领域，在自然语言处理领域也有广泛应用 [50]。而在 YIPS 网络的训练中采用 Fine-tuning 方法，旨在将主干网络在图像分类任务中学习到的特征提取知识迁移到 YIPS 网络中，虽然 YIPS 针对的路径规划任务与图像分类任务有较大差异，网络输入也有明显的不同，但是本文的实验结果表明，两个任务在特征提取知识上有相通之处，通过知识迁移可以加快 YIPS 网络的训练速度并提高训练效果，同时也一定程度上有利于克服路径规划训练数据不足 (相比与 ImageNet) 带来的泛化误差较大的问题。

### 3.2.3 损失函数设计

YIPS 神经网络的预测任务有二：采样点的类别预测以及采样点的配置预测，前者为分类任务，后者为回归任务。因此设计如式3-2所示，由分类损失  $J_c$  和回归损失  $J_r$  组成的损失函数  $J$ 。其中损失系数  $\lambda_r$  是用于平衡分类任务和回归任务的训练速度的超参数，通过调整  $\lambda_r$  使两个损失  $J_c$  和  $J_r$  的收敛速度一致可以提高

训练效果，本文中选取  $\lambda_r = 1e+04$  来实现该目标。

$$\begin{aligned} J &= J_r + J_c \\ J_c &= + \sum_b^B E(\hat{p}_b, S(t_{p_b})) \\ J_r &= \lambda_r \sum_b^B \hat{p}_b E([\hat{x}_b, \hat{y}_b, \hat{\phi}_b], S([t_b^x, t_b^y, t_b^\phi])) \end{aligned} \quad (3-2)$$

其中， $B$  表示预设采样点序列的长度， $\lambda_r$  为损失系数， $(\hat{x}, \hat{y}, \hat{\phi})$  和  $\hat{p}$  分别为采样点配置和类别的真实值即标签值， $S(\bullet)$  表示 Sigmoid 函数， $E(\bullet)$  表示二元交叉熵函数， $(t^x, t^y, t^\phi)$  和  $t^p$  是神经网络输出层对应于配置  $(x, y, \phi)$  和类别  $p$  的直接输出值，取值范围为  $[-\infty, \infty]$ ，通过 Sigmoid 函数将其映射到  $[0, 1]$  后作为网络预测值。

由式3-2可知，对网络预测的采样点序列，会依次计算序列的每一采样点在配置预测和类别预测上的损失，再将所有采样点的损失加和得到该网络预测的总损失。要指出的是，对于采样点的对应真实值，即标签采样点，如果标签采样点是必要类采样点，则其类别值  $\hat{p} = 1$ ，反之  $\hat{p} = 0$ ，因此在计算采样点的配置预测损失时，实际上只有当其对应的标签采样点是必要的，才会计算该采样点的配置损失。

$$L(z) = \ln \frac{z}{1-z} \quad (3-3)$$

$$J_r = |t^x - L(\hat{x})|^2 + |t^y - L(\hat{y})|^2 + |t^\phi - L(\hat{\phi})|^2 \quad (3-4)$$

其中， $z$  为输入值， $L$  为式3-3所示的 Logit 函数。

对采样点的类别预测损失，本文采用常用于分类任务的损失函数二元交叉熵。而对于配置预测损失，不同于 YOLOv3[46] 等方法中采用均方差函数，本文采用二元交叉熵函数计算损失，原因在于二元交叉熵函数可以有效避免指数溢出问题。采用均方差函数计算的单个采样点配置量的预测损失如式3-4所示，需要先利用 Logit 函数(如式3-3所示)计算该配置量对应的真实值的 logit 值，再利用 logit 值计算均方差，由于 Logit 函数在输入值  $z$  趋于 0 时输出值  $L(z)$  会趋于负无穷大，从而导致计算过程中可能出现下溢出的现象，造成程序运行的不稳定。为了避免这个问题，采用均方差函数计算时需要对配置的真实值进行预处理以

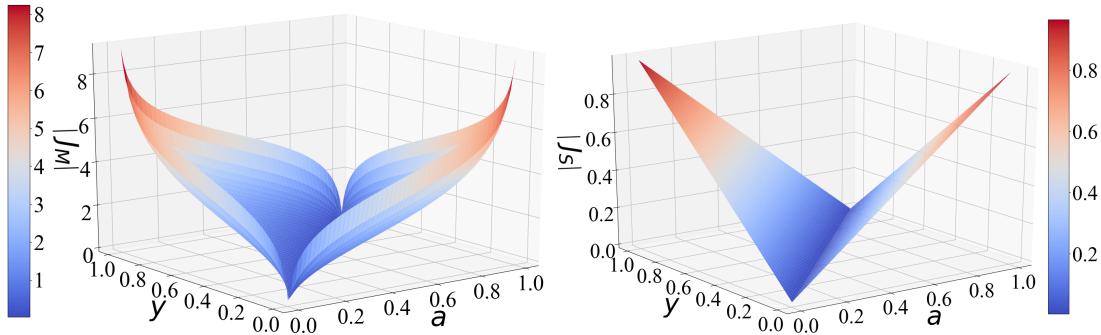


图 3-6 均方差损失和二元交叉熵损失的梯度绝对值关于网络预测值  $a$  和真实值  $y$  的函数图像。左侧图像为均方差损失的梯度绝对值  $|J_M|$ , 右侧为二元交叉熵损失的梯度绝对值  $|J_S|$ 。

避免溢出, 相比于采用二元交叉熵的损失计算方法增加了预处理的工作量。

$$J_S = (a - y)z \quad (3-5)$$

$$J_M = (L(a) - L(y))z \quad (3-6)$$

另外, 采用二元交叉熵函数获得的梯度比均方差函数更稳定。设  $a$  为输入  $z$  对应的网络预测值,  $a$  对应的真实值为  $y$ , 则使用交叉熵函数和均方差函数计算的梯度分别如式3-5和式3-6所示。设定输入  $z = 1$ , 则梯度  $|J_M|$  和  $|J_S|$  关于  $a$  和  $y$  的函数图像如图3-6所示, 进一步从图中截取出  $y$  分别为 0.3、0.5 和 0.7 时,  $|J_M|$  和  $|J_S|$  关于  $a$  的函数曲线如图3-7所示。对于均方差函数, 除  $y = 0.5$  时外, 其余情况下梯度曲线都是不对称的, 当  $y < 0.5$  时, 如  $y = 0.3$ ,  $a < y$  段的梯度变化率要大于  $a > y$  段, 当  $y > 0.5$  时, 如  $y = 0.7$ , 情况正好相反。这说明, 在  $y$  和  $a$  的绝对差  $\delta = |y - a|$  不变的情况下, 均方差函数计算的梯度与真实值有关, 当  $y < 0.5$  时,  $a < y$  时的梯度要大于  $a > y$  时的梯度, 而  $y > 0.5$  时, 情况则相反, 因此  $\delta$  的大小不能独立的反映出梯度的大小。与之相比, 交叉熵函数的梯度曲线更平衡, 其对称性不因  $y$  的变化而变化, 无论  $y$  取何值, 梯度的变化率都是相同的, 这意味着, 无论  $a < y$  还是  $a > y$ , 对于任意给定的  $y$ , 只要绝对差  $\delta$  是相同的, 计算得到的梯度就是相同的(不考虑正负), 独立于  $y$ , 因此  $\delta$  的大小可以独立的反映出梯度的大小。

综上所述, 虽然使用均方差函数和交叉熵函数都可以反映出预测值与真实值之间的差异, 但是后者比前者计算的梯度更稳定, 不受真实值变化的影响, 梯度值始终与绝对差  $\delta$  保持相同的线性关系。依照损失函数的梯度应尽可能准确稳

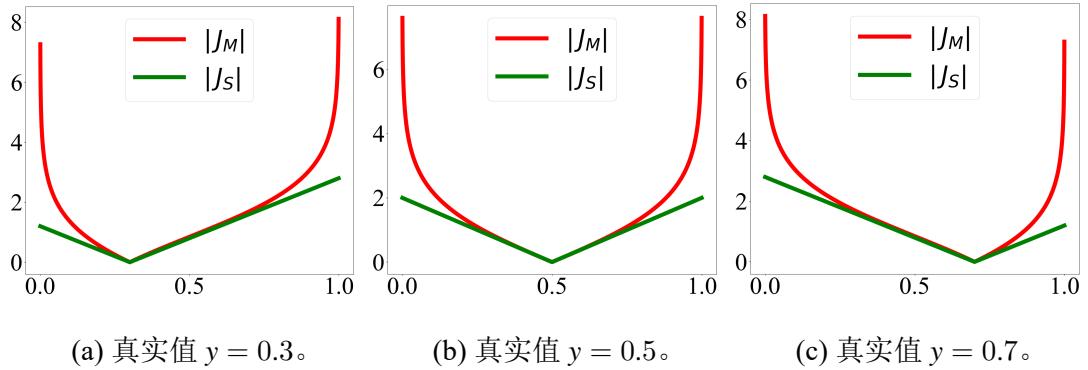


图 3-7 真实值  $y$  分别为 0.3、0.5 和 0.7 时，均方差损失函数的梯度绝对值  $|J_M|$  和二元交叉熵损失函数的梯度绝对值  $|J_S|$  关于预测值  $a$  的函数曲线图。所有曲线图中横坐标均为  $a$ ，纵坐标为梯度值。

定地反映出预测值与真实值之间的差异的设计原则，二元交叉熵损失函数相比而言是更好的选择。同时，本文的实验也证明了以二元交叉熵作为损失函数的训练效果要优于均方差损失。

### 3.2.4 任务融合

YIPS 路径规划网络与目前主流的目标检测网络的设计是相似的，都是在针对图像分类任务的主干网络上嫁接针对其他特定任务（路径规划或目标检测等）的头部网络，通过迁移图像分类任务的知识以更好的满足特定任务需求。这样的网络设计可以带来的一个好处就是可以通过共享同一主干网络来实现路径规划任务和目标检测任务的融合，即如图3-8所示的任务融合架构。进行任务融合的好处在于通过整合目标检测网络和路径规划网络可以一定程度上减少自动驾驶系统对存储资源以及 CPU 和 GPU 等算力资源的需求。自动驾驶系统中存在许多基于深度学习的功能单元，尤其是在感知和决策子系统中，这些单元往往拥有不同的网络结构，从而导致系统中存在大量的神经元，要实时运行这些神经元对系统的算力要求很高，进一步提高了系统的实现成本。通过将不同任务的神经网络的共通部分进行整合，将能有效减少系统内神经元的数量，从而降低对算力和存储资源的需求 [24]。因此，通过融合路径规划任务和目标检测任务，可以减少在自动驾驶系统中引入 YIPS 路径规划网络时带来的额外开销。

将目标检测任务和路径规划任务进行融合的一种直接的方法是，用 Fine-tuning 方法同时在目标检测和路径规划数据集上训练已共享主干的目标检测网

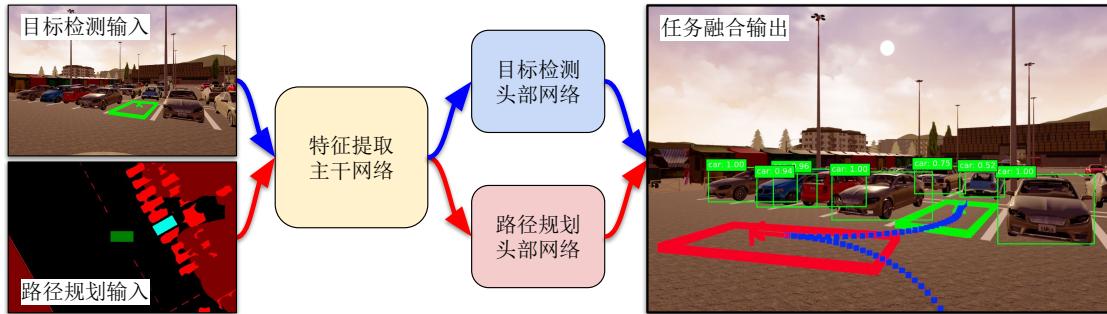


图 3-8 目标检测任务和路径规划任务的任务融合架构。其中，在任务融合输出中，地面上的红色矩形和箭头代表 YIPS 网络预测的采样点，蓝色虚线为 YIPS 网络规划的路径，绿色矩形和箭头代表目标泊车位置。框出车辆的绿色矩形代表目标检测算法的输出。

络和路径规划网络，这种方法虽然可以同时针对两个任务对主干网络的权重进行微调，从而获取最佳权重，但是在实际操作中，要均衡两种任务的数据集大小是相当困难的，而且要训练针对不同任务的大型网络也是相当耗时的。考虑到路径规划网络的泛化误差可以利用采样规划算法进行优化和校正，而目标检测网络的预测校正相较而言更困难，因此，两个任务对预测精度的要求是不一样的，应该优先考虑保证目标检测网络的预测性能。

因此本文提出直接将目标检测网络的主干作为 YIPS 网络的主干，且 YIPS 网络不再对主干网络进行微调，仅训练路径规划头部网络的任务融合方法，本文中称为直接融合方法。直接融合方法能在不破坏现有目标检测网络的情况下实现其与 YIPS 路径规划网络的融合，在最大程度保障目标检测网络性能的同时以更低的成本将 YIPS 路径规划网络引入已有的自动驾驶系统。直接融合方法虽然会因为限制 YIPS 网络对主干网络的微调而一定程度上降低了 YIPS 学习能力，但是本文的实验表明，这种限制所导致的学习能力下降是有限的，更不会导致网络训练失效，同时由于二阶段规划方法在第二阶段会对 YIPS 规划的初始路径进行优化，可以进一步消除这种限制带来的影响。

### 3.3 第二阶段：SO-RRT\* 路径优化算法

#### 3.3.1 采样优化算法设计

由于 YIPS 神经网络存在泛化误差，因此无法保证其规划的初始泊车路径都是无碰撞且可行的，同时即便 YIPS 规划的初始路径直接可用，其路径质量仍有

继续提高的空间。因此在本文提出的二阶段泊车规划方法的第二阶段，利用基于采样的路径规划算法对 YIPS 网络规划的初始路径进行优化。同时，因为在第二阶段中，算法的主要任务是优化路径，而非探索配置空间以寻找可行路径，因此为了最大化优化效果，本文基于 Bi-RRT\* 算法，提出了基于采样的路径优化方法 SO-RRT\*。SO-RRT\* 的路径优化流程如图3-1右框所示，SO-RRT\* 将 YIPS 网络预测的采样点序列作为采样启发，通过 YIPS 网络预测误差的最佳拟合高斯模型来偏置采样，以寻找采样点附近区域中的更优甚至最优的采样点，从而获取更优的采样点序列，进而生成更优的泊车路径。

---

**算法 5 SO-RRT\*( $\mathcal{P}, \mathcal{X}_{start}, \mathcal{X}_{goal}, \mathcal{W}, \mathcal{F}$ )**


---

```

1:  $T_s \leftarrow (\mathcal{V}_s \leftarrow \{\mathcal{X}_{start}\}, \mathcal{E}_s \leftarrow \emptyset), T_g \leftarrow (\mathcal{V}_g \leftarrow \{\mathcal{X}_{goal}\}, \mathcal{E}_g \leftarrow \emptyset)$ 
2:  $T_a \leftarrow T_s, T_b \leftarrow T_g$ 
3:  $c_{best} \leftarrow \infty, \mathcal{P}_{best} \leftarrow \emptyset$ 
4: for  $i$  in  $1 \dots, N$  do
5:    $\mathcal{X}_{rand} \leftarrow \text{SampleFromPath}(\mathcal{P}, \mathcal{W})$ 
6:    $\mathcal{X}_{parent} \leftarrow \text{CostLeast}(\mathcal{V}_a, \mathcal{X}_{rand}, \mathcal{W})$ 
7:   if  $\text{Cost}(\mathcal{X}_{nearest}^*) + \mathcal{D}(\mathcal{X}_{nearest}^*, \mathcal{X}_{new}) + \text{CostToGo}(\mathcal{X}_{new}) < c_{best}$  then
8:      $\mathcal{V}_a \leftarrow \mathcal{V}_a \cup \{\mathcal{X}_{new}\}, \mathcal{E}_a \leftarrow \mathcal{E}_a \cup \{(\mathcal{X}_{nearest}^*, \mathcal{X}_{new})\}$ 
9:      $\text{Rewire}(\{\mathcal{X}\}_{near}, \mathcal{X}_{new}, \mathcal{E}_a, \mathcal{F}, \mathcal{W})$ 
10:     $\text{ConnectGraphs}^*(c_{best}, \mathcal{P}_{best}, \mathcal{V}_b, \mathcal{X}_{new}, \mathcal{F}, \mathcal{W})$ 
11:     $\text{BranchAndBound}(T_a, T_b)$ 
12:   end if
13:    $\text{SwapTrees}(T_a, T_b)$ 
14: end for
15: return  $\mathcal{P}_{best}$ 

```

---

SO-RRT\* 的算法伪代码如算法5所示，相比于 Bi-RRT\*(如算法1所示) 其主要的拓展有三处，分别是行5、6和10。行5处的拓展的作用是实现以 YIPS 网络预测的采样点为基准的高斯偏置采样，该拓展处所使用的函数 SampleFromPath 的伪代码如算法6所示， SampleFromPath 函数首先利用子函数 SelectSample 顺序地从采样点序列中获取一采样点作为采样基准点  $(x, y, \phi)$ ，然后循环地利用预设的三

---

**算法 6** SampleFromPath( $\mathcal{P}, \mathcal{W}$ )

---

```

1:  $(x, y, \theta) \leftarrow \text{SelectSample}(\mathcal{P})$ 
2: loop
3:    $\Delta x \leftarrow \mathcal{N}_x(\mu_x, \sigma_x)$ ,  $\Delta y \leftarrow \mathcal{N}_y(\mu_y, \sigma_y)$ ,  $\Delta \theta \leftarrow \mathcal{N}_\theta(\mu_\theta, \sigma_\theta)$ 
4:    $X_{rand} \leftarrow (x, y, \theta) + (\Delta x, \Delta y, \Delta \theta)$ 
5:   if CollisionFree( $X_{rand}, \mathcal{W}$ ) then
6:     break
7:   end if
8: end loop
9: return  $X_{rand}$ 

```

---

**算法 7** CostLeast( $\mathcal{V}_a, X_{rand}, \mathcal{W}$ )

---

```

1:  $X_{parent} \leftarrow \emptyset$ ,  $c_{best} \leftarrow \infty$ 
2: for  $X \in \mathcal{V}_a$  do
3:    $cost \leftarrow \text{Cost}(X) + \mathcal{D}(X, X_{rand})$ 
4:   if  $cost < c_{best} \wedge \text{CollisionFree}(X, X_{rand}, \mathcal{W})$  then
5:      $X_{parent} \leftarrow X$ ,  $c_{best} \leftarrow cost$ 
6:   end if
7: end for
8: return  $X_{parent}$ 

```

---

**算法 8** ConnectGraphs\*( $c_{best}, \mathcal{P}_{best}, \mathcal{V}_b, X_{new}, \mathcal{F}, \mathcal{W}$ )

---

```

1:  $X_{nearest} \leftarrow \text{Nearest}(\mathcal{V}_b, X_{new})$ 
2: if  $\text{Cost}(X_{new}) + \mathcal{D}(X_{new}, X) + \text{Cost}(X) < c_{best}$  then
3:   if CollisionFree( $X_{new}, X, \mathcal{W}$ ) then
4:      $c_{best} \leftarrow \text{Cost}(X_{new}) + \mathcal{D}(X_{new}, X) + \text{Cost}(X)$ 
5:      $\mathcal{P}_{best} \leftarrow \text{GeneratePath}(X_{new}, X)$ 
6:   end if
7: end if

```

---

个高斯分布分别计算在  $x$ 、 $y$  和  $\phi$  维上的随机偏置值，再将偏置值叠加到基准采样点上以获得偏置采样点  $X_{rand}$ ，如果  $X_{rand}$  经过函数 CollisionFree 的检查后确认为自由配置点，则退出循环并返回  $X_{rand}$ ，否则继续循环直至获取无碰撞偏置采样点。

行6处的拓展则是为了优化树的扩展过程，加快算法收敛到最优解的速度。函数 CostLeast 的伪代码如算法7所示，函数会查询树  $T_a$  的顶点集  $\mathcal{V}_a$  中的所有顶点，并返回其中令采样点  $X_{rand}$  的  $cost$  最小且与  $X_{rand}$  间存在无碰撞路径的顶点  $X_{parent}$  作为最优父顶点，顶点的  $cost$  即根节点到该顶点的路径长度。由于 CostLeast 函数通过搜索整棵树得以获取全局最优父顶点，因此相比 RRT\* 和 Bi-RRT\* 仅获取树中局部最优父顶点的做法，能在保证算法的渐进最优属性的同时通过扩展更优的子路径来加快规划算法收敛的速度。虽然搜索整棵树会导致算法的时间复杂度增加，但由于 SO-RRT\* 主要负责优化而非搜索路径，所以由此带来配置空间探索效率的下降是可以接受的，并不影响二阶段泊车规划方法整体的路径规划效率，同时，由于函数 BranchAndBound 对树的修剪作用，极大的减少了树的顶点数量，从而使得搜索整棵树带来的算法的时间复杂度的增加十分有限，保证了 SO-RRT\* 的路径优化速度。

行10处扩展的目的在于优化两棵树  $T_a$  与  $T_b$  之间的连接过程，增加树的连接可能带来的路径优化收益，进一步提高算法收敛速度。函数 ConnectGraphs\* 的伪代码如算法8所示，其功能与 Bi-RRT\* 中的函数 ConnectGraphs 的功能基本相同，即检查通过偏置采样点连接另一棵树  $T_b$  是否会产生比当前已获得的路径更优的路径，不同之处在于 ConnectGraphs\* 函数会查询树  $T_b$  中所有的顶点，以获取树  $T_b$  中离偏置采样点  $X_{rand}$  最近的顶点，即全局最优的可能连接点  $X_{connect}$ ，而非如 ConnectGraphs 那样获取局部最优连接点，通过获取更优的子路径可以加速算法收敛。

此外，在 SO-RRT\* 中，使用函数 CostToGo( $\bullet$ ) 求解给定配置到目标配置的路径长度时，函数 CostToGo( $\bullet$ ) 会利用 Reed-Shepp 曲线进行求解，以获取两配置间符合车辆运动学约束的最短路径。通过使给定配置到目标配置的启发式距离更接近车辆真实行驶距离，进一步提高规划路径的质量，同时可以降低将不必要的采样点加入树的可能性，约束树的尺寸。

### 3.3.2 YIPS 预测误差建模

SO-RRT\* 算法使用高斯偏置方法对基准采样点进行偏置采样，其优化 YIPS 网络规划的初始路径的效率很大程度上取决于偏置采样对初始采样点的优化效率，而偏置采样的方式取决于预设的高斯分布，因此要提高 SO-RRT\* 的路径优化性能，更合理的设置  $x$ 、 $y$  和  $\phi$  三个维度上的高斯偏置分布是必不可少的。YIPS 网络的泛化误差主要由两部分组成：采样点序列长度预测误差和采样点配置预测误差，高斯偏置采样所能校正的主要是后者，对于前者则可以通过在配置空间中均匀采样来克服。采样点配置预测误差是由预测配置  $(x, y, \phi)$  与标签配置  $(\hat{x}, \hat{y}, \hat{\phi})$  间的差值  $(\Delta x = \hat{x} - x, \Delta y = \hat{y} - y, \Delta \phi = \hat{\phi} - \phi)$  定义的，

独立考虑  $\phi$  维上的误差  $\Delta\phi$ 。首先所有可能的误差值可以构成样本空间  $\Omega_\phi = \{\Delta\phi | \Delta\phi = i, i \in [0, 2\pi]\}$ ，该空间显然不为空。其次，对于定义在  $\Omega_\phi$  空间上的事件集  $\mathcal{F} = \{A_i | A_{ij} = i \leq \Delta\phi \leq j; i, j \in [0, 2\pi]; i <= j\}$ ，任意事件  $A_{ij} = i \leq \Delta\phi \leq j$  及其补集  $A_{ij}^c = \{0 \leq \Delta\phi < i \text{ and } j \leq \Delta\phi < 2\pi\}$  均包含在  $\mathcal{F}$  中，且任意事件的组合  $\bigcup_{i=1}^{\infty} A_i$  也包含在  $\mathcal{F}$  中，因此事件集  $\mathcal{F}$  是定义在  $\Omega_\phi$  上的可度量集。再次，在事件集  $\mathcal{F}$  上定义函数  $P : \mathcal{F} \rightarrow [0, 1]$ ，则对任意两事件  $A_{ij}$  和  $A_{jk}$ ， $i \leq j \leq k$ ，有  $P(A_{ij} + A_{jk}) = P(A_{ij}) + P(A_{jk}) = P(A_{ik})$ ，且  $P(\Omega_\phi) = 1$ ，故  $P$  为事件集  $\mathcal{F}$  上的概率度量，也即累计分布函数。因此误差  $\Delta\phi$  实际上可以看作是一个随机变量  $\Phi : \Omega_\phi \rightarrow \mathcal{F}$ 。同理， $x$  和  $y$  维上的误差也可视为随机变量  $X : \Omega_\phi \rightarrow \mathcal{F}$  和  $Y : \Omega_\phi \rightarrow \mathcal{F}$ 。

通过拟合随机变量  $X$ 、 $Y$  和  $\Phi$  的分布将有利于对预测误差进行预判从而提高对误差的校正能力。本文提出利用高斯分布对随机变量的分布进行拟合，先获取分布的最佳拟合高斯分布，再用最佳拟合高斯分布去指导 SO-RRT\* 的偏置采样过程，以实现针对 YIPS 网络预测误差的偏置采样，提高偏置采样的优化效率。使用高斯分布的原因在于高斯分布的广泛性且易于应用和分析，同时，在采样算法的启发式中，高斯偏置采样启发也是相对高效易用且成熟的启发方式。

对随机变量  $X$ 、 $Y$  和  $\Phi$  的分布进行最佳高斯拟合的过程如图3-9所示，首先以标准高斯分布的分位数为横坐标，以随机变量值为纵坐标将未知分布可视化到概率图上，这种概率图也称为 P-P 图，通过概率图可以直观看出分布的高斯性，

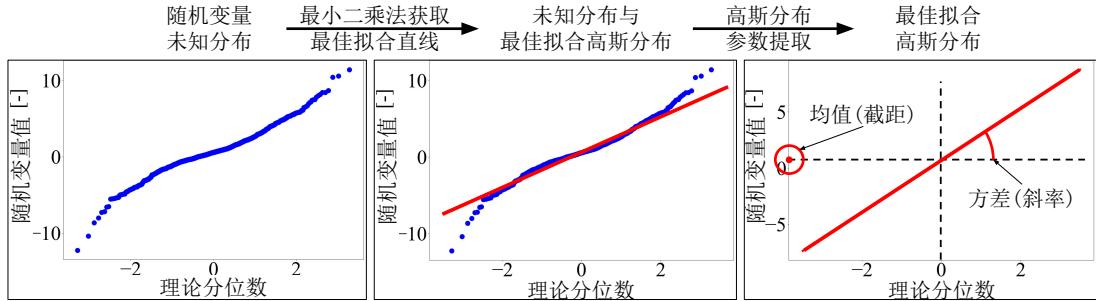


图 3-9 对神经网络预测误差的分布进行高斯拟合的过程。

即未知分布与高斯分布之间的相似性，分布的高斯性越强，则分布在概率图上越接近于直线。然后基于概率图，利用最小二乘法回归分布的最佳拟合直线，即最佳拟合高斯分布。最后提取最佳拟合直线的截距和斜率作为最佳拟合高斯分布的均值和方差。

### 3.4 泊车规划数据集制作

为了充分训练 YIPS 路径规划网络，生成大量细节丰富的泊车场景作为训练样本是十分必要的，同时逼真的泊车场景也是验证泊车规划算法性能的重要前提。根据1.2小节对不同自动驾驶模拟器的评估，本文选择场景丰富度相对更高的 CARLA[25] 模拟器作为制作泊车规划场景的仿真平台。

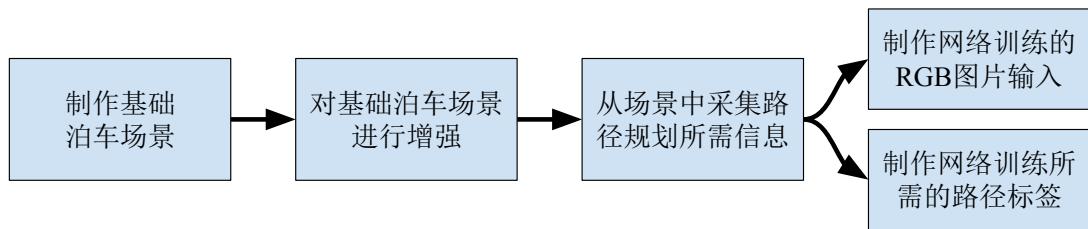


图 3-10 泊车规划场景数据集的制作流程图。

数据集的制作流程如图3-10所示。在制作基础泊车场景阶段，从 CARLA 模拟器 (0.9.6 版本) 已有 7 张地图中选取 87 个泊车场景作为基础场景，每个场景包含三个基本信息：车辆起始配置、目标配置和障碍物分布，这三个信息与定义规划场景的三元组  $(X_{start}, X_{goal}, \mathcal{W})$  中的元素一一对应。87 个基础泊车场景主要可分为四类：规则停车场泊车、规则路边停车区域泊车、半规则乡村道路区域泊车和无规则开放区域泊车，四类场景的结构化程度依次减弱。图3-11展示了这四

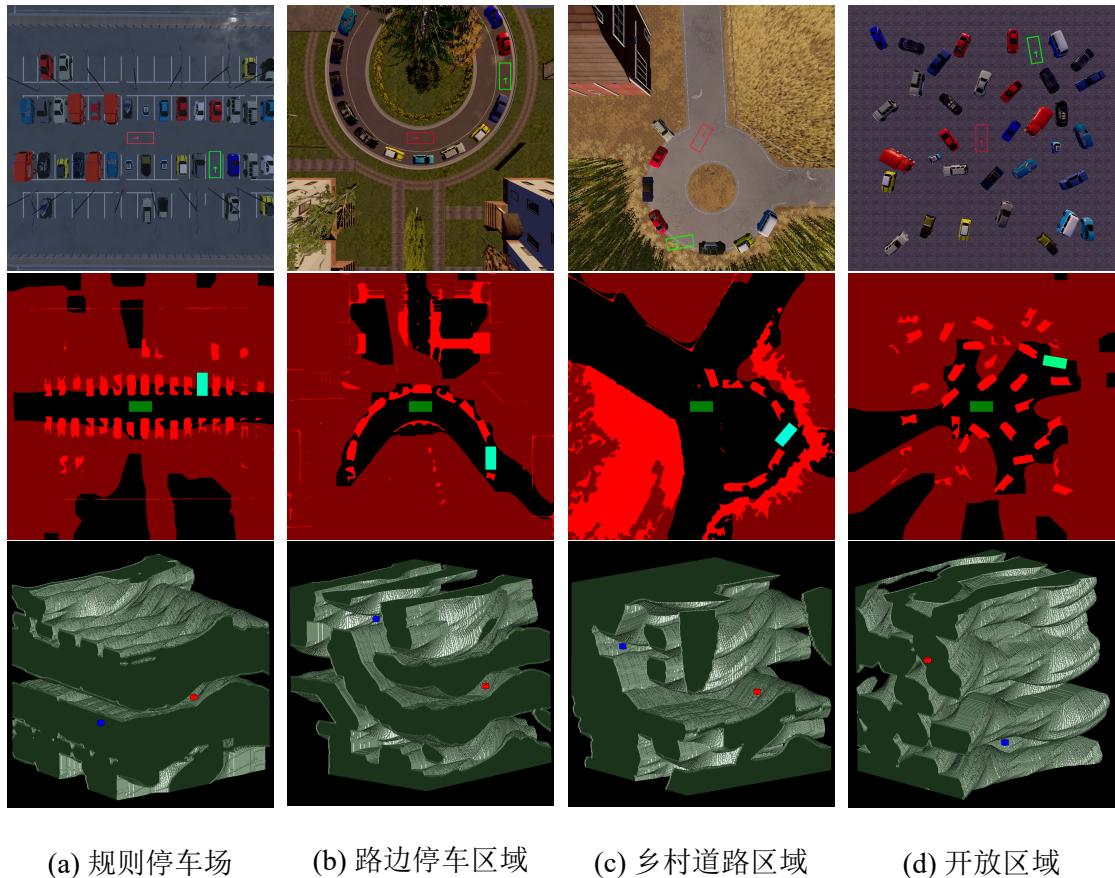
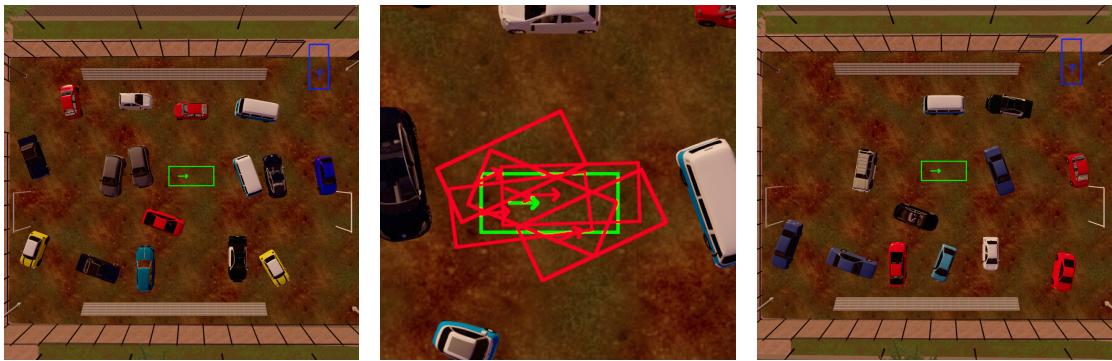


图 3-11 四类典型基础泊车场景的示例图以及对应的编码 RGB 图片和配置空间。其中，第一列为泊车场景示例图，从左往右依次为：规则停车场泊车、规则路边停车区域泊车、半规则乡村道路区域泊车和无规则开放区域泊车。第二列为依次对应的编码了起始配置、目标配置和占用栅格图的 RGB 图片，分辨率为  $600 \times 600$ ，图中绿色和青蓝色矩形表示起始和目标配置，红色、深红色和黑色像素区域代表工作空间中的障碍、未知和可行区域。第三列为依次对应的配置空间，截取配置空间中大小为  $30 \text{ m} \times 30 \text{ m} \times 2\pi$  的子空间，以  $300 \times 300 \times 300$  分辨率显示，其中红点和蓝点分别表示起始和配置（已放大 8 倍以可视化），绿色三维流形代表配置空间  $C_{obs}$ 。

类典型场景的示例图以及对应的配置空间和编码 RGB 图片。从四类场景的 RGB 图片可以看出，停车场场景和路边停车区域场景的可行区域（图中黑色区域）是相对比较规则的，同时，由于这两类场景的泊车区域一般仅供结构类似的四轮车辆使用，因此泊车区域附近的障碍物也比较规则。相比而言，乡村道路区域场景的可行区域虽然也比较规则，但是由于乡村道路非结构化的特点，停车区域附近有很多植被之类的不规则障碍物，场景的不规则性更强，需要神经网络过滤的干扰信息更多。而开放区域泊车场景相对于前三类场景，可行区域是完全不规则



(a) 基础泊车场景示例。绿色 (b) 起始配置增强示例, 红色 (c) 障碍物分布增强示例。对和蓝色方框与箭头分别表示 方框与箭头表示随机偏置获 障碍物车辆的种类和数量进起始和目标配置。 (b) 方框与箭头表示随机偏置获 行随机改变以实现增强。

图 3-12 基础泊车场景的增强示例图。

表 3-1 本文中泊车规划的目标车辆林肯 MKZ2017 的参数表。

长度	宽度	轴距	轮距	转向半径	转向曲率
4.925 m	2.116 m	2.850 m	1.575 m	5.003 m	$0.200 \text{ m}^{-1}$

的, 而且由于泊车区域停靠的车辆摆放凌乱, 使得传感器所识别到的障碍物也显得形态各异, 因此场景规则性最弱。从配置空间看, 结构相似的场景的配置空间也是相似的, 如图3-11所示的同样是圆形泊车区域的第二类和第三类场景, 其配置空间都具有相似的圆弧结构。而结构相似度不高的场景, 如图示的第一类和第四类场景, 其配置空间的差异也是十分明显的。

对基础泊车场景的增强分为两部分, 第一部分为对起始配置进行增强, 在确保起始配置无碰撞的情况下对起始配置进行随机偏置, 从而生成更多的训练数据, 如图3-12b所示, 对起始配置增强后获取额外的五个起始配置。第二部分为对障碍物分布进行增强, 如图3-12c所示, 在基础场景中会预设一个允许放置障碍物车辆的配置点组成的集合, 通过随机从集合中选择配置点, 并为配置点随机选择障碍物车辆的种类, 来实现障碍物分布的增强。在本文的实现中, 对每个基础场景的起始配置随机增强 16 次, 障碍物分布随机增强 10 次, 从而使泊车场景的数量增加了 160 倍。

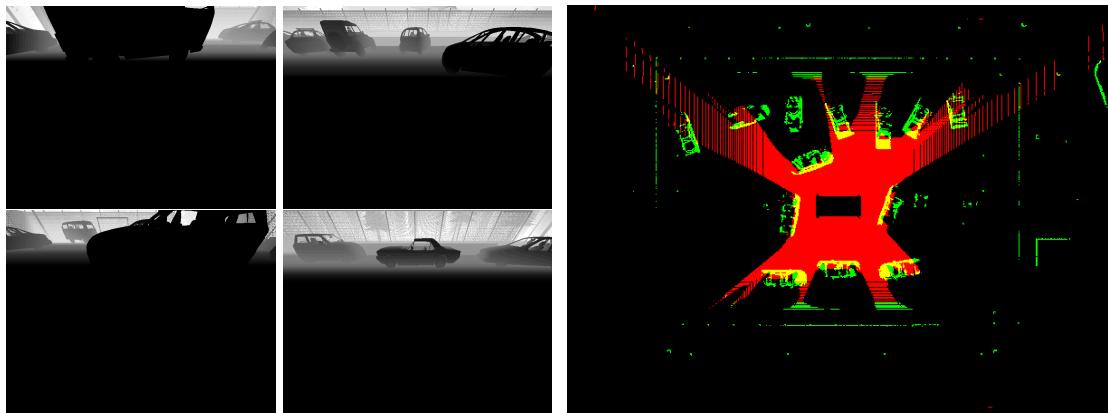
从场景中采集的规划所需信息包括起始配置、目标配置和占用栅格图, 其中



图 3-13 从场景中采集数据时所用的仿真车辆以及仿真深度相机的安装位置示意图。其中，左图为仿真车辆示意图，车辆型号为林肯 MKZ2017，右图为仿真深度相机的安装位置示意图，绿色框内的文字为前后左右四台深度相机的安装位置和姿态信息。方括号“[]”内为车身局部坐标系中相机的欧拉角，依次为绕 X、Y 和 Z 轴的旋转角，圆括号“()”内为车身局部坐标系中相机的位置坐标，依次为 X、Y 和 Z 轴。

起始和目标配置是易于获取的，而占用栅格图的制作相对复杂。为了获取精细的占用栅格图，需要在仿真环境中设置传感器阵列对传感数据进行融合，为了加速数据制作过程，本文采用仿真深度相机来实现传感器阵列的融合效果，实现快速且精细的占用栅格图生成。用于搭载深度相机传感器的仿真车辆和传感器安装位置如图3-13所示。为了制作占用栅格图，首先要通过深度相机获取车辆四个方向的深度图像，如图3-14a所示，然后将图像转换为三维空间中的点云并将四个传感器的点云进行坐标系融合，如图3-14b所示，再从融合的点云中过滤掉属于路面的非障碍物部分，最后将过滤后的点云投影到 XY 平面，利用数学形态学的闭操作处理后最终生成占用栅格图，如图3-15所示。进行闭操作的原因是为了减少图像中的散点，使障碍物特征更明显。与利用单一 64 线激光雷达生成的占用栅格图相比，在保证生成速度的情况下，利用深度图制作的栅格图更精细，只要是传感器能看见的障碍物车辆，都能生成更为完整的车辆轮廓。

要制作用于网络训练的 RGB 图片输入是相对简单的，只需要将起始和目标配置下的车辆姿态以一定的几何形状绘制到 RGB 图片上即可，在本文的实现中，将车辆以矩形形状绘制在 RGB 图片的 G 通道中，矩形长宽为车辆的实际长宽值，起始和目标配置分别对应于像素值 255 和 127，将占用栅格图置于 R 通道中，同时因为对于朝向角相差 180 度的两个位置相同的配置无法根据 RGB 图中的矩形



(a) 在3-12a所示场景中，前后左右四个位置  
的深度相机传感器的所采集的深度图，以灰  
度图片格式显示，已经过直方图均衡以增强  
可视化程度。

(b) 在3-12a所示场景中，根据深度相机传感  
器数据生成的点云的平面示意图，其中红色  
区域表示属于路面等非障碍物的点云，绿色  
像素表示属于障碍物车辆的点云。

图 3-14 深度相机传感器生成的深度图及基于深度图生成的点云。

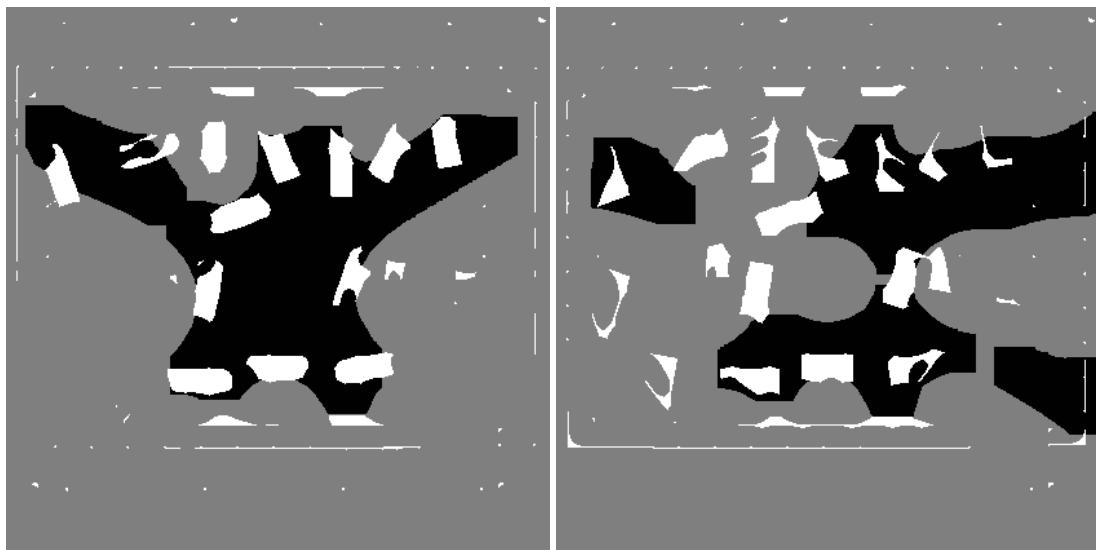


图 3-15 在3-12a所示场景中，分别基于深度相机传感器和 64 线激光雷达制作的占用栅格图。其中白色、灰色和黑色像素分别表示障碍物区域、未知区域，和可行区域，占用栅格图的尺寸为  $60 \text{ m} \times 60 \text{ m}$ ，分辨率为  $0.1 \text{ m}/\text{cell}$ 。左侧为基于深度相机传感器制作，右侧为基于中央放置的单一 64 线激光雷达传感器制作。

的方向进行区别，所以，将配置的朝向角映射到  $[0, 255]$  后置于 B 通道中。所得  
到的 RGB 图片如图3-11中第二列子图所示。

利用基于采样的规划算法如 RRT\*、Bi-RRT\* 等为每个泊车场景生成最优采样点序列作为网络训练的标签则相对困难，因为采样规划算法具有渐进优化和概率完备的特点，所以如果采样时间不足，会导致生成的路径质量不高甚至无法生成路径。而由于场景数量较多，又不允许在每个场景上花费太多时间。因此本文采用 OSE 启发式来加速采样规划算法的收敛，提高生成的标签路径的质量。此外，考虑到 OSE 启发式并不能有效适应所有场景，因此同时利用 GBS 启发的采样规划算法作为对照，从两种方法中选取更优的结果作为标签路径。OSE 启发式和 GBS 启发式的算法伪代码分别如算法4和3所示。另外，本文选择更适合于泊车规划的 Bi-RRT\* 算法作为采样规划算法，其伪代码如算法1所示，规划算法的目标车辆为如图3-13所示的林肯 MKZ2017，其车辆相关参数如表3-1所示，OSE 启发和 GBS 启发的采样规划算法的采样次数限制分别为每个场景 500 次和 1000 次。最终经过筛选后获得由 10962 组不同泊车场景输入和对应采样点序列标签组成的泊车规划数据集。

### 3.5 本章总结

在本章中，首先介绍本文提出的二阶段泊车运动规划方法的系统框架，方法分为两个阶段，在第一阶段，利用 YIPS 路径规划神经网络规划初始泊车路径，在第二阶段，利用 SO-RRT\* 路径优化算法进一步优化初始路径，提高方法规划的泊车路径的质量。然后对路径规划任务的可学习性以及 YIPS 神经网络的输入和输出进行了分析，并在此基础上介绍了 YIPS 网络的设计方法和网络架构，YIPS 神经网络以编码了起始配置、目标配置和占用栅格图的 RGB 图片为输入，为泊车车辆预测采样点序列，再通过 Reed-Shepp 曲线连接采样点序列最终生成初始泊车路径。最后详细介绍了 SO-RRT\* 路径优化算法以及 YIPS 网络泛化误差的最佳拟合高斯建模方法，SO-RRT\* 以 YIPS 网络预测的采样点序列为启发，通过高斯偏置采样优化采样点序列，从而实现对初始路径的优化。而为了提高 SO-RRT\* 的优化能力，基于 P-P 图对 YIPS 网络预测误差进行最佳高斯拟合建模，利用拟合高斯模型去指导 SO-RRT\* 偏置采样过程以提高偏置采样效率。此外，本章中还介绍了用于 YIPS 网络训练和规划算法验证的泊车规划数据集的制作流程。



## 第4章 神经网络评估及验证

本章首先对二阶段泊车规划方法的核心组成部分 YIPS 神经网络的预测性能进行评估，并对配置预测的泛化误差进行高斯性分析和建模。然后对神经网络设计中的关键部分进行了实验验证，主要包括验证网络的输入编码方式、网络训练方法、损失函数和主干网络结构对网络学习能力即训练效果的影响。此外，本章还评估了任务融合的直接融合方法对 YIPS 网络学习能力产生的影响。

### 4.1 实验平台参数

本章中，YIPS 神经网络的所有训练均在性能相当的显卡 NVIDIA Tesla P100 和 NVIDIA Tesla V100 上进行，基于 Tensorflow[51] 的 Keras 深度学习框架，训练集和验证集的样本数量分别是 8221 和 2741，批次大小均是 16。无特别说明时，YIPS 网络的主干网络为 VGG-19[43]，以 Fine-tuning 方法训练 200 个 epoch，采用 cosine 型学习率衰减策略，初始学习率和损失系数分别为  $3e-05$  和  $1+04$ ，优化器为 Adam [52] 且系数  $\beta_1$  和  $\beta_2$  分别为 0.9 和 0.999，采用 L2 正规化且系数为  $1e-4$ ，激活函数 Leaky ReLU 的系数  $\alpha$  为 0.1。

### 4.2 网络预测性能评估

#### 4.2.1 网络预测组成

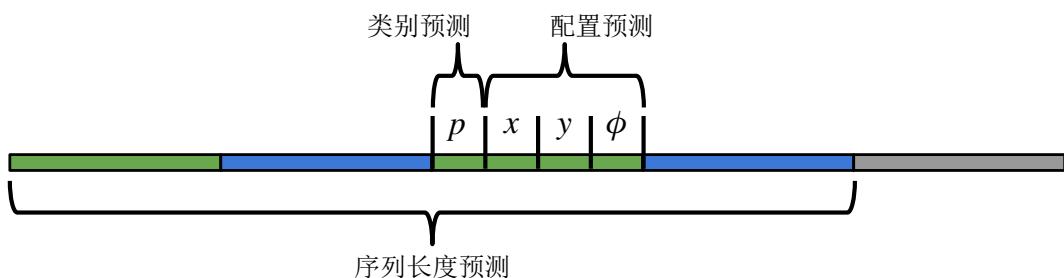


图 4-1 YIPS 路径规划网络预测组成。对于给定预测，其主要由三部分组成：采样点的类别预测、采样点配置预测及采样点序列长度预测，其中序列长度预测又以类别预测为基础，被分类为必要的采样点的总数即是 YIPS 网络预测的采样点序列长度。

如图4-1所示，YIPS 路径规划网络的预测由三个部分组成：采样点类别预测、采样点配置预测和采样点序列长度预测，其中序列长度预测以类别预测为基础，被分类为必要的采样点的总数即是 YIPS 网络预测的采样点序列长度，同时，需要指出的是，预测的序列长度为不考虑起始和目标配置的序列长度，起始和目标配置是给定的，不需要进行预测。序列长度预测和类别预测统称分类预测。

### 4.2.2 分类预测评估

分类预测包括类别预测和序列长度预测。对于类别预测，其预测的置信度可由类别预测值  $p$  直接给出，同时采样点的类别是二元的，只有必要采样点和非必要采样点两类，因此可以利用 PR(Precision-Recall, 精确度-召回率) 曲线和 AP(Average Precision, 平均精确度) 值有效地评估 YIPS 网络的类别预测性能，即网络从规划场景中识别和提取采样点的能力，其中召回率对第二阶段优化而言更关键，如果召回率太低，会导致 SO-RRT\* 能偏置采样的区域太小，从而对路径优化产生负面影响。此外，为了进一步考察 YIPS 网络对无碰撞采样点的分类预测能力，引入衡量采样点碰撞度指标 IOV(Intersection-area Over Vehicle-area)，即在占用栅格图上，车辆占用区域与障碍物区域的重合面积与车辆占用区域面积之比，计算公式如式4-1。通过设置 IOV 的阈值就可以基于碰撞度判断预测的有效性，如可以设置阈值为 0.125，则当采样点的碰撞区域与车辆占用区域之比小于 0.125 时，认为该采样点预测有效，反之无效。阈值为 0，则意味着只有无碰撞的采样点预测才有效。

$$\text{IOV} = \frac{|Veh \cap Obs|}{|Veh|} \quad (4-1)$$

其中， $Veh$  代表构成占用栅格图上车辆所占区域的点集， $Obs$  代表构成占用栅格图上障碍物所占区域的点集。

对于序列长度预测，其以类别预测为基础，预测值等于网络预测的必要类采样点的总数。考虑到在应用中更关注，对于给定规划场景，YIPS 网络的序列长度预测的准确率，因此以正样本占所有样本的比例即准确度 (Accuracy, Acc) 作为衡量序列长度预测性能的主要指标，以符号 Acc 表示。同时评估 YIPS 网络对不同长度的序列的预测准确度，网络对不同长度的序列的预测能力可以一定程度上反映网络对不同难度的规划任务的求解能力，对不同长度序列的预测准确

度以符号  $\text{Acc}^a$  表示,  $a$  为序列长度。

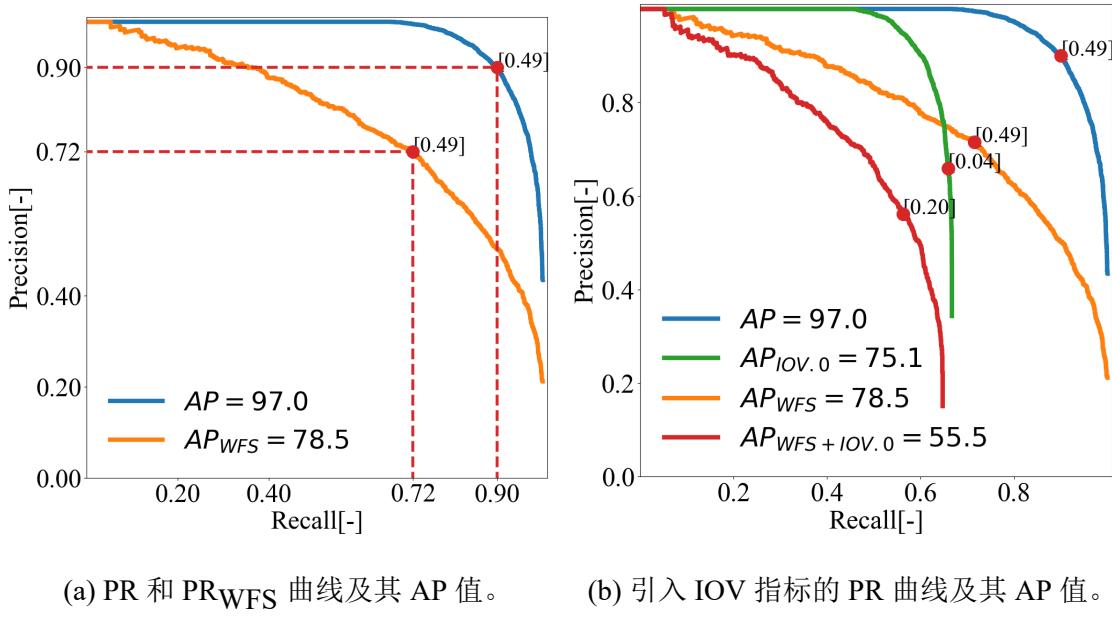


图4-2 YIPS神经网络类别预测的PR曲线及其AP值。图4-2b中,下标 $IOV.a$ 表示 $IOV$ 阈值为 $0.a$ , $IOV.0$ 即 $IOV$ 阈值为0.0。图中红点为PR曲线的平衡点,红点的数字标签如“[0.49]”表示该平衡点对应的分类阈值为0.49。

图4-2a显示了在不考虑 $IOV$ 指标时采样点分类性能的评估结果,图中除了PR曲线以外,还同时绘制了 $PR_{WFS}$ 曲线,即在忽略对序列首个采样点的预测(Without the First Sample, WFS)的情况下PR曲线,之所以不计入对首个采样点的预测,是因为对任意规划场景,如果其起始和目标配置间存在直接可行路径(即无需通过采样规划,仅利用Reeds-Shepp曲线就可以获取的路径),那么该规划场景无需YIPS对其进行规划,所以,凡输入到YIPS网络中的规划场景,则默认采样点序列至少有一个采样点。因此网络对首个采样点的预测精确率会高于其他采样点,通过计算 $PR_{WFS}$ 曲线及其平均精度 $AP_{WFS}$ ,可以更有效评估网络对非默认采样点的识别性能。如图4-2a所示,在计入序列首个采样点时, $AP$ 值相当高,为97.0,而忽略首个采样点后, $AP$ 值降至78.5,由此可见YIPS对首个采样点的预测精度确实高于其余采样点,对首个采样点的计入会一定程度上使类别预测性能的评估结果过于乐观。同时,这也说明了,通过学习YIPS网络也发现了预设序列的首个采样点是恒必要的,因此对首个采样点的预测非常自信而且准确度也很高。除去首个采样点的影响后,YIPS网络对其他采样点的分

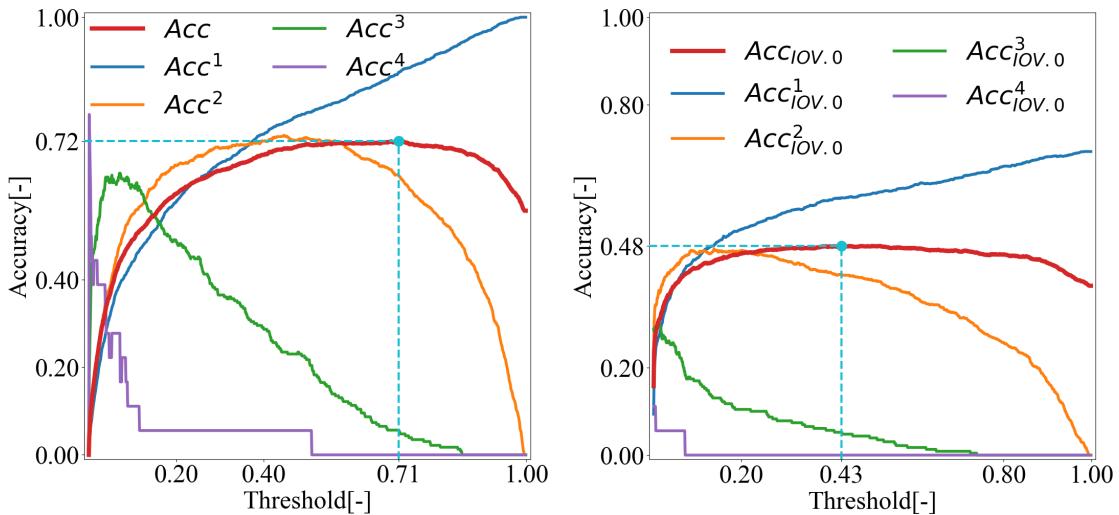
类性能也还是相当高的， $AP_{WFS}$  值接近于 80，说明 YIPS 网络具有相当良好的采样点识别分类能力。

表 4-1 不同 IOV 阈值下，YIPS 神经网络采样点类别预测 AP 值。

IOV 阈值	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875
$AP_{IOV}$	75.1	93.7	95.6	96.4	96.9	97.0	97.0	97.0
$AP_{IOV+WFS}$	55.5	75.0	77.5	78.2	78.5	78.5	78.5	78.5

引入 IOV 指标后的 PR 曲线如4-2b所示，图中的 IOV 阈值为 0，可见，在进一步考虑采样点的碰撞度后，YIPS 网络的类别预测性能有所下降， $AP$  和  $AP_{WFS}$  指标分别下降至 75.1 和 50.6。但由于 YIPS 网络的预测误差可以在第二阶段路径优化中得到校正，因此即便采样点存在一定程度的碰撞，通过校正后也容易得到无碰撞的优化采样点。所以在考虑碰撞度作为采样点有效性指标时，IOV 阈值不宜过低。表4-1显示了不同 IOV 阈值下类别预测的 AP 值，由表可见，当 IOV 阈值大于 0.5 后， $AP_{IOV}$  和  $AP_{WFS+IOV}$  值趋于稳定，均不再增加。由此可见，YIPS 网络预测的采样点的碰撞度 IOV 值不超过 0.625，且绝大部分在 0.5 以下，这提示在第二阶段优化中，对采样点的偏置范围应大于车辆占用区域面积的一半以上，以确保能校正采样点配置预测误差造成的碰撞。

图 4-3a 显示了 YIPS 网络的序列长度预测的准确度-分类阈值曲线，由图可见，随着分类阈值增加，准确度先增后减，并在阈值为 0.71 时取得最大值 72%，此时，采样点的类别预测精确度为 95%，但召回率仅为 85%，而当阈值为 0.49，即为类别预测 PR 曲线平衡点处的分类阈值时，准确度轻微下降至 71%，但召回率增至 90%，所以相比而言以平衡点处的分类阈值来对采样点进行分类能更好的兼顾序列预测性能和采样点类别预测性能。同时，由图可见，网络对不同长度序列的预测准确度差异比较大，其中，对长度为 1 和 2 的序列的预测准确度明显高于长度为 3 和 4 的序列，在总准确度 Acc 达到最大值时，对长度为 3 和 4 的序列的预测准确度  $Acc^3$  和  $Acc^4$  已低于 10%。造成这一现象的原因在于数据集中不同序列长度的样本的占比不平衡，如表4-2所示，序列长度小于等于 2 的样本占了数据集的 90%，因此网络对这两种长度的序列的预测准确度自然更高。而造成数据集不平衡的原因可能在于，数据集中大部分的规划任务都不难，长度小



(a) 序列长度预测准确度与分类阈值曲线。 (b) 序列长度预测 IOV-准确度与分类阈值曲线。

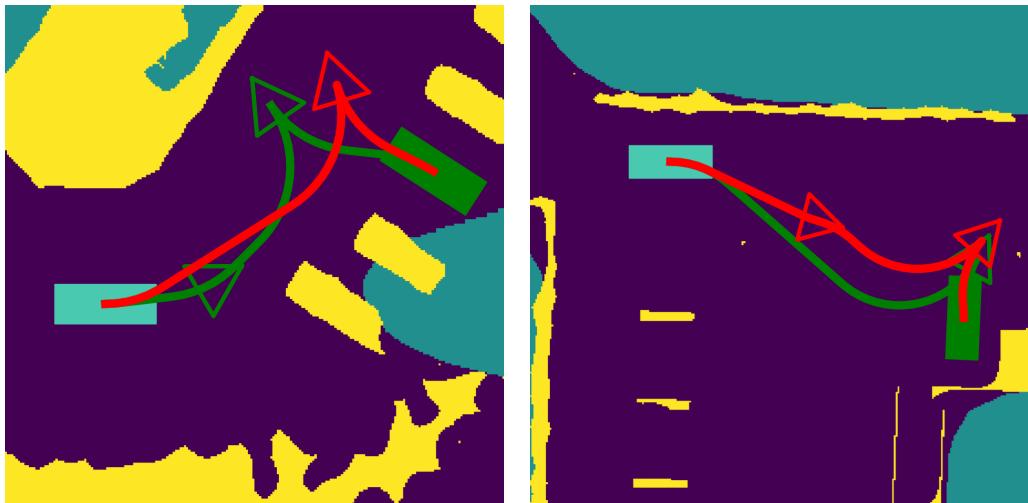
图 4-3 YIPS 神经网络序列长度预测准确度-分类阈值曲线。图中,  $Acc$  表示序列预测准确度 (Accuracy), 无上标表示对所有长度的序列的预测准确度。 $Acc$  的上标  $a$  表示对长度为  $a$  的序列的预测准确度, 下标  $IOV.a$  表示  $IOV$  阈值为  $0.a$ ,  $IOV.0$  即  $IOV$  阈值为  $0.0$ 。青蓝色虚线和青蓝色点标识了准确度的最大值和对应的分类阈值。

表 4-2 数据集中不同序列长度的样本的占比。

序列长度	1	2	3	4
样本比例	55.48 %	34.87 %	8.68 %	0.97 %

于等于 2 的采样点序列足以应付。另外, 序列长度为 3 和 4 的样本的产生也可能是因为用于生成标签路径的采样规划算法在小部分场景中未完全收敛, 如果给予算法更长的规划时间, 序列长度为 3 和 4 的样本的可以转化为长度为 1 和 2 的样本。

图 4-3b 显示了引用  $IOV$  指标对序列长度预测准确度的影响, 由图可见, 在  $IOV$  阈值为 0, 即无碰撞采样点才认为有效的苛刻限制下, 序列长度预测的准确度下降至 48 %, 不同长度的序列的预测准确度也均下降。由此可见, YIPS 网络在预测无碰撞采样点方面表现的比较挣扎。但值得注意的是, 由于 YIPS 网络存在学习能力, 网络会给出有别于对应标签的路径, 如图 4-4 所示, 网络预测的序列长度不同于标签序列长度, 但是路径仍是可行的, 预测路径的形态与标签路径



(a) 预测序列短于标签序列，路径可行。 (b) 预测序列长于标签序列，路径可行。

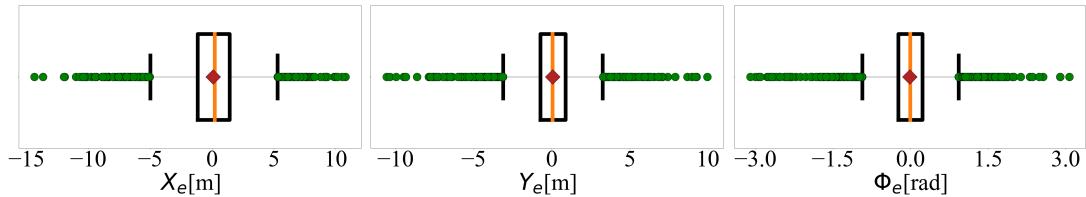
图 4-4 预测序列长度不同于标签序列长度时，预测路径与对应的标签路径的对比。其中，红色曲线表示 YIPS 网络的预测路径，红色三角形表示预测采样点。绿色曲线表示标签路径，绿色三角形表示标签采样点。黄色、青蓝色和紫色像素区域分别表示障碍物、未知和可行区域。

不同但相似，因此仅通过序列长度准确度并不能完全反映出 YIPS 网络的路径规划性能。

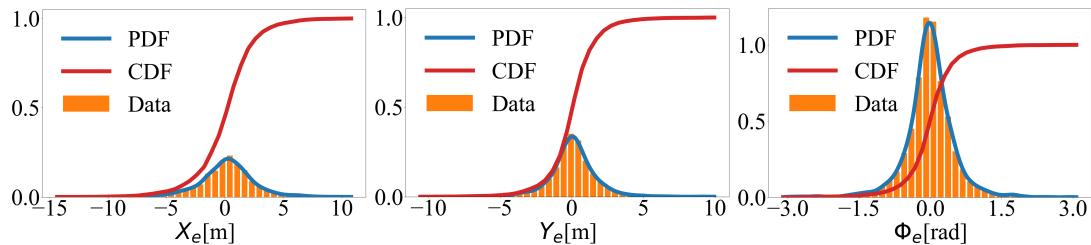
### 4.2.3 配置预测误差分析及建模

对于 YIPS 网络的配置预测性能的评估，本文通过分析其配置预测误差的分布来实现。YIPS 网络预测的采样点配置是由  $x$ 、 $y$  和  $\phi$  三个坐标量组成的元组即  $(x, y, \phi)$ ，根据 3.3.2 节中的分析，每个坐标维度上的误差都可以视为一个独立的随机变量，设  $x$ 、 $y$  和  $\phi$  三个坐标维度上的误差随机变量分别为  $X_e$ 、 $Y_e$  和  $\Phi_e$ ，则 YIPS 网络的配置预测误差可视为随机变量组成的三元组  $(X_e, Y_e, \Phi_e)$ 。测试集上的配置预测误差计算方法为：对于每一预测采样点，如果其对应的标签采样点的类别为必要，即  $\hat{p} = 1$ ，则计入该点的配置预测误差  $(X_e, Y_e, \Phi_e) = (\hat{x} - x, \hat{y} - y, \hat{\phi} - \phi)$ ，反之忽略该采样点的配置预测。

YIPS 网络在测试集上的配置预测误差  $X_e$ 、 $Y_e$  和  $\Phi_e$  的箱线图如图 4-5 所示，其概率密度函数 (PDF) 和累计分布函数 (CDF) 曲线绘制如图 4-6 所示。由箱线图可见， $X_e$ 、 $Y_e$  和  $\Phi_e$  的均值都接近于 0，说明配置预测误差中不存在样本无关的固定误差。对于  $X_e$ ，其方差较大为 6.14 m，但从箱线图可看出约 50 % 左右的误



(a) 均值: 0.09, 方差: 6.14。(b) 均值: 0.05, 方差: 2.92。(c) 均值: -0.01, 方差: 0.27。

图 4-5 YIPS 网络的配置预测误差  $X_e$ 、 $Y_e$  和  $\Phi_e$  的箱线图。

(a) 偏度: -0.29, 峰度: 2.95。(b) 偏度: -0.23, 峰度: 3.82。(c) 偏度: -0.52, 峰度: 5.66。

图 4-6 YIPS 网络的配置预测误差  $X_e$ 、 $Y_e$  和  $\Phi_e$  的概率密度函数 (PDF) 和累计分布函数 (CDF) 曲线。

差值在  $[-1, 1]$  之间，同时绝大部分误差值分布在  $[-5, 5]$  之间，因此方差偏大主要是受较大的异常值影响，整体而言预测误差还是集中在较小的区间  $[-5, 5]$  内，这一点，通过其 PDF 曲线可以直观看出。 $Y_e$  的均值和方差都小于  $X_e$ ，可见 YIPS 对配置坐标  $x$  和  $y$  的预测能力是不同，相比而言，对  $y$  坐标的预测精度更高。而对于  $\Phi_e$ ，其方差仅为 0.27 即 15 度，预测精度是相当高的，从箱线图和 PDF 曲线都可以直观的看出其误差值相当的集中， $[-1.5, 1.5]$  的区间几乎涵盖了所有误差值。此外，从误差的 PDF 和 CDF 可看出，虽然  $X_e$ 、 $Y_e$  和  $\Phi_e$  的分布有所差异，但都跟高斯分布具有很强的相似性。

进一步比较采用不同主干网络时，YIPS 网络的配置预测误差  $X_e$ 、 $Y_e$  和  $\Phi_e$  的分布，如图4-7所示，可见对于不同的主干网络，配置预测误差的分布基本相同，分布相似程度很高，可见配置误差分布的基本形态与网络的配置预测性能无关，预测准确度的差异只会影响分布的峰度，对分布的偏度影响不大，这提示了 YIPS 网络的配置预测误差确实存在相对固定的分布规律，其分布不是完全随机的。

为了进一步考察配置预测误差分布的高斯性，将  $X_e$ 、 $Y_e$  和  $\Phi_e$  的 CDF 曲线

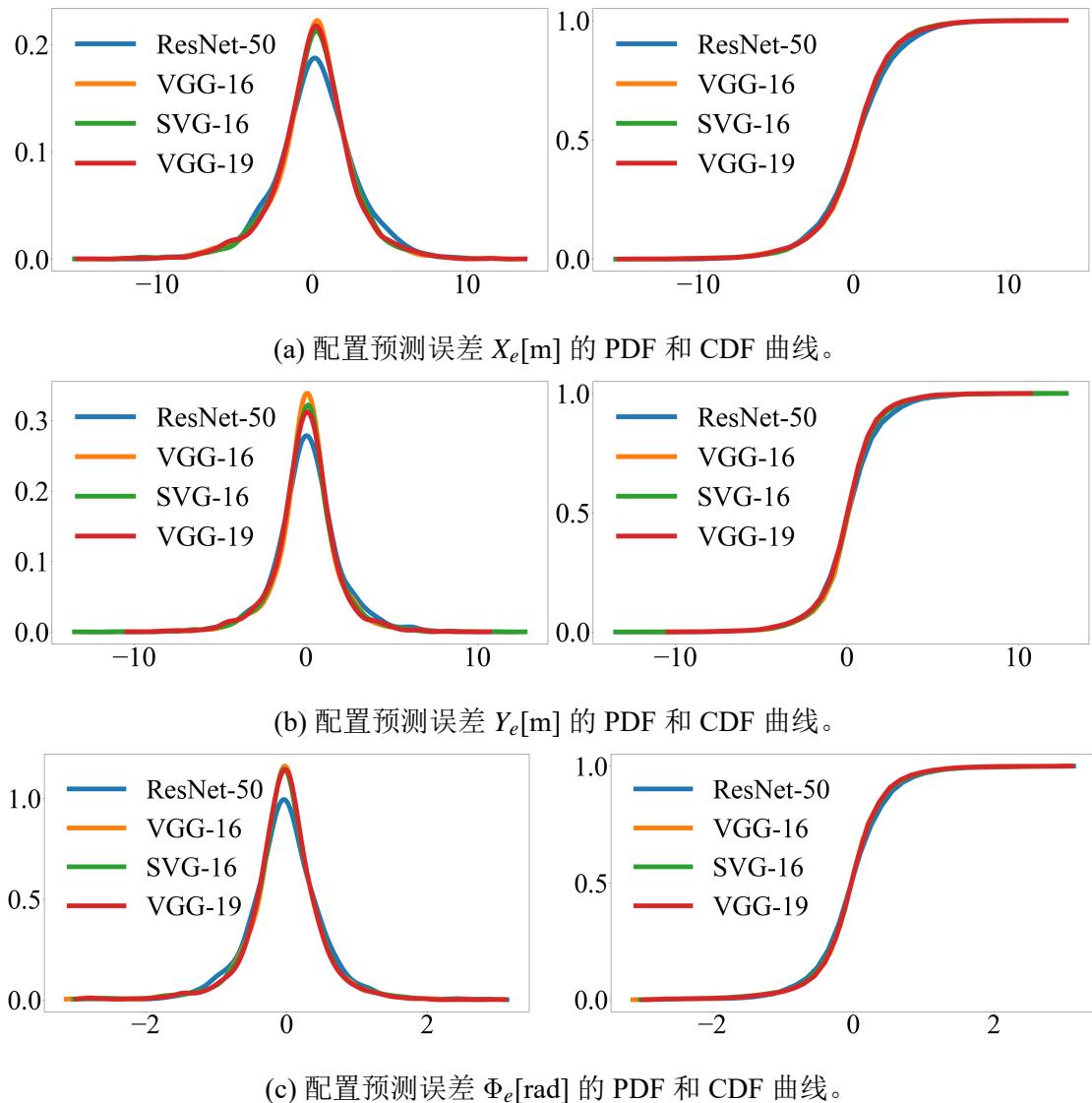


图 4-7 采用不同类型主干网格的 YIPS 网络的配置预测误差  $X_e$ 、 $Y_e$  和  $\Phi_e$  的 PDF 和 CDF 曲线对比。不同主干网络的训练方法参见表4-7。

表示到 P-P 图上，并利用最小二乘法获取其最佳拟合高斯分布 (Best-fit Gaussian Distribution)，如图4-8所示，由图可见，误差  $X_e$ 、 $Y_e$  和  $\Phi_e$  的分布与其最佳拟合高斯分布的吻合程度是比较高的，尤其是在  $[-2, 2]$  区段内，误差分布与 Best-fit 直线基本重合。但在  $[-2, 2]$  区段外，误差分布逐渐偏离 Best-fit 直线，这表明利用高斯分布对误差分布进行拟合时，并不能很好地拟合区间  $[-2, 2]$  外的误差分布，区间外的拟合结果存在较大的偏差。

但表4-3中对配置预测误差大于  $2\sigma$  的路径的无碰撞比例的统计结果却表明，配置误差大于  $2\sigma$  的路径中超过一半的路径是无碰撞的。其中  $\sigma$  为最佳拟合高斯

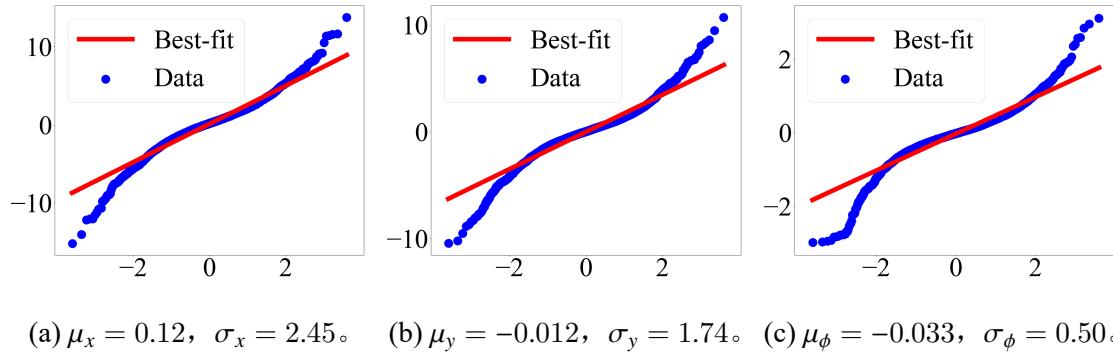


图 4-8 YIPS 网络的配置预测误差及其最佳拟合高斯分布的 P-P 图。P-P 图的横坐标为标准高斯分布的分位数，纵坐标为随机误差值，从左到右分别为  $X_e$ 、 $Y_e$  和  $\Phi_e$ 。 $(\mu_x, \sigma_x)$ 、 $(\mu_y, \sigma_y)$  和  $(\mu_\phi, \sigma_\phi)$  分别为  $X_e$ 、 $Y_e$  和  $\Phi_e$  的最佳拟合高斯分布的参数(均值，方差)。

表 4-3 配置预测误差大于  $2\sigma$  的路径的无碰撞比例。 $(\mu_x, \sigma_x)$ 、 $(\mu_y, \sigma_y)$  和  $(\mu_\phi, \sigma_\phi)$  分别为  $X_e$ 、 $Y_e$  和  $\Phi_e$  的最佳拟合高斯分布的参数(均值，方差)。

配置误差限制条件	无碰撞路径比例
$ X_e - \mu_x  > 2\sigma_x$	45.83 %
$ Y_e - \mu_y  > 2\sigma_y$	50.59 %
$ \Phi_e - \mu_\phi  > 2\sigma_\phi$	51.92 %
$ Y_e - \mu_y  > 2\sigma_y \wedge  \Phi_e - \mu_\phi  > 2\sigma_\phi$	56.52 %
$ X_e - \mu_x  > 2\sigma_x \wedge  Y_e - \mu_y  > 2\sigma_y \wedge  \Phi_e - \mu_\phi  > 2\sigma_\phi$	63.64 %

分布的方差。这提示，并非所有误差较大的路径都是不可行的路径，相反虽然这些预测路径与标签路径差距较大，但是大部分是无碰撞可行路径。图4-9展示了此类配置误差较大的路径与标签路径的对比，由图可见，这些大误差预测路径实际上给出了有别于标签的另一种可行路径。如图4-9a、4-9b和4-9d，YIPS 网络规划出了与标签路径对称的路径，这反映出 YIPS 网络确实具有从训练数据中学习如何规划路径的能力，可以通过学习给出有别于标签路径但仍可行的替代路径。另外，图4-9e和4-9f还反映出即便由于配置预测存在误差而导致预测路径发生碰撞，但由于碰撞度不高，通过第二阶段优化后仍容易获得可行路径。因此即便利用最佳拟合高斯分布对误差进行建模会在区间  $[-2\sigma, 2\sigma]$  外产生较大的拟合偏差，但是较大的配置偏差大概率意味着网络给出了有别于便签的替代路径，因此，不宜将较大的配置误差视为需要大幅校正的较大预测错误，而应视为与区间

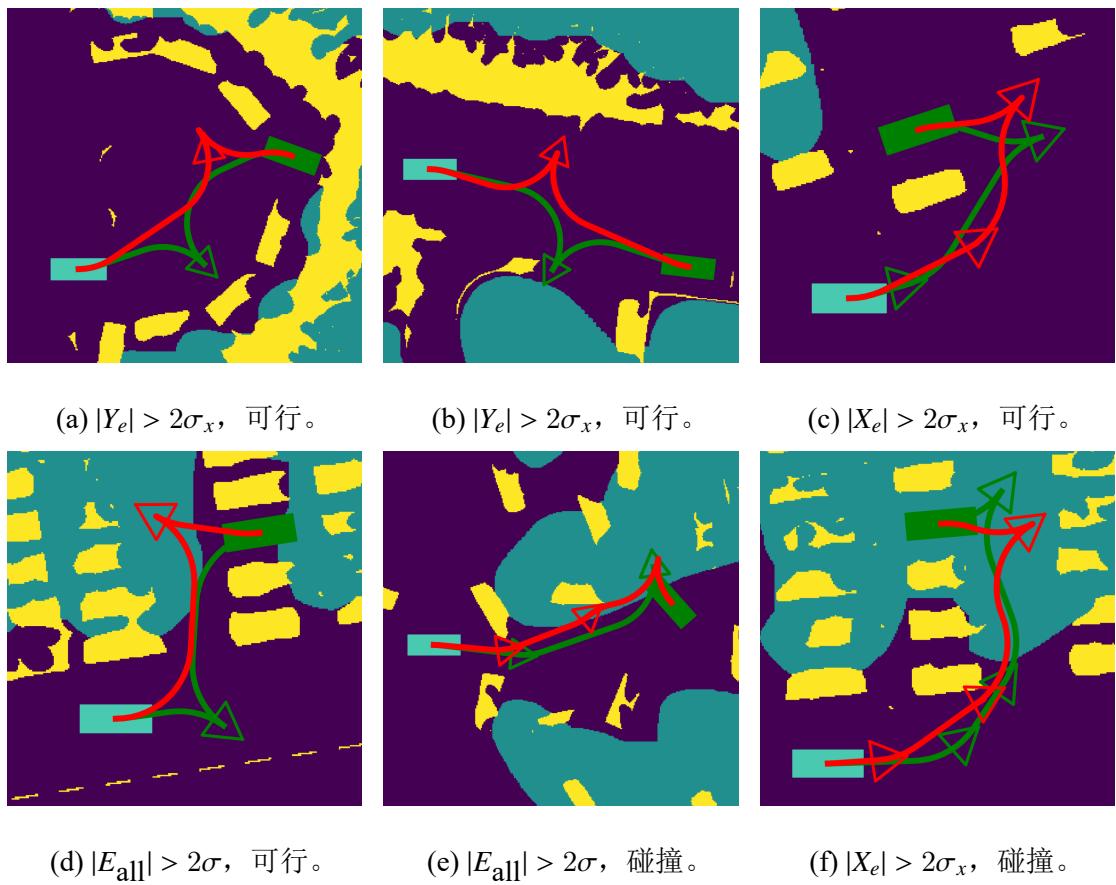


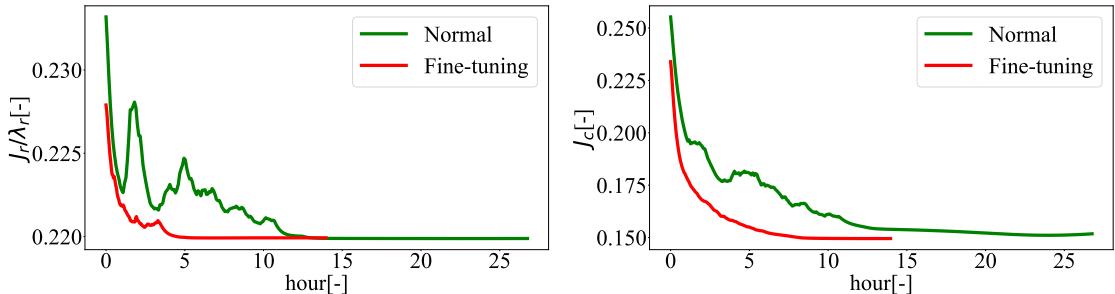
图 4-9 配置预测误差大于  $2\sigma$  的路径与对应的标签路径的对比。其中,  $|E_{all}| > 2\sigma$  等价于  $|X_e| > 2\sigma_x \wedge |Y_e| > 2\sigma_y \wedge |\Phi_e| > 2\sigma_\phi$ , 红色曲线和红色三角形分别表示 YIPS 网络规划的路径和其采样点。绿色曲线和绿色三角形分别表示标签路径和标签采样点。黄色、青蓝色和紫色像素区域分别表示障碍物、未知和可行区域。

$[-2\sigma, 2\sigma]$  内误差相同的较小的预测偏差。所以利用最佳拟合高斯分布对配置预测误差进行建模仍是合理的, 最佳拟合高斯分布可以有效地反映出配置误差的真实分布的特征。

### 4.3 网络模型设计验证

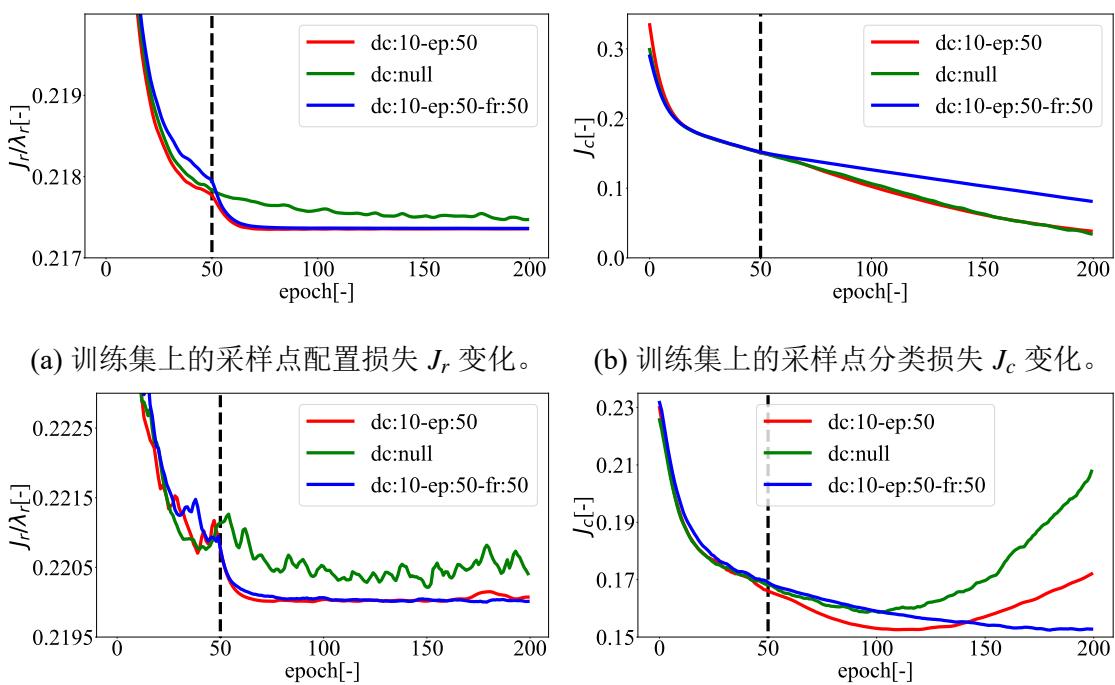
#### 4.3.1 网络训练方法验证

本小节将对使用 Fine-tuning 方法对 YIPS 网络进行训练的有效性进行验证。Fine-tuning 训练方法的流程为: 首先在 Imagenet 数据集上预训练特征提取主干网络再训练整个 YIPS 网络, 在训练整个网络时, 先冻结主干网络(使主干网络的权重不可学习), 仅训练路径规划头部网络, 将头部子网训练至接近收敛后, 再解



(a) 验证集上的采样点配置损失  $J_r$  变化。 (b) 验证集上的采样点分类损失  $J_c$  变化。

图 4-10 Fine-tuning 训练方法和常规训练方法的训练效果对比。其中，横坐标为训练小时数，纵坐标为验证集上的训练损失。绿色和红色曲线分别代表常规训练方法和 Fine-tuning 训练方法。



(a) 训练集上的采样点配置损失  $J_r$  变化。 (b) 训练集上的采样点分类损失  $J_c$  变化。

图 4-11 学习率变化和主干网络解冻对网络训练效果的影响。其中，“dc:10”代表学习率缩减为原来的 1/10，“ep:50”代表在 epoch = 50 时对学习率进行变化，“fr:50”在 epoch = 50 时对主干网络进行解冻。

冻主干网络，以较小的学习率同时训练主干和头部网络至损失收敛。图4-10对比了采用 Fine-tuning 方法和常规训练方法在验证集上的训练效果对比，由图可见，Fine-tuning 方法的训练时间比常规方法少了约 10 小时，同时在验证集的配置和分类损失上均略优于常规训练方法，而且训练过程更加稳定。

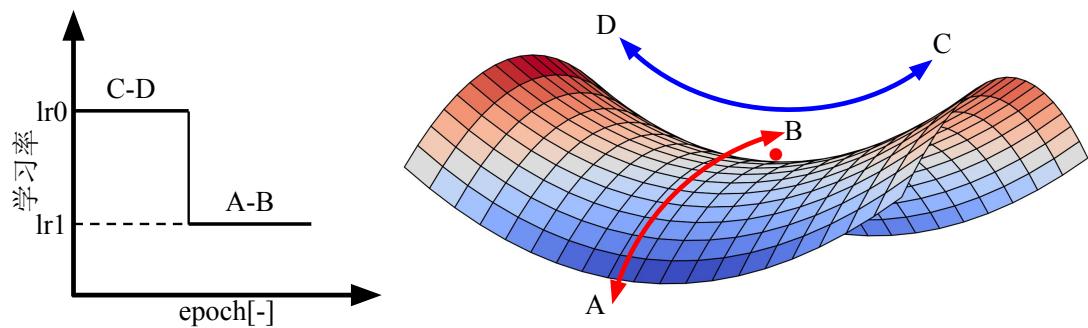


图 4-12 学习率调整可以帮助网络训练越过“鞍点 (saddle)” [53]。左图为学习率调整方案示例图，其中，右图为鞍点的示意图，其中红色圆点为鞍点，蓝色和红色箭头示意了权重更新维度 C-D 和 A-B。

此外，在 Fine-tuning 训练流程中，适时解冻主干网络并以更小的学习率对网络进行训练的操作方法对提高网络训练效果也是十分有效的。图4-11验证了这种操作方法对训练效果的影响。由图可见，解冻主干网络对 YIPS 网络的学习能力有明显的提高作用，释放主干网络后虽然可学习的参数量增加了一倍，但是没有增加网络的过拟合趋势，反而使得网络的学习能力得到提升，在配置损失上，使得验证集上的损失变化更平稳，在分类损失上，消除了未解冻主干网络时的过拟合现象。同时，由图可见，在  $\text{epoch} = 50$  时，将学习率减小为原学习率的  $1/10$  可以加速配置损失和分类损失的收敛，从而提高网络的训练效果，其原因在于通过对学习率进行调整可以避免权重更新陷入“鞍点 (saddle)” [53]。图4-12展示了学习率变化对权重更新过程的影响，鞍点的存在可能使权重更新陷入  $C \leftrightarrow D$  维，而错过可以进一步降低损失的  $A \leftrightarrow B$  维，通过降低学习率，可以使权重更新进入  $A \leftrightarrow B$  维，从而越过鞍点。但是学习率调整带来的优化程度不是一致的而是与任务性质是相关的，由图可见学习率的调整对配置损失的优化程度比分类损失更高，学习率衰减后配置损失出现迅速下降的趋势，而分类损失的下降趋势则较平缓。

因此综上，利用 Fine-tuning 方法进行网络训练是有益的。预训练的主干网络可以给 YIPS 网络提供一个十分有效的初始权重设置，而解冻主干网络和学习率衰减可以提高网络学习能力并帮助权重更新越过鞍点。同时，Fine-tuning 的有效性也说明了，尽管路径规划任务与图像分类任务在任务输入和任务目标上都有很大差异，但是特征提取知识却有共通性，可以迁移共享。

### 4.3.2 损失函数对比

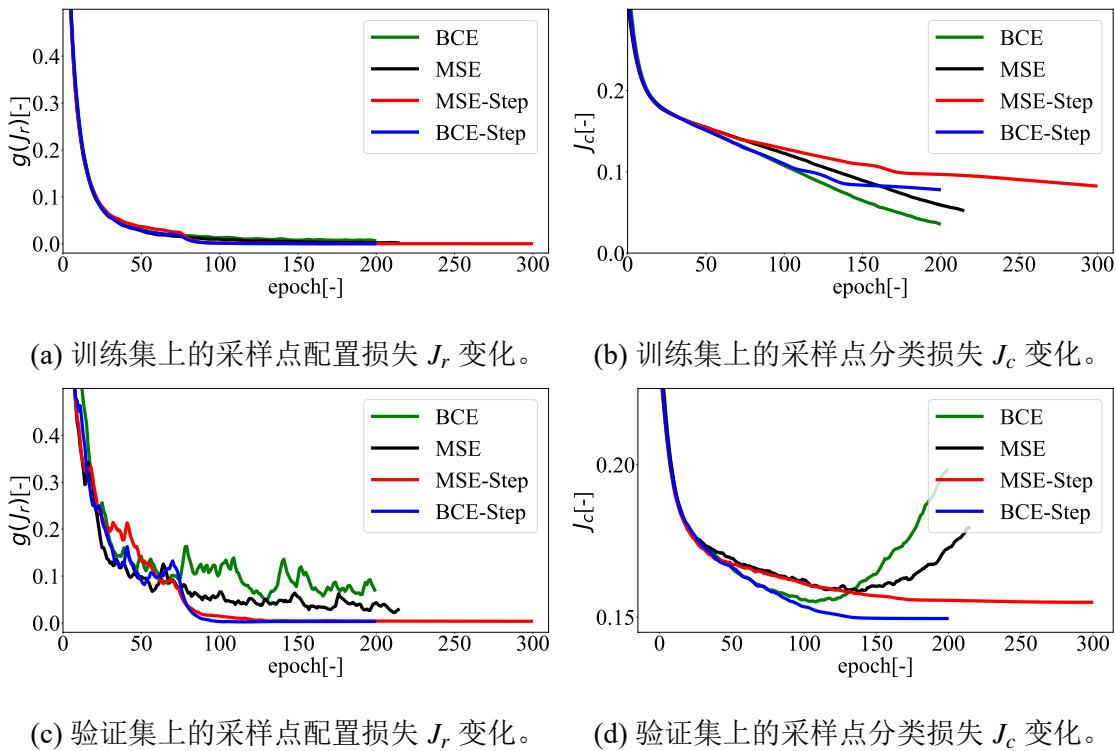


图 4-13 不同的配置损失计算方法对网络训练的影响。BCE 和 MSE 曲线分别代表采用二元交叉熵函数和均方差函数计算配置损失且全程冻结主干网络时 YIPS 网络的训练过程，BCE-Step 和 MSE-Step 曲线则代表在 epoch=75 时解冻主干网络并采用阶梯型学习率衰减策略的训练过程。(a)(c) 子图中的纵坐标  $g(J_r) = [J_r - \min(J_r)] / [\max(J_r) - \min(J_r)]$ ，通过函数  $g(\bullet)$  对配置损失进行规范以消除损失系数不同带来的配置损失值差异。

如节3.2.3所述，计算 YIPS 网络的配置损失即回归损失的方法有两种，分别是二元交叉熵 (BCE) 方法和均方差 (MSE) 方法，分别采用这两种损失计算方法进行网络训练的结果对比如图4-13所示，在全程冻结主干网络，仅训练头部网络的情况下，BCE 方法与 MSE 方法在训练集的配置损失上相差无几，但 BCE 方法在验证集上的损失变化相比而言更动荡一些，可能的原因是采用 BCE 方法训练时选取的损失系数  $\lambda_r$  更大，导致损失变化更剧烈，而在分类损失上，BCE 方法的过拟合程度和速度相对更大。而解冻主干网络并采用阶梯型的学习率衰减策略训练时，虽然训练策略的调整对 BCE 方法和 MSE 方法都有优化作用。但就分类损失而言，采用 BCE 方法时分类损失的收敛速度更快，收敛损失也更小。可见使用 BCE 方法时 YIPS 网络在分类任务上的学习能力更强，训练策略调整能

释放的网络学习能力更多。同时，两种方法在训练和验证集上的配置损失收敛结果无明显差异，这说明了将 BCE 方法应用在回归损失计算上能达到与 MSE 方法相同的效果。

因此综上，BCE 方法比 MSE 方法更适用于同时进行分类和回归任务训练的 YIPS 网络，可以更好的平衡分类和回归任务的训练过程。从网络输出的角度看，造成这一现象的可能原因是采用 BCE 方法计算配置损失时，回归和分类损失的计算方法是相同的，因此在不考虑输出的实质含义的情况下，对于网络而言，采样点分类任务和配置回归任务是同一个任务，而非两个不同的任务，从而使多任务学习退化为单任务学习，降低了学习难度。从损失函数的梯度来看，原因可能在于采用 BCE 方法计算配置损失，可以使两个任务的梯度变化更具有相似性，这种相似性可能有助于分类损失的优化，而采用 MSE 方法计算则无这种相似性，从而一定程度上影响了分类任务的学习。

### 4.3.3 输入编码方式对比

本小节将对 YIPS 网络的不同输入编码方式进行对比。将起始和目标配置以及占用栅格图编码成 RGB 图片的方式有多种，不同的编码方法对网络的学习存在一定的影响，尤其是起始和目标配置的编码方式。起始和目标配置只是配置空间的坐标点，并不具备直接的图像信息，因此确定起始和目标配置在 RGB 图片上的编码形状和面积大小是值得探究的问题。同时，如果将起始和目标配置转换到起始配置的局部坐标系中，则对于在局部坐标系中的规划算法，起始配置信息是可以忽略，但对于 YIPS 网络，是否也可以忽略仍需要验证。另外，在一些运动规划的实践中并不关注未知区域，而是将未知区域直接视为可行区域，此时，将未知区域编码入 RGB 图片还是否必要也是需要验证的。

本文实验了如图4-14所示的 7 种不同的编码方式，它们的主要区别在于起始和目标配置在 RGB 图片上的编码形状和面积大小、是否编码起始配置以及占用栅格图是否包含未知区域。这 7 种不同的编码方式对应的训练结果如表4-4所示。实验结果表明，将基准编码方式中的矩形面积增大会对网络的学习产生负面的影响。矩形的面积过大虽然可以一定程度上减少配置损失，但会显著增加分类损失，原因可能在于增加矩形面积虽然可以在卷积过程中更好地保留起始和目标配置信息，从而有利于采样点配置预测任务，但是过大的矩形会遮挡占用栅格

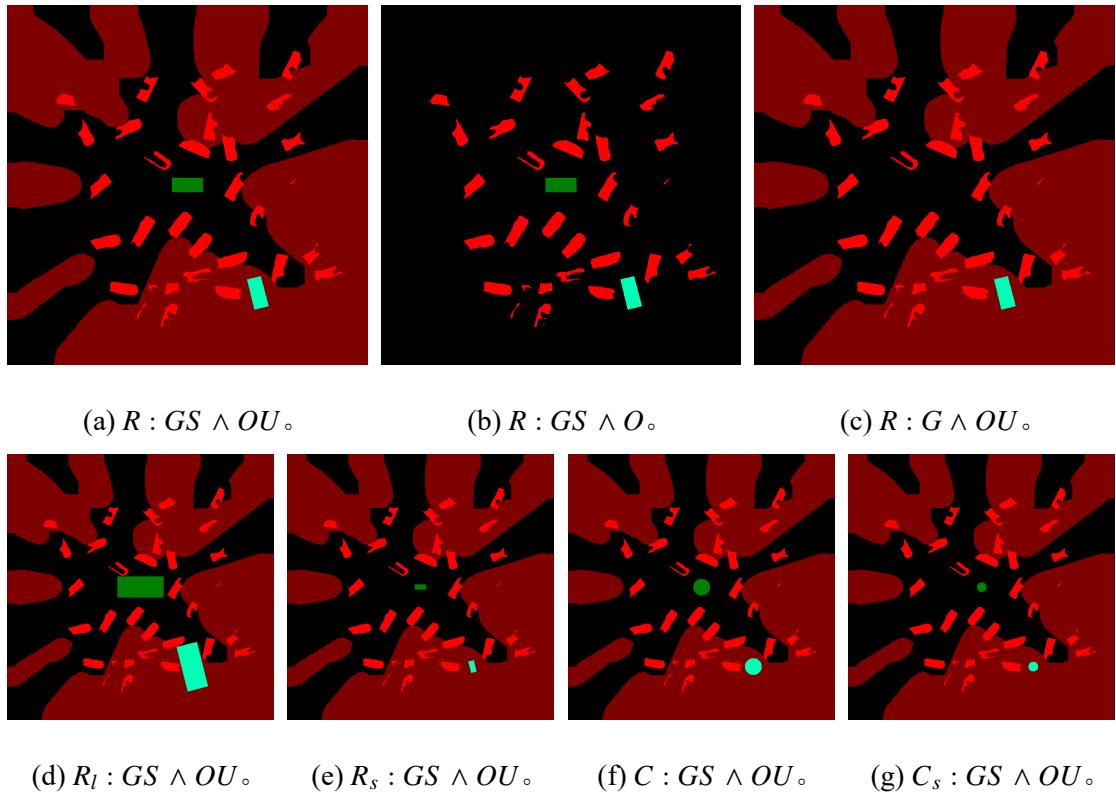


图 4-14 7 种不同的网络输入编码方式。从左到右从上到下，依次为：基准编码方式(即将起始和目标配置编码为长宽等于车辆实际长宽的矩形并编码完整的占用栅格图)、将基准中的占用栅格图的未知区域去除、将基准中的起始配置编码去除、将基准中的矩形长宽放大一倍、将基准中的矩形长宽缩小一半、将基准中的矩形替换为等面积的圆形、将基准中的矩形替换为直径为矩形宽度的圆形。编码方式命名规则为 {配置编码形状}:{被编码的配置}  $\wedge$  {被编码的占用栅格图区域}。圆形和矩形的配置编码形状分别以  $R$  和  $C$  表示，下标  $l$  和  $s$  分别表示编码形状的面积相比基准更大和更小；目标和起始配置以  $G$  和  $S$  表示； $O$  和  $U$  分别代表占用栅格图的障碍物和未知区域。

图，从而污染障碍物分布信息，因此对采样点的分类任务的产生了负面影响。而将基准编码方式中的矩形面积缩小，会导致起始和目标配置在卷积过程中保留的信息过少，不仅会增加配置损失，也会导致分类损失增加，因为在采样点的分类任务中也不能缺失过多的起始和目标配置信息。

而将基准编码中的矩形替换成圆形同样会对网络学习造成负面影响，将矩形替换成等面积的圆形时，虽然配置损失基本不变，但分类损失有所增加，减少圆形直径到矩形宽度时，分类损失也不会得到改善。可能的原因是，圆形编码会丢失一部分的车辆朝向信息，因此导致了分类损失的增加，但圆形的编码形状仍

表 4-4 图4-14所示的 7 种网络输入编码方式对应的网络训练结果对比。

网络输入编码方式	验证集的配置损失 $J_r$	验证集的分类损失 $J_c$
$R : GS \wedge OU$	2199.5	<b>0.1496</b>
$R_b : GS \wedge OU$	<b>2198.5</b>	0.1541
$R_s : GS \wedge OU$	2200.1	0.1520
$C : GS \wedge OU$	2199.8	0.1514
$C_s : GS \wedge OU$	2199.3	0.1522
$R : G \wedge OU$	2198.8	0.1510
$R : GS \wedge O$	2199.8	0.1513

能有效保留起始和目标配置的位置信息，因此使采样点的配置预测仍有成效。

此外，实验结果还表明，不显式地在输入中编码起始配置虽然一定程度上增加分类损失，但是配置损失有所减少，综合来看，网络训练效果差异不大。对于占用栅格图上的未知区域，即便规划任务中可能并不关注未知区域，但是将未知区域表示到输入中仍有利于网络的采样点分类任务，原因可能在于，在已知可行区域的情况下，未知区域是障碍物区域的补集，提供了障碍物分布信息的重要补充，由于传感器对障碍物轮廓的探测是不全面且分辨率不高的，导致障碍物区域会比较破碎，一些特征在卷积中容易丢失，此时未知区域作为补充可以弥补卷积过程中信息的丢失。

综上所述，虽然不同编码方式并不会导致网络训练失效，但基准编码方式是相对而言最好的编码方式，可以更好的保留起始和目标配置以及占用栅格图的信息。同时相比于不编码起始配置的编码方式，其编码的可读性更强，对泊车规划场景的信息表达更完整。

#### 4.3.4 主干网络对比

本小节将对不同类型主干网络的适用性进行对比。对于 YIPS 神经网络，主干网络在很大程度上决定了网络的特征提取能力，因此对网络的学习能力有决定性影响。理论上，所有在图像分类任务中有良好表现的神经网络都具有较好的特征识别和提取能力，因此均适合用作 YIPS 网络的主干网络，但是由于输入 YIPS 网络的规划场景图片是人工图片而非在自然景象下拍摄出的自然图片，因

此虽然在人眼看来，人工图片和自然图片都具有相似的抽象特征，如物体轮廓、颜色等，但是两者在像素值分布和物体特征等方面有明显的不同，如自然图片中有动植物和建筑物等形态各异的物体，且物体均具有阴影和纹理等丰富特征，而在人工图片中，物体的类别和特征都是相对单一的。此外，虽然是针对相同的图像分类任务，但不同网络识别和提取的特征不尽相同且往往不是人所能理解的。所以不同的主干网络在诸如 Imagenet 等自然图片数据集上训练后，其特征提取能力是否可以直接迁移到 YIPS 网络中是不确定的。本文对比了 5 种主流的图像分类网络的训练效果，实验结果如表4-5所示，5 种网络的参数量和在 Imagenet 数据集上的图像分类准确度如表4-6所示。

表 4-5 采用不同类型主干网络的 YIPS 的训练结果对比。其中，VGG-19、VGG-16 和 ResNet-50 的起始学习率是 3e-05，而 Xception 和 DarkNet-53 的起始学习率为 3e-06，所有训练的损失系数均为 1e+04 且采用阶梯型学习率衰减策略，衰减 epoch 依表格中顺序依次为 [75, 105, 135]、[70, 95, 110]、[30, 140, 170]、[50, 80] 和 [25, 50]，在第一次衰减的同时解冻主干网络。

特征提取主干网络	训练集 $J_r$	训练集 $J_c$	验证集 $J_r$	验证集 $J_c$
VGG-19 [43]	2173.4	0.0772	<b>2199.2</b>	<b>0.1495</b>
VGG-16 [43]	2173.4	0.0818	2200.7	0.1499
ResNet-50 [20]	2173.4	<b>0.0733</b>	2200.6	0.1575
Xception [54]	2173.4	0.1685	2277.2	0.2011
DarkNet-53 [46]	2173.4	0.2020	2270.1	0.2059

表 4-6 表4-5所比较的 5 种主干网络的参数量和在 Imagenet[45] 数据集上的分类准确度。

特征提取主干网络	Top-1 准确率	Top-5 准确率	参数量	网络总参数量
VGG-19	74.50 %	92.00 %	20.0 M	38.7 M
VGG-16	74.40 %	91.90 %	14.7 M	33.4 M
ResNet-50	77.15 %	93.29 %	19.1 M	40.7 M
Xception	79.00 %	94.50 %	16.1 M	38.0 M
DarkNet-53	77.20 %	93.80 %	35.4 M	57.2 M

由表4-5可见，VGG-19 和 VGG-16 作为主干网络时 YIPS 网络的泛化误差要

表 4-7 采用不同类型主干网络的 YIPS 的类别预测性能对比。其中  $mAP_{WFS+IOV}$  和  $mAP_{IOV}$  表示不同 IOV 阈值  $[0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75]$  下  $AP_{WFS+IOV}$  和  $AP_{IOV}$  的平均值, 下标  $IOV.0$  即 IOV 阈值为 0.0。

主干网络	AP	$AP_{WFS}$	$mAP_{IOV}$	$mAP_{WFS+IOV}$	$AP_{IOV.0}$	$AP_{WFS+IOV.0}$
VGG-19	<b>97.0</b>	<b>78.5</b>	<b>93.6</b>	<b>75.0</b>	75.1	<b>55.5</b>
VGG-16	96.8	77.5	93.4	74.0	<b>75.2</b>	54.5
ResNet-50	96.5	74.4	92.8	70.4	75.1	50.6

表 4-8 采用不同类型主干网络的 YIPS 的序列长度预测准确度对比。 $Acc_{IOV}^{max}$  表示给定 IOV 阈值下预测准确度 Acc 的最大值,  $mAcc_{IOV}^{max}$  则表示不同 IOV 阈值下  $Acc_{IOV}^{max}$  的平均值。 $Acc_{IOV}^{max-1}$  表示当预测准确度 Acc 的取得最大值时, 针对长度为 1 的序列的预测准确度。 $mAcc_{IOV}^{max-1}$  同理表示不同 IOV 阈值下  $Acc_{IOV}^{max-1}$  的平均值。IOV 阈值的所有取值为  $\{0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75\}$ 。

主干网络	$mAcc_{IOV}^{max}$	$mAcc_{IOV}^{max-1}$	$mAcc_{IOV}^{max-2}$	$mAcc_{IOV}^{max-3}$	$mAcc_{IOV}^{max-4}$
VGG-19	<b>67.9 %</b>	<b>82.7 %</b>	<b>60.6 %</b>	<b>5.5 %</b>	0.0 %
VGG-16	67.5 %	82.6 %	59.0 %	8.0 %	0.0 %
ResNet-50	65.5 %	79.9 %	58.8 %	2.5 %	0.0 %

明显小于以 ResNet-50、Xception 和 DarkNet-53 为主干网络的情况。对比表4-5中各主干网络的参数量和对应的 YIPS 网络的总参数量可知, 这种训练效果的差异并不是由参数量的差异引起的, 如 ResNet-50、Xception 和 DarkNet-53 的参数量要明显多于 VGG-16, 但增加的参数量未能使前者的训练损失优于后者, 而 VGG-19 虽然参数量也明显多于 VGG-16, 但却获得了相较而言更好的训练结果。进一步比较不同主干网络在 Imagenet[45] 数据集上的图像分类准确度, 可以发现训练结果的好坏与主干网络在图像分类任务上的准确度存在一定的联系, ResNet-50、Xception 和 DarkNet-53 的 Top-1 准确度均大于 77.00 %, 而 VGG-19 和 VGG-16 的准确度仅在 74.00 % 左右, 因此, 在图像分类任务上准确度高的网络不一定适合作为 YIPS 网络的主干, 网络在图像分类任务上的性能越高, 其作为 YIPS 网络的主干时 YIPS 的学习能力可能越差。

4-7和4-8进一步比较了分别使用 VGG-19、VGG-16 和 ResNet-50 作为主干

的 YIPS 网络的类别预测和序列长度预测性能，综合而言，以 VGG-19 为主干时 YIPS 网络的采样点分类性能最好，其序列长度预测准确度  $mAcc_{IOV}$  也是最高的，同时在类别预测和序列长度预测性能上，VGG-16 和 VGG-19 的性能均高于 ResNet-50。此外，对于所有主干网络，针对长度为 4 的序列的预测准确度均为 0 %，提示了数据集类别分布不均对 YIPS 网络预测带来的负面影响无法通过主干网络进行克服。

综上所述，在所比较的 5 种主干网络中，VGG-19 是其中性能最好的一种，VGG-16 与 VGG-19 性能接近，ResNet-50 差于前两种，而 Xception 和 DarkNet-53 是相对性能最差的。因此可以说，只有 VGG 网络才能比较成功的将图像分类任务的特征提取知识迁移到 YIPS 的路径规划任务中。类似的现象在基于神经网络的风格迁移领域 [55] 也有出现，几乎所有的成功的风格迁移网络都基于 VGG 网络 [56][57]。参考已有研究，VGG 网络比其他网络更适合作为 YIPS 的主干网络的可能原因有二：其一是 VGG 网络可以抑制卷积神经网络反向传播时产生的棋盘效应 (Checkerboard Artifacts)[58]，从而使反向传播更稳定，进而提高网络从非自然图片中提取特征的能力。原因其二是 VGG 网络相比与其他网络，识别和提取鲁棒特征 (Robust Features) 的能力更强 [59]，而规划场景图片与自然图片在鲁棒特征上的相似性很高，而在非鲁棒特征上相似度较低，因此以预训练的 VGG 为主干网络时，YIPS 网络的特征提取能力更出色。鲁棒性特征指人可以理解和识别的特征，非鲁棒特征即人不可见但在高维空间中被神经网络识别出的特征。

### 4.3.5 任务融合验证

本小节将对节3.2.4提出的直接融合方法进行验证。因为基于 VGG 的卷积网络更适合作为 YIPS 网络的主干，所以选用以 VGG-16 为主干的目标检测网络 SSD[38] 作为 YIPS 网络的任务融合对象。称 SSD 网络的主干网络为 SVG-16，以 SVG-16 为主干，通过直接融合方法训练的 YIPS 网络的训练结果如表4-9所示，为了提供参照，表中同时展示了 Fine-tuning 方法训练的以 VGG-19 和 VGG-16 为主干的 YIPS 网络的训练结果，以及采用直接融合方法训练的以 VGG-19 为主干的 YIPS 网络的训练结果。由表可见，相比于 Fine-tuning 方法的训练结果，直接融合方法的训练结果并不差，虽然不对 SVG-16 的权重进行解冻和微调会一定程度上影响网络学习，但是负面影响在可接受范围，而且由于未对主干网络解冻，

表 4-9 基于 SVG-16 的直接融合方法训练结果。SVG-16 代表从预训练的目标检测网络 SSD 中剥离出来的主干网络 VGG-16。

主干网络	训练方法	验证集 $J_r$	验证集 $J_c$	训练时间
SVG-16 [38]	直接融合	2199.6	0.1515	<b>10.4 h</b>
VGG-19	直接融合	2200.1	0.1548	15.9 h
VGG-19	Fine-tuning	<b>2199.2</b>	<b>0.1495</b>	13.6 h
VGG-16	Fine-tuning	2200.7	0.1499	16.1 h

表 4-10 直接融合方法与 Fine-tuning 方法训练的 YIPS 网络的采样点类别预测性能对比。其中  $mAP_{WFS+IOV}$  和  $mAP_{IOV}$  表示不同 IOV 阈值  $[0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75]$  下  $AP_{WFS+IOV}$  和  $AP_{IOV}$  的平均值，下标 IOV.0 即 IOV 阈值为 0.0。

主干网络	训练方法	AP	$AP_{WFS}$	$mAP_{IOV}$	$mAP_{WFS+IOV}$
SVG-16	直接融合	96.8	76.6	93.1	73.0
VGG-19	Fine-tuning	<b>97.0</b>	<b>78.5</b>	<b>93.6</b>	<b>75.0</b>
VGG-16	Fine-tuning	96.8	77.5	93.4	74.0

表 4-11 直接融合方法与 Fine-tuning 方法训练的 YIPS 网络的序列长度预测性能对比。  
 $Acc_{IOV}^{max}$  表示给定 IOV 阈值下预测准确度 Acc 的最大值， $mAcc_{IOV}^{max}$  则表示不同 IOV 阈值下  $Acc_{IOV}^{max}$  的平均值。 $Acc_{IOV}^{max-1}$  表示当预测准确度 Acc 的取得最大值时，针对长度为 1 的序列的预测准确度。 $mAcc_{IOV}^{max-1}$  同理表示不同 IOV 阈值下  $Acc_{IOV}^{max-1}$  的平均值。IOV 阈值的所有取值为  $\{0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75\}$ 。

主干网络	训练方法	$mAcc_{IOV}^{max}$	$mAcc_{IOV}^{max-1}$	$mAcc_{IOV}^{max-2}$	$mAcc_{IOV}^{max-3}$
SVG-16	直接融合	66.3 %	77.6 %	<b>63.1 %</b>	<b>9.6 %</b>
VGG-19	Fine-tuning	<b>67.9 %</b>	<b>82.7 %</b>	60.6 %	5.5 %
VGG-16	Fine-tuning	67.5 %	82.6 %	59.0 %	8.0 %

训练收敛的速度也更快，训练耗时更短。此外，与迁移自图像分类任务的主干 VGG-19 相比，同样采用直接融合方法训练的情况下，目标检测主干 SVG-16 的训练效果要更好，可见，目标检测任务中的知识比图像分类任务的知识更适用于路径规划任务。

表4-10和4-11进一步比较了分别使用直接融合方法与 Fine-tuning 方法训练的 YIPS 网络在类别和序列长度预测性能上的差异。对于类别预测，直接融合方法训练的 YIPS 网络虽然预测性能有所下降，但与 Fine-tuning 方法训练的最好结果相比，在各个指标上均差距不大。对于序列长度预测性能，直接融合方法训练的 YIPS 虽然在总预测准确度指标  $mAcc_{IOV}$  上的表现不如 Fine-tuning 方法训练的 YIPS，但针对长度为 2 和 3 的序列，其预测准确度反而更高，证明从目标检测任务中迁移过来的知识可以一定程度上改善数据集类别分布不平衡带来的负面影响。

因此，综上所述，基于直接融合方法，通过共享 SVG-16 主干将 YIPS 路径规划网络和 SSD 目标检测网络整合在一起，从而将路径规划任务和目标检测任务进行任务融合是可行，虽然不对网络主干进行微调会降低网络的学习能力，但从目标检测任务中迁移的知识可以一定程度上弥补学习能力的不足。

## 4.4 本章总结

本章中，对二阶段泊车规划方法的核心部分 YIPS 路径规划网络进行了性能评估和设计验证。评估结果表明 YIPS 网络的采样点类别预测 AP 值可达 97.0，采样点序列长度预测准确度可达 72%。对 YIPS 网络的配置预测误差进行的分析表明，网络在配置的  $x$ 、 $y$  和  $\phi$  维度上不存在样本无关的固定误差，同时配置误差分布具有明显的高斯性，可利用最佳拟合高斯分布对配置预测误差进行合理建模。此外本章还对 YIPS 网络设计中的关键部分进行了验证：通过 Fine-tuning 方法迁移图像分类任务中的知识确实有利于 YIPS 网络在路径规划任务中的表现。相比于均方差计算方法，采用二元交叉熵方法计算配置损失更有利于网络的训练。将起始和目标配置编码为长宽等于车辆实际长宽的矩形并编码完整的占用栅格图是最合理的输入编码方式。不同的主干网络对 YIPS 网络的学习能力影响很大，而 VGG 网络最适合作为 YIPS 的主干网络。此外，本章还通过实验证明了采用直接融合方法将目标检测任务和路径规划任务进行任务融合的可行性。



## 第 5 章 泊车规划效果验证与评估

本章中，首先将本文提出的二阶段泊车规划方法与基于 OSE 启发式和 GBS 启发式的 Bi-RRT\* 泊车规划方法进行对比以验证方法的有效性和先进性。然后进一步对 YIPS 网络的初始路径规划性能和 SO-RRT\* 算法的初始路径优化性能进行了评估，最后验证了用高斯分布对 YIPS 网络预测误差进行建模的必要性。

### 5.1 实验平台参数

本章中包括 SO-RRT\*、Bi-RRT\* 和 OSE 算法在内的所有路径规划及启发算法均运行在 Intel Xeon E5@3.20GHz 处理器中的单核上，以 Python 作为算法主要实现语言。YIPS 神经网络则在 NVIDIA Tesla P100 显卡上运行，基于 Tensorflow[51] 的 Keras 深度学习框架，同样以 Python 作为主要实现语言。用于测试的数据集大小为 2741，即包含 2741 个规划场景及其对应的标签路径的测试集，数据集制作方法同节3.4所述。用于碰撞检查的占用栅格图的尺寸和分辨率为  $60\text{ m} \times 60\text{ m}$  和  $0.1\text{ m}/\text{cell}$ 。在所有实验中，非特殊说明情况下，SO-RRT\* 的偏置采样过程中使用的高斯模型均为 YIPS 配置预测误差的最佳拟合高斯模型。

### 5.2 泊车运动规划方法对比

#### 5.2.1 实验设置与评价参数说明

本节将把本文提出的二阶段泊车规划算法与两种主流的泊车规划算法即基于 GBS 启发的 Bi-RRT\* 和基于 OSE 启发的 Bi-RRT\* 算法进行对比，以验证和评估本文提出的规划方法的有效性和先进性。其中，基于 OSE 启发的 Bi-RRT\* 算法是目前最为先进的泊车规划方法之一 [12][22]。本文实验中，OSE 启发式的参数配置如表5-1所示，以 OSE 为启发的 Bi-RRT\* 的偏置采样高斯模型的参数如表5-2所示。GBS 启发式的高斯模型参数如表5-3所示。二阶段泊车规划方法采用以 VGG-19 为主干的 YIPS 网络作为第一阶段路径规划网络，第二阶段优化算法 SO-RRT\* 的偏置采样高斯模型的参数如表5-4所示。所有规划方法的规划目标车辆相同，其参数如表3-1所示。

表 5-1 本文实验中采用的 OSE 启发式的参数配置表。

最小圆半径	最大圆半径	与障碍物最小间距	相邻节点数	最大转向曲率
0.200 m	3.029 m	1.058 m	32	0.200 m <sup>-1</sup>

表 5-2 以 OSE 为启发的 Bi-RRT\* 的偏置采样高斯模型参数。Bi-RRT\* 依次遍历 OSE 启发输出的圆形节点序列的每一圆形节点，以圆形节点的圆心为基准点进行偏置采样，表中的符号  $r$  表示圆形节点的半径。 $(\mu_x, \sigma_x)$ 、 $(\mu_y, \sigma_y)$  和  $(\mu_\phi, \sigma_\phi)$  分别为高斯模型在配置的  $x$ 、 $y$  和  $\phi$  维度上的 (均值, 方差)。

$\mu_x$ [m]	$\sigma_x$ [m]	$\mu_y$ [rad]	$\sigma_y$ [rad]	$\mu_\phi$ [rad]	$\sigma_\phi$ [rad]
0.00	$r/3$	0.00	$r/3$	0.00	$\pi/4$

表 5-3 GBS 启发的高斯模型参数表， $r$  和  $\theta$  为采样点的局部极坐标系的极径和极角， $(\mu_r, \sigma_r)$  和  $(\mu_\theta, \sigma_\theta)$  分别为高斯模型在这两个坐标维度上的 (均值, 方差)。 $\phi$  为车辆朝向角， $(\mu_\phi, \sigma_\phi)$  为高斯模型在朝向角维度上的 (均值, 方差)。

$\mu_r$ [m]	$\sigma_r$ [m]	$\mu_\theta$ [rad]	$\sigma_\theta$ [rad]	$\mu_\phi$ [rad]	$\sigma_\phi$ [rad]
2.00	0.50	0.00	$\pi/4$	0.00	$\pi/6.$

表 5-4 SO-RRT\* 的偏置采样高斯模型参数。 $(\mu_x, \sigma_x)$ 、 $(\mu_y, \sigma_y)$  和  $(\mu_\phi, \sigma_\phi)$  分别为高斯模型在配置的  $x$ 、 $y$  和  $\phi$  维度上的 (均值, 方差)。

$\mu_x$ [m]	$\sigma_x$ [m]	$\mu_y$ [rad]	$\sigma_y$ [rad]	$\mu_\phi$ [rad]	$\sigma_\phi$ [rad]
0.087	2.436	0.052	1.659	-0.014	0.500

为了有效对比规划算法的场景适应性，本文选取了 6 个指标对算法规划性能进行评价，分别是：(1) 算法规划成功率 SR<sub>0</sub>，即算法成功规划的场景数与测试集规划场景总数之比；(2) 仅允许算法规划 0.1s 时的规划成功率 SR<sub>0.1</sub>；(3) 算法规划首条可行路径的平均时间 mTTFP；(4) 在不计入启发式和神经网络运算时间时，算法规划首条可行路径的平均时间 mTTFP<sub>WIT</sub>；(5) 首条路径的平均长度 mLOP<sub>TTFP</sub>；(6) 规划首条路径 0.1s 后算法输出的路径的平均长度 mLOP<sub>TTFP.1</sub>。指标中的平均指测试集中所有规划场景下的测量结果的平均。通过这 6 个指标

可以综合评价算法的规划成功率、规划速度、以及规划路径质量，从而有效对比算法的场景适应性。另外，需要指出的是，对于 GBS 启发的 Bi-RRT\* 算法，由于其启发计算过程即是在 Bi-RRT\* 的采样计算过程，因此不存在额外的启发计算耗时，故 mTTFP<sub>WIT</sub> 即是 mTTFP，两者等同。

此外，三种规划方法分别在测试集上运行三次，每次运行时，只允许在每个规划场景中规划一次，最大采样次数为 500，选取三次运行中规划成功率 SR 最高的一次作为最终运行结果，以一定程度上消除随机性。

### 5.2.2 对比结果分析

表 5-5 本文提出的二阶段泊车规划方法与 OSE 启发 Bi-RRT\* 规划方法以及 GBS 启发 Bi-RRT\* 规划方法的规划性能对比表。表中的规划性能评价指标的说明以及规划方法的参数设置详见节 5.2.1(实验设置与评价参数说明)。

规划方法	SR	SR <sub>0.1</sub>	mTTFP/mTTFP <sub>WIT</sub>	mLOP <sub>TTFP</sub>	mLOP <sub>TTFP.1</sub>
二阶段规划	99.8 %	90.0 %	76.81/53.58 ms	24.92 m	22.71 m
OSE 启发规划	99.1 %	20.6 %	530.44/74.70 ms	25.02 m	22.83 m
GBS 启发规划	98.1 %	76.1 %	166.50/166.50 ms	26.83 m	24.10 m

本文提出的二阶段泊车规划算法与基于 GBS 启发和基于 OSE 启发的 Bi-RRT\* 算法在测试集上的规划性能对比如表 5-5 所示。

首先，通过对比指标 SR 和 mTTFP 可见，在规划时间充分的情况下，三种规划算法的成功率都可以到达 99 % 左右，但是二阶段规划方法的规划速度明显快于其余两种方法，平均首条可行路径规划时间为 76.81 ms，而 OSE 启发规划和 GBS 启发规划均超过 100 ms，其中 OSE 启发规划甚至超过 0.5 s。由此可见，相比其余两种方法，二阶段规划方法不仅规划成功率高，而且规划速度更快，可以在事件触发的控制系统中实现平均帧率 10 FPS 以上的实时规划，且规划成功率可达 99.8 %。

进一步比较指标 SR<sub>0.1</sub> 可见，在仅允许算法规划 0.1s 时，二阶段规划方法的规划成功率仍能保持 90 %，而 GBS 启发规划的成功率降至 76.1 %，OSE 启发规划的成功率仅为 20.6 %，这说明，其一，二阶段规划方法在时间触发的控制系统中也可以实现成功率达 90 % 的实时 (10 FPS) 规划，显著地快于其他两种方法。

其二，二阶段规划方法的规划速度不仅在大部分规划场景下快于其他两种方法，规划速度稳定性也明显更优，场景适应性更强。

另外，通过对比指标  $mLOP_{TTFP}$  和  $mTTFP$  可见，二阶段规划方法的首条路径平均长度是最短的，分别比 OSE 启发规划方法和 GBS 启发规划方法短了 0.1 m 和 1.91 m。而允许三种方法对初始路径进行 0.1 s 的优化后，二阶段方法输出的路径仍是最短的，平均比 OSE 启发规划方法和 GBS 启发规划方法短 0.12 m 和 1.39 m。由此可见，相比于其他两种方法，二阶段方法不仅规划速度更快，路径质量也更优。

此外，通过对比指标  $mTTFP_{WIT}$  可见，在不考虑启发式和神经网络运算时间的情况下，二阶段规划方法规划首条路径的时间仅为 53.58 ms，耗时明显少于 OSE 启发规划的 74.70 ms 和 GBS 启发规划的 166.50 ms。从启发规划的角度看，这说明了 YIPS 网络的启发效果要优于 OSE 和 GBS，在 YIPS 启发下，虽然 SO-RRT\* 的采样速度慢于 Bi-RRT\*，但是收敛速度却更快，而且求解的首条路径的长度也更短。

综上所述，与 OSE 启发规划方法和 GBS 启发规划方法相比，本文提出的二阶段泊车规划方法不仅规划速度更快，而且规划成功率更高，同时规划路径的质量也更优，整体而言泊车规划性能更优异，场景适应性更强。此外，以测试集的规划表现来看，本文提出的二阶段泊车规划方法可以实现成功率 99 % 以上的实时（平均帧率 10 FPS 以上）路径规划，而 OSE 启发规划和 GBS 启发规划均无法达到实时性要求。而从启发采样的角度看，实验也证明了，相比于 OSE 的可行区域启发，YIPS 网络提供的采样点序列（即路径）启发的启发效果更好，启发规划收敛速度更快。

## 5.3 YIPS 网络初始路径规划评估

### 5.3.1 路径规划性能评估

本节将对 YIPS 网络的初始路径规划性能进行评估。对于 YIPS 路径规划网络的路径规划性能，本文以四个指标进行衡量，分别是网络平均推断时间  $mIT$  (mean Inference Time)、规划路径的平均长度  $mLOP$  (mean Length Of Paths)、规划成功率  $SR$  (Success Rate) 以及路径碰撞度  $IOV$  (依式4-1定义)，其中在规划成功率  $SR$

中也引入 IOV 指标，以更好评估规划路径中不同碰撞程度的路径的占比。

表 5-6 采用不同类型主干的 YIPS 网络在测试集上的路径规划性能评估结果。mIT 为网络平均推断时间；mLOP 为规划路径的平均长度， $\Delta$  为规划路径与标签路径的平均长度之差；SR 为规划成功率，下标 IOV.125 表示允许路径 IOV 不大于 0.125 时的规划成功率；mIOV 表示路径碰撞度 IOV 的平均值， $\text{IOV}_{\max}$  则表示路径碰撞度 IOV 的最大值。除了以 SVG 为主干时采用直接融合方法训练，其余情况下均以 Fine-tuning 方法训练。

主干网络	mIT [ms]	mLOP/ $\Delta$ [m]	SR	SR <sub>IOV.125</sub>	SR <sub>IOV.5</sub>	mIOV/IOV <sub>max</sub>
VGG-19	23.23	22.45/0.65	<b>24.5 %</b>	72.4 %	<b>95.3 %</b>	0.12/2.45
SVG-16	20.06	22.44/0.64	22.2 %	71.3 %	94.6 %	0.12/2.79
VGG-16	19.58	<b>22.40/0.60</b>	22.5 %	<b>72.7 %</b>	95.2 %	<b>0.12/1.95</b>
ResNet-50	<b>17.48</b>	22.66/0.86	23.7 %	65.3 %	91.0 %	0.16/3.85

采用不同类型主干的 YIPS 网络在测试集上的评估结果如表5-6所示。由表可见，YIPS 网络的推断速度很快，不同主干的 mIT 均在 20 ms 左右，即推断帧率为 50 FPS，完全满足实时性要求。其中以 ResNet 为主干时，得益于 ResNet 网络更快的浮点运算速度，mIT 可达 17 ms，即推断帧率为 58.8 FPS。同时，对于采用不同类型主干的 YIPS 网络，规划路径与标签路径的长度差均小于 1 m，其中最差的结果仅为 0.86m，而最好的结果可达 0.6 m，因此，就路径长度 mLOP 而言，YIPS 网络规划的路径已经相当接近最优解，有助于减轻第二阶段优化算法 SO-RRT\* 的寻优负担。

另外，YIPS 网络的规划成功率不高，最好的结果仅为 24.22 %，不足四分之一，但是，进一步考虑 IOV 阈值为 0.125 时，成功率 SR 上升至 70 % 左右，最好的结果为 73.04 %，这表明超过 70 % 的碰撞路径的碰撞面积不足车辆占用面积的 1/8，路径整体碰撞度是比较低的，而 IOV 阈值提高到 0.5 后，SR 进一步上升至 95 % 左右。因此，虽然由于在损失设计中未能充分考虑路径碰撞产生的损失，削弱了网络对路径是否碰撞的识别能力，但整体而言网络对路径是否存在碰撞仍有一定判断能力。

此外，由表还可以看出，对于采用不同类型主干的 YIPS 网络，其 IOV 最小可达 0.117，而  $\text{IOV}_{\max}$  的最小值为 1.95，即车辆占用区域面积的两倍左右，可见尽管占比不高但 YIPS 网络规划的路径中存在部分碰撞度很大的路径。图5-1展

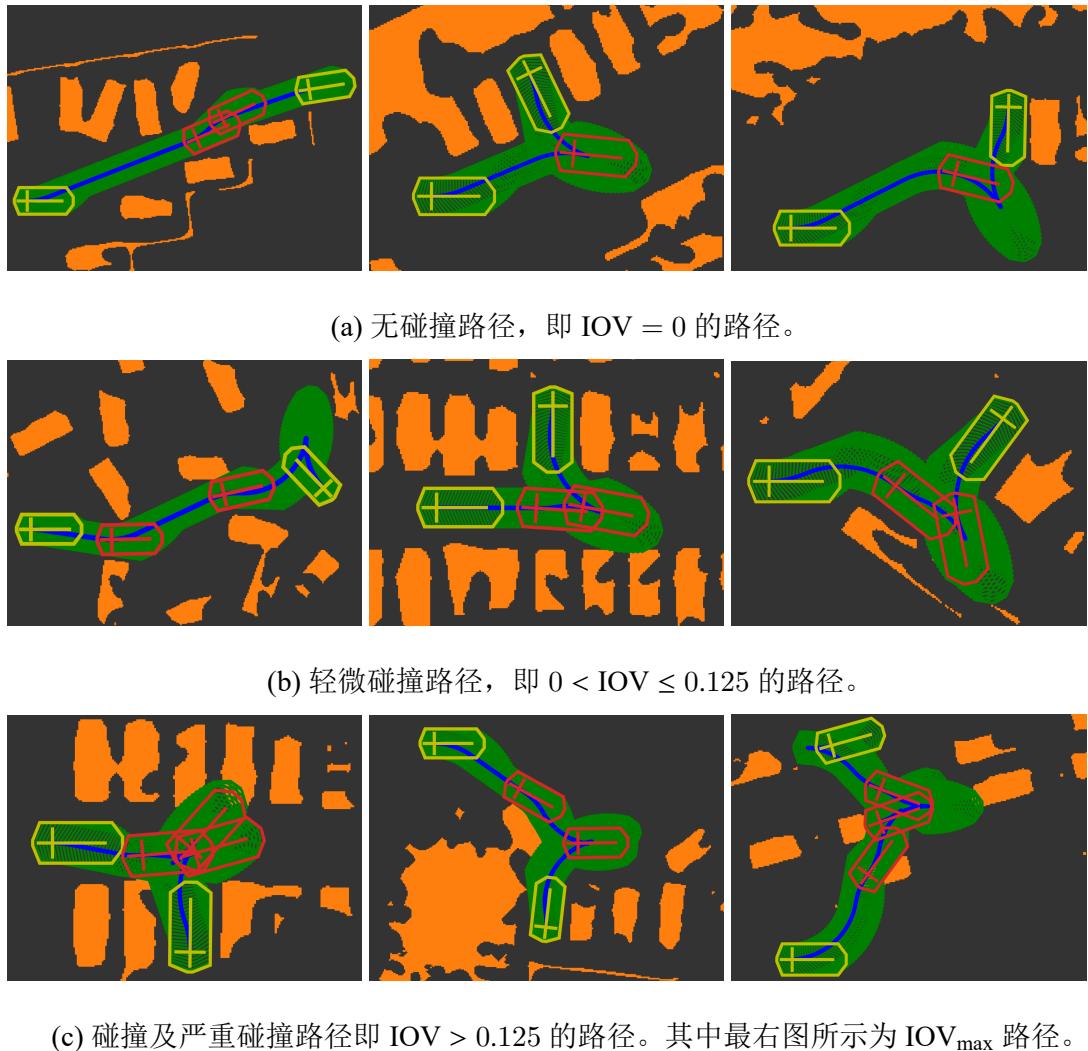


图 5-1 不同碰撞程度的 YIPS 网络预测路径。其中, 橙色块表示障碍物, 绿色轮廓为路径覆盖的区域, 蓝色曲线为预测路径, 黄色多边形和十字构成起始和目标配置, 黄色十字的交叉点为车辆后轴中点。红色多边形和十字形构成 YIPS 神经网络预测的采样点。

示了不同碰撞程度的预测路径。由图可以看到  $\text{IOV} < 0.125$  的路径的碰撞程度是十分轻微的, 对其进行优化校正也相对容易。而对于碰撞较为严重的场景, 即  $\text{IOV} > 0.25$  的场景, 虽然路径碰撞区域较大, 但网络所预测的采样点的分布仍是相对合理的, 基本分布在关键的绕障点附近, 这样的分布仍有助于 SO-RRT\* 对路径的优化校正, 当然, 同时也对 SO-RRT\* 的优化能力提出了挑战, 尤其是对图5-1c最右所示的预测严重偏离情况。

最后, 由表还可以看出, YIPS 网络的规划性能与其采样点类别和配置预测性能是正相关的, 预测性能越好, 规划路径质量和规划成功率越高。此外, 与

性能相对最高的 VGG-19 为主干的 YIPS 网络相比，采用直接融合方法训练并以 SVG-16(即目标检测网络 SSD 的主干网络)为主干的 YIPS 网络并没有出现明显的不足，两者性能差距并不明显。再次说明了采用直接融合方法对目标检测和路径规划任务进行任务融合的可行性。

综合而言，YIPS 网络可以实时规划出碰撞度不高的初始路径，有助于提高路径规划方法整体的规划速度，对实现实时路径规划十分重要。但 YIPS 网络独立完成给定规划任务的成功率不高，大部分规划路径存在轻微碰撞，因此对 YIPS 网络规划的路径进行第二阶段优化将十分必要。

### 5.3.2 规划速度评估

与其他基于深度神经网络的规划方法相同，本文所提出的二阶段路径规划方法的规划速度非常依赖于深度神经网络的推断速度，即 YIPS 路径规划网络的规划速度。因此为了评估 YIPS 网络的规划速度以及规划速度的场景适应性，将 YIPS 网络在测试集中不同场景的推断速度与 OSE 启发式 [12] 的推断速度进行对比，OSE 启发式是目前主流的采样规划算法启发式中综合性能最好的一种。

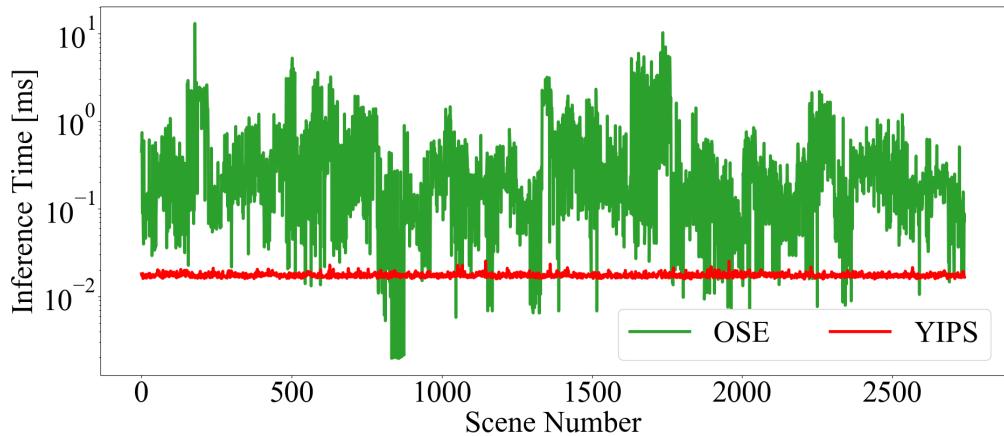


图 5-2 YIPS 神经网络与 OSE 启发式在不同规划场景中的推断时间对比。

YIPS 网络与 OSE 启发式在测试集的不同规划场景中的推断时间对比结果如图5-2所示。由图可见，总体而言 YIPS 网络的规划速度要明显快于 OSE 启发式，这得益于 YIPS 神经网络可以并行计算的特点。同时 YIPS 网络的规划速度的稳定性也更高，推断时间曲线相比 OSE 启发式几乎没有震荡，这得益于端到端的卷积神经网络的另一个特点，即网络输入尺寸固定后网络推断时间与输入基本

无关。OSE 启发式虽然在部分场景中比 YIPS 网络的推断时间更短，但是规划时间依赖于场景的结构，对于不同场景的适应性远不如后者。

## 5.4 SO-RRT\* 算法初始路径优化验证

### 5.4.1 寻优方式评估

本小节将对比两种不同的父节点搜索方式对 SO-RRT\* 路径优化性能的影响。SO-RRT\* 本质上是一种采样规划算法，其采样点的父节点的搜索方式，即算法5中函数 CostLeast 的实现方式，对算法探索规划空间和寻找最优解的性能都有较大的影响。父节点的搜索方式主要有两种，一种是贪婪方式 (Greedy)，即将树中离采样点最近的顶点作为采样点的父节点，另一种是本文采用的最优方式 (Optimal)，即将树中能使采样点 Cost 最小的顶点作为采样点的父节点。Greedy 方式因为无需在搜索时对顶点与采样点间的路径进行碰撞检查，所以单位时间内采样次数更多，使算法能更快的探索规划空间，但是与 Optimal 方式相比，其输出路径的质量往往更差。

为了评估这两种搜索方式对 SO-RRT\* 优化性能的影响，选取如图5-3所示的 6 种典型的规划场景，通过对比不同场景下 SO-RRT\* 算法的收敛曲线进行评估。收敛曲线即算法输出路径的平均长度 mLOP 随规划时间变化的曲线，通过收敛曲线可以容易看出算法规划首条可行路径的速度以及收敛到最优解的速度。为了消除算法随机性的影响，对于每个规划场景，SO-RRT\* 算法将运行 100 次，每次运行从自由配置空间中采样 500 次，以充分探索规划空间。同时考虑到每个场景的最优路径长度均不相同，依式5-1对路径长度进行规范化，规范化后收敛曲线的纵坐标为  $\widehat{mLOP}$ 。

$$\widehat{LOP} = LOP / LOGP \quad (5-1)$$

其中，LOP 为给定路径长度，LOGP 为给定规划场景的测地线长度，即起始和目标配置间的 Reed-Shepp 曲线的长度。

分别采用 Greedy 和 Optimal 搜索方式的 SO-RRT\* 算法的收敛曲线如图5-4所示，图中还绘出了标签路径长度 (图中蓝色虚线 Label) 和测地线长度 (图中红色虚线 Geodesic) 两条参考线，绘制前者是为了便于将 SO-RRT\* 的收敛结果与标签路径进行对比，绘制后者是为了便于判断收敛结果与最优解之间的差距。此外，

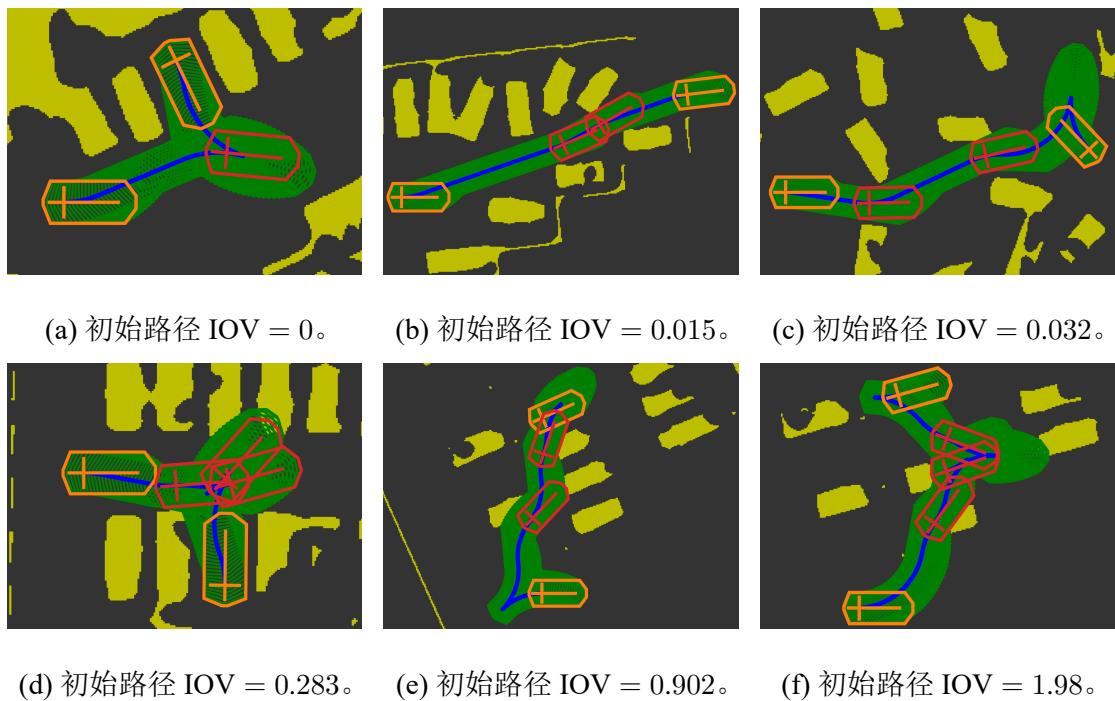


图 5-3 用于评估不同搜索方式对 SO-RRT\* 优化性能影响的 6 种典型的规划场景。从上至下从左至右, YIPS 网络规划的初始路径的 IOV 逐次递增。所有子图中, 黄色色块表示障碍物, 绿色轮廓为初始路径覆盖的区域, 蓝色曲线为初始路径, 橙色多边形和十字形构成起始和目标配置, 十字形的交叉点为车辆后轴中点。红色多边形和十字形构成 YIPS 神经网络预测的采样点。

若曲线在测地线参考线下方收敛, 则说明算法在该场景中失效。由图可见, 对于采用 Greedy 方式能规划成功的场景, 即图 5-3a、5-3c、5-3d 和 5-3f 所示四个场景, 采用 Greedy 方式的收敛曲线越过测地线长度和标签路径长度参考线的速度都快于采用 Optimal 方式的曲线, 这表明单位时间内采样次数更多的 Greedy 方式确实有利于算法以更快的速度探索规划空间。但这四个场景中, 采用 Optimal 方式的收敛速度明显快于采用 Greedy 方式, 也证明了在树中进行全局搜索虽然会减少单位时间的采样次数, 但是能加快 SO-RRT\* 的收敛速度, 提高 SO-RRT\* 优化 YIPS 网络规划的初始路径的能力。

另外, 在图 5-3b 所示的场景中, 虽然路径的碰撞程度不高, IOV 仅为 0.015, 但是由于障碍物紧密分布在采样点附近, 使得采样点附近的可通行区域非常狭窄, 在这种情况下, Optimal 方式由于每次搜索父节点时都会对顶点和采样点间的路径进行碰撞检查, 这一过程相当于在采样点局部区域的内进行了多次的可行

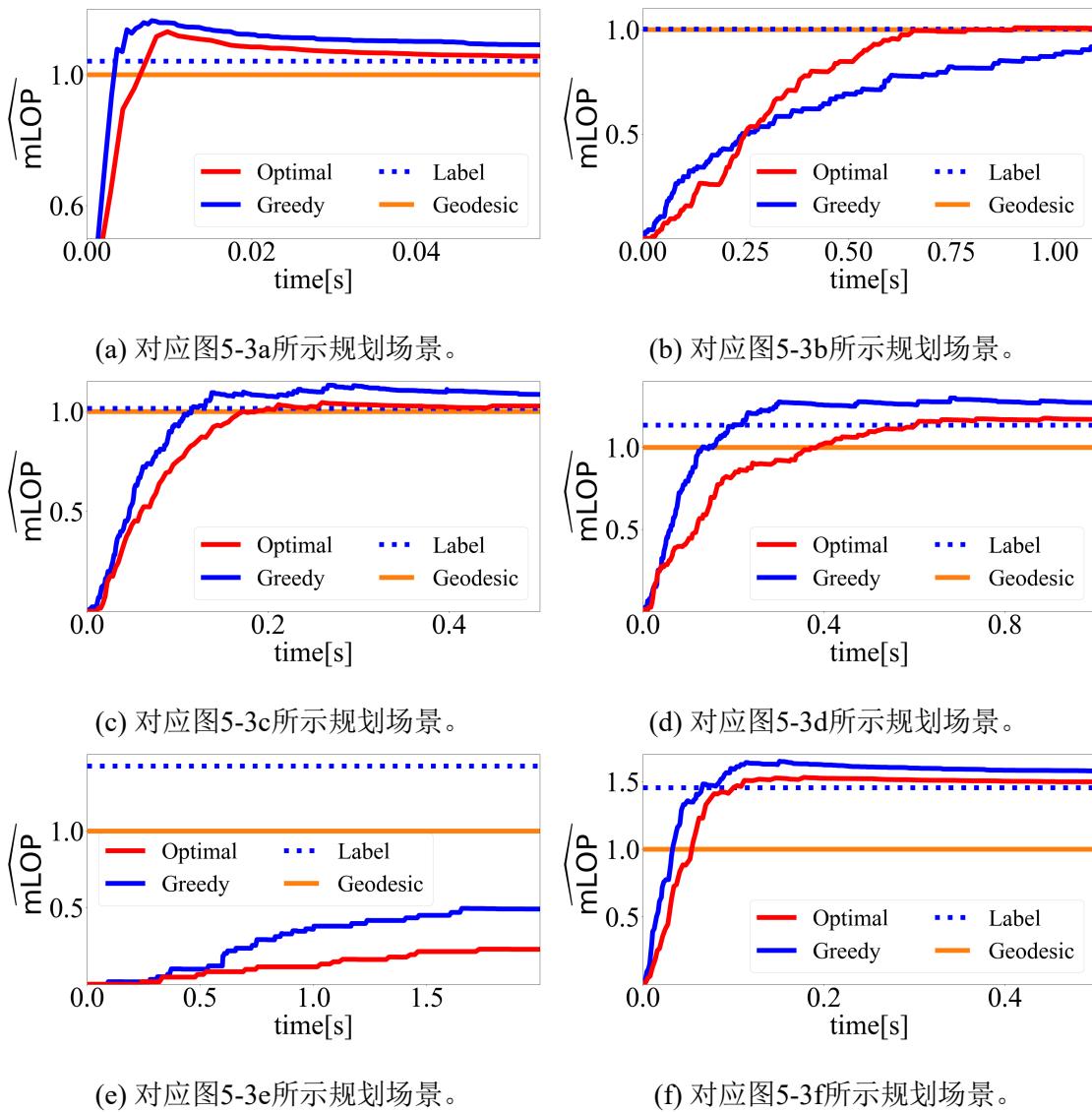


图 5-4 分别采用贪婪 (Greedy) 和最优 (Optimal) 父节点搜索方式的 SO-RRT\* 的规划收敛率曲线对比图。曲线图的纵坐标为依式5-1规范化后的平均路径长度 mLOP，横坐标为规划时间。子图中蓝色虚线 Label 为标签路径长度参考线，橙色实线为测地线长度参考线。

路径搜索，促进了全局搜索可行路径的过程，使得算法得以成功规划，而 Greedy 方式虽然通过牺牲这样的局部搜索过程获得了更快的采样速度，但是由于可行区域的狭窄，许多位置合理的采样点会因为父顶点的不合理导致路径碰撞而被放弃，最终导致算法规划失败。此外，当 YIPS 预测的路径发生严重碰撞时，如图5-3f，只要 YIPS 预测的采样点是合理的，即分布在绕障区域附近，那么两种搜索方式都能快速有效的校正 YIPS 网络的预测偏差，且规划路径长度接近于标签路径，路径示例如图5-5b。但当 YIPS 预测的采样点呈现如5-3e所示的严重偏

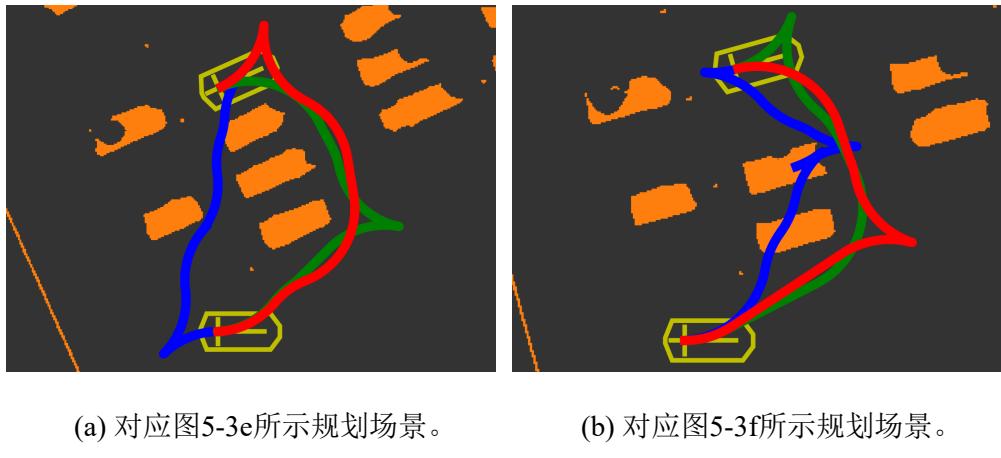


图 5-5 SO-RRT\* 路径算法对 YIPS 网络规划的初始路径的优化结果。所有子图中，橙色色块表示障碍物，黄色多边形和十字形构成起始和目标配置，十字形的交叉点为车辆后轴中点。红色曲线为 SO-RRT\* 的规划路径，蓝色曲线为 YIPS 网络的预测路径，绿色曲线为标签路径。

离时，无论采用何种搜索方式都不能使算法规划成功。但需要指出的是，这里说的无法规划成功，是从概率的角度来说的，即算法在 100 次运行中，规划失败的次数远远大于成功次数，算法无法稳定地规划成功。考虑到随机性，算法在规划场景 5-3e 中还是可以规划出如图 5-5a 所示的成功结果。

因此，综合而言，采用 Optimal 方式更有利于 SO-RRT\* 对初始路径的优化，不仅可以加速算法收敛，而且可以提高算法的优化成功率。

#### 5.4.2 优化性能评估

本小节将对 SO-RRT\* 的路径优化性能进行评估。SO-RRT\* 优化初始泊车路径的过程也可以看做是 YIPS 网络预测的采样点序列启发 SO-RRT\* 采样规划的过程，因此对 SO-RRT\* 优化性能的评估可以采用节 5.2.1 所述的规划性能评价指标。在此基础上，为了将规划结果与标签路径进行对比，将路径长度评价指标与标签路径长度作差，作差后长度通过添加上标  $\Delta$  以区别，如作差后的 mLOP<sub>TTFP</sub>，以  $mLOP_{TTFP}^\Delta$  表示，作差后长度越小表示路径质量越接近标签路径。

SO-RRT\* 对采用不同类型主干的 YIPS 网络的路径优化结果如表 5-7 所示，每个规划场景 SO-RRT\* 只运行一次，最大采样次数为 500。由表可见，尽管 YIPS 网络的规划成功率仅在 22% 左右（见表 5-6），但经过 SO-RRT\* 优化 0.1 s 后，规划成功率均可以达到 90% 左右，如果允许 SO-RRT\* 采样更久的时间，采样次数

表 5-7 SO-RRT\* 对采用的不同类型主干的 YIPS 网络的优化结果。表中的 mLOP 即算法采样 500 次后输出的路径的长度指标，其余规划性能评价指标的说明参见节 5.2.1(实验设置与评价参数说明)。 $mLOP_{TTFP}^{\Delta}$  表示路径长度评价指标  $mLOP_{TTFP}$  减去标签路径平均长度后的差值，以体现规划路径质量与标签路径质量的差距，其他路径长度评价指标同理。YIPS 网络除了以 SVG 主干时以直接融合方式训练，其余情况下均以 Fine-tuning 方法训练。

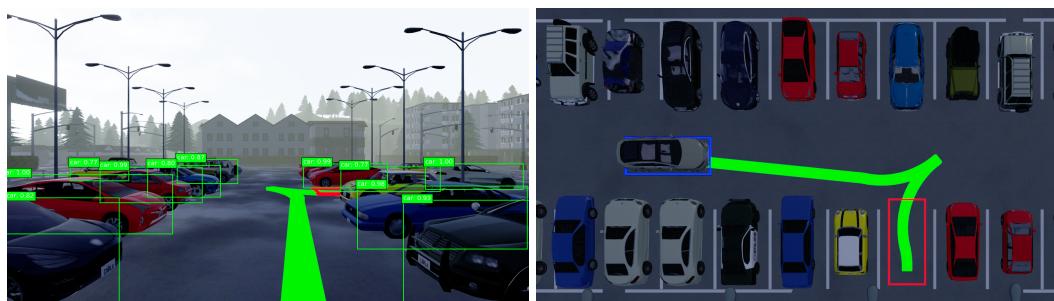
主干网络	$SR_{0.1}$	SR	$mTTFP$	$mLOP_{TTFP}^{\Delta}$	$mLOP_{TTFP.1}^{\Delta}$	$mLOP^{\Delta}$
VGG-19	91.8 %	<b>99.8 %</b>	<b>53.58</b> ms	<b>3.13</b> m	<b>0.92</b> m	<b>0.08</b> m
SVG-16	91.4 %	99.7 %	64.31 ms	3.18 m	0.99 m	0.11 m
VGG-16	<b>91.9</b> %	99.7 %	62.31 ms	3.23 m	0.92 m	0.10 m
ResNet-50	89.2 %	99.5 %	79.22 ms	3.34 m	1.13 m	0.14 m

表 5-8 规划场景的 TTFP 的比例分布表。给定场景的 TTFP 即该规划场景下 SO-RRT\* 规划首条路径的时间。

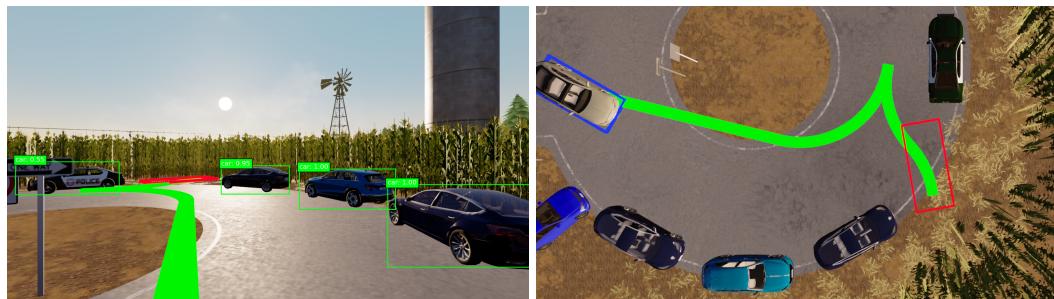
TTFP 取值区间	(0.00, 0.05]	(0.05, 0.10]	(0.10, 0.50]	(0.50, 1.00]	(1.00, 9.00]
规划场景比例	85.4 %	6.4 %	6.3 %	0.9 %	0.9 %

达到 500 次时，规划成功率可以进一步提高到 99 % 以上，由此可见，SO-RRT\* 的校正成功率相当高。另外，SO-RRT\* 的校正速度也相当快，对于以 VGG-19 为主干的 YIPS 网络，其规划出首条路径的平均时间为 53.58 ms，考虑到网络的平均推断时间为 23.23 ms，因此，整体二阶段路径规划的首条路径平均生成时间为 76.81 ms，即 13.55 FPS，满足 10 FPS 的实时性要求。此外，SO-RRT\* 的优化速度也相当快，虽然生成的首条可行路径的长度平均比标签路径要长 3.11 m 左右，但是优化 0.1s 后就可以将长度差缩小到 1.0 m 左右，进一步采样至 500 次，则长度差可以缩小至 0.1 m。

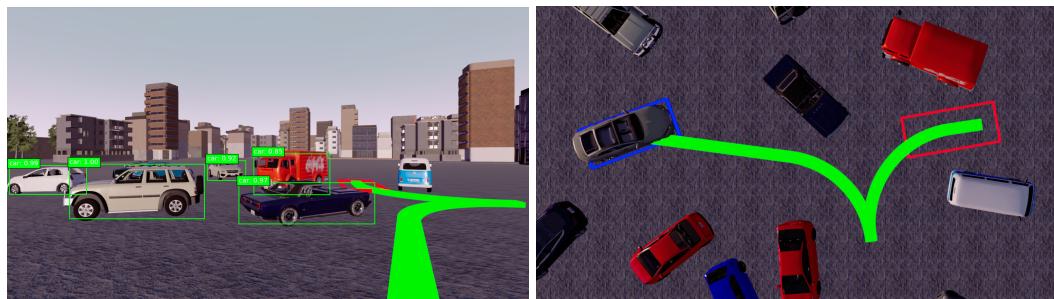
此外，由表还可见，二阶段规划方法的规划性能与 YIPS 网络预测性能是密切相关的，尽管 SO-RRT\* 具有很强的优化能力，但是并不能完全消除 YIPS 网络预测性能不足对方法整体规划性能的影响，如预测性能相对最差的 ResNet-50 为主干的 YIPS 网络，其经优化后的结果也是相对最差的。同时，与 Fine-tuning 方法训练的最好的结果相比，以直接融合方法训练的 YIPS 网络的规划成功率和规划质量并无明显不足，这说明，虽然直接融合方法训练的 YIPS 网络会出现轻



(a) 规则停车场泊车场景下的任务融合输出，左侧为车前视角，右图为俯视角。



(b) 半规则乡村道路泊车场景下的任务融合输出，左侧为车前视角，右图为俯视角。



(c) 无规则开放区域泊车场景下的任务融合输出，左侧为车前视角，右图为俯视角。

图 5-6 不同类型泊车场景下目标检测与路径规划的任务融合输出。所有子图中，带标签绿色框为目标检测网络 SSD 输出的检测框，绿色曲线为二阶段规划方法规划的泊车路径。红色方框为目标停车位置，蓝色方框泊车车辆起始位置。

微的学习能力不足，但通过 SO-RRT\* 的优化，学习能力不足的影响可以完全消除。因此，通过直接融合方法将路径规划任务与目标检测任务进行融合是完全可行的。图5-6中展示了将本文提出的二阶段泊车规划算法与 SSD 目标检测算法进行直接融合，同时进行路径规划和目标检测任务的示例结果。

进一步考察 SO-RRT\* 规划首条路径的时间 TTFP 的分布情况，将所有规划成功的场景按 TTFP 值的大小划分为分成 5 组： $0.00 \text{ s} < \text{TTFP} \leq 0.05 \text{ s}$ 、 $0.05 \text{ s} < \text{TTFP} \leq 0.10 \text{ s}$ 、 $0.10 \text{ s} < \text{TTFP} \leq 0.50 \text{ s}$ 、 $0.50 \text{ s} < \text{TTFP} \leq 1.00 \text{ s}$  和  $1.00 \text{ s} < \text{TTFP} \leq$

9.00 s，每组的场景数量占场景总数的比例如表5-8所示，由表可见，SO-RRT\* 在 85.4% 的规划场景中仅需不足 50 ms 的时间就能规划出首条路径，在超过 90% 的规划场景中只需要少于 0.10 s 的时间就能规划出首条路径。因此整体而言，在 YIPS 网络预测的采样点序列的指导下，SO-RRT\* 在规划空间中寻找可行路径的速度是相当快的，大部分场景下，规划速度都能满足 10 FPS 的实时性要求。但同时，对于占比为 0.9% 的边缘规划场景，SO-RRT\* 的规划时间会超过 1.00 s，甚至在极端情况下，可能需要 10 s 左右才能规划到首条可行路径。

### 5.4.3 最佳拟合高斯模型验证

本小节将验证在 SO-RRT\* 算法偏置采样过程中应用 YIPS 网络配置预测误差的最佳拟合高斯模型的必要性和有效性，并评估不同的高斯模型对 SO-RRT\* 路径优化性能的影响。选取三种不同的高斯模型与最佳拟合高斯模型进行对比，这四种高斯模型的参数如表5-9所示，表中的 Best-fit 模型即为最佳拟合高斯模型，其余三种对比模型分别称为 Larger、Similar 和 Narrower 模型，这三种模型与 Best-fit 模型相比，在配置的  $x$ 、 $y$  和  $\phi$  维度上的方差更大、相近和更小。图5-7展示了这四种高斯模型的  $2 - \sigma$  偏置采样范围。

表 5-9 偏置采样高斯模型的参数表。 $(\mu_x, \sigma_x)$ 、 $(\mu_y, \sigma_y)$  和  $(\mu_\phi, \sigma_\phi)$  分别为高斯模型在配置的  $x$ 、 $y$  和  $\phi$  维度上的(均值, 方差)。

高斯模型	$\mu_x$ [m]	$\sigma_x$ [m]	$\mu_y$ [m]	$\sigma_y$ [m]	$\mu_\phi$ [rad]	$\sigma_\phi$ [rad]
Best-Fit	0.131	2.442	0.071	1.780	-0.029	0.507
Larger	0.000	4.000	0.000	4.000	0.000	1.047
Similar	0.000	2.000	0.000	2.000	0.000	0.524
Narrower	0.000	1.000	0.000	1.000	0.000	0.262

分别采用这四种高斯模型进行偏置采样的 SO-RRT\* 算法的规划性能如表5-10所示。由表可见，采用偏置范围更小的 Narrower 模型可以缩短 SO-RRT\* 规划的首条路径的长度，同时更小的采样范围也有利于 SO-RRT\* 的收敛，在 mLOP<sub>TTFP</sub> 以及 mLOP<sub>TTFP.1</sub> 这两个指标上，采用 Narrower 模型的 SO-RRT\* 均是最优的。但同时，更小的偏置范围会显著的削弱 SO-RRT\* 的校正能力，在仅允许算法规划 0.1s 时，其规划成功率不足 90%，仅为 86.7%。此外，虽然直觉上偏置采样范

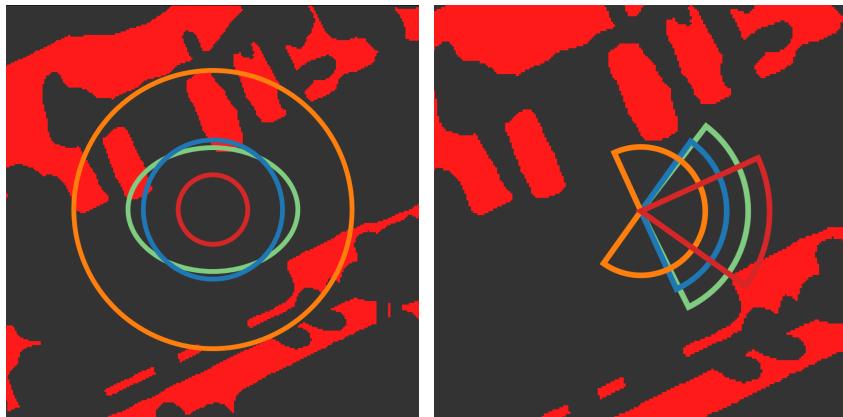
(a) 采样点位置  $(x, y)$  的偏置范围。(b) 采样点朝向角  $\phi$  的偏置范围。图 5-7 表5-9所示的四种高斯模型  $2 - \sigma$  偏置采样范围。Best-fit、Larger、Similar 和 Narrower 高斯模型对应的颜色分别为青蓝色、橙色、蓝色和红色。

表 5-10 采用不同高斯模型进行偏置采样的 SO-RRT\* 算法的路径优化性能对比。表中的 mLOP 即算法采样 500 次后输出的路径的长度指标，其余规划性能评价指标的说明参见节5.2.1(实验设置与评价参数说明)。

高斯模型	$SR_{0.1}$	SR	mTTFP	$mLOP_{TTFP}$	$mLOP_{TTFP.1}$	mLOP
Best-Fit	<b>91.9 %</b>	<b>99.7 %</b>	<b>62.31 ms</b>	25.02 m	22.70 m	21.88 m
Larger	89.2 %	99.8 %	66.35 ms	28.04 m	22.50 m	22.11 m
Similar	91.5 %	99.6 %	67.98 ms	24.94 m	22.05 m	21.88 m
Narrower	86.7 %	97.3 %	124.54 ms	<b>23.66 m</b>	<b>21.97 m</b>	<b>21.84 m</b>

围越大，SO-RRT\* 校正能力会越强，规划成功率会越高，但是实验结果表明，拥有最大偏置范围的 Larger 模型在  $SR_{0.1s}$  的指标上仅比 Narrower 模型好不足 2 %，为四个模型中倒数第二差，而  $SR_{500}$  指标上，虽然是最好的但是也不过比第二好的 Best-fit 模型高 0.1 %，更大的偏置范围所带来的成功率收益相当有限，相反，显著削弱了 SO-RRT\* 寻找首条可行路径的速度，同时，也导致 SO-RRT\* 规划的首条路径的平均长度偏大，达到了 28 m，相比于第二差的 Best-fit 模型长了近 3 m。与 Best-fit 模型偏差范围基本相同的 Simliar 模型在各项规划性能指标上都与 Best-fit 模型相近，虽然规划首条路径的时间比 Best-fit 平均长了 5.67 ms，但 Similar 模型在  $mLOP_{TTFP.1}$  指标上优于 Best-fit。由此可见，并不需要完全依照最佳拟合高斯模型进行偏置采样，利用参数相近的模型来指导偏置采样也能达

到相近的效果，Best-fit 模型并非是最优的偏置采样模型，但为合理配置高斯偏置模型提供了重要的参考。在实际应用中，几乎不可能提前获取 YIPS 网络的所有可能输入，然后通过对配置误差进行建模从而获取完美的偏置模型，通过获取 YIPS 网络在已知数据集上的最佳拟合高斯模型，再以最佳拟合高斯模型为参考设计偏置范围相当的高斯偏置模型是更为合理和可行的选择。

## 5.5 本章总结

本章中，首先将本文提出的二阶段泊车规划方法与先进的基于 OSE 启发和 GBS 启发的 Bi-RRT\* 规划方法进行了比较，结果表明，本文提出的二阶段泊车规划方法在规划成功率、规划首条路径的速度以及规划路径的质量三个方面都优于后两种规划方法，在测试集上，本文提出的二阶段泊车规划方法可在满足平均帧率 10 FPS 的实时性要求的同时达到 99 % 以上的规划成功率。然后对本文提出的 YIPS 网络的初始路径规划性能进行了评估，YIPS 网络的规划速度最快可达 50 FPS，完全满足实时性要求且速度明显快于 OSE 启发式，同时其规划路径中 95 % 以上碰撞面积少于车辆占用区域面积的一半。最后对本文提出的 SO-RRT\* 初始路径优化算法的优化性能进行了评估。SO-RRT\* 在 0.1s 的优化时间内就可以将 YIPS 网络规划的初始路径的可行率提高到 90 % 以上，且优化后路径平均长度与最优值差距少于 1 m。此外，在本章中还验证了在 SO-RRT\* 算法偏置采样过程中应用 YIPS 网络配置预测误差的最佳拟合高斯模型的必要性和有效性。最佳高斯拟合可以有效的指导 SO-RRT\* 的采样过程并提高优化性能，尽管最佳拟合高斯模型并非是最优的偏置采样模型，但为合理配置偏置采样模型提供了重要的参考。

## 第 6 章 总结与展望

本章将对本文工作进行总结，并对本文工作中存在的不足以及可进一步研究之处进行分析和展望。

### 6.1 总结

本文的研究工作的主要贡献在于提出了一种基于深度学习的二阶段自主泊车运动规划方法。在方法的第一阶段，通过设计基于深度卷积神经网络的 YIPS 路径规划网络，实现了端到端的采样点序列预测，即泊车路径规划。在方法的第二阶段，利用基于 Bi-RRT\* 设计的 SO-RRT\* 路径优化算法，对 YIPS 网络规划的初始泊车路径进行优化校正，以实现更高质量的实时泊车规划。此外，本文研究中，还对 YIPS 网络的泛化误差进行了最佳高斯拟合，并利用最佳高斯拟合分布指导 SO-RRT\* 算法的偏置采样过程，最大化 SO-RRT\* 算法的路径优化能力，进一步提高二阶段泊车规划的成功率和规划质量。以下按章节顺序对本文内容进行概括总结。

在本文第 2 章中，对与本文工作相关的理论和技术原理进行了介绍，具体介绍了利用几何代数方法对本文所研究的自主泊车运动规划问题进行规范和简化的过程，并概括叙述了泊车规划中应用基于采样的规划算法的原因和常用的采样规划算法，以及先进的启发式。此外，还简述了工作空间表示和积分曲线求解两个运动规划的重要子问题。最后针对性的介绍了与本文神经网络设计相关的基于深度学习的目标检测算法的原理和网络设计。

在本文第 3 章中，介绍了本文提出的基于深度学习的二阶段自主泊车运动规划方法。首先阐述了第一阶段中用于为泊车车辆规划初始泊车路径的 YIPS 神经网络的设计，包括网络输入输出、网络架构以及损失函数设计等。然后对第二阶段中用于优化初始泊车路径的 SO-RRT\* 算法的设计进行了介绍，并介绍了对 YIPS 泛化误差进行最佳高斯拟合的方法以及在 SO-RRT\* 算法偏置采样过程中应用最佳高斯拟合分布的方式。最后介绍了基于 CARLA 自动驾驶模拟器进行泊车规划场景数据集制作的方法和流程。

在本文第 4 章中，首先对 YIPS 网络的预测性能进行了评估，结果表明 YIPS 网络的采样点类别预测 AP 值可达 97.0，采样点序列的长度预测准确度可达 72%，同时，配置预测误差中不存在固定误差，其分布具有明显的高斯性，可利用最佳拟合高斯分布进行合理建模。然后对 YIPS 网络的几处关键设计进行了验证，包括输入编码方式、网络训练方法、配置损失函数和主干网络选择等。此外，还验证了采用直接融合方法对 YIPS 网络进行训练的可行性。

在本文第 5 章中，首先将本文提出的二阶段泊车规划方法与两种先进的泊车规划方法进行了对比，结果表明，二阶段泊车规划方法在规划成功率、规划首条路径的速度以及规划路径的质量三个方面都更优，可在满足平均帧率 10 FPS 的实时性要求的同时达到 99% 以上的规划成功率。然后对本文提出的 YIPS 路径规划网络的初始路径规划性能进行了评估，结果表明，YIPS 神经网络的推断速度最快可达 50 FPS，规划的路径中 95% 以上碰撞面积少于车辆占用区域面积的一半。最后对 SO-RRT\* 的初始路径优化性能进行验证，结果表明 SO-RRT\* 在 0.1s 的优化时间内就可将初始泊车路径的可行率提高到 90% 以上，且优化后路径平均长度与最优值的差距少于 1 m。此外，实验结果还表明，YIPS 网络预测误差的最佳高斯拟合分布可以为合理配置高斯偏置模型提供重要的参考，采用直接融合方法对目标检测和路径规划任务进行融合是可行的。

## 6.2 展望

在本文提出的二阶段泊车规划方法中，YIPS 路径规划网络直接预测无碰撞可行路径的成功率仍有待提高，在测试集中，其规划路径的可行率不足 25%，若能进一步提高路径可行率，将可以进一步提高方法整体的规划速度。另一个值得进一步研究的问题是，对于 YIPS 网络，为何 VGG 网络作为其主干时，规划效果要明显好于被广泛认为更为先进的 ResNet、DarkNet 和 Xception 网络，这背后的原因仍有待探究。此外，在本文中，将泊车环境简化为静态的环境，如何将方法扩展到复杂的动态泊车环境中也是值得进一步研究的问题。

## 参考文献

- [1] Behere S, Törngren M. A Functional Reference Architecture for Autonomous Driving [J]. Information and Software Technology, 2016, 73: 136–150.
- [2] Paden B, Čáp M, Yong S Z, *et al.* A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles [J]. IEEE Transactions on Intelligent Vehicles, 2016, 1 (1): 33–55.
- [3] Shan Y, Yang W, Chen C, *et al.* CF-pursuit: A Pursuit Method with A Clothoid Fitting and A Fuzzy Controller for Autonomous Vehicles [J]. International Journal of Advanced Robotic Systems, 2015, 12 (9): 134.
- [4] Tian W, Salscheider N O, Shan Y, *et al.* A Collaborative Visual Tracking Architecture for Correlation Filter and Convolutional Neural Network Learning [J]. IEEE Transactions on Intelligent Transportation Systems, 2019.
- [5] Chen L, Shan Y, Tian W, *et al.* A Fast and Efficient Double-Tree RRT\*-Like Sampling-Based Planner Applying on Mobile Robotic Systems [J]. IEEE/ASME Transactions on Mechatronics, 2018, 23 (6): 2568–2578.
- [6] Canny J. The Complexity of Robot Motion Planning [M]. MIT press, 1988.
- [7] Schwartz J T, Sharir M, Hopcroft J E. Planning, Geometry, and Complexity of Robot Motion [M]. Intellect Books, 1987.
- [8] Dorst L, Trovato K. Optimal Path Planning by Cost Wave Propagation in Metric Configuration Space [C]. In Mobile Robots III, 1989: 186–197.
- [9] Kuwata Y, Fiore G A, Teo J, *et al.* Motion Planning for Urban Driving Using RRT [C]. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008: 1681–1686.
- [10] Khaksar W, Sahari K S M, Hong T S. Application of Sampling-based Motion Planning Algorithms in Autonomous Vehicle Navigation [J]. Autonomous Vehicle, 2016, 735.
- [11] Karaman S, Frazzoli E. Sampling-based Algorithms for Optimal Motion Planning [J]. The International Journal of Robotics Research, 2011, 30 (7): 846–894.
- [12] Chen C, Rickert M, Knoll A. Path Planning with Orientation-aware Space Exploration Guided Heuristic Search for Autonomous Parking and Maneuvering [C]. In 2015 IEEE Intelligent Vehicles Symposium (IV), 2015: 1148–1153.
- [13] Chen C, Rickert M, Knoll A. Combining Space Exploration and Heuristic Search in Online Motion Planning for Nonholonomic Vehicles [C]. In 2013 IEEE Intelligent Vehicles Symposium (IV), 2013: 1307–1312.
- [14] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths [J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4 (2): 100–107.
- [15] Caltagirone L, Bellone M, Svensson L, *et al.* LIDAR-based Driving Path Generation Using Fully Convolutional Neural Networks [C]. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017: 1–6.

- [16] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation [C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3431–3440.
- [17] Pfeiffer M, Schaeuble M, Nieto J, *et al.* From Perception to Decision: A Data-driven Approach to End-to-end Motion Planning for Autonomous Ground Robots [C]. In 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017: 1527–1533.
- [18] Chaulwar A, Botsch M, Utschick W. A Hybrid Machine Learning Approach for Planning Safe Trajectories in Complex Traffic-scenarios [C]. In 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016: 540–546.
- [19] Chaulwar A, Botsch M, Utschick W. A Machine Learning Based Biased-sampling Approach for Planning Safe Trajectories in Complex, Dynamic Traffic-scenarios [C]. In 2017 IEEE Intelligent Vehicles Symposium (IV), 2017: 297–303.
- [20] He K, Zhang X, Ren S, *et al.* Deep Residual Learning for Image Recognition [C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770–778.
- [21] Pérez-Higueras N, Caballero F, Merino L. Learning Human-aware Path Planning with Fully Convolutional Networks [C]. In 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018: 1–5.
- [22] Banzhaf H, Sanzenbacher P, Baumann U, *et al.* Learning to Predict Ego-vehicle Poses for Sampling-based Nonholonomic Motion Planning [J]. IEEE Robotics and Automation Letters, 2019, 4 (2): 1053–1060.
- [23] Jordan M, Perez A. Optimal Bidirectional Rapidly-Exploring Random Trees, MIT-CSAIL-TR-2013-021 [R]. 2013. <http://dspace.mit.edu/bitstream/handle/1721.1/79884/MIT-CSAIL-TR-2013-021.pdf>.
- [24] Zeng W, Luo W, Suo S, *et al.* End-to-end Interpretable Neural Motion Planner [C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 8660–8669.
- [25] Dosovitskiy A, Ros G, Codevilla F, *et al.* CARLA: An Open Urban Driving Simulator [C]. In Proceedings of the 1st Annual Conference on Robot Learning, 2017: 1–16.
- [26] Codevilla F, Santana E, López A M, *et al.* Exploring the Limitations of Behavior Cloning for Autonomous Driving [C]. In Proceedings of the IEEE International Conference on Computer Vision, 2019: 9329–9338.
- [27] Shah S, Dey D, Lovett C, *et al.* Airsim: High-fidelity Visual and Physical Simulation for Autonomous Vehicles [C]. In Field and Service Robotics, 2018: 621–635.
- [28] Lozano-Perez. Spatial Planning: A Configuration Space Approach [J]. IEEE Transactions on Computers, 1983, C-32 (2): 108–120.
- [29] Reif J H. Complexity of the Mover’s Problem and Generalizations [C]. In 20th Annual Symposium on Foundations of Computer Science (SFCS 1979), 1979: 421–427.
- [30] Barraquand J, Latombe J-C. A Monte-Carlo Algorithm for Path Planning with Many Degrees of Freedom [C]. In Proceedings., IEEE International Conference on Robotics and Automation, 1990: 1712–1717.
- [31] Lindemann S R, LaValle S M. Current Issues in Sampling-based Motion Planning [C]. In Robotics Research. The Eleventh International Symposium, 2005: 36–54.

- [32] Thrun S. Probabilistic Robotics [J]. Communications of the ACM, 2002, 45 (3): 52–57.
- [33] Dubins L E. On Curves of Minimal Length with A Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents [J]. American Journal of Mathematics, 1957, 79 (3): 497–516.
- [34] Reeds J, Shepp L. Optimal Paths for A Car that Goes both Forwards and Backwards [J]. Pacific Journal of Mathematics, 1990, 145 (2): 367–393.
- [35] Bertolazzi E, Frego M. On the G2 Hermite Interpolation Problem with Clothoids [J]. Journal of Computational and Applied Mathematics, 2018, 341: 99–116.
- [36] Fraichard T, Scheuer A. From Reeds and Shepp’s to Continuous-curvature Paths [J]. IEEE Transactions on Robotics, 2004, 20 (6): 1025–1035.
- [37] Redmon J, Divvala S, Girshick R, *et al.* You Only Look Once: Unified, Real-Time Object Detection [C]. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 779–788.
- [38] Liu W, Anguelov D, Erhan D, *et al.* SSD: Single Shot Multibox Detector [C]. In European Conference on Computer Vision, 2016: 21–37.
- [39] Lin T, Goyal P, Girshick R, *et al.* Focal Loss for Dense Object Detection [C]. In 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 2999–3007.
- [40] Ren S, He K, Girshick R, *et al.* Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [M] // Cortes C, Lawrence N D, Lee D D, *et al.* Advances in Neural Information Processing Systems 28. Curran Associates, Inc., 2015: 2015: 91–99.
- [41] Dai J, Li Y, He K, *et al.* R-FCN: Object Detection via Region-based Fully Convolutional Networks [C]. In Advances in Neural Information Processing Systems, 2016: 379–387.
- [42] He K, Gkioxari G, Dollár P, *et al.* Mask R-CNN [C]. In 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 2980–2988.
- [43] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [C] // Bengio Y, LeCun Y. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
- [44] Lin T-Y, Maire M, Belongie S, *et al.* Microsoft Coco: Common Objects in Context [C]. In European Conference on Computer Vision, 2014: 740–755.
- [45] Deng J, Dong W, Socher R, *et al.* Imagenet: A Large-scale Hierarchical Image Database [C]. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248–255.
- [46] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement [EB/OL]. <https://arxiv.org/abs/1804.02767>, Apr 8 2018.
- [47] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [C]. In Proceedings of the 32nd International Conference on Machine Learning - Volume 37, 2015: 448–456.
- [48] Maas A L, Hannun A Y, Ng A Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models [C]. In Proceedings of the International Conference on Machine Learning, 2013: 3.
- [49] Tan C, Sun F, Kong T, *et al.* A Survey on Deep Transfer Learning [C] // Kúrková V, Manolopoulos Y, Hammer B, *et al.* In Artificial Neural Networks and Machine Learning – ICANN 2018, Cham, 2018: 270–279.

- [50] Howard J, Ruder S. Universal Language Model Fine-tuning for Text Classification [C]. 01 2018: 328–339.
- [51] Abadi M, Agarwal A, Barham P, *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. <https://www.tensorflow.org/>.
- [52] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization [C] // Bengio Y, LeCun Y. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
- [53] Buduma N, Locascio N. Fundamentals of Deep Learning: Designing Next-generation Machine Intelligence Algorithms [M]. O'Reilly Media, Inc., 2017.
- [54] Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions [C]. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 1800–1807.
- [55] Jing Y, Yang Y, Feng Z, *et al.* Neural Style Transfer: A Review [J]. IEEE Transactions on Visualization and Computer Graphics, 2019: 1–1.
- [56] Mordvintsev A, Pezzotti N, Schubert L, *et al.* Differentiable Image Parameterizations [EB/OL]. <https://distill.pub/2018/differentiable-parameterizations>, Jul 25 2018.
- [57] Nakano R. A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Adversarially Robust Neural Style Transfer [EB/OL]. <https://distill.pub/2019/advex-bugs-discussion/response-4>, Aug 6 2019.
- [58] Odena A, Dumoulin V, Olah C. Deconvolution and Checkerboard Artifacts [EB/OL]. <http://distill.pub/2016/deconv-checkerboard>, Oct 17 2016.
- [59] Ilyas A, Santurkar S, Tsipras D, *et al.* Adversarial Examples Are Not Bugs, They Are Features [M] // Wallach H, Larochelle H, Beygelzimer A, *et al.* Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019: 2019: 125–136.

## 攻读硕士学位期间科研成果

### 学术论文:

1. Fan Li, Yunxiao Shan, and Kai Huang, DeepPlanning: Deep Learning-based Planning Method for Autonomous Parking[C], Neural Information Processing Systems Conference, 2020 (投稿中)

### 发明专利:

1. 名称：一种基于深度卷积神经网络的自动驾驶车辆路径规划方法  
申请号：201910537330.7  
发明人：黄凯、**李樊**、单云霄。



## 致 谢

谨此向我的导师致以衷心的感谢和崇高的敬意！本论文的工作是在导师的悉心指导下完成的。

感谢实验室的同学在科研和学习上给予我的帮助。他们开创性的研究拓展了我的学术视野，无数次的探讨使我的研究工作有了长足的进展。

由衷感谢我的室友，与他们的讨论交流使我收益匪浅，是他们令我的学习生活都更加充满动力。

衷心的感谢我的家人对我的关心、支持和理解，没有他们对我的关心、鼓励和支持，我无法顺利完成学业。

最后，感谢所有曾经教育和帮助过我的所有老师。衷心地感谢为评阅本论文而付出宝贵时间和辛勤劳动的专家和教授们！

李樊

二〇二〇年五月二十六日