

分类号 TP957

学号 0123456

U D C

密级 公开

工学硕士学位论文

基于深度学习的气动流场快速精确预测

硕士生姓名 陈东林

学科专业 计算机科学与技术

研究方向 大规模科学与工程计算

指导教师 邓小刚 教授

协助指导教师 高翔 助理研究员

国防科技大学研究生院

二〇二〇年十月

Fast and Accurate Flow Prediction Based on Deep Learning

Candidate: Chen Donglin

Supervisor: Prof. Deng Xiaogang

Associate Supervisor: Associate Prof. Gao Xiang

A dissertation

Submitted in partial fulfillment of the requirements

for the degree of Master of Engineering

in Computer Science and Technology

Graduate School of National University of Defense Technology

Changsha, Hunan, P. R. China

October, 2020

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科技大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目： 基于深度学习的气动流场快速精确预测

学位论文作者签名： _____ 日期： 年 月 日

学 位 论 文 版 权 使 用 授 权 书

本人完全了解国防科技大学有关保留、使用学位论文的规定。本人授权国防科技大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密学位论文在解密后适用本授权书。)

学位论文题目： 基于深度学习的气动流场快速精确预测

学位论文作者签名： _____ 日期： 年 月 日

作者指导教师签名： _____ 日期： 年 月 日

目 录

摘要	i
ABSTRACT	iii
第一章 绪论	1
1.1 研究背景及意义	1
1.2 研究现状	2
1.2.1 基于传统机器学习的气动优化	3
1.2.2 基于深度学习的气动优化	5
1.3 主要工作和创新点	8
1.4 论文结构	9
第二章 研究基础与相关技术	10
2.1 研究问题定义	10
2.2 计算流体动力学基础理论	12
2.2.1 格子 Boltzmann 方法的基本原理	12
2.2.2 雷诺平均 N-S (RANS) 方程	15
2.3 深度学习相关模型介绍	17
2.3.1 卷积自编码网络	17
2.3.2 基于深度学习的图像分割模型	19
2.3.3 Pix2Pix 网络模型	22
2.3.4 图卷积神经网络	23
2.4 本章小结	25
第三章 基于深度卷积网络的流场预测方法	26
3.1 引言	26
3.1.1 动机分析	26
3.1.2 研究思路	26
3.1.3 基于笛卡尔网格的流场数据表示	27
3.2 基于 U-net 网络的气动流场预测	28
3.2.1 稳态流场预测深度模型 (FlowDNN)	28
3.2.2 物理损失函数	30
3.2.3 注意力机制	31
3.2.4 神经网络剪枝	33

3.3 实验设置与结果分析	33
3.3.1 数据集与参数设置	33
3.3.2 实验结果与分析	36
3.4 本章小结	44
第四章 基于图神经网络的流场模拟加速方法	45
4.1 引言	45
4.1.1 动机分析	45
4.1.2 研究思路	45
4.1.3 OpenFOAM 网格数据结构	46
4.2 基于 GCN 网络的气动流场模拟加速	49
4.2.1 图结构数据表示	49
4.2.2 基于 GCN 网络的气动流场模拟加速	50
4.3 实验设置与结果分析	51
4.3.1 数据集与参数设置	51
4.3.2 实验结果与分析	53
4.4 本章小结	53
第五章 总结与展望	55
致谢	56
参考文献	57
作者在学期间取得的学术成果	62

表 目 录

表 3.1 FlowDNN 的网络结构细节	29
表 3.2 模型训练超参数设置	34
表 3.3 不同深度学习方法的各项性能指标比较	36
表 3.4 不同深度学习方法的二维速度场预测结果	38
表 3.5 不同深度学习方法的二维速度场预测结果与真值的差异比较	39
表 4.1 OpenFOAM 网格文件信息	46

图 目 录

图 1.1 气动优化技术的发展历程	3
图 1.2 基于智能学习的气动优化技术	4
图 2.1 利用传统 CFD 方法研究和解决流体问题的流程示意图	10
图 2.2 基于深度学习的气动优化模型工作流程示意图	11
图 2.3 D2Q9 模型和 D3Q19 模型速度离散示意图	14
图 2.4 卷积自编码网络示意图	18
图 2.5 逆卷积操作原理	18
图 2.6 FCN 网络结构图	20
图 2.7 FCN 采用的 skip layer 方法	20
图 2.8 U-net 网络结构图 ^[1]	21
图 2.9 Pix2Pix 网络结构示意图	22
图 2.10 多层 GCN 网络示意图	25
图 3.1 SDF 原理示意图	28
图 3.2 二元法原理示意图	28
图 3.3 FlowDNN 网络结构示意图	29
图 3.4 CAM 和 SAM 结构示意图 ^[2]	32
图 3.5 神经网络剪枝流程示意图	33
图 3.6 训练集和测试集数据样本可视化示例	34
图 3.7 RoI 示意图	36
图 3.8 不同深度学习方法的预测性能指标在测试集上的分布	38
图 3.9 不同损失函数对流场预测结果的影响	40
图 3.10 不同损失函数对流场预测结果物理一致性的影响	40
图 3.11 注意力机制对 MRE _{RoI} 在测试集上分布的影响	41
图 3.12 注意力机制对模型训练的影响	41
图 3.13 ReLU 和 ELU 激活函数	42
图 3.14 不同激活函数对模型训练的影响	43
图 3.15 网络剪枝对 MRE 的影响	43
图 4.1 points 文件结构	47
图 4.2 faces 文件结构	47
图 4.3 面的 owner 与 neighbor 的关系	47
图 4.4 owner 文件结构	48
图 4.5 neighbour 文件结构	48

图 4.6 boundary 文件结构	48
图 4.7 基于 GCN 网络的气动流场加速方法示意图	49
图 4.8 翼型 e342 和翼型 NACA 0012 几何外形示意图	51
图 4.9 实验算例二维流场区域示意图	52
图 4.10 内插数据集上速度场和压力场的模拟结果比较	53
图 4.11 外推数据集上速度场和压力场的模拟结果比较	54
图 4.12 GCN+CFD 方法在不同测试集上的模拟效率	54

摘要

气动优化设计是航空航天飞行器设计等应用的重要组成部分，随着飞行器产业的发展和应对复杂现实环境需求的提升，气动优化往往涉及许多相互交织影响的因素，容易导致设计优化空间“爆炸”。全阶的计算流体力学（CFD）方法能够针对特定流动状态进行高精度的模拟，但往往耗时较长难以进行全面的设计空间探索；虽然基于传统机器学习方法的优化技术在构建代理模型、降阶模型等方面有广泛的应用，但该方法多适用于简单气动优化问题，具有一定的局限性。近年来，深度学习相关研究不断深入并推动诸多应用领域进一步发展，为提升气动优化效率，构建高效、准确的气动评估方法提供了新的途径。本文主要研究内容如下：

- (1) 通过对对比研究气动流场模拟任务和图像回归预测任务的相似性和不同点，本文提出了一种基于深度卷积网络的气动流场预测方法。首先，针对流场数据的表示问题，提出基于笛卡尔网络的流场数据表示方法，将流场数据转化为深度神经网络可以接受的形式。其次，针对边界层等区域流场物理量变化大、包含重要信息多特点，本文提出了嵌入注意力模块的深度卷积网络，能够有效学习该区域流动规律，提升模型预测准确率；基于流体运动遵循的质量守恒定律和动量守恒定律，设计了物理约束的损失函数，有效保证了深度学习预测模型预测结果和 CFD 求解器的物理一致性。为了全面的评估气动流场预测模型的性能，本文定义了新的性能指标，在测试集上的对比试验结果表明该方法能有效提升气动流场模拟效率，同时将预测误差控制在 5% 以内。
- (2) 针对深度卷积网络难以处理非欧式空间流场数据的问题，本文提出了一种基于图神经网络的气动流场模拟加速方法。基于图结构对流场数据进行表示，能够有效减少数据转换过程中的信息丢失，实现流场几何外形、边界条件和流动控制变量的有效融合；基于图卷积网络的流模拟加速模块能够提取流场数据特征，为 CFD 求解器提供接近收敛状态的初始场，减少 CFD 迭代计算的时间。实验结果表明该方法能够在加速气动流场模拟的同时，从根本上保证了模拟结果的有效性。

本文针对飞行器气动优化设计中的典型问题，基于深度学习技术和方法，提出针对流场数据的表示方法，搭建气动流场预测模型和气动流场模拟加速模型，通过嵌入物理约束和融合传统 CFD 方法进一步优化模型预测结果，有效提升气动

流场模拟效率，为基于深度学习的智能气动流场和气动性能预测技术在复杂飞行器气动优化设计中的应用奠定基础。

关键词: 气动优化; 计算流体力学; 深度学习; 图神经网络; 物理一致性

ABSTRACT

National University of Defense Technology is a comprehensive national key university based in Changsha, Hunan Province, China. It is under the dual supervision of the Ministry of National Defense and the Ministry of Education, designated for Project 211 and Project 985, the two national plans for facilitating the development of Chinese higher education.

NUDT was originally founded in 1953 as the Military Academy of Engineering in Harbin of Heilongjiang Province. In 1970 the Academy of Engineering moved southwards to Changsha and was renamed Changsha Institute of Technology. The Institute changed its name to National University of Defense Technology in 1978.

Key Words: Aerodynamic Optimization; Computational Fluid Dynamics; Deep Learning; Graph Neural Network; Physical Consistency

第一章 緒論

CFD 数值模拟在航空航天飞行器气动优化设计等应用中发挥着重要作用，但是全阶 CFD 模拟经济、时间成本高昂，限制了设计人员进行全面、快速的设计空间探索和设计迭代。为了以更省时省力、又快又好地获得最优解，机器学习等智能算法和技术在气动优化的寻优算法及构建代理模型、降阶模型等方面有广泛的应用，但是传统机器学习方法适用于处理低维数据，难以推广到复杂的 CFD 应用场景。随着人工智能领域的快速发展，以深度学习为代表的智能方法和技术在处理高维复杂数据上展现了强大的学习能力，从而为快速、精确气动评估提供了新思路和新方法。

1.1 研究背景及意义

计算流体力学（Computational Fluid Dynamics, CFD）作为了解和探索流体运动的手段，在航空航天，交通运输，石油勘探，天气预测，水利工程和石油化工等工程领域发挥着极其重要的作用。近年来，随着计算机性能的提升，CFD 的应用前景进一步扩大，渗透到生活的方方面面。小到吸管设计，大到汽车制造，生活中随处可见 CFD 的身影。在 2020 年新冠疫情期间，CFD 研究者通过对喷嚏飞沫的流动状态进行研究，得出喷嚏飞沫可以漂浮 8 米远的结论，为疫情防控做出了重要的贡献^[3]。

复杂飞行器气动优化设计是 CFD 的重要应用领域，以增升减阻为目标的气动优化是飞行器设计的重要组成部分。如文献^[4]指出，对于某民航客机，起飞升阻比每提高 1%，可增加 14 个乘客；着陆最大升力系数每增加 1%，则可增加 22 个乘客。由此可见，提升气动优化效率，实现快速、精准的气动评估对于推动飞行器设计发展具有重要意义。尽管传统的全阶 CFD 数值模拟可针对特定状态获得较高精度的评估结果，但是时间、经济等成本巨大。为了以更省时省力、又快又好地获得最优解，许多研究者在优化设计时对原始模型进行简化处理，包括采用更为简单的代理模型^[5] 和降阶模型^[6]，代替复杂的 CFD 评估过程，减少计算开销；引入基于机器学习的智能方法和技术等。因此，在保证模拟精度的基础上，不断地提升气动模拟的效率是数值模拟领域现阶段的研究热点，具有极大的实际应用价值。然而加速和优化气动评估目前面临着以下挑战：

- (1) 气动优化往往涉及许多相互交织影响的因素。以常见的超临界机翼优化为例，除需考虑巡航升阻力系数、升阻比、力矩系数等设计点性能外，抖振、阻力发散等非设计点动态特性也需考虑；同时，优化还存在一些必要约束，如机翼厚度、油箱容积、前后缘装置等。海量设计参数容易导致设计空间

“爆炸”，仅依靠传统的 CFD 数值模拟极大地限制了对复杂飞行器进行全面的设计空间探索。

- (2) 机器学习等智能算法和技术，通过部分或者全部代替复杂的 CFD 评估过程，在原始模型简化处理方面有广泛的应用。然而，大部分机器学习技术借助于发展成熟的机器学习方法和技术，属于浅层学习方法，随着 CFD 所模拟的问题越加的复杂，其预测精度和应用范围相对有限。此外，传统机器学习算法复杂度将随着样本数量和模型精度的提高呈指数级增长，目前应用范围多限于一些容易获得训练样本的二维简单优化问题。

自 2006 年 Hinton 等提出深度信念网络^[7]以来，在 GPU、TPU 等高性能计算机硬件的助力下，以深度学习为代表的智能方法和技术迅速发展，成功应用于图像分类与识别，医疗诊断，视频预测，自然语言处理等诸多领域。在学术界，深度学习算法和技术包括其相关应用也是世界各大研究实验室和顶尖大学的研究热点，比如斯坦福大学、纽约大学、加拿大蒙特利尔大学等成为研究深度学习的重镇，在国际顶尖会议和期刊上涌现了许多关于深度学习的研究内容；在工业界，包括众多互联网公司在内的知名企业专门设立了深度学习实验室，促进深度学习技术转化落地。

由于采用了复杂和更深层次的模型结构，深度学习模型更善于从数据中提取特征，而不是依靠人工构建特征，极大提升了归纳能力，能够自主进行特征选择，可以从大量的候选特征中剔除无用特征再进行回归和分类，具备深层次学习能力，尤其适合于归纳、分析高维、时空相关的流场数据，展现出广阔的应用前景。

一方面，深度学习方法拥有巨大的潜力解决 CFD 领域所面临的问题；另一方面，基于深度学习的气动评估及其在气动优化中的应用尚处于起步阶段，已有应用多为二维简单外形算例，多参考深度学习在其他领域应用较为成熟的方法技术，尤其是深度卷积神经网络在计算机视觉领域的研究成果，深度学习还没有与空气动力学实现交叉融合。因此，研究基于深度学习的气动优化技术对于解决气动优化领域所面临的问题和满足设计空间探索的实际需求具有重大意义。

1.2 研究现状

如图1.1所示，早在 20 世纪 70 年代就出现气动优化的应用，相关研究主要聚焦于使用小扰动方程、全势流等快速方法进行气动评估^[8, 9]，结合梯度类优化算法对气动性能进行优化。在 20 世纪 80 年代，以遗传算法为代表的进化类算法和以蚁群算法与模拟退火为代表的启发式算法逐渐得到应用^[10-12]，这类方法不需要引入梯度信息，因其优化过程具有随机性，使用者无需关心优化的细节，所以在当

时被广泛应用到气动优化过程。但是，进化类算法依赖大量适应度函数分析，优化效率较低。伴随方法^[13]的提出再次推动了基于梯度的优化方法的发展，通过求解伴随场，有效提升了求解梯度的效率。为了结合两类算法的优点，混合算法在2000年以后开始得到广泛应用，实现局部寻优和全局寻优能力的均衡，通过将优化分阶段提高搜索效率等。

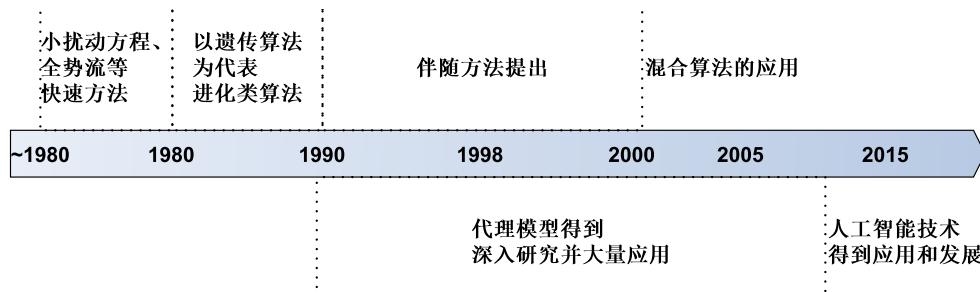


图 1.1 气动优化技术的发展历程

随着机器学习技术和人工神经网络的发展，许多学者对以智能学习为基础的气动流场和气动性能预测方法进行了大量研究，基于智能学习的气动优化也是领域研究热点和本文关注的重点。通过总结相关文献，本文对基于智能学习的气动优化技术和研究进行了整理和分类。如图1.2所示，基于智能学习的气动优化技术概括而言主要有两种：一是利用传统机器学习方法和技术构建优化器、代理模型、降阶模型来减少计算开销，采用数据驱动方式，通过对大量训练样例的“学习”构建数据间内部关系的模型，对未知数据进行预测；二是基于深度学习方法和技术对气动特性和气动性能进行预测，利用深度神经网络强大的归纳学习能力，提取高维、时空相关流场数据中的重要信息，构建气动流场、气动性能快速预测模型。

1.2.1 基于传统机器学习的气动优化

传统机器学习方法在气动优化中最典型的应用就是基于响应面法（response surface methodology, RSM）、多项式回归函数（Polynomial Regression, PR）、径向基函数（Radial Basis Functions, RBF）和人工神经网络（Artificial Neural Networks, ANN）等方法构建模型，快速从已计算的样本中预测设定的目标函数，从而达到代替 CFD 数值模拟的效果。此类优化方法通常被称为基于代理模型的优化方法。

Kanazaki 和 Tanaka 等人^[14]针对飞机降落和接近失速条件下机翼升力系统的

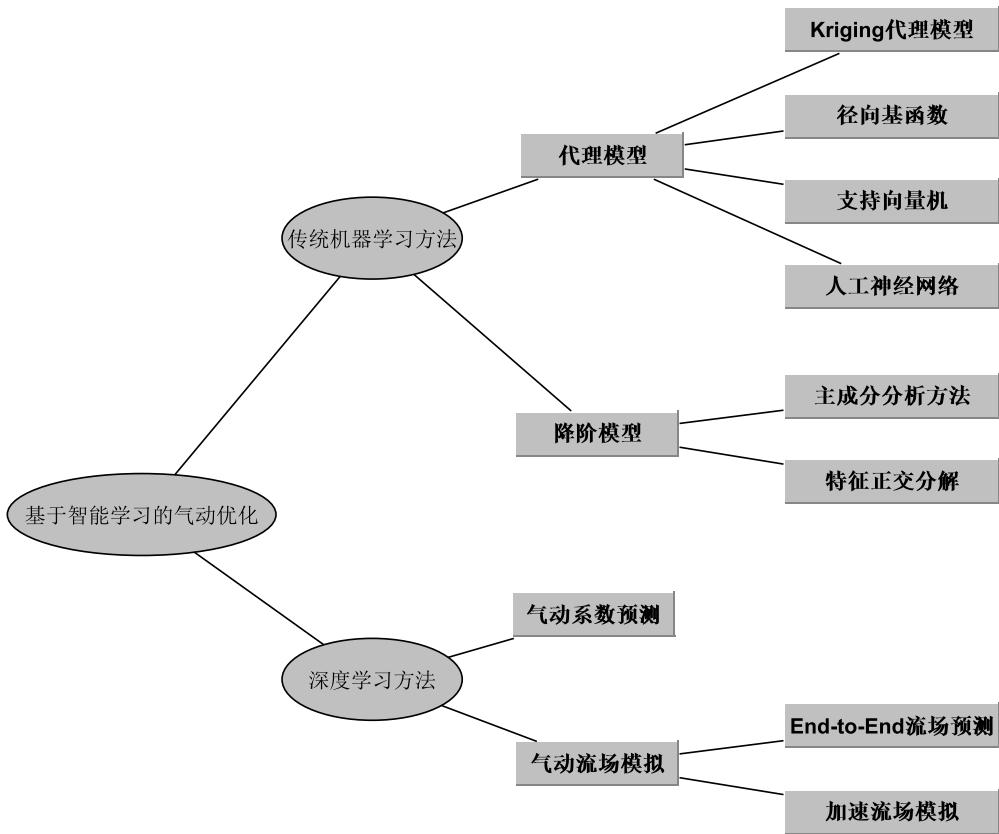


图 1.2 基于智能学习的气动优化技术

设计和优化问题，首先利用基于雷诺平均 Navier-Stokes 的 CFD 方法获取训练样本，设计了基于克里格（Kriging）代理模型的最大化升力系数的目标函数，实验结果表明相对于传统的遗传算法克里格模型显著降低了计算开销，同时实现了更好的性能。此外，作者还利用数据挖掘中的方差泛函分析方法分析了影响升力系统设计的主要因素。

张彬乾和罗烈等人^[15]针对飞翼布局设计中气动与隐身设计矛盾更为突出的问题，采用高精度气动和隐身计算方法，利用 RBF 神经网络代替复杂目标函数的评估，结合遗传算法和松散式代理模型管理方法，建立了翼型多目标优化设计平台。实验结果证明基于 RBF 神经网络的代理模型能有效模拟和评估翼型等几何形状与气动和隐身特性之间存在的强烈非线性关系。

Ju 和 Zhang 等人^[16]为了解决蒙特卡罗模拟方法耗时长、计算开销大的问题，深入比较了人工神经网络、径向基函数和支持向量机模型（Support Vector Machine, SVM）的拟合精度，结果表明 SVM 精度较高，并以此基于 SVM、遗传算法和蒙特卡罗方法针对翼型表面粗糙度的不确定性对翼型升阻系数进行了鲁棒性优化，实验结果表明，遗传算法 - 支持向量回归模型可以很好地捕获从表面粗糙度到机

翼气动性能的不确定性传播。

基于代理模型的方法主要关注在给定输入数据的条件下，模型预测结果的准确性。当研究输入变量之间的关系或者代理模型难以处理高维数据时，往往需要借助以主成分分析（Principal Component Analysis, PCA）和特征向量分解（Poper Orthogonal Decomposition, POD）为代表的机器学习方法。比如，Kapsoulis 和 Dimitrios 等人^[17]针对气动优化中维数灾难问题，整合进遗传算法和 PCA 方法，在交叉和变异阶段使用 PCA 对设计变量进行降维，在降低优化复杂度的同时促使新个体的变化方向沿着方差最大的方向进行，以提高优化效率。此外，PCA 和 POD 还可以用来对偏微分方程组进行降阶，得到降阶模型，再结合基于代理模型的方法，直接对气动流场进行预测。

1.2.2 基于深度学习的气动优化

近年来，以深度学习为代表的人工智能技术取得了令人瞩目的发展和成就，深度学习等智能方法、技术也为快速、精确气动评估提供了新手段。根据气动优化的目标不同，可以将基于深度学习的气动优化技术分为气动系数预测技术和气动流场模拟技术。气动系数预测技术利用深度神经网络对感兴趣的气动系数（比如飞行器升力系数，升阻比等）进行回归预测，不关心流体在流场中运动的细节，属于深度学习在气动优化中的浅层运用。

Zhang 和 Sun 等人^[18]探讨了卷积神经网络技术的适应性以实现空气动力学元建模任务。针对可变的流动条件和物体的几何形状，即具有多种形状的翼型，具有多个流动马赫数，雷诺数和攻角的流动条件，训练了多个卷积神经网络以预测机翼的升力系数。证明利用深度学习对空气动力学任务进行元建模的有效性。将卷积神经网络结果与多层感知机的结果进行比较，发现基于卷积神经网络的元建模方法与多层感知机在学习能力方面具有可比性；更重要的是，在几何表示约束最小的条件下，卷积神经网络模型预测准确性更高。

陈海和钱炜祺等人^[19]提出了一种基于深度学习的翼型气动系数预测方法，有效克服了以往方法依赖翼型设计参数以及算法复杂度随预测精度的提高呈指数级增长等缺点。该方法能够在只给出翼型图像的基础上，使用深度卷积神经网络根据翼型图像对其气动力系数进行了建模。研究结果表明显示模型预测的翼型气动系数获得了较高的精度，说明深度学习在翼型气动系数预测方面具有很好的应用前景。

利用深度学习技术对气动流场进行模拟，研究者可以观察到物理量（比如速度和压力）在流场中的分布，获取更多有用的信息，对飞行器设计等工作进行针对性的优化和调整。本文进一步对基于深度学习的流场模拟进行了分类：基于深

度学习的流场预测方法和基于深度学习的流场模拟加速方法。前者使用深度学习技术全部取代 CFD 过程实现 End-to-End（端到端）流场预测，后者将深度学习方法作为加速模块并与传统的 CFD 求解器进行融合。

在基于深度学习的流场预测方法中，除了在神经网络模型训练阶段，需要利用 CFD 求解器准备训练数据以外，在模型训练完成后，推理过程不再依赖 CFD 方法，深度学习模型可以独立地对给定的几何体和流场条件进行预测。近年来，在基于深度学习的流场预测方向涌现了大量工作。

惠心雨和袁泽龙等人^[20]为了克服传统 CFD 计算需要耗费大量的计算时间与成本的缺陷，提出了一种基于深度学习的非定常周期性流场的预测框架，可以实时生成给定状态的高可信度的流场结果。将条件生成对抗网络（Generative Adversarial Networks, GAN）与卷积神经网络相结合，改进条件生成对抗网络对生成样本的约束方法，建立了基于深度学习策略采用改进的回归生成对抗网络模型，并与常规的条件生成对抗网络模型的预测结果进行对比。研究表明，基于改进的回归生成对抗网络的深度学习策略能准确预测出指定时刻的流场变量，且总时长比 CFD 数值模拟减少至少 1 个量级。

Guo 和 Li 等人^[21]针对非一致稳态层流流动建立了基于卷积神经网络（Convolutional Neural Network, CNN）的预测模型，其深度网络架构中包含了卷积编码和卷积解码（Encoder-Decoder）两部分，采用基于笛卡尔网格计算得到的符号距离函数（Signed Distance Function, SDF）作为输入，以 LBM（Lattice Boltzmann Methods）求解器的模拟结果（速度场）作为训练样本标签。相对于基于二进制的几何外形表示，SDF 值不仅提供了局部几何外形细节，同时包含了全局几何结构的丰富信息。作者针对二维简单形状、包含三维球体和圆柱的管道流等算例速度场进行了训练和预测，获得了较高的精度。深度神经网络预测耗时较 GPU 平台上 LBM 模拟快了 2 个数量级，较 CPU 平台上 LBM 模拟则快了 4 个数量级。该方法可以在早期设计阶段设计迭代中提供快速反馈，为构建实时、交互式设计优化空间探索提供了可能。

在 Guo 和 Li 等人的研究基础上，Bhatnagar 和 Afshar 等人^[22]进行了改进，采用类似的深度神经网络架构，以 3 个翼型、4 个雷诺数、21 个攻角构成的 RANS 模拟结果作为训练数据，预测未知翼型、雷诺数、攻角的测试算例的速度场和压力场。与 Guo 等人的研究不同的是，作者在训练输入中增加了雷诺数、攻角信息（在深度神经网络的全连接层处输入），且在损失函数中引入了 gradient sharpening 和 L1 正则化，提高了模型泛化能力和预测精度。

Miyanawala 和 Jaiman^[23]使用 CNN 预测了不同的阻流体的低雷诺数下的阻流体的空气动力系数。他们提出了一种使用 CNN 和随机梯度下降法的数据驱动方

法，用于在非定常流动问题中对 Navier-Stokes 方程进行模型简化。

Lee 和 You 等人^[24] 使用深度学习技术对在未获知的雷诺数条件下非定常圆柱绕进行了流场预测，考虑到守恒律对预测性能的影响，评测了 GAN 和 multi-scale CNN 等四种深度神经网络的预测效果。实验结果表明，对于短期预测，四种神经网络都实现了较好的预测效果；对于长期预测，考虑物理守恒的 GAN 和 multi-scale CNN 相较于不考虑物理守恒的 multi-scale CNN 实现了更好的性能，不考虑物理守恒的 GAN 实现了在四种深度神经网络中表现最好。该结果证明 GAN 深度神经网络充分利用了无监督训练，因此它们可以应用于先验未知的基础物理学的问题。

Filipe 和 Thomas 等人^[25] 使用图神经网格对流场进行了预测。基于图神经网络的方法利用网格点之间的拓扑信息和网格点上的特征向量作为输入，训练得到基于网格点的流场（包括速度场和压力场）。此外，该方法还采用粗网格上采样的结果作为输入，引入可导的求解器对粗网格点进行调整，增加了模型的泛化能力。

因为在进行预测时不需要进行耗时的迭代计算，流场预测方法的显著特点是预测效率非常高，但是预测结果的有效性即满足流体运动的规律往往难以得到保证。许多研究者针对基于深度学习的预测结果有效性进行了探索。

Ling 和 Templeton 等人^[26] 针对雷诺平均湍流模型提出了一种基于深度神经网络预测模型 TBNN。该架构使用具有不变张量的乘法层来实现将伽利略不变性嵌入预测的雷诺应力张量。实验结果证明了这种神经与通用神经网络相比，预测结果的准确性和物理一致性得到了提升。

Wang 和 Kashinath^[27] 等人提出了一种新颖的混合深度学习模型 TF-Net，该模型将表示学习和湍流仿真技术相结合。TF-Net 利用湍流的多尺度行为来设计可训练的尺度分离算子，以分别对不同尺度范围进行建模。我们提供了 TF-Net 和基线的详尽比较，并观察到了预测误差和所需物理量（包括散度，湍流动能和能谱）。采用不同的评估指标来评估深度学习模型对物理系统的预测性能包括准确性和物理一致性。

在基于深度学习的流场模拟加速方法中，传统的 CFD 求解器仍然参与流场模拟的过程，通过使用深度学习模型减少迭代计算量，达到计算流场模拟的效果。相较于流场预测方法，流场模拟加速方法的模拟效率会有所下降，但是由于 CFD 求解器参与了流场模拟并且输出了最终了模拟结果，使得流场模拟结果的有效性得到了充分的保证。具有代表性的工作是文献^[28]，Octavi 和 Abhinav 等人针对流场模拟提出了一种基于深度学习的加速器 CFDnet。将稳态流场模拟分为三个部分：基于 CFD 求解器的 warmup，基于深度学习模型的 Inference，基于 CFD 求解器的 Refinement。在 CFD 迭代计算过程中初始阶段和最终阶段之间，利用深度神经网络建立映射代替大部分“invalid”迭代计算。在内插数据集和外推数据集上

对 CFDnet 的效果进行了测试，结果表明，CFDNet 满足特定领域物理求解器的收敛约束，同时在层流和湍流模拟方面可达到 1.9-7.4x 的加速比。此外，通过测试 CFDNet 对训练期间看不见的新几何形状的预测来证明其泛化能力。在这种情况下，CFDnet 满足 CFD 求解器的收敛标准，同时收敛速度仍明显优于仅使用传统 CFD 求解器的对照组。

综上所述，相较于传统机器学习方法，基于深度学习的气动特性和气动性能预测具有以下优势：1) 深度模型可以自适应提取输入特征，无需进行人工筛选和精简，可极大地减少设计变量数目限制。2) 深度学习很好地保留了坐标点间的空间相邻关系，其特征提取过程具有空间上的平移不变性和缩放不变性。3) 深度学习由于能够提取更深层次的特征，因而拥有更强大的归纳能力，不仅可以获得气动性能预测，还可以将流场特征（如压力分布等）作为预测输出。但是，使用深度学习技术解决气动流场优化问题，不能是简单的迁移运用，需要根据应用场景的不同，在流场模拟效率、模拟结果精度和有效性等方面进行优化，又快又好地解决气动优化问题。

1.3 主要工作和创新点

本文在深入分析现有基于智能学习的气动优化技术的基础上，受到深度卷积网络和图神经网络在计算机视觉领域成功应用的启发，将其引入来提升气动评估的效率。本文侧重对稳态流动（即达到稳定状态后，流体中压力，速度等物理量都不随时间变化）问题的研究，针对现有工作将深度学习技术应用到气动优化领域的一些不足，在流场数据表示，网络架构搭建与优化，嵌入物理模型约束和性能评价指标等方面展开相关研究。本文贡献了两个测试数据集，用于测试与分析基于不同深度学习网络的气动优化方法，并得到相应结论。总结而言，本文的主要工作和创新点如下：

- (1) 提出了一种基于深度卷积网络 (Deep Neural Networks, DNN) 的气动流场预测方法。首先，针对深度网络的输入数据的表示问题，提出了针对流场流场数据的特征表示方法；利用基于 LBM (Lattice Boltzmann Method) 的数值模拟求解器构造了两种不同流动条件下的数据集；其次，基于图像分割领域的深度网络模型，设计了适用于流场预测的网络架构，同时引入了注意力机制和神经网络剪枝技术，分别对模型的预测精度和推理速度进行了优化；研究流体流动的特性，基于质量守恒和动量守恒的控制方程，设计了嵌入物理模型约束的损失函数。在性能评价指标方面，除了深度学习训练常用的相对评价误差以外，本文提出了新的评价指标用来评估气动预测模型在感兴趣区域 (Region of interest, RoI) 和物理守恒性等方面的预测效果。实验结果表明该

方法能够对给定的输入数据进行快速的预测，相较于 LBM 求解器预测速度提升了四个数量级，同时保证预测误差低于 5%。

- (2) 提出了一种基于图神经网络（Graph Neural Networks, GNN）的气动模拟加速方法。首先，利用网格生成软件生成的网格文件，基于算例的边界条件设置，构造了带有拓扑信息和节点特征信息的图结构输入数据，用于训练图神经网络；其次，利用图卷积网络（Graph Convolutional Network, GCN）构建气动模拟的加速模块。完成模型训练后，将模型的推理结果反馈给 CFD 求解器，为求解器迭代计算过程提供一个更接近于稳定状态的初场，减少 CFD 的模拟时间。实验结果表明，基于 GNN 的气动模拟加速方法能够有效提升流场模拟的效率，此外，通过 CFD 求解器对 GCN 的模拟结果进行优化，保证了模拟结果的预测一致性。

1.4 论文结构

本文主要研究基于深度学习方法和技术的气动流场模拟方法，利用深度神经网络和图神经网络设计实现了气动流场的预测方法和模拟加速方法，论文的主要结构如下：

第一章：主要介绍了传统 CFD 模拟在实际工程应用中遇到的挑战，智能学习方法和技术在气动优化方向的研究现状以及论文的主要工作。

第二章：通过分析传统 CFD 模拟流程定义了研究问题，介绍了计算流体力学相关理论基础；针对第三章、第四章研究内容，重点介绍了相关的深度学习知识与技术，包括常用的深度卷积网络模型和图神经网络模型。

第三章：介绍基于深度卷积神经网络的气动流场预测方法，提出针对流场数据的表示方法。基于图像分割网络设计气动流场预测架构，通过嵌入注意力模块和设计嵌入物理约束的损失函数对流场预测模型进行针对性的优化。介绍利用 CFD 求解器生成实验数据集的流程，定义新的流场预测结果的评价指标。通过实验和结果分析，对模型预测结果的准确率、有效性以及流场模拟效率进行了全面的评估。

第四章：介绍基于图神经网络的气动流场模拟加速方法，提出基于图结构的流场数据表示方法，基于图卷积网络设计气动流场模拟加速模块，利用 CFD 求解器对模型模拟结果进行优化，在提升流场模拟效率的同时保证流场模拟结果的有效性。在测试集上对提出的方法进行实验论证，分析实验结果。

第五章：对本文的工作进行总结和评价，分析现有研究工作的不足和缺陷，对未来的研究方向和研究内容进行展望。

第二章 研究基础与相关技术

近年来，深度学习一直是计算机领域乃至多学科交叉应用领域的研究热点，不仅在计算机视觉方向有着广泛的应用，在光学、材料和航空航天等领域也出现了相关研究。尤其对于大数据应用场景，基于深度学习的方法已经成为重要的数据挖掘手段。对于气动优化乃至整个 CFD 领域而言，深度学习方法在此领域获得发展并取得成功只是时间问题。为了便于阐述本课题工作，本章首先定义了本课题要解决的问题，即明确利用深度学习方法和技术要解决的问题和要达到的效果；其次介绍了 CFD 相关的基础理论，主要包括建立在宏观层次的 Navier-Stokes (N-S) 方程和介观层次的格子 Boltzmann 方法 (Lattice Boltzmann Methods, LBM)；最后介绍了四种与本文工作紧密相关的深度神经网络和图神经网络模型。

2.1 研究问题定义

利用传统 CFD 方法对流体问题（包括气动优化问题）进行研究的流程如图2.1所示。

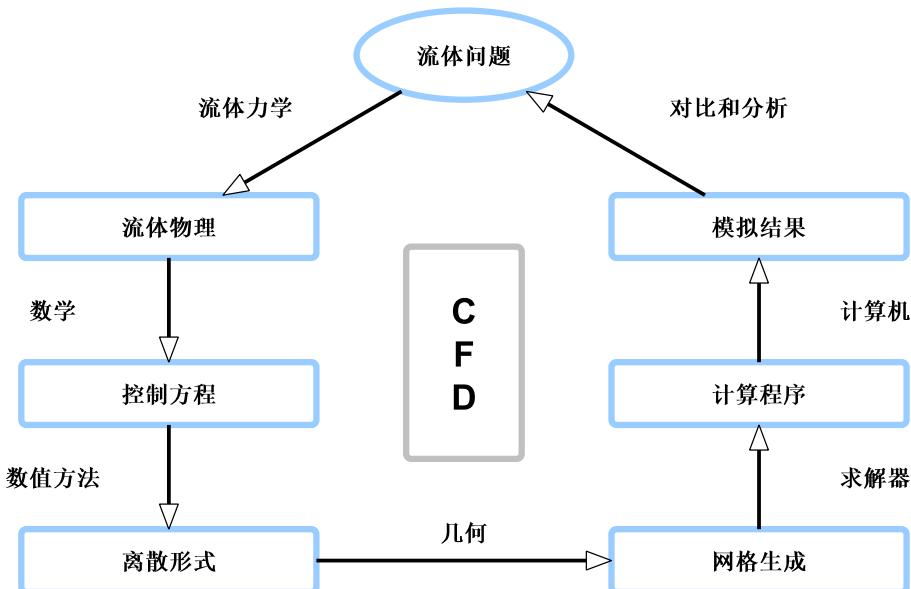


图 2.1 利用传统 CFD 方法研究和解决流体问题的流程示意图

首先利用流体力学知识对流体问题进行理论分析，建立物理模型；根据流体运动遵循的基本规律，利用数学方法推导出流体流动的控制方程，宏观层面上有粘性流动的 Navier-Stokes 方程和无粘流动的欧拉方程；针对特定的流体问题，明确其物理基础和假设，简化控制方程、确定边界条件和初始条件。由于 CFD 方法

是利用数值计算对流体流动进行模拟，必须利用数值方法对时间和空间进行离散，以便进行迭代计算求解。对空间的离散方法包括有限差分方法，有限元方法和有限体积方法；对时间的离散通常包括适用于定常流体问题求解的显示方法和适用于非定常问题的求解的隐式方法。确定离散形式之后，需要对几何进行划分，即生成网格，在网格上对控制方程进行求解。网格生成之后，选择合适的 CFD 求解器，依据初始条件和边界条件对求解器的参数进行设置，最后利用计算机进行迭代计算至模拟结果收敛。

这种全阶 CFD 模拟往往耗时较长，难以满足全面、快速的设计空间探索需求。为了加速 CFD 模拟进程，提升气动优化效率，本文针对二维稳态流动问题，提出了两种基于深度学习的气动优化模型，图2.2展示了这两种模型的工作流程。

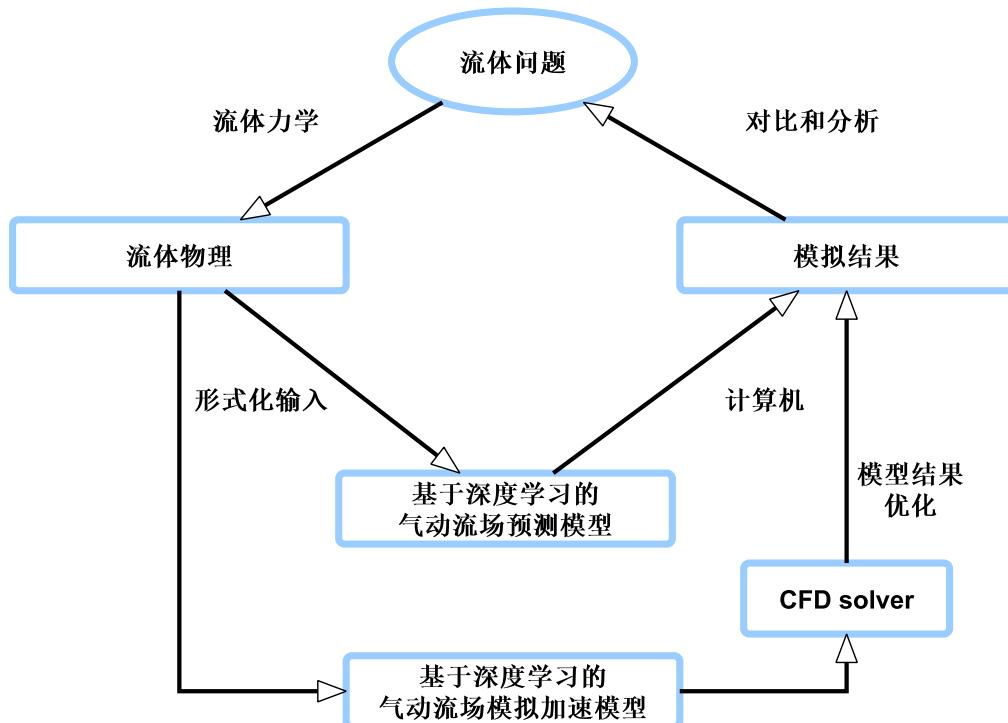


图 2.2 基于深度学习的气动优化模型工作流程示意图

在气动流场预测模型中，将边界条件和几何外形形式化为适合神经网络训练的输入，利用深度学习方法构建端到端的预测模型，通过神经网络正向推理过程，得到对应的气动流场预测结果；气动流场模拟加速模型的流程与气动流场预测模型类似，不同之处在于模型训练完成后，将深度学习模型的结果输入到 CFD 求解器中，为 CFD 求解过程提供一个更加接近收敛状态的初场，从而达到减少迭代计算量、加速模拟的效果。

2.2 计算流体动力学基础理论

根据对液体和气体在不同尺度上动力学规律的描述，计算流体力学方法可分为三类：宏观、介观和微观。

宏观尺度上，假设流体连续地充满整个空间，流体被假设为连续介质。满足质量守恒、动量守恒以及能量守恒；在数学上，流体可由欧拉方程组、N-S 方程组进行描述；在数值计算上，通过各种离散方法将欧拉方程组或 N-S 方程组离散成各种代数方程。介观尺度（分子自由程尺度）上，流体被离散为一系列流体粒子；在数学上，流体由统计力学方程描述；在数值计算上，构造符合一定物理规律的演化机制，通过演化得到与物理规律相符的数值结果。微观尺度上，不再假设流体介质为连续的，通过对每一个分子的运动进行模拟计算，然后在采用不同的方式进行统计平均，以获得流体的宏观运动规律。因为要对每一个分子的运动进行模拟计算，微观层面的方法往往需要消耗大量的计算资源。根据与本文研究内容的相关性，本节重点介基于介观尺度的绍格子 Boltzmann 方法和基于宏观尺度的雷诺平均 N-S 方程。

2.2.1 格子 Boltzmann 方法的基本原理

2.2.1.1 BGK 模型

Boltzmann 方程的基本思想是在任何一个宏观系统中，每一个分子的微观运动都遵循力学规律，因此只要算出大量分子的个别运动就可以确定系统的宏观参数；求出每一分子处于某种状态下的概率，通过统计的方法得出系统的宏观参数。基于以上思想，物理学家 Boltzmann 提出方程的三大假设：

- (1) 分子相互碰撞只考虑二体碰撞，认为 3 个及 3 个以上分子碰撞的概率很小；
- (2) 各个分子的速度分布是不依赖于另外的分子而独立存在；
- (3) 外力不影响局部碰撞的动力学行为。

定义速度分布函数 f 是空间位置矢量 \vec{r} ，分子速度矢量 $\vec{\xi}$ 以及时间 t 的函数。根据 f 的定义有：

$$n = \int f(\vec{r}, \vec{\xi}, t) d\xi \quad (2.1)$$

n 即为 t 时刻， \vec{r} 处单位体积内的分子数。根据假设，速度分布函数 f 可由两项引起改变，第一项是分子的运动，第二项是分子的碰撞，先考虑没有碰撞的情况， $m\vec{a}$ 为作用在每个分子上的外力， m 是分子质量，任意分子，如果在时间间隔 dt

内无碰撞，则分子位置由 \vec{r} 变为 $\vec{r} + d\vec{r}$, 速度变为 $\vec{\xi} + \vec{a}dt$, 则原 t 时刻, 在 $d\vec{r}d\vec{\xi}$ 中的气体分子全部转移到 $\vec{r} + d\vec{r}$, $\vec{\xi} + \vec{a}dt$ 的 $d\vec{r}d\vec{\xi}$ 中, 即有

$$f(\vec{r} + d\vec{r}, \vec{\xi} + \vec{a}dt, t + dt) d\vec{r}d\vec{\xi} = f(\vec{r}, \vec{\xi}, t) d\vec{r}d\vec{\xi} \quad (2.2)$$

在 $(\vec{r}, \vec{\xi}, t)$ 处进行 Taylor 展开, 化简有分子的运动对速度分布函数 f 的影响:

$$\left(\frac{\partial f}{\partial t} \right)_{\text{运动}} = -\vec{\xi} \cdot \frac{\partial f}{\partial \vec{r}} - \vec{a} \cdot \frac{\partial f}{\partial \vec{\xi}} \quad (2.3)$$

考虑分子间的碰撞, 应用刚体碰撞模型, 根据碰撞前后动量守恒和能量守恒可得碰撞对速度分布函数的影响:

$$\left(\frac{\partial f_1}{\partial t} \right)_{\text{碰撞}} = \iint (f'_1 f'_2 - f_1 f_2) d_D^2 |\vec{g}| \cos \theta d\Omega d\vec{\xi}_2 \quad (2.4)$$

其中 $\vec{g} = f_1 - f_2$, f_1 , f_2 为碰撞前分子速度, f'_1 , f'_2 是碰撞后分子速度; d_D 是分子直径, $d\Omega$ 表示球面微元在第一个分子的固定角。

综合公式2.3和2.4有:

$$\left(\frac{\partial f}{\partial t} \right) = \left(\frac{\partial f}{\partial t} \right)_{\text{运动}} + \left(\frac{\partial f}{\partial t} \right)_{\text{碰撞}} \quad (2.5)$$

化简即有:

$$\left(\frac{\partial f}{\partial t} \right) + \vec{\xi} \cdot \frac{\partial f}{\partial \vec{r}} + \vec{a} \cdot \frac{\partial f}{\partial \vec{\xi}} = \iint (f'_1 f'_2 - f_1 f_2) d_D^2 |\vec{g}| \cos \theta d\Omega d\vec{\xi}_1 \quad (2.6)$$

由于碰撞项计算涉及复杂的非线性积分, 所以 Boltzmann 方程难以求解。因此, Bhatnagar, Gross 和 Krook 提出使用一个简单的算子 Ω_f 替代方程2.6右边碰撞项, 称为 BGK 近似模型。最简单的算子可以认为碰撞使 f 趋于平衡分布 f^{eq} 。设改变率与 $(f^{eq} - f)$ 成正比, 系数为 ν , 为碰撞频率即弛豫时间的导数 $1/\tau_0$, 则 Boltzmann 方程简化为:

$$\left(\frac{\partial f}{\partial t} \right) + \vec{\xi} \cdot \frac{\partial f}{\partial \vec{r}} + \vec{a} \cdot \frac{\partial f}{\partial \vec{\xi}} = \nu [f^{eq}(\vec{r}, \vec{\xi}, t) - f(\vec{r}, \vec{\xi}, t)] \quad (2.7)$$

等式右边即为线性算子 Ω_f 。

2.2.1.2 格子 Boltzmann 方程

格子 Boltzmann 方程是 BGK 方程的进一步离散形式, 这一离散形式包括了速度离散、时间离散、空间离散。对于时间离散, 由于微观粒子时刻在做无规则

的热运动，因此微观粒子的速度是连续的，其速度方向和大小有无穷个，但粒子的运动并不会显著影响流体的宏观运动。因此可以将分子速度简化为有限维速度空间， $\{\vec{e}_0, \vec{e}_1, \dots, \vec{e}_N\}$ ， N 表示速度种类数。连续的速度分布函数 f 也被相应离散为 $\{f_0, f_1, \dots, f_N\}$ ，其中 $f_\alpha = f_\alpha(\vec{r}, \vec{e}_\alpha, t)$ ， $\alpha = 0, 1, \dots, N$ 。于是可得离散的 Boltzmann 方程：

$$\frac{\partial f_\alpha}{\partial t} + \vec{e}_\alpha \cdot \nabla f_\alpha = -\frac{1}{\tau_0} (f_\alpha - f_\alpha^{eq}) + F_\alpha \quad (2.8)$$

其中 f_α^{eq} 分子局部平衡分布； F_α 为离散速度空间的外力项。在速度离散的基础上，进一步进行时间离散和空间离散，对公式2.8积分，采用矩形法对公式右边项进行逼近有：

$$f_\alpha(\vec{r} + \vec{e}_\alpha \delta_t, t + \delta_t) - f_\alpha(\vec{r}, t) = -\frac{1}{\tau_0} [f_\alpha(\vec{r}, t) - f_\alpha^{eq}(\vec{r}, t)] + \delta_t F_\alpha(\vec{r}, t) \quad (2.9)$$

方程2.9即为格子 Boltzmann 方程。DnQm 模型^[29]是格子 Boltzmann 方程基本模型， n 表示空间维数， m 表示速度的类型，常见的有 D2Q7 模型、D2Q9 模型、D3Q15 模型和 D3Q19 模型等，图2.3给出了 D2Q9 和 D3Q19 的示意图。

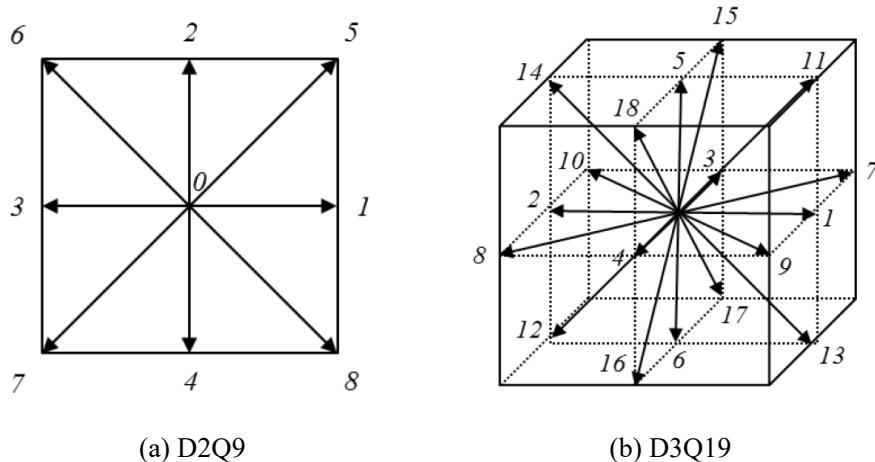


图 2.3 D2Q9 模型和 D3Q19 模型速度离散示意图

D2Q9 模型适用于二维流动问题，将速度根据大小和方向的不同分为了 9 类；D3Q19 模型适用于三维流动问题，速度离散也更加复杂，将速度根据大小和方向的不同分为了 19 类。

LBM 方法相对于宏观方法具有精度高的特点，与传统方法相比，对流项（碰撞项）是线性的，算法简单；由于粒子的运动只有碰撞和迁移，LBM 方法编程容

易；此外，LBM 运算具有局部性，每个粒子只与周围相邻粒子有关，局部运算可同步进行，适合并行计算。因为以上优点，LBM 方法在 CFD 领域应用广泛，利用深度学习技术提升 LBM 方法的模拟效率具有重要意义。

2.2.2 雷诺平均 N-S (RANS) 方程

流体流动一般由以下三个基本定律来控制：(1) 质量守恒定律；(2) 牛顿第二定律；(3) 能量守恒定律。基于这三个基本的物理学定理构建的流动模型，将导出一组控制方程，比如适合粘性流动的 Navier-Stokes 方程 (N-S 方程)。

流体流动有不同的流动状态，当流速很小时，流体分层流动，互不混合，称为层流；当流速逐渐增大，流体的流线开始出现波浪状的摆动，摆动的频率及振幅随流速的增加而增加，此种流况称为过渡流；当流速增加到很大时，流线不再清楚可辨，流场中有许多小漩涡，层流被破坏，相邻流层间出现滑动和混合，这时的流体作不规则运动，这种运动称为湍流。

湍流运动过程十分复杂，在数学上表现出极高的非线性，使用数值模拟方法解决湍流一直是流体力学研究的难点。虽然 N-S 方程能够准确地描述湍流运动的细节，但求解这样一个复杂的方程会花费大量的精力和时间。

目前在 CFD 领域，解决湍流问题的方法主要包括直接数值模拟 (Direct numerical simulation, DNS)、大涡模拟 (Large eddy simulation, LES) 和雷诺平均 (Reynold average Navier-Stokes, RANS)。其中 DNS 和 LES 对网格精细度要求较高，在工程应用上还处于尚不成熟的阶段。为了在精度和效率上满足实际工程需要，研究者提出了基于时均理论的雷诺平均 N-S 模型，RANS 在工程上应用也最多。对于本文研究的稳态不可压流动，在笛卡尔坐标系下有，连续性方程：

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (2.10)$$

对动量方程采用二阶迎风对流格式离散：

$$\begin{aligned} u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\mu}{\rho} \nabla^2 u \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} + \frac{\mu}{\rho} \nabla^2 v \\ u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} + \frac{\mu}{\rho} \nabla^2 w \end{aligned} \quad (2.11)$$

对方程2.10和2.11进行雷诺平均有：

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho \bar{u}_i) = 0 \quad (2.12)$$

$$\frac{\partial}{\partial t} (\rho \bar{u}_i) + \frac{\partial}{\partial x_i} (\rho \bar{u}_j \bar{u}_i) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \bar{\sigma}_{ij}}{\partial x_j} + \frac{\partial}{\partial x_j} (-\rho \bar{u}'_i \bar{u}'_j) \quad (2.13)$$

其中 \bar{u}_i 为雷诺平均速度分量, ρ 为密度, p 为压强, \bar{u}'_i 平均脉动速度, $\partial \bar{\sigma}_{ij}$ 为应力张量分量。

一般地, 在对湍流流动的 N-S 方程平均后, 得到的平均方程中会包含未知的雷诺应力项, 导致了方程求解的不封闭问题。因此需要根据湍流运动规律以寻找附加条件和关系式构建湍流模型使方程封闭。此外, 在进行雷诺平均的过程中损失了部分流动细节, 引入湍流模型对于找回这些细节十分必要。对于 RANS 湍流模拟, 根据 Boussinesq^[30] 假设, 粘性系数由层流部分和湍流部分构成, 即 $\nu = \nu_L + \nu_T$, 其中 ν_L 是层流运动粘性系数 (laminar kinematic viscosity), ν_T 是湍流运动粘性系数 (eddy-viscosity variable), 由湍流模型方程计算得到。

常用的湍流模型可根据所采用的微分方程数进行分类为: 零方程模型、一方程模型、两方程模型、四方程模型、七方程模型等^[31]。本文使用的湍流模型是 Spalart-Allmaras (SA) 模型^[32], 是一种主要用在航空领域的一方程湍流模型, 对墙壁束缚 (wall-bounded) 流动, 显示出很好的效果。SA 模型认为在自由剪切流中能量和信息由大尺度流动流向小尺度流动, 涡粘系数只有产生项和扩散项, 所以满足以下基本运输方程:

$$\begin{aligned} \frac{DF}{Dt} &= \frac{\partial F}{\partial t} + (u \cdot \nabla) F \\ &= \text{Diffusion} + \text{Production} - \text{Destruction} \end{aligned} \quad (2.14)$$

其中 F 外力, diffusion 是扩散项, production 是产生项, destruction 是损失项。对于湍流运动粘性系数 ν_T 应用该运输方程有:

$$\begin{aligned} \frac{D\nu_T}{Dt} &= \frac{1}{\sigma} [\nabla \cdot ((\nu_L + \nu_T) \nabla \nu_T) + c_{b2} (\nabla \nu_T)^2] + c_{b1} \tilde{S} \nu_T - \\ &\quad c_{w1} f_w \left[\frac{\nu_T}{d} \right]^2 \end{aligned} \quad (2.15)$$

其中 σ 普朗特数, c_{b1} 和 c_{b2} 为闭合常数, d 表示到壁面的最短距离。 \tilde{S} 可以通过 d 和速度 u 计算得到, f_w 是一个关于 \tilde{S} 和 ν_T 无量纲函数, 用于解决在边界层外部损失项收敛慢的问题。

虽然许多湍流模型在某些特定的应用场景取得成功, 但至今仍未有一个有效的通用的湍流模型, 这也直接导致了 RANS 缺少普适性。本文首先基于特定流场条件, 模拟生成流场数据集, 然后利用深度学习方法从大量流场数据学习流场运

动规律，最后得到能够进行快速精准流场预测的代替模型。

2.3 深度学习相关模型介绍

深度学习是机器学习的重要分支，自 2006 年提出以来，深度学习理论和技术以及获得了长足的发展，常见的深度学习模型有深度信念网络^[7]，自动编码器^[33]，卷积神经网络^[34]，递归神经网络^[35]，生成对抗网络^[36]等。此外，深度学习在与强化学习、图网络结合方面也非常成功，比较前沿的研究领域有深度强化学习^[37]和图神经网络^[38]等。深度学习的快速发展为解决气动优化提供了新的思路和方法，利用深度学习方法提升气动优化效率的核心思想是基于神经网络构建从输入到输出的映射函数，从而代替或者加速 CFD 求解器的迭代计算过程，不同于常规的图像分类任务，深度神经网络需要在大量数据中学习到从给定输入到对应输出的表示方法。

针对流场数据中的欧式空间数据，经过类比分析，我们发现图像处理中的 image-to-image^[1, 39-42] 的转换方法尤其适用于气动优化场景。对于非欧式空间数据，本文引入了基于图神经网络的架构进行模型训练。如何将流场数据处理成为深度神经网络可接受的形式将在第三章详细阐述，本节重点介绍三类用于图像回归任务的深度神经网络和图神经网络中的图卷积网络。关于深度学习的其他基础理论知识，包括网络基础结构单元，损失函数，优化算法等，可参见文献^[43]。

2.3.1 卷积自编码网络

卷积自编码网络（Convolutional Autoencoders）是自编码网络^[33]的变体。自编码网络及其变体都有类似的网络结构：编码器和解码器。传统的自编码器是一种无监督学习算法，数据没有标签，输入数据 X 经过编码器处理得到输入数据的特征表示 z ，编码结果经过解码器得到输出 X' ，具体过程可以表示为：

$$z = g(X; \phi) \quad (2.16)$$

$$X' = f(z; \theta) \quad (2.17)$$

其中 $g(\bullet; \phi)$ 和 $f(\bullet; \theta)$ 分别表示编码器和解码器， ϕ 和 θ 是相应的参数。

损失函数一般可以定义为输入 X 和输出 X' 的最小均方误差（Mean Squared Error, MSE）：

$$Loss_{MSE} = \min_{\phi, \theta} \|X - X'\|_2^2 \quad (2.18)$$

一般而言， z 的维度远小于输入 X 的维度，网络通过这样编码和解码的方式，实现对输入数据的降维且尽量不损失数据信息。

但是传统的自编码器在处理图片格式数据时，由于采用了全连接操作，忽略了图像中的空间关系，数据切片和数据堆叠会导致信息大量丢失。为了克服这一缺点，卷积自编码网络采用卷积层来构造自编码器。图2.4是卷积自编码网络的示意图，深色部分代表编码器，通常由卷积层和池化层构成，卷积层负责信息提取，池化层负责空域下采样；浅色部分代表解码器，由卷积层和上采样层构成，有时也利用逆卷积操作替代卷积和上采样操作进行原始信息的复原。

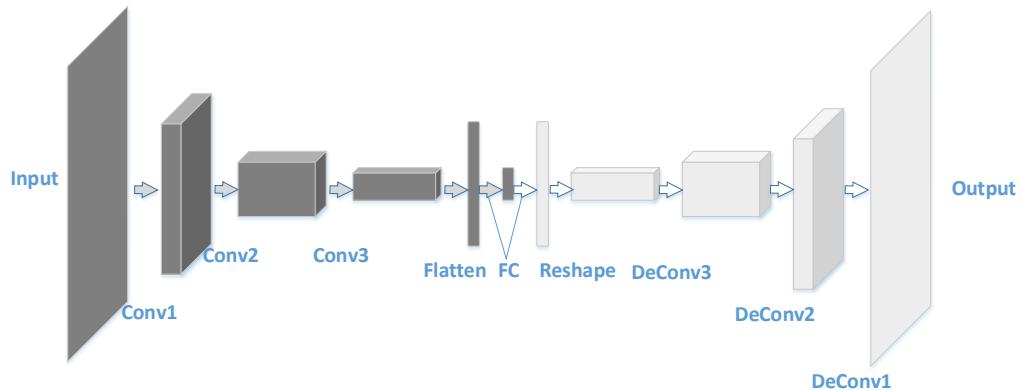


图 2.4 卷积自编码网络示意图

逆卷积操作原理如图2.5所示，逆卷积操作可以看出是常规卷积操作的逆过程，不同点在于，为了还原原始输入的尺寸，通常需要进行填充（padding）操作（如图2.5(b)中的空白区域）。

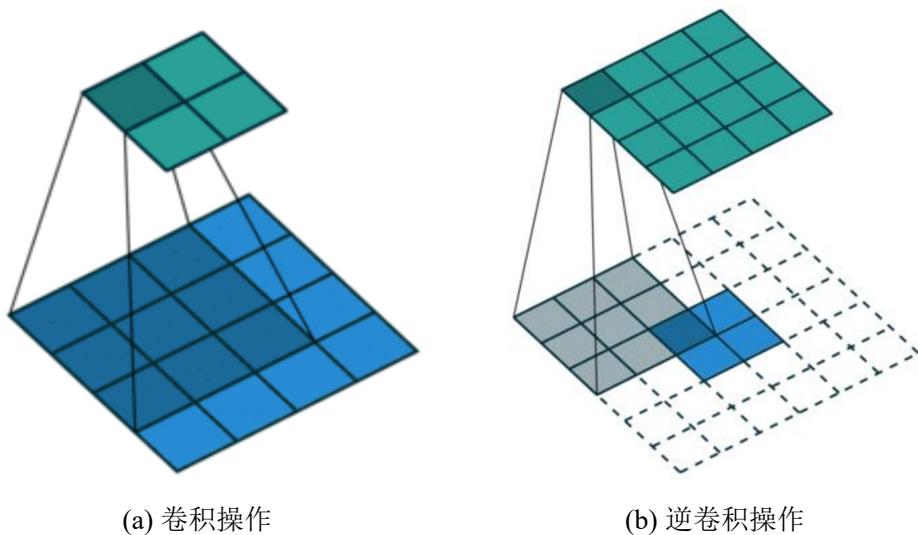


图 2.5 逆卷积操作原理

对于无监督学习，在卷积神经网络的编码器和解码器的衔接处，利用全连接

层，也可以提取带图像数据特征表示；在进行有监督学习时，网络更关注解码器的输出 Y' 而不是 z ，损失函数转化为：

$$Loss_{MSE} = \min_{\phi, \theta} \|Y - Y'\|_2^2 \quad (2.19)$$

从而可以利用卷积神经网络进行有监督学习任务，在气动流场模拟领域已有基于卷积自编码网络开展的工作^[21]。

2.3.2 基于深度学习的图像分割模型

图像分割一直是计算机视觉领域的难题，也是该领域的研究热点。所谓图像分割是指根据灰度、彩色、空间纹理、几何形状等特征把图像划分成若干个互不相交的区域，使得这些特征在同一区域内表现出一致性或相似性，而在不同区域间表现出明显的不同。传统的方法有基于阈值的分割方法；基于区域的图像分割方法；基于边缘检测的分割方法；基于小波分析和小波变换的分割方法；基于遗传算法的分割方法等^[44]。

近年来，深度学习方法开始应用到图像分割，通过搭建神经网络，对训练样本进行有监督学习，得到图形分割的模型。根据分割应用任务不同，图像分割可分为普通分割、语义分割和实例分割。其中：普通分割是指对分属不同区域的像素点进行分类；语义分割是在分类的基础上识别出每一块区域的语义；实例分割在语义分割的基础上，进一步对每个识别目标进行编号。本节重点介绍两种经典的语义分割网络：全卷积网络和 U-net 网络，分别适用于自然图像分割和医疗图像分割。

2.3.2.1 全卷积网络

2015 年 Long 等人提出的全卷积网络（Fully Convolutional Networks, FCN）用于图像语义分割^[45]。自从提出后，FCN 已经成为语义分割的基本框架，后续算法都借鉴了该框架的思想。

如图2.6所示，FCN 参考了图像分类网络中的 VGG16^[46] 网络架构。图像分类网络只能对整个图片进行分类而不能识别每个像素点的类别。为了实现逐像素分类的目的，FCN 用卷积层替换了 VGG 网络中的全连接层，最后利用逆卷积的上采样方法将特征图恢复成原始图片大小，从对整张图片的稀疏分类转换成对每个像素点进行密集分类（dense prediction），达到图像分割的目的。

FCN 的另一个特点是利用了全局信息和局部信息。经过多次卷积和池化操作以后，得到的图像越来越小，分辨率越来越低，最后产生了高维特征图。如果直接对进行上采样至原始图片大小，会产生模糊的分割结果。为了解决这一问题，FCN 使用了如图2.7所示的 skip layer 的方法。

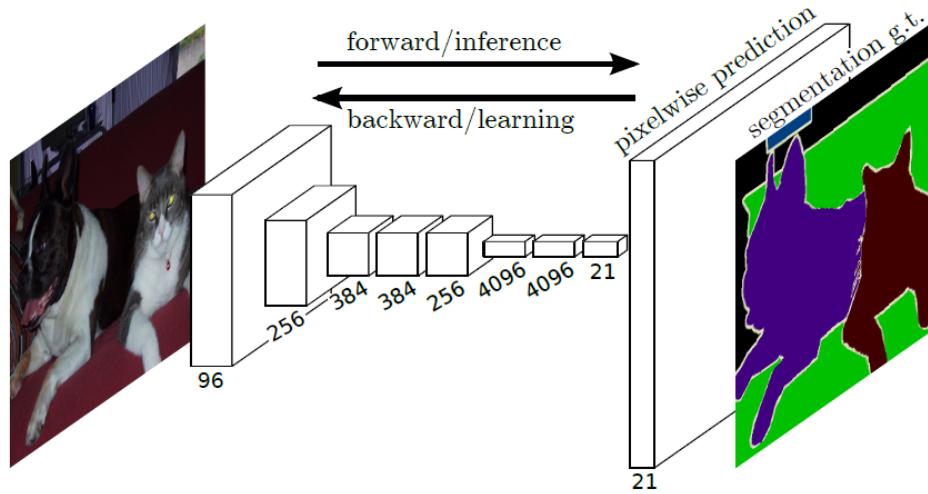


图 2.6 FCN 网络结构图

对于 FCN-32s，高层得到的粗糙层（conv7）进行 32 倍上采样操作，再对每个点进行 softmax 逻辑回归处理，得到每一个像素点的分类。

对于 FCN-16s，先将 conv7 的结果进行 2 倍上采样，再将上采样结果与浅层的精细层（pool4）进行逐点相加，最后进行 16 倍的上采样操作得到与原始输入尺寸相同的图像分割结果。

对于 FCN-8s，先将 conv7 的结果进行 4 倍上采样，将 pool4 的结果进行 2 倍上采样，再将上采样结果与更浅层的精细层（pool3）进行逐点相加，最后进行 8 倍上采样操作和 softmax 处理。

通过 skip layer 的方法，融合多层特征图，有效整合了粗粒度的语义信息和细粒度的位置信息，有利于提高分割准确性。尽管 8 倍上采样的 FCN 的分割效果已经有了很大提升，但是结果还是比较模糊，对分割区域边界不敏感；此外，对每个像素点单独进行分类，没有充分考虑图像的空间上的联系。

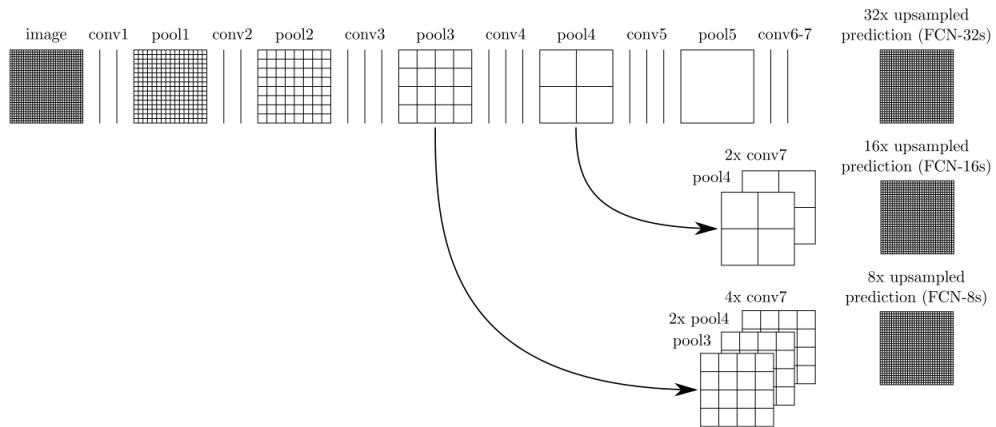


图 2.7 FCN 采用的 skip layer 方法

2.3.2.2 U-net 网络

2015 年，Ronneberger 等人^[1]提出了 U-net 网络结构，U-net 是基于 FCN 的一种语义分割网络，适用于做医学图像的分割。U-net 修改并扩展了 FCN 网络结构，使它在使用少量的数据进行训练的情况下获得精确的分割结果。其主要思想是在下采样网络的后面补充一个对称的上采样网络，多个上采样层增加了网络的参数和表示能力，有利于提升输出结果的分辨率。对称的网络结构形似英文字母“U”，所以被称为 U-net。

U-net 网络结构如图2.8所示：蓝 / 白色框表示特征图；蓝色箭头表示 3x3 卷积层，用于特征提取；灰色箭头表示 skip-connection，用于特征融合；红色箭头表示池化层，用于降低维度；绿色箭头表示上采样 upsample，用于恢复维度；青色箭头表示 1x1 卷积，用于输出结果。其中灰色箭头中的 copy 就是 atenate 操作，将深层的特征图和浅层的特征图在通道方向拼接；crop 是为了让两者的长宽一致，保证拼接后的特征图大小一致。

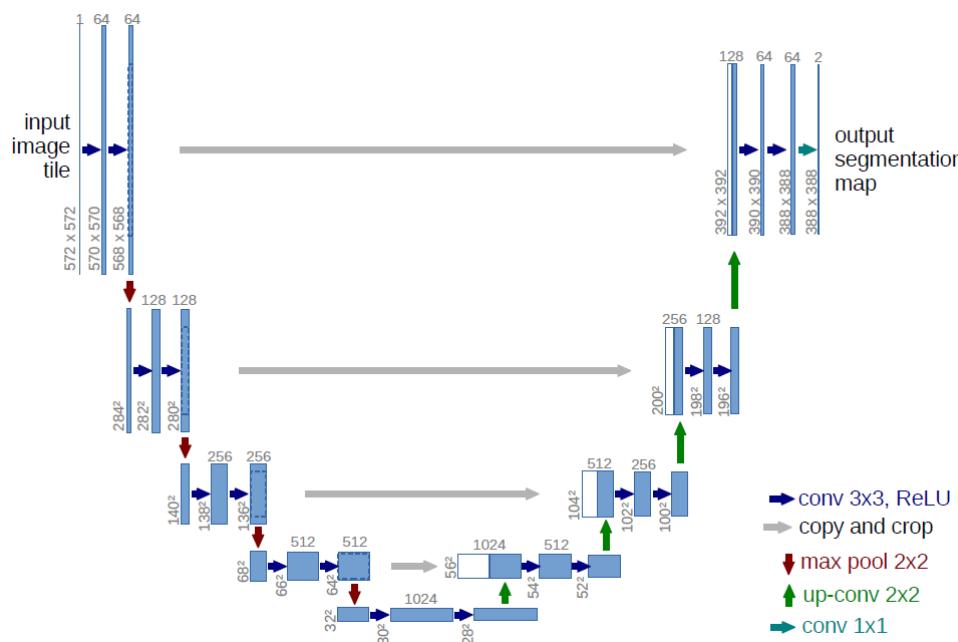


图 2.8 U-net 网络结构图^[1]

U-net 主体结构包括一个捕获上下文信息的收缩路径和一个允许精确定位的对称拓展路径：收缩部分和扩展部分都有 4 个采样层。这种架构延续了 Encoder-Decoder 的思想。Encoder 由卷积操作和下采样操作组成，文中所用的卷积结构统一为 3x3 的卷积核，填充为 0，步长为 1；Decoder 部分采用的上采样的方式与 FCN 中的反卷积不同，为双线性插值。此外，为了更好融合位置信息和语义信息，U-net 拓展了 FCN 中 skip layer 的思想，在“U”形网络的对称部分添加了 skip

connection。与 FCN 的加操作不同，U-net 使用了叠操作（concatenation）增加了特征的厚度，保留了更多浅层的位置信息，这使得上采样层可以在浅层特征与深层特征在训练时进行自适应选择，这对语义分割任务来说更有优势。

由于在医疗图像分割上取得巨大成功，许多研究者针对不同的图像分割任务对 U-net 进行了改进，产生了许多 U-net 变体包括 V-Net^[47]、UNet++^[48]、U-NetPlus^[49] 和 3D U2-Net^[50] 等，提升了模型的推理速度和精度。

2.3.3 Pix2Pix 网络模型

对于气动流场预测，核心问题是实现从输入到输出的映射。除了前文提到的自编码网络和图像分割网络，基于生成对抗网络（Generative Adversarial Network, GAN）的图像风格迁移模型也在解决像素到像素的映射问题上展示出的强大潜力。GAN 在图像生成、风格迁移、超分辨重建等领域已经得到了广泛的应用，本节我们重点介绍 Pix2Pix 网络模型^[40]。

Pix2Pix 网络基于条件生成对抗网络（Conditional Generative Adversarial Network, cGAN）^[51]，通过添加条件约束信息来指导完成图像转换任务，比如从标签图合成相片，从线稿图重构对象，给图片上色等。和传统的 GAN 网络类似，在 Pix2Pix 模型训练过程中，迭代地训练生成器和判别器，生成器尽可能生成接近真实的样本，企图“欺骗”判别器；判别器尽可能识别出真实的样本和生成的样本，获得更高的得分。这样的对抗训练过程类似博弈游戏，随着训练的进行，生成器和判别器的能力不断提升直到达到令人满意的效果。由于添加了约束条件，Pix2Pix 网络的输入不再是普通 GAN 网络中的随机变量，其网络结构示意图如图2.9所示：

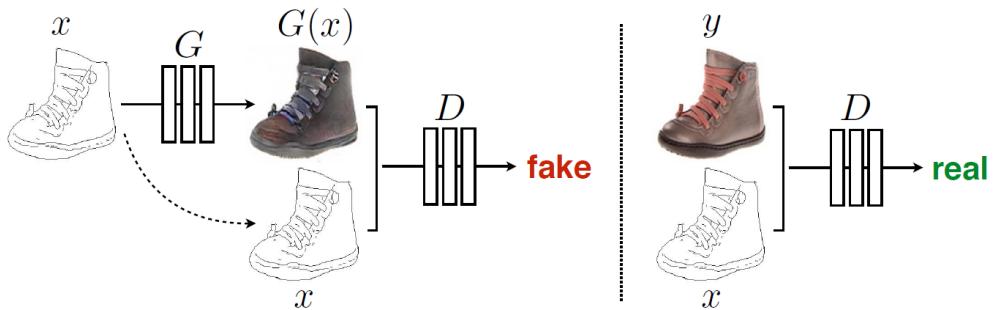


图 2.9 Pix2Pix 网络结构示意图

其中 x 是条件输入， y 是对应的标签，每个输入 x 唯一对应一个标签 y 。在训练生成器时， x 进过生成器 G 得到生成图像 $G(x)$ ；在训练判别器时，输入 x 和对应生成图像 $G(x)$ 或标签 y 通过通道维度的叠操作进行拼接，一同送入判别器，判别器会输出概率值，表示是否为一对真实样本。

Pix2Pix 网络利用类似 U-net 的结构代替自编码网络作为生成器，在输入和输出之间存在很多可以共享的低级信息，采用 skip-connection 的结构有利于传递这些底层信息和重构图像。对于判别器网络，Pix2Pix 提出了 PatchGAN 架构。传统 GAN 判别器通常对生成样本整体进行判断，对于图片而言，直接输出整张图片是真实样本的概率。而图像转换任务中关注像素到像素的转换效果，所以在这里提出了分块判断的算法，在图像的每个块上去判断是否为真，输出平均预测结果。

条件 GAN 采用的损失函数通常为：

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) = & \mathbb{E}_{x,y}[\log D(x, y)] + \\ & \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]\end{aligned}\quad (2.20)$$

$$\mathcal{L}_{cGAN}(G) = \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \quad (2.21)$$

其中 z 为输入随机变量，在 Pix2Pix 网络输入仅为 x 。此外，为了保证像素级低频信息的预测精度，Pix2Pix 网络的损失函数还引入了 L_1 损失项：

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x_1, x_2, y} [\|y - G(x_1, x_2)\|_1] \quad (2.22)$$

考虑到训练的最终目标是获得一个性能良好的生成器，同时生成器和判别器进行着对抗的训练，所以训练的最终目标是：

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (2.23)$$

其中 λ 是 L_1 损失函数项的权重系数。

2.3.4 图卷积神经网络

在深度学习发展过程中，卷积神经网络（Convolutional Neural Network, CNN）和循环神经网络（Recurrent Neural Network, RNN）在图像识别、语义分割、自然语言处理等领域发挥了重要作用。无论是图像还是语言，在 CNN 或 RNN 进行处理时都被转换成结构规则的数据形式，都属于欧式空间的数据。然而现实生活中有很多不规则的数据结构，如社交网络、化学分子结构、生物网络等；即使是语言，其内部实际是复杂的树形结构，只是 RNN 将其处理成向量的形式便于神经网络训练。图结构通常被用来表示这些非欧式空间的数据，但是在图结构中，每个节点周围的结构千差万别，不同于图像和向量化的语言数据具有平移不变性，因此不可能直接利用 CNN 或 RNN 进行处理。

近年来，人们对深度学习方法在图上的拓展进行了大量的研究，研究人员借鉴

CNN、RNN 以及图嵌入技术的思想，定义和设计了专用于处理图结构数据的神经网络，即图神经网络（Graph Neural Network, GNN）。文献^[52] 将 GNN 分为五大类：图卷积网络（Graph Convolution Networks, GCN）、图注意力网络（Graph Attention Networks）、图自编码器（Graph Autoencoders）、图生成网络（Graph Generative Networks）和图时空网络（Graph Spatial-temporal Networks）。本节重点介绍图卷积神经网络。

GCN 方法主要分为两大类，基于谱（spectral-based）和基于空间（spatial-based）。基于谱的方法将空域信号利用傅里叶变换转换到频域进行分析处理，完成在空域无法完成的操作。对于无向图有正则化拉普拉斯矩阵：

$$\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad (2.24)$$

\mathbf{I}_n 为单位矩阵， \mathbf{A} 为图的邻接矩阵， \mathbf{D} 为对角矩阵。对 \mathbf{L} 进行特征值分解有：

$$\mathbf{L} = \mathbf{U} \Lambda \mathbf{U}^T \quad (2.25)$$

\mathbf{U} 是特征向量构成的矩阵， Λ 是对角矩阵，对角线上的值为 \mathbf{L} 的特征值。对输入图信号 \mathbf{x} 的傅里叶变换和傅里叶逆变换分别被定义为

$$\mathfrak{F}(\mathbf{x}) = \mathbf{U}^T \mathbf{x} \quad (2.26)$$

$$\mathfrak{F}^{-1}(\hat{\mathbf{x}}) = \mathbf{U} \hat{\mathbf{x}} \quad (2.27)$$

$\hat{\mathbf{x}}$ 为傅里叶变换的结果。定义对输入图信号的图卷积操作为：

$$\begin{aligned} \mathbf{x} *_G \mathbf{g} &= \mathfrak{F}^{-1}(\mathfrak{F}(\mathbf{x}) \odot \mathfrak{F}(\mathbf{g})) \\ &= \mathbf{U} (\mathbf{U}^T \mathbf{x} \odot \mathbf{U}^T \mathbf{g}) \end{aligned} \quad (2.28)$$

其中 \mathbf{g} 是定义的滤波器， \odot 表示哈达玛积运算。基于频谱的图卷积网络都遵循这样的模式，它们之间关键的不同点在于选择的滤波器不同。基于谱的方法从图信号处理的角度引入滤波器来定义图卷积，根据图谱理论和卷积定理，将数据由空域转换到频域做处理，理论基础坚实。基于空间的方法将图卷积表示为从邻域聚合特征信息，直接在图空间上定义卷积操作，灵活性更强。

对于图卷积算法的实现主要包括三个步骤：i) 发送：每一个节点将自身的特征信息经过变换后发送给邻居节点。这一步是在对节点的特征信息进行抽取变换；

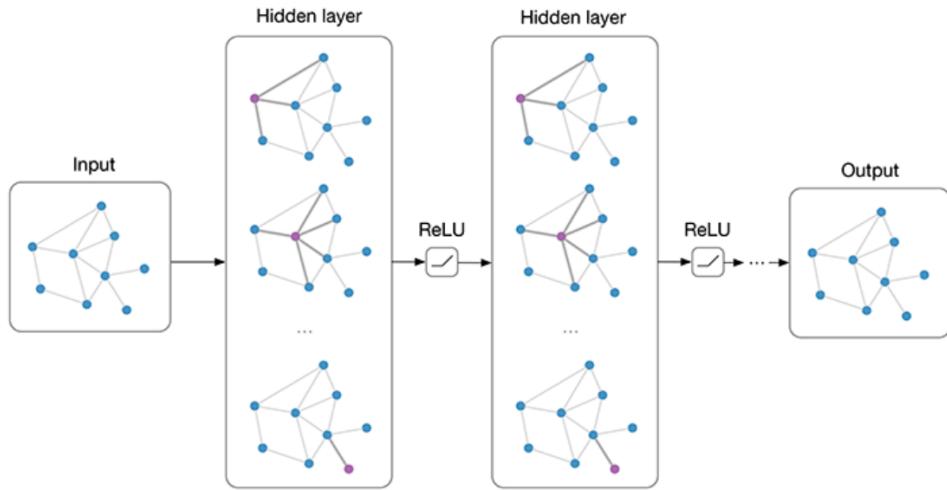


图 2.10 多层 GCN 网络示意图

ii) 接收：每个节点将邻居节点的特征信息聚集起来。这一步是在对节点的局部结构信息进行融合；iii) 变换：把前面的信息聚集之后做非线性变换，增加模型的表达能力。图2.10展示了一个简单的 GCN 网络，从图中可以发现数据的拓扑结构在训练过程中始终保持不变，可以在训练中重复利用。每一层的节点特征维度是不固定的，可以根据训练任务的复杂程度进行灵活的调整。

概括而言，图卷积算子的作用就是将每个节点的特征与其邻居节点的特征加权平均后传播到下一层，具有以下特点：1) 局部参数共享，计算节点特征值时，按一定规律将参与聚合的节点分为若干个子集，同一个子集内的节点采用相同的权重；2) 节点感受域正比于层数，每一层图卷积运算只能和相邻的节点进行特征聚合；通过增加 GCN 层数，每个节点上参与运算的其他节点信息就更多；3) 适用于任意拓扑结构的节点与图，能同时对节点特征信息与图结构信息进行端对端学习。

2.4 本章小结

本章首先通过对比传统 CFD 求解器和基于深度学习的气动优化模型的工作流程，明确了利用深度学习技术和方法要解决的 CFD 问题。然后介绍了 CFD 相关理论知识，重点介绍了在宏观和介观层面对流体运动规律的描述。最后分析了气动流场模拟任务和深度学习中图像回归预测任务的相似性，介绍了与本文研究内容紧密相关的深度网络模型。

第三章 基于深度卷积网络的流场预测方法

本章介绍了基于深度神经网络的气动流场预测方法，首先将流场数据通过合适的方法表示为深度神经网络可以接受的输入。然后借鉴图像分割领域中的 U-net 网络设计了气动流场预测网络 FlowDNN，根据气动流场预测的实际需要，通过嵌入注意力模块网络架构对 FlowDNN 进行了改进，设计了遵循守恒定律的物理损失函数。最后利用基于 LBM 的求解器生成实验数据集，定义了新的预测模型评测指标并与三种基线网络进行了全面的比较。

3.1 引言

3.1.1 动机分析

为了解决传统 CFD 方法进行流场模拟时间、经济成本较高的问题，本文尝试利用数据驱动的深度卷积网络实现端到端流场预测，提升气动流场模拟的效率。基于深度卷积网络的气动特性和气动性能预测具有以下优势：1) 深度卷积网络不需要对输入特征进行人工筛选和精简，可极大地减少设计变量数目限制，设计者能对几何外形的坐标点进行直接控制，设计变量可以细化为几何外形所有离散点的空间坐标，实现精细的外形优化。2) 深度卷积网络可以得到更加符合物理直觉的模型。如对几何外形坐标点和气动性能的关系进行建模，输入数据是一个二阶或三阶张量，很好地保留了坐标点间的空间相邻关系，其特征提取过程具有空间上的平移不变性和缩放不变性。3) 深度卷积网络由于能够提取更深层次的特征，因而拥有更强大的归纳能力，不仅可以获得气动性能预测，还可以将流场特征（如压力分布等）作为预测输出。

当前，基于深度学习的气动评估及其在气动优化中的应用尚处于起步阶段，多参考深度学习在其他领域应用较为成熟的方法技术，尤其是深度卷积神经网络在计算机视觉领域的研究成果，在深度学习优化技术应用于气动优化场景方面缺少进一步的研究，深度学习还没有与空气动力学实现交叉融合。因此，本文提出构建嵌入物理约束的深度卷积网络模型，利用深度学习优化技术增强模型流场数据特征提取能力，使其更加适用于处理气动优化问题。

3.1.2 研究思路

流场预测任务和图像回归预测任务相似，要解决的核心问题是利用深度神经网络学习输入到输出的映射关系。通过对图像回归预测相关方法和技术的研究，本文基于广泛用于图像分割的 U-net 网络实现对气动流场的快速准确预测。

在流场数据表示方面，通过合适的数据表示方法将几何体、边界条件等流场信息表示为深度卷积网络可以接受的矩阵形式。针对流场预测任务特点，对 U-net 网络架构进行改进。相对于图像分割任务，流场预测任务对模型的表示能力要求更高：在进行图像分割时，模型只需要对每个像素点进行准确的分类，即可有效区分图像的不同区域，是一种分类任务；在进行流场预测时，流场预测模型需要预测每个格点上的流场物理量，是一种回归任务。因此不能简单地将图像分割中的模型迁移应用到流场预测任务中，需要在网络架构上进行针对性的改进。此外，流场数据不同于图片数据，虽然两者都具有较强的时空相关性，但是流场数据是流体流动稳定后的结果，流场数据的形成遵循一定的物理规律，在进行深度卷积网络训练时需要合理嵌入物理约束保证预测结果的有效性。

3.1.3 基于笛卡尔网格的流场数据表示

利用深度学习技术对气动流场进行预测首先要解决的问题就是流场数据的表示问题，即如何将边界条件，物理场（如速度矢量场）等表示为神经网络可以接受的形式。在此我们介绍两种方法符号距离函数（Signed Distance Function, SDF）和二元法（binary representation）。

符号距离函数 对于二维笛卡尔图像，图像域 $\Omega \subset R^2$ 上的每个笛卡尔网格点为 (i, j) ，使用 $f(i, j)$ 表示符号距离函数符号。对于几何图形经过的点， $f(i, j)$ 值为 0。设边界点集合为 Z ，则有：

$$Z = \{(i, j) \in R^2 : f(i, j) = 0\} \quad (3.1)$$

当点在几何体内部时， $f(i, j) < 0$ ，当点在几何体外部时， $f(i, j) > 0$ ，符号距离函数 $D(i, j)$ 具体计算公式如下：

$$D(i, j) = \min_{(i', j') \in Z} |(i, j) - (i', j')| \operatorname{sign}(f(i, j)) \quad (3.2)$$

$D(i, j)$ 表示给定点 (i, j) 到几何边界的最短距离。图3.1是利用 SDF 表示二维几何图形的示意图。表示的几何图形使用白线表示，形状为圆形。

二元法 图3.2是二元法原理示意图，先将几何图形投射到笛卡尔网格上，使用 $B(i, j)$ 表示二元法，则当点在几何体内部或边界时， $B(i, j) = 1$ ，当点在几何体外部时， $B(i, j) = 0$ ，通过这样表示就能将几何图形表示为一张“人工图像”。相对于 SDF，二元法进行了进一步简化，每个点到边界的物理距离没有显示的表示出来，但是对于图像数据而言，我们认为每个点的位置信息都在坐标信息中有所体现。因此，从简化预处理的角度出发，实验中使用了二元法对流场边界条件和几

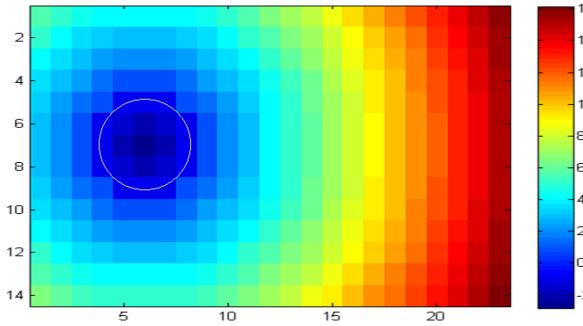


图 3.1 SDF 原理示意图

何图形进行表示。

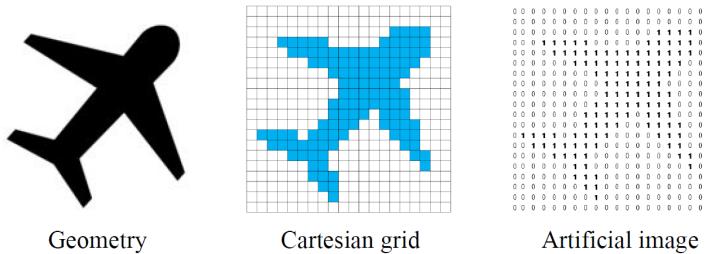


图 3.2 二元法原理示意图

无论是 SDF 和二元法，在输入流场时表示的边界条件和几何图形，在模型完成训练后进行流场预测时，每个笛卡尔网格点上的数值就不再表示几何图形，而是预测的物理量。通常输出特征图大小与输入相同，不同维度的特征图分别表示不同物理量在流场中的分布。

3.2 基于 U-net 网络的气动流场预测

3.2.1 稳态流场预测深度模型 (FlowDNN)

本章基于 U-net 网络优化设计了专用于稳态流场预测的深度神经网络架构 FlowDNN，图3.3展示了 FlowDNN 网络结构。

黑色实线箭头表示卷积层和黑色虚线箭头表示反卷积层，棕色箭头表示嵌入注意力模块的 skip connection。流场边界和几何外形经过预处理得到“人工图像”作为网络输入。输出是预测的二维速度场，预测值和真值的损失函数为物理损失函数，用于神经网络反向传播。

表3.1展示了 FlowDNN 网络结构的具体细节。对于输入和输出而言， $128 \times 128 \times 1$ 表示只有 1 个通道且大小为 128×128 的特征图。对于（逆）卷积层来说， 4×4 表示卷积核的大小，16 表示卷积核的数量，2 表示步长。作为 U-net 的变体，FlowDNN 也具有“U”形架构，包括 7 个下采样模块和 7 个上采样模块分别用于

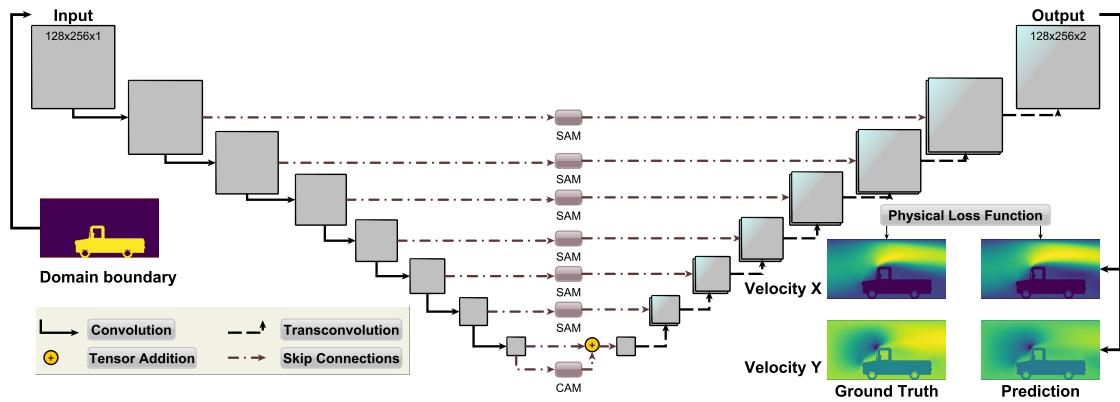


图 3.3 FlowDNN 网络结构示意图

表 3.1 FlowDNN 的网络结构细节

	编码部分	解码部分
<i>Input</i>	128 x 256 x 1	1 x 2 x 128
(DE)CONV1	4 x 4, 16, 2	2 x 2, 128, 2
(DE)CONV2	4 x 4, 32, 2	4 x 4, 128, 2
(DE)CONV3	4 x 4, 32, 2	4 x 4, 64, 2
(DE)CONV4	4 x 4, 64, 2	4 x 4, 64, 2
(DE)CONV5	4 x 4, 64, 2	4 x 4, 32, 2
(DE)CONV6	4 x 4, 128, 2	4 x 4, 32, 2
(DE)CONV7	2 x 2, 128, 2	4 x 4, 2, 2
<i>Output</i>	1 x 2 x 128	128 x 256 x 2

编码和解码。每个下采样模块包括一个卷积层，一个激活层和一个批标准化层；卷积核的填充（padding）为 1，步长为 2，用来实现下采样以提取高维抽象特征。由于使用了逆卷积操作，可能会使预测结果出现棋盘效应^[53]，影响预测的精度。所以为了消除棋盘效应的影响，本文卷积核大小设置为步长 2 的整数倍，而不是常用的 3x3 或者 5x5 等尺寸。前六个卷积核的感受视野为 4x4，最底部的卷积核大小为 2x2，因为此时输入特征图大小仅为 1x2（填充之后为 3x4）。上采样层结构类似，不同的是使用逆卷积操作进行特征图的扩展，且不再使用批标准化。每个下采样模块和上采样模块通过 skip connection 连接。为了更好的融合浅层的特征，本文在 skip connection 上嵌入了注意力模块（细节参考3.2.3节），增加模型的非线性和学习能力，加强不同维度信息的交互。

3.2.2 物理损失函数

将气动流场预测问题转化为图像回归预测问题后，在深度学习中常使用均方误差或均方根误差等损失函数来指导模型训练，使预测结果在每个像素点上尽可能接近真值。考虑到流体运动遵循一定的流动规律，仅仅依靠传统的 L_1 或 L_2 损失函数来约束神经网络训练，可能会导致模型给出的误差较低但是不符合物理规律的预测结果。因此，本文基于流体流动的物理规律，提出了融合物理损失项和传统 L_1 损失项的物理损失函数，以“软”约束的方式，指导模型训练，在保证预测结果准确率的前提下，得到满足物理一致性的结果。对于二维定常不可压流体流动问题，其宏观控制方程可以简化为：

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.3)$$

$$\frac{\partial(uu)}{\partial x} + \frac{\partial(uv)}{\partial y} = \frac{\partial\tau_{xx}}{\partial x} + \frac{\partial\tau_{yx}}{\partial y} - \frac{\partial p}{\partial x} \quad (3.4)$$

$$\frac{\partial(vu)}{\partial x} + \frac{\partial(vv)}{\partial y} = \frac{\partial\tau_{xy}}{\partial x} + \frac{\partial\tau_{yy}}{\partial y} - \frac{\partial p}{\partial y} \quad (3.5)$$

$$e_{in} + \frac{u^2 + v^2}{2} = e \quad (3.6)$$

其中方程3.3是根据质量守恒定律定义的连续性方程，方程3.4和3.5是根据牛顿第二定义的动量方程（x方向和y方向动量分量均守恒），方程3.6是在不考虑源项和扩散项的条件下得到的能量守恒方程。具体而言， u 和 v 分别代表 x 方向和 y 方向的速度分量， τ_{xx} ， τ_{yx} ， τ_{xy} 和 τ_{yy} 是粘性应力张量， p 表示压力， e_{in} 是单位质量物质的内能， e 是单位质量物质的总能量。本文不考虑能量守恒方程并忽略了压力场的作用，依据连续性方程和动量守恒方程并结合传统深度学习训练中的损失函数 L_1 设计了新的物理损失函数：

$$L_{\text{physical}} = \alpha_1 L_1 + \alpha_2 L_{\text{mass}} + \alpha_3 L_{\text{momentum}} \quad (3.7)$$

其中 α_1 ， α_2 和 α_3 是三个损失项的权重。对于二维几何有：

$$L_1 = \frac{1}{2mn_xn_y} \sum_{l=1}^m \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} (|u_{ij}^l - \bar{u}_{ij}^l| + |v_{ij}^l - \bar{v}_{ij}^l|) \quad (3.8)$$

$$L_{\text{mass}} = \frac{1}{m(n_x-2)(n_y-2)} \sum_{l=1}^m \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \left| \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)_{ij}^l - \left(\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} \right)_{ij}^l \right| \quad (3.9)$$

$$L_{\text{momentum}} = \frac{1}{m(n_x-2)(n_y-2)} \sum_{l=1}^m \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \\ \left\{ \left| \left[\left(\frac{\partial(uu)}{\partial x} + \frac{\partial(uv)}{\partial y} \right)_{ij}^l - \frac{1}{\text{Re}} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)_{ij}^l \right] - \left[\left(\frac{\partial(\bar{u}\bar{u})}{\partial x} + \frac{\partial(\bar{u}\bar{v})}{\partial y} \right)_{ij}^l - \frac{1}{\text{Re}} \left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} \right)_{ij}^l \right] \right| + \right. \\ \left. \left| \left[\left(\frac{\partial(vu)}{\partial x} + \frac{\partial(vv)}{\partial y} \right)_{ij}^l - \frac{1}{\text{Re}} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)_{ij}^l \right] - \left[\left(\frac{\partial(\bar{v}\bar{u})}{\partial x} + \frac{\partial(\bar{v}\bar{v})}{\partial y} \right)_{ij}^l - \frac{1}{\text{Re}} \left(\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} \right)_{ij}^l \right] \right| \right\} \quad (3.10)$$

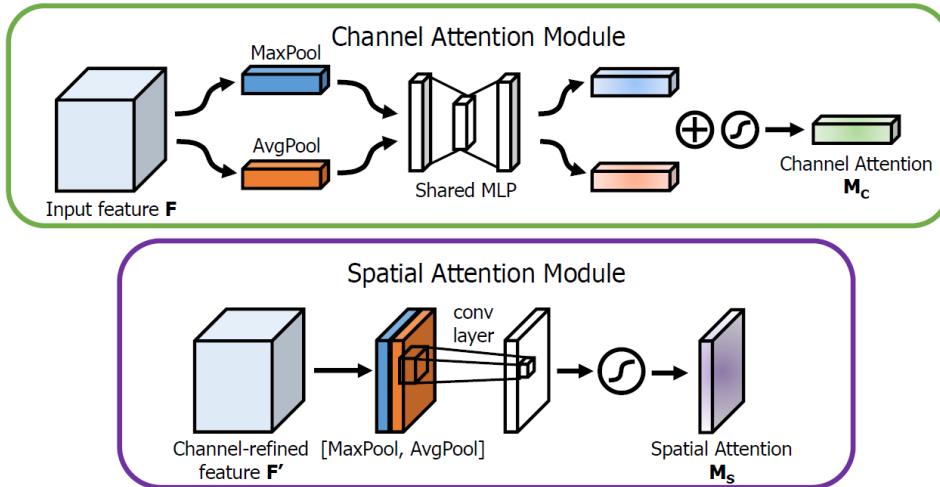
其中 m 是神经网络训练中训练样本批大小 (batch size), l 表示某个样本, n_x 和 n_y 是沿 x 方向和 y 方向的像素 (网格) 数量, u 和 v 分别是 x 方向和 y 方向的速度分量, \bar{u} 和 \bar{v} 代表相应方向上的预测速度分量。 L_{mass} 是基于质量守恒定律的损失函数, 评估在预测流场和参考流场 (由 CFD 求解器计算得到, 即深度学习训练中的真值) 中每个单元质量变化的差异。同理, L_{momentum} 是基于动量守恒定律的损失函数, 比较预测流场和参考流场中 x 和 y 方向上的动量变化的差异。 Re 代表固定的雷诺数 (Reynolds number), 流动条件通常通过描述惯性比的无量纲雷诺数来量化, 以区分是否为有粘流动问题。在这里, 一阶和二阶偏导数是使用一阶和二阶中心差分格式^[54], 以 x 方向的速度 u 为例有:

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial x} &= \frac{1}{2} (u_{i+1,j} - u_{i-1,j}); \\ \frac{\partial^2 u_{i,j}}{\partial x^2} &= u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \end{aligned} \quad (3.11)$$

3.2.3 注意力机制

在计算机视觉中图像分割或者目标检测等任务中, 研究者通常希望网络能够更多的关注一些特定的区域, 比如目标所在的区域, 注意力机制^[55]是常用的优化方法之一。类似地, 在气动流场预测任务中, 也有类似的需求。CFD 研究者通常更加关注几何体周围的流场, 这部分区域一般称为感兴趣区域 (Region of Interest, ROI)。原因是这部分区域流场变化快, 蕴含的信息多, 一些关键的气动系数 (如升阻比, 压力系数等) 也与几何体周围的流场更相关。经过以上分析, 我们在气动流场预测网络结构中引入注意力机制, 期望能够提升 ROI 区域的预测精度。

FlowDNN 嵌入了两个轻量级注意力模块: 通道注意力模块 (channel attention module, CAM) 和空间注意力模块 (spatial attention module, SAM)^[2]。CAM 和 SAM 分别可以从通道维度和空间维度自适应提取神经网络需要重点关注的信息, 图3.4是 CAM 和 SAM 的结构示意图。

图 3.4 CAM 和 SAM 结构示意图^[2]

CAM 通过全局池化操作和多层感知机对通道维度的信息进行了整合，SAM 通过全局池化操作和卷积操作对空间维度的信息进行了整合，具体工作原理如下：

$$\mathbf{F}_c = \mathbf{M}_c(F) \otimes F \quad (3.12)$$

$$\mathbf{F}_s = \mathbf{M}_s(F) \otimes F \quad (3.13)$$

$$\mathbf{M}_c(F) = \sigma(\text{MLP}(\text{GAP}(F)) + \text{MLP}(\text{GMP}(F))) \quad (3.14)$$

$$\mathbf{M}_s(F) = \sigma(\text{Conv}(\text{GAP}_c(F) \oplus \text{GMP}_c(F))) \quad (3.15)$$

其中 $F \in R^{C \times H \times W}$ 表示输入特征图（C、H、W 分别表示通道数和特征图的高度及宽度）， $\mathbf{M}_c \in R^{C \times 1 \times 1}$ 和 $\mathbf{M}_s \in R^{1 \times H \times W}$ 分别代表 CAM 和 SAM。 \otimes 表示元素乘法（element-wise multiplication），经 CAM 和 SAM 处理后的输出特征图 $\mathbf{M}_c(F)$ 和 $\mathbf{M}_s(F)$ 需要与输入特征图 F 相乘，保证输入和最终输出的大小和维度一致。公式3.14和3.15展示了 CAM 和 SAM 的具体操作：CAM 先对输入特征图在空间域进行全局最大池化（global max pooling, GMP）和全局平均池化（global average pooling, GAP），分别得到一个单通道向量，再将该向量输入一个多层次感知机（multi-layer perceptron, MLP）进行通道方向的信息融合，最后将 GMP 和 GAP 的结果相加，通过 sigmoid 激活函数得到 $\mathbf{M}_c(F)$ ；SAM 也对输入特征图进行了 GMP 和 GAP，不同的是池化操作是在通道方向进行；利用 \oplus 操作，将池化的结果进行叠加，然后利用卷积操作对空间维度的信息进行融合。

由于 SAM 模块中含有卷积操作，神经网络剪枝会对 SAM 模块产生影响，所以在设置注意力模块时仅在“U”形 FlowDNN 的底部设置了 SAM 模块，其他 skip connections 仅嵌入 CAM 模块；在进行神经网络剪枝时，与 SAM 模块相连的卷积层和逆卷积层不参与剪枝，保证 SAM 模块不受影响。CAM 模块通过多层次感

知机进行通道维度的信息融合，因此不受神经网络剪枝的影响。

3.2.4 神经网络剪枝

神经网络剪枝（network pruning）通过裁剪卷积层中卷积核大小或数量及其关联层参数大小，减小模型的计算量和体积，从而加快模型部署后的推理速度，是一种减小模型大小和降低模型计算复杂度的常用方式。在气动流场预测模型训练完成后，我们对网络进行了神经网络剪枝^[56]，主要有两方面考虑：a) 一方面，相对于基线模型 FlowDNN 的网络参数较多。通过对网络进行剪枝，可以证明 FlowDNN 对气动流场预测性能的提升不是简单的增加网络参数量；b) 另一方面，由于神经网络的冗余性，适当的剪枝并不会对预测精度产生明显的影响，但是剪枝后模型的推理时间将有效减少。对于复杂的大规模气动流场模拟，剪枝对于提升预测性能意义重大。

神经网络剪枝要解决的核心问题就是“剪哪里”，即评估卷积核重要性的标准，常见的标准有：内核权重的大小，特征映射激活值的均值、标准差和比例，激活值和预测值之间的互信息等。本文采用了文献^[57]的方法，通过基于泰勒展开的方法近似评估每个卷积核被剪枝后损失函数的变化，进而根据卷积核的重要性对网络进行裁剪。图3.5是神经网络剪枝的流程示意图：1) 载入训练好的模型;2) 通过剪枝算法对每个卷积核的重要性进行评估;3) 剔除重要性最低的卷积核;4) 利用训练数据对剪枝后的网络进行微调;5) 判断网络剪枝是否满足预设条件（参数量等），如果不满足，返回第二步；如果满足就结束网络剪枝。

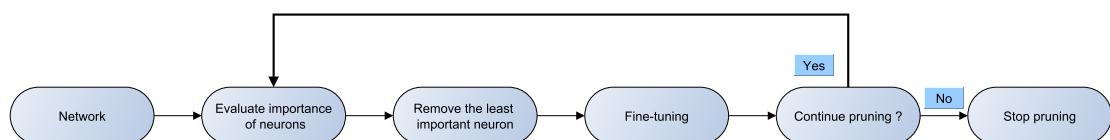


图 3.5 神经网络剪枝流程示意图

3.3 实验设置与结果分析

3.3.1 数据集与参数设置

3.3.1.1 基于 LBM 求解器生成数据集

实验算例设置为二维定常不可压层流的外流问题，几何外形为不同形状的二维汽车外形。我们使用基于 LBM 方法的开源库 MECHSYS¹对二维流场进行模拟。为了验证模型的泛化能力，我们将训练集设置为 3000 个简单几何体的组合形状，

¹ <http://mechsys.nongnu.org>

包括三角形、圆形、椭圆和方形等，不同样本中几何体大小和位置不同；验证集和测试集分别是 22 和 44 个不同形状的汽车外形。图3.6是部分训练集和测试集数据样本可视化结果。

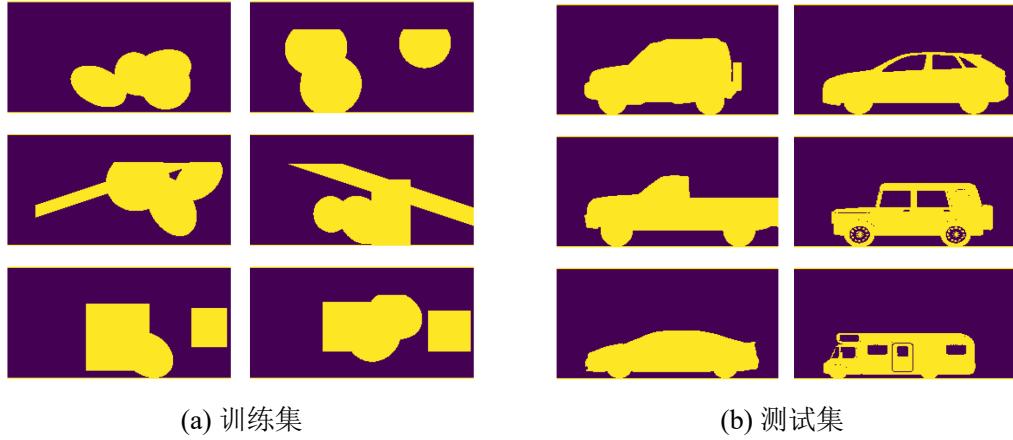


图 3.6 训练集和测试集数据样本可视化示例

在进行 LBM 流场模拟时，雷诺数设置为 400 以获取层流模拟结果；气流流向与 x 方向平行，采用 D2Q9 模型对速度矢量进行离散。所有 LBM 求解程序都在 GPU 上并行执行，迭代 10000 时间步，保证计算结果的收敛性。神经网络训练输入和输出特征图尺寸相同，输入为基于二元法表示的几何外形，输出为二维速度场，笛卡尔网格大小为 256x128。

3.3.1.2 参数设置

表 3.2 模型训练超参数设置

	Learning rate (Lr)	Lr decay interval	Batch size	Size of the filters	Pruning number	Weights of L_{physical}
Optimum	4×10^{-4}	25	16	4	1	(1, 5, 25) / 3
Tuning range	10^{-5} $\sim 10^{-3}$	20~50	4~64	2~8	1~5	-

FlowDNN 基于 PyTorch 深度学习框架编程实现，在实验中，我们利用英伟达 Tesla V100 GPU 实现模型的训练、参数调优和神经网络剪枝。深度学习训练有许多超参数，包括批大小学习率等，表3.2展示了一些重要的超参数调优的范围以及最终的设置。模型训练使用 Adam 优化算法，为了得到稳定收敛的结果，模型在数据集上训练 400 轮次 (epoch)。初始学习率设置为 4×10^{-4} ，每 25 个 epochs 将

学习率更新，乘以衰减因子 0.9。学习率衰减的好处是在算法优化的前期会加速学习，使得模型更容易接近局部或全局最优解，随着迭代次数增加，学习率会逐渐进行减小，保证模型在训练后期不会有太大的波动，从而更加接近最优解。批大小设置为 16，意味着模型在进行推理时可以同时对 16 个不同初始条件流场进行预测。在剪枝过程中，我们对每次剪枝的数量进行了探究，最终决定每次迭代只剪枝一个神经元，因为一次剪枝过多会破坏神经网络结构，导致预测准确率急剧下降。每个剪枝迭代步之后，对剪枝后的网络进行微调，训练 40 个 epochs 是神经网络收敛。对激活函数，本文考虑两种常用的激活函数 ReLU (rectified linear units) 和 ELU (exponential linear units)，其中 ELU 激活函数在文献^[58] 中被推荐使用，两者的比较结果可见3.3.2.4节。对于损失函数的权重，我们先固定 α_1 为 1，然后调整 α_2 和 α_3 使得损失函数的每一项在数值上对总的损失函数贡献相当；由于我们有损失函数有三项，将 α_1 、 α_2 和 α_3 分别除以 3，即为最终损失函数的权重。

3.3.1.3 性能评估标准

基线模型

本文引入了三个基线模型，用来比较和评估 FlowDNN 对稳态流场的预测结果。这三个基线模型分别参考文献^[1, 21, 59]，方便起见，三个模型分别被简称为 C-Net，T-Net 和 U-Net。

FIX:简单介绍三个基线模型

流场预测结果评价指标

对于深度神经网络模型的预测结果，本文使用平均相对误差（Mean Relative Error, MRE）比较预测流场和真值之间的差异，对气动流场的预测准确率进行总体的评估。除了全场模拟结果，CFD 领域专家通常对某些特定的区域感兴趣，比如流场边界层（即几何体表面附近流场），在这些区域中流场变化剧烈，往往包含着更多有用的信息，我们称之为感兴趣区域（Regions of Interest, ROI）。对于本文使用的汽车外部流场，我们定义 ROI 区域如图3.7中红色方框选中区域所示。ROI 不是固定的，对于不同的汽车外形，红色方框自适应地恰好选中整个汽车。在确定了 ROI 之后，即可求得该区域的平均相对误差 MRE_{ROI} 。根据流体流动遵循的质量守恒定律和动量守恒定律，我们设计了 MRE_{ma} 和 MRE_{mo} 分别评测模型预测结果与真值的质量守恒一致性和动量守恒一致性。

在测试集有 N 个样本的情况下，对于二维流场有：

$$MRE = \frac{1}{N} \sum_{l=1}^N \left(\sum_{i=1}^{n_x} \sum_{j=1}^{n_y} (|u_{ij}^l - \bar{u}_{ij}^l| + |v_{ij}^l - \bar{v}_{ij}^l|) / \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} (|u_{ij}^l| + |v_{ij}^l|) \right) \quad (3.16)$$

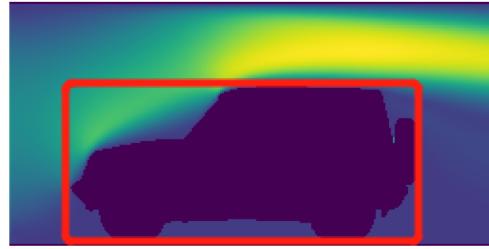


图 3.7 RoI 示意图

$$MRE_{RoI} = \frac{1}{N} \sum_{l=1}^N \left(\sum_{i=1}^{n_s} (|u_i^l - \bar{u}_i^l| + |v_i^l - \bar{v}_i^l|) / \sum_{i=1}^{n_s} (|u_i^l| + |v_i^l|) \right) \quad (3.17)$$

$$MRE_{ma}/MRE_{mo} = \frac{1}{N} \sum_{l=1}^N \left(\sum_{i=1}^{n_x} \sum_{j=1}^{n_y} |g_{ij}^l - \bar{g}_{ij}^l| / \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} |g_{ij}^l| \right) \quad (3.18)$$

其中 n_s 表示在 RoI 中的格子数。在公式3.18中， g 和 \bar{g} 分别表示在真值和预测结果的每个格子点上质量（或动量）的变化值。

3.3.2 实验结果与分析

3.3.2.1 总体预测性能比较

通过大量实验，我们将基于不同深度学习方法的预测结果和基于 LBM 方法的求解器模拟结果进行了对比。表3.3展示了 FlowDNN 和基线模型在测试集上预测准确率，推理时间和模型参数量等方面的性能表现，其中 AM 表示嵌入了注意力模块， P 代表神经网络剪枝；FlowDNN 均使用物理损失函数进行训练。

表 3.3 不同深度学习方法的各项性能指标比较

Method	MRE	MRE_{RoI}	MRE_{ma}	MRE_{mo}	Runtime (ms)	Parameters (Mb)
LBM	-	-	-	-	3300×16	-
C-Net	14.31%	30.99%	37.44%	45.60%	9.29	180.25
T-Net	24.65%	59.71%	79.09%	82.38%	8.47	7.45
U-Net	14.74%	13.14%	30.78%	44.42%	16.15	32.96
FlowDNN	7.91%	23.56%	18.28%	22.46%	3.52	13.70
FlowDNN w/ AM	5.34%	9.16%	12.34%	15.69%	4.51	13.74
FlowDNN w/ AM & P	4.77%	8.87%	12.14%	14.63%	3.62	7.40

在模型参数量方面，因为 C-Net 网络架构中使用了全连接层，每个神经元

与其前一层的所有神经元进行全连接，导致网络参数量远大于其他模型，大小为 180.25Mb。在基线模型中，T-Net 模型参数量最小为 7.45Mb，约是剪枝之前 FlowDNN 模型大小的一半。此外，在 FlowDNN 嵌入注意力模块之后，模型大小几乎没有改变，体现出轻量级注意力模块的灵活性。在推理时间方面，在利用深度学习模型进行流场预测时我们将批大小设置为 16，即一次推理过程可以得到 16 个算例的预测结果。对于每个算例，LBM 求解器通过迭代计算求解控制方程得到收敛的二维速度场结果，在 GPU 测试平台上程序的平均运行时间为 3300ms。对于深度学习预测模型，尽管 U-Net 模型参数不是最大，但推理时间最长，这可能和网络中卷积核的大小和采样的方式有关。FlowDNN 的推理时间最短为 3.52ms，在嵌入注意力模块之后推理时间增加了近四分之一，经过神经网络剪枝后推理时间基本和嵌入注意力模块之前相当。

对于指标 MRE 和 MRE_{RoI} ，模型的预测结果基本上是 MRE_{RoI} 远大于 MRE ，说明对于气动流场而言，几何体表面附近相较于远场更难以预测，反映出该区域速度变化更加剧烈，这符合流体流动的一般特点。然而我们观察到 U-Net 对于全场和 RoI 区域的预测性能接近甚至在 RoI 区域更准确，这一结果可能和 U-Net 的网络结构相关，但具体的原因需要进一步探究。相较于三种基线模型，在不引入注意力机制和神经网络剪枝的情况下，使用物理损失函数训练 FlowDNN 可以使 MRE 降低至 7.91%，接近三种基线模型中表现最好的 C-Net 的一半。但是 FlowDNN 对 RoI 区域的预测效果并不理想， MRE_{RoI} 仅为 23.56%，不仅远大于 U-Net 的 13.14%，而且是 MRE 预测误差的三倍。因此，本文引入了注意力机制希望能提升 FlowDNN 对几何体表面附近流场的预测效果。

对于指标 MRE_{ma} 和 MRE_{mo} ，不同深度学习方法的预测结果均体现出 MRE_{mo} 大于 MRE_{ma} 的趋势，可以得知相较于质量守恒，预测结果往往更难以保证动量守恒的一致性。由于 FlowDNN 均是使用物理损失函数训练，所以与基线模型相比， MRE_{ma} 和 MRE_{mo} 误差比较小。关于损失函数对预测结果的影响可见3.3.2.2节。此外，注意力模块也有效降低了 MRE_{ma} 和 MRE_{mo} 误差。

图3.8展示了 MRE 、 MRE_{RoI} 、 MRE_{ma} 和 MRE_{mo} 四种指标在测试集上分布情况。小提琴图外围的曲线宽度代表数据点分布的密度，内部的黑色长条表示中间二分之一的样本分布的范围，白点表示所有样本的中位数。观察三种基线模型的预测结果分布，除了 U-Net 在指标 MRE_{RoI} 上表现比较稳定之外，其他情况的预测误差都表现出很大的波动，即在某些样本上模型预测性能较好，在另一些样本上预测性能则较差，比如 U-Net 对某个算例的预测结果的 MRE 误差甚至超过了 50%。相比之下，FlowDNN 不仅在各项性能指标上平均误差较小，而且表现出良好的预测稳定性，对于测试集上不同的算例，FlowDNN 能将流场预测结果的

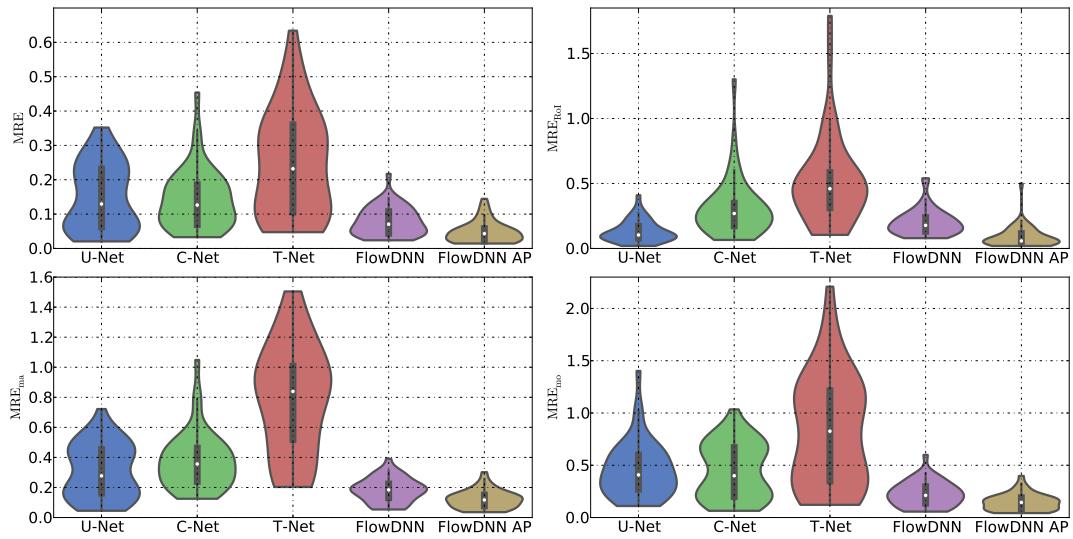


图 3.8 不同深度学习方法的预测性能指标在测试集上的分布

误差控制在较小的范围。

表 3.4 不同深度学习方法的二维速度场预测结果

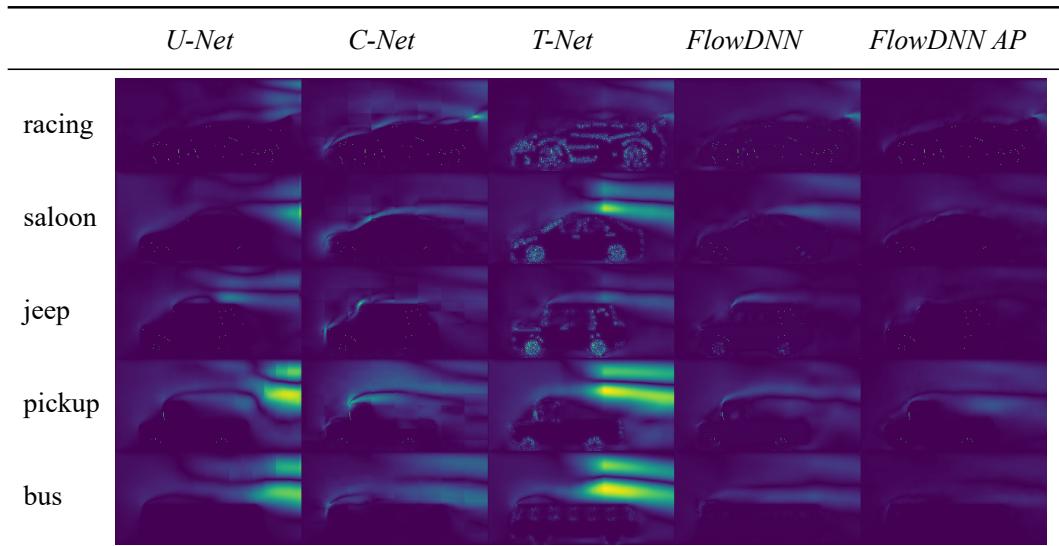
	LBM	<i>U-Net</i>	<i>C-Net</i>	<i>T-Net</i>	FlowDNN	FlowDNN AP
racing						
saloon						
jeep						
pickup						
bus						

表3.4展示了基于不同深度学习方法预测的流场速度场的可视化结果。FlowDNN AP 表示使用了注意力机制和神经网络剪枝的 FlowDNN。对于不同的汽车外形，FlowDNN 的二维速度场预测结果都体现出与 LBM 求解器模拟结果的高度一致性。观察 *U-Net* 和 *T-Net* 的预测结果，我们发现对于部分汽车外形如 *pickup*（皮卡）和 *bus*（公共汽车），几何体上方的流场会出现明显的断层，原因可能是模型的泛化能力有限，对于不同的汽车外形的预测结果可能出现过拟合现象。虽然 *C-Net* 对流场速度场的预测结果没有出现断层的现象，但是可以看出速度场可视化结果模糊，这是预测结果精度不够导致的。此外，*C-Net* 预测结果具有明显的分块现象，这和 *C-Net* 在进行卷积操作时使用了较大的步长有

关，导致数据的局部性信息有较大程度的损失。

为了对不同深度学习方法的流场预测结果进行进一步分析，本文比较了不同深度学习方法的二维速度场预测结果与真值的差异。如表3.5所示，深度学习模型的预测误差主要集中在几何体周围以及几何体尾部区域。这是因为流体绕物体运动时，物体表面附近形成很薄的边界层涡旋区。这些薄边界层或分离流涡旋区在物体后部表面分离后，将顺流而下，在物体后面形成紊乱的，充满大大小小旋涡的尾流。如果物体是钝体，尾流能保持很远距离，并对处于尾流中的其他物体产生影响。所以在几何体周围以及几何体尾部区域中，流体的运动规律更加复杂，对深度模型的学习能力提出了更高的要求。对比 FlowDNN 和三个基线模型的预测结果与真值的差异，可以直观地看出 FlowDNN 能够对二维速度场进行准确的预测，有效降低几何体周围以及几何体尾部区域的预测误差。

表 3.5 不同深度学习方法的二维速度场预测结果与真值的差异比较



3.3.2.2 物理损失函数

为了探究物理损失函数对流场预测结果的影响，尤其是对预测结果和 CFD 求解器模拟结果的物理一致性的影响，本文分别使用 $L_{physical}$ 和 L_1 损失函数对 FlowDNN 进行了训练。

图3.9展示了不同损失函数在 MRE 、 MRE_{RoI} 、 MRE_{ma} 和 MRE_{mo} 四项性能指标的表现。可以看到，相对于传统的 L_1 损失函数，使用物理损失函数训练预测模型可以有效地降低 MRE_{ma} （从 37.83% 降低至 18.28%），质量守恒误差减小了一半。物理损失函数对降低动量守恒误差 MRE_{mo} 也有一定效果，我们分析 $L_{physical}$ 对 MRE_{ma} 和 MRE_{mo} 影响不同的原因和损失函数权重有关。

我们对物理损失函数对预测结果的物理一致性的影响进行了进一步探究。

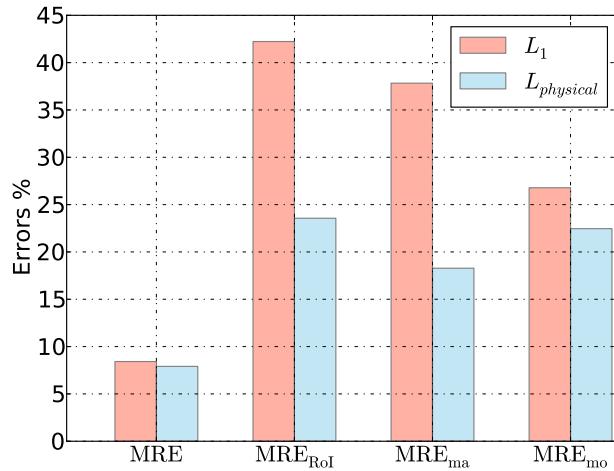


图 3.9 不同损失函数对流场预测结果的影响

图3.10展示了真值和不同损失函数预测结果在每个格子点上质量和动量的变化值。可以看出质量和动量的变化主要集中在定义的 RoI 区域和流场尾部区域，这也导致了使用物理损失函数训练的流场预测模型，其预测结果的 MRE 并没有显著降低，而 MRE_{RoI} 从 42.23% 降低至 23.56%。此外，我们观察到 L_1 的预测结果会出现不符合流体流动规律的情况，即 $\Delta mass$ 位于几何体内部，物理损失函数可以有效减少这种情况的发生，增强预测结果和 CFD 求解器模拟结果的物理一致性。

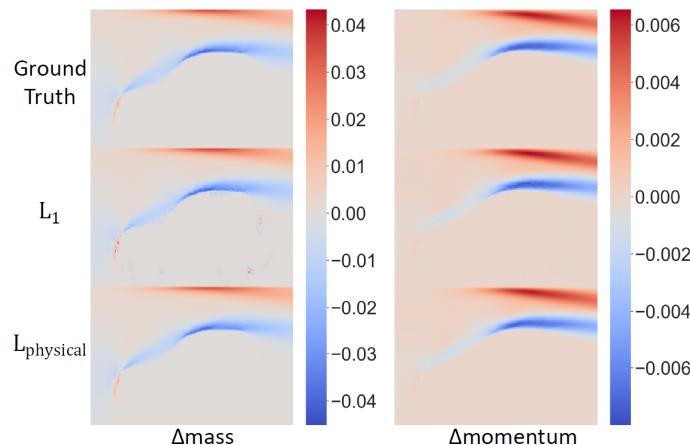


图 3.10 不同损失函数对流场预测结果物理一致性的影响

3.3.2.3 注意力机制

为了提升模型的表示能力和非线性，提高 FlowDNN 对 RoI 区域的预测精度，我们在 FlowDNN 的 skip connections 上嵌入了 SAM 和 CAM 模块。从表3.3中可知，嵌入注意力模块的 FlowDNN 有效增加了神经网络对 RoI 区域信息的提取能力，使 RoI 区域的平均相对误差 MRE_{RoI} 下降至 9.16%，同时提升了全场的预测准

确率。图3.11展示了在使用和不使用注意力模块的情况下，FlowDNN 的 MRE_{RoI} 误差在测试集上的分布。在未使用注意力机制之前，测试集上 MRE_{RoI} 误差集中在 18% 左右；使用注意力机制之后， MRE_{RoI} 误差集中在 6% 左右，而且误差分布更加集中，提升了 FlowDNN 的流场预测稳定性。

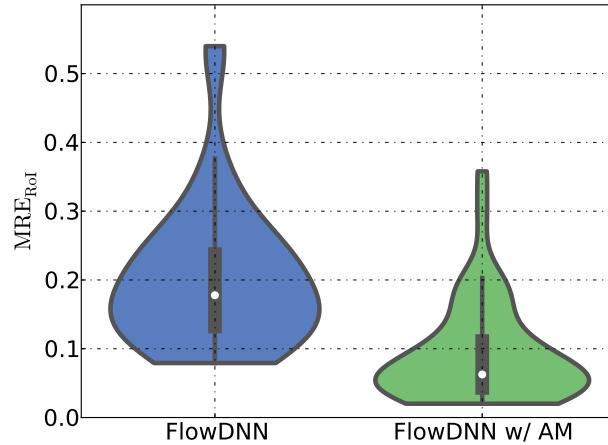


图 3.11 注意力机制对 MRE_{RoI} 在测试集上分布的影响

图3.12展示了注意力机制对预测模型训练的影响。相比不使用注意力机制的情况，嵌入了 CAM 和 SAM 模块的 FlowDNN 随着训练的进行收敛地更快，在 50 个 epoch 时验证上的损失函数已经收敛到较低的水平，在 150 个 epoch 时验证集上的误差已经趋于稳定。对比两条曲线的振幅，嵌入注意力模块的蓝线在垂直方向上的振幅更小，模型训练过程更加稳定。在两条误差曲线都趋于稳定时，使用注意力模块的 FlowDNN 的误差收敛至更低的水平。

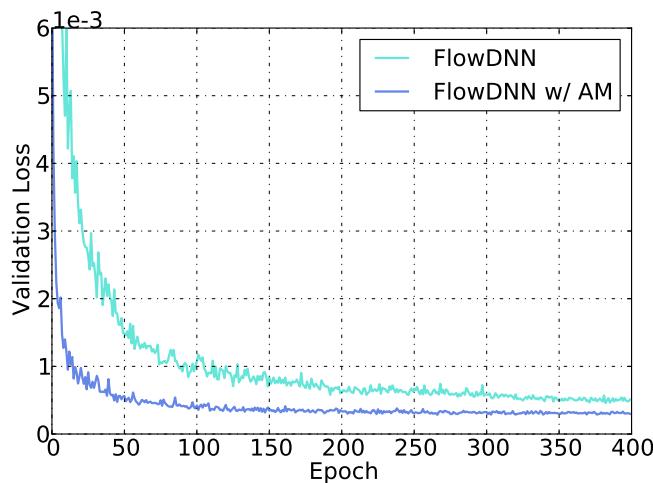


图 3.12 注意力机制对模型训练的影响

由此可见，在 FlowDNN 中合理地嵌入注意力模块，不仅能大幅提升模型在 RoI 区域的预测精度，还有助于加速模型训练收敛，缩短模型训练的时间，同时提高模型的预测稳定性。

3.3.2.4 激活函数

激活层是深度神经网络的重要组成部分，对神经网络模型的预测效果有着十分重要的影响。本文比较了常用于气动预测模型的 ReLU 和 ELU 两种激活函数对气动流场预测模型的影响。图3.13展示了两种激活函数的工作原理。

对 ReLU 激活函数有：

$$\varphi(x) = \max(0, x) \quad (3.19)$$

对 ELU 激活函数有：

$$\varphi(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases} \quad (3.20)$$

ReLU 和 ELU 的 $x > 0$ 的部分作用效果是相同的，保留神经元的原始输入 ($y = x$)，都能有效克服梯度消失的问题^[60]；在 $x < 0$ 的部分，ReLU 会将神经元的输出置为 0，ELU 则将输出映射到 (-1,0) 之间。

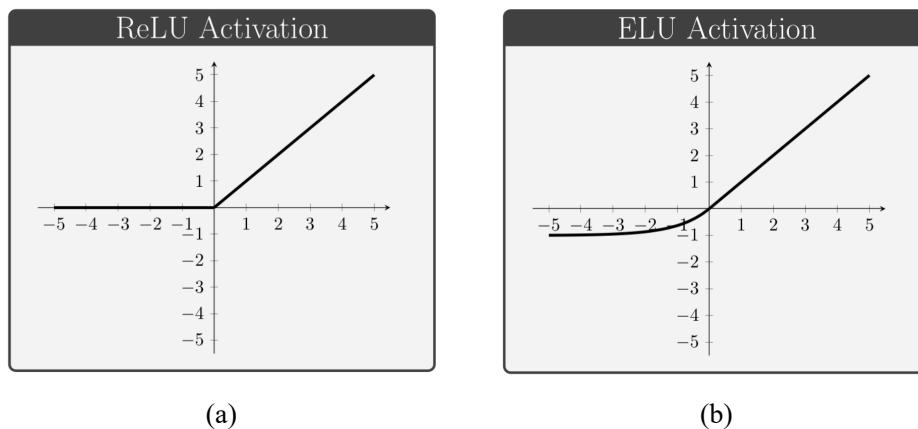


图 3.13 ReLU 和 ELU 激活函数

图3.14展示了 ReLU 和 ELU 两种激活函数对模型训练的影响。为了保证模型收敛和避免模型训练产生梯度爆炸，在激活函数进行处理之前我们对输入数据进行了批标准化（batch normalization, BN）。在使用 BN 的情况下，使用 ReLU 作为激活函数模型随着训练收敛地更快；训练稳定后，在测试集上的误差也小于使用 ELU 作为激活函数的情况。我们分析 ELU 激活函数表现较差的原因可能是 ELU 本身能够将激活单元的输出均值往 0 推近，和批标准化作用相似。所以 ELU 配合

BN 操作使用时，因为多次数据标准化处理，预测性能反而会较差。因此我们对仅使用 ELU 激活函数的情况进行了测试，结果表明，仅使用 ELU 和 ReLU+BN 的效果相当。考虑到 ReLU 激活函数能够将部分神经元输出置为 0，增加神经网络的稀疏性，有利于神经网络剪枝，所以 FlowDNN 采用了 ReLU 作为激活函数。

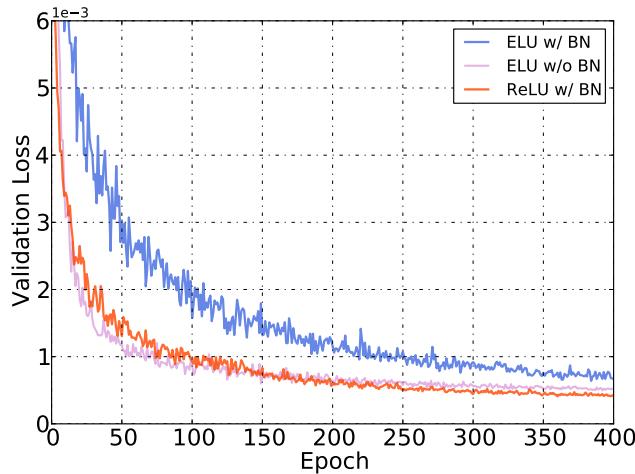


图 3.14 不同激活函数对模型训练的影响

3.3.2.5 神经网络剪枝

在选定损失函数，利用注意力机制对 FlowDNN 进行优化之后，我们对 FlowDNN 进行了神经网络剪枝。引入神经网络剪枝主要出于两方面的考虑：一方面剪枝后的预测模型参数减少，可以证明 FlowDNN 的预测性能提升不是简单地增加参数量；另一方面，通过神经网络剪枝精简网络结构，加快预测模型推理速度。

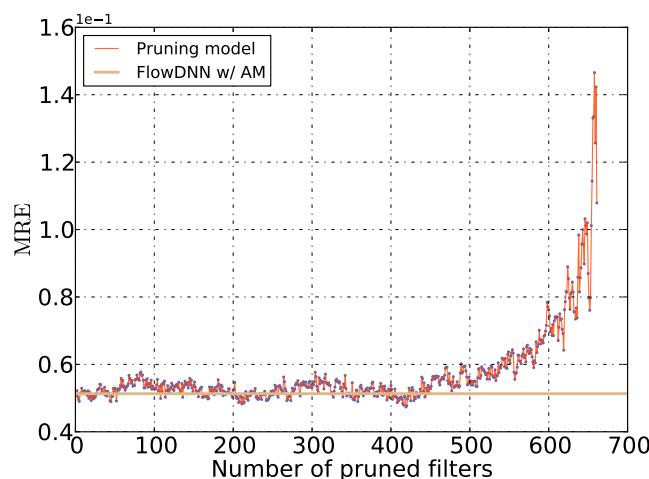


图 3.15 网络剪枝对 MRE 的影响

图3.15展示了神经网络剪枝对 FlowDNN 预测结果的平均相对误差的影响。其中平行于坐标横轴的是剪枝前 FlowDNN 的 MRE 误差，随着剪枝的进行，模型的 MRE 误差一直在基线上下波动；在剪枝神经元数量超过 500 之后，随着剪枝继续进行，破坏了神经网络结构，导致 MRE 误差急剧上升。本文选取 MRE 误差最小的剪枝模型作为最终的气动流场预测模型，整个二维速度场预测结果的平均相对误差仅为 4.77%。

3.4 本章小结

本章针对传统 CFD 方法进行流场模拟效率低的问题，提出一种基于深度卷积网络的端到端流场预测方法。首先分析了基于深度卷积网络进行流场预测的优势，对比了流场模拟任务和传统图像转换任务的相似点与不同点。其次研究了基于笛卡尔网络的流场数据表示方法，包括 SDF 方法和二元法。然后提出了基于 U-net 网络的气动流场预测模型 FlowDNN，具体包括：针对流体运动特点，通过改进 U-net 网络架构；基于流体流动的物理规律，提出融合物理损失项和传统 L_1 损失项的物理损失函数；针对边界层等 ROI 区域预测准确率低的问题，引入注意力模块；通过神经网络剪枝进一步优化模型预测效率。最后定义了新的流场预测结果评价指标，与传统 CFD 求解器和三个基线模型进行了全面的性能比较。

实验表明，相对于三个基线模型，FlowDNN 在测试集上的全场预测准确率、ROI 区域准确率更高，预测结果与 CFD 模拟结果的物理一致性更高；与基于 LBM 的传统 CFD 方法相比，流场模拟效率提升 14000x，同时将预测误差控制在 5% 以内。

第四章 基于图神经网络的流场模拟加速方法

本章介绍了基于图神经网络的气动流场模拟加速方法，首先利用 CFD 网格生成软件在算例几何模型上生成非结构网格，基于网格单元拓扑连接信息、边界条件和初始条件提取图神经网络训练的输入数据；利用 CFD 求解器获取图神经网络训练的真值。然后基于图卷积神经网络设计了气动流场模拟加速模型，在模型训练完成后，使用 CFD 求解器对气动流场模拟加速模型的输出结果进行优化。最后在测试集上比较了气动流场模拟加速方法与传统 CFD 数值模拟方法的气动流场模拟效率。

4.1 引言

4.1.1 动机分析

利用深度学习方法对气动流场进行预测的主要思想是将气动流场模拟问题转化为图像回归预测问题，因此需要将流场数据转化为矩阵的形式。在处理流场中非结构数据时，则需要通过采样的方式将数据转化为结构规则的笛卡尔网格形式，这样就会导致预测的结果在平滑区域（远场）过采样，而在流场边界层（几何体附近区域）欠采样，导致预测流场结果在几何体周围轮廓模糊，不能真实的反映物理量在边界层的分布。在传统基于网格的 CFD 模拟方法中，一般通过加密边界层的网格密度以更加细致地捕捉边界层的速度和压力等物理量的变化，使用均匀的粗粒度的网格进行模拟根本不能得到准确的结果甚至导致计算不收敛。此外，流场预测结果由深度学习模型输出，在复杂流动条件下，无法保证深度学习模型能够对流体运动规律进行准确学习，输出符合流体运动规律的预测结果。

基于以上考虑，本文在利用图神经网络进行流场模拟方面进行了探索，对气动流场模拟的效率和预测结果的有效性进行了权衡，提出了基于图神经网络的流场模拟加速方法。

4.1.2 研究思路

与基于传统卷积神经网络的流场预测方法相比，基于图神经网络的流场模拟加速方法主要有两点不同：1) 流场数据表示方法通用性更强。无论是结构网格还是非结构网格，利用图能够灵活地对网格单元的拓扑结构进行表示，边界条件和初始条件可以处理为节点特征向量。由于图结构利用网格对流场域进行了全尺寸模拟，不需要利用采样方法将流场表示为笛卡尔网格，所以最大程度保留了原始流场数据中的信息，一些控制流体流动的全局变量也可以在图神经网络训练时用

于每一个图节点上。2) 从根本上保证流场模拟结果的有效性。利用 CFD 求解器对深度学习模型的预测结果进行优化，将计算收敛的结果作为最终的流场预测结果。虽然深度学习模型在此方法中只是作为加速模块部分代替了 CFD 求解器的工作，导致气动流场模拟效率低于基于深度学习的流场预测方法，但由于气动流场模拟结果最终由 CFD 求解器输出，因此输出结果满足求解器计算收敛条件和流体流动物理规律。

由于图神经网络进行训练时需要保证图的拓扑结构保持不变，因此本章主要针对给定几何外形在不同流动条件下气动流场模拟效率进行了研究。

4.1.3 OpenFOAM 网格数据结构

本文基于 OpenFOAM 软件进行网格生成和流场模拟。OpenFOAM 是基于 C++ 程序语言开发用于场运算和操作的类库，主要包括两大类：求解器（solvers）和工具（utilities）。用户可以根据特定的连续介质流体力学问题在标准的求解器基础上进行开发。工具主要有前处理工具和后处理工具两类。前处理工具包括几何处理、网格处理、边界条件设置等，后处理工具包括数据提取与绘图、可视化处理等。OpenFOAM 主要网格文件及其包含的信息如表4.1所示。

表 4.1 OpenFOAM 网格文件信息

文件	存储信息
points	所有节点的三维坐标
faces	构造面单元的节点编号
owner	与面单元相对应 owner 体单元编号
neighbour	与面单元相对应 neighbour 体单元编号
boundary	边界条件设置

points 文件以矢量场的形式存储所有节点坐标，单位为米，文件主要内容如图4.1所示，节点的编号即为其坐标数据在 points 列表中的位置，初始编号为 0。

faces 文件存储面单元 - 节点拓扑结构数据，文件主要内容如图4.2所示。面单元最少由 3 个节点组成（三角形），其构造节点总数为每一行数据第一个整数（任意不小于 3 的值），每一行括号内的整数为面单元构造节点的编号，表示其在 points 列表中的位置。面单元的编号即为数据在 faces 列表中的位置，初始编号为 0。整个 faces 列表包含边界面单元。

如图4.3所示，owner 体单元与 neighbour 体单元的标识基于公共面单元 face 外法向量 (S_f) 的方向，即由 owner 单元指向 neighbour 单元。owner 单元与 neighbour

```

19  5146           // 节点总数 Number of points
20  (
21  // 数据格式 Data format
22  // (x坐标 y坐标 z坐标)
23  // (#x #y #z)
24  (1 0 0)          // 节点#0 Point #0
25  (5 5 0)          // 节点#1 Point #1
26  (5 -5 0)         // 节点#2 Point #2
27  (-5 -5 0)
28  (-5 5 0)

```

图 4.1 points 文件结构

```

19  17396 // 面单元总数 Number of faces
20  (
21  // 数据格式 Data format
22  // 面单元构造节点总数(构造节点编号)
23  // #number of points for face(#p1 #p2 #p3 ...)
24  4(423 2791 5092 2724) // 面单元#0 Face #0
25  4(423 1205 3573 2791) // 面单元#1 Face #1
26  4(1205 2724 5092 3573) // 面单元#2 Face #2
27  4(2681 5049 3137 769) // 面单元#3 Face #3
28  4(1181 769 3137 3549) // 面单元#4 Face #4
29  4(2681 1181 3549 5049)
30  4(423 2791 3495 1127)
31  4(2724 1127 3495 5092)

```

图 4.2 faces 文件结构

单元有时亦被称为左单元与右单元。

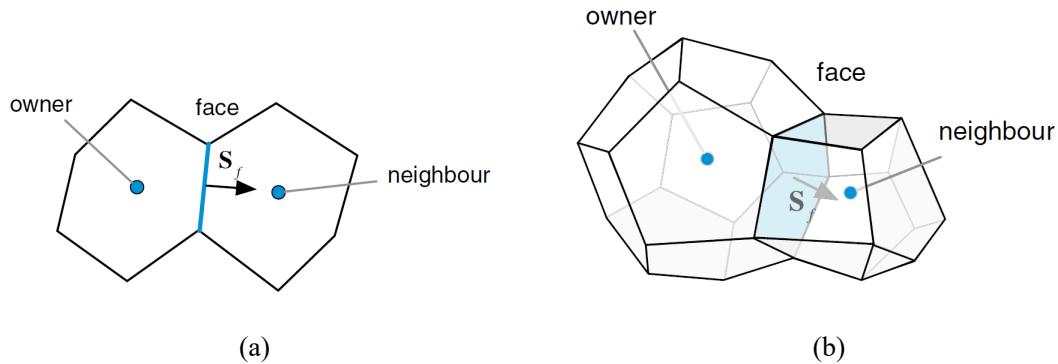


图 4.3 面的 owner 与 neighbor 的关系

owner 文件存储 owner 单元 - 面单元拓扑结构数据，文件主要内容如图4.4所示。owner 列表存储相对应面单元的 owner 单元编号，初始编号为 0，owner 列表的位置对应面单元的编号，亦对应于其在 faces 列表中的位置。任何一个面单元总存在一个 owner 单元，因此 owner 单元总数与面单元总数 nFaces 相等。

neighbour 文件存储 neighbour 单元 - 面单元拓扑结构数据，文件主要内容

```

20 17396 // owner 单元总数(等于面单元总数) Number of owner cells
21 (
22 0 // 面#0的owner单元编号 #owner of face #0
23 0 // 面#1的owner单元编号 #owner of face #1
24 0 // 面#2的owner单元编号 #owner of face #2
25 1 // 面#3的owner单元编号 #owner of face #3
26 1 // 面#4的owner单元编号 #owner of face #4
27 1
28 2

```

图 4.4 owner 文件结构

如图4.5所示。neighbour 列表存储相对应面单元的 neighbour 单元编号，初始编号不为 0，neighbour 列表的位置对应面单元的编号，亦对应于其在 faces 列表中的位置。需要特别说明的是，因为边界面单元没有 neighbour 单元，neighbour 单元总数与内部面单元总数 nInternalFaces 相等。

```

20 7309 // neighbour 单元总数(等于内部面单元总数) Number of neighbour cells
21 (
22 2 // 面#0的neighbour单元编号 #neighbour of face #0
23 995 // 面#1的neighbour单元编号 #neighbour of face #1
24 3139 // 面#2的neighbour单元编号 #neighbour of face #2
25 22 // 面#3的neighbour单元编号 #neighbour of face #3
26 1453 // 面#4的neighbour单元编号 #neighbour of face #4
27 3153
28 1739

```

图 4.5 neighbour 文件结构

boundary 文件存储整个网格的边界信息，比如边界区域名称、边界类型 type、面单元个数 nFaces 以及起始面单元编号 startFace 等，文件主要内容如图4.6所示。

```

18 7 // 边界总数 Number of boundaries
19 (
20   front // 边界名称 #boundary patch name
21   {
22     type empty; // 边界类型 #patchtype
23     physicalType patch;
24     nFaces 4941; //面单元总数 #number of face in patch set
25     startFace 7309; //起始面单元编号 #starting face index for patch
26   }
27   back
28   {
29     type empty;
30     physicalType patch;
31     nFaces 4941;
32     startFace 12250;
33   }
34   exit
35   {
36     type patch;
37     physicalType patch;
38     nFaces 17;
39     startFace 17191;
40   }

```

图 4.6 boundary 文件结构

OpenFOAM 不支持二维网格，对于导入的外部二维网格，OpenFOAM 会自动给网格加一个厚度，然后将前后面（front 和 back）边界类型设置为空（empty）。

4.2 基于 GCN 网络的气动流场模拟加速

针对流场数据的非结构性和深度学习模型预测结果的可靠性，本章提出了基于图卷积神经网络的气动流场模拟加速方法。图4.7展示了基于 GCN 网络的气动流场加速方法的工作流程，主要包括三个部分：a) 图数据表示部分；b) 基于 GCN 网络的气动流场加速部分；c) 基于 CFD 求解器的模拟结果优化部分。

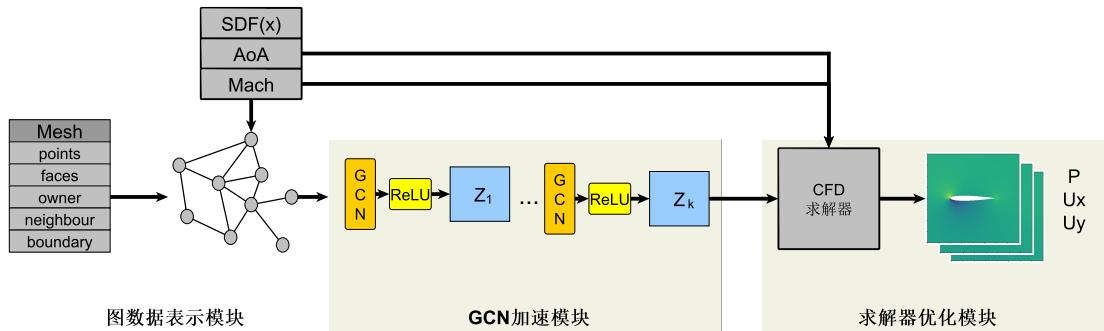


图 4.7 基于 GCN 网络的气动流场加速方法示意图

4.2.1 图结构数据表示

本章定义图结构 $G = (X, E)$ 来表示 OpenFOAM 网格信息，其中 $X \in R^{N \times d}$ 表示图节点，节点数目为 N ，节点特征维度为 d 。 E 表示边，每条边连接两个图节点，考虑到流体在网格中是相互流通的，所有的边均定义为无向边。

不同于基于网格节点进行流场模拟的求解器 SU2^[61]，OpenFOAM 是基于有限体积方法离散的求解器，基于控制体单元进行流场计算和模拟。因此，每个节点对应网格中的体单元，节点数和体单元总数相同；对于二维流场，节点特征值包括体心坐标 (C_x, C_y) ，体单元的边界属性 $\mathcal{O} B_l$ 。其中体心坐标可以通过 `points` 文件中点坐标进行插值得到；体单元的边界属性由 `boundary` 文件中所属面单元的边界类型决定。此外，本章将更多流场信息嵌入图节点特征向量中，包括控制流体流动的马赫数（Mach）、攻角（AoA）等全局变量。利用距离符号函数计算的给定图节点到几何边界的最短距离 (D_s) ，更好地帮助图神经网络学习节点的位置距离关系。每条边对应两个体单元的共有面，边总数和内部面单元总数相同，两个体单元分别为该面单元的 `owner` 和 `neighbor`。

通过以上表示可以得到基于体单元的图数据结构，图节点 $X = \{x_1, x_2 \dots, x_n\}$ ，每个图节点 X_i 的特征维度为 6 包括 $C_x, C_y, B_l, \text{Mach}, \text{AoA}$ 和 D_s ；边 $E = \{e_1, e_2 \dots, e_m\}$ ，每条边 e_i 连接两个图节点 $x_j, x_k \in X$ 且 $j \neq k$ 。这种基于网格结构的数据表示可以灵活处理各种复杂条件下的流场环境，而且能够扩展到三维的

流场数据表示，具有很强的通用性。全局的流场控制信息可以通过特征向量嵌入的方式作用在每个图节点上，与流场几何位置信息有效融合，充分参与图神经网络的训练。

4.2.2 基于 GCN 网络的气动流场模拟加速

利用图数据结构本章将流场数据表示为图神经网络可以接受的输入，该图数据包括了网格结构信息、流场边界信息和流场控制条件信息等。本章利用图卷积网络对形式化表示后的流场数据进行学习，输出流场稳定时速度场、压力场等相关物理量的预测结果。算法4.1展示了基于图卷积网络的气动流场模拟加速建模整体过程。

算法 4.1 基于图卷积网络的气动流场模拟加速建模

输入： 图数据结构： $Z_0 = X = [C_x, C_y, B_l, AoA, Mach, D_s]$

输出： 二维气动流场模拟结果： $Y = [U_x, U_y, P]$

- 1: 构建图卷积网络模型，随机初始化权重矩阵
 - 2: 计算图形的拉普拉斯矩阵 $L = D - W$
 - 3: 定义图卷积操作 GCN，确定激活函数
 - 4: **for** $i = 1$ to $K - 1$ **do**
 - 5: $Z_i = \text{ReLU}(\text{GCN}_i(Z_{i-1}))$
 - 6: **end for**
 - 7: $Y_{temp} = \text{GCN}_K(Z_{K-1})$, $Y_{temp} = [U_x, U_y, P, nut, nuTilda]$
 - 8: $Y_{temp} \Rightarrow S_0$
 - 9: $Y = S_t = simpleFoam(S_0)$
-

基于文献^[38]中的 GCN 模块，本章构建了用于气动流场模拟加速的图卷积网络，包括 K 个图卷积层，前 $K - 1$ 个图卷积层后接一个激活层。层与层之间的传播方式如下：

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (4.1)$$

其中 $H^{(l)} \in R^{N \times d^{(l)}}$ 是第 l 层图结构特征表达，维度是 $N \times d^{(l)}$ ， N 表示图节点数目， $d^{(l)}$ 是节点特征维度。矩阵 $\tilde{A} = A + I$ ，其中 A 是图节点的邻接矩阵， I 是单位矩阵，表示每个节点在下一层状态不仅与自己的邻居有关还与节点本身有关。矩阵 \tilde{D} 是 \tilde{A} 的度矩阵，用于归一化处理； $W^{(l)}$ 是权重矩阵，在图卷积网络训练过程中不断调整； σ 代表非线性激活函数 ReLU。图的拓扑结构在训练中保持不变，因此图卷积算子中 $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ 可以在训练之前确定并重复用于图卷积计算。

图卷积网络模型输出为 Y_{temp} ，包括 $[U_x, U_y, P, nut, nuTilda]$ ，其中粘度系数

nut 和湍流运动粘度 nuTilda 是 OpenFOAM 模拟计算的中间结果，有助于构造 CFD 求解器的初始场。结合流场控制条件攻角和马赫数，可以将气动流场预测结果转化为 OpenFOAM 中 CFD 求解器的初始场。通过训练，GCN 网络能够有效提取流场数据特征，学习流体流动规律；在新的给定流场条件下，GCN 网络能够输出一个接近真实模拟结果的预测流场。因此，相对于传统 CFD 模拟，基于 GCN 的流场模拟方法在该预测结果的基础上在利用 CFD 求解器进行预测结果优化，能够减少迭代计算的时间，从而实现模拟效率的提升。

4.3 实验设置与结果分析

4.3.1 数据集与参数设置

4.3.1.1 基于 OpenFOAM 求解器生成数据集

实验算例设置二维不可压湍流条件下固体外部流场，几何外形选取为空气动力学中经常研究的翼型。机翼一般都有对称面。平行于机翼的对称面截得的机翼截面，称为翼剖面，通常也称为翼型。翼型的几何形状是机翼的基本几何特性之一。翼型的气动特性，直接影响到机翼及整个飞行器的气动特性，在空气动力学理论和飞行器设计中具有重要的地位。图4.8展示了翼型 e342 和翼型 NACA 0012 的几何外形示意图^[62]。

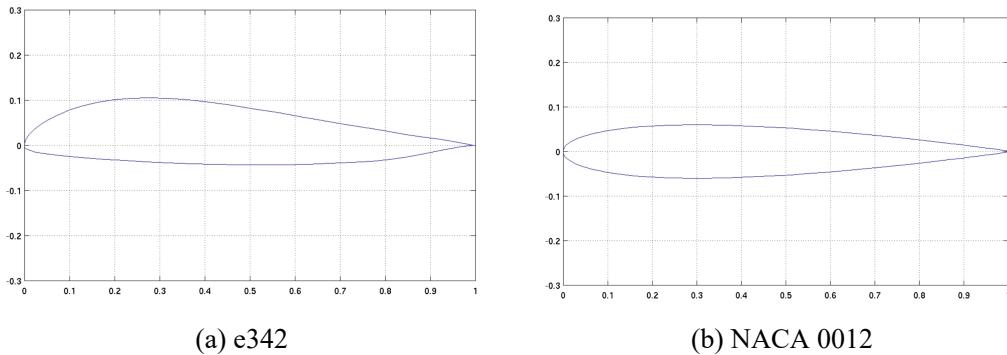


图 4.8 翼型 e342 和翼型 NACA 0012 几何外形示意图

在获取翼型的几何模型之后，利用 Gmsh^[63] 进行非结构网格的生成。Gmsh 是一个开源的具有内置 CAD (Computer Aided Design) 引擎和后期处理工具的三维有限元网格生成器，可以方便地在 Python 脚本中调用 Gmsh 提供的接口对翼型进行网格生成。全部网格信息保存在.msh 文件中，再利用 gmshToFoam 指令将.msh 文件导入 OpenFOAM 中，就能得到4.1.3节介绍的网格文件，完成网格生成任务。

在几何外形确定为翼型 e342 的基础上，本文在不同的攻角和来流马赫数条件

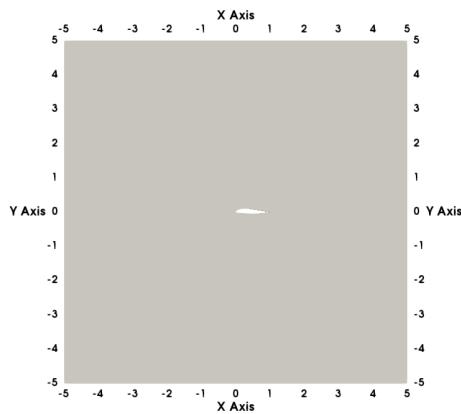


图 4.9 实验算例二维流场区域示意图

下，利用 OpenFOAM 软件对翼型周围流场进行模拟。图4.9展示了实验算例的流场区域设置，二维流场在 x 方向和 y 方向的范围均为 -5~5，翼型几何体长度范围 0~1。求解器使用 OpenFOAM 中不可压缩稳态流动求解器 `simpleFoam`，压力速度耦合采用的 `simple` 算法，湍流模型使用 SA 一方程模型。计算的时间步长设置为 1，迭代次数设置为 5000 保证在不同的流场条件下得到收敛的流场。根据攻角 (AoA) 和流场来流马赫数 (Mach number) 划分训练集和测试集，内插数据集和外推数据集划分如下：

内插数据集:

$$\text{AoA}_{train} = \{-10, -9, \dots, 9, 10\}$$

$$\text{Mach}_{train} = \{0.2, 0.3, 0.35, 0.4, 0.5, 0.55, 0.6, 0.7\}$$

$$\text{AoA}_{test} = \{-10, -9, \dots, 9, 10\}$$

$$\text{Mach}_{test} = \{0.25, 0.45, 0.65\}$$

外推数据集:

$$\text{AoA}_{train} = \{-10, -9, \dots, 9, 10\}$$

$$\text{Mach}_{train} = \{0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55\}$$

$$\text{AoA}_{test} = \{-10, -9, \dots, 9, 10\}$$

$$\text{Mach}_{test} = \{0.65, 0.6, 0.7\}$$

通过 AoA_{train} 和 Mach_{train} 与 AoA_{test} 和 Mach_{test} 的组合分别构成训练集和测试集的流场条件，其中训练集样本大小为 168，测试集数据大小为 63。在内插数据集中，训练集和测试集的数据不同，但是两个集合的参数有着相似的范围，广泛用于流场模拟相关工作中[21, 22]。为了进一步测试基于 GCN 网络的流场模拟加速方法的泛化能力，本章设置了外推数据集，其测试集与训练集的马赫数分别

处于不同的范围。

4.3.1.2 参数设置

本章在 PyTorch 深度学习框架下，基于 PyTorch Geometric 库提供的 GCN 模块设计了气动流场加速网络模型。实验在 Ubuntu18.04 系统下利用 GEFORCE RTX 2080Ti 显卡进行。模型训练使用 Adam 优化算法，学习率设置为 5×10^{-5} ，模型在数据集上训练 5000 轮次（epoch）。GCN 网络训练损失函数使用常用的最小绝对值误差 L_1 损失函数。

在生成数据集和带回 CFD 求解器进行预测结果优化均使用 OpenFOAM v7，对于速度、压力和湍流运动粘度的残差收敛条件均设置为 1×10^{-5} 。在进行代数方程求解时，本章使用 GAMG (generalised geometric-algebraic multi-grid) 多重网格求解器对压力进行求解，该求解器广泛用于加速 CFD 收敛；速度和湍流运动粘度使用 smoothSolver 光滑求解器进行求解。所有求解器容错误差均为 1×10^{-8} ，光滑器使用 GaussSeidel。

4.3.2 实验结果与分析

本节将基于 GCN 模型的模拟结果称为 *GCN* 模拟结果；将基于 GCN 模型的模拟结果并进一步利用 CFD 求解器进行优化的结果称为 *GCN + CFD* 模拟结果；将传统的 CFD 求解器计算结果称为 *P_CFD* 模拟结果。

FIX:更新数据

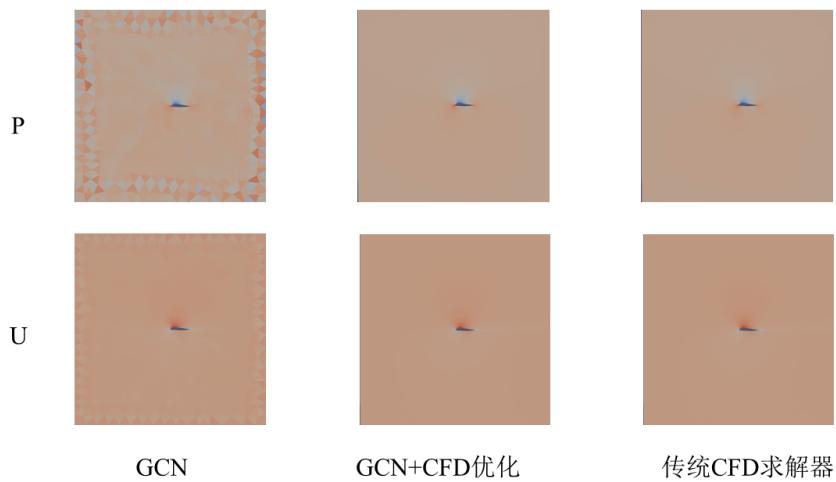


图 4.10 内插数据集上速度场和压力场的模拟结果比较

4.4 本章小结

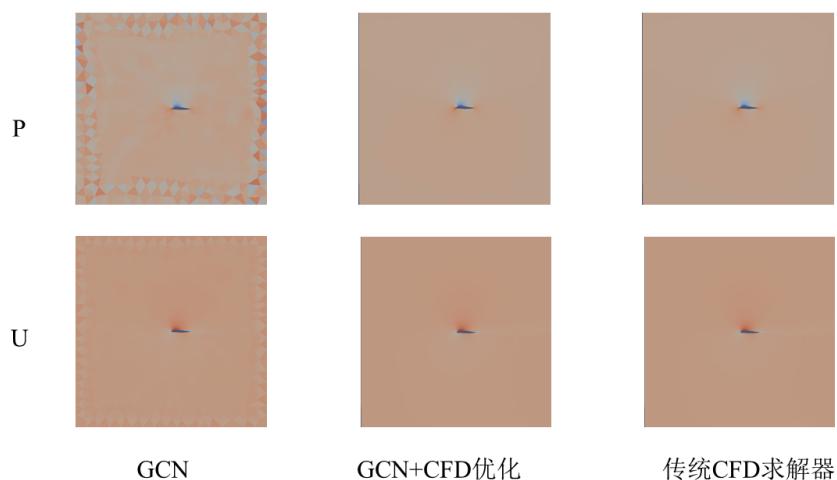


图 4.11 外推数据集上速度场和压力场的模拟结果比较

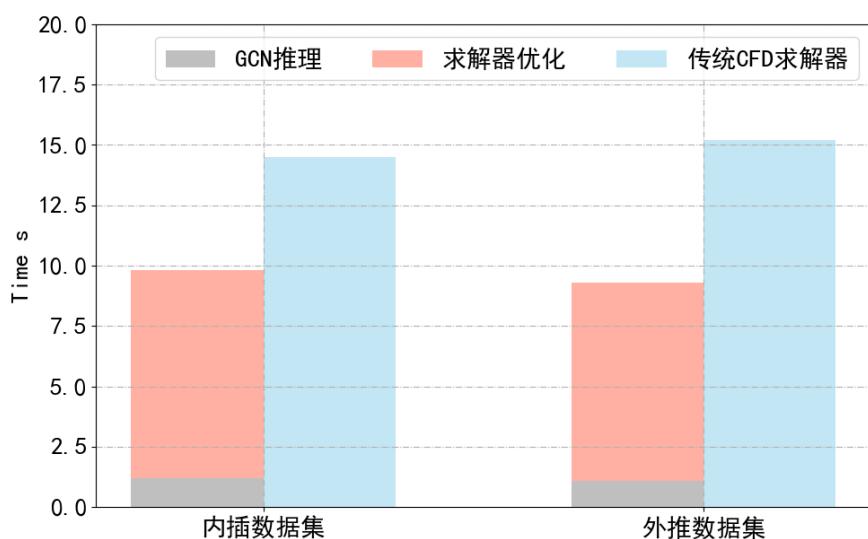


图 4.12 GCN+CFD 方法在不同测试集上的模拟效率

第五章 总结与展望

致 谢

衷心感谢导师 xxx 教授和 xxx 副教授对本人的精心指导。他们的言传身教将使我终生受益。

感谢 NUDTPAPER，它的存在让我的论文写作轻松自在了许多，让我的论文格式规整漂亮了许多。

参考文献

- [1] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation [C] // Navab N, Hornegger J, Ill W M W, et al. In Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III. 2015.
- [2] Woo S, Park J, Lee J, et al. CBAM: Convolutional Block Attention Module [C]. In ECCV. 2018.
- [3] Bourouiba L. Turbulent Gas Clouds and Respiratory Pathogen Emissions: Potential Implications for Reducing Transmission of COVID-19 [J/OL]. JAMA. 2020, 323 (18): 1837–1838. <https://doi.org/10.1001/jama.2020.4756>.
- [4] van Dam C. The aerodynamic design of multi-element high-lift systems for transport airplanes [J]. 2002, 38 (2): 101–144.
- [5] Quinta A A. Aircraft Multidisciplinary Design Optimization Using Design of Experimental Theory and Response Surface Modeling Methods [J]. Ph. D. Dissertation, Department o Aerospace Engineering. 1997.
- [6] Legresley P, Alonso J, Alonso P. Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics [J]. 2001.
- [7] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. science. 2006, 313 (5786): 504–507.
- [8] Hicks R, Murman E, Vanderplaats G. An assessment of airfoil design by numerical optimization [J]. 1974.
- [9] Vanderplaats G, Hicks R, Murman E. Application of Numerical Optimization Techniques to Airfoil Design [J]. 1975.
- [10] Jones B R, Crossley W A, Lyrantzis A S. Aerodynamic and Aeroacoustic Optimization of Rotorcraft Airfoils via a Parallel Genetic Algorithm [J]. Journal of Aircraft. 2000, 37 (6): 1088–1096.
- [11] Vicini A, Quagliarella D. Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm [J]. Aiaa Journal - AIAA J. 1997, 35: 1499–1505.
- [12] Obayashi S, Takanashi S. Genetic algorithm for aerodynamic inverse optimization problems [C]. In Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALERIA. First International Conference on (Conf. Publ. No. 414). 1995.

- [13] Jameson A, Martinelli L. Aerodynamic shape optimization techniques based on control theory [M]. Springer Berlin Heidelberg, 2000.
- [14] Kanazaki M, Tanaka K, Jeong S, et al. Multi-Objective Aerodynamic Exploration of Elements' Setting for High-Lift Airfoil Using Kriging Model [J]. Journal of the Japan Society for Aeronautical Spaceences. 2007, 54 (3): 419–426.
- [15] 张彬乾, 罗烈, 陈真利, et al. 飞翼布局隐身翼型优化设计 [J]. 航空学报. 2014, 35 (4): 957–967.
- [16] Ju Y, Zhang C, Ma L. Artificial intelligence metamodel comparison and application to wind turbine airfoil uncertainty analysis [J]. Advances in Mechanical Engineering. 2016, 8.
- [17] Kapsoulis D, Tsiakas K, Trompoukis X, et al. Evolutionary multi-objective optimization assisted by metamodels, kernel PCA and multi-criteria decision making techniques with applications in aerodynamics [J]. Applied Soft Computing. 2018, 64: 1–13.
- [18] Zhang Y, Sung W, Mavris D. Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient [C]. 01 2018.
- [19] 陈海, 钱炜祺, 何磊. 基于深度学习的翼型气动系数预测 [J]. 空气动力学学报. 2018, 36 (002): 294–299.
- [20] 惠心雨, 袁泽龙, 白俊强, et al. 基于深度学习的非定常周期性流动预测方法 [J]. 空气动力学学报. 2019, 37 (03).
- [21] Guo X, Li W, Iorio F. Convolutional Neural Networks for Steady Flow Approximation [C]. In SIGKDD. 2016.
- [22] Bhatnagar S, Afshar Y, Pan S, et al. Prediction of aerodynamic flow fields using convolutional neural networks [J]. Computational Mechanics. 2019: 1–21.
- [23] Miyanawala T P, Jaiman R K. An Efficient Deep Learning Technique for the Navier-Stokes Equations: Application to Unsteady Wake Flow Dynamics [J]. 2017.
- [24] Lee S, You D. Data-driven prediction of unsteady flow over a circular cylinder using deep learning [J]. Journal of Fluid Mechanics. 2019, 879: 217–254.
- [25] Belbute-Peres F D A, Economou T D, Kolter J Z. Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction [J]. 2020.
- [26] Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance [J]. Journal of Fluid Mechanics. 2016, 807: 155–166.
- [27] Wang R, Kashinath K, Mustafa M, et al. Towards Physics-informed Deep Learning for Turbulent Flow Prediction [C]. In KDD. 2020.

- [28] Obiols-Sales O, Vishnu A, Malaya N, et al. CFDNet: a deep learning-based accelerator for fluid simulations [C]. In International Conference on Supercomputing, ICS. 06 2020: 1–12.
- [29] Y, H, Qian, et al. Lattice BGK Models for Navier-Stokes Equation [J]. EPL (Europhysics Letters). 1992.
- [30] Schmitt F G. About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity [J]. Comptes Rendus Mécanique. 2007, 335 (9-10): 617–627.
- [31] Zang Y, Street R L, Koseff J R. A dynamic mixed subgrid : cale model and its application to turbulent recirculating flows [J]. Physics of Fluids A: Fluid Dynamics. 1998, 5 (12): 3186.
- [32] Spalart P, Allmaras S. A One-Equation Turbulence Model for Aerodynamic Flows [J]. AIAA. 1992, 439.
- [33] Bengio, Yoshua, Courville, et al. Representation Learning: A Review and New Perspectives [J]. IEEE Transactions on Pattern Analysis Machine Intelligence. 2013, 35 (8): 1798–1828.
- [34] Lecun Y, Bottou L. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE. 1998, 86 (11): 2278–2324.
- [35] Williams R, Zipser D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks [J]. Neural Computation. 2014, 1 (2): 270–280.
- [36] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Nets [J]. ArXiv. 2014.
- [37] Fran?oisLavet V, Fonteneau R, Ernst D. Playing Atari with Deep Reinforcement Learning [J]. Computer Ence. 2015.
- [38] Kipf T N, Welling M. Semi-Supervised Classification with Graph Convolutional Networks [J]. 2016.
- [39] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. 2015.
- [40] Isola P, Zhu J-Y, Zhou T, et al. Image-to-Image Translation with Conditional Adversarial Networks [C]. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. 2017.
- [41] Zhu J-Y, Park T, Isola P, et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss [C]. In Computer Vision (ICCV), 2017 IEEE

International Conference on. 2017.

- [42] Amodio M, Krishnaswamy S. TraVeLGAN: Image-To-Image Translation by Transformation Vector Learning [C]. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. 2019.
- [43] Minar M R, Naher J. Recent Advances in Deep Learning: An Overview [J]. 2018.
- [44] 周莉莉, 姜枫. 图像分割方法综述研究 [J]. 计算机应用研究. 2017, 34 (007): 1921–1928.
- [45] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation [J]. IEEE Transactions on Pattern Analysis Machine Intelligence. 2015, 39 (4): 640–651.
- [46] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer ence. 2014.
- [47] Milletari F, Navab N, Ahmadi S A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation [J]. 2016.
- [48] Zhou Z, Rahman Siddiquee M M, Tajbakhsh N, et al. UNet++: A Nested U-Net Architecture for Medical Image Segmentation [C]. 07 2018.
- [49] Hasan S M K, Linte C A. U-NetPlus: A Modified Encoder-Decoder U-Net Ar-chitecture for Semantic and Instance Segmentation of Surgical Instrument [J]. 2019.
- [50] Huang C, Han H, Yao Q, et al. 3D U²-Net: A 3D Universal U-Net for Multi-Domain Medical Image Segmentation [J]. 2019.
- [51] Mirza M, Osindero S. Conditional Generative Adversarial Nets [J]. 2014.
- [52] Wu Z, Pan S, Chen F, et al. A Comprehensive Survey on Graph Neural Networks [J]. IEEE Transactions on Neural Networks and Learning Systems. 2019.
- [53] Odena A, Dumoulin V, Olah C. Deconvolution and checkerboard artifacts [J]. Distill. 2016, 1 (10): e3.
- [54] Blazek J. Computational fluid dynamics: principles and applications [M]. Butterworth-Heinemann, 2015.
- [55] Hu J, Shen L, Albanie S, et al. Squeeze-and-Excitation Networks [J]. IEEE Transactions on Pattern Analysis Machine Intelligence. 2017, PP (99).
- [56] Liu Z, Sun M, Zhou T, et al. Rethinking the Value of Network Pruning [C]. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. 2019.

- [57] Molchanov P, Tyree S, Karras T, et al. Pruning Convolutional Neural Networks for Resource Efficient Inference [C]. In ICLR. 2017.
- [58] Hamdan M K A, Rover D, Just J. Mass Estimation from Images using Deep Neural Network and Sparse Ground Truth [J/OL]. CoRR. 2019, abs/1908.04387. <http://arxiv.org/abs/1908.04387>.
- [59] Thuerey N, Weissenow K, Prantl L, et al. Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows [J]. AIAA Journal. 2019: 1–12.
- [60] Nair V, Hinton G E. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair [C]. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. 2010.
- [61] Economou T D, Palacios F, Copeland S R, et al. SU2: An Open-Source Suite for Multiphysics Simulation and Design [J]. AIAA Journal. 2015: 1–19.
- [62] Selig M. UIUC airfoil data site [M]. Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign, 1996.
- [63] Geuzaine C, Remacle J-F. Gmsh reference manual: the documentation for Gmsh, a finite element mesh generator with built-in pre-and post-processing facilities [J]. URL <http://www.geuz.org/gmsh>. 2008.

作者在学期间取得的学术成果

发表的学术论文

- [1] **Donglin Chen**, Xiang Gao, Chuanfu Xu, Shizhao Chen, Jianbin Fang, Zhenghua Wang, and Zheng Wang. FlowGAN: A Conditional Generative Adversarial Network for Flow Prediction in Various Conditions. International Conference on Tools with Artificial Intelligence (ICTAI, CCF C) 2020.
- [2] **Donglin Chen**, Jianbin Fang, Chuanfu Xu, Shizhao Chen, Zheng Wang. Characterizing Scalability of Sparse Matrix-Vector Multiplications on Phytium FT-2000+. International Journal of Parallel Programming (IJPP, SCI IF 1.244, 中科院 SCI 分区 3 区). 48(1): 80-97, 2020.
- [3] **Donglin Chen**, Jianbin Fang, Chuanfu Xu, Shizhao Chen, Zheng Wang. Optimizing Sparse Matrix-Vector Multiplications on an ARMv8-based Many-Core Architecture. International Journal of Parallel Programming (IJPP, SCI IF 1.244, 中科院 SCI 分区 3 区). 47(3): 418-432, 2019.
- [4] **Donglin Chen**, Xiang Gao, Chuanfu Xu, Siqi Wang, Shizhao Chen, Jianbin Fang, and Zheng Wang. FlowDNN: A Physics-Informed Deep Neural Network for Fast and Accurate Flow Prediction. (Frontiers of Information Technology & Electronic Engineering FITEE, SCI under review.)
- [5] Shizhao Chen, Jianbin Fang, **Donglin Chen**, Chuanfu Xu, Zheng Wang. Adaptive Optimization of Sparse Matrix-Vector Multiplication on Emerging Many-Core Architectures. IEEE International Conference on High Performance Computing and Communications (HPCC) 2018: 649-658.