

学校代码: 10284

分类号: TP181

密级: 公开

U D C: 004.8

学号: MF21330047



南京大學

硕士学位论文

论文题目 基于空洞卷积和空间注意力的
长短期时序预测研究

作者姓名 李菲菲

专业学位类别(领域) 电子信息(计算机技术)

研究方向 时间序列预测

导师姓名 申富饶教授

2024年5月27日

答辩委员会主席 戴新宇 教授

评 阅 人 武港山 教授

徐明华 教授

论文答辩日期 2024年5月16日

研究生签名:

导师签名:

Research on Long and Short Term Time Series Forecasting based on Dilated Convolution and Spatial Attention

by

Li Fei-Fei

Supervised by

Professor Shen Fu-Rao

A dissertation submitted to

the graduate school of Nanjing University

in partial fulfilment of the requirements for the degree of

MASTER

in

Computer Technology



Department of Computer Science and Technology

Nanjing University

May 27, 2024

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于空洞卷积和空间注意力的长短期时序预测研究

计算机技术 专业 2021 级硕士生姓名：李菲菲

指导教师（姓名、职称）：申富饶 教授

摘 要

时间序列预测在数据分析和决策规划中极为重要，它通过历史数据预测未来趋势，为决策提供洞察。本文聚焦于时间序列预测的两大关键任务：长期时间序列预测和短期时空预测。前者任务的难点在于如何有效地提取并建模长期的时间依赖，而后者任务的关键在于如何精确地建模复杂的空间依赖性，并将其与动态时间依赖相结合。本研究针对这些挑战进行了深入研究，并提出两种算法，通过实验验证了方法的有效性。本文的主要研究内容如下：

1. 本文提出了一种浅层多尺度空洞卷积网络 MSDCN，旨在提升对时间序列数据长期依赖特征的捕捉能力。该网络利用空洞卷积的扩张特性，在不增加模型参数的情况下，通过不同尺度的卷积核和空洞扩张，扩大模型的感知范围。此外，模型整合了自回归模块，以更好地模拟时间序列数据中的线性关系。在多个公共数据集上的实验验证了模型在时间复杂度和空间复杂度上的优势。

2. 本文提出了一种基于稀疏动态图的空间注意力网络结构 ThermoNet，它对空间注意力生成的动态图进行了优化。该结构引入空间注意力机制来生成动态空间图，以捕捉节点间的动态交互。模型使用硬阈值和热力学启发的 Softmax 函数变体进行稀疏化，从而强化关键连接并降低非关键连接的影响。在四个交通数据集上的实验结果验证了模型的有效性。

3. 本文开发了交通流量可视化与预测系统，该系统提供历史交通流量查询、短期交通流量查询和长期交通流量查询功能。通过模型预测和数据分析，系统为用户提供了直观的流量可视化界面和智能出行建议，展现了本文所提出的算法在实际应用中的潜力和价值。

关键词：时间序列预测；多尺度空洞卷积；动态图；空间注意力机制

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Long and Short Term Time Series Forecasting
based on Dilated Convolution and Spatial Attention

SPECIALIZATION: Computer Technology

POSTGRADUATE: Li Fei-Fei

MENTOR: Professor Shen Fu-Rao

Abstract

Time series forecasting is crucial for data analysis and decision-making, predicting future trends from past data. This paper addresses two key time series forecasting tasks: long-term forecasting and short-term spatio-temporal forecasting. The main challenges are effectively modeling long-term temporal dependencies and accurately capturing complex spatial dependencies alongside dynamic temporal ones. To tackle these, we introduce two algorithms, validated through experiments. The main research is as follows:

1. This paper proposes MSDCN, a shallow multi-scale dilated convolution network designed to enhance the detection of long-term dependency features within time series data. The model leverages the expansion properties of dilated convolutions to broaden the model's receptive field without increasing the model's parameters. It also incorporates an autoregressive module to better simulate linear relationships within time series data. Experiments across several public datasets have demonstrated the model's superior performance in terms of time and space complexity.

2. This paper introduces ThermoNet, a spatial attention network architecture based on sparse dynamic graphs. This structure employs a spatial attention mechanism to generate dynamic spatial graphs that capture the dynamic interactions between nodes. The model utilizes a hard threshold function and a thermodynamically inspired variant of the Softmax function for sparsification, highlighting key spatial connections while diminishing the impact of non-essential ones. Experiments on four traffic datasets have validated the model's effectiveness.

3. This paper develops a traffic flow visualization and forecasting system that provides functionalities for querying historical, short-term, and long-term traffic flows. By integrating future predictions and data analysis, the system offers users an intuitive interface for traffic flow visualization and intelligent travel recommendations, demonstrating the practical applicability and value of the algorithms proposed in this paper.

KEYWORDS: Time Series Forecasting; Multi-Scale Dilated Convolution; Dynamic Graph; Spatial Attention Mechanism

目 录

中文摘要	I
ABSTRACT	III
目 录	V
插图清单	IX
附表清单	XI
第一章 绪论	1
1.1 研究背景及意义	1
1.2 研究现状与挑战	2
1.3 研究内容与贡献	6
1.4 论文的组织结构	8
第二章 相关工作	11
2.1 时间依赖建模	11
2.1.1 基于 Transformer 的方法	11
2.1.2 基于多层感知机的方法	13
2.1.3 基于卷积神经网络的方法	15
2.2 空间依赖建模	18
2.2.1 基于静态图的方法	18
2.2.2 基于动态图的方法	21
2.3 本章小结	23
第三章 基于浅层空洞卷积的长期时间序列预测算法	25

3.1	研究动机	25
3.2	算法设计	26
3.2.1	问题定义	26
3.2.2	模型结构	27
3.2.3	多尺度空洞卷积模块	28
3.2.4	自回归模块	29
3.2.5	损失函数	30
3.3	实验与分析	30
3.3.1	实验设置	30
3.3.2	对比实验	33
3.3.3	消融实验	36
3.3.4	模型分析	39
3.4	本章小结	43
第四章 基于稀疏动态图的短期时空预测算法		45
4.1	研究动机	45
4.2	算法设计	47
4.2.1	问题定义	47
4.2.2	模型结构	48
4.2.3	数据嵌入层	49
4.2.4	多头注意力机制	50
4.2.5	稀疏空间图生成	51
4.2.6	图卷积层	54
4.3	实验与分析	54
4.3.1	实验设置	54
4.3.2	对比实验	56
4.3.3	消融实验	57
4.3.4	模型分析	61
4.4	本章小结	62

第五章 交通流量可视化与预测系统	63
5.1 系统研发背景	63
5.2 系统设计	64
5.2.1 系统需求分析	64
5.2.2 系统架构设计	65
5.3 系统实现	67
5.3.1 系统开发环境	67
5.3.2 系统功能模块	67
5.4 系统效果展示	69
5.5 本章小结	72
第六章 总结与展望	73
参考文献	75
致 谢	83
简历与科研成果	85

插图清单

1-1	序列数据分解	2
1-2	本文章节组织结构	8
2-1	不同 Attention 改进比较	12
2-2	DLinear 模型示意图	14
2-3	NBeats 模型示意图	15
2-4	空洞因果卷积示意图	16
2-5	TimesNet 模型将一维时间序列转化为二维	17
2-6	DCRNN 模型示意图	20
2-7	GMAN 模型空间注意力组内和组间计算	22
3-1	时间序列可以被视为不同尺度成分的线性或非线性叠加	25
3-2	MSDCN 网络结构图	27
3-3	自回归模块对预测结果的影响	38
3-4	不同卷积块输出表示的可视化	39
3-5	不同输入长度下的预测性能比较	40
3-6	预测结果可视化	43
4-1	预定义图分布	46
4-2	自学习动态图分布	47
4-3	ThermoNet 网络结构图	48
4-4	多头注意力机制示意图	50
4-5	不同 β 对应的分布	53
4-6	动态空间图可视化	61
4-7	在 PEMS04 数据集上不同 β 上界对预测结果的影响	62

5-1	某地图软件实时交通情况	64
5-2	交通流量可视化和预测平台系统架构	66
5-3	历史流量查询模块实现逻辑	68
5-4	未来流量预测模块实现逻辑	68
5-5	系统首页界面	69
5-6	历史交通流量查询界面	70
5-7	历史交通流量查询结果	70
5-8	短期交通流量预测查询界面	71
5-9	交通流量预测界面展示	72

附表清单

3-1	长期时间序列预测实验数据集	31
3-2	在 ETT 数据集上 MSDCN 实验结果	33
3-3	在其他数据集上 MSDCN 实验结果	34
3-4	单变量时序预测 MSDCN 实验结果	35
3-5	不同尺度卷积模块对预测结果的影响	36
3-6	不同通道分组数对预测结果的影响	37
3-7	MSDCN 与主流模型的时间效率和空间效率对比	42
4-1	常用时空数据集预定义图稀疏度	45
4-2	短期时空预测实验数据集	54
4-3	在 PEMS04 和 PEMS08 数据集上 ThermoNet 实验结果	56
4-4	在 METR-LA 和 PEMS-BAY 数据集上 ThermoNet 实验结果	57
4-5	时间模块和空间模块对预测结果的影响	58
4-6	在 METR-LA 数据集上不同 Embedding 对预测结果的影响	59
4-7	在 PEMS04 数据集上不同 Embedding 对预测结果的影响	60
4-8	在 PEMS04 数据集上不同稀疏操作对预测结果的影响	60

第一章 绪论

1.1 研究背景及意义

随着互联网技术的飞速发展，一系列尖端技术如智能传感器、全球定位系统（GPS）以及移动通信终端等，已经广泛融入到人们的日常生活与工业生产之中，极大地丰富了人们的生活方式并提升了工作效率。在个人生活领域，智能手表等可穿戴设备能够实时监测并记录用户的地理位置、活动步数以及健康状况等关键信息，为用户提供了前所未有的便利与健康保障。在工业生产领域，智能传感器的应用使得环境温度、湿度等关键参数的自动采集成为可能，为生产流程提供了精确的数据支持，从而优化了生产效率和产品质量。在交通领域，GPS技术的普及使得驾驶者能够依据实时交通信息规划出最优的行驶路线，显著提高了出行的效率和安全性。

这些技术的应用不仅依赖于大量数据的收集与分析，而且也产生了庞大的数据资源。在这些数据资源中，时间序列数据因其记录了随时间推移而发生的动态变化而显得尤为宝贵。这些数据包括但不限于气象数据^[1]、电力数据^[2]、金融数据^[3]和交通流量数据^[4]等。通过对这些时间序列数据进行深入的研究，不仅可以理解过去和现在的变化趋势，还可以预测未来的发展动向，为个人、企业和政府的决策提供科学、及时且有效的支持。例如，个人可以根据天气预报来合理安排户外活动；企业可以根据预测的电力需求变化来优化资源配置和能源管理；政府则可以利用交通流量预测来改善城市交通管理，推动智慧城市的建设与发展。因此，时间序列预测任务的研究能够促进生产生活的智能化，为人们做出决策和统筹规划提供有力的建议，具有十分重要的意义。

与此同时，在第四次工业革命大背景下，以人工智能等技术为技术突破口的工业革命正在触及着经济社会的各个领域。在时间序列预测领域，传统的统计学方法由于对非线性关系的处理能力较弱以及对长期依赖关系的建模能力不足，

导致其在处理非线性、非平稳、大规模和高维度等复杂的时间序列数据时预测效果不佳，难以满足日益复杂的实际需求。而深度学习方法的蓬勃发展为这一领域的研究注入了新的活力。深度学习技术从 2006 年重新兴起后，在计算机视觉领域、自然语言处理领域等都发挥了重要的作用。而深度学习模型中的循环神经网络 (RNN)^[5-6]、长短期记忆网络 (LSTM)^[7]和 Transformer^[8]等模型架构，能够有效地对时间序列进行建模，提供了预测的准确性和鲁棒性。随着深度学习技术的不断向前发展和广泛应用，时间序列预测领域的深度学习研究也不断迎来了新的突破和进展，为时间序列预测模型的提升和优化提供了更多的可能性，从而为时间序列预测模型在实际生产和生活场景中的应用奠定了坚实的技术基础。

1.2 研究现状与挑战

时序数据是指在某一时间段内按照特定的采样时间间隔，通过传感器或其他智能设备对变量进行监测记录，从而收集到的一组随着时间变化的连续序列数据。与图像或文本数据不同，时序数据具有以下特征：

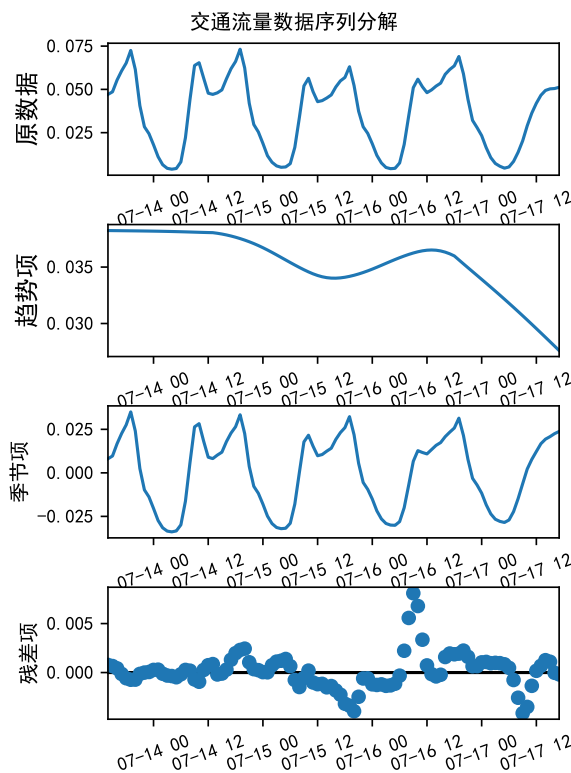


图 1-1 序列数据分解

- 动态性：时间序列数据是带有时间戳的一组随时间变化的序列数据。这意味着随着时间的推移，数据会发生变化，可能会呈现出趋势、季节性或周期性的变化模式。如图 1-1 所示，时间序列数据一般可以分解为趋势项、季节项和残差项^[9-10]。
- 噪声性：在时序数据的采集过程中，由于传感器误差或其他环境因素的影响，往往会引入一些噪声数据。这些噪声数据是随机的，并且可能会对数据的分析和预测产生影响。
- 时滞相关性：时间序列数据中的观测值通常与前面的观测值相关联。这种时滞相关性意味着当前时刻的数据可能受到之前时刻数据的影响，例如，某个月的销售额可能与前几个月的销售额相关。
- 时空相关性：一些时序数据具有明显的空间特征，比如高速公路上多个速度监测点之间收集到的数据在空间上具有关联性。这种时空相关性表明数据不仅在时间上相关，而且在空间上也存在一定的关联性。

针对时间序列数据具有的上述特点，目前时间序列预测主流的研究方法主要分为三类：基于传统统计学的时间序列预测算法、基于机器学习的时间序列预测算法以及基于深度学习的时间序列预测算法。

基于传统统计学的时间序列预测算法主要利用时间序列数据的统计特性进行建模和分析。这些方法包括了移动平均法、指数平滑法和差分自回归移动平均模型等。移动平均法^[11]对历史序列数据进行简单平均的方式来获得预测值，通过平均的方式消除随机因素对数据的影响。这种方法适用于数据变动缓慢、呈现出一定的稳定性趋势的情况。指数平滑法^[11]进一步改进了移动平均法，采用加权平均的方式，对不同时刻的值赋予不同的权重。权重随观测时间的久远程度呈指数型下降，距离预测时间点最远的观察值被赋予最小的权值。这种方法中，参数 α 控制着权重下降的速度， α 越大，更靠近预测时间点的观察值对预测值的贡献越大。差分自回归移动平均模型 (ARIMA)^[12]则结合了自回归模型和移动平均模型，并通过差分操作消除非平稳性。然而，传统方法局限于捕捉线性关系，难以应对非线性关系。对于非平稳性数据，需要进行多次差分才能变成平稳数据，这使得传统方法难以处理大规模和复杂性数据。

基于机器学习的时间序列预测算法一般会预先提取历史序列数据中的统计特性，如滞后阶数、移动平均值等特征，算法主要包含了决策树^[13]、随机森林^[14]和

支持向量机^[15]等。决策树算法将历史序列数据的每一个时间点的值作为单独的一个实例，并根据每个时间点的数值进行节点划分。在涉及到空间数据的情况下，可以建立多个决策树，每个决策树对应于一段特定的时空区域。随机森林算法能够自动对特征进行排序，并自动选择最重要的特征，最终将多个决策树的结果集成起来，以提高预测的准确性。支持向量机算法则通过将时序数据映射到高维的特征空间，通过定义边界的方式来学习输入和输出的关系。然而，基于机器学习的算法通常需要人工提取数据特征，且算法的性能很大程度上取决于特征提取的质量。虽然一些算法可以自动提取特征，但在实际应用中，仍然需要领域专家的知识 and 经验来确保提取的特征能够准确地捕获数据中的重要信息。此外，随着数据量的增加和问题复杂度的提高，通用的特征提取方法会变得不够灵活且不够有效。

随着数据规模的不断扩大和数据内容的日益复杂，基于深度学习的时间序列预测算法受到了越来越多的关注。这些算法能够有效地捕捉数据中的复杂模式，并在多种预测任务中展现出卓越的性能。Transformer 是常用的序列模型，然而，其中的自注意力机制存在一定的冗余，且固有的排列不变性并不适用于时序建模，因此，LogTrans^[16]、Informer^[17]和 Autoformer^[18]等模型对 Transformer 模型中的自注意力机制进行改进，旨在降低自注意力机制的时间和空间复杂度。此外，一些研究通过分解序列的方式对时间维度进行建模，例如 NBeats^[19]、NHits^[20]模型利用基函数的方式将数据解耦为趋势项、季节项和剩余项，然后分别对这些分量进行预测，从而取得了不错的效果。然而，这些方法依赖于某些特定的基函数，尽管在某些情况下能够取得令人满意的结果，但在处理复杂且多变的数据时，它们可能难以有效捕捉数据中的多尺度特征，从而限制了模型的拟合能力。而基于卷积神经网络的 MICN^[21]、TimesNet^[22]等模型使用普通的卷积运算来提取局部依赖关系，这些方法通常只能提取短期的局部依赖关系，缺乏对多尺度时间信息的提取，对于长期依赖关系的建模相对不足。

在处理具有空间相关性的时间序列数据时，传统的卷积神经网络(CNN)^[23]在处理结构化数据（图像数据等）时表现出色，能够有效提取空间特征，然而，它们却难以处理抽象的复杂拓扑图结构。为了解决这一问题，很多研究^[24-25]提出利用图卷积神经网络(GCN)^[26]来提取空间相关性，这是一种专门设计来处理非结构化空间数据的方法。GCN 的核心优势在于其能够捕捉数据中的空间相关性，

这一能力的基础是有效地构建和表示空间图。一些研究工作^[27-28]尝试在预定义的距离矩阵上提取空间信息，但这种方法存在局限性，因为预定义的距离矩阵可能无法准确反映节点之间的真实空间关系，从而削弱了模型在捕捉空间特征方面的能力。在现实世界的应用场景中，节点之间的空间关系是高度动态变化的。鉴于此，众多研究致力于学习隐藏的空间关系。例如，GWNEN^[29]通过自学习图的方法自动学习节点间的邻接关系，而 MTGNN^[30]则利用潜在相关层来学习节点间的动态邻接矩阵，从而更好地适应空间关系的动态变化。此外，利用空间自注意力机制^[30-34]能够生成动态的邻接矩阵，并通过深层模块的叠加，逐步提取多种类型的空间相关性。这些机制使得模型能够更加精细地捕捉空间特征，并适应节点间复杂的相互作用。尽管空间注意力机制在提取时空数据中的空间相关性方面取得了一定的进展，但在当前的研究实践中，该机制仍面临一些挑战。具体而言，空间注意力生成的动态图可能会包含冗余连接，而这些连接并非都对模型的预测性能至关重要。在实际应用中，通常只有一部分节点间的连接对预测结果有显著影响，而大量的非关键连接可能会引入不必要的噪声，增加了模型过拟合的风险。随着模型层数的加深，这种误差可能会逐渐积累，影响模型的泛化能力和预测准确性。

总体来看，现有的时间序列预测研究工作仍存在如下挑战：

- 局部与全局依赖的提取：长期全局依赖通常涉及序列中跨越较长时间的模式和趋势，而短期局部依赖则关注相邻时间点之间的短时关系。模型若过分强调长期依赖性，可能会忽视序列中由突发事件或局部模式引起的短期变化，这些变化对于精确预测同样至关重要。相反，如果模型过度敏感于短期波动，可能会忽略长期趋势的影响。现有研究在预测过程中无法充分兼顾到序列中的局部变化和全局趋势；
- 时间和空间复杂度：时间序列预测方法普遍面临高时间和空间复杂度的问题。这一挑战在处理大规模数据集或在资源受限的环境中尤为明显。高复杂度不仅增加了模型运行的计算成本，也提高了存储需求。在实际应用中，这可能导致模型无法在合理时间内完成预测任务，或难以在资源受限的设备上部署，限制了这些方法的实际应用范围。因此，研究者需要探索更高效的算法，以降低模型的复杂度，使其更适合于实际应用场景。
- 动态空间关系建模：对于具有显著空间关系的时间序列数据，基于地理距

离构建的预定义图是静态的，不足以建模随时间变化的空间节点关系，因此需要动态建模节点之间的空间相关性。现有研究虽然对节点之间的关系进行了动态建模，但节点关系存在冗余，弱相关性连接的存在会使得学习到的图结构不够明确，可能会导致模型错误地赋予弱相关性连接过多的权重，从而稀释了强相关性连接的影响力。

1.3 研究内容与贡献

时间序列预测任务根据预测目标的时间范围可以分为长期时间序列预测和短期时间序列预测。对于长短期时间的判定，并没有公认的界定标准。比如梁宏涛等人^[35]按照时间跨度分为了四类：一天以内为超短期预测，两周以内是短期预测，三年以内是中期预测，三年以外是长期预测。而汪维泰等人^[36]将小于三十分钟的视为短时预测，大于三十分钟的预测视为长时预测。在本文的研究中，将一天内的预测视为短期预测，一天以外的预测视为长期预测。

长期时间序列预测需要同时考虑全局和局部的时间依赖模式，以更准确地预测长期趋势。短期预测对预测的短时准确度更加精确，响应实时变化。此外，有一些时间序列数据具有显式的空间关系，可以辅助时间依赖建模。由于在长期时间序列预测任务中，模型面临的主要挑战在于准确预测较长时间段内的数据趋势。在这种情况下，空间依赖性对预测结果的影响相对较小。因而本文在长期时间序列预测任务中不考虑空间关系的建模。而在短期时间序列预测中，加入空间依赖的建模。本文的研究重点集中在两个主要问题上：第一，针对不具有显式空间关系的数据，如何有效地进行长期时间序列预测；第二，对于具有显式空间关系的数据，如何进行短期时空序列预测，以充分利用空间信息来提高预测的准确性和效率。

针对目前研究工作中存在的挑战，结合时间序列预测应用的实际场景，本文首先设计了一种多尺度的空洞卷积网络模型来更好地同时捕获长期时间序列中的局部依赖和全局依赖，该模型是一种浅层结构。在达到较好的预测性能情况下，在时间复杂度和空间复杂度上都具有优势。进一步地，针对于具有明显空间关系的时间序列数据，本文设计了一种稀疏化的空间自注意力机制来动态地生成空间关系图，使得模型更关注于节点间的强相关连接，从而提升模型的预测

性能。最后，本文基于上述的两种算法，开发了一款交通流量预测软件，旨在为用户提供详尽的交通流量预测服务。软件能够提供未来一周内的长期交通流量预测以及最近一小时内的短期交通流量预测，使用户能够根据这些长短期预测信息做出更为合理的出行规划。本文的主要贡献如下：

1. 本文设计了一种多尺度的浅层空洞卷积网络模型 **MSDCN**。该模型的核心在于利用空洞卷积模块的空洞性质，以在不增加计算成本的前提下，灵活调整卷积核的感受野大小。这一特性使得 **MSDCN** 能够有效地捕获时间序列数据中的局部细节与全局趋势，进而提升对序列依赖关系的建模能力。**MSDCN** 还将多尺度空洞卷积模块与传统自回归模型相结合。这种结合不仅增强了模型对数据中固有线性依赖性的捕捉能力，而且简化了复杂的非线性建模过程。为了验证 **MSDCN** 模型的性能，本文在多个标准时间序列数据集上进行了广泛的实验。实验结果显示，**MSDCN** 在推理速度和预测准确率方面均优于当前主流的预测模型。此外，通过一系列消融实验和深入的分析实验，本文进一步证实了 **MSDCN** 中各个模块的有效性，以及该模型相较于其他研究方法的整体优势。

2. 本文提出了一种受热力学启发的基于稀疏动态空间图的注意力网络模型 **ThermoNet**。**ThermoNet** 模型通过空间注意力机制构建了随时间而变化的空间动态图，并运用硬阈值函数与 **Softmax** 函数的变体对注意力矩阵进行稀疏化处理。这种设计旨在强化模型对空间图中关键连接的关注，同时减少对非关键连接的依赖。**ThermoNet** 模型分别对预定义图和动态图进行图卷积，以充分提取空间依赖关系。本研究选取交通流量领域作为应用场景，在四个公共的交通数据集上进行了实验，并对 **ThermoNet** 网络模型中重要的结构进行了一系列消融实验，以验证方法的有效性。

3. 本文设计了一个智能的交通流量可视化与预测系统。系统以网页应用的形式提供服务，具有良好的用户交互性。系统支持查询历史流量的详细分析数据，使得用户可以全面了解特定区域的历史路况。此外，系统支持未来流量预测，一方面，支持查询未来长期流量数据，用户可以做长期规划；另一方面，支持查询未来短期流量数据，用户可以实时调整出行路线，有效避开拥堵区域。通过这一系统，用户可以合理进行出行规划，减少因交通拥堵带来的时间损失，从而提高城市交通满意度。

1.4 论文的组织结构

本文分为六个章节，文章组织结构如图1-2所示，每个章节的主要内容如下：

第一章为绪论，首先介绍了时间序列预测研究在生产生活中的应用场景及应用价值，并介绍了三种主流的时间序列预测研究方法。随后介绍了时序数据的特点、时间序列预测研究的现状及面临的挑战。最后阐述了本文为解决这些挑战所做的研究工作。

第二章为相关工作，介绍了时间序列预测建模任务中关注的两个核心维度：时间依赖以及空间依赖。对于时间依赖建模，本章从基础模型的角度出发，介绍了三类不同的方法。对于空间依赖建模，本章从空间图的构建方式出发分别介绍了基于静态图的方法以及基于动态图的方法，为后续介绍研究工作提供坚实基础。

第三章提出了一种多尺度的浅层空洞卷积网络模型 MSDCN，用于解决长期时间序列预测问题。本章详细地阐述了所提出方法的研究动机，并介绍了网络模型的各个模块和实现细节，通过一系列消融实验和分析实验进一步验证了方法的有效性。

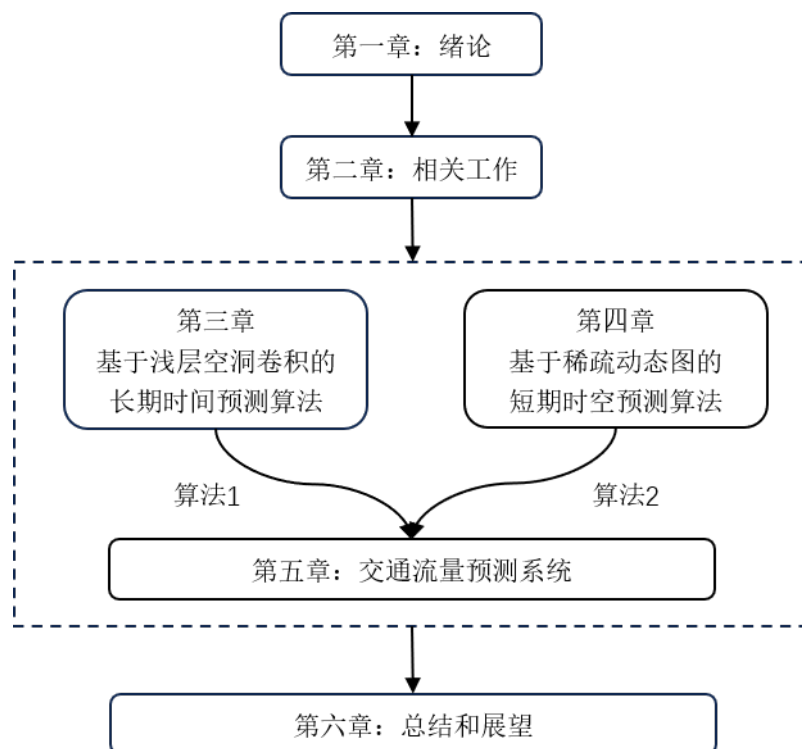


图 1-2 本文章节组织结构

第四章提出了一种基于稀疏化动态图的注意力网络模型 ThermoNet, 用于解决短期时空预测问题。本章针对预定义图的缺陷以及当前研究中动态生成图的不足, 提出了采用多头稀疏空间注意力机制生成动态图的结构, 并通过消融实验验证了模型各个模块的作用。

第五章利用第三、四章提出的长期时间序列预测算法和短期时空预测算法, 搭建了交通流量可视化与预测系统。该系统支持未来流量预测功能, 进一步体现了本文所提出算法的应用价值。

第六章为本文的总结和展望, 对全文的研究工作进行了全面的回顾与总结, 并进一步分析了本文提出方法的不足之处, 并展望了未来研究的可能方向。

第二章 相关工作

2.1 时间依赖建模

无论是长期时间序列预测还是短期时空序列预测任务，时间依赖建模都是非常重要的。最近针对时间依赖建模的研究主要分为三类：基于 Transformer 的方法、基于多层感知机的方法以及基于卷积神经网络的方法。

2.1.1 基于 Transformer 的方法

Transformer 模型^[8]首次提出于 2017 年，最初设计用于自然语言处理 (NLP) 任务，并随后被应用于其他领域。该模型的核心特性是多头注意力机制 (Multi-Head Attention)，它允许模型在处理序列数据时，对序列中的每个元素分配不同的权重。与循环神经网络 (RNN)^[6]和长短期记忆网络 (LSTM)^[7]逐步处理序列的方式不同，Transformer 通过在编码器 (Encoder) 和解码器 (Decoder) 之间采用自注意力机制，显著缩短了信息传播的路径。这种设计对于捕捉序列数据的上下文依赖性至关重要，因为它能够更加有效地处理长距离的依赖关系，从而提升了模型对序列整体语义的理解。

原始 Transformer 模型的自注意力机制在输入序列长度为 L 时，计算复杂度为 $O(L^2)$ 。在长期时间序列预测问题中，往往需要很长的输入序列来建模，而随着序列长度 L 的逐步增加，原始 Transformer 模型的时间复杂度和空间复杂度都呈二次方增长，这极大地限制了 Transformer 模型在处理长序列数据时的有效性和可行性。不少研究针对 Transformer 的注意力机制进行改进，如图 2-1 所示，(a) 图展示了原始的注意力机制；(b) 图是一种稀疏化的注意力机制，减少不必要的注意力计算来降低复杂度，代表模型 Reformer^[37]、Informer^[17]等；(c) 图展示了以 Log 形式稀疏化的注意力机制，同样通过减少不必要的注意力计算来降低复杂度，代表模型 LogTans^[16]；(d) 图展示了自相关机制，以发现序列级别的依赖

关系，代表模型 **Autoformer**^[18]。下面将分别介绍这些具有代表性的模型。

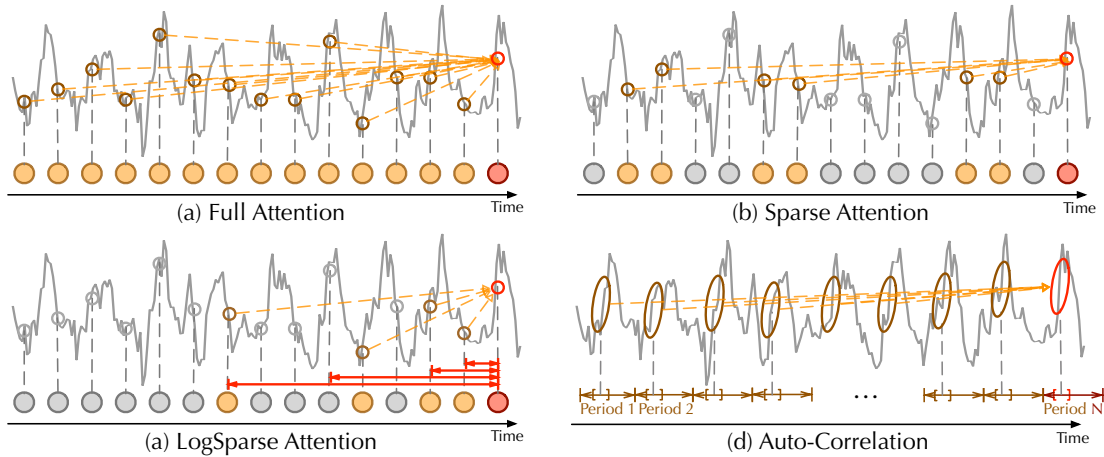


图 2-1 不同 Attention 改进比较¹

自注意力分数揭示了序列内部元素之间的相关性，这是理解和处理序列数据时至关重要的信息。然而，值得注意的是，自注意力分数往往呈现出长尾分布的特征，其中大部分的注意力权重集中在少数几个关键的元素上，而其他大部分元素的注意力权重则相对较小。这表明，在序列中只有少数的元素对当前的预测或决策具有显著的作用。因此，可以将原始自注意力机制进行稀疏化。**LogTrans** 模型^[16]提出了对数稀疏自注意力机制 (**LogSparse Self-Attention**)，让每个点只关注前一个具有指数步长的点，也就是只选择具有指数步长的点参与点积运算，从而将计算复杂度降低到了 $O(L \log L)$ 。类似地，**Informer** 模型^[17]提出了概率稀疏自注意力机制 (**ProbSparse Self-Attention**)，通过 **KL** 散度计算 **Query** 与 **Key** 的相似度，进一步挑选出相似度最高的 u 个 **Query** 参与计算，减少了参与 **Attention** 计算的 **Query** 和 **Key** 的个数，从而在不丢失重要注意力分数的同时，降低了计算复杂度。此外，**Informer** 还提出了自注意力蒸馏，通过卷积和最大池化减少网络维度和网络参数量。**Reformer** 模型^[37]使用局部敏感哈希 (**Locality-Sensitive Hashing**) 来代替点积注意力机制，通过哈希算法将高维空间中相似的向量以高概率分配到相同的哈希值。在每个哈希桶内，模型独立地执行点积注意力计算，从而实现了对自注意力机制的高效近似。

虽然通过将自注意力矩阵稀疏化降低了原始自注意力机制的时间复杂度和空间复杂度，但是自注意力机制具有排列不变性的特性，即使可以使用各种类型的位置编码技术保留排序信息，但在其上应用自注意力后仍然不可避免地会丢

¹ 图片引用自文献^[18]

失时间信息。对于自然语言处理等语义丰富的应用领域来说，通常不用关注序列顺序，比如，即使交换一个句子中的某些单词位置，句子的语义在很大程度上也会得到保留。然而，在分析时间序列数据时，时间序列数值数据本身不包含任何语义信息，模型更应该关注于对一组连续的点之间的时间变化进行建模，而较少地关注逐点的语义信息。因此，研究者需要在自注意力机制的基础上，引入一些与时序数据特点有关的结构。针对此问题，Autoformer 模型^[18]提出了自相关机制（Auto-Correlation Mechanism），包括了基于周期的依赖关系以及时延信息聚合模块。自相关机制首先将时域向量转换到频率域，通过快速傅里叶变换计算序列级别的依赖关系，然后根据周期长度对其进行对齐和聚合。此外，Autoformer 还提出了一种深度分解架构，将预测结果优化和序列分解交替进行，即从隐变量中逐步地分解趋势项和周期项，实现一种渐进式的分解，从而渐进式地提取长序列中的时间依赖。同样为了让模型关注序列的依赖信息，PatchTST 模型^[38]、Crossformer 模型^[39]将时间序列分割成多个子序列级别的片段（Patches），并对每一段分别进行编码，保留了时间序列的局部语义信息，从而有利于捕捉序列的局部依赖信息。针对时间序列往往具有非平稳性的特点，Non-stationary Transformer 模型^[40]针对性地改进了注意力机制。其先将序列平稳化，然后在计算注意力时引入了非平稳性，通过一系列的理论推导得到平稳化前和平稳化后的序列注意力表达式，并利用该理论提出了去平稳化注意力机制（De-stationary Attention），从而减轻深度模型的过度平稳化预测问题。

2.1.2 基于多层感知机的方法

基于 Transformer 的模型是提取长序列中元素间相关性的较为成功的解决方案，然而虽然基于 Transformer 的模型提出了各种结构来匹配时间序列数据具有子序列依赖的特点，但其自注意力机制固有的排列不变性不可避免地会造成时间信息的丢失。DLinear 方法^[41]在其实验中通过随机打乱输入数据的形式，得到了以下结论：即使打乱数据，只简化 Transformer 模型的方法 Informer 预测性能不会发生太大的变化。相反，那些融合了序列依赖关系的模型，例如 Autoformer 和 FEDformer^[42]，却在打乱数据后预测性能表现出较大幅度的下降。这一发现揭示了设计序列相关的结构在保留时间关系方面的重要性。

近期的研究^[41,43-44]发现，仅依赖多层感知机（MLP）结构构建的时间序列

预测模型也能取得较为满意的预测结果。这些研究表明，在适当的设计和训练下，MLP 模型能够有效捕捉时间序列数据的复杂模式，并在预测任务中表现出色。尽管 MLP 不具备处理序列数据长距离依赖的固有能力，但其在某些场景下的预测性能仍不容忽视，这为时间序列预测领域提供了一种简洁而有效的建模选择。

LightTS 模型^[45]通过将一维时间序列数据转换为二维结构，有效地组织和处理时序信息。该模型采用连续采样和间隔采样的策略，将原始数据映射到一个二维平面上。在这个二维表示的基础上，LightTS 运用多层感知机结构，通过时间投影和空间投影两个关键步骤对数据进行深度特征提取和转换。相比于 LightTS 模型，DLinear^[41]和 NLinear^[41]方法则更为简洁。如图2-2所示，DLinear 方法通过执行移动平均操作来提取原始输入数据的趋势分量，并从原始数据中移除这一趋势分量 (Trend)，从而得到剩余的剩余分量 (Reminder)。随后，这两个分量分别通过一个线性层进行处理，其输出特征经过合并，形成最终的预测结果。这种基于趋势分解的方法显著提升了线性层的表达能力和预测性能。NLinear 方法则更为简洁，它仅使用一个线性层，并在该层前后应用归一化操作。尽管这两种方法在结构上极为简练，但它们在多个公共数据集上的性能指标均优于一些复杂的基于 Transformer 的模型，如 FEDformer、Autoformer 和 Informer 等。并且其计算成本、训练时间和推理时间都远小于基于 Transformer 的模型。此外，谷歌发布的 TSMixer 模型^[46]同样证实了基于多层感知机的模型在时间序列预测领域的潜力。TSMixer 模型通过采用线性层替代 Transformer 架构中的注意力机制，提出了一种新颖的时间序列预测方法。该模型借鉴了计算机视觉领域 MLP-Mixer^[47]的方法论，通过在多个多层感知器之间交替执行时间维度和特征维度的混合操作，实现了对时间序列数据中的时间模式和交叉变量的有效捕捉。

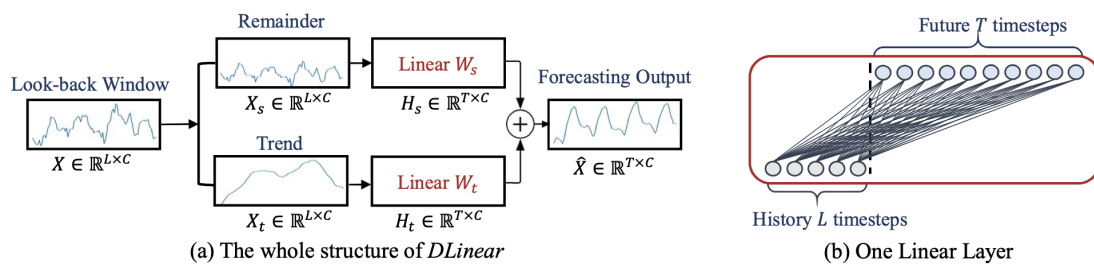


图 2-2 DLinear 模型示意图¹

1 图片引用自文献^[41]

NBeats 系列模型^[19-20,48]则提出了一种可解释的时序学习架构，其核心思想在于将时间序列预测转化为各种基础函数的组合。如图2-3所示，NBeats 整体结构是一个基于前向和后向残差连接的全连接层堆栈式的深度神经网络架构。在这一架构中，前向分支负责逐步聚合预测结果，而后向分支则负责删除前一个模块中可以很好近似的部分，从而使后续模块能够专注于那些尚未被良好预测的部分。这种设计不仅提高了模型的预测精度，也简化了下游模块的预测任务。NBeats 进一步引入了两种具有高度可解释性的模块——趋势模型 (Trend Model) 和季节性模型 (Seasonality Model)，分别用于捕捉时间序列数据的趋势和周期性成分。这些模块的设计基于先验知识，通过确定周期函数、多项式等函数形式，使得模型能够学习到时间序列的独特特性。而随着预测长度的增加，NBeats 误差会增大。为了解决这一问题，NHITS 模型^[20]在 NBeats 的基础上进行了改进，引入了最大池化和插值技术，来对不同尺度进行预测。

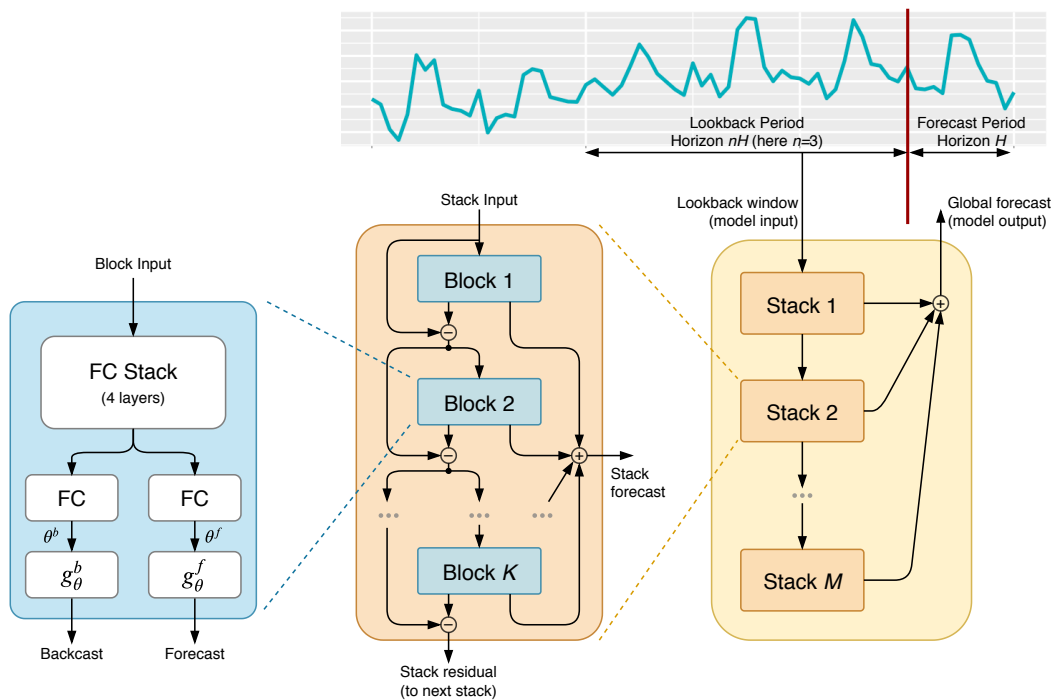


图 2-3 NBeats 模型示意图¹

2.1.3 基于卷积神经网络的方法

卷积神经网络 (CNN) 的核心机制是卷积层和池化层，最开始用于解决计算机视觉任务中的图像识别任务。在这些任务中，由于图像数据 (不考虑颜色通

¹ 图片引用自文献^[19]

道)本质上是二维的,包含了宽度和高度信息,因二维卷积层与图像的自然二维结构相匹配而得到了较为广泛的应用。对于时间序列数据,由于其本质上是一维的,即数据点随时间单向排列,因此大多数研究使用一维卷积。一维卷积层通过在序列上滑动卷积核,能够捕捉时间序列中的局部模式和长期依赖关系。

时间卷积网络(TCN)^[49]通过采用因果卷积和空洞卷积操作,实现了对序列数据的高效并行处理。如图2-4所示,因果卷积的设计确保了模型在生成当前输出时仅使用当前及之前的序列元素,从而避免了未来信息的泄露。而空洞卷积则通过在卷积核中引入扩张因子,扩大了模型的感受野,使得模型能够在不增加参数数量的前提下,捕捉到更广泛的上下文信息。TCN的这种设计使得模型能够在保持未来信息隔离的同时,有效地捕捉序列数据中的长期依赖关系。TCN的并行处理特性特别适合于处理大规模序列数据集,显著提升了模型在训练和推理阶段的效率。

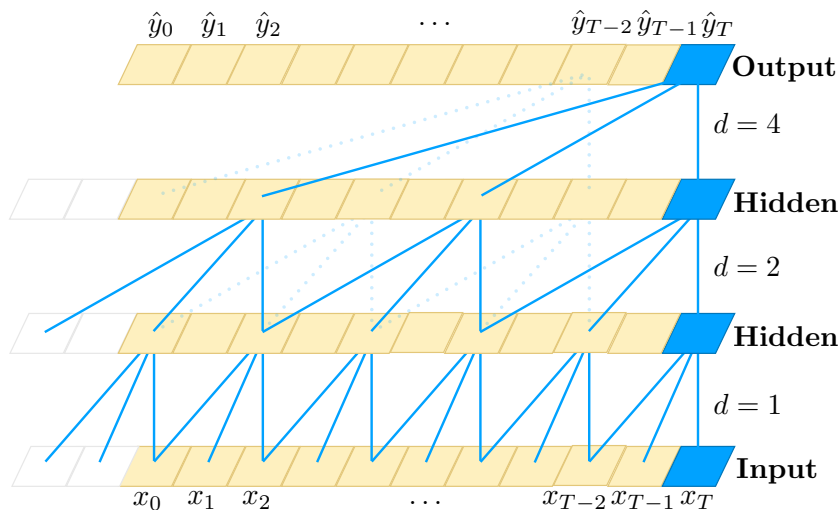


图 2-4 空洞因果卷积示意图¹

LSTNet^[50]模型则充分利用了循环神经网络和卷积神经网络的优势,通过循环神经网络提取长期依赖关系,同时利用卷积神经网络捕获短期局部依赖模式。尽管这种结合提高了模型的表达能力,但循环神经网络存在梯度消失的问题,并且传统的卷积网络结构在处理长期依赖关系时通常需要更深的网络结构,这不仅增加了模型的空间和时间复杂度,也可能导致过拟合和低计算效率。为了解决这些问题,MICN^[21]模型采用一种多尺度等距卷积,这是一种类似于因果卷积的结构,旨在更有效地提取数据中的局部和全局特征。尽管这种方法在一定程

¹ 图片引用自文献^[49]

度上克服了传统卷积网络的限制，但它与 TCN 模型存在相似的缺陷，即对于每个输入，模型只能看到位于该点之前时刻的输入。然而，在时间序列预测问题中，模型输入本身并不包含未来序列，这限制了其对时序信息的提取能力。针对这一问题，SCINet^[51]模型通过树状结构对原始时间序列进行分解，每次下采样都将时间序列分解为具有不同时间分辨率的子序列。SCINet 使用不同的卷积滤波器对这些子序列进行特征提取，从而学习到多分辨率下的动态时间特征。

与上述方法不同，TimesNet^[22]模型则创新性地在一维时序数据扩展到二维空间进行分析。如图2-5所示，在 TimesNet 中，模型首先通过傅里叶变换计算得到时间序列数据强度最大的 k 个频率以及对应的 k 个周期，这一步骤使得模型能够识别时间序列中的主要周期性模式。然后根据这 k 个周期对原始的一维时间序列进行折叠，转化成多个不同频率的二维的时间序列。随后，在二维时间序列上，利用经典的 Inception 卷积网络^[52]提取丰富的特征表示，Inception 网络通过并行的不同尺寸的卷积核和池化层来捕获数据的多尺度特征。最后，TimesNet 对提取到的特征进行信息聚合和融合，得到最终的输出。TimesNet 提出的将一维转化为二维序列的思路给研究带来了新的视角，为时间序列分析开辟了新的研究方向。然而，TimesNet 具有较高的时间和空间复杂度，在实时预测场景中，其应用可能会受到限制。

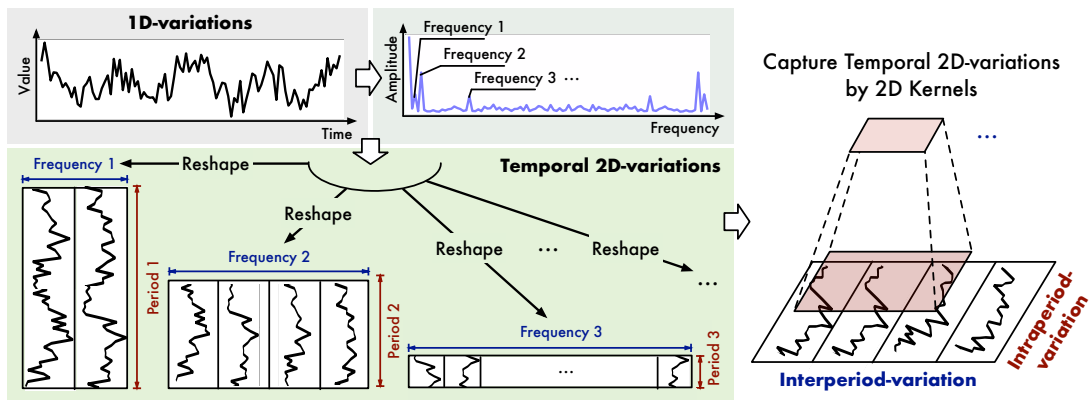


图 2-5 TimesNet 模型将一维时间序列转化为二维¹

1 图片引用自文献^[22]

2.2 空间依赖建模

在处理具有明显空间关系的时间序列数据时,为了实现更准确的预测,研究者必须同时考虑时间和空间两个维度的相关性。时间序列数据不仅在时间轴上展现出连续性和动态变化,而且在空间分布上也常常表现出复杂的相互作用和依赖性。因此,必须考虑时间和空间的联合建模。而空间依赖建模建立在空间图上,空间图的构建方式一般可以分为静态图(包括预定义图和自适应图)以及动态图。本节将重点介绍空间依赖性建模的方法,依据图的构建方式,分别介绍基于静态图和基于动态图的建模策略。在介绍这些方法时,本文也将简要介绍这些模型将时间依赖性建模与空间依赖性建模进行有效聚合的方式。

2.2.1 基于静态图的方法

设空间图的结构 $\mathcal{G} = (V, A)$, 其中 V 表示图中的节点, $|V| = N$, 代表共有 N 个节点。 A 表示邻接矩阵, 其定义了节点间的连接关系。邻接矩阵的构建的方式有很多种, 本文介绍四种常用的构建方法:

- **Distance Graph (距离图)**^[24,27] 其主要思想是采用阈值高斯核^[53], 根据节点间的距离计算节点连接的权重, 从而构建一个能够反映地理空间邻近性的图结构。然后设定一个合理的阈值, 限制高斯核的影响范围, 确保只有邻近的节点间存在显著的连接, 具体的计算方式如下

$$A_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}(v_i, v_j)^2}{2\sigma^2}\right), & \text{如果 } \text{dist}(v_i, v_j) \leq k, \\ 0, & \text{其他,} \end{cases} \quad (2-1)$$

其中, v_i, v_j 表示节点, $\text{dist}(v_i, v_j)$ 表示从节点 v_i 到节点 v_j 的地理距离, σ 表示距离的标准差, k 表示阈值。

- **Neighbor Graph (相连图)** 根据节点间是否连通定义的邻接阵, 又称 0-1 矩阵。具体的计算方式如下:

$$A_{ij} = \begin{cases} 1, & \text{如果 } v_i \text{ 和 } v_j \text{ 相邻/连通,} \\ 0, & \text{其他.} \end{cases} \quad (2-2)$$

- **Functionality Graph (功能图)**^[54-55] 在实际应用中, 具有相似功能的地方往往其时序数据具有高度相关性, 比如靠近学校的路口和靠近医院的路口的交通流量数据通常会展现出类似的模式。设有 K 个不同的功能划分, 对于每个节点 v_i , 定义一个对应的功能向量 $F_i = \{f_{i,1}, f_{i,2}, \dots, f_{i,K}\}$, 其中每个元素 $f_{i,k}$ 代表节点 v_i 对应第 k 个功能类别的存在性。通过计算节点间功能向量的皮尔逊相关系数, 可以构建功能图, 具体构建方式如下:

$$A_{ij}^F = \begin{cases} \frac{\sum_{k=1}^K (f_{i,k} - \bar{F}_i)(f_{j,k} - \bar{F}_j)}{\sqrt{\sum_{k=1}^K (f_{i,k} - \bar{F}_i)^2} \sqrt{\sum_{k=1}^K (f_{j,k} - \bar{F}_j)^2}}, & \text{如果 } i \neq j, \\ 0, & \text{其他,} \end{cases} \quad (2-3)$$

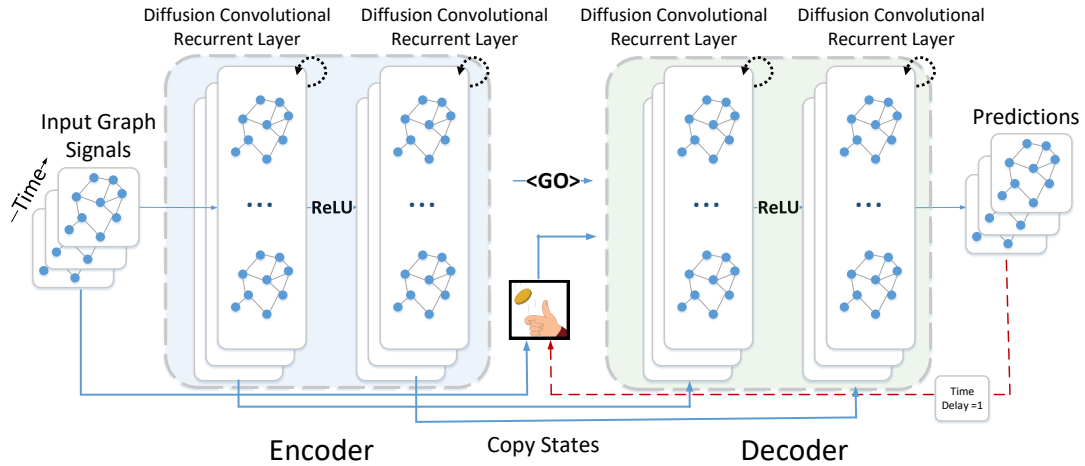
其中, 功能越相似的节点在图中对应的值越大。

- **Temporal Pattern Similarity Graph (时序相似图)**^[28,56-57] 设每个节点 v_i 的时间序列向量为 $T_i = \{t_{i,1}, t_{i,2}, \dots, t_{i,L}\}$, 其中, L 是序列长度, 每个元素 $t_{i,l}$ 代表节点在时间步 l 对应的值。通过计算序列间的皮尔逊相关系数或者运用动态时间规整 (DTW) 算法^[58], 可以构建时序相似图, 以皮尔逊相关系数为例, 具体构建方式如下:

$$A_{ij}^T = \begin{cases} \frac{\sum_{p=1}^P (t_{i,p} - \bar{T}_i)(t_{j,p} - \bar{T}_j)}{\sqrt{\sum_{p=1}^P (t_{i,p} - \bar{T}_i)^2} \sqrt{\sum_{p=1}^P (t_{j,p} - \bar{T}_j)^2}}, & \text{如果 } i \neq j, \\ 0, & \text{其他.} \end{cases} \quad (2-4)$$

在上述四种构建空间图的方法中, 距离图和相连图的创建通常可以直接从传感器的地理位置信息中获得, 这是在数据收集阶段最容易获取的数据类型。而功能图的构建需要对每个节点进行详细的功能标定, 需要额外的数据分析工作。时序相似图则需要选定一个合适的序列长度, 并计算在这个长度上的时序相似, 而且时序相似可能会随着时间变化而变化, 这时静态的时序相似图可能会失效。

DCRNN^[27]模型是最经典的模型之一, 后续很多模型都基于 DCRNN 结构搭建。在预定义距离图的基础上, DCRNN 将具有时空特征的交通流量数据建模为有向图上的扩散过程, 如图2-6所示。DCRNN 用扩散卷积代替了原始 GRU 的全连接部分, 构成了扩散卷积循环神经网络。通过在图上的双向随机游走来捕获空间依赖性, 同时采用具有计划采样的编码器-解码器架构来捕获时间依赖性。

图 2-6 DCRNN 模型示意图¹

STGCN^[24]同样也利用了图卷积在距离图上提取空间特征，但它结合了门控时间卷积和图卷积构成了纯卷积块结构，通过堆叠的时空卷积块来提取时空特征。上述两种模型都使用了距离图来构建邻接矩阵，MLSTGCN^[28]则使用了多图构建的方式，不仅包含了上述四种图，还包括了直方图分布图，共五种类别的图。在同一类型图内部以及不同图之间提取节点的相关性，通过多图学习来实现信息上的增益与互补，进而提升模型性能。

然而，这些模型通常依赖于预定义的图结构，这可能无法准确反映实际的空间关系。首先，预定义图结构的构建过程可能引入误差，例如数据收集过程中的错误或缺失，这可能导致模型对空间关系的误判。其次，预定义的邻接矩阵并不总是能够真实地反映节点间的物理连接性。以距离矩阵为例，即使两个节点在地理上相邻，它们也可能在流量上没有直接的相关性；相反，相隔较远的节点可能由于道路网络的连通性而表现出强烈的空间相关性。因此，不少研究提出了采用自适应图的方式来自适应地学习图。

AGCRN^[59]和 Graph WaveNet^[29]通过可学习的节点嵌入乘积来构建自适应图，自动学习隐藏的空间依赖信息，摆脱了对预定义邻接矩阵的依赖，其构建方式为：

$$A = \text{Softmax}(\text{ReLU}(E_1 E_2^T)), \quad (2-5)$$

其中， $E_1, E_2 \in R^{N \times c}$ 是可学习的节点嵌入。在得到自适应图后，AGCRN 采用了节点自适应参数学习的图卷积与 GRU 相集成的网络结构。而 Graph WaveNet

¹ 图片引用自文献^[27]

则通过图卷积网络和门控 TCN 的串行结构来提取时空依赖。

MTGNN^[30]设计了具有混合跳播层的图卷积模块以及具有扩张初始层的时间卷积模块，两种模块相互交织来捕获时空相关性。同时，其引入了另一种自适应图构建方式：

$$\begin{aligned}
 M_1 &= \tanh(\alpha E_1 \Theta_1), \quad M_2 = \tanh(\alpha E_2 \Theta_2), \\
 A &= \text{ReLU}(\text{Tanh}(\alpha(M_1 M_2^T - M_2 M_1^T))), \\
 \text{for } i &= 1, 2, \dots, N, \quad idx = \text{argtopk}(A[i, :]), \\
 A[i, -idx] &= 0,
 \end{aligned} \tag{2-6}$$

其中， $E_1, E_2 \in \mathbb{R}^{N \times C_{embed}}$ 是可学习的节点嵌入， Θ_1, Θ_2 是线性层参数， α 是超参， argtopk 表示只取与每个节点关系最大的 k 个节点，可以稀疏化邻接矩阵。但是，需要注意的是，在训练的过程中，自适应图会不断地更新和学习，在训练结束后自适应图却仍然是固定的，而现实世界中的空间关系可能随时间动态变化。因而必须开发能够动态学习和适应真实世界变化的图结构的模型。

2.2.2 基于动态图的方法

随着注意力机制在众多领域的成功应用，很多研究利用注意力机制的特性来动态计算节点之间的空间相关性。ASTGNN^[33]模型通过在静态邻接矩阵的基础上添加动态的权重关系来实现动态图的构建。模型利用其设计的时间趋势感知多头注意力机制产生的特征表征 $Z_t^{(l-1)}$ 作为输入，计算空间相关权重矩阵 S_t ：

$$S_t = \text{Softmax} \left(\frac{Z_t^{(l-1)} Z_t^{(l-1)T}}{\sqrt{d_{model}}} \right) \in \mathbb{R}^{N \times N}, \tag{2-7}$$

然后，计算静态邻接矩阵 A 与空间相关权重矩阵 S_t 之间的逐元素点积，从而生成了动态图，即 $A' = A \odot S_t$ 。由于空间相关权重矩阵的输入 Z_t 是动态变化的，动态图 A' 由于输入 Z_t 的变化也会发生动态变化。

StemGNN^[60]模型引入了一种潜在相关层，该层能够在无需使用预定义邻接矩阵的情况下，直接从数据本身自动学习并揭示节点间的潜在相关性。该层首先将序列输入到一层 GRU 模型中提取原始数据的时间特征，然后对 GRU 输出

的最后一层隐藏状态 R 使用自注意力的方式计算邻接矩阵。在得到邻接矩阵后, StemGNN 将其转换到频域, 并使用谱图卷积网络提取空间依赖。此外, StemGNN 也将时间序列分解到频率域上, 在此基础上学习了频域上的时序特征表示, 从而在频率层面捕捉时空依赖。

GMAN^[32]模型在计算空间注意力时, 首先将前一层的隐藏状态与节点嵌入进行拼接, 以此作为计算空间注意力的基础。为了应对大规模节点带来的高时空复杂度问题, GMAN 模型引入了分组空间注意力网络, 如图2-7所示。具体而言, GMAN 模型将 N 个节点随机划分为 G 个组, 其中每组包含了 $M = N/G$ 个顶点。在组内, 模型通过计算节点间的注意力权重来捕捉局部空间特征。随后, 应用最大池化操作以提取每个组的代表性特征, 从而实现了对局部空间特征的有效汇总。在组间层面, GMAN 模型进一步计算组间的注意力, 以生成全局空间特征。这一步骤有助于模型理解不同组之间的相互关系, 并为每个组分配全局上下文信息。最终, 模型将每个组的全局空间特征与节点的局部空间特征进行融合, 形成每个节点的综合空间特征表示。通过这种分组策略, GMAN 降低了空间注意力的计算复杂度, 并且利用组内和组间两个渠道计算节点特征。但其随机划分分组的方式, 可能会导致完全不相似的节点被划分到一组, 此时计算组内的代表性特征就可能会干扰到特征提取的过程, 影响模型对节点特征的代表性。

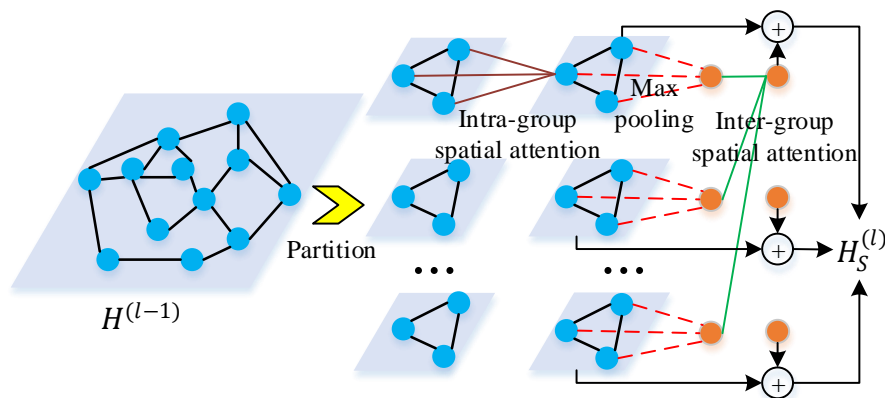


图 2-7 GMAN 模型空间注意力组内和组间计算¹

PDFormer^[61]模型同样也使用了空间注意力机制来提取动态图, 其创新之处在于引入了两个图掩码矩阵, 用以优化邻接矩阵的构建。由于空间自注意力机制生成的是全面的节点间关系, 实际上只有部分节点对之间的相互作用是至关重要的。为了聚焦于这些关键连接, PDFormer 利用掩码矩阵对邻接矩阵进行筛

¹ 图片引用自文献^[32]

选，从而剔除那些非必要的节点联系。这两个掩码矩阵的构建方法与静态图的构建有所类似，包括了地理掩码矩阵 M_{geo} 和语义掩码矩阵 M_{sem} 。地理掩码矩阵 M_{geo} 的设计基于节点间的地理邻近性，通过设定一个阈值来判断节点间的距离是否足够近。如果两个节点之间的地理距离小于这个阈值，则在掩码矩阵中对应的权重设置为 1；否则，权重设置为 0。这样的设计有助于模型专注于局部区域内的节点相互作用。语义掩码矩阵 M_{sem} 则关注节点间的时间序列数据的语义相似性。PDFormer 采用动态时间规整 (DTW) 方法来衡量节点历史时间序列数据的相似度，并选择相似度最高的 K 个节点作为每个节点的语义邻居。这一步骤使得模型能够识别并保留那些在时间序列特征上具有强相关性的节点连接。通过结合这两个掩码矩阵，PDFormer 对节点间的连接进行了筛选，从而捕获更为关键的空间依赖。但这种掩码的方式基于一定的先验知识，可能会导致连接信息的丢失。

2.3 本章小结

本章从时间依赖建模和空间依赖建模两个角度介绍了时间序列预测任务的相关工作。对于时间依赖建模，本章从基础模型分类角度详细阐述了基于 Transformer 的方法、基于多层感知机的方法以及基于卷积神经网络的方法，并指出了这些方法在处理长期依赖关系时存在的局限性。本文第三章将针对长期依赖建模提出一种基于多尺度空洞卷积神经网络的模型。对于空间依赖建模，本章从空间图的构建方式详细阐述了基于静态图的方法以及基于动态图的方法。本文第四章将通过具体数据进一步分析目前研究中使用静态图和动态图的不足之处，并提出一种基于稀疏动态图的网络模型。

第三章 基于浅层空洞卷积的长期时间序列预测算法

在长期时间序列预测任务中，模型的目标是基于历史数据预测未来较长一段时间内的序列变化，这对于多种应用场景中的决策规划至关重要。本章首先分析了建模这类任务时现有方法存在的问题，随后提出了基于浅层多尺度空洞卷积的网络结构（Multi-Scale Dilated Convolution Network, MSDCN）。本章进行了一系列对比试验和消融实验，实验结果表明，相较于 TimesNet 等主流网络，MSDCN 网络不仅实现了更快的训练和推理速度，还能提供更精确的预测结果。

3.1 研究动机

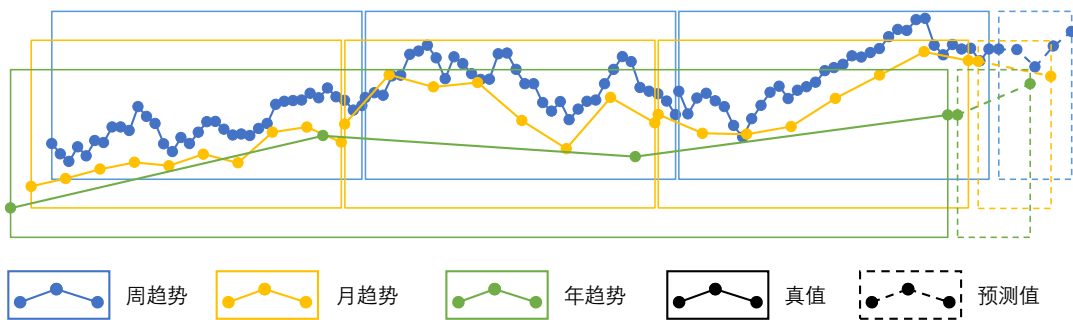


图 3-1 时间序列可以被视为不同尺度成分的线性或非线性叠加

时间序列建模的本质是有效提取时间序列的不同尺度成分，并根据提取出的成分进行相应任务的建模过程。如图 3-1 所示，一条时间序列数据可以被视为周趋势、月趋势、年趋势等多尺度成分的线性或非线性叠加。多尺度信息不仅包含了时间序列的宏观趋势，还蕴含了微观层面的波动和异常。这些信息对于预测未来发展趋势、识别潜在的风险点、以及制定有效的决策策略都具有不可估量的价值。因此，研究能够捕捉和利用多尺度信息的建模方法，对于提升时间序

列预测性能非常重要。一些基于时序分解的方法^[18,41]通过应用移动平均和池化等技术对时间序列数据进行预先分解，分离出时间序列中的不同成分，进而分别对不同成分建模。然而，它们在执行移动平均操作时存在一个显著的局限性：仅依赖于单一尺寸的卷积核。这种局限性限制了方法的灵活性，导致它们难以提取时间序列中的多尺度信息。而采用多尺度的普通卷积核进行多尺度信息提取，虽然可行，但这种方法往往需要较大的卷积核尺寸来捕获更广泛的尺度信息。随着卷积核尺寸的增加，模型所需的参数量也会相应增加，这不仅会导致计算资源的大量消耗，还可能导致过拟合问题，尤其是在数据量有限的情况下。

空洞卷积的扩张特性能够缓解上述问题。最初，空洞卷积在图像领域被引入^[62]，旨在克服传统卷积和池化操作在下采样时的局限性。通过在卷积核之间引入间隔（即空洞），空洞卷积扩大了卷积操作的感受野，使其能够更好地捕捉更远处的像素信息。因此，网络能够在不增加参数负担的情况下，捕获更广阔的上下文信息。空洞卷积的这种特性不仅在图像处理领域有用，用于时间序列建模也具有一定的优势。通过使用不同尺度的空洞卷积，时序模型不仅可以捕获到不同尺度的趋势和模式，而且避免了因参数增加而导致的计算负担和过拟合风险。

基于上述背景，本文提出了一种浅层多尺度空洞卷积的网络模型 MSDCN，该方法采用不同大小的空洞卷积来捕捉时间序列中的多尺度特征。MSDCN 模型通过在不同大小的卷积核中采用以指数级增长的空洞数，更加灵活地捕捉时间序列中不同尺度的成分信息。这种基于空洞卷积的多尺度时间序列建模方法具有一定的优势，它不仅能够更好地捕捉时间序列中的局部和全局特征，还可以适应不同尺度的趋势和模式。因此，该方法在时间序列预测任务中具有潜在的应用价值。

3.2 算法设计

3.2.1 问题定义

长期时间序列预测是一项利用历史数据预测未来较长时间跨度的时序值的任务。具体而言，该问题可以形式化定义如下：输入数据表示为：

$$\mathbf{X} = [X_1^t, X_2^t, \dots, X_C^t]_{t=1}^T \in \mathbb{R}^{T \times C}, \quad (3-1)$$

其中, T 是回看窗口长度, 即用于预测的历史数据的时间步数; C 是变量个数, X_i^t 表示在时刻 t 第 i 个变量的值。

预测输出表示为

$$\mathbf{Y} = [X_1^t, X_2^t, \dots, X_C^t]_{t=T+1}^{L+T} \in \mathbb{R}^{L \times C}, \quad (3-2)$$

其中, L 是预测窗口长度, 即想要预测的时间步数。在本问题中, 使用过去的 T 个时间步的时序信息, 预测未来 L 步的时序, 且 $L > T$ 。问题的目标是学习一个映射函数 \mathbf{f} , 该函数能够根据输入的历史时序数据 \mathbf{X} 来预测未来的时序数据 \mathbf{Y} , 即

$$\mathbf{Y} = \mathbf{f}(\mathbf{X}). \quad (3-3)$$

3.2.2 模型结构

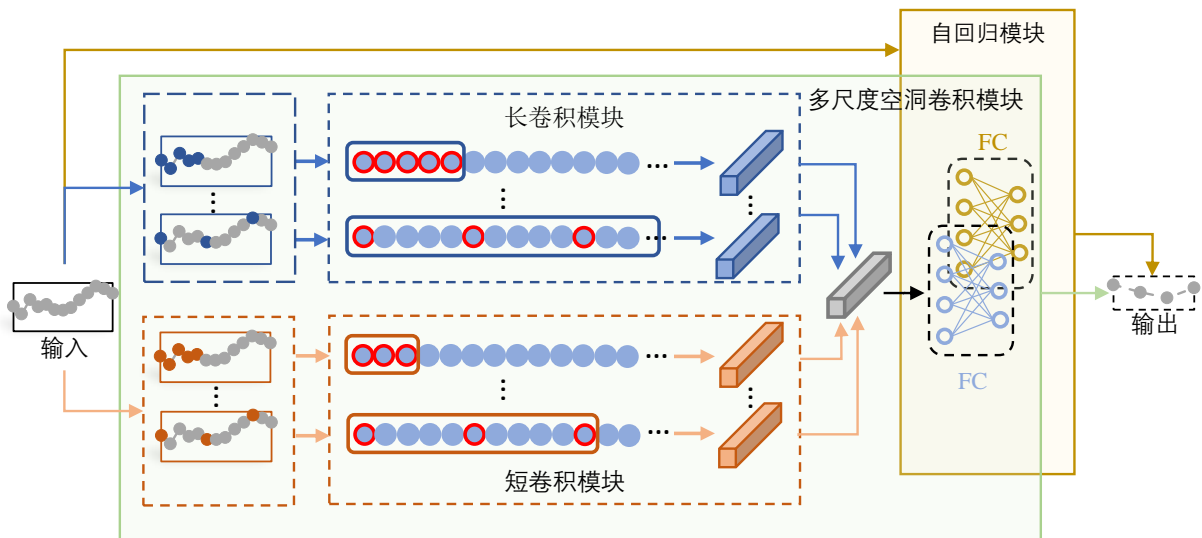


图 3-2 MSDCN 网络结构图

本章提出了一种浅层多尺度空洞卷积网络结构。如图 3-2 所示, 整体网络结构可以分为两个模块: 多尺度空洞卷积模块 (图中绿色框) 和自回归模块 (图中黄色框)。其中, 多尺度卷积模块负责从输入数据 \mathbf{X} 中提取特征, 通过一系列不同尺度的空洞卷积核实现, 以捕获时间序列数据中的多尺度信息。长卷积模块和短卷积模块均由多个卷积块构成, 每个卷积块专注于生成特定尺度的特征表示。这些不同尺度的特征表示随后被融合, 并通过前馈神经网络层进行进一步表征。

融合后的特征与自回归模块的输出相结合，形成最终的预测序列。为了增强模型的数值稳定性和训练过程的稳定性，本章提出的方法在将序列数据输入模型前和输出预测结果前引入了一种归一化操作（在图中未明确表示），此操作来源于 NLinear 模型^[41]。这种归一化操作有助于减少训练过程中的数值波动，从而提高模型的稳定性和预测准确性。接下来，本节将详细介绍网络结构中多尺度空洞卷积模块和自回归模块的设计原理和实现细节。

3.2.3 多尺度空洞卷积模块

本文使用两种具有相同结构但卷积核大小不同的卷积模块来对输入数据进行不同尺度的特征提取。长卷积模块采用较大的核尺寸，代表更大的感受野，而短卷积模块采用更小的核尺寸，代表较小的感受野。每个卷积模块由多个不同的一维卷积块并行排列而成。具体地，假设长卷积模块包含 n 个卷积块，短卷积模块具有 m 个卷积块。这些卷积块分别产生如下输出：

$$h_i = \text{ConvBlock}_i(\mathbf{X}), \quad i = 1, \dots, n, n+1, \dots, n+m, \quad (3-4)$$

其中， $h_i \in \mathbb{R}^{C \times T}$ 是第 i 个卷积块的输出。每一个卷积块是由空洞一维卷积 DilatedConv1d、批归一化 BatchNorm1d 和线性整流激活函数 ReLU 组成的序列结构，即

$$\begin{aligned} u_i &= \text{DilatedConv1d}(\text{Padding}(\mathbf{X})), \\ h_i &= \text{ReLU}(\text{BatchNorm1d}(u_i)). \end{aligned} \quad (3-5)$$

网络结构中的 1 维卷积采用深度卷积，其中输入通道和输出通道的数量等于数据集中变量的个数，即将每个变量进行独立建模。同时，为了捕获时间序列中的多尺度特征，本文采用了空洞卷积，而非普通卷积。空洞卷积通过在卷积核中引入间隔来实现，这种间隔，即空洞大小决定了卷积核观察时间序列的间隔大小，从而允许网络在不同的尺度上提取特征。为了捕捉更广泛的上下文信息，本文设计了 K 个并行的一维深度卷积层。在这一模块中，每个卷积层的空洞数按 2 的指数倍递增，即空洞数从 1 开始，逐层加倍：

$$\text{dilation factors} = \{1, 2^0 + 1, 2^1 + 1, \dots, 2^K + 1\}, \quad (3-6)$$

K 是一个可调整的参数。这样的设计能够在不增加额外参数的情况下，有效地扩大了每个卷积层的感受野，并且可以灵活地设计多种尺度的卷积层，使得网络能够在保持计算效率的同时，捕捉到时间序列数据中的长短距离依赖关系。

对于不同的卷积块，学习权重 $W \in \mathbb{R}^{C \times (n+m)}$ ，并利用该权重将不同卷积块得到的特征表示进行融合：

$$\begin{aligned} H &= [h_1, h_2, \dots, h_{n+m}], \\ M &= \sum_i^{n+m} H_i \odot W'_i, \end{aligned} \quad (3-7)$$

其中， $H \in \mathbb{R}^{C \times T \times (n+m)}$ 是卷积块特征表示 h_i 的横向拼接， W' 是 W 重复 T 次得到的权重， \odot 代表逐点乘积， $M \in \mathbb{R}^{C \times T}$ 是融合后的特征表示。该融合后的特征表示 M 随后被送入一个前馈神经网络层进行进一步地预测，生成预测分量：

$$\hat{Y}_c = W_1 M^T + b_1, \quad (3-8)$$

其中， $W_1 \in \mathbb{R}^{L \times T}$ 是权重， $b_1 \in \mathbb{R}^L$ 是偏置。

3.2.4 自回归模块

为了使神经网络更容易学习到时间序列中较为规律的简单模式，本文借鉴了 ResNet^[63] 的思想，引入了自回归预测模块。自回归方法在传统时间序列分析中是一种常用的随机过程建模技术，该方法基于变量自身的历史信息进行回归分析，从而构建模型以预测未来的数值。数学上，自回归模型可以表示为一个关于变量历史值的线性函数，其中当前时间步的值是前一时间步（或多个时间步）值的函数，数学公式可以表示为

$$Y_t = \sum_{i=1}^p \phi_i Y_{t-i} + \epsilon_t, \quad (3-9)$$

其中， ϕ 是系数， ϵ_t 是白噪声误差项。通过将自回归模块整合到神经网络中，模型可以捕捉到时间序列中潜在的线性动态依赖关系。本文使用线性层来表示自

回归模块，即预测分量：

$$\hat{Y}_h = W_2 \mathbf{X} + b_2, \quad (3-10)$$

其中， $W_2 \in \mathbb{R}^{L \times T}$ 是权重， $b_2 \in \mathbb{R}^L$ 是偏置。

最终预测值为两个预测分量的组合：

$$\hat{Y} = \hat{Y}_c + \hat{Y}_h, \quad (3-11)$$

其中， \hat{Y}_c 由公式 3-8 得到， \hat{Y}_h 由公式 3-10 得到。

3.2.5 损失函数

时间序列数据集中往往伴随着噪声的存在，这些噪声可能会诱导模型在训练过程中过分关注异常值，从而干扰模型对正常数据模式的学习。为了解决这一问题，本文引入了 Huber Loss 作为损失函数，以增强模型对异常值的鲁棒性。Huber Loss^[64]融合了均方误差 (MSE) 和平均绝对误差 (MAE) 两种损失函数的特点。具体而言，当预测误差小于 δ 时，损失函数采用均方误差函数；当预测误差大于 δ 时，损失函数采用平均绝对误差函数。这种损失函数的设计可以防止模型过度学习异常离群点，从而提高模型的鲁棒性。Huber Loss 的具体形式如下：

$$loss = \begin{cases} 0.5 (y_i - \hat{y}_i)^2, & \text{如果 } |y_i - \hat{y}_i| < \delta, \\ \delta * (|y_i - \hat{y}_i| - 0.5 * \delta), & \text{其他,} \end{cases} \quad (3-12)$$

其中， δ 为正数，是一个可调整的超参数。

3.3 实验与分析

3.3.1 实验设置

数据集

本文在长期时间序列预测问题的五类公共的基准数据集（共八个数据集）上进行了实验。以下是每个数据集的详细描述：

- (1) ETT 数据集^[17]包含了从 2 个电力变压器采集到的数据, 包括负载、油温等变量, 数据集的时间范围为 2016 年 7 月到 2018 年 7 月。根据采样粒度和采样地区的不同, ETT 数据集被划分为四个子数据集: ETTm1、ETTm2、ETTh1、ETTh2。其中 m 表示每十五分钟采样一次数据, h 表示每小时采样一次数据, 时间字母后面的 1 和 2 表示数据集来自中国的两个不同地区;
- (2) Illness 数据集^[65]包含了来自美国疾病控制与预防中心报告的流感疾病患者人数数据, 数据集的时间范围为 2002 年 1 月到 2020 年 6 月, 每周采样一次;
- (3) Weather 数据集^[66]包含来自马克斯·普朗克生物地球化学研究所气象站的 21 个气象指标, 数据集的时间范围为 2020 年 1 月到 2021 年 1 月, 每十分钟采样一次;
- (4) Electricity 数据集^[67]包含了从 321 个客户端采集到的用电量数据, 数据集的时间范围为 2012 年 1 月到 2014 年 12 月, 每小时采样一次;
- (5) Traffic 数据集^[68]包含了加利福尼亚州高速公路上 862 个传感器测量的道路占用率数据, 数据集的时间范围为 2016 年 7 月到 2018 年 7 月, 每小时采样一次。

更具体地, 每个数据集的变量数、采样频率以及时间步长见表 3-1。

表 3-1 长期时间序列预测实验数据集

数据集	ETTm1	ETTm2	ETTh1	ETTh2	Illness	Weather	Electricity	Traffic
变量数	7	7	7	7	7	21	321	862
采样频率	15 分钟	15 分钟	1 小时	1 小时	1 周	10 分钟	1 小时	1 小时
时间步长	69680	69680	17420	17420	966	52696	26304	17544

为了确保实验的公平性, 本文严格遵循了以往工作的数据集划分标准。具体地, 按照时间顺序将数据集分为训练集、验证集和测试集。对于 ETT 数据集, 采用 6: 2: 2 的划分比例; 其他数据集采用 7: 1: 2 的划分比例。对于 Illness 数据集, 模型的输入长度为 36, 输出长度为 24、36、48、60。而对于其他数据集, 模型的输入长度为 96, 输出长度为 96、192、336、720。对于不同的数据集, 这些输入和输出长度代表了不同的时间跨度, 具体的时间跨度可以结合输入/输出长度以及表 3-1 中的采样频率计算。

评估指标

实验采用的评估指标是均方误差函数 (Mean Squared Error, MSE) 和平均绝对误差 (Mean Absolute Error, MAE), 用于衡量模型的性能。MSE 计算了预测值和真值之间差异的平方, 而 MAE 计算了预测值与真实值之间差异的绝对值, 受异常值的影响较小。MSE 和 MAE 的计算公式如下:

$$MSE = \frac{1}{n} \sum_{j=1}^n (\hat{y}_j - y_j)^2, \quad (3-13)$$

$$MAE = \frac{1}{n} \sum_{j=1}^n |\hat{y}_j - y_j|, \quad (3-14)$$

其中, n 是变量数量, \hat{y} 是模型预测值, y 是真值。

实现细节

本章实验在 GPU 型号为 NVIDIA GeForce GTX 1080 Ti 上进行。Pytorch 框架使用 1.10.1 版本, 优化器使用 Adam^[69] 优化算法。Adam 优化算法结合了动量法和自适应学习率的特性, 是一种比较常用的优化算法。Adam 算法在训练过程中可以自适应地调整学习率, 具有较好的收敛性和泛化性能, 可以有效地优化神经网络模型。实验过程中, 学习率设为 1×10^{-4} , batch 大小设为 32。

基线方法

本章提出的方法与以下基线方法进行对比:

- (1) 基于 Transformer 方法的模型: ETSformer^[70]、Autoformer^[18]、Informer^[17]。这三种模型都对原始 Transformer 的注意力机制进行了改进或替换, 并且前两个模型在模型中加入了与时序特征有关的改进。ETSformer 模型则引入了指数平滑注意力和频率注意力, Autoformer 引入了序列自相关模块来替换注意力机制, 而 Informer 模型将注意力机制稀疏化;
- (2) 基于多层感知机的模型: DLinear^[41]、NLinear^[41]、LightTS^[45]。尽管这三种模型在结构设计上较为简洁, 但它们展现出了出色的预测性能, 并且在计

算复杂度和空间复杂度方面都很低；

- (3) 基于卷积神经网络的模型: TimesNet^[22]。与本文研究相似, 均采用了多尺度的卷积神经网络, TimesNet 基于傅里叶变换将一维时间序列转化为了不同频率的二维序列, 并且在其上应用 Inception 卷积网络结构。

3.3.2 对比实验

为了验证本章提出方法的有效性, 本章进行了两个主要实验: 多变量长时间序列预测和单变量长时间序列预测。实验结果如下:

多变量长序列预测

在 ETT 数据集上的实验结果展示在表 3-2, 其他数据集的实验结果见表 3-3。表中的“Avg”行表示了四种输出长度下的平均性能指标。在方法名称后标注星号(*)的指标结果是基于方法的官方代码运行得到的, 而其他方法的实验结果则来自 TimesNet 论文^[22]。表中最好的结果用加粗体表示, 第二好结果用下划线表示。观察表中数据可以看出, 在 ETT 数据集上, 本章提出的方法相比于其他

表 3-2 在 ETT 数据集上 MSDCN 实验结果

方法	MSDCN		TimesNet		NLinear*		DLinear		LightTS		ETSformer		Autoformer		Informer	
	-	-	2023	2023	2023	2023	2023	2023	2022	2022	2022	2022	2021	2021	2021	2021
评估指标	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.334 0.365	<u>0.338</u> 0.375	0.356 0.376	0.345 <u>0.372</u>	0.374 0.400	0.375 0.398	0.505 0.475	0.672 0.571							
	192	<u>0.380</u> 0.385	0.374 <u>0.387</u>	0.394 0.395	<u>0.380</u> 0.389	0.400 0.407	0.408 0.410	0.553 0.496	0.795 0.669							
	336	<u>0.411</u> 0.404	0.410 <u>0.411</u>	0.427 0.416	0.413 0.413	0.438 0.438	0.435 0.428	0.621 0.537	1.212 0.871							
	720	<u>0.477</u> 0.439	0.478 <u>0.450</u>	0.489 0.451	0.474 0.453	0.527 0.502	0.499 0.462	0.671 0.561	1.166 0.823							
	Avg	0.400 0.398	<u>0.400</u> <u>0.406</u>	0.416 0.409	0.403 0.407	0.435 0.437	0.429 0.425	0.588 0.517	0.961 0.734							
ETTh2	96	0.174 0.256	0.187 0.267	<u>0.182</u> <u>0.265</u>	0.193 0.292	0.209 0.308	0.189 0.280	0.255 0.339	0.365 0.453							
	192	0.239 0.298	0.249 0.309	<u>0.246</u> <u>0.304</u>	0.284 0.362	0.311 0.382	0.253 0.319	0.281 0.340	0.533 0.563							
	336	0.296 0.332	0.321 0.351	<u>0.306</u> <u>0.341</u>	0.369 0.427	0.442 0.466	0.314 0.357	0.339 0.372	1.363 0.887							
	720	0.395 0.392	0.408 0.403	<u>0.408</u> <u>0.398</u>	0.554 0.522	0.675 0.587	0.414 0.413	0.433 0.432	3.379 1.338							
	Avg	0.276 0.320	0.291 0.333	<u>0.286</u> <u>0.327</u>	0.350 0.401	0.409 0.436	0.293 0.342	0.327 0.371	1.410 0.810							
ETTh1	96	0.379 0.390	<u>0.384</u> 0.402	0.393 0.400	0.386 <u>0.400</u>	0.424 0.432	0.494 0.479	0.449 0.459	0.865 0.713							
	192	0.428 0.417	<u>0.436</u> 0.429	0.449 0.433	0.437 0.432	0.475 0.462	0.538 0.504	0.500 0.482	1.008 0.792							
	336	0.465 0.436	0.491 0.469	0.485 0.449	<u>0.481</u> 0.459	0.518 0.488	0.574 0.521	0.521 0.496	1.107 0.809							
	720	0.468 0.453	0.521 0.500	<u>0.471</u> <u>0.462</u>	0.519 0.516	0.547 0.533	0.562 0.535	0.514 0.512	1.181 0.865							
	Avg	0.435 0.424	0.458 0.450	<u>0.449</u> <u>0.436</u>	0.456 0.452	0.491 0.479	0.542 0.510	0.496 0.487	1.040 0.795							
ETTh2	96	0.286 0.333	0.340 0.374	<u>0.289</u> <u>0.338</u>	0.333 0.387	0.397 0.437	0.340 0.391	0.346 0.388	3.755 1.525							
	192	0.364 0.385	0.402 0.414	<u>0.375</u> <u>0.390</u>	0.477 0.476	0.520 0.504	0.430 0.439	0.456 0.452	5.602 1.931							
	336	0.407 0.420	0.452 0.452	<u>0.414</u> <u>0.426</u>	0.594 0.541	0.626 0.559	0.485 0.479	0.482 0.486	4.721 1.835							
	720	0.418 0.438	0.462 0.468	<u>0.423</u> <u>0.444</u>	0.831 0.657	0.863 0.672	0.500 0.497	0.515 0.511	3.647 1.625							
	Avg	0.369 0.394	0.414 0.427	<u>0.375</u> <u>0.400</u>	0.559 0.515	0.602 0.543	0.439 0.452	0.450 0.459	4.431 1.729							

表 3-3 在其他数据集上 MSDCN 实验结果

方法	MSDCN		TimesNet		NLinear*		DLinear		LightTS		ETSformer		Autoformer		Informer		
	-		2023		2023		2023		2022		2022		2021		2021		
评估指标	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
Electricity	96	<u>0.175</u>	0.265	0.168	<u>0.272</u>	0.198	0.275	0.197	0.282	0.207	0.307	0.187	0.304	0.201	0.317	0.274	0.368
	192	0.183	0.271	<u>0.184</u>	0.289	0.198	<u>0.278</u>	0.196	0.285	0.213	0.316	0.199	0.315	0.222	0.334	0.296	0.386
	336	<u>0.199</u>	0.287	0.198	0.300	0.212	<u>0.293</u>	0.209	0.301	0.230	0.333	0.212	0.329	0.231	0.338	0.300	0.394
	720	0.238	0.320	0.220	<u>0.320</u>	0.254	0.326	0.245	0.333	0.265	0.360	<u>0.233</u>	0.345	0.254	0.361	0.373	0.439
	Avg	<u>0.199</u>	0.286	0.192	0.295	0.216	<u>0.293</u>	0.212	0.300	0.229	0.329	0.208	0.323	0.227	0.338	0.311	0.397
Traffic	96	0.619	<u>0.366</u>	0.593	0.321	0.647	0.388	0.650	0.396	<u>0.615</u>	0.391	0.607	0.392	0.613	0.388	0.719	0.391
	192	0.577	<u>0.342</u>	0.617	0.336	0.600	0.364	<u>0.598</u>	0.370	0.601	0.382	0.621	0.399	0.616	0.382	0.696	0.379
	336	0.591	0.348	0.629	0.336	0.607	0.367	<u>0.605</u>	0.373	0.613	0.386	0.622	0.396	0.622	<u>0.337</u>	0.777	0.420
	720	0.630	<u>0.365</u>	0.640	0.350	0.645	0.387	0.645	0.394	0.658	0.407	<u>0.632</u>	0.396	0.660	0.408	0.864	0.472
	Avg	0.604	<u>0.355</u>	<u>0.620</u>	0.336	0.625	0.376	0.625	0.383	0.622	0.392	0.621	0.396	0.628	0.379	0.764	0.416
Weather	96	0.169	0.217	<u>0.172</u>	<u>0.220</u>	0.202	0.240	0.196	0.255	0.182	0.242	0.197	0.281	0.266	0.336	0.300	0.384
	192	0.215	0.259	<u>0.219</u>	<u>0.261</u>	0.248	0.277	0.237	0.296	0.227	0.287	0.237	0.312	0.307	0.367	0.598	0.544
	336	0.269	0.299	<u>0.280</u>	<u>0.306</u>	0.300	0.313	0.283	0.335	0.282	0.334	0.298	0.353	0.359	0.395	0.578	0.523
	720	<u>0.350</u>	0.352	<u>0.365</u>	<u>0.359</u>	0.373	0.360	0.345	0.381	0.352	0.386	0.352	0.388	0.419	0.428	1.059	0.741
	Avg	0.251	0.282	<u>0.259</u>	<u>0.287</u>	0.281	0.298	0.265	0.317	0.261	0.312	0.271	0.334	0.338	0.382	0.634	0.548
ILI	24	2.222	<u>0.936</u>	<u>2.317</u>	0.934	2.662	1.054	2.398	1.040	8.313	2.144	2.527	1.020	3.483	1.287	5.764	1.677
	36	<u>2.192</u>	0.915	1.972	<u>0.920</u>	2.487	1.040	2.646	1.088	6.631	1.902	2.615	1.007	3.103	1.148	4.755	1.467
	48	2.164	0.938	<u>2.238</u>	<u>0.940</u>	2.406	1.024	2.614	1.086	7.299	1.982	2.359	0.972	2.669	1.085	4.763	1.469
	60	<u>2.287</u>	<u>0.946</u>	2.027	0.928	2.475	1.037	2.804	1.146	7.283	1.985	2.487	1.016	2.770	1.125	5.264	1.564
	Avg	<u>2.216</u>	<u>0.934</u>	2.139	0.931	2.507	1.039	2.616	1.090	7.382	2.003	2.497	1.004	3.006	1.161	5.137	1.544

模型有了显著的提升。与 ETT 数据集相比，其他数据集具有更多的变量（比如 Traffic 数据集）和更复杂的数据波动性（比如 Illness 数据集）。在这些数据集上，本章提出的方法提升略小。详细的结果分析如下：

通过对表 3-2 和表 3-3 的综合分析，可以观察到，本章提出的方法在 80% 的实验场景中均取得了最优的预测结果。这表明，该方法在不同应用领域的时间序列数据集上均展现出了预测性能的提升。与目前表现最佳的基于卷积神经网络的模型 TimesNet 相比，本章提出的方法在均方误差（MSE）指标上实现了 2.4% 的降低，在平均绝对误差（MAE）指标上实现了 2.2% 的降低。这一改进验证了本章设计的浅层多尺度一维空洞卷积模块相较于 TimesNet 中使用的深层二维普通卷积模块，在预测性能上具有优势。因此，本章提出的方法通过空洞卷积对时间序列数据进行多尺度下采样，有效地捕获了数据的多尺度特征。此外，本章的方法与 TimesNet 相比，计算复杂度和空间复杂度都具有很大的优势，这将在第 3.3.4 小节模型分析部分进一步分析。

同时，与基于 Transformer 的模型 ETSformer、Autoformer、Informer 相比，MSDCN 模型也具有优势。尤其是，当与最佳的基于 Transformer 的模型 ETSformer 进行比较时，本章提出的方法在 MSE 指标上取得了 9.4% 的降低，在 MAE 指标

上取得了 11.0% 的降低。这一显著的性能提升表明，即使不依赖于复杂的 Transformer 架构，本章提出的方法也能够有效地处理长时间序列预测任务。

进一步地，与表现最佳的基于多层感知机的模型 NLinear 相比，本章提出的方法在 MSE 指标上实现了 9.4% 的降低，在 MAE 指标上实现了 4.2% 的降低。NLinear 模型采用了自回归结构，并对输入进行了前后“归一化”处理。本章提出的方法可以被视为对 NLinear 模型的扩展和改进，结合了 NLinear 模型和新设计的多尺度卷积模块。实验结果表明，所设计的卷积模块显著增强了 NLinear 模型的表达能力和预测性能。

单变量长序列预测

在本研究中，设计的浅层多尺度卷积模块基于变量独立性原则，通过通道独立性机制对每个变量进行独立建模。然而，在前馈神经网络层中，不同变量的建模过程共享了相同的模型参数。为了验证本模型在单变量预测任务中的有效性，本研究选取了 ETT 数据集中的油温数据列作为实验的单变量预测目标。实验结果汇总于表 3-4。在实验设置中，本文遵循了与以往研究一致的回看窗口长度，即设定为 336。

表 3-4 单变量时序预测 MSDCN 实验结果

方法	MSDCN		PatchTST/64		PatchTST/42		TimesNet		NLinear		DLinear		Autoformer		Informer		
	-	-	2023		2023		2023		2023		2023		2021		2021		
评估指标	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh1	96	0.052	0.177	0.059	0.189	0.055	0.179	0.056	0.183	<u>0.053</u>	<u>0.177</u>	0.056	0.180	0.071	0.206	0.193	0.377
	192	0.067	0.202	0.074	0.215	0.071	0.205	0.072	0.208	<u>0.069</u>	<u>0.204</u>	0.071	0.204	0.114	0.262	0.217	0.395
	336	<u>0.078</u>	<u>0.225</u>	0.076	0.220	0.081	0.225	0.083	0.225	0.081	0.226	0.098	0.244	0.107	0.258	0.202	0.381
	720	0.077	0.220	0.087	0.236	0.087	0.232	0.089	0.236	<u>0.080</u>	<u>0.226</u>	0.189	0.359	0.126	0.283	0.183	0.355
	Avg	0.068	0.206	0.074	0.215	0.074	0.210	0.075	0.213	<u>0.071</u>	<u>0.208</u>	0.104	0.247	0.105	0.252	0.199	0.377
ETTh2	96	0.118	0.269	0.131	0.284	0.129	0.282	0.136	0.286	<u>0.129</u>	<u>0.278</u>	0.131	0.279	0.153	0.306	0.213	0.373
	192	0.155	0.312	0.171	0.329	<u>0.168</u>	0.328	0.182	0.336	0.169	<u>0.324</u>	0.176	0.329	0.204	0.351	0.227	0.387
	336	<u>0.174</u>	0.335	0.171	<u>0.336</u>	0.185	0.351	0.216	0.369	0.194	0.355	0.209	0.367	0.246	0.389	0.242	0.401
	720	0.196	0.355	<u>0.223</u>	<u>0.380</u>	0.224	0.383	0.245	0.396	0.225	0.381	0.276	0.426	0.268	0.409	0.291	0.439
	Avg	0.161	0.318	<u>0.174</u>	<u>0.332</u>	0.177	0.336	0.195	0.347	0.179	0.335	0.198	0.350	0.218	0.364	0.243	0.400
ETTh1	96	0.026	<u>0.122</u>	<u>0.026</u>	0.123	0.026	0.121	0.029	0.127	0.026	0.122	0.028	0.123	0.056	0.183	0.109	0.277
	192	<u>0.040</u>	0.151	0.040	0.151	0.039	<u>0.150</u>	0.046	0.162	0.039	0.149	0.045	0.156	0.081	0.216	0.151	0.310
	336	0.052	0.175	0.053	0.174	0.053	<u>0.173</u>	0.060	0.188	0.052	0.172	0.061	0.182	0.076	0.218	0.427	0.591
	720	0.072	0.208	<u>0.073</u>	0.206	0.074	<u>0.207</u>	0.081	0.219	0.073	0.207	0.080	0.210	0.110	0.267	0.438	0.586
	Avg	0.047	0.164	<u>0.048</u>	0.164	0.048	0.163	0.054	0.174	0.048	<u>0.163</u>	0.054	0.168	0.081	0.221	0.281	0.441
ETTh2	96	0.062	0.182	0.065	0.187	0.065	0.186	0.066	0.186	<u>0.063</u>	<u>0.182</u>	0.063	0.183	0.065	0.189	0.088	0.225
	192	<u>0.091</u>	<u>0.226</u>	0.093	0.231	0.094	0.231	0.102	0.240	0.090	0.223	0.092	0.227	0.118	0.256	0.132	0.283
	336	0.091	0.226	0.121	0.266	0.120	0.265	0.132	0.277	<u>0.117</u>	<u>0.259</u>	0.119	0.261	0.154	0.305	0.180	0.336
	720	0.166	<u>0.319</u>	0.172	0.322	0.171	0.322	0.185	0.336	<u>0.170</u>	0.318	0.175	0.320	0.182	0.335	0.300	0.435
	Avg	0.103	0.238	0.113	0.252	0.113	0.251	0.121	0.260	<u>0.110</u>	<u>0.246</u>	0.112	0.248	0.130	0.271	0.175	0.320

实验结果显示,本章提出的方法 MSDCN 取得了显著的性能优势:与 NLinear 模型相比,本章提出的方法在均方误差 (MSE) 指标上实现了 4.8% 的降低;与 PatchTST/64 模型相比,降低了 6.8%;与 TimesNet 模型相比,更是显著降低了 12.9%。这些结果一致表明,本章提出的方法在处理单变量长序列预测问题时,仍然能够实现卓越的性能。

3.3.3 消融实验

多尺度卷积模块

为了验证本文设计的卷积模块的有效性,本文设计了四个实验,第一个实验只保留自回归模块,以评估模型在没有多尺度卷积模块辅助下的基本预测能力。第二个实验在自回归模块的基础上仅加入长卷积模块,第三个实验在自回归模块的基础上仅加入短卷积模块,以评估不同尺度特征的贡献。第四个实验在自回归模块的基础上同时加入长卷积模块和短卷积模块,实验结果如表 3-5 所示。

表 3-5 不同尺度卷积模块对预测结果的影响

卷积模块		评估指标	ETTM2					Electricity				
长卷积模块	短卷积模块		96	192	336	720	Avg	96	192	336	720	Avg
×	×	MSE	0.182	0.246	0.306	0.408	0.286	0.198	0.198	0.212	0.254	0.216
		MAE	0.265	0.304	0.341	0.398	0.327	0.275	0.278	0.293	0.326	0.293
✓	×	MSE	0.176	0.239	0.300	0.398	0.278	0.176	0.184	0.200	0.241	0.200
		MAE	0.257	0.299	0.335	0.392	0.321	0.265	0.272	0.287	0.321	0.286
×	✓	MSE	0.177	0.242	0.302	0.407	0.282	0.194	0.193	0.208	0.250	0.211
		MAE	0.256	0.300	0.336	0.399	0.323	0.271	0.274	0.289	0.322	0.289
✓	✓	MSE	0.174	0.239	0.296	0.395	0.276	0.175	0.183	0.199	0.239	0.199
		MAE	0.256	0.298	0.332	0.392	0.320	0.265	0.271	0.287	0.320	0.286

从表中可以看出,长卷积模块和短卷积模块的分别引入都可以改善预测结果。对于采样粒度为 15 分钟的 ETTm2 数据集,加入长卷积模块和短卷积模块的预测结果差别并不明显,这是因为 ETTm2 数据集受到短期因素的影响和受到长期因素的影响较为均衡。而对于采样粒度为 1 小时的 Electricity 数据集,仅加入长卷积模块比仅加入短卷积模块的预测效果显著提升,这是因为 Electricity 数据集受到长期因素的影响较大。总体而言,在同时引入两个模块后,预测效果达到最佳水平。这说明,多尺度的联合建模对于提取多尺度的时间依赖信息是非

常重要的，同时也进一步验证了本章提出的方法的有效性。

通道独立性

在实际应用中，对于不具有显式空间依赖关系的变量，时间序列的多个变量之间通常存在隐式的相互依赖关系，其中一个变量的变化可能会随着另一个变量的变化而发生。然而，时间序列数据具有分布漂移的特性，测试数据集与训练数据集、验证数据集的数据分布可能存在较大差异。在长时间序列预测问题中，若考虑多个变量之间的隐性依赖关系，对多变量进行融合建模时，有时会忽视原始变量在很长一段时期内分布的变化。最近研究^[71]发现在长时间序列预测问题中，相比于变量联合预测，独立预测能够取得更好的预测结果，并且他们通过严谨的理论论证表明独立预测有助于缓解分布漂移现象。比如，基于 Transformer 的模型 PatchTST^[38]就采用了独立建模的方式，也取得了不错的效果。本文设计的方法也采用独立建模的方式。

本文采用的卷积网络结构为分通道卷积，即变量之间没有相互交互。为了探究本文所设计的卷积网络结构中通道依赖性对预测结果的影响，本文进行了一组实验。在预测长度为 720 的实验设置下，通过改变卷积中通道的分组个数来确定不同变量间是否交互，实验结果如表 3-6 所示。其中， $groups = 1$ 表示所有变量共享一个卷积核，即所有变量之间是有交互的， $groups = \text{变量数}$ 表示每一个变量对应一个独立的卷积核，即每一个卷积核只处理一个变量的信息。实验记录指标包含了均方误差 MSE、平均绝对误差 MAE、推理时间和参数量。

通过比较实验结果，可以观察到在本文设计的网络结构中，分组卷积具有以下优势：首先，分组卷积减少了标准卷积的通道冗余，从而降低了网络训练参数

表 3-6 不同通道分组数对预测结果的影响

分组数	评估指标	ETTm2	ETTh1	Electricity	Traffic
1	MSE	0.425	0.502	0.233	0.633
	MAE	0.414	0.482	0.330	0.374
	推理时间	25.90s	2.80s	10.14s	4.92s
	参数量	569.36KB	556.78KB	45.37MB	284.02MB
变量数	MSE	0.395	0.468	0.238	0.630
	MAE	0.392	0.453	0.320	0.365
	推理时间	20.98s	2.51s	9.02s	4.54s
	参数量	549.51KB	547.43KB	711.14KB	230.69KB

量,有助于避免过拟合问题。对于单个卷积,标准卷积的参数量为 $k \times c_{in} \times c_{out}$,而分组的参数量为 $k \times \frac{c_{in}}{c_{in}} \times c_{out}$ 。其次,由于参数量的减少,分组卷积可以降低模型在训练和推理阶段的计算开销,从而提升推理速度。此外,分组卷积有利于模型学习每个变量的独特特征,解决单个变量在输入和输出间可能存在的分布漂移现象而难以预测的问题,进而能够获得更精准的预测结果。同时,通过比较不同数据集在不同分组数设置下的提升幅度,可以发现 Electricity 和 Traffic 数据集的提升幅度较小。这是因为相较于 ETTm2 和 ETTh1 数据集,Electricity 和 Traffic 数据集的数据序列更为平稳,分布漂移现象相对不明显,因此在这两个数据集上,不同的分组数设置对预测结果的影响较小。

自回归模块

为了验证自回归模块对本文提出的网络结构 MSDCN 的作用,本文设计了三个实验:第一个实验是去除网络中的卷积模块,即只保留自回归模块;第二个实验是去除网络中的自回归模块,即只保留卷积模块;第三个实验是完整的 MSDCN 网络,包含了卷积模块和自回归模块。实验结果如图 3-3 所示。

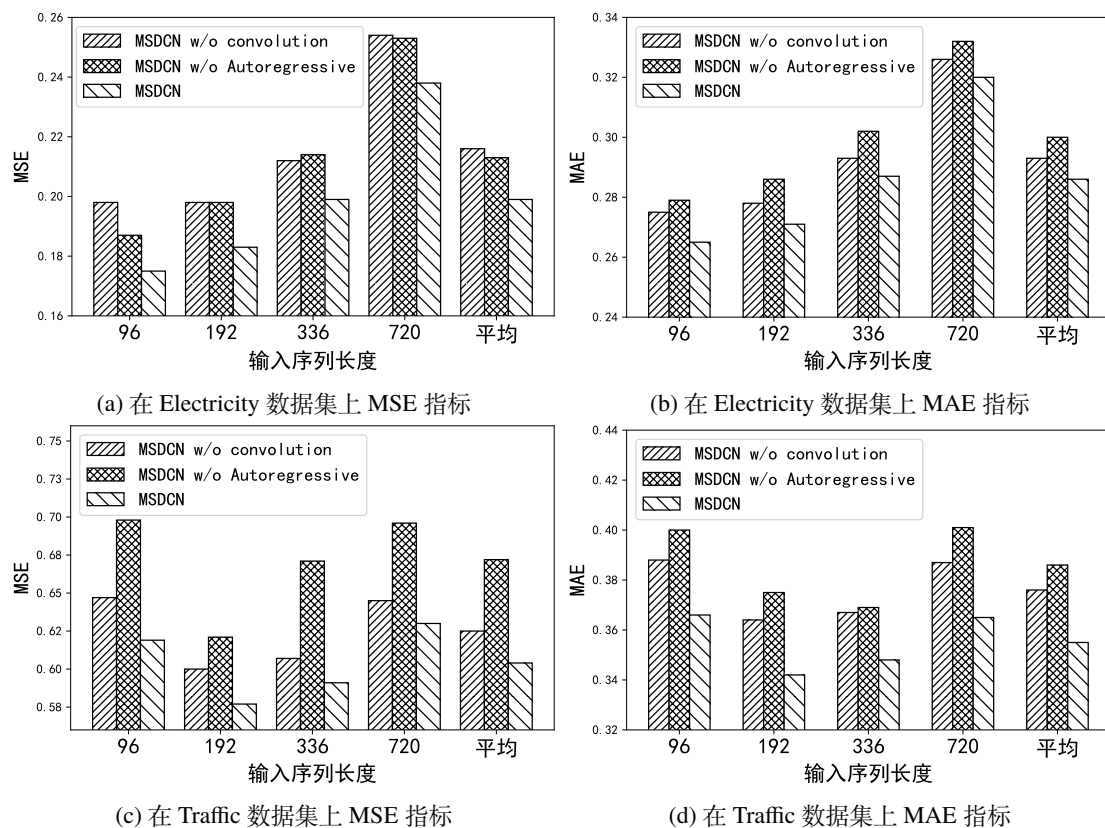


图 3-3 自回归模块对预测结果的影响

从实验结果可以观察到，只用自回归模块进行预测的表现并不差，在加入卷积模块后，网络性能有所提升。在 *Electricity* 数据集上获得了 7% 的性能提升，在 *Traffic* 数据集上获得了 3% 的性能提升。这说明对于具有一定周期模式的时间序列，自回归模块能够捕捉到周期性的特征，并能进行较准确的预测，表现并不差。尽管如此，由于自回归模块本质上是一层线性变换，它在处理时间序列中更为复杂的模式时存在局限性。本章设计的多尺度卷积模块旨在解决这一问题。通过多尺度卷积操作，该模块能够识别并利用时序数据中的复杂模式和趋势，从而显著提高预测的准确性。实验结果验证了当加入本文设计的卷积模块后，能够增强模型性能。

3.3.4 模型分析

多尺度表征

为了验证多尺度卷积网络是否有效地学习了多尺度特征表示，本研究对卷积层的输出进行了可视化分析，结果展示于图 3-4。在本次实验中，选取了长度为 96 的时间序列作为输入，其中时间步长被映射到横轴，而卷积层的输出值则对应纵轴。图中展示了多个子图，每个子图对应不同尺度的卷积层输出，按照从大到小的尺度顺序自上而下排列。考虑到卷积模块中使用了 **ReLU** 激活函数，所有负值输出均被置零，而未标记 ID 的区域则意味着相应的卷积层未能激活，即

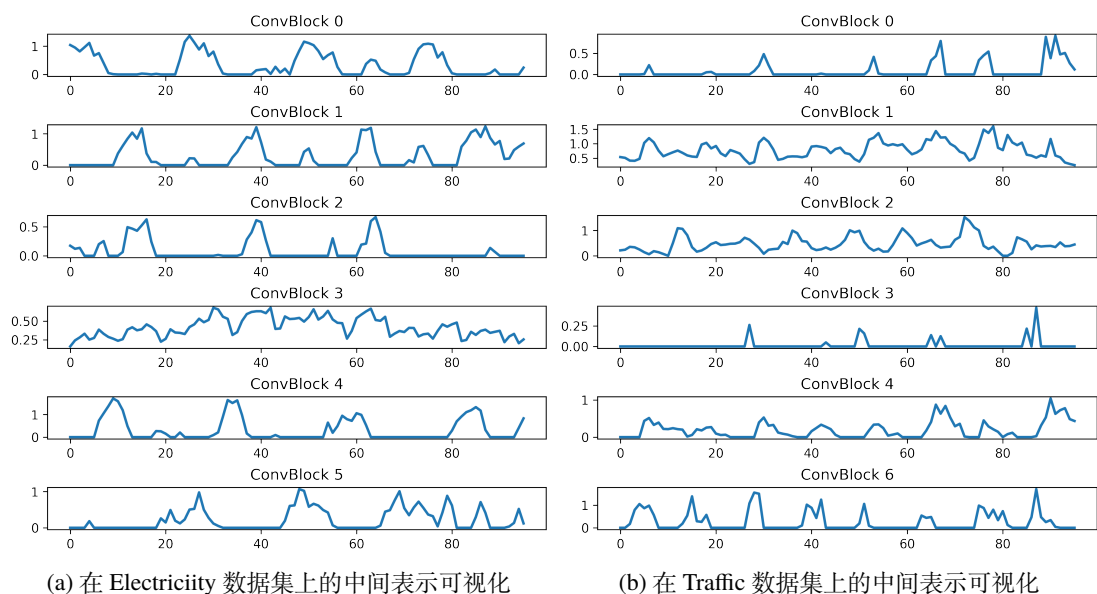


图 3-4 不同卷积块输出表示的可视化

其输出被完全抑制。

在本文设计的多尺度模块中，这些 ConvBlock 的输入都是一样的，通过分析图 3-4，可以观察到不同尺度的卷积层能够提取出形态各异特征曲线，这些抽象的曲线本质上反映了具有不同周期性和趋势性的时间序列子模式。这一发现验证了本章提出的方法能够通过多尺度卷积层捕捉时间序列数据中的多样化特征，进而更精准地建模数据的周期性和趋势性。因此，本章提出的方法在处理时间序列数据时展现了有效性和鲁棒性，能够从不同尺度上提取特征，并捕捉到数据中的多种周期和趋势子模式。

不同输入长度下的预测性能比较

在固定输出序列长度的前提下，增加输入序列的长度使得模型能够接触到更丰富的历史信息，这有助于模型更深入地理解时间序列数据中的长期和短期

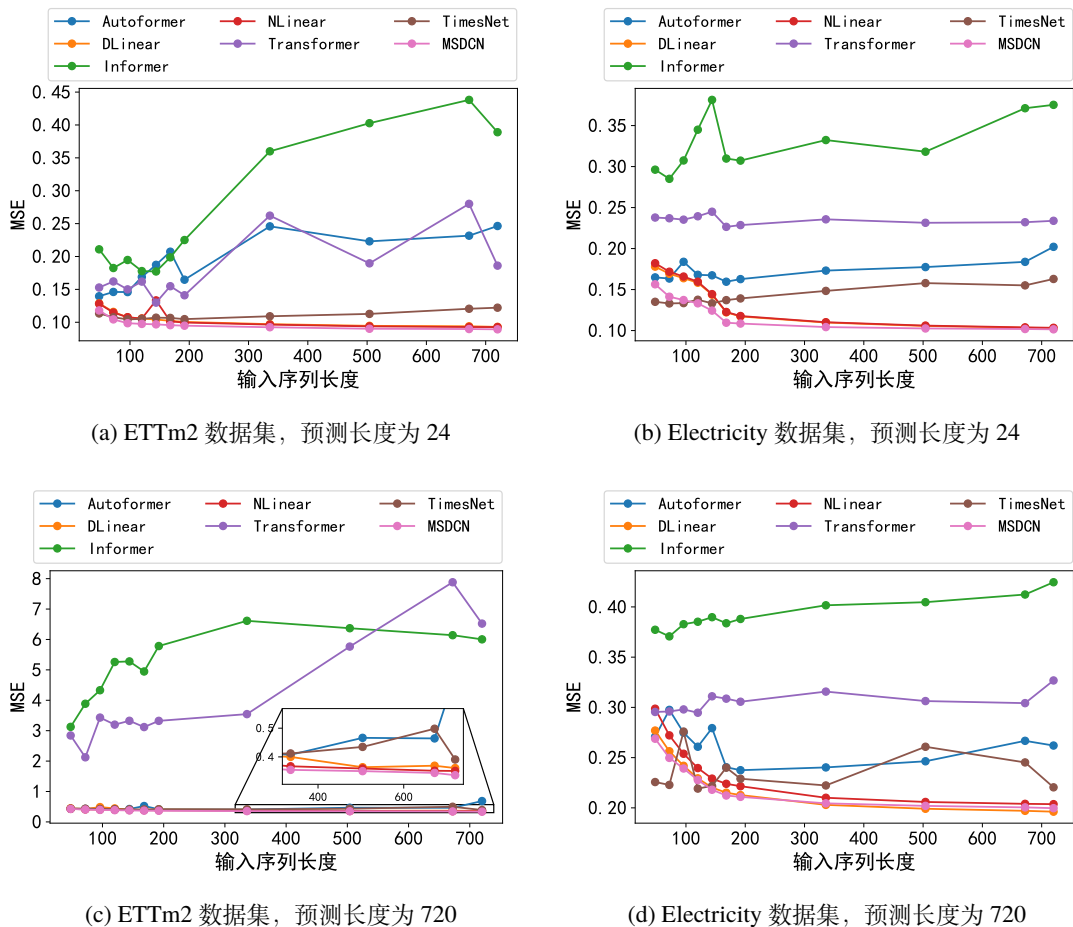


图 3-5 不同输入长度下的预测性能比较

依赖性以及趋势变化，从而提升对未来趋势的预测精度。然而，随着输入序列长度的增加，模型所面临的建模任务也变得更加复杂。在较长的输入序列中，可能混杂着噪声和冗余信息，这对模型的学习能力提出了更高的要求。理论上，一个优秀的模型应当能够在输出序列长度固定的情况下，随着输入序列长度的增加而提升其预测性能，或者至少维持性能不下降。

为了评估本章提出的模型与其他最优模型在不同输入长度下的性能差异，本文在 ETTm2 数据集和 Electricity 数据集上进行了四组实验，这些实验的预测长度固定为 24 或 720，输入序列长度分别为 48、72、96、120、144、168、192、336、504、672、720，实验结果如图 3-5 所示。图中水平轴表示输入序列长度，垂直轴表示均方误差。为了更清楚地展示差异，部分图进行了放大处理。

从图中可以看出，随着输入序列长度的增加，Informer、Autoformer、TimesNet 等模型的预测 MSE 呈现出上升趋势，表明它们的预测性能随着序列长度的增加而逐渐下降。这可能是因为这些模型在处理更长的输入序列时，开始捕捉到冗余信息，造成了过拟合，从而影响了预测的准确性。相比之下，DLinear 和 NLinear 模型在大多数情况下展现出相似的性能，并且随着输入序列长度的增加，它们的预测性能有所提升。本文提出的 MSDCN 模型不仅在输入序列长度较短时优于其他模型，并且即使在较长的输入序列下也一样能保持其性能，这表明本章提出的方法能有效利用输入序列中的有效信息，从中提取有用的特征进行准确的预测。

效率分析

在实时时间序列预测的应用场景中，模型的性能不仅体现在预测准确性上，还包括其处理速度和资源消耗。一个高效的预测模型能够迅速响应输入数据，提供及时且准确的预测结果，这对于相关人员制定快速决策、规避潜在风险至关重要。

为了评估所提出的 MSDCN 模型与主流模型在时间和空间效率方面的性能差异，本研究在 Electricity 数据集上进行了一系列的实验比较，如表 3-7 所示。实验中，本文将 MSDCN 模型及对比的主流模型分别运行了 5 次，以计算各自的平均效率指标。这些指标包括参数量、内存占用、乘累加操作（MACs）数量以及模型的推理时间。参数量是衡量模型复杂度的一个重要指标，它反映了模型

表 3-7 MSDCN 与主流模型的时间效率和空间效率对比

模型	参数量	内存	MACs	推理时间
MSDCN	239.67K	19.16MB	8.75M	3.68s
DLinear	139.68K	13.48MB	0.14M	1.50s
Autoformer	14.91M	126.33MB	4.18G	62.39s
PatchTST	7.59M	800.96MB	12.69G	80.42s
TimesNet	301.75M	1628.31MB	1.23T	1491.58s

中可训练参数的总数。内存占用指标则指示了模型在一次前向或后向传递过程中所需的存储空间大小。MACs 是评估计算量的关键指标，它表示模型在执行过程中所进行的乘法和累加操作的总次数。推理时间则是衡量模型实际应用效率的直接指标，即模型处理输入数据并输出预测结果所需的时间。

从表中可以看出，DLinear 模型由于其结构中仅包含少量的池化层和线性层，因此具有最少的参数量，导致其推理速度最快。而本章所提出的 MSDCN 模型在推理速度上紧随其后，且相较于基于 Transformer 架构的模型显示出更快的效率。更重要的是，与同样基于卷积神经网络的 TimesNet 模型相比，MSDCN 模型在推理速度上有显著提升。TimesNet 模型首先通过嵌入层对数据进行转换，并使用参数高效的 Inception 结构来获取中间表示。这些操作会导致网络参数量和计算复杂度显著增加。相比之下，MSDCN 模型使用浅层的卷积层来提取特征表示，因此模型参数量和计算量更少，从而推理速度更快。这些结果表明，在长期时间序列预测任务中，本文提出的模型在推理速度方面表现出明显的优势，能够在实时预测及决策应用中发挥作用。

预测结果可视化

图 3-6 展示了 MSDCN 方法在 Electricity 数据集和 Traffic 数据集上的预测结果。在图中，纵轴代表变量的数值，而横轴代表序列的长度，其中横轴的 0 到 95 区间对应于输入序列，96 到 432 区间对应于预测输出序列。蓝线代表实际观测值，而橘线代表 MSDCN 模型的预测值。

通过对比实际值和预测值的曲线，可以观察到 MSDCN 方法在两个数据集上的预测结果均与实际值的趋势和周期性保持了较高的一致性。然而，图中也揭示了 MSDCN 方法在处理突发变化时的局限性，尤其是在图 3-6(a) 中所示的第

三个峰值。这类突发变化往往由多种复杂因素共同作用引起，模型在缺乏足够上下文信息的情况下，可能难以准确预测这些变化。这表明，尽管 MSDCN 在捕捉时间序列的整体趋势和周期性方面表现出色，但在处理由复杂因素引起的局部突发变化时，仍需进一步的改进和优化。

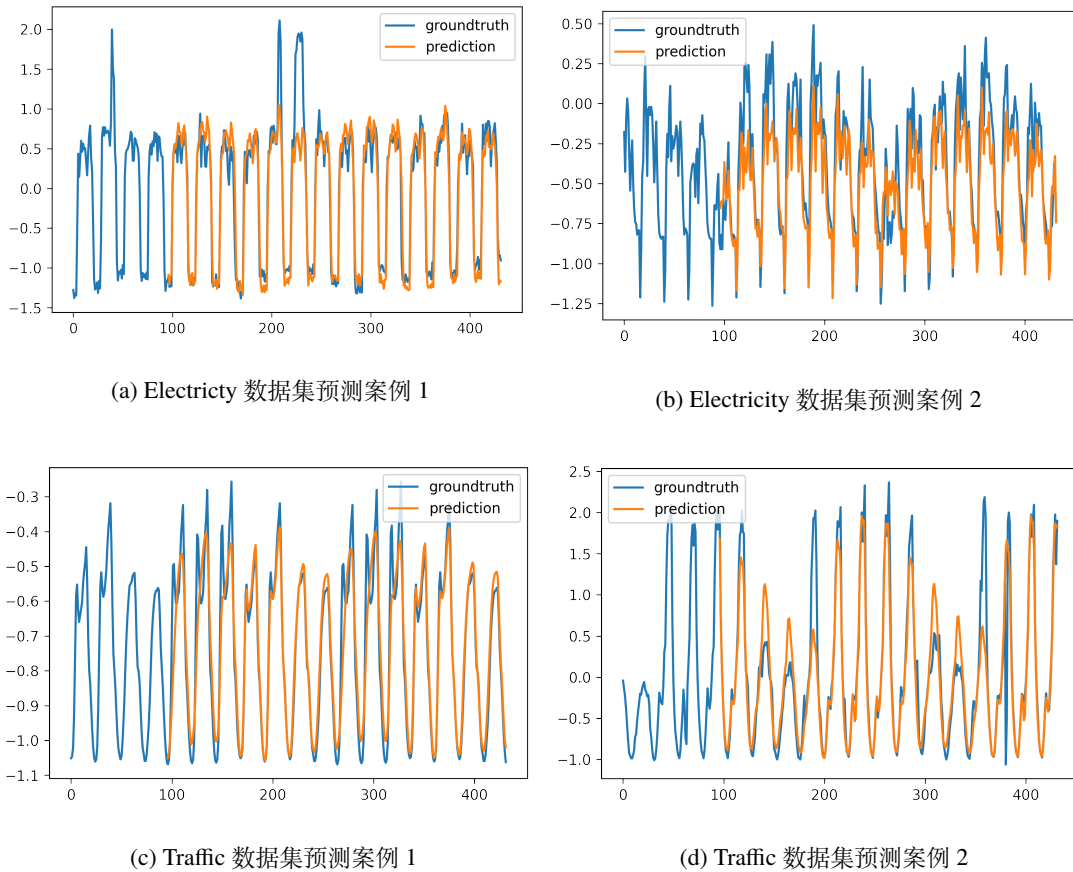


图 3-6 预测结果可视化

3.4 本章小结

本章介绍了一种基于浅层空洞卷积的长序列预测算法 MSDCN。该算法由多尺度空洞卷积模块和自回归模块组成。卷积模块由多个不同尺度的空洞卷积构成，这些空洞卷积层通过在不增加参数量和计算量的前提下，扩大了模型的感受野，使得模型能够同时捕捉到序列中局部和全局的时序依赖关系。这种设计避免了深层网络结构可能带来的梯度消失问题，同时保持了模型的轻量化和计算效率；自回归模块则用于辅助卷积模块捕捉线性动态关系，以提高预测准确性。

为了全面评估 MSDCN 算法的性能，本文设计了一系列的消融实验和对比实验。这些实验不仅验证了 MSDCN 在不同输入长度下的预测准确性，而且还深入分析了各个模块在模型中的作用和贡献。此外，本文还对 MSDCN 的时间效率和空间效率进行了评估，以展示其在资源利用方面的优势。实验结果表明，MSDCN 在长期时间序列预测任务上表现出色，与当前主流的预测模型相比，无论是在预测精度、时间效率还是空间效率方面，都展现了显著的优势。这些结果证实了 MSDCN 算法的实用性和适用性，表明其能够广泛应用于不同领域的时间序列预测问题，并提供快速而准确的预测结果。

第四章 基于稀疏动态图的短期时空预测算法

时间序列数据中，存在一类数据，其特性不仅随时间变化，而且在空间分布上表现出显著的相关性。这类数据的空间关系为预测提供了额外的维度。交通网络数据是日常生活中典型的时空数据类型，本章将以交通数据为主要研究对象，研究短期时空预测算法。本章提出一种受热力学启发的基于稀疏动态空间图的注意力网络模型（Thermodynamic-inspired Dynamic Graph Sparse Attention Network, ThermoNet），它基于稀疏空间图结构，并受到热力学原理的启发。通过对比实验和消融实验，本文验证了所提出方法的有效性。

4.1 研究动机

表 4-1 常用时空数据集预定义图稀疏度

数据集	节点数	稀疏度
METR-LA	207	0.035
PEMS-BAY	325	0.022
PEMS04	307	0.002
PEMS08	170	0.005

在时空预测研究中，地理空间中的每个采集点通过其位置相互关联，形成了复杂的空间关系网络。为了有效地捕捉和表示这些空间关系，研究者通常采用预定义图的构建方法。最常见的预定义图构建方式是第 2.2.1 小节中提到的距离图构建方式，其核心是利用高斯核函数，根据节点的地理距离来计算邻接矩阵。表 4-1 展示了利用距离图构建得到的常用交通数据集预定义图的稀疏度（非零个数/全部个数）。图 4-1 则进一步可视化了预定义图的连接强度分布。从表格和图中可以观察到，通过距离图构建的预定义图呈现出较高的稀疏性。这种预定义

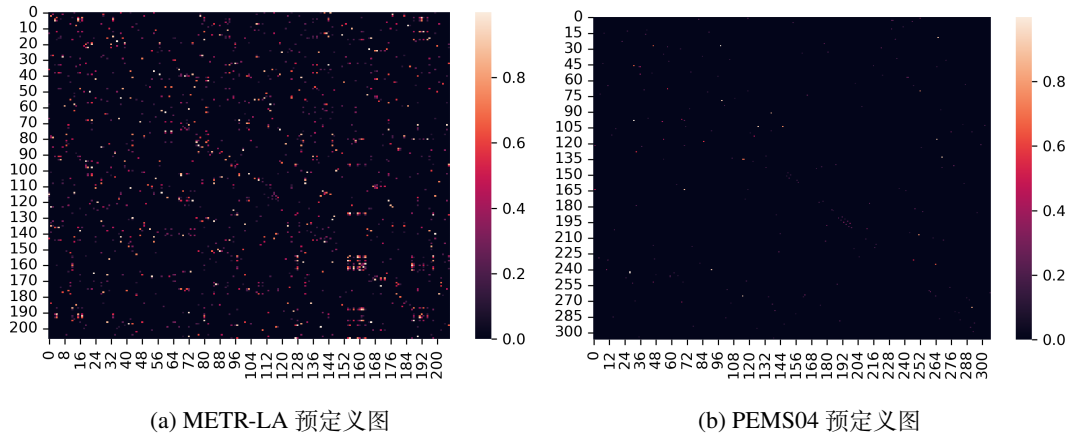


图 4-1 预定义图分布

图的构建基于一个地理空间假设：即所有事物均存在一定程度的相关性，而空间上相近的事物之间的关联性更为显著。这一假设促使研究者采用基于距离的权重分配方法，以量化地理空间中的相互作用。然而，城市结构的复杂性及其功能区域的划分引入了时空数据的空间层次性。这种层次性意味着，尽管地理位置相距较远，但功能相似的区域可能展现出强烈的空间依赖性。因而，高度稀疏的预定义图可能并未能充分表示出实际存在的隐式强连接。

此外，随着时间的推移，节点间的连接强度也可能会发生变化。因而，静态的预定义图也难以适应空间动态变化的需求。研究者需要探索更为动态和灵活的图构建方法，以更准确地捕捉时空数据的复杂性，并适应其随时间变化的特性。当前，不少研究^[61,72]致力于动态图的生成，然而，尽管这些研究中模型生成的动态空间图的稀疏度达到了 1.0，即所有节点之间都存在连接，但往往存在一个问题：大多数节点间的连接强度都相当接近，导致关系强弱难以区分，从而形成了一种均匀分布的连接模式，如图 4-2(a) 所示。图 4-2(b) 直方图则进一步展示了左侧图的分布直方图。

针对动态图中连接强度分布较为均匀的局限性，对图结构进行稀疏化处理显得尤为重要。稀疏化处理不仅能够优化模型的计算效率，避免在计算过程中处理大量冗余的弱连接，而且还能够促进网络在学习过程中更加关注强连接。通过这种方式，模型能够更有效地识别和利用空间关系中的关键信息，从而提高时空预测的准确性和可靠性。

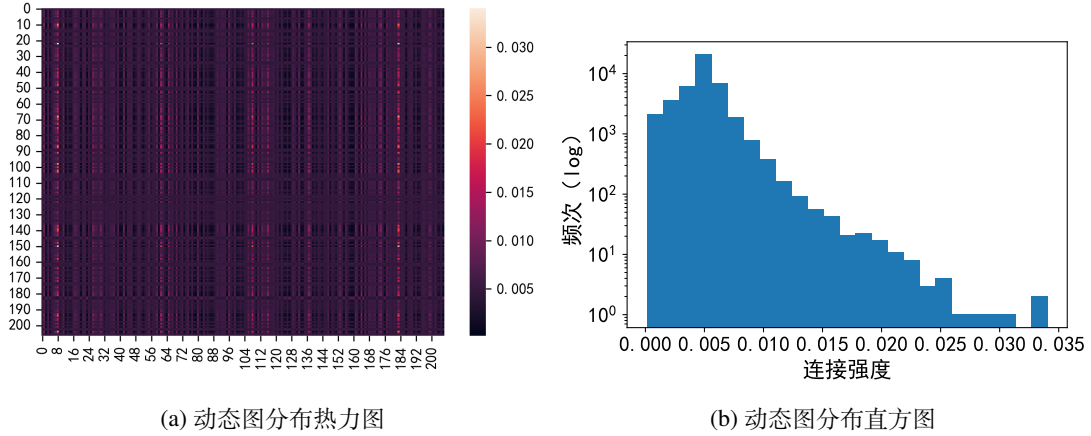


图 4-2 自学习动态图分布

4.2 算法设计

针对预定义图缺乏随时间变化的空间动态性以及动态图缺乏密度限制，本文提出了一种基于稀疏动态图的短期时空预测算法 ThermoNet。本节将首先明确研究问题的定义，随后详细介绍 ThermoNet 模型的整体架构以及构成各个模块的实现细节。

4.2.1 问题定义

短期时空预测问题定义如下：给定预定义空间有向图 $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ ，其中 \mathcal{V} 是节点集合，且 $|\mathcal{V}| = N$ ， N 为节点个数， \mathcal{E} 是边集合， $W \in \mathbb{R}^{N \times N}$ 是节点的邻接矩阵。在图 \mathcal{G} 上观察到的流量表示为：

$$\mathbf{X} = [X_1^t, X_2^t, \dots, X_N^t]_{t=1}^T \in \mathbb{R}^{T \times N}, \quad (4-1)$$

其中， T 为回看窗口长度， X_i^t 表示在时刻 t 第 i 个节点的值。预测值为

$$\mathbf{Y} = [X_1^t, X_2^t, \dots, X_N^t]_{t=T+1}^{t=T'} \in \mathbb{R}^{T' \times N}, \quad (4-2)$$

其中， T' 是预测窗口长度。问题的目标是学习一个映射函数 h 来预测短期内的时序数据，即给定图 \mathcal{G}

$$\mathbf{Y} = \mathbf{h}(\mathbf{X}; \mathcal{G}). \quad (4-3)$$

注意，本节所探讨的时空预测问题与本文第 3.2.1 小节中所述的长期时间序列预测问题存在一定差异。具体而言，本节关注的问题涉及的变量（节点）在空间上存在明确的相互关联性，这是第 3.2.1 小节中所未涉及的。此外，本节的目的是预测短期内的变量状态，其预测的时间跨度相较于第 3.2.1 小节中讨论的长期预测而言，显著较短。

4.2.2 模型结构

如图 4-3 所示，ThermoNet 模型结构由数据嵌入层、编码层和解码层组成。数据嵌入层用于学习原始数据的内在关系，让模型能够生成表达相关性的向量。编码层用来捕捉数据中的时间依赖和空间依赖关系，由两部分组成：首先通过原始 Transformer 的编码层对数据中的时间依赖关系进行学习，其次利用多头稀疏空间注意力生成动态空间图，并用图卷积层来提取空间依赖关系。最后用线

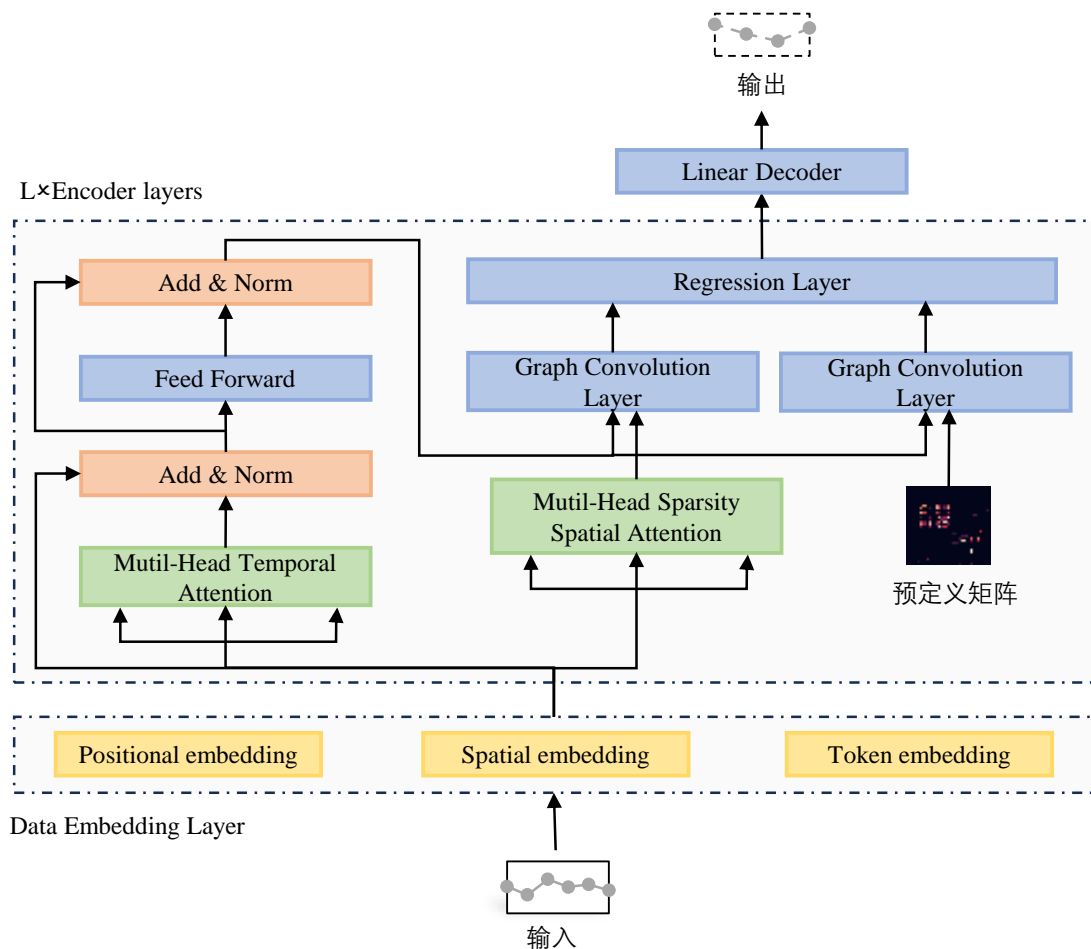


图 4-3 ThermoNet 网络结构图

性层作为解码层，输出预测结果。接下来，本节将详细介绍模型结构中的关键模块，包括数据嵌入层的设计、多头注意力机制的实现、动态稀疏空间图的构建以及图卷积层的实现。

4.2.3 数据嵌入层

数据嵌入层将时间序列数据转换为高维数据，包含了三种嵌入技术：位置嵌入（Positional Embedding）、空间嵌入（Spatial Embedding）、标识嵌入（Token Embedding）。

位置嵌入在没有循环结构的网络结构中引入了序列的相对位置信息，使得模型能够区分序列中的不同位置。位置嵌入使用不同频率的正弦和余弦函数来生成编码，编码公式如公式 4-4 所示。

$$\begin{aligned} PE(pos, 2i) &= \sin(pos/10000^{2i/d_{\text{model}}}), \\ PE(pos, 2i+1) &= \cos(pos/10000^{2i/d_{\text{model}}}), \end{aligned} \quad (4-4)$$

其中， pos 表示位置， i 表示序列的维度索引。

空间嵌入将空间中的 N 个离散的点映射到高维空间中连续的向量空间，为每个节点赋予一个特征表示向量。通过一个嵌入矩阵 $W_{\text{spatial}} \in \mathbb{R}^{N \times d_{\text{model}}}$ 来实现，其中 N 表示节点个数。

标识嵌入通过一维卷积将原始数据进行维度扩展：

$$X_{\text{token}} = \text{Conv1d}(X), \quad (4-5)$$

这种嵌入方式通过滑动卷积核的方式，在生成某一时间点数据的编码时，不仅考虑了当前时间点的数据，还考虑了周围时间点数据的局部特征，使得编码具有更丰富的时间上下文信息。

最终，经过数据嵌入层后，原始输入数据 $X \in \mathbb{R}^{N \times T \times 1}$ 被转化为 $X_{\text{embed}} \in \mathbb{R}^{N \times T \times d_{\text{model}}}$ ：

$$X_{\text{embed}} = X_{\text{position}} + X_{\text{spatial}} + X_{\text{token}}. \quad (4-6)$$

4.2.4 多头注意力机制

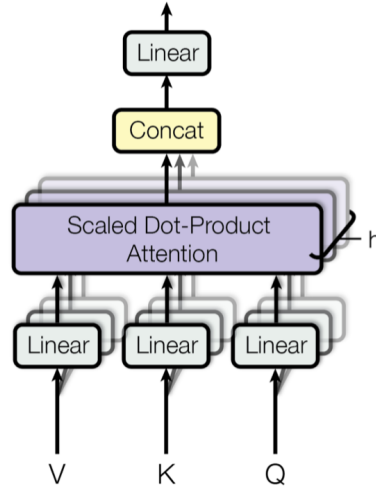


图 4-4 多头注意力机制示意图¹

数据嵌入层的输出 X_{embed} 为第一层编码层的输入，记作 $Z^{(0)} = X_{\text{embed}}$ 。如图 4-3 所示，ThermoNet 的编码层中有两种类型的多头注意力机制：第一种为多头时间注意力机制，其在时间维度上进行计算；第二种是多头稀疏空间注意力机制，其在空间维度上进行计算。本节介绍多头时间注意力机制，在此基础上，下一小节将介绍本文设计的多头稀疏空间注意力机制。多头注意力机制整体结构如图 4-4 所示。

注意力机制的核心是计算注意力权重分布，将 Query 与一组键值对 (Key-Value) 映射到输出。设前一个编码层的输出为 $Z^{(l-1)}$ ，Query、Key、Value 生成方式为：

$$Q = W_q Z^{(l-1)}, K = W_k Z^{(l-1)}, V = W_v Z^{(l-1)}, \quad (4-7)$$

其中， $Q, K, V \in \mathbb{R}^{N \times T \times d_{\text{model}}}$ ， $W_q, W_k, W_v \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ 为可学习的参数。

本文采用缩放点积方式计算注意力权重矩阵 $A \in \mathbb{R}^{T \times T}$ ：

$$A = \sigma \left(\frac{QK^T}{\sqrt{D_k}} \right), \quad (4-8)$$

其中， $\sigma(\cdot)$ 是 Softmax 函数。易知，注意力权重矩阵表示了时间维度上的依赖关

¹ 图片引用自文献^[8]

系，根据输入数据动态变化。通过加权值的方式，产生注意力输出：

$$\text{Attention} = \sigma \left(\frac{QK^T}{\sqrt{D_k}} \right) V. \quad (4-9)$$

在上述注意力机制的基础上，多头注意力机制通过并行地执行多个注意力计算来实现其功能，其使用不同的权重矩阵 W 来独立地投影输入，生成多组不同的 Q 、 K 、 V 矩阵，进而计算出多个注意力输出：

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \end{aligned} \quad (4-10)$$

其中， head_i 表示第 i 个注意力头的输出， W^O 是权重矩阵。通过这种方式，多头注意力机制允许模型关注不同表示子空间的信息。

4.2.5 稀疏空间图生成

在第 4.1 节中，本文指出了仅依赖于与距离矩阵相关的预定义图来进行时空预测存在不足，动态的节点相关图对于捕捉时空数据的复杂性至关重要。为此，ThermoNet 网络采用了一种基于空间注意力机制的方法来构建动态节点相关图，并通过对注意力权重矩阵实施稀疏性约束，实现了对动态图稀疏度的有效控制。本小节首先介绍动态空间图的生成过程，随后介绍如何在生成的空间图上执行稀疏化操作。

相较于时间注意力机制，空间注意力关注空间维度。其映射得到的 Q, K, V 维度为 $\mathbb{R}^{N \times T \times d_{\text{model}}}$ 。随后经过激活函数 σ' 得到初始的空间注意力矩阵 $M \in \mathbb{R}^{N \times N}$ ：

$$M_n = \sigma' \left(\frac{Q_n K_n^T}{\sqrt{D_k}} \right), \quad (4-11)$$

σ' 是 Softmax 函数的变体，其具体计算细节在稍后公式 4-15 中介绍。这里本文只保留了注意力权重矩阵，而没有进一步计算 Value 向量的加权输出。

为了消除注意力权重矩阵中的微小值并实现稀疏化，本文引入了硬阈值函数来筛选出高于特定阈值的注意力权重，从而生成了稀疏的空间图 A_s 。具体的

计算过程如下：

$$A_s = g(M_n, \tau) = \begin{cases} 0, & \text{如果 } M_n(i, j) < \tau, \\ M_n(i, j), & \text{其他,} \end{cases} \quad (4-12)$$

其中， $M_n(i, j)$ 为 M_n 矩阵中第 i 行、第 j 列的元素。 τ 为预定义的阈值，在实验中， τ 设定为 $\frac{1}{N}$ ， N 为节点数。生成的空间图 $A_s \in \mathbb{R}^{T \times N \times N}$ ，其随着时间步长的变化而变化。

硬阈值函数可以对权重矩阵中较小的值进行直接过滤，但并没有改变注意力权重的分布情况。本文结合热力学中的玻尔兹曼分布提出了 Softmax 函数的一种变体，让重要部分位置的注意力分数更大，不重要部分的注意力分数更小。热力学中的玻尔兹曼分布可以表示为

$$P(z_i) = \frac{e^{Q(z_i)/T}}{\sum_{j=1}^N e^{Q(z_j)/T}}, \quad (4-13)$$

其中， $Q(\cdot)$ 为状态的能量， T 是系统温度，在高温极限下，所有状态都近似相同，分布趋于均匀。当温度 T 逐渐减小时，玻尔兹曼分布逐渐从离散均匀分布转变为偏向于高能量状态的分布。当 $T = 1$ 时，为 Softmax 函数：

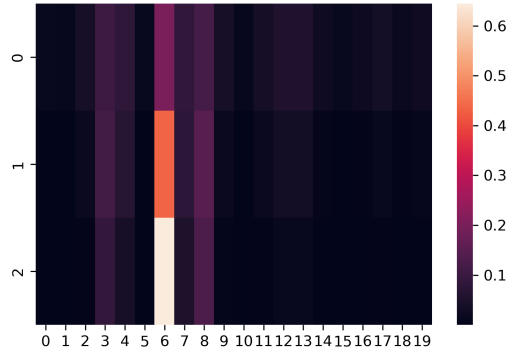
$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}}. \quad (4-14)$$

本文用 β 替换公式 4-13 中的 $1/T$ 部分，得到 $\sigma'(\cdot)$ 函数：

$$\sigma'(z_i) = \frac{e^{\beta z_i}}{\sum_{i=1}^n e^{\beta z_i}}, \quad (4-15)$$

假设随机生成了 20 个数，分别设置 $\beta = 1.0, 2.0, 3.0$ ，可以得到如图 4-5 所示的分布，从上到下分别对应 1.0, 2.0, 3.0，可以看出与前述分析一致， β 越大，分数分布越偏向于“两极化”，重要的位置注意力分数更大。

为了找到最优的 β 值，本文所使用的 β 是动态调整的， β 的变换范围是根据 β 的上界和下界生成的指数型序列。此外，预设动态生成图需达到的稀疏度 ρ ，用于指导动态生成图的稀疏度。该稀疏度目标基于数据集预定义图稀疏度和

图 4-5 不同 β 对应的分布

训练过程中数据集所能达到的最小稀疏度来综合确定的，它是一个可调整的超参数。在训练过程中，如果模型生成的邻接矩阵对于超过 70% 的样本未能达到预设的稀疏度 ρ ，那么 β 值将逐步增加以促进生成更稀疏的图。反之，如果模型已经能够满足稀疏度要求，则 β 的更新将停止。 β 的更新规则详见算法 1 所示。

算法 1 β 更新规则

输入: $\beta_{(i-1)}$: 第 $i-1$ 训练轮 β 值; j : β -range 的索引.

参数: I : 总训练轮数; L, U : β 上下界; sp : 生成图的稀疏度; W : 总样本数;
 V : 达到预设稀疏度 ρ 的样本数.

输出: $\beta_{(i)}, j$.

- 1: β -range \leftarrow LogFunction(I, L, U);
 - 2: $A_s^{(l)} \leftarrow g(M_n, \tau)$;
 - 3: $r \leftarrow$ ComputeSparsityFunction($A_s^{(l)}$);
 - 4: **if** $r < \rho$ **then**
 - 5: **if** $\frac{V}{W} > 0.7$ **then**
 - 6: $j \leftarrow j$;
 - 7: $\beta_{(i)} \leftarrow \beta_{(i-1)}$;
 - 8: **else**
 - 9: $j \leftarrow j + 1$;
 - 10: $\beta_{(i)} \leftarrow$ SelectFunction(β - range, j);
 - 11: **end if**
 - 12: **else**
 - 13: $j \leftarrow j$;
 - 14: $\beta_{(i)} \leftarrow \beta_{(i-1)}$;
 - 15: **end if**
 - 16: **return** $\beta_{(i)}, j$.
-

4.2.6 图卷积层

图卷积层的核心在于通过节点间的信息交换来提炼节点的表示。对于任何一个给定节点，图卷积操作首先聚合其邻近节点的信息，从而形成该节点的中间层表示。接着，通过应用线性变换和随后的非线性激活函数，对中间表示进行进一步的处理和转换。

由于预定义图表示了节点之间的地理距离，动态空间生成图表示了随着时间变化的节点动态关系，这两种图对模型学习都是有用的，因此本文分别对动态空间生成图 A_s 和预定义图 A_g 执行图卷积并将结果拼接。图卷积操作的输入节点特征为提取时间依赖模块的输出 $X_t^{(l)}$ ，输入图分别为 A_s 和 A_g ，输出为 $h^{(l)}$ ：

$$\begin{aligned} GCN^{(l)}(X_t^{(l)}) &= \left[\text{ReLU}(A_s^{(l)} X_t^{(l)} W_s), \text{ReLU}(A_g X_t^{(l)} W_g) \right] \\ h^{(l)} &= \text{ReLU}(\text{BatchNorm}(g^{(l)})), \end{aligned} \quad (4-16)$$

其中， W_s 、 W_g 是可学习的投影矩阵。

4.3 实验与分析

4.3.1 实验设置

数据集

为了验证 ThermoNet 的效果，本文在四个交通数据集上进行了实验，包括 METR-LA^[73]、PEMS-BAY^[74]、PEMS04^[74]、PEMS08^[74]数据集，这些数据集中每一列代表着某一个传感器采集的交通流量数据，每个数据集的统计特性如表 4-2 所示。

为确保实验公平性，本文沿用了以往工作的数据集划分比例。对于 METR-

表 4-2 短期时空预测实验数据集

数据集	节点数	采样频率	时间步长
METR-LA	207	5 分钟	34272
PEMS-BAY	325	5 分钟	52116
PEMS04	307	5 分钟	16992
PEMS08	170	5 分钟	17856

LA 和 PEMS-BAY 数据集, 70% 的数据用于训练集, 10% 的数据用于验证集, 20% 的数据用于测试集。对于 PEMS04 和 PEMS08 数据集, 60% 的数据用于训练集, 20% 的数据用于验证集, 20% 的数据用于测试集。在所有实验中, 输入序列的长度和预测输出的长度均固定为 12, 对应于 1 小时的时间跨度。

评估指标

实验采用的评估指标是均方误差函数 (MSE)、平均绝对误差 (MAE) 和平均绝对百分比误差 (MAPE)。MSE 和 MAE 的计算见公式 3-13 和公式 3-14, MAPE 的计算公式如下:

$$\text{MAPE} = \frac{1}{n} \sum_{j=1}^n \left(\left| \frac{\hat{y}_j - y_j}{y_j} \right| \right) * 100, \quad (4-17)$$

其中, n 是变量个数, \hat{y} 是模型预测值, y 是真值。为了与以往工作保持一致, 在 METR-LA 和 PEMS-BAY 数据集上, 本文评估预测序列维数为 3、6、12 的单一预测性能, 即评估第 15 分钟、第 30 分钟以及第 60 分钟的预测性能。而在 PEMS04 和 PEMS08 数据集上, 本文计算了 12 个步长的平均预测性能。

实现细节

实验在 GPU 型号为 NVIDIA RTX 2080Ti 上进行。Pytorch 框架使用 1.10.0 版本, 优化器使用 Adam 优化算法, 损失函数采用 HuberLoss。学习率设为 1×10^{-4} , batch 大小设为 32。预定义图矩阵采用 DCRNN 方法^[27]的生成方式, 与基线方法使用的预定义图矩阵一致。

基线方法

本章所提出的方法与以下基线方法对比:

- STGCN^[24]: 提出了只由卷积神经网络构成的学习框架, 通过时空卷积模块将两个门控时间卷积和空间图卷积整合到一起。只使用卷积结构使得训练速度更快, 参数更少;

- DCRNN^[27]: 将交通流建模为有向图上的扩散过程, 以此来捕获复杂的时空依赖。通过模拟图上的随机游走来捕获节点之间的空间关系, 并使用带有计划采样的编解码器架构来捕获时间依赖性;
- GMAN^[32]: 设计时空门控融合机制, 自适应地融合时空注意力机制提取的信息。并且提出转移注意力机制, 提取历史时间步与未来时间步的依赖关系, 将历史特征转移到未来表征, 避免误差累积现象;
- STTN^[75]: 提出空间-时间 Transformer 网络范式, 交替地建模空间和时间依赖性, 动态捕获隐藏空间模式和时间依赖性。
- GTS^[76]: 动态地发现图之间的关系, 将图结构和预测模型一起学习, 而不需要预定义的图结构。同时通过优化图分布上的均值性能来学习概率图模型, 以便可以可微分地采样离散图;
- STNorm^[77]: 提出两种归一化模块——时间归一化和空间归一化, 减少系统的复杂外部影响并突出数据的内部特性, 可以集成到其他网络架构中, 提高架构的预测性能。

4.3.2 对比实验

在 PEMS04 和 PEMS08 数据集上的实验表现如表 4-3 所示。从结果中可以观察到, 本文提出的方法取得了较好的性能提升。鉴于 PEMS04 和 PEMS08 数据集的预定义图极为稀疏, 依赖仅预定义图的建模方法, 如 STGCN 和 DCRNN, 未能充分挖掘数据的潜在结构, 表现得相对较差, 而使用动态图建模的方法 GMAN、GTS 和 STNorm 也表现出了不错的性能。本文所提出的方法相比于其他方法, 将动态空间注意力机制进行稀疏化并生成动态图的方式能够更好地捕捉到图中的强关系连接。

表 4-3 在 PEMS04 和 PEMS08 数据集上 ThermoNet 实验结果

数据集	指标	STGCN	DCRNN	GMAN	STTN	GTS	STNorm	ThermoNet
PEMS04	MAE	21.76	22.74	19.14	19.48	20.96	<u>18.96</u>	18.86
	RMSE	34.77	36.58	31.60	31.91	32.95	<u>30.98</u>	28.72
	MAPE	13.87	14.75	13.19	13.63	14.66	12.69	<u>13.10</u>
PEMS08	MAE	17.84	18.19	<u>15.31</u>	15.48	16.49	15.41	14.90
	RMSE	27.12	28.18	<u>24.92</u>	24.97	26.08	24.77	23.03
	MAPE	11.21	11.24	10.13	10.34	10.54	<u>9.76</u>	9.45

表 4-4 在 METR-LA 和 PEMS-BAY 数据集上 ThermoNet 实验结果

数据集	预测范围	指标	STGCN	DCRNN	GMAN	STTN	GTS	STNorm	ThermoNet
METR-LA	15min	MAE	2.88	2.77	2.80	2.79	<u>2.75</u>	2.81	2.70
		RMSE	5.74	5.38	5.55	5.48	<u>5.27</u>	5.57	5.14
		MAPE	7.60	7.30	7.41	7.19	<u>7.12</u>	7.40	6.93
	30min	MAE	3.47	3.15	<u>3.12</u>	3.16	3.14	3.18	3.11
		RMSE	7.24	6.45	6.49	6.50	<u>6.33</u>	6.59	6.17
		MAPE	9.60	8.80	8.73	8.53	8.62	8.47	<u>8.53</u>
	60min	MAE	4.59	3.60	3.44	3.60	3.59	3.57	<u>3.56</u>
		RMSE	9.40	7.59	<u>7.35</u>	7.60	7.44	7.51	7.24
		MAPE	12.70	10.50	10.07	<u>10.16</u>	10.25	10.24	10.23
PEMS-BAY	15min	MAE	1.36	1.38	1.34	1.36	1.37	<u>1.33</u>	1.32
		RMSE	2.96	2.95	2.82	2.87	2.92	<u>2.82</u>	2.80
		MAPE	2.90	2.90	2.81	2.89	2.85	2.76	<u>2.81</u>
	30min	MAE	1.81	1.74	1.62	1.67	1.72	1.65	<u>1.65</u>
		RMSE	4.27	3.97	3.72	3.79	3.86	3.77	<u>3.75</u>
		MAPE	4.20	3.90	3.63	3.78	3.88	<u>3.66</u>	3.73
	60min	MAE	2.49	2.07	1.86	1.95	2.06	1.92	1.97
		RMSE	5.69	4.74	4.32	4.50	4.60	<u>4.45</u>	4.47
		MAPE	5.80	4.90	4.31	4.58	4.88	<u>4.46</u>	4.62

在 METR-LA 和 PEMS-BAY 数据集上的实验表现如表 4-4 所示。与 PEMS0x 数据集相比，这两个数据集的预定义图较为密集，因而 STGCN、DCRNN 方法的预测表现与其他方法的差距不太大。同时，注意到，本文所提出的方法在预测 60 分钟的流量时，表现均不是最佳的，主要是因为 ThermoNet 方法主要改进点在于动态空间图的构建，而对于短期预测问题中的较长期时间依赖并没有进行过过多的探索。虽然本文第三章所提出的方法针对长期时序预测问题，但其所采用的输入是较长的，因而可以使用多尺度的空洞卷积对输入提取长期依赖。但本章讨论的问题中空间图的构建与输入序列的长度密切相关。当输入序列过长时，空间复杂度会显著增加。此外，根据之前的实验探索，适度增加短期预测问题中的输入长度对短期预测性能的提升作用有限。这一发现有待在未来的研究中进一步验证和探索。

4.3.3 消融实验

为验证 ThermoNet 网络中各个模块的作用，本文做了四组消融实验，分别是时间/空间模块、预定义图/动态生成图、数据嵌入以及空间图稀疏策略的消融实验。本文也将生成的动态空间图进行可视化。接下来，本文将分别介绍这些实验。

表 4-5 时间模块和空间模块对预测结果的影响

预测范围	15min			30min			60min		
指标	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o M_T	2.86	5.31	7.53	3.26	6.27	8.99	3.76	7.44	10.64
w/o M_S	2.99	5.94	7.96	3.62	7.33	10.31	4.54	9.16	13.80
w/o A_g	2.75	5.24	7.20	3.12	6.20	8.63	3.55	7.24	10.28
w/o A_s	2.81	5.35	7.35	3.27	6.55	9.14	3.86	7.86	11.35
ThermoNet	2.70	5.14	6.93	3.11	6.17	8.53	3.56	7.24	10.23

第一组和第二组消融实验在 METR-LA 数据集上进行, 实验结果合并并在表 4-5 中展示, 表 4-5 中的四种模型变体分别为

- w/o M_T : 去除网络结构中的多头时间注意力机制, 即去除图 4-3 左边第一列的网络层, 将数据嵌入层的输出当作图卷积层的输入, 只保留提取空间依赖关系的模块;
- w/o M_S : 去除网络结构中的多头空间注意力机制和所有图卷积层, 只保留提取时间依赖关系的模块;
- w/o A_g : 不使用预定义的生成图, 只使用网络生成的动态空间图做图卷积;
- w/o A_s : 不使用网络生成的动态空间图卷积, 只使用预定义的生成图做图卷积。

时间/空间模块

从表 4-5 的前两行的实验结果可以观察到, 移除时间模块 (w/o M_T) 和空间模块 (w/o M_S) 均会导致预测性能显著下降。此外, 仅保留时间模块的预测效果不如仅保留空间模块的情况, 这一发现强调了在短期交通流量预测中, 考虑节点间空间关系的重要性。这些结果表明, 空间模块在建模交通流量的短期变化中起到了关键作用, 而时间模块虽然也很重要, 但在没有空间模块支持的情况下, 其预测效能有限。因此, 为了实现精确的短期交通流量预测, 必须综合考虑时间和空间两个维度对预测结果的共同影响。这种综合考量不仅能够提升模型的预测能力, 还能够更全面地捕捉交通系统的动态特性。

预定义图/动态生成图

从表 4-5 中后三行的实验结果可以观察到，无论是基于地理距离构建的预定义关系图，还是由 ThermoNet 网络动态生成的空间关系图，两者对于短期交通流量预测模型的性能均具有显著作用。实验数据表明，移除任何一个关系图均会导致预测准确性的降低。进一步分析发现，相较于动态空间关系图，预定义关系图的移除对性能的影响较小。这一结果表明，尽管预定义关系图能够反映节点间的真实地理距离，从而对模型预测有所帮助，但动态空间关系图在捕捉时空数据中的复杂空间依赖性方面展现出更高的灵活性和有效性。因此，在进行短期交通流量时空预测时，动态空间关系图的引入是非常有必要的，此外，也验证了本文所提出的多头稀疏空间注意力机制和双图卷积层模块在建模动态空间依赖关系方面的有效性。

三种数据嵌入

位置嵌入 (Positional Embedding) 和标识嵌入 (Token Embedding) 都可以用于捕捉时间序列数据中的时序依赖性，前者引入了序列中元素的相对位置信息，后者引入了邻近时间点的局部时间特征。大部分研究^[32-33]只使用了位置嵌入和空间嵌入 (Spatial Embedding) 对原始数据进行特征提取，但也有研究^[17,22]通过引入标识嵌入来增强对时间信息的捕捉能力。因而本文对三种嵌入的组合进行了实验，探究其有效性。实验结果如表 4-6 和表 4-7 所示。

综合分析预测指标，本文发现各种嵌入技术组合的预测指标表现呈现以下排序：Positional+Spatial+Token<Positional+Spatial<Spatial+Token<Positional。通过以上排序，本文可以得到以下结论：(1) 位置嵌入 (Positional Embedding) 和空间嵌入 (Spatial Embedding) 的引入，都可以使得模型性能显著提升。这说

表 4-6 在 METR-LA 数据集上不同 Embedding 对预测结果的影响

Embedding			15min			30min			60min		
Positional	Spatial	Token	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
✓			2.86	5.38	7.51	3.33	6.47	9.27	3.90	7.67	11.35
	✓	✓	2.80	5.26	7.20	3.16	6.22	8.56	3.63	7.32	10.40
✓	✓		2.72	5.12	6.93	3.11	6.12	8.48	3.55	7.20	10.19
✓	✓	✓	2.70	5.12	7.00	3.10	6.13	8.59	3.55	7.27	10.45

表 4-7 在 PEMS04 数据集上不同 Embedding 对预测结果的影响

Embedding			PEMS04		
Positional	Spatial	Token	MAE	RMSE	MAPE
✓			20.41	30.61	14.44
	✓	✓	19.23	29.26	14.11
✓	✓		19.12	29.05	12.94
✓	✓	✓	18.86	28.72	13.10

明了在时空数据建模中，考虑数据的序列顺序性以及节点的空间表示是非常重要的；(2) 当比较位置嵌入 + 空间嵌入与标识嵌入 (Token Embedding) + 空间嵌入的组合时，前者的性能更好，这说明引入序列的相对位置信息比引入局部特征信息更为关键，对于捕捉时空数据的复杂性具有更大的贡献。(3) 在位置嵌入和空间嵌入基础上，对于 METR-LA 数据集，引入标识嵌入对预测性能的提升并不显著；而在 PEMS04 数据集上，引入标识嵌入后模型性能有显著提高。这一差异可能源于 PEMS04 数据集有更为复杂的时序依赖关系。因此，基于上述实验结果，本文选择采用在位置嵌入和空间嵌入的基础上引入标识嵌入来提取关键的局部序列依赖信息，从而增强模型对时序特征的捕捉能力。

稀疏空间图

本文所提出的方法对生成的动态空间图采用了两种稀疏化策略：一种是硬阈值函数，另一种是基于自学习变量 β 的 Softmax 函数变体。由于 PEMS04 数据集在四个实验数据集的预定义图是最稀疏的，生成的动态图对预测是更为重要的。因此，本文选择了 PEMS04 数据集来深入探讨稀疏化操作对预测性能的影响，实验结果如表 4-8 所示。从实验结果中可以发现，单独采用任一种稀疏操

表 4-8 在 PEMS04 数据集上不同稀疏操作对预测结果的影响

操作		PEMS04		
硬阈值函数	Softmax 变体	MAE	RMSE	MAPE
		18.95	29.07	13.12
✓		19.05	29.06	13.86
	✓	19.11	29.29	13.15
✓	✓	18.86	28.72	13.10

作都没有带来性能的提升，然而，当同时采用两种稀疏操作时，虽然性能提升有限，但仍可见到一定程度的提升。采用 Softmax 函数变体会使得连接强度逐渐偏向“两极化”，让不重要的相关连接值更小，此时对这种不重要的值进行裁剪，才是比较合理的。相比之下，仅使用硬阈值函数可能会导致一些相对重要的连接被错误地移除，从而对预测产生不利影响。因而，本文所提出的这两种稀疏操作需要同时施加在动态生成的空间图上才能有一定的效果。

4.3.4 模型分析

动态空间图可视化

如图 4-6 所示，本文利用在 PEMS04 数据集上训练完成的模型进行了推理过程，并得到了相应的稀疏空间图的可视化结果。为了更清晰地展示结果，此处仅对 207 个节点进行了可视化处理。生成的空间图稀疏度为 0.46，显著高于 PEMS04 预定义图的稀疏度 0.002。这再次验证了预定义图不足以满足复杂的空间依赖，动态的空间图有利于模型提取动态空间依赖信息。

在生成的稀疏空间图中，仅有少数节点间的连接以“微亮的光点”形式呈现，而其余大部分节点间的连接权重较小，几乎不可见。这一现象表明，本文所提出的模型是有选择性的，倾向于强调那些对预测结果影响最大的节点连接。这种选择性强调不仅揭示了模型对于关键连接的敏感性，而且也指出了在动态空间图中，只有少数连接对模型的预测性能起着决定性作用。

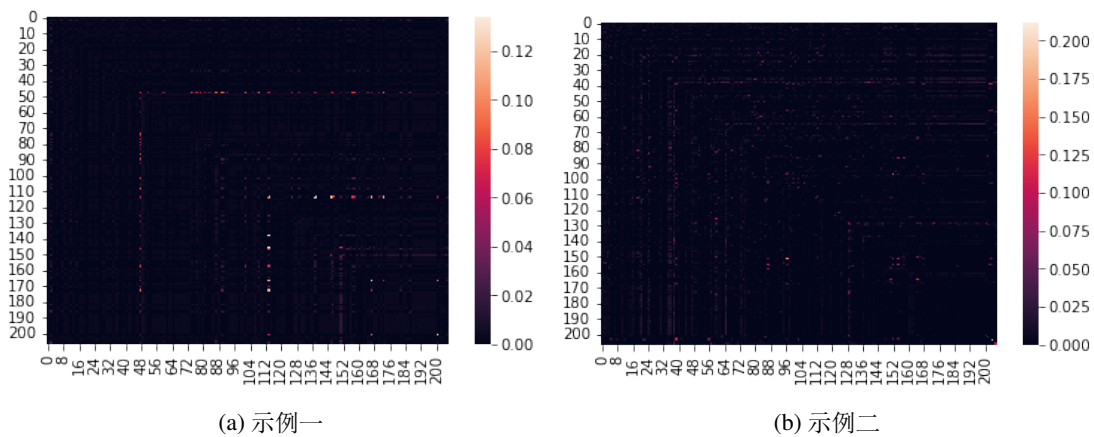


图 4-6 动态空间图可视化

β 上界分析

在本文所提出的方法中，引入了一个动态调整的参数 β ，其值随着训练过程逐步增加，在满足一定条件后停止增加。参数 β 的变化范围受到预设 β 上界限制，即 β 的上界决定了其增长速率。较大的 β 上界会导致更快的增长速度，不利于寻找 β 的最佳值。而过小的上界则可能不足以实现稀疏的效果。为了探究不同 β 上界对模型性能的影响，本文对 $\beta = 1, 5, 10, 20, 50, 100$ 进行了实验，如图 4-7 所示。实验结果显示，当 β 上界设定为 1 时，相当于保持 Softmax 函数不变，模型性能较差。而当 β 上界设为 20 时，模型表现出最佳性能，此时确定的最佳 β 值为 3.03。

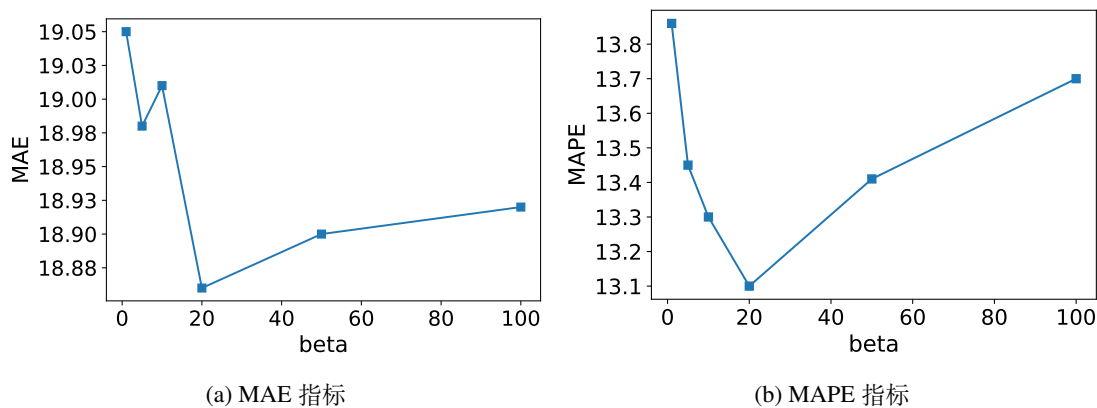


图 4-7 在 PEMS04 数据集上不同 β 上界对预测结果的影响

4.4 本章小结

本章首先对时空数据中的预定义图和以往模型生成的动态图进行分析。通过这一分析，本章指出了在时空预测任务中采用动态图的必要性，以及对动态图进行稀疏化处理的潜在优势。在此基础上，本章提出了一种基于稀疏图结构的短期时空预测网络模型 ThermoNet，并介绍了模型中的关键模块。随后，本章设计并进行了一系列对比实验和消融实验。这些实验结果不仅确保了研究动机的合理性，也验证了 ThermoNet 网络的有效性。

第五章 交通流量可视化与预测系统

为了验证本文提出的算法在时间序列预测领域的应用效果，本章节构建了一个针对交通流量的可视化与预测平台，并在此平台上应用了本文所提出的算法。交通流量预测任务可分为两种主要场景：第一种场景旨在预测未来较长时间内的交通流量，这要求算法能够准确捕捉并预测交通流量的长期趋势；第二种场景则关注短时间内的交通流量变化，需要算法能够实时提供各观测点的短期流量预测。本章所开发的系统能够同时满足上述两种预测需求。接下来，本章将详细阐述系统的研发背景、需求分析、架构设计、开发环境和功能模块的实现细节以及系统的整体效果。

5.1 系统研发背景

随着中国经济的快速发展和人民生活水平的显著提高，汽车已逐渐成为普通家庭的必备交通工具。根据国家相关部门的统计数据，2019 年底，中国的汽车保有量已达到 2.6 亿辆¹。而到了 2023 年底，这一数字进一步增长至 3.36 亿辆²。这一显著的增长反映了国内汽车消费需求的持续上升。然而，汽车数量的增加也带来了一系列城市交通问题，尤其是交通拥堵现象日益严重。特别是在上下班的高峰时段以及节假日期间，城市道路和高速公路上车辆行驶缓慢，显著增加了人们的通勤时间，严重影响了人们的出行效率和体验。为了缓解这一问题，政府相关部门已经开始倡导错峰出行策略，鼓励市民在非高峰时段进行出行，以分散交通压力。

智慧交通系统的构建对于提升交通管理效率至关重要。智慧交通系统利用先进的信息技术，如物联网、大数据分析和人工智能等，收集和大量的交通

1 <https://www.mps.gov.cn/n2254314/n6409334/c6852472/content.html>

2 <https://www.mps.gov.cn/n2254098/n4904352/c9384864/content.html>

数据。如图5-1所示，这些系统能够实时分析交通状况，及时向驾驶者提供路况信息，帮助他们规划出更加高效的出行路线。然而，尽管现有的交通导航软件能够提供基本的起终点路线规划，它们在地图更新的时效性和导航准确性方面仍存在不足。这可能导致规划的路线存在误差，驾驶员只能查看规划路线中各路路口的流量情况，无法查询单个路口的实时流量和未来流量。此外，现有软件提供的流量信息具有一定的滞后性。本章旨在开发一款细粒度的交通流量可视化与预测平台。该平台提供详细的单个路口的历史流量与未来流量信息，为驾驶员自行选择路线提供参考。



图 5-1 某地图软件实时交通情况

5.2 系统设计

本节首先从用户的实际需求出发，对使用该系统的场景进行了深入的需求分析。通过这一过程，本文确保了系统设计能够精准地满足用户的期望和业务需求。在此基础上，本文进一步设计了系统的详细架构。

5.2.1 系统需求分析

结合已有的交通出行平台的不足，本章节所设计系统的具体需求如下：

- 用户友好的交互界面。系统界面的设计应注重直观性和简洁性，确保用户能够轻松地浏览和操作系统。同时，系统必须提供一套完善的用户信息管理功能。这包括但不限于实现用户账户的注册、登录验证以及密码重置等基本操作。这些功能是确保用户能够安全、便捷地访问和使用系统的基础。
- 地理区域路口可视化。系统需提供用户自定义查询功能，允许用户选择特定的城市和区域。基于选定区域的路网信息，系统应生成主要路口的地理信息图，以直观展示各路口的位置分布。
- 历史和实时交通流量数据分析。系统应具备收集历史与实时交通流量数据的能力，并能够对这些数据进行综合统计与分析。通过对数据进行分析，系统将提供详尽的数据报告，使用户能够对历史交通流量模式有一个全面的理解。
- 未来交通流量数据预测。系统不仅需支持用户在行驶过程中实时查看路况信息，还应提供出行前的长期与短期交通流量预测服务。这将使用户能够根据预测结果，提前规划更为合理的出行路线。
- 未来出行细粒度建议。基于对未来交通流量的预测分析，系统应进一步结合各路口及周边区域的流量信息，为用户提供精细化的出行建议。这些建议将帮助用户在不同时间段内做出更优的出行决策。

基于以上详细需求分析，本研究旨在开发一个智能交通流量管理平台。该平台致力于满足用户在多个层面的需求，包括但不限于对历史和未来交通流量信息的查询，以及对出行规划的个性化建议。

5.2.2 系统架构设计

基于本系统的需求，本文进一步详尽规划了系统的架构设计，如图 5-2 所示。系统架构主要由三个关键组件构成：应用服务器、数据库服务器和训练服务器。每个组件的具体功能和作用如下：

- 应用服务器部署了 Web 应用程序，负责处理用户的请求和响应。应用服务器不仅实现了用户交互逻辑，还负责执行数据分析和业务逻辑处理。应用服务器接收来自用户的查询请求，调用相应的数据获取、处理和分析服务，并将结果以可视化的形式展现在 Web 界面上，提供直观的用户体验；

- 数据库服务器负责数据的存储和管理。不仅保存了用户的基本信息，还存储了区域路口的交通流量信息。数据库的设计确保了数据的完整性、安全性和高效访问，为系统的数据分析、模型训练和模型推理提供了数据基础；
- 训练服务器配备有高性能的计算资源，用于本文所提出的算法的训练和推理任务。训练服务器能够处理大量的数据，并执行复杂的算法。此外，训练服务器还负责模型的持续优化和更新。

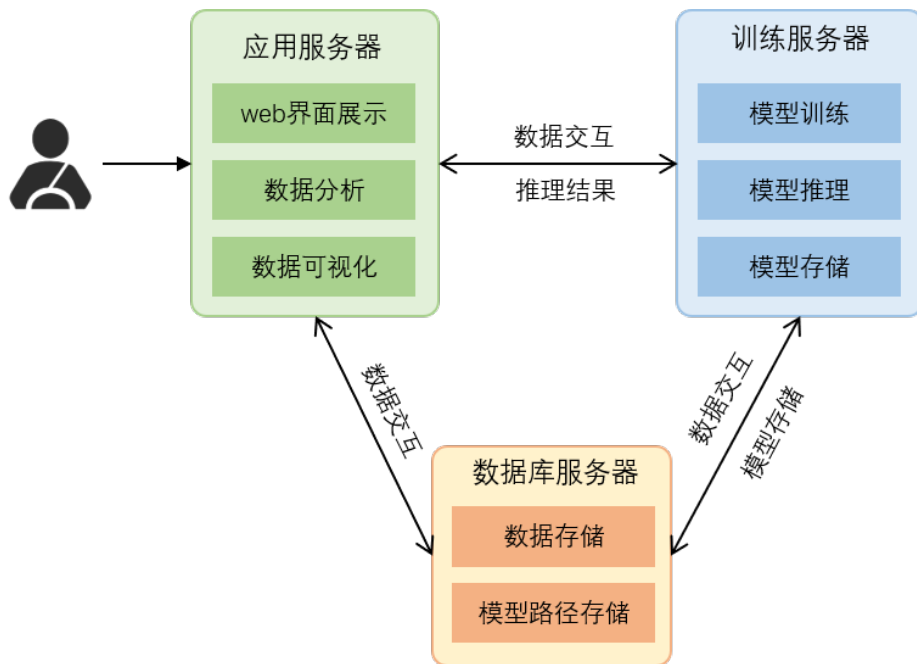


图 5-2 交通流量可视化和预测平台系统架构

三类组件间的主要交互流程如下：

- 用户交互与数据可视化：当用户通过 Web 界面发起查询时，应用服务器向数据库服务器请求必要的数据库，进行统计分析，并将分析结果显示至 Web 界面供用户查看。
- 模型训练与存储：模型训练阶段，训练服务器从数据库服务器批量提取所需数据。训练完成后，训练服务器将训练好的模型的存储路径更新至数据库服务器中，以便后续的推理任务调用。
- 预测推理与结果返回：对于预测功能，应用服务器首先从数据库服务器获取必要的历史和实时数据以及所需模型的存储路径，然后向训练服务器发送推理请求。训练服务器接收请求后，利用已训练好的模型进行推理分析，并将最终的预测结果返回给应用服务器，由应用服务器展示给用户。

5.3 系统实现

根据 5.2 节所述的系统需求和系统架构设计，本文实现了一个交通流量可视化和预测平台。在本节中，本文将介绍系统的开发环境，包括所采用的编程语言、软件工具和硬件资源等。此外，本节还将详细介绍各个系统功能模块的具体实现细节。

5.3.1 系统开发环境

交通流量可视化与预测系统属于基于深度学习的的海数据分析平台。本系统将应用服务和训练服务部署在同一台 Ubuntu 服务器上，由于深度学习模型的训练和推理过程对计算资源有较高的要求，本系统选择了配备 NVIDIA RTX 2080Ti 显卡和 Intel Xeon E5-2678 v3 (12 核) CPU 的 Ubuntu 服务器来支持这些计算密集型任务。这种硬件配置确保了模型训练和应用服务的顺畅运行，同时提供了足够的性能来处理复杂的数据分析工作。在软件实现方面，本系统采用了 Pytorch 作为主要的深度学习框架。系统开发采用了 Streamlit 框架，这是一个专为快速创建数据驱动型 Web 应用而设计的 Python 库。利用 Python 3.8.5 的库和功能，Streamlit 将数据分析脚本转换为交互式 Web 应用程序变得简单快捷。

5.3.2 系统功能模块

本系统主要的功能模块有两个：历史流量查询模块和未来流量预测模块。本小节将结合流程图详细介绍两大功能模块的实现逻辑。

历史流量查询模块允许用户查询特定时间段内的交通流量历史数据。通过这一模块，用户可以分析历史时间上的交通模式，提供历史流量参考。具体的实现逻辑如流程图 5-3 所示，当用户指定了查询的区域和时间范围后，系统首先对该区域进行粒度分析。系统从数据库中提取指定时间段内该区域及其邻近区域的交通流量数据，计算区域平均流量，并进行相互比较。基于比较结果，系统生成具体的通行建议：如果目标区域的平均流量显著低于邻近区域，表明该区域可能存在拥堵情况，系统会建议用户选择绕行路线；否则，系统将推荐用户正常通行。进一步地，系统对区域内的各个路口进行细致的分析，通过可视化工具展示每个路口的历史流量数据，并识别出在历史时间段内经常出现拥堵的时段。

利用这些信息，系统为用户提供每个路口的具体通行建议，帮助用户在规划路线时避开潜在的拥堵点。

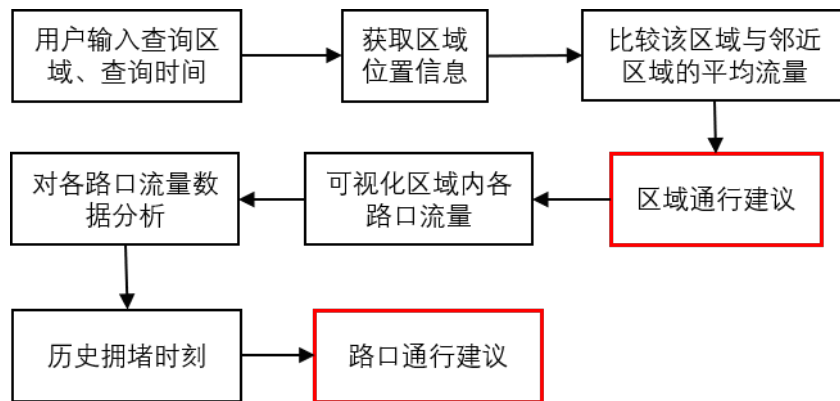


图 5-3 历史流量查询模块实现逻辑

未来流量预测模块提供对交通流量的长期和短期预测功能。包含了长期流量预测和短期流量预测两种模块。这两种模块的逻辑流程相似，只是使用的预测模型和预测的时间范围有所不同。长期流量预测专注于预测较长时间跨度（如未来一周）的流量预测，使用本文所提出的 MSDCN 算法进行预测；而短期流量预测模块则专注于预测即将到来的 1 小时内的流量变化，使用本文所提出的 ThermoNet 算法进行预测。下面统一介绍这两种预测模块的实现逻辑，如图 5-4 所示。当用户输入查询区域后，系统自动获取当前时间，并从数据库中检索模型推理所需的历史流量序列输入数据。随后，系统根据用户的需求调用相应的预测模型进行流量预测。预测完成后，系统将预测结果以可视化的形式展示给用户，同时对预测的流量值与当前流量进行比较分析。基于这一分析，系统提供路口通行建议。

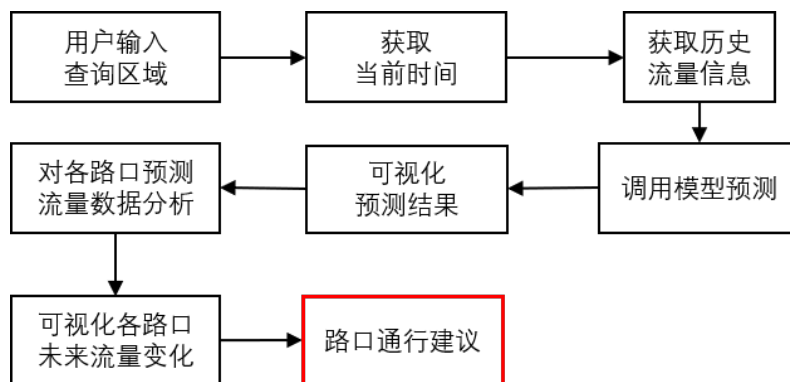


图 5-4 未来流量预测模块实现逻辑

5.4 系统效果展示

在本节中，本文将通过一些直观的可视化页面展示本章中实现的交通流量可视化与预测系统的主要功能和效果。

图 5-5 提供了系统首页的概览，其中详细介绍了系统的核心功能，并提供了一个开放的反馈窗口，以便用户可以提出建议和反馈，鼓励用户参与系统的改进和优化过程。导航栏包括了系统介绍、用户信息、历史交通流量查询、短期交通流量查询、长期交通流量查询五种条目。接下来将分别介绍后三种主要的条目。



图 5-5 系统首页界面

图 5-6 展示了系统的历史交通流量查询界面。在此界面中，用户能够通过输入具体的城市名称、区域、以及查询所需的日期和时间范围来定制化其查询请求。用户需指定查询的起始日期和结束日期，以及相应的起始时间和结束时间，以确定所需要分析的历史流量数据的时间窗口。用户填入的日期和时间均不能超过当前系统日期和时间。完成这些必要的输入后，用户点击提交按钮以触发查询操作。

图 5-7 展示了历史交通流量查询结果。为了更加清晰地呈现数据，网页左侧的导航栏在此视图被隐藏。图 5-7(a) 展示区域数据分析的结果，其中系统首先直观地呈现了查询区域内主要路口的地理位置，随后基于历史流量数据提供了针对性的区域内通行建议。比如，图 5-7 展示的区域通行建议为“您选择的这一

区域在此时段较为拥堵，建议您选择其他路线”。图 5-7(b) 描绘了路口级别的详细数据分析结果。系统首先以表格形式展示了所选区域内各主要路口随时间变化的交通流量数值，并通过黄色高亮标记出流量最小（即最拥堵）的值，以使用户快速识别潜在的拥堵点。随后，系统明确指出了历史数据显示的各路口最拥堵的具体时刻，并通过曲线图形式可视化地展示了各路口流量随时间的变化趋势，增强了数据的可读性。



图 5-6 历史交通流量查询界面

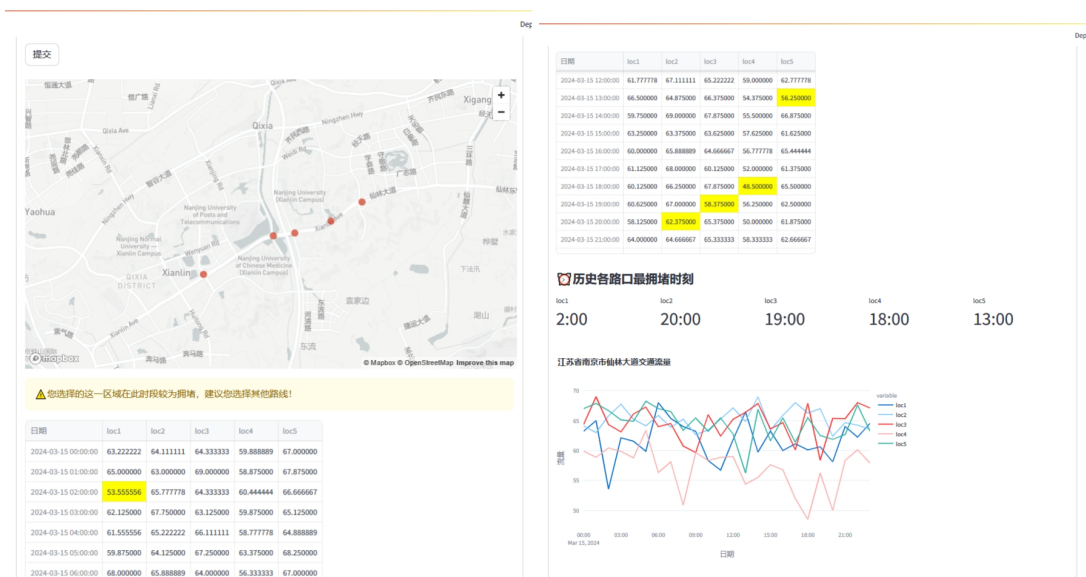


图 5-7 历史交通流量查询结果

图 5-8 展示了系统的短期流量查询界面。长期流量预测查询界面在设计上与短期流量查询界面保持一致性。在这个界面上，用户被要求输入特定的城市和区域信息，以便系统能够针对相应的地理位置提供短期交通流量预测。与历史流量查询不同，短期流量查询界面中的时间参数是由系统根据当前时间自动设定的，确保用户总是获得最新的预测数据。

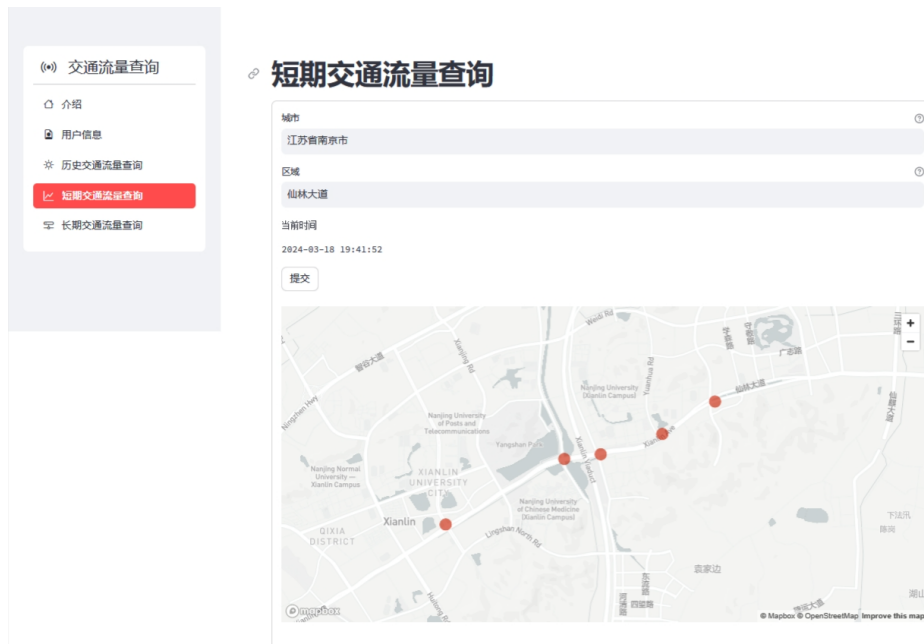


图 5-8 短期交通流量预测查询界面

图 5-9 分别展示了短期和长期交通流量预测结果。在图 5-9(a) 界面中，系统利用本文所提出的 ThermoNet 算法对即将到来的一小时内各个路口的交通流量进行了预测，预测结果通过直观的可视化曲线图形式展现。用户可以将光标放置在图上详细地查看曲线上每个点的具体值，同时通过绿色（下降）或灰色（提高）的箭头给出了在未来一天内各个路口的预计流量变化趋势和幅度，使用户直观地感受到各个路口的路况情况。随后，系统通过对预测的流量数据进行分析，给出了路口的综合性通行建议，比如图中所给出的建议为“该区域内路口 loc3、loc5 较为通畅，建议通行；该区域内路口 loc1、loc4 较为拥堵，建议绕行”。在图 5-9(b) 中，系统应用了本文提出的 MSDCN 算法来预测未来一周内各个路口的交通流量情况。与上述短期预测界面类似地，此界面基于长期预测流量数据综合分析，呈现了预测流量可视化、流量变化趋势和幅度以及路口通行建议。其中，长期流量预测曲线图的横轴展示了比短期预测更广泛的时间跨度，从而允许用户观察并分析更长周期内的交通流量趋势。



(a) 短期交通流量预测界面

(b) 长期交通流量预测界面

图 5-9 交通流量预测界面展示

5.5 本章小结

本章介绍了交通流量可视化与预测系统的详细设计和实现细节。本系统的开发背景基于对智慧城市建设需求的深入理解，特别是对于交通流量管理的智能化和数据驱动的需求。进一步地，本章通过对现有交通地图软件调研，明确了系统的需求，并进一步设计了系统的架构。同时，本章详细介绍了系统开发和部署环境以及各个系统功能模块的作用，确保了系统能够在稳定和高效的环境中运行。最后，本章通过一系列可视化的网页界面截图，展示了系统的实际使用流程和效果。

第六章 总结与展望

随着时间序列数据量的不断增长，挖掘历史数据中的潜在价值并据此对未来序列进行准确预测变得日益重要。本研究聚焦于时间序列预测任务，从长期时间序列预测和短期时空序列预测两个维度进行了深入的研究。本文设计并实现了一个交通流量可视化与预测系统，以将理论研究与实际应用相结合。系统应用了本文所提出的两种算法，展示了本文研究内容在实际场景中的应用潜力。

长期时间序列预测的研究主要面临如何有效建模长期依赖关系的挑战，这种依赖关系通常涉及多个不同的时间尺度。为了捕捉这些长期依赖性，模型通常需要处理较长的输入序列，一方面有助于提取关键的时间信息，但另一方面也不可避免地导致了高时间复杂度和空间复杂度的问题。针对这一问题，本研究提出了一种浅层多尺度空洞卷积网络结构，旨在提高模型的效率和性能。本研究中设计的并行多尺度空洞卷积层能够同时捕捉长期时间序列中的局部和全局依赖性，并通过加权融合的方式整合多尺度的特征表示。这种架构显著扩展了模型对序列的感受野，使得模型能够在保持较少参数的前提下，有效地处理更长时间的序列数据。此外，本研究还将传统的自回归模型整合到所提出的架构中，以提取序列中隐含的线性表征。本文提出的轻量级模型架构在保持高预测精度的同时，显著降低了模型的时间和空间复杂度，使得模型更加适合于实际应用场景。

对于一些具有明显空间关系的数据，其短期预测往往需要空间关系的辅助预测。尽管当前研究采用空间注意力机制来捕获节点间的依赖性，但在现实世界的数据集中，节点数量众多，而真正具有强相关性的节点对却相对有限。这导致注意力矩阵包含了大量冗余信息，因此本文提出了两种对空间注意力进行稀疏化的操作，分别为硬阈值函数和热力学系统启发的 Softmax 函数变体，旨在保留关键的节点连接。同时，本文提出的方法不仅对预定义的静态图进行图卷积操作，还对由空间注意力生成的动态图进行卷积，从而全面利用节点间的相互作用信息，有效地整合了静态和动态的空间关系，达到了较好的性能。为验证本

文提出方法中各个模块的有效性，本文进行了对模型包含的各个模块进行了充分的消融实验以及详尽的分析，实验结果验证了模型的有效性。

在上述两个研究的基础上，本文开发了交通流量可视化预测系统，为人们提供长期和短期的交通流量参考，帮助人们做决策，便利人们的生活。该系统允许用户选定特定的查询区域，并能够提供该区域的历史交通流量数据、短期（未来一小时）预测以及长期（未来一周）预测。系统通过数据可视化，将复杂的交通流量信息转化为直观的图形展示，使用户能够轻松分析交通状况。此外，系统对数据进行了智能分析，能够根据实时和历史数据提供区域化的出行建议，从而优化用户的出行计划。

结合本文的研究工作，未来的研究可以在以下几个方面进行拓展和深入探索：

1. 时间序列数据具有分布漂移现象，即测试数据和训练、验证数据分布不一致。这可能会导致模型训练时所依赖的数据分布将不再适用于测试数据，造成预测结果的较大误差，尤其对于长期预测来说误差会更大。在本文的研究中，本文采用了独立建模的方式来缓解分布漂移现象，实验结果表明了相比于联合建模，独立建模的方式预测效果更优。但这一建模方式可能会忽略掉多变量间的隐式关系，从而造成信息的丢失。因此，如何解决分布漂移问题，通过算法来识别和适应数据分布的变化，还需要进一步的理论研究和实验探索。

2. 对于具有明显空间关系的时序数据，需要进一步考虑如何有效地从数据中同时提取空间依赖关系和时间依赖关系，并促进这两种依赖性之间的相互作用，以实现更精准的短期预测。本文在短期时空预测方面的研究主要针对空间依赖建模，还需进一步探索空间依赖模块和时间依赖模块的有效融合机制。此外，当前一些研究提出了基于多层感知机的时空预测模型，并且预测性能良好。未来研究可以继续探索基于多层感知机的时空预测模型，通过精心的设计结构，利用轻量高效的网络结构来提取时空依赖。

3. 针对交通流量可视化与预测系统，未来可以进一步增加和优化系统功能，比如可以将短期流量查询和长期流量查询统一到一个界面中，用户可以自定义想要查询的未来时间跨度，从而便捷地获取所需信息。

参考文献

- [1] KAREVAN Z, SUYKENS J A. Transductive LSTM for time-series prediction: An application to weather forecasting[J]. *Neural Networks*, 2020, 125: 1-9.
- [2] PANG Y, YAO B, ZHOU X, et al. Hierarchical electricity time series forecasting for integrating consumption patterns analysis and aggregation consistency[C]// *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018: 3506-3512.
- [3] QIN Y, SONG D, CHENG H, et al. A dual-stage attention-based recurrent neural network for time series prediction[C]// *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2017: 2627-2633.
- [4] WANG J, JIANG J, JIANG W, et al. Towards efficient and comprehensive urban spatial-temporal prediction: A unified library and performance benchmark[J]. *ArXiv preprint arXiv:2304.14343*, 2023.
- [5] JORDAN M I. Serial order: A parallel distributed processing approach[G]// *Advances in Psychology: vol. 121*. Elsevier, 1997: 471-495.
- [6] ELMAN J L. Finding structure in time[J]. *Cognitive Science*, 1990, 14(2): 179-211.
- [7] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. *Neural Computation*, 1997, 9(8): 1735-1780.
- [8] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// *Advances in Neural Information Processing Systems: vol. 30*. 2017: 6000-6010.
- [9] CLEVELAND R B, CLEVELAND W S, MCRAE J E, et al. STL: A seasonal-trend decomposition[J]. *Journal of Official Statistics*, 1990: 3-73.

- [10] WEN Q, GAO J, SONG X, et al. RobustSTL: A robust seasonal-trend decomposition algorithm for long time series[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 5409-5416.
- [11] HYNDMAN R J, ATHANASOPOULOS G. Forecasting: Principles and practice[M]. 2018.
- [12] BOX G E, JENKINS G M, REINSEL G C, et al. Time series analysis: Forecasting and control[M]. John Wiley & Sons, 2015.
- [13] KIM T, YUE Y, TAYLOR S, et al. A decision tree framework for spatiotemporal sequence prediction[C]//Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2015: 577-586.
- [14] TYRALIS H, PAPACHARALAMPOUS G. Variable selection in time series forecasting using random forests[J]. Algorithms, 2017, 10(4): 114.
- [15] MILLER J A, ALDOSARI M, SAEED F, et al. A survey of deep learning and foundation models for time series forecasting[J]. ArXiv preprint arXiv:2401.13912, 2024.
- [16] LI S, JIN X, XUAN Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[C]//Advances in Neural Information Processing Systems: vol. 32. 2019: 5243-5253.
- [17] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 35: 12. 2021: 11106-11115.
- [18] WU H, XU J, WANG J, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting[C]//Advances in Neural Information Processing Systems: vol. 34. 2021: 22419-22430.
- [19] ORESHKIN B N, CARPOV D, CHAPADOS N, et al. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting[C]//International Conference on Learning Representations. 2020.

- [20] CHALLU C, OLIVARES K G, ORESHKIN B N, et al. N-HiTS: Neural hierarchical interpolation for time series forecasting[C] // Proceedings of the AAAI Conference on Artificial Intelligence: vol. 37: 6. 2023: 6989-6997.
- [21] WANG H, PENG J, HUANG F, et al. MICN: Multi-scale local and global context modeling for long-term series forecasting[C] // International Conference on Learning Representations. 2022.
- [22] WU H, HU T, LIU Y, et al. TimesNet: Temporal 2D-variation modeling for general time series analysis[C] // International Conference on Learning Representations. 2023.
- [23] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [24] YU B, YIN H, ZHU Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting[J]. ArXiv preprint arXiv:1709.04875, 2017.
- [25] SONG C, LIN Y, GUO S, et al. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting[C] // Proceedings of the AAAI Conference on Artificial Intelligence: vol. 34: 01. 2020: 914-921.
- [26] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks[J]. ArXiv preprint arXiv:1609.02907, 2016.
- [27] LI Y, YU R, SHAHABI C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting[J]. ArXiv preprint arXiv:1707.01926, 2017.
- [28] SHAO W, JIN Z, WANG S, et al. Long-term spatio-temporal forecasting via dynamic multiple-graph attention[J]. ArXiv preprint arXiv:2204.11008, 2022.
- [29] WU Z, PAN S, LONG G, et al. Graph wavenet for deep spatial-temporal graph modeling[J]. ArXiv preprint arXiv:1906.00121, 2019.

- [30] WU Z, PAN S, LONG G, et al. Connecting the dots: Multivariate time series forecasting with graph neural networks[C] // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 753-763.
- [31] LIN H, BAI R, JIA W, et al. Preserving dynamic attention for long-term spatial-temporal prediction[C] // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 36-46.
- [32] ZHENG C, FAN X, WANG C, et al. Gman: A graph multi-attention network for traffic prediction[C] // Proceedings of the AAAI Conference on Artificial Intelligence: vol. 34: 01. 2020: 1234-1241.
- [33] GUO S, LIN Y, WAN H, et al. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 34(11): 5415-5428.
- [34] YAN H, MA X, PU Z. Learning dynamic and hierarchical traffic spatiotemporal features with transformer[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(11): 22386-22399.
- [35] 梁宏涛, 刘硕, 杜军威, 等. 深度学习应用于时序预测研究综述[J]. 计算机科学与探索, 2023, 17: 1285-1300.
- [36] 汪维泰, 王晓强, 李雷孝, 等. 时空图神经网络在交通流预测研究中的构建与应用综述[J]. 计算机工程与应用, 1-15.
- [37] KITAEV N, KAISER L, LEVSKAYA A. Reformer: The efficient transformer[C] // International Conference on Learning Representations. 2019.
- [38] NIE Y, H. NGUYEN N, SINTHONG P, et al. A time series is worth 64 words: Long-term forecasting with transformers[C] // International Conference on Learning Representations. 2023.
- [39] ZHANG Y, YAN J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting[C] // International Conference on Learning Representations. 2023.

- [40] LIU Y, WU H, WANG J, et al. Non-stationary transformers: Exploring the stationarity in time series forecasting[C]//Advances in Neural Information Processing Systems: vol. 35. 2022: 9881-9893.
- [41] ZENG A, CHEN M, ZHANG L, et al. Are transformers effective for time series forecasting?[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 37: 9. 2023: 11121-11128.
- [42] ZHOU T, MA Z, WEN Q, et al. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting[C]//International Conference on Machine Learning. 2022: 27268-27286.
- [43] DAS A, KONG W, LEACH A, et al. Long-term forecasting with TiDE: Time-series dense encoder[J]. ArXiv preprint arXiv:2304.08424, 2023.
- [44] YI K, ZHANG Q, FAN W, et al. Frequency-domain MLPs are more effective learners in time series forecasting[C]//Advances in Neural Information Processing Systems: vol. 36. 2024.
- [45] ZHANG T, ZHANG Y, CAO W, et al. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures[J]. ArXiv preprint arXiv:2207.01186, 2022.
- [46] CHEN S A, LI C L, YODER N, et al. Tsmixer: An all-mlp architecture for time series forecasting[J]. ArXiv preprint arXiv:2303.06053, 2023.
- [47] TOLSTIKHIN I O, HOULSBY N, KOLESNIKOV A, et al. Mlp-mixer: An all-mlp architecture for vision[C]//Advances in Neural Information Processing Systems: vol. 34. 2021: 24261-24272.
- [48] OLIVARES K G, CHALLU C, MARCJASZ G, et al. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx[J]. International Journal of Forecasting, 2023, 39(2): 884-900.
- [49] BAI S, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. ArXiv preprint arXiv: 1803.01271, 2018.

- [50] LAI G, CHANG W C, YANG Y, et al. Modeling long- and short-term temporal patterns with deep neural networks[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018.
- [51] LIU M, ZENG A, CHEN M, et al. SCINet: Time series modeling and forecasting with sample convolution and interaction[C]//Advances in Neural Information Processing Systems: vol. 35. 2022: 5816-5828.
- [52] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.
- [53] SHUMAN D I, NARANG S K, FROSSARD P, et al. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains[J]. IEEE Signal Processing Magazine, 2013, 30(3): 83-98.
- [54] ZHANG X, CAO R, ZHANG Z, et al. Crowd flow forecasting with multi-graph neural networks[C]//International Joint Conference on Neural Networks. 2020: 1-7.
- [55] GENG X, LI Y, WANG L, et al. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 3656-3663.
- [56] FANG Z, LONG Q, SONG G, et al. Spatial-temporal graph ode networks for traffic flow forecasting[C]//Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2021: 364-373.
- [57] LI M, ZHU Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 35: 5. 2021: 4189-4196.
- [58] BERNDT D J, CLIFFORD J. Using dynamic time warping to find patterns in time series[C]//Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. 1994: 359-370.

- [59] BAI L, YAO L, LI C, et al. Adaptive graph convolutional recurrent network for traffic forecasting[C] // Advances in Neural Information Processing Systems: vol. 33. 2020: 17804-17815.
- [60] CAO D, WANG Y, DUAN J, et al. Spectral temporal graph neural network for multivariate time-series forecasting[C] // Advances in Neural Information Processing Systems: vol. 33. 2020: 17766-17778.
- [61] JIANG J, HAN C, ZHAO W X, et al. PDFFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction[J]. ArXiv preprint arXiv:2301.07945, 2023.
- [62] YU F, KOLTUN V, FUNKHOUSER T. Dilated residual networks[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 636-644.
- [63] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.
- [64] HUBER P J. Robust estimation of a location parameter[J]. The Annals of Mathematical Statistics, 1964: 73-101.
- [65] CDC. Illness[Z]. <https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>. 2021.
- [66] Wetterstation. Weather[Z]. <https://www.bgc-jena.mpg.de/wetter/>. 2021.
- [67] TRINDADE A. Electricity[Z]. <https://doi.org/10.24432/C58C86>. 2015.
- [68] California. Traffic[Z]. <https://pems.dot.ca.gov/>. 2017.
- [69] KINGMA D P, BA J. Adam: A Method for stochastic optimization[C] // International Conference on Learning Representations. 2015.
- [70] WOO G, LIU C, SAHOO D, et al. Etsformer: Exponential smoothing transformers for time-series forecasting[J]. ArXiv preprint arXiv:2202.01381, 2022.

- [71] HAN L, YE H J, ZHAN D C. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting[J]. ArXiv preprint arXiv:2304.05206, 2023.
- [72] LIU H, DONG Z, JIANG R, et al. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting[C] // Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 4125-4129.
- [73] JAGADISH H V, GEHRKE J, LABRINIDIS A, et al. Big data and its technical challenges[J]. Communications of the ACM, 2014, 57(7): 86-94.
- [74] CHEN C, PETTY K, SKABARDONIS A, et al. Freeway performance measurement system: Mining loop detector data[J]. Transportation Research Record, 2001, 1748(1): 96-102.
- [75] XU M, DAI W, LIU C, et al. Spatial-temporal transformer networks for traffic flow forecasting[J]. ArXiv preprint arXiv:2001.02908, 2020.
- [76] SHANG C, CHEN J, BI J. Discrete graph structure learning for forecasting multiple time series[J]. ArXiv preprint arXiv:2101.06861, 2021.
- [77] DENG J, CHEN X, JIANG R, et al. St-norm: Spatial and temporal normalization for multi-variate time series forecasting[C] // Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2021: 269-278.

致 谢

三年光阴如白驹过隙，转瞬即逝。这段时间里，我经历了许多，有收获，也有遗憾；有欣喜，也有失落。在前进的道路上，我遇到了不少困难，但同时也结识了许多有趣的人和事，从中汲取了不少经验，增长了不少才干。在此，我由衷地感谢所有曾经帮助和支持过我的老师、同学以及我的家人。

首先，我要由衷感谢我的导师申富饶老师。尽管申老师每周都有教学任务，但他总是能够抽出时间与我们每个人单独讨论。在每次讨论中，老师总是耐心地指导我，帮助解决我在科研中遇到的困难，并强调科研应该紧密结合实际需求，具有实际应用价值。我对老师的激励和鼓励深表感激，我会牢记老师的教导，并在未来的工作中努力将工作落实到实处，真正做有价值的工作。同时，我还要感谢赵健老师在我撰写英文论文过程中的帮助，他细致地修改了论文中的语法和逻辑错误，并提供了详细的修改建议。

然后，感谢 RINC 实验组的各位同学们，大家庭的和谐氛围让我们在遇到问题时总能得到及时的帮助和支持。特别要感谢韩峰学长和郭苏涵学姐在科研上对我的指导和帮助。同时，也感谢同级的小伙伴们，我们一起度过了研究生三年，共同学习、共同进步，一同探索科研的乐趣。

其次，感谢我身边的家人和朋友们，他们在我无助和迷茫的时候，总是无条件地给予我支持和鼓励，赠予我前行的力量。他们的陪伴和关心让我感到无比幸运和温暖，是我不断奋斗的动力来源。在人生的旅途中，有了他们的支持，我相信无论遇到什么困难，都能勇往直前，勇敢面对。

最后，不知未来如何，但愿永葆初心。念念不忘，必有回响。

简历与科研成果

基本信息

李菲菲，女，汉族，2000年3月出生，山西吕梁人。

教育背景

2021年9月—2024年6月 南京大学计算机科学与技术系 硕士

2017年9月—2021年6月 南京大学计算机科学与技术系 本科

攻读硕士学位期间完成的学术成果

- Feifei Li, Suhan Guo, Feng Han, Jian Zhao, Furaio Shen. "Multi-Scale Dilated Convolution Network for Long-Term Time Series Forecasting". arXiv preprint arXiv: 2405.05499, 2024.
- 申富饶, 李菲菲, 郭苏涵, 韩峰, 赵健. 《一种基于多尺度空洞卷积的时间序列预测方法》。(专利申请号 202410110896.2)

攻读硕士学位期间参与的科研课题

- 科技部重大项目“基于神经可塑性的脉冲网络高效学习机制与类脑智能系统”(参与课题年限 2021年9月—2024年6月), 负责时间序列预测相关问题的研究。
- 国家自然科学基金面上项目“面向增量式无监督学习的新型神经网络研究”(参与课题年限 2023年1月—2024年6月), 负责神经网络模型相关研究。