目 录

[0 引言 1](#_Toc199018231)

[1 相关工作概述 2](#_Toc199018232)

[1.1 城市细粒度流量推断 2](#_Toc199018233)

[1.2 城市流量数据预测 3](#_Toc199018234)

[1.3 城市流量数据超分辨率 3](#_Toc199018235)

[1.4 交通流量预测 3](#_Toc199018236)

[2 基于时空对比自监督的城市细粒度流量预测模型 4](#_Toc199018237)

[2.1 引言 4](#_Toc199018238)

[2.2 方法描述 5](#_Toc199018239)

[2.2.1 定义 6](#_Toc199018240)

[2.2.2 空间自监督学习 7](#_Toc199018241)

[2.2.3 时间自监督学习 8](#_Toc199018242)

[2.2.4 外部因素融合 10](#_Toc199018243)

[2.2.5 微调 UrbanSTC 10](#_Toc199018244)

[2.3 实验与分析 11](#_Toc199018245)

[2.3.1 实验设置 11](#_Toc199018246)

[2.3.1.1 数据集 11](#_Toc199018247)

[2.3.1.2 训练细节与超参数 11](#_Toc199018248)

[2.3.2 对比最新方法 11](#_Toc199018249)

[2.3.3 消融实验 15](#_Toc199018250)

[2.3.4 端到端和两阶段对比 17](#_Toc199018251)

[2.3.5 时间对比采样分析 17](#_Toc199018252)

[2.3.6 融合外部因素研究 18](#_Toc199018253)

[2.3.7 配置和参数分析 18](#_Toc199018254)

[2.3.8 城市细粒度流量推断可视化 21](#_Toc199018255)

[2.4 本章小节 21](#_Toc199018256)

[3 基于时空吸引学习的不可观测数据下细粒度城市流量推理 22](#_Toc199018257)

[3.1 引言 22](#_Toc199018258)

[3.2 方法描述 23](#_Toc199018259)

[3.2.1 符号说明与问题定义 23](#_Toc199018260)

[3.2.2 城市流量数据补全网络 23](#_Toc199018261)

[3.2.3 城市流量数据补全网络 25](#_Toc199018262)

[3.2.4 最终目标函数 26](#_Toc199018263)

[3.3 实验与分析 26](#_Toc199018264)

[3.3.1 实验设置 26](#_Toc199018265)

[3.3.1.1 数据集 26](#_Toc199018266)

[3.3.1.2 对比方法 27](#_Toc199018267)

[3.3.1.3 训练细节与超参数 27](#_Toc199018268)

[3.3.2 与基线方法的对比分析 27](#_Toc199018269)

[3.3.3 消融分析 29](#_Toc199018270)

[3.3.4 参数分析 30](#_Toc199018271)

[4 基于自适应时空上下文学习的交通流量预测模型 30](#_Toc199018272)

[4.1 引言 30](#_Toc199018273)

[4.2 方法描述 31](#_Toc199018274)

[4.3 实验与分析 35](#_Toc199018275)

[4.3.1 实验设置 35](#_Toc199018276)

[4.3.2 方法对比 36](#_Toc199018277)

[4.3.3 消融实验 37](#_Toc199018278)

[4.3.4 参数分析 38](#_Toc199018279)

[4.3.5 鲁棒性分析 39](#_Toc199018280)

[4.4 本章小结 41](#_Toc199018281)

[5 结论（结论和建议） 41](#_Toc199018282)

[参考文献 42](#_Toc199018283)

插图清单

[图 1 整体研究路线 1](#_Toc199018284)

[图 2 城市细粒度流量推断 5](#_Toc199018285)

[图 3 UrbanSTC整体架构 6](#_Toc199018286)

[图 4 空间超分辨率推理网络预训练 8](#_Toc199018287)

[图 5 外部信息融合 10](#_Toc199018288)

[图 6 融实验可视化 16](#_Toc199018289)

[图 7 P1数据集上硬采样和权重采样性能比较 17](#_Toc199018290)

[图 8 不同通道数实验结果 19](#_Toc199018291)

[图 9 阈值λ影响曲线 19](#_Toc199018292)

[图 10 模型参数研究 20](#_Toc199018293)

[图 11 数据集上不同方法之间的推理误差可视化 21](#_Toc199018294)

[图 12 两种粒度层次的城市流量图示意图 23](#_Toc199018295)

[图 13 UrbanSTA 框架结构 24](#_Toc199018296)

[图 14 （a）STA 模块在任务中的有效性分析；（b）FIN 模块在任务中的有效性分析 27](#_Toc199018297)

[图 15 在 TaxiBJ 数据集（缺失率 20%，上采样因子 4）上的不同超参数设置下模型性能对比 29](#_Toc199018298)

[图 16 交通预测框架 31](#_Toc199018299)

[图 17 ASTCL模型框架图 33](#_Toc199018300)

[图 18不同图的构建方式 34](#_Toc199018301)

[图 19 时空联合学习分析 38](#_Toc199018302)

[图 20 参数分析 39](#_Toc199018303)

附表清单

[表 1 P1不同比例训练数据结果 12](#_Toc199018304)

[表 2 P2不同比例训练数据结果 13](#_Toc199018305)

[表 3 P3不同比例训练数据结果 13](#_Toc199018306)

[表 4 P4不同比例训练数据结果 14](#_Toc199018307)

[表 5 BikeNYC不同比例训练数据结果 15](#_Toc199018308)

[表 6 消融实验 16](#_Toc199018309)

[表 7 端到端和两阶段对比 17](#_Toc199018310)

[表 8 TaxiBJ外部因素实验结果1 18](#_Toc199018311)

[表 9 TaxiBJ外部因素实验结果2 18](#_Toc199018312)

[表 10 P1数据集上评估模型的效率 20](#_Toc199018313)

[表 11 数据集统计信息 26](#_Toc199018314)

[表 12 TaxiBJ 数据集上的对比实验结果 28](#_Toc199018315)

[表 13 BikeNYC 数据集上的对比实验结果 28](#_Toc199018316)

[表 14 消融实验结果 29](#_Toc199018317)

[表 15 交通流量预测实验结果 36](#_Toc199018318)

[表 16 交通流量预测实验结果 36](#_Toc199018319)

[表 17 消融实验 38](#_Toc199018320)

[表 18 跨城市预测实验结果 39](#_Toc199018321)

[表 19 跨域预测实验结果 40](#_Toc199018322)

# 0 引言

随着人工智能技术的快速发展，智慧城市建设已成为全球城市化进程的重要方向。智慧交通系统作为一种融合通信技术、人工智能与物联网技术的新型治理范式应运而生。其核心目标是通过构建数据驱动的智能决策体系，实现交通资源的最优配置与动态调控。随着信息技术的快速发展，尤其是传感器技术和大数据处理能力的显著提升，交通数据的种类和来源日益多样化，为智慧交通系统的部署和优化提供了坚实的数据基础。

人工智能与深度学习技术的持续进步，为智慧交通系统赋予了全新的认知与决策能力。城市交通预测则是智慧交通系统的核心构成部分，它通过对海量历史数据的深度分析与实时信息的整合处理，运用多种数据分析方法与模型，预测未来交通流量的变化趋势、速度的波动情况以及拥堵状况的发生概率等，从而为交通管理决策和出行者的出行决策提供有力的数据支持和决策依据。近年来，城市交通预测问题因其广泛的应用价值受到学术界和工业界的关注。然而，城市交通系统具有显著的动态性和时空异质性，如何有效挖掘历史数据中的时空关联并提供可靠、准确的预测结果，成为当前亟待解决的关键问题。

然而，当前研究仍面临三大关键挑战：、资源受限环境下的高效模型训练，传感器数据缺失条件下的高精度预测以及动态外部干扰下的鲁棒时空建模。针对这些问题，本研究系统梳理了现有研究的局限性，并提出了一系列创新性解决方案，以推动智慧交通预测技术的进一步发展。本研究的整体思路如图1所示。

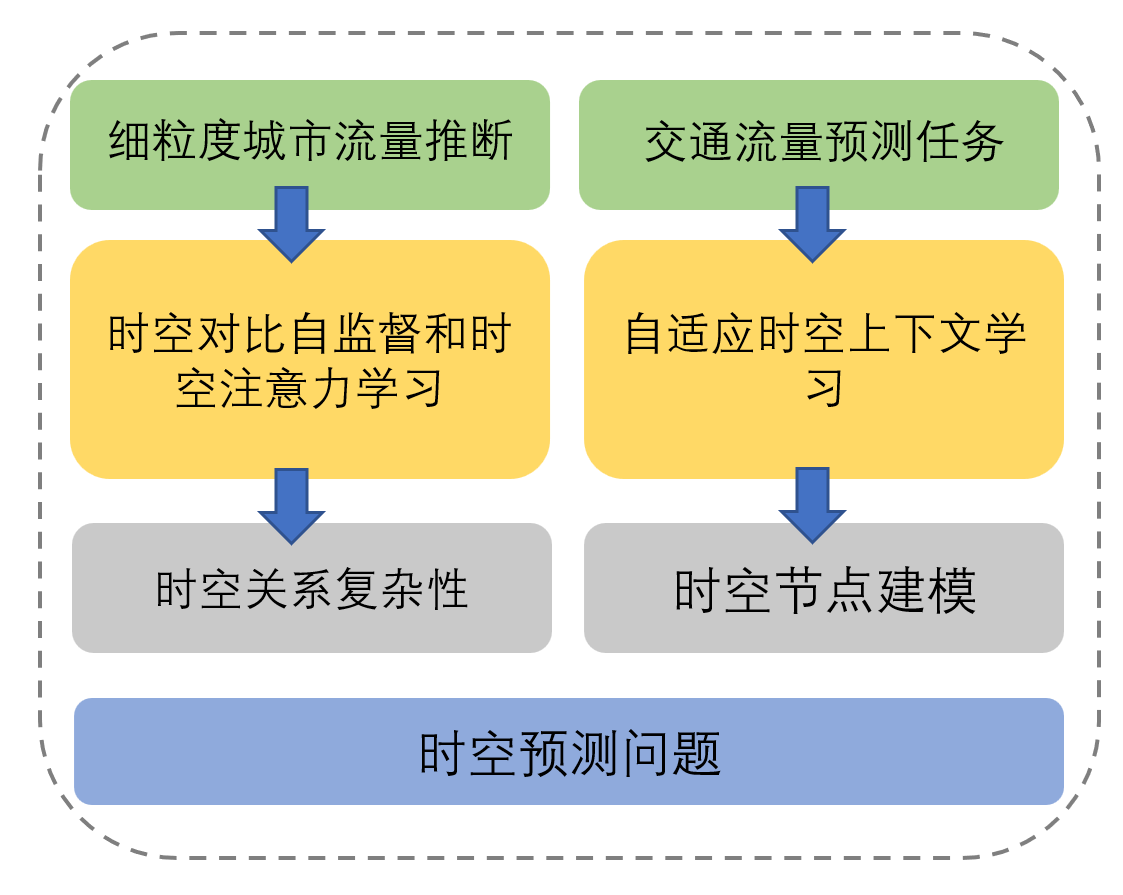


图 1 整体研究路线

细粒度流量推断通常需要复杂的深度学习模型，而现实场景中（如边缘计算设备）往往存在计算资源受限的问题。现有方法在有限训练数据或低算力环境下性能显著下降，亟需轻量化且高效的学习策略。传统交通流量监测依赖密集部署的传感器网络，导致高昂的硬件成本和能源消耗。尽管基于历史数据的补全算法能够部分缓解这一问题，但在高缺失率场景下（如传感器稀疏布局），现有模型的预测精度仍难以满足实际需求。并且交通系统受天气、突发事件等外部因素影响，时空模式呈现高度动态性。现有模型大多假设数据分布稳定，难以适应实时变化的交通状态，导致预测鲁棒性不足。研究难点总结如下：

1. 低资源场景的模型优化：如何通过自监督学习等策略，在有限计算资源下提升细粒度预测效率？
2. 高缺失率下的时空建模：如何在传感器稀疏布局的情况下，利用不完整的历史数据实现高精度补全？
3. 动态时空模式的鲁棒建模：如何设计自适应框架，以应对交通网络中节点关联的动态变化和外部干扰？

针对智慧交通预测中的一系列问题，本研究结合时空关系学习与细粒度推断方法，设计三种算法解决上述城市流量预测问题，分别对应本研究 2 至 4 章。在第 1 章中，对现有相关工作进行了论述。在第 5章中，对本研究工作进行总结，并展望未来工作。

在第2章中，为提升资源受限环境下的训练效率，设计了双重自监督任务，区域对比学习：通过拉近相似职能区域的表征距离，增强空间特征判别能力；下采样重构任务：将粗粒度流量图下采样后重构为原始分辨率，降低任务难度并提升模型泛化性。此外，提出硬采样和权重采样策略，优化正负样本选择效率。

在第3章中，本研究针对传感器稀疏部署场景，提出了一种基于时空图神经网络的多缺失率流量预测方法。通过建模历史时间片段的时空依赖性，该算法能够有效填补缺失区域的流量值，显著降低硬件部署成本。

在第4章中，针对动态交通环境，提出了一种节点级预测框架，其核心创新包括：自适应时空上下文构建：为目标节点融合历史时序特征和空间邻域信息，通过联合卷积和注意力机制提取局部-全局协同特征；动态图神经网络：基于特征相似性实时生成序列图，捕捉节点间动态关联。

# 1 相关工作概述

## 城市细粒度流量推断

城市细粒度流量推断旨在根据观测的粗粒度流量数据来推断城市中的细粒度流量, 它也是图像超分辨率任务(SISR)在交通领域的迁移[11,12]。Liang等人[1]首先提岀了一个名为UrbanFM的神经网络来解决城市细粒度流量推断任务，它主要借助残差神经网络架构[13]和该任务特有的结构性约束。UrbanFM设计了一个M^2-Normalization层，它可以控 制粗细粒度流量图对应区域的概率分布值。Shen等人[14]网基于超分辨率方案设计了一个受天气影响的城市细粒度流量预测器(WFRFP)模型，WFRFP探索天气状况与流量分布之间的关系，并根据相应的粗粒度流量图缩小预测区域的范围。然而，所提出的网络架构严重依赖于经验堆叠的深度神经网络。为了解决网络层数过深这个问题，Chen等人从微分方程[15]和神经网络相结合的角度介绍了深度神经网络模型，他们将神经网络的训练 和预测视为常微分方程问题。由于神经常微分方程(NODE)的引入，Zhou等人发现NODE 可以作为核心模块来解决城市细粒度流量推断问题，提出了一种基于ODE结构的城市细粒度流量推断(FODE)[2]。它可以解决前一种方法的数值不稳定问题并且不会造成额外的内存计算成本，FODE的关键思想是在每个ODE块中加入仿射耦合层来避免梯度不准确的问题。FODE和UrbanFM的区别在于前者使用的是ODE块而不是残差神经单元，尽管上述模型取得了成功，但现有技术大多依赖于大量参数和复杂的神经网络架构。

## 1.2 城市流量数据预测

城市流量数据预测在城市计算领域中占据重要地位，研究方法多种多样[9,10,11]。在深度学习方法中，DeepST 模型[12]被提出，用于建模时间与空间特征，并能同时提取外部因素的影响。另一典型模型 ST-ResNet [13]则通过残差单元与特征融合组件以学习更复杂的时空特征。

与传统的城市流量数据预测不同，本文研究的是在粗粒度观测数据存在缺失情况下，推理细粒度城市流量的任务。

## 1.3 城市流量数据超分辨率

深度学习方法是当前推理细粒度城市流量的主流手段[14]。UrbanFM 模型[15]通过引入外部因子融合网络以提取外部特征，从而预测城市交通流量。为进一步提升 UrbanFM 的推理性能，Ouyang 等人[16]提出了 UrbanPy 模型，分为两个部分：推理网络用于生成细粒度流量分布，通用融合子网络用于增强整体性能。

然而，上述研究多基于理想数据，现实中普遍存在缺失问题。Li 等[6]首次针对该问题提出了 MT-CSR 多任务网络模型，设计了 CMPNet 补全网络，综合考虑局部空间依赖性与全局 POI 相似性以填补粗粒度流量图的缺失；此外还提出了 SRNet 超分辨率网络，用以捕捉粗细粒度数据之间的复杂关联。

## 1.4 交通流量预测

交通流量预测是智慧交通系统中的一个重要研究方向，旨在基于历史数据对未来一段时间的交通信息进行预测。基于图学习的相关方法近年来被广泛应用于交通预测领域。DCRNN[25] 通过扩散卷积网络建模空间关系，并利用循环神经网络捕捉时间依赖性，从而实现对交通数据的时空特性进行有效表征。一些研究[30-32] 使用节点间的欧氏距离来度量空间相关性，这种方法虽然能够直观地反映交通节点在地理空间中的分布关系，但仍存在局限性。因此，研究[33-38] 利用节点特征去构建动态交通网络，并通过图神经网络 (GNNs) 在动态交通图上进行建模，从而更加灵活地捕获时空关系。在深层 GNN中，多次迭代的消息传递会导致节点特征趋同，导致过平滑现象[39-40] 。为解决这些问题，基于注意力机制的交通预测模型[41-46] 受到研究者的广泛关注，显著提升了模型对远距离时空依赖关系的建模能力。

同时，研究者们探索了多层感知机（MLP）在交通预测领域的应用[47-49] ，旨在降低模型复杂度并减少训练开销。类似地，基于神经架构搜索的交通预测模型[50-53] 通过自动化搜索最优模型架构，进一步节省了人工设计成本，同时保证了预测结果的准确性。近年来，面向复杂交通场景的预测问题（如少样本交通预测和跨城市交通预测）也受到广泛关注，研究者致力于提升模型在复杂交通场景的预测鲁棒性和准确性。其中，少样本交通预测问题聚焦于在数据稀缺的条件下实现准确的流量预测，常通过迁移学习[54-56] 和元学习[57] 等技术来提升模型的泛化能力。跨城市预测模型[58-62] 则关注如何将已训练的模型迁移到新的城市场景，其核心挑战在于如何有效应对不同城市间交通模式的差异性。由此可见，城市交通预测作为智慧城市建设的核心环节，已吸引了众多研究者的广泛关注。随着交通数据的多样性和复杂性不断增加，构建鲁棒性强、泛化能力优异的交通预测模型已成为当前的重点研究方法。

# 2 基于时空对比自监督的城市细粒度流量预测模型

## 2.1 引言

随着城市化的发展趋势，智能交通系统己经成为智慧城市领域的重要组成部分之一。 城市规划者和管理者的一个重要需求对细粒度的城市流量进行监控，并在出现交通拥堵、公共风险等情况时发出警告[1-4]。观测细粒度的城市流量需要在城市中部署大量监控设备，尽管带来了一定便利，但同时也消耗了大量电力资源。例如，管理人员在运营监测设备方面的采购、人力和维护费用非常高，这增加了政府的财政压力。因此，从粗粒度的流量图中推断出细粒度的流量变化显得至关重要。为了解决以上问题，研究者们提出了城市细粒度流量推断(FUFI)，通过观测到的粗粒度流量数据来对细粒度流量进行估计。图2(a)和2(b)展示了同一城市区域，但分割尺度不同，左侧子图为粗粒度地图（32x32），右侧子图为细粒度地图（64x64）。FUFI的目标是基于粗粒度数据对细粒度流量图进行准确预测。FUFI也被视为图像超分辨率的一个变种，但它具有独特的结构约束，即细粒度区域的流量总和严格等于对应的超区域的流量总和。

尽管许多研究者[1,2,4]在FUFI问题中取得了不错的进展，但他们所提出的模型大多需要复杂的神经网络结构、大量的参数和长期的训练时间。自监督学习在处理此类问题具有明显的优势，在计算机视觉领域[31-34]和自然语言处理领域[35,36]表现良好。这些模型在大量未标记数据或者少量标记数据中表现出优异的表示学习能力。

迄今为止，现有的对比自监督学习策略并不能直接用于FUFI问题，具体来说存在如下几个挑战：(1)空间对比自监督学习：本质上，一个区域的流量主要受周围区域的影响。然而，当两个区域具有相同的城市功能(例如，商业中心、住宅区域和旅游区域)时。它们具有相似的流量分布状况。如图2(a)所示，即使区域A和B之间相隔距离很远，它们也具有相似的流量特性。以前的FUFI研究主要关注相邻区域的相关性，而忽略了语义相似性。 此外，现有的对比自监督常使用整个流量图来进行对比，但是忽略了区域层面的对比情况。因此，如何设计一种有效的空间对比学习方法是需要解决的主要问题。(2)时间对比自监督学习：现有的FUFI方法旨在从当前时刻的粗粒度流量图来预测相 应的细粒度流量图。从自监督的角度分析，这种方法并没有充分利用时间信息。城市细粒度流量预测不仅是从当前时刻推断出来，而且还受到其他时刻的影响。此外，一个区域整体交通流量变化具有很强的周期属性，这表明连续时间语义相似有助于流量推断，未能使用此信息将会导致性能不佳。(3)外部因素：外部因素在FUFI[1,4]中扮演着至关重要的角色，例如，在通勤高峰时段， 主干道的车流量远大于其他时间段。当恶劣天气发生时，人们更强倾向于在室内而不是室外。各种外部因素对现实世界的细粒度推断有着不同影响。

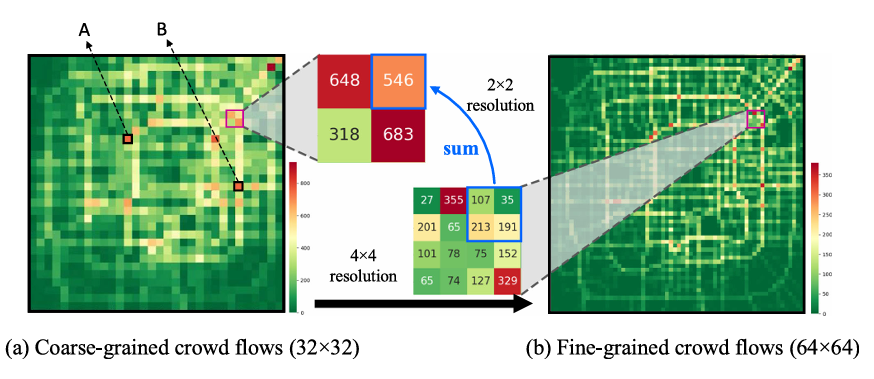


图 2 城市细粒度流量推断

本研究提出了一个名为UrbanSTC的时空对比自监督方法。该方法创新点总结如下：首先，在空间方面设计区域对比和空间超分辨推理网络，根据周围属性和语义相似性来识别区域之间的空间底层关系。区域对比侧重基于内在空间特征探索区域层面流量之间的相似性；空间推理削减原本城市细粒度推断的难度并且学习同比例的推理方法。其次，提出硬采样和权重采样两种时间对比采样方法。对于给定的一组三元流量图，它能够将锚点与正样本之间的距离拉近，同时使锚点和负样本之间的距离远离。设计一个结合三个预训练编码器的微调网络来进行城市细粒度流量预测。最后，将外部因素纳入微调的UrbanSTC网络，从事件和天气条件中提取有用信息。

## 2.2 方法描述

本研究提出了一个名为UrbanSTC的方法，用于挖掘城市规划过程中的细粒度流量信息。UrbanSTC主要包含三个部分：空间自监督、时间自监督和微调阶段。首先分别通过基于区域对比和空间推理的空间自监督和基于时间对比的时间自监督模块进行预训练，然后将以上三个预训练的编码器参数保存到最终网络进行微调，完成城市细粒度流量预测，整体架构如图3所示。



图 3 UrbanSTC整体架构

### 2.2.1 定义

本节介绍一些与细粒度流量预测基本符号和概念。

#### 2.2.1.1 网格流量图

给定时刻,假设是一个城市的流量图，根据经纬度将该流量图划分为的区域，如图2(a)所示，每个区域表示观测到的流量。

#### 2.2.1.2 超区域与子区域

图2(a)和(b)是同一个城市区域，但有两种不同的划分尺度，左子图是粗粒度流量图(32x32),右子图表示细粒度流量图(64x64)。表示控制粗粒度和细粒度图之间分辨率变化的比例因子。图1表示M=2时的示例。本研究使用超区域和子区域分别定义较大的网格及其较小区域[1-4]。

#### 2.2.1.3 结构性约束

细粒度子区域的流量之和严格等于对应的超区域。

其并且。

#### 2.2.1.4 城市细粒度流量推断

给定一个粗粒度流量图和上采样因子，本项任务的目标是在结构性约束的条件下推断出细粒度流量图。

### 2.2.2 空间自监督学习

城市流量数据具有典型的空间特征。受自监督学习的启发，本研究提供了两种空间视角的自监督任务：区域对比和空间超分辨率推理网络。

区域级别对比预训练：区域对比自监督致力于挖掘区域层面的流量关系。在任何时刻，粗粒度流量图中有许多区域具有相似或者不相似的流量特性。在图3中，浅蓝色区域(Reg)描绘了区域级别对比学习的示例。假设图中黑色选框是一个锚点区域，具有红色和蓝色选框的区域可以通过计算与的语义距离分别被视为正样本和负样本, 如公式(2)和(3)所示：

其中是流量图中的候选区域，是区分正负样本的阈值。由于区域之间的语义 距离不同，本方法希望在其高级表示中保留这些属性，即与正区域样本足够 近,让所有负区域样本远离，其中K1和K2分别是正负区域样本的数量。

对于粗粒度流量图，首先通过非线性编码器将其投影到低级隐藏特征,本方法将这个组件命名为区域级别编码器以便用于之后的微调阶段。然后，被用于标准化操作[45]并改变维度得,最后使用个隐藏单 元的感知机生成高级语义特征。对于之前的对比损失函数，比如InfoNCE只能严格选择一个样例作为正样本[40,41]。本方法将来自粗粒度流量图中的部分区域视为正样本，其余的视为负样本，这与[46]中的策略类似。对于每个，随机选择区域锚点，并通过基于预先设定的阈值计算欧几里得距离来区域正负样本，本研究对比损失函数表示为：

其中，，并且表示两个特征之间的相似度函数(内积)。通过这种方法，正样本之间特征相互拉近，而与负样本特征远离。

空间超分辨推理网络预训练：给定一个粗粒度流量图和上采样因子, FUFI目标是在结构性约束的前提下学习一个推理网络来预测细粒度流量图.其中最重要的机制是如何将一个粗粒度区域拆分成个细粒度单元: 。为了模拟这个过程，本方法在预训练任务中设计了空间推理网络，如图4所示。

本方法的意图是使用更粗粒度的映射模式来作为辅助任务。详细来说，首先基于粗粒度流量图和上采样因子得到一个下采样粗粒度流量图，其中下采样粒度流量图的每个区域等于中对应的流量之和。然后本方法构造一个空间超分辨率推理网络：从推断。这个辅助任务能够学习的模式并且有利于提高周围流量的推理能力。

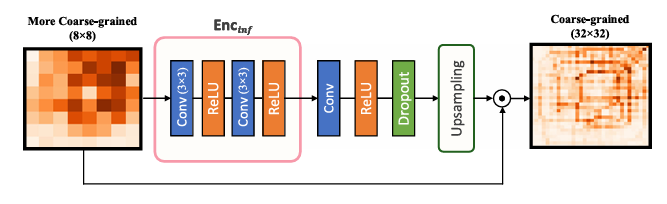


图 4 空间超分辨率推理网络预训练

对于任意下采样粗粒度流量图，首先使用两层带有3x3卷积对其进行编码，并 且每个卷积操作后紧跟非线性变化，如图4所示。两个卷积层作为特征学习网络将映射到低级隐藏特征，这个组件被命名为空间超分辨率编码器。然后利用之前FUFI方法在其网络末端进行分布式上采样[1]：使用-Normalization网络进行结构性约束。最终该模块的损失函数由MSE进行计算

其中表示推理网络中的所有可学习参数，这个网络结构和函数类似最终微调 UrbanSTC，详情请参阅微调UrbanSTC部分。

### 2.2.3 时间自监督学习

现有的FUFI研究仅基于其粗粒度流量图来推断细粒度流量图，而忽略了不同时刻 相似的流量也会有助于推断。在此，设计了一个时间对比自监督网络(TCS)来提取时间 维度上的相似信息。对于任何时刻t,本方法可以得到一个锚点；,然后通过识别样本 之间的相似性来收集它的正样本集和负样本集，其中和是选择的正负时间样本数量。

TCS构建了一个自监督辅助任务，缩小了锚点和正样本之间的编码特征并使负样本 远离。TCS编码器具有与空间超分辨率推理网络相似的结构。它将粗粒度流量 图；投影到低级隐藏特征,然后采用归一化和全局平均池化操作。最后使用带有ReLu激活函数的多层感知机(MLP)对其进行非线性变换。将编码器特征转变为高级语义特征。如图3浅黄色区域，接下来将介绍如何选取它们。

硬采样：首先使用一种直接的方法来选择与锚点最近和最远的样本作为其正负样本对。样本之间的距离通过欧几里得距离进行计算:

其中是当前时刻的粗粒度流量图，是其他时刻流量图。如图3中时间自监 督模块，有绿、红和蓝色三种类型的样本，它们分别代表锚点、正样本和负样本，硬釆样方法的目的是选择与当前锚点最近的（正）样本并找到最远（负）样本。值得注意的是, 时间对比方法在视频处理中被广泛应用，例如[37,38],它们只是选择一个时间窗内作为正样本，将其余的视频帧放入到负样本集中。然而交通流量预测问题[47]具有高频的周期特性，因此本方法选择计算锚点和所有训练样本之间的距离。

|  |
| --- |
| 算法2-1：权重采样 |
| 输入：原始的粗粒度流量数据 |
| 输出：三元组数据 |
| 1. for do |
| 2. 建立大顶堆和小顶堆； |
| 3. for do |
| 4. if then |
| 5. 计算x和y之间的欧式距离； |
| 6. 调整大顶堆和小顶堆； |
| 7. 分别选择Top-K个正负样本； |
| 8. 通过公式（7）得到； |
| 9. 通过公式（8）得到； |

重采样：考虑到硬采样不能充分利用所有时间样本之间的相关性，在本节中提出一种权重采样的方法。具体来说，使用加权组合的方式选择Top-K正负样本：

其中表示锚点与第个样本之间的欧式距离。具体的算法细节如算法2-1所示。

TCS使用Triplet loss[48]来优化预训练模型。给以定一个三元组数据, Triplet loss确保（锚点）和（正样本）之间在潜在语义空间上彼此拉近并且远离 （负样本）。将这个损失函数定义为：

其中是可学习的非线性映射函数，是超参数。值得注意得是，Triplet约束可 以更加灵活地适用不用级别的类内方差，从而保证了各个时刻之间的差异。

### 2.2.4 外部因素融合

外部因素（例如温度、风速、天气和节假日）影响次区域的流量分布。例如：人们更倾向于在节假日走出办公区。当阴天暴雨来临时，人们更喜欢待在室内而不是室外。 因此，本工作还需要考虑到这些外部因素。

将外部因素分为连续特征和离散特征。其中，包括温度和风速在内的连续特征可直接拼接成向量。离散特征包括：时刻、日期、节假日和天气条件，本工作使用UrbanFM[1]中的方法来初始化外部信息，分类特征通过嵌入层转化为低维向量并且相互拼接成 , 最终将两个向量进行信息融合()。

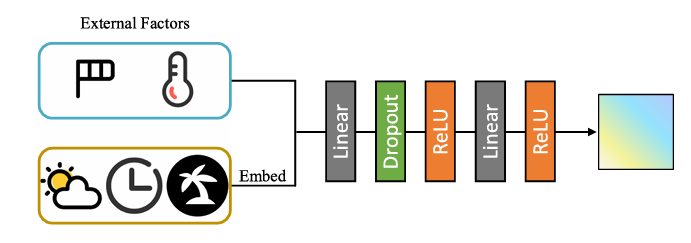


图 5 外部信息融合

注：外部因素分为连续特征（蓝色区域）和离散特征（黄色区域）。

如图5所示。使用具有非线性变换的多层感知机提供外部信息,本方法通过非线性变换，将不同的外部因素收敛到一个隐藏状态将其视为流量图的偏差， 在前面的部分中，本方法只使用了没有外部信息的粗粒度流量图进行预训练。最后在微 调阶段对初始的流量图和外部信息进行张量相加作为模型输入.

### 2.2.5 微调 UrbanSTC

在上述预训练任务完成了三个编码器：区域级别对比编码器器、空间超分辨 率推理编码器器和时间对比编码器器。如图3所示，三个编码器用于下游任务微调，首先，本方法通过编码器组合三个低级隐藏特征：

其中Concat是张量拼接操作。Decoder是一个带有ReLU激活函数的卷积层，它被 用来解码三个低级隐藏特征。除此之外，本方法采用另一个卷积层和PixelShuffle层， 它们通过上采样因子重新排列特征并且增大特征尺寸，在PixelShuffle层的末端紧跟 ReLU激活函数。经过上述操作,本方法得到了一个尺寸扩大倍的特征。接下来，使用一个通道数为1的3x3卷积来得到细粒度分布的隐藏特征。由于FUFI问题的结构性约束，MSE损失函数不能较好的应用在此类问题上，参照 UrbanFM[8]和FODE[2]中的分布式上釆样，本研究选择一个-Normalization使子区域的流量总和等于其对应的超区域：

其中是中的第行和列，表示概率分布。-Normalization旨在学习从粗粒度流量图到细粒度流量图的概率映射。最终本方法通过来推断细粒度流量图。均方误差(MSE)作为损失函数：

其中表示UrbanSTC模型，表示该模型中所有可学习参数。

## 2.3 实验与分析

在本章节中进行了复杂的实验来验证模型的有效性。以下将由实验设置、出租车数据集的实验结果、自行车数据集的实验结果和实验分析组成。

### 2.3.1 实验设置

### 2.3.1.1 数据集

本研究在两个具有代表性的数据集上进行了实验验证：TaxiBJ和BikeNYC。

TaxiBj[1,4]，该数据是从北京出租车流量中以时间间隔为30分钟收集而来，包括四 个不同的时期：P1到P4。

BikeNYC(https://citibikenyc.com/system-data),该数据是从一个公开网站收集而来，该网站包含从2019年1月1日到 3月31日纽约市的自行车流量数据。本研究将城市区域划分为40x20的网络作为粗粒度 流量图，并定义80x40的细粒度流量图。

### 2.3.1.2 训练细节与超参数

本方法的模型和基线是基于TITAN Xp GPU的Pytorch框架进行训练，网络用一阶 动量为0.9和二阶动量为0.999的Adam优化器进行训练[53]，初始学习率设置为le-3并 且每过50个迭代周期后学习率减半，这样可以在收敛点附近更加平滑的搜索。min-batch 等于16；模型基本的通道数为128。

### 2.3.2 对比最新方法

#### 2.3.2.1 评价指标

本研究使用三个被广泛使用的指标来评估不同的方法：均方根误差(RMSE)、平均绝对误差(MAE)和平均绝对百分比误差(MAPE)。

其中是细粒度流量图的预测值；是真实值；是预测值的总数。

#### 2.3.2.2 实验结果

首先使用不同比例的训练数据在TaxiBJ上评估UrbanSTC和基线的性能。如表1到4所示，由于所有的实验结果方差均在0.000-0.002范围之内，因此结果展示省 略方差。对表格数据总结如下：

（1）UrbanSTC在整个时间跨度(P1-P4)上均优于所有基线方法。通过与最先进的方 法相比，UrbanSTC在100%训练数据的TaxiBJ-Pl上，其RMSE、MAE和MAPE平均 提高了 2.51%、3.80%和 10.51%。

（2）值得注意的是当训练数据减少时，UrbanSTC依然可以达到最好的效果。以 TaxiBJ-Pl(20%的训练数据)为例，UrbanSTC在RMSE、MAE和MAPE ±分别相对提高了 7.10%， 12.24%和 10.91%。

以上结果表明，UrbanSTC在训练资源有限的情况下具有自身优势，这与本工作最 初的动机是一致的，即时空对比自监督可以更好地学习数据特征表示并提高FUFI性能。 图像超分辨率方法SCN[52]在20%到60%的TaxiBJ数据上表现优于其他基线的RMSE, 而其MAE和MAPE上有所下降。这是由于SCN是依据均方误差损失函数的最先进图像超分辨率方法。然而，大多数图像超分辨率方法并不适用FUFI,因为它们在设计模型 时并没考虑结构约束。本方法与UrbanFM、UrbanPy和FODE相比，时空对比学习方法 的UrbanSTC可以具有更好的潜在表示和优越的性能。

表 1 P1不同比例训练数据结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | Pl(20%)  MAE | MAPE | RMSE | Pl (60%)  MAE | MAPE | RMSE | Pl(100%)  MAE | MAPE |
| MEAN | 20.918 | 12.019 | 4.469 | 20.918 | 12.019 | 4.469 | 20.918 | 12.019 | 4.469 |
| HA | 4.794 | 2.269 | 0.339 | 4.793 | 2.258 | 0.338 | 4.772 | 2.251 | 0.336 |
| SRCNN | 4.737 | 2.767 | 0.804 | 4.506 | 2.587 | 0.712 | 4.275 | 2.430 | 0.642 |
| ESPCNN | 4.552 | 2.583 | 0.682 | 4.264 | 2346 | 0.558 | 4.208 | 2.318 | 0.546 |
| DeepSD | 4.532 | 2.535 | 0.652 | 4.883 | 2.834 | 0.805 | 4.128 | 2.248 | 0.516 |
| VDSR | 4.546 | 2.556 | 0.669 | 4.198 | 2.279 | 0.527 | 4.054 | 2.186 | 0.485 |
| SRResNet | 4.734 | 2.800 | 0.844 | 4.276 | 2.437 | 0.654 | 4.079 | 2.291 | 0.580 |
| LapSRN | 4.676 | 2.738 | 0.801 | 4.309 | 2.432 | 0.635 | 4.083 | 2.255 | 0.542 |
| IMDN | 4.696 | 2.748 | 0.794 | 4.251 | 2.376 | 0.601 | 4.085 | 2.253 | 0.538 |
| SCN | 4.395 | 2.491 | 0.661 | 4.096 | 2.250 | 0.536 | 3.965 | 2.162 | 0.494 |
| UrbanFM | 4.560 | 2.343 | 0.398 | 4.195 | 2.140 | 0.350 | 4.042 | 2.062 | 0337 |
| UrbanPy | 4.665 | 2.471 | 0.547 | 4.112 | 2.077 | 0.349 | 3.944 | 1.998 | 0.333 |
| FODE | 4.476 | 2.304 | 0.391 | 4.161 | 2.116 | 0.344 | 4.002 | 2.044 | 0.336 |
| UrbanSTC | **4.083** | **2.022** | **0302** | **3.941** | **1.962** | **0.301** | **3.845** | **1.922** | **0.298** |
|  | +7.10% | +12.24% | +10.91% | +3.78% | +5.54% | +10.95% | +2.51% | +3.80% | +10.51% |

注：最好的结果是粗体，次好结果是下划线。

表 2 P2不同比例训练数据结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | P2(20%)  MAE | MAPE | RMSE | P2(60%)  MAE | MAPE | RMSE | P2(100%)  MAE | MAPE |
| MEAN | 26.729 | 15.350 | 5364 | 26.729 | 15.350 | 5.364 | 26.729 | 15.350 | 5364 |
| HA | 6.568 | 2.889 | 0.358 | 5.669 | 2.620 | 0338 | 5.512 | 2.576 | 0334 |
| SRCNN | 5.613 | 3.201 | 0.837 | 5.172 | 3.036 | 0.801 | 4.978 | 2.896 | 0.748 |
| ESPCNN | 5.461 | 3.062 | 0.738 | 4.934 | 2.779 | 0.637 | 5.072 | 2.957 | 0.749 |
| DeepSD | 5.412 | 2.991 | 0.704 | 4.716 | 2.585 | 0.546 | 4.909 | 2.738 | 0.625 |
| VDSR | 5.449 | 3.024 | 0.727 | 4.954 | 2.795 | 0.660 | 4.429 | 2.402 | 0.475 |
| SRResNet | 5.801 | 3.420 | 0.992 | 4.702 | 2.760 | 0.653 | 4.548 | 2.573 | 0.605 |
| LapSRN | 5.717 | 3.343 | 0.931 | 4.818 | 2.753 | 0.673 | 4.555 | 2.556 | 0.569 |
| IMDN | 5.790 | 3.547 | 1.123 | 4.710 | 2.792 | 0.755 | 4.476 | 2.608 | 0.661 |
| SCN | 5.222 | 2.932 | 0.721 | 4.487 | 2.475 | 0.528 | 4.336 | 2.381 | 0.490 |
| UrbanFM | 5.546 | 2.855 | 0.433 | 4.588 | 2.365 | 0336 | 4.414 | 2.272 | 0.318 |
| UrbanPy | 5.528 | 2.803 | 0.485 | 4.464 | 2.276 | 0.334 | 4.315 | 2.210 | 0.323 |
| FODE | 5.362 | 2.734 | 0.395 | 4.538 | 2.331 | 0.323 | 4.366 | 2.248 | 0.317 |
| UrbanSTC | **4.975** | **2.424** | **0.297** | **4.347** | **2.185** | **0.288** | **4.225** | **2.136** | **0.288** |
|  | +4.73%% | +11.34% | +17.04% | +2.62% | +4.00% | +10.84% | +2.09% | +3.35% | +9.15% |

注：最好的结果是粗体，次好结果是下划线。

表 3 P3不同比例训练数据结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | P3(20%)  MAE | MAPE | RMSE | P3(60%)  MAE | MAPE | RMSE | P3(100%)  MAE | MAPE |
| MEAN | 27.442 | 16.029 | 5.612 | 27.442 | 16.029 | 5.612 | 27.442 | 16.029 | 5.612 |
| HA | 5.833 | 2.741 | 0.337 | 5.731 | 2.707 | 0.331 | 5.675 | 2.670 | 0.328 |
| SRCNN | 5.581 | 3.317 | 0.906 | 5.082 | 2.936 | 0.718 | 4.891 | 2.817 | 0.673 |
| ESPCNN | 5.273 | 3.013 | 0.717 | 5.091 | 2.888 | 0.656 | 4.853 | 2.716 | 0.579 |
| DeepSD | 5.257 | 2.935 | 0.666 | 4.960 | 2.749 | 0.583 | 4.720 | 2.580 | 0.510 |
| VDSR | 5.285 | 2.982 | 0.699 | 4.786 | 2.626 | 0.536 | 4.616 | 2.522 | 0.495 |
| SRResNet | 5.578 | 3.352 | 0.945 | 4.934 | 2.857 | 0.705 | 4.658 | 2.648 | 0.602 |
| LapSRN | 5.832 | 3.535 | 1.019 | 5.041 | 2.920 | 0.721 | 4.641 | 2.589 | 0.550 |
| IMDN | 5.635 | 3.493 | 1.077 | 4.908 | 2.930 | 0.788 | 4.690 | 2.765 | 0.704 |
| SCN | 5.090 | 2.899 | 0.694 | 4.670 | 2.593 | 0.549 | 4.514 | 2.494 | 0.506 |
| UrbanFM | 5.299 | 2.738 | 0.379 | 4.761 | 2.456 | 0.336 | 4.578 | 2.356 | 0.314 |
| UrbanPy | 5.342 | 2.827 | 0.529 | 4.743 | 2.443 | 0.362 | 4.436 | 2.272 | 0.318 |
| FODE | 5.165 | 2.686 | 0.380 | 4.712 | 2.434 | 0.331 | 4.536 | 2.345 | 0.319 |
| UrbanSTC | **4.781** | **2.383** | **0.287** | **4.512** | **2.271** | **0.288** | **4382** | **2.215** | **0.285** |
|  | +6.07% | +11.28% | +14.84% | +3.38% | +6.70% | +12.99% | +1.22% | +2.51% | +9.24% |

注：最好的结果是粗体，次好结果是下划线。

表 4 P4不同比例训练数据结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | P4(20%)  MAE | MAPE | RMSE | P4(60%)  MAE | MAPE | RMSE | P4(100%)  MAE | MAPE |
| MEAN | 19.049 | 11.070 | 4.192 | 19.049 | 11.070 | 4.192 | 19.049 | 11.070 | 4.192 |
| HA | 4.306 | 2.067 | 0.319 | 4.209 | 2.043 | 0.319 | 4.201 | 2.039 | 0.320 |
| SRCNN | 4.048 | 2369 | 0.668 | 3.799 | 2.182 | 0.569 | 3.813 | 2.188 | 0.571 |
| ESPCNN | 3.983 | 2.290 | 0.600 | 4.112 | 2.430 | 0.684 | 3.914 | 2.277 | 0.607 |
| DeepSD | 3.980 | 2.240 | 0.562 | 3.924 | 2.215 | 0.552 | 3.662 | 2.030 | 0.472 |
| VDSR | 3.952 | 2.239 | 0.573 | 3.655 | 2.015 | 0.462 | 3.555 | 1.948 | 0.431 |
| SRResNet | 4.118 | 2.463 | 0.738 | 3.761 | 2.184 | 0.591 | 3.630 | 2.067 | 0.523 |
| LapSRN | 4.467 | 2.753 | 0.884 | 3.705 | 2.103 | 0.530 | 3.679 | 2.118 | 0.544 |
| IMDN | 4.100 | 2.530 | 0.818 | 3.703 | 2.203 | 0.635 | 3.848 | 2340 | 0.720 |
| SCN | 3.798 | 2.154 | 0.550 | 3.573 | 1.987 | 0.467 | 3.486 | 1.927 | 0.439 |
| UrbanFM | 4.054 | 2.126 | 0.373 | 3.677 | 1.908 | 0.323 | 3.559 | 1.841 | 0.305 |
| UrbanPy | 3.959 | 2.088 | 0.413 | 3.644 | 1.889 | 0.332 | 3.470 | 1.801 | 0.313 |
| FODE | 3.912 | 2.042 | 0.350 | 3.627 | 1.879 | 0.314 | 3.529 | 1.828 | 0.304 |
| UrbanSTC | **3.640** | **1.837** | **0.278** | **3.474** | **1.769** | **0.282** | **3.416** | **1.742** | **0.278** |
|  | +4.16% | + 10.04% | +12.85% | +2.77% | +5.85% | +10.19% | + 1.56% | +3.28% | +8.55% |

注：最好的结果是粗体，次好结果是下划线。

表5展示了 BikeNYC数据集上的结果比较，由于无法获得该数据集的外部因素， 因此不会在实验中添加此类信息。在本次实验中当上采样因子为2时，基线DeepSD和 SRCNN相同，因此删除DeepSD。

BikeNYC数据集比TaxiBJ更加稀疏。尽管如此，UrbanSTC依然在RMSE和MAE 指标上平均提高了 2.93%和5.60%o由于BikeNYC数据集极其稀疏，评价指标MAPE不可用。

表 5 BikeNYC不同比例训练数据结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | BikeNYC(20%) | | BikeNYC (60%) | | BikeNYC (100%) | |
| RMSE | MAE | RMSE | MAE | RMSE | MAE |
| MEAN | 3.776 | 1.281 | 3.776 | 1.281 | 3.776 | 1.281 |
| HA | 1.498 | 0.359 | 1.511 | 0.365 | L502 | 0.364 |
| SRCNN | 1.419 | 0.452 | 1.228 | 0.373 | 1.262 | 0.413 |
| ESPCNN | 1.458 | 0.489 | 1.302 | 0.402 | L295 | 0.411 |
| VDSR | 1.888 | 0.838 | 1.616 | 0.700 | 1.476 | 0.626 |
| SRResNet | 1.843 | 0.891 | 1.607 | 0.712 | 1.443 | 0.600 |
| LapSRN | 1.582 | 0.635 | 1392 | 0.516 | 1.320 | 0.464 |
| IMDN | 1.407 | 0.521 | 1.292 | 0.447 | 1.220 | 0.402 |
| SCN | 1331 | 0.424 | 1.191 | 0.356 | 1.162 | 0.332 |
| UrbanFM | 1.405 | 0316 | 1.215 | 0.283 | L172 | 0.263 |
| UrbanPy | 1.381 | 0.315 | 1.271 | 0.286 | 1.126 | 0.250 |
| FODE | 1.293 | 0302 | 1.167 | 0.265 | 1.134 | 0.253 |
| UrbanSTC | **1.267** | **0.276** | **1.146** | **0.246** | **1.093** | **0.236** |
|  | +2.01% | +8.61% | +1.80% | +7.17% | +2.93% | +5.60% |

注：最好的结果是粗体，次好结果是下划线。

### 2.3.3 消融实验

为了分析UrbanSTC每个模块的贡献，本小结进行消融实验分析。本研究只汇报了 TaxiBJ数据集的评估指标(P1到P4的平均实验结果)，因为BikeNYC的实验结果具 有类似的结论。所有的结果显示在表6中。术语“Reg”表示区级级别对比预训练； “Inf”表示空间超分辨率推理预训练；“TCS”表示时间对比预训练。

如表6所示，任何两个模块的组合都要比单一模块效果好，这证明了本工作所提 模块的有效性。仅考虑单一策略时，时间对比的性能要优于区域级别对比和空间超分辨 率推理网络。空间对比包含“Reg”和“Inf”，本研究发现“Inf”的效果要优于“Reg”， 主要因为“Reg”编码器的卷积核是1x1而“Inf”编码器的卷积核是3x3,较大的卷积核尺寸有利于编码器捕获更多的信息。“Reg” + “TCS”和“Inf” + “TCS”组合结果略 差与最终模型，表明这种先验知识同时考虑了空间的时间信息，对于城市细粒度流量推 断具有重要的意义。

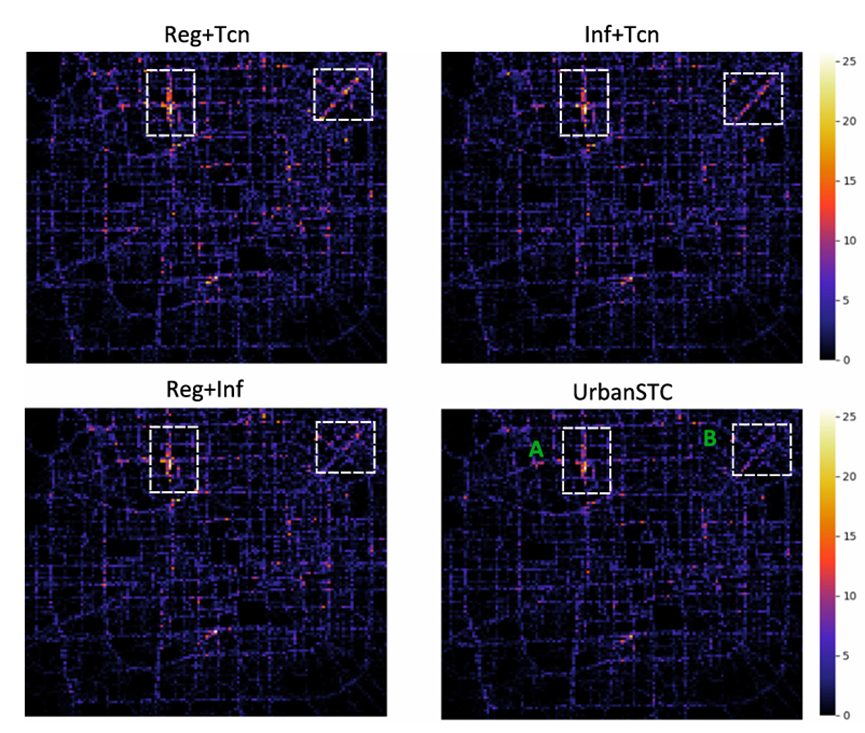


图 6 融实验可视化

表 6 消融实验

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 区域级别对比 | 空间超分辨率推  理网络 | 时间对比 | TaxiBJ | | |
| RMSE | MAE | MAPE |
| √ |  |  | 4.118 | 2.100 | 0.311 |
|  | √ |  | 4.019 | 2.040 | 0.297 |
|  |  | √ | 4.008 | 2.027 | 0.290 |
| √ | √ |  | 3.970 | 2.009 | 0.289 |
| √ |  | √ | 3.983 | 2.009 | 0.287 |
|  | √ | √ | 3.975 | 2.008 | 0.288 |
| √ | √ | √ | **3.967** | **2.004** | **0.287** |

注：最好的结果是粗体，次好结果是下划线。

为了更好地呈现消融实验结果，本研究在图6中展示了一些可视化对比。图6表示UrbanSTC和消融实验的推理误差其中越亮的区域代表更大的误差。区域A （土城西路）和B （三元桥）是北京的主要交通干道。可以明显得观测到，UrbanSTC 比其他消融实验取得了更好的结果，这证明了所提模型的结果能够更好地捕获数据的时空特征。

### 2.3.4 端到端和两阶段对比

为了验证两阶段训练比端到端训练效果更好，本研究在TaxiBJ（Pl到P4的平均实验 结果）和BikeNYC数据集上进行实验。端到端模型集成了三个所提出的模块，即同时 将粗细粒度流量图引入空间自监督、时间自监督和外部因素学习，并对这三个损失函数 进行整体优化。如表7所示，可以清晰地发现两阶段的实验结果优于端到端的训练过程，端到端训练方法需要调整每个损失函数之间的平衡因子，而两阶段训练方法不需要调整辅助任务之间的平衡。自监督学习的优势在于两阶段训练。辅助任务帮助提前学习 数据的内部特征，然后微调阶段学习相应的标签信息［41，42，54，55］。

表 7 端到端和两阶段对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | TaxiBJ  MAE | MAPE | RMSE | BikeNYC  MAE | MAPE |
| End End | 3.980 | 2,053 | 0.294 | 1.120 | 0.245 | 0.077 |
| Two-stage | **3.958** | **1.998** | **0.284** | **1.093** | **0.236** | **0.073** |
| A | +0.55% | +2.68% | +3.40% | +2.41% | +3.67% | +5.19% |

### 2.3.5 时间对比采样分析

为了评估硬采样和权重采样的效果，本研究展示了在TaxiBJ-P1的实验结果。如图7所示，当使用的训练数据低于60%时，权重采样优于硬采样。这是因为权重采样方 法可以综合使用前Top-K个相关样本，而硬采样只选择最相似或者最不相似的数据。随 着训练数据量的增加，硬采样开始表现出比权重采样更好的性能。当训练数据较小时, 很难找到全局最相似的样本，但可以使用Top-K相似样本代替。随着训练数据量的增加 更容易找到最相似的样本，硬采样方法可以获得更好的结果。因此在不同的训练场景中 可以采用两种方法相结合。

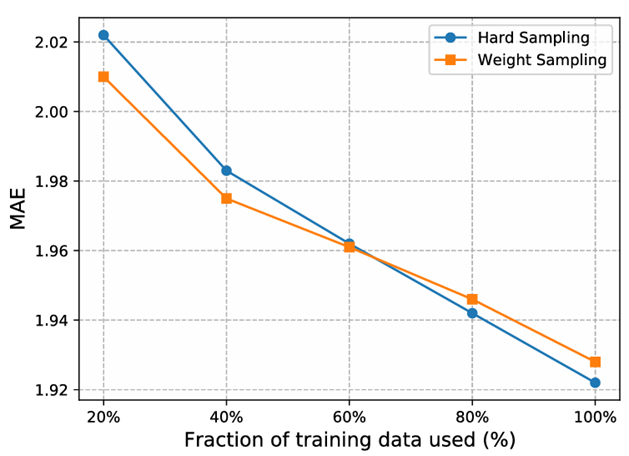


图 7 P1数据集上硬采样和权重采样性能比较

### 2.3.6 融合外部因素研究

FUFI问题存在复杂的外部因素，为了验证本工作方法中外部信息的有效性，引入外部因素并对不同时间跨度的TaxiBJ数据集进行了实验。本研究只将UrbanSTC与可用的基线进行比较，如表8和9所示，UrbanSTC+E在所有时间跨度上的性能都优于其他模型,这表明UrbanSTC和外部因素结合可以提高模型性能。值得注意的是，即使一些精心设计外部信息融合的模型；本研究所提的UrbanSTC具有简单的网络结构来 捕获外部信息。

表 8 TaxiBJ外部因素实验结果1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | Pl  MAE | MAPE | RMSE | P2  MAE | MAPE |
| UrbanFM+E | 3.970 | 2.023 | 0.334 | 4.355 | 2.239 | 0317 |
| UrbanPy+E | 3.909 | 1.981 | 0.330 | 4.353 | 2.230 | 0.327 |
| FODE+E | 3.915 | 1.996 | 0.332 | 4.348 | 2.235 | 0316 |
| UrbanSTC | 3.845 | 1.922 | 0.298 | 4.225 | 2.136 | 0.288 |
| UrbanSTC+E | **3.841** | **1.917** | **0.292** | **4.209** | **2.125** | **0.284** |

注：“+E”表示包含外部因素的模型；最好的结果为粗体显示。

表 9 TaxiBJ外部因素实验结果2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | RMSE | P3  MAE | MAPE | RMSE | P4  MAE | MAPE |
| UrbanFM+E | 4.530 | 2.335 | 0.321 | 3.528 | 1.824 | 0.303 |
| UrbanPy+E | 4.466 | 2.294 | 0.323 | 3.498 | 1.817 | 0.317 |
| FODE+E | 4.505 | 2.329 | 0.314 | 3.505 | 1.821 | 0.311 |
| UrbanSTC | 4.382 | 2.215 | 0.285 | 3.416 | 1.742 | 0.278 |
| UrbanSTC+E | **4.376** | **2.210** | **0.283** | **3.404** | **1.738** | **0.275** |

注：“+E”表示包含外部因素的模型；最好的结果为粗体显示。

### 2.3.7 配置和参数分析

在本节中，本研究尝试探索UrbanSTC方法在各种环境中的学习能力。对比不同的通 道数(32、64、128)，如图8所示。图8(a)表明通道数越多，UrbanSTC具有更好的性能。此外，图8(b)(c)(d)表明更大的通道数量可以提高学习收敛速度。

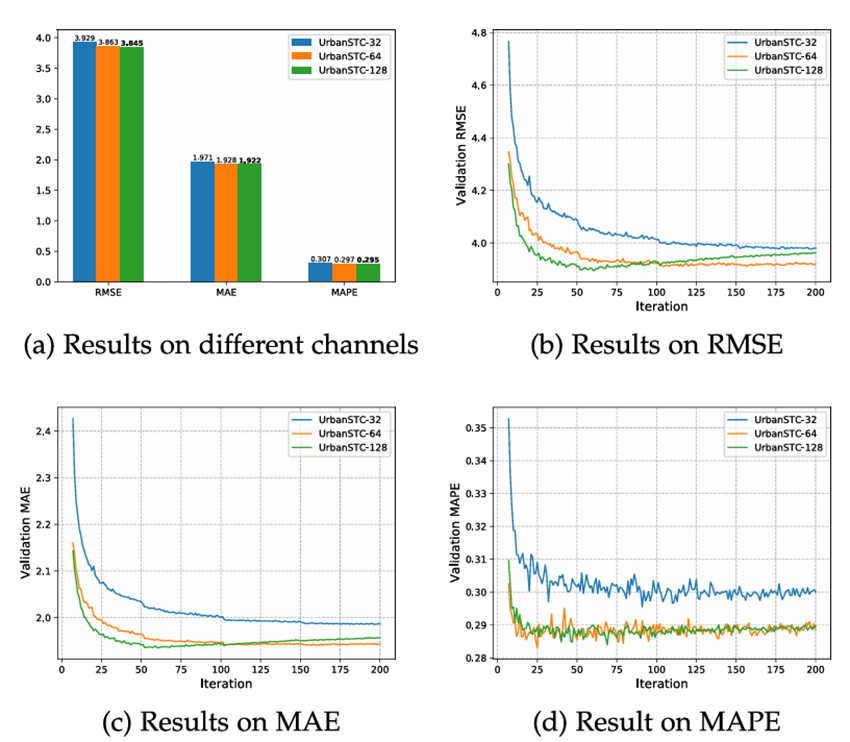


图 8 不同通道数实验结果

本研究分析了 2在区域级别对比自监督中的影响。图9展示了 2在不同设置下的不同性能。区域级别对比根据阈值久判断哪些区域是正负样本。实验结果表明，在TaxiBJ 数据集上阈值为le-4时取得最好效果，而BikeNYC数据上阈值为5e-5产生最佳性能。

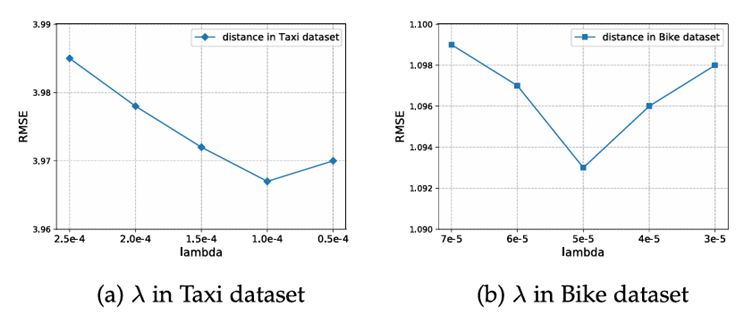


图 9 阈值λ影响曲线

对于模型参数分析，图10(a)表示在不同比例的训练数据情况下UrbanSTC比其他 基线模型具有较好的结果。图10(b)表示传统的图像超分辨率方法，例如：IMDN、VDSR 和SRResNet,由于固有的结构性差异，不适用于FUFI问题。尽管SRResNet和UrbanFM 具有相似的结构，但UrbanFM中的M2-Normalization层有助于解决FUFI问题。UrbanPy 使用级联结构将原始任务分解为多个子任务来预测细粒度的城市流量，这会导致计算复 杂度的增加。FODE利用OED模块替换UrbanFM中的ResNet结构。因为上述模块可 以看成一个连续ODE算子的离散化，能够提高收敛速度，减少模型参数数量。对于 UrbanSTC,本研究设计了几个自监督辅助任务能够让编码器捕获丰富的时空信息。实验结 果证明，所提模型在相对较少的参数下取得了最好的效果。

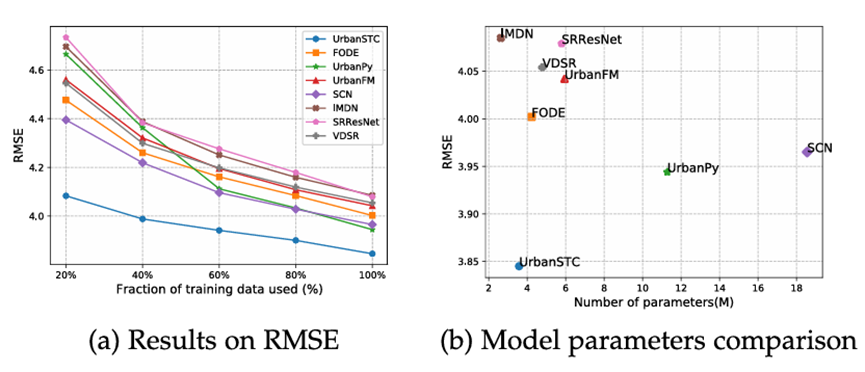


图 10 模型参数研究

本研究还在训练和推理时间方面对UrbanSTC和基线进行了比较。即使UrbanSTC包 含两个阶段，其训练时间也少于之前的FUFI模型，如表10所示。这主要是因为所提 出的编码器结构简单，但可以通过几个精心设计的自监督辅助任务来捕获丰富的时空数 据特征。总体而言，实验结果证明UrbanSTC参数量少、训练时间短、准确率高。

表 10 P1数据集上评估模型的效率

|  |  |  |
| --- | --- | --- |
| Method | Training Time | Inference Time |
| VDSR | 0.050s | 0.012s |
| SRResNet | 0.073s | 0.019s |
| IMDN | 0.052s | 0.015s |
| SCN | 0.260s | 0.110s |
| UrbanFM | 0.125s | 0.036s |
| UrbanPy | 0.313s | 0.130s |
| FODE | 0.135s | 0.036s |
| UrbanSTC | 0.049s | 0.012s |

### 2.3.8 城市细粒度流量推断可视化

图11显示了由UrbanSTC和其他三个基线的推理误差，其中较亮的像素表示较大的误差。总体而言，UrbanSTC获得更详细的推理效果和更少的全局误差。为了更好地可视化推理质量，本研究选择UrbanSTC明显优于其他方法的四个区域（A、 B、C和D）。 A区域为三元桥（市区入口）； B区域为北京动物园（游客众多）；C和 D区域表示北京站和北京西站。与现有的FUFI方法相比，观察到UrbanSTC在上述方面取得了很大的进步。此外，UrbanSTC从热力图中显示出比其他方法更暗的色调，这恰好对应了表3中的实验结果。

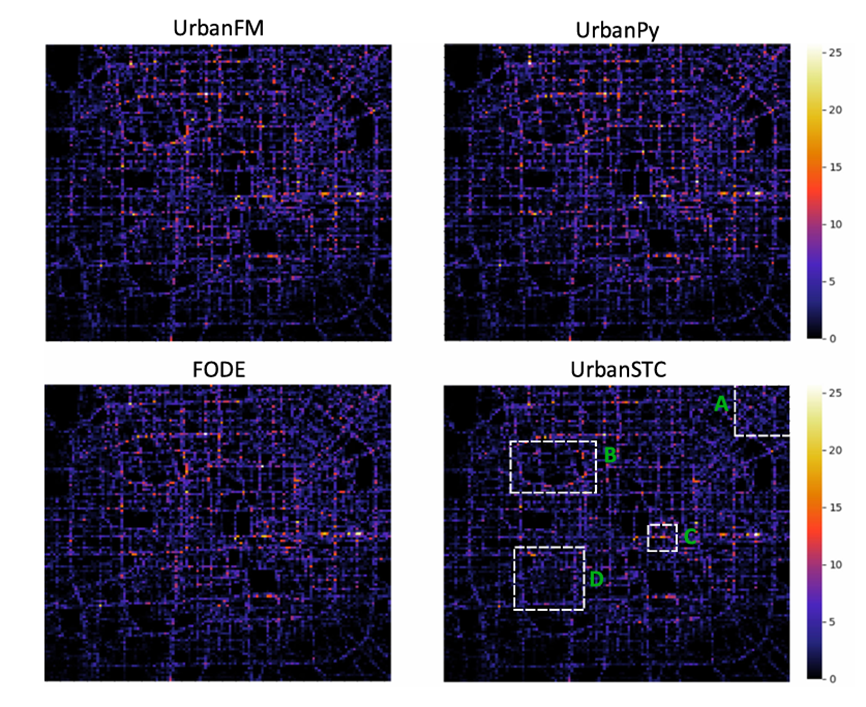


图 11 数据集上不同方法之间的推理误差可视化

## 2.4 本章小节

在本研究中，针对城市细粒度流量推断问题提出了一种名为UrbanSTC的时空对比自监督的方法。所提出的模型可以从城市流量中提取丰富的时空特征，能够从空间和时间相关性两个方面建立自监督辅助任务。对于空间相关性，区域对比和空间超分辨率推理网络为捕捉区域级别流量和流量推断模式之间的相似性做出了巨大的贡献。此外，本研究设计了两种基于时间属性的采样策略。模型的整体结构遵循自监督训练模式：预训练和微调。通过精心设计的自监督辅助任务，使简单的网络可以从复杂的城市流量中学习高级表征能力。对两个真实世界的数据集进行了详细的实验来比较UrbanSTC和其他最先进方法之间的性能。结果不仅表明所提方法优于其他所有方法，而且在训练数据有限时依旧表现出较高性能。

# 3 基于时空吸引学习的不可观测数据下细粒度城市流量推理

## 3.1 引言

随着智慧城市建设的加速推进[1]，城市流量预测成为城市化与智慧城市发展的关键组成部分。高精度的城市流量预测对城市管理者具有重要意义，有助于未来智慧城市的持续建设[2]。然而，仅通过部署传感器获取城市流量数据在实际中往往难以满足需求[3,4,5]，受制于实施成本、地域限制等因素，部分区域流量数据缺失成为普遍现象[6]。

当前关于细粒度城市流量推理的研究仍相对有限[7,8]，且多集中于将缺失值预测与细粒度推理视作两个独立任务，而忽略了现实中两者往往需要同时处理[6]。为此，本文提出了 UrbanSTA 模型，旨在解决以下三个主要挑战：

1. 城市流量图的不完整性：由于传感器分布不均和区域覆盖受限，导致流量图中存在缺失值；
2. 粗细粒度流量之间的复杂相关性：一个粗粒度区域通常对应多个相邻细粒度区域，推理难度大；
3. 城市流量图中空间属性的相互依赖性：区域流量通常受其邻域及相似区域流量变化的影响。

UrbanSTA 包含两个子网络：STA 模块采用非对称编码-解码结构，通过提取空间与时间特征来预测缺失值；FIN 模块则在空间注意力机制下利用预测出的粗粒度流量推断细粒度流量，同时采用分布式上采样机制以恢复原始结构约束关系。本文的主要贡献如下：

1. 城市流量图中空间属性的相互依赖性：区域流量通常受其邻域及相似区域流量变化的影响。
2. 设计分布式上采样策略以建模粗细粒度城市流量之间的复杂关系；
3. 考虑空间旋转不变性与时间周期性，引入时空吸引约束损失以增强时空表征能力。

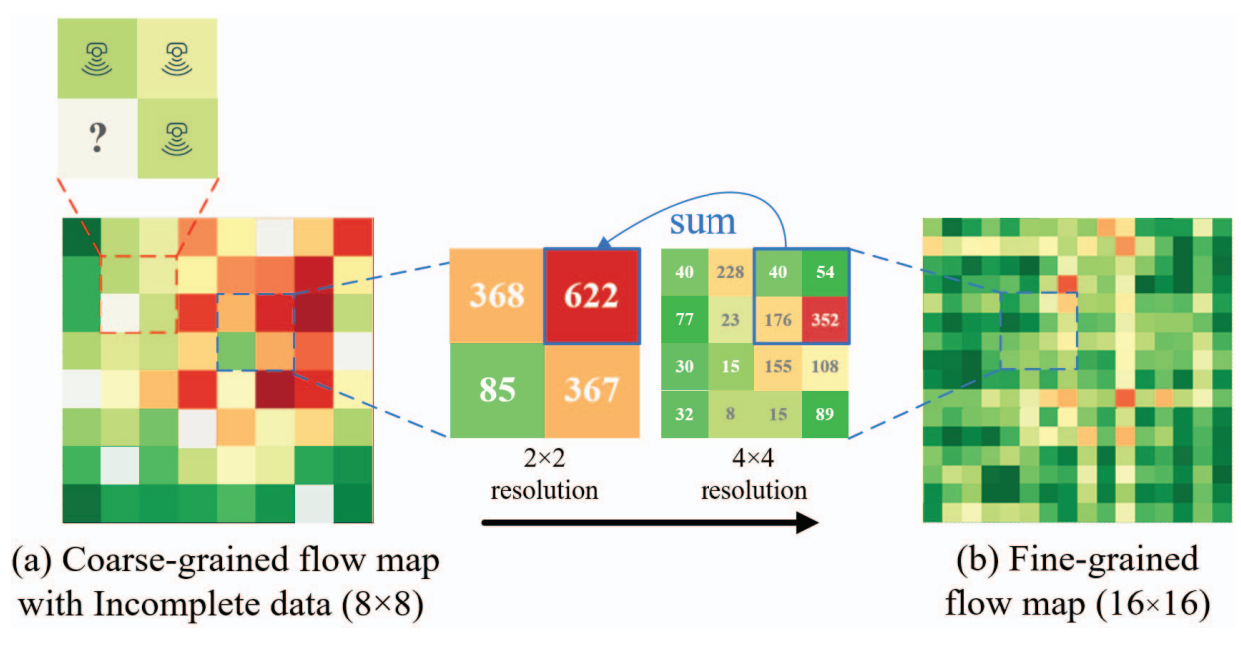
****

图 12 两种粒度层次的城市流量图示意图

## 3.2 方法描述

本文提出的 UrbanSTA 模型如图 13 所示，旨在处理不可观测数据下的细粒度城市流量推理问题。

### 3.2.1 符号说明与问题定义

#### 3.2.1.1 定义1：粗粒度与细粒度城市流量图

一般而言，原始传感器所采集的 数据被定义为粗粒度城市流量图 ，通过推理可得细粒度城市流量图 ，其中 为上采样因子。

#### 3.2.1.2 定义2：不完整的粗粒度城市流量图

在现实中，由于某些区域长期缺失观测数据，粗粒度城市流量图呈现出固定位置的缺失模式，记作。

#### 3.2.1.3 定义3：结构约束

每个粗粒度区域的流量 等于其对应细粒度区域集合的流量总和，即：

其中满足，。

#### 3.2.1.4 目标定义

给定上采样因子 和一组历史及当前的不完整粗粒度流量图 ，本研究的目标是推理当前完整的细粒度城市流量图 。

### 3.2.2 城市流量数据补全网络

#### 3.2.2.1 初始化

**掩码**。为了对缺失区域进行城市流量数据补全，我们定义掩码操作，缺失区域为 1，其他区域为 0。

**划分为子块**。我们将地图 按网格划分为规则的非重叠二维子块 ，然后我们将随机缺失的子块记为 ，剩余区域为 。

#### 3.2.2.2 STA 编码器

**线性嵌入**。我们的编码器通过可训练的线性投影 对未遮蔽的子块进行嵌入：

其中， 表示位置嵌入， 表示时间嵌入。

**分离的时空注意力模块**。首先，STA 编码器通过一系列分离的时空 Transformer 块处理嵌入向量 ，其中时间和空间注意力依次分离进行。对于每一层 ，我们计算查询、键、值向量：

其中， 表示层归一化， 表示多个注意力头的索引， 表示注意力头数，潜在维度为 。

分离时序注意力通过点积计算。对于每个块 ，我们首先计算 中的每个块以及在其他时间处于相同空间位置的所有块的时间注意力，如下所示：

其中， 表示 激活函数， 表示每个子块与时间维度上的所有同位置子块进行 次比较。然后，将这些来自所有注意力头的矢量拼接如下：

接着，作为新编码送入空间注意力模块。根据上式从 中重新获得 ，我们同时计算时间步 中每个子块与所有不同空间位置子块之间的空间注意力，具体如下：

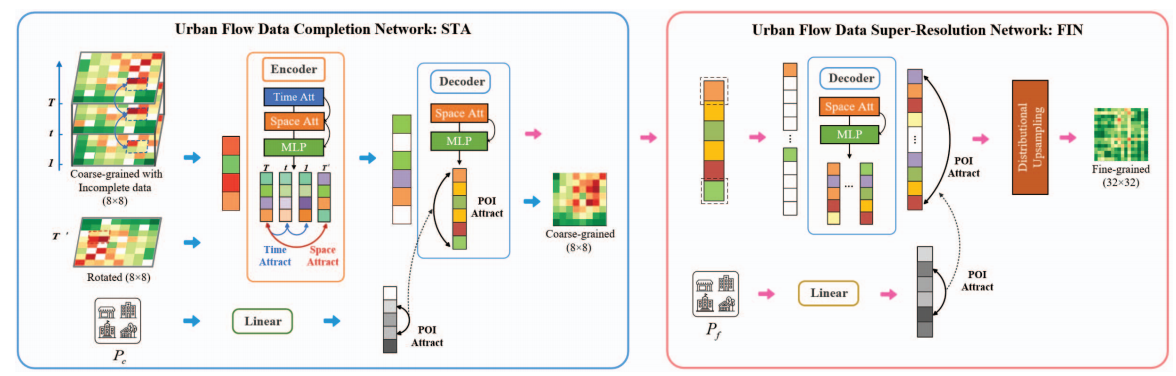


图 13 UrbanSTA 框架结构

其中 表示每个子块在空间维度被比较 次。注意，分离注意力每个子块仅需 次比较，较其他自注意力计算方法更加高效[17]。

最终，所有 向量按照上述方式拼接，并通过多层感知机（MLP）进一步处理，结合残差连接获得 。

**时间与空间吸引**。我们在时间维度设计时间周期性损失 ，用于提取周期性信息。当前时刻 可以从历史数据中提取三个时间属性：接近性（closeness）、周期性（period）与趋势性（trend）。时间周期性损失定义为：

其中 表示周期性与趋势性特征图， 与 分别为周期性与趋势性时间片数。

为提升 Transformer 的局部特征捕捉能力，我们提出空间旋转损失 ，基于一定程度的旋转不变性。

将粗粒度流量图 随机旋转角度 得到 ，再分块并计算其空间注意力嵌入特征为 ，定义如下：

其中 表示旋转还原函数，用于将旋转后的特征恢复至原始位置。

#### 3.2.2.3 STA 解码器

**重建目标**。STA 解码器的输入是完整的粗粒度城市流量图子块集合 ，该集合包括编码后的未遮蔽子块 与掩码子块 ：

其中 表示较小维度的线性投影层， 为排序函数，用于将所有子块恢复至初始 的顺序， 为解码器的位置嵌入。

随后，向量 被输入至空间注意力模块以捕捉空间信息 。解码器特征经线性投影并重构为完整的粗粒度城市流量图 ：

其中 为 reshape 函数，将子块重构为二维图， 为线性投影矩阵。

**POI 吸引**。我们收集每个区域的兴趣点（POI）类别分布（如购物、医疗、教育和居民），记作粗粒度和细粒度的 POI 特征为 与 ，其中 表示兴趣点类别数[18]。

我们使用线性投影融合 类 POI，随后（如图 13 蓝色框底部）在粗粒度 POI 图中选取锚点区域 ，从候选区域 中依据距离与阈值选择 个正样本 与 个负样本 。

我们提出 POI 对比损失 ，旨在拉近正样本区域对应的流量特征 ，同时拉远负样本 ，定义如下：

其中 表示两个嵌入特征之间的相似度函数（例如内积）。

### 3.2.3 城市流量数据补全网络

**细粒度重构**。粗粒度特征 的关系需要被重构为细粒度特征关系，定义如下：

其中 表示学习得到的向量，代表缺失的子块。我们为每个粗粒度子块向量补充 个细粒度子块向量，从而形成细粒度关系映射图。共计 组 个子块向量作为输入。

**超分辨率推理**。随后，我们通过空间注意力模块进一步提取特征。粗粒度子块被丢弃，仅保留剩余 组细粒度子块，重组为二维图 ：

其中 表示按照区域位置将一维子块组合为二维图的重构函数， 表示线性投影。与此同时，我们选择 归一化策略，使细粒度流量之和等于其对应的粗粒度流量，定义如下：

其中 表示细粒度城市流量的分布概率。

### 3.2.4 最终目标函数

为了同时实现城市流量数据补全与超分辨率推理，我们提出了一个多任务学习网络，并采用两阶段的训练策略。

第一阶段为预训练阶段，针对数据补全网络，使用像素级均方误差（MSE）损失、时间周期损失、空间旋转损失与 POI 对比损失进行联合训练，定义如下：

其中 为超参数， 表示时间步 的粗粒度城市流量图真实值。

第二阶段训练中，我们在训练经过预训练的 STA 模块的基础上，引入 FIN 模块，加入像素级 MSE 损失与 POI 对比损失，最终实现最优总体UrbanSTA如下：：

其中 为超参数， 表示时间步 的细粒度城市流量图真实值。

## 3.3 实验与分析

### 3.3.1 实验设置

### 3.3.1.1 数据集

数据集的统计信息如表 11 所示。在实验中，我们将数据划分为互不重叠的训练集、验证集和测试集，比例为 2:1:1。

表 11 数据集统计信息

|  |  |  |
| --- | --- | --- |
| **数据集** | TaxiBJ | BikeNYC |
| 时间跨度 | 7/1/2013 - 10/31/2013 | 1/1/2019 - 3/31/2019 |
| 时间间隔 | 30分钟 | 1小时 |
| 粗粒度 | 8 × 8 | 8 × 8 |
| 细粒度 | 16 × 16 / 32 × 32 | 16 × 16 / 32 × 32 |
| 放大因子（M） | 2 / 4 | 2 / 4 |
| POI信息 | √ | √ |

### 3.3.1.2 对比方法

我们将 UrbanSTA 与六种基线方法进行了比较。其中，UrbanFM、UrbanPy、FODE 为细粒度流量推理方法；MT-CSR、STA-UP、STA-FD 为考虑缺失数据的细粒度流量推理方法。值得注意的是，STA-UP 和 STA-FD 分别是将 STA 模块与 UrbanPy 和 FODE 结合后的版本。

### 3.3.1.3 训练细节与超参数

子块大小设置为 1。STA 模型预训练时的学习率设为 ，批大小为 256；UrbanSTA 完整模型训练时的学习率为 ，批大小为 128。

### 3.3.2 与基线方法的对比分析

**性能对比**。不同方法的计算结果汇总如表 12 和表 13 所示。实验结果表明，UrbanSTA 在 TaxiBJ 数据集上的多数指标中取得了最佳性能。相比之下，UrbanFM、UrbanPy 和 FODE 忽略了数据缺失问题。而在 BikeNYC 数据集上，UrbanSTA 在多数情形中也获得了次优表现。由于 UrbanSTA 模型采用 Transformer 网络结构，在小规模数据集上的表现略逊于基于 CNN 的方法。

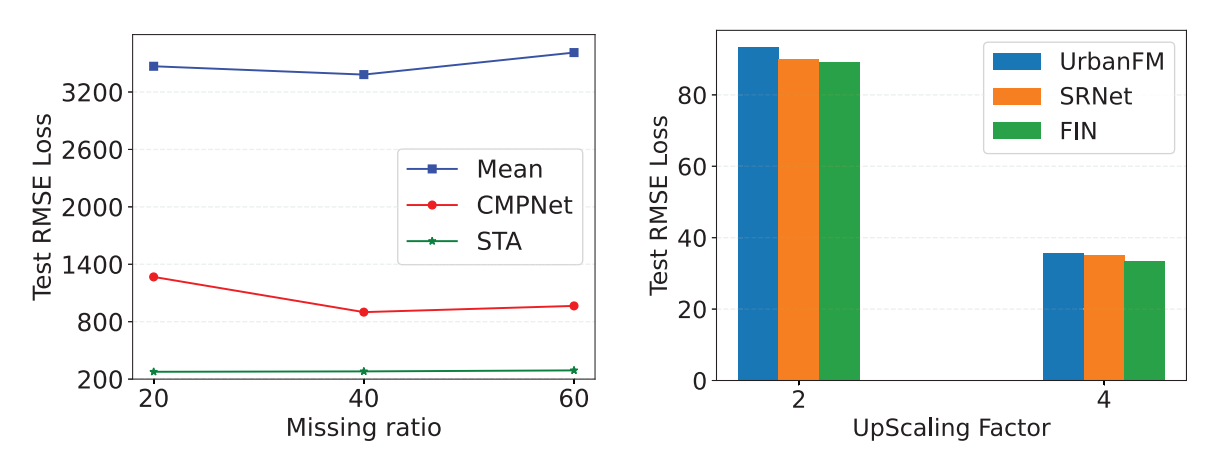


图 14 （a）STA 模块在任务中的有效性分析；（b）FIN 模块在任务中的有效性分析

**模型在两项任务上的效果分析**。我们在 TaxiBJ 数据集上将 STA 模块与平均填补（Mean）和 CMPNet 进行了对比。如图 14(a) 所示，STA 模块明显优于两个基线方法。FIN 模块是一个不依赖数据补全的超分辨率模块。我们使用 TaxiBJ 数据集将 FIN 与 UrbanPy 及 SRNet 进行了比较，结果如图 14(b) 所示。结果表明，在不同上采样因子条件下，FIN 的表现始终略优于 UrbanPy 和 SRNet。

表 12 TaxiBJ 数据集上的对比实验结果

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 模型 |  | | | UrbanFM | UrbanPy | FODE | MT-CSR | STA-UF | STA-FD | UrbanSTA |
| TaxiBJ | 2 | 20% | MAE | 73.25 | 118.87 | 74.39 | 75.81 | 67.62 | **62.99** | 63.03 |
| RMSE | 116.20 | 190.13 | 116.75 | 123.35 | 98.84 | 92.49 | **92.36** |
| 40% | MAE | 113.65 | 174.04 | 116.19 | 117.06 | 72.86 | 66.95 | **65.06** |
| RMSE | 174.22 | 255.15 | 176.04 | 186.86 | 105.26 | 98.32 | **96.01** |
| 60% | MAE | 132.48 | 190.76 | 133.62 | 134.32 | 73.64 | **68.89** | 69.91 |
| RMSE | 202.88 | 274.95 | 201.69 | 214.16 | 108.43 | **102.99** | 103.95 |
| 4 | 20% | MAE | 25.89 | 31.02 | 30.77 | 26.74 | 25.23 | 25.71 | **23.19** |
| RMSE | 40.67 | 48.26 | 46.54 | 44.71 | 38.69 | 40.03 | **37.11** |
| 40% | MAE | 35.53 | 43.50 | 39.59 | 36.34 | 25.84 | 24.84 | **24.36** |
| RMSE | 52.73 | 63.14 | 57.24 | 58.75 | 40.00 | 39.36 | **38.98** |
| 60% | MAE | 39.71 | 45.48 | 42.94 | 39.80 | 26.09 | 25.20 | **24.59** |
| RMSE | 60.65 | 66.24 | 62.45 | 65.20 | 40.87 | 40.49 | **39.95** |

表 13 BikeNYC 数据集上的对比实验结果

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 模型 |  | | | UrbanFM | UrbanPy | FODE | MT-CSR | STA-UF | STA-FD | UrbanSTA |
| BikeNYC | 2 | 20% | MAE | 0.88 | 0.91 | 0.83 | 0.80 | 0.81 | 0.82 | **0.79** |
| RMSE | 2.17 | 2.24 | **1.82** | 1.89 | 2.05 | 1.84 | 1.87 |
| 40% | MAE | 1.07 | 1.20 | 1.09 | 1.02 | 1.08 | **0.87** | 1.44 |
| RMSE | 2.31 | 2.84 | 2.33 | 2.49 | 2.75 | **1.92** | 3.15 |
| 60% | MAE | 1.15 | 1.31 | 1.17 | 1.12 | 1.04 | 0.91 | **0.89** |
| RMSE | 2.48 | 3.23 | 2.49 | 2.58 | 2.24 | **2.00** | 2.09 |
| 4 | 20% | MAE | 0.27 | 0.28 | 0.27 | **0.26** | 0.33 | **0.26** | 0.36 |
| RMSE | **0.94** | 1.07 | **0.94** | 1.02 | 0.97 | 0.95 | 1.03 |
| 40% | MAE | 0.33 | 0.35 | 0.33 | 0.30 | 0.34 | **0.28** | 0.30 |
| RMSE | 1.04 | 1.34 | 1.04 | 1.16 | 1.00 | 0.99 | **0.98** |
| 60% | MAE | 0.35 | 0.35 | 0.35 | 0.33 | 0.35 | **0.32** | **0.32** |
| RMSE | 1.10 | 1.34 | 1.10 | 1.25 | **1.02** | 1.06 | **1.02** |

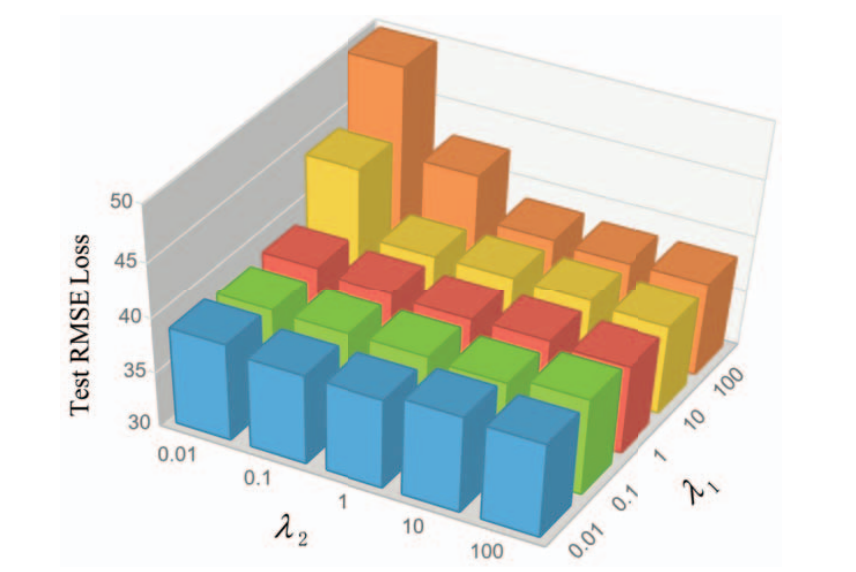


图 15 在 TaxiBJ 数据集（缺失率 20%，上采样因子 4）上的不同超参数设置下模型性能对比

### 3.3.3 消融分析

我们在 TaxiBJ 数据集上对 UrbanSTA 及其若干变体进行了对比分析，计算结果如表 14 所示。结果表明，STA 模块优于其七种变体，表明空间旋转、时间周期以及粗粒度 POI 对比损失在补全阶段均具有积极作用。

同时，我们还开展了对比实验以验证细粒度 POI 约束对 FIN 模块性能的影响。表14 总结了相关结果，添加细粒度 POI 约束显著提升了超分辨率推理任务的表现。

表 14 消融实验结果

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 空间 | |  | √ |  |  | √ | √ |  | √ |  | | √ | √ |
| 时间 | |  |  | √ |  | √ |  | √ | √ | √ | √ |
|  | |  |  |  | √ |  | √ | √ | √ | √ | √ |
|  | |  |  |  |  |  |  |  |  |  | √ |
| 数据补全 | MAE | 226.72 | 210.55 | 214.86 | 209.31 | 212.22 | 210.21 | 212.27 | 204.02 | 超分辨率 | MAE | 24.20 | 24.10 |
| RMSE | 307.90 | 289.40 | 290.67 | 287.84 | 288.76 | 287.21 | 288.07 | 274.80 | RMSE | 38.47 | 38.29 |

### 3.3.4 参数分析

我们分析了模型在不同参数设置下对 RMSE 指标的敏感性。图 15 总结了该实验结果，其中我们分别在 范围内调节最终目标函数中两个超参数 与 。实验结果显示，FIN 模块的作用大于 STA 模块。因此，在实际实验中，我们对 STA 模块进行轻微微调，增强对 FIN 模块的约束，最终设置为 ，，从而实现最佳性能。

# 4 基于自适应时空上下文学习的交通流量预测模型

## 4.1 引言

随着现代城市的快速发展和城市化进程的推进，交通预测已成为智能交通系统和智慧城市建设中的关键要素。精准的交通预测对于拥堵管理、路径规划及应急响应等应用场景至关重要。通过分析历史数据，城市交通系统旨在预测路网未来状态（如流量、速度与密度等指标）[1]。有效的交通预测能显著提升城市交通系统运行效率、缩短出行时间，并改善居民整体生活质量。

近年来，随着城市交通系统复杂度的持续提升，交通预测研究呈现爆发式增长。这类研究通常将交通网络建模为时空图结构：交通传感器对应图中的节点，节点间的相互依赖关系则构成边[2]。图神经网络（GNN）因其能通过消息传递机制聚合邻域节点信息的特点[1][3][4]，已成为交通预测的主流方法。但基于GNN的模型存在过度平滑问题，可能导致节点表征趋同[5][6]；同时由于处理复杂图结构时的高计算与内存开销，其效率瓶颈也备受诟病[7]。此外，受注意力机制在时空建模中成功应用的启发，学界相继提出多种基于注意力的模型[8][9]。这些方法通过加权时空特征的重要性，利用注意力机制来提升模型性能。

现有交通预测方法通常基于大规模数据集中存在隐含不变时空模式的假设。然而，由于时空数据固有的动态特性，历史训练数据与未来测试数据之间可能出现不可预见的分布偏移[10]。交通模式极易受多种外部因素影响，包括不可预测的社会事件、气候变化导致的天气异常、区域建设规划以及城市经济发展等。例如在某些意外期间，大规模封锁政策导致交通流量骤减，这类突发情况可能引发交通模式的剧烈变化，从而显著降低预测模型的有效性。

为应对这一挑战，我们从一个新颖的视角探索交通预测问题，强调以快速自适应方式学习有效时空表征的重要性。我们提出基于日粒度数据的自适应交通预测方法，仅利用单日数据即可训练出适应动态交通场景的模型。这种每日自适应机制使模型能够快速响应由外部因素引起的交通模式变化。通过每日增量训练，模型既能捕捉最新交通趋势，又能及时应对特殊事件或极端天气等突发状况。这种实时响应能力对于维持动态环境中交通预测的准确性与可靠性至关重要。如图16所示，该方法不仅达到了理想的预测精度，还能灵活适应跨城市、跨领域等多种应用场景。

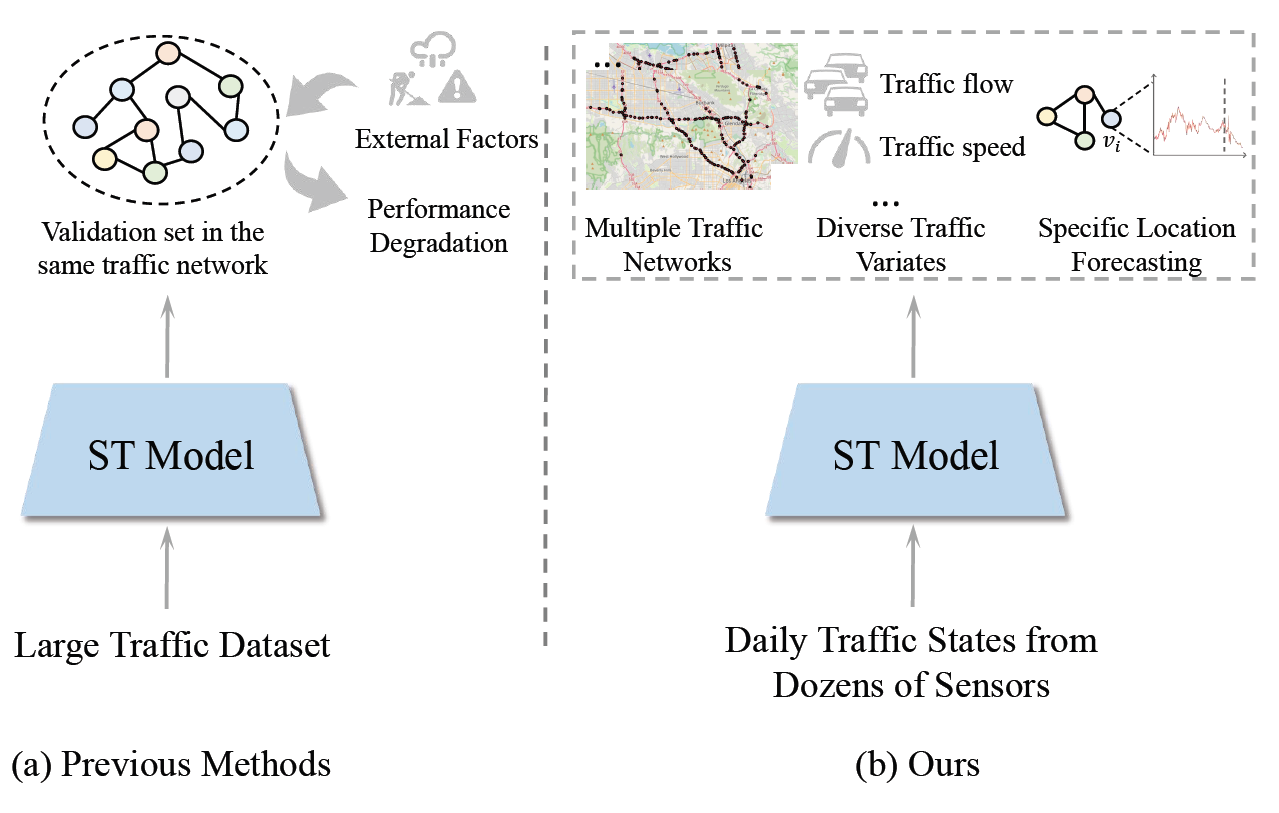


图 16 交通预测框架

与传统方法依赖长期历史数据挖掘不变时空模式不同，本研究聚焦实时交通信息与应用场景特性，提出了一种新型自适应时空上下文学习方法（ASTCL），可实现高精度的日粒度交通预测。本方法创新性地设计了节点级预测任务——以目标节点（如单个交通传感器）的信息聚合作为模型输入，而非全路网数据。这种设计具有三重优势：精准定位：针对特定关注区域进行定向预测，相比城市级路网预测更具成本效益；强迁移性：训练完成的模型可突破特定路网结构限制，适用于城市内部其他区域或跨城市应用；数据增效：基于图结构的节点级预测使训练样本量与交通节点数量成正比，有效缓解日粒度预测中的数据稀缺问题。

针对单一节点数据存在的单调性与稀疏性缺陷，本研究通过构建自适应时空上下文进行全局信息补偿：基于语义相似度，为每个目标节点动态整合相同时段内其他传感器的历史数据。这种上下文增强机制通过聚合具有时空语义关联的节点信息，显著提升了数据多样性与时序覆盖度。为建模动态上下文中的节点关系，本工作提出动态序列图结构，其边权重随时间序列动态演化。在此基础上，创新性地设计了时空联合模块，通过联合注意力机制与联合卷积操作，同步捕捉空间、时间及时空交叉维度的复杂关联。ASTCL仅包含数千参数，模型复杂度与计算需求显著低于主流方法，四个真实交通数据集上的实验表明，ASTCL均稳定超越现有最优基线模型，各项指标提升幅度达1.2%-5.7%。

## 4.2 方法描述

本研究将由交通传感器收集的历史交通数据表示为 ，其中，代表可用传感器的数量，表示时间范围，代表所研究的交通特征。在此基础上，采用滑动窗口机制将数据序列划分为多个时间区间，每个窗口大小为 ，从而得到 。通常情况下，直接作为模型输入，并依据的大小进行多轮训练。然而，在基于单日数据的交通预测场景中，由于每天的数据量有限，的取值通常仅为数百，导致生成的训练样本数量显著受限。

为应对这一挑战，本研究提出一种基于节点级预测的训练任务。具体而言，对于交通网络中的目标节点（即交通传感器），模型以其自适应时空上下文作为输入，预测其未来 个时间步的交通状态，而非整个交通图的状态。通过这一设计，训练样本数量将随着交通传感器数量的增加而成倍增长，从而有效缓解数据稀缺问题。该预测任务的定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

其中，为预测结果。

尽管节点级预测能够扩充训练样本数量，但单个节点的交通数据往往缺乏多样性，难以提供足够的信息支持模型的稳定训练。近期研究表明，节点的交通状态主要受到其局部邻域的影响，因此通过融合相邻节点的信息构建目标节点的局部或区域上下文。然而，这些方法未能充分考虑全局空间的影响，限制了模型的预测能力。为此，本研究提出自适应时空上下文的概念，旨在从全局视角为目标节点引入更加多样化的数据，使模型能够捕捉更广泛的交通环境及其对目标节点的影响模式。在时间窗口 内，目标节点 的特征可表示为：。本文的目标是自适应地识别对目标节点相关性强的节点集合，并将这些节点聚合为自适应时空上下文。

为了量化不同节点间交通状态的相关性，本研究采用快速动态时间规整（fastDTW）算法 ，该方法基于经典的动态时间规整（DTW）技术，通过fastDTW计算得到的数值可用于衡量节点间交通模式的相似程度，从而有效评估两个节点在相同时段内的序列相似性。节点与之间的相关性权重通过高斯核函数定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (40) |

其中，表示与之间的相关性权重，权重值越高表明二者关系越密切；表示fastDTW计算；为所有值的标准差。基于的值，本研究选择与目标节点交通模式最相似的个节点，并将其特征拼接以构建自适应时空上下文。目标节点本身也被包含在该自适应上下文中，以确保其自身的时空模式得到保留。这种方法使模型能够动态地从全局交通网络中选择相关信息，从而增强预测的鲁棒性和准确性。

ASTCL的模型设计与实现，主要包括时空联合学习、动态序列建模和模型预测三个方面的内容，ASTCL模型的整体架构如图17所示。ASTCL的核心任务是从自适应时空上下文中捕捉复杂的时空相关性，并预测交通状态的未来趋势。在时间维度上，历史交通数据揭示了随时间推移不断重复的模式和趋势；在空间维度上，目标节点的交通状况通常受到自适应上下文中其他节点的影响，这种影响源于交通节点间的语义相似性。以往的研究通常采用独立的空间、时间和时空编码器，并将它们按序堆叠以提取各维度的特征。然而，这种堆叠策略不仅增加了模型的深度和复杂性，还可能导致不同维度之间交互信息的丢失。本研究提出了两种联合时空学习方法：联合注意力机制和联合卷积。这两种操作能够在单层编码器中同时学习空间、时间和时空表示，同时保持较低的计算复杂度。

给定自适应时空上下文作为输入，ASTCL首先对输入数据应用实例归一化，归一化后的数据将在模型的最终预测阶段进行反归一化。如图17所示，归一化后的时空上下文经过一层卷积神经网络得到自适应时空上下文的初始特征嵌入，特征维度从转换为，得到特征张量 。为了学习更高层次的时空表示，本文将特征提取模块进行了次堆叠，对于第 层的输出，第一步的卷积计算过程如下：

|  |  |  |
| --- | --- | --- |
|  |  | (41) |

其中，是学习到的特征嵌入； 是第 层的输出；\ 表示卷积操作；是激活函数；是可学习参数。

为了充分利用自注意力机制在捕捉长距离依赖关系方面的优势，本研究设计了一种联合时空注意力机制，旨在单阶段内高效建模空间、时间和时空相关性。具体而言，首先将 根据特征通道均匀地划分为四部分，得到 、、 和 ，每部分都是具有相同特征维度的张量。然后，针对每个特征划分利用不同的注意力机制来捕捉各种依赖关系。对于，本项目利用多头空间自注意力机制来捕捉同一时间步内节点之间的空间依赖关系。具体地，对于时间戳 ，首先通过特征变换得到查询（Q）、键（K）和值（V）矩阵，公式如下：

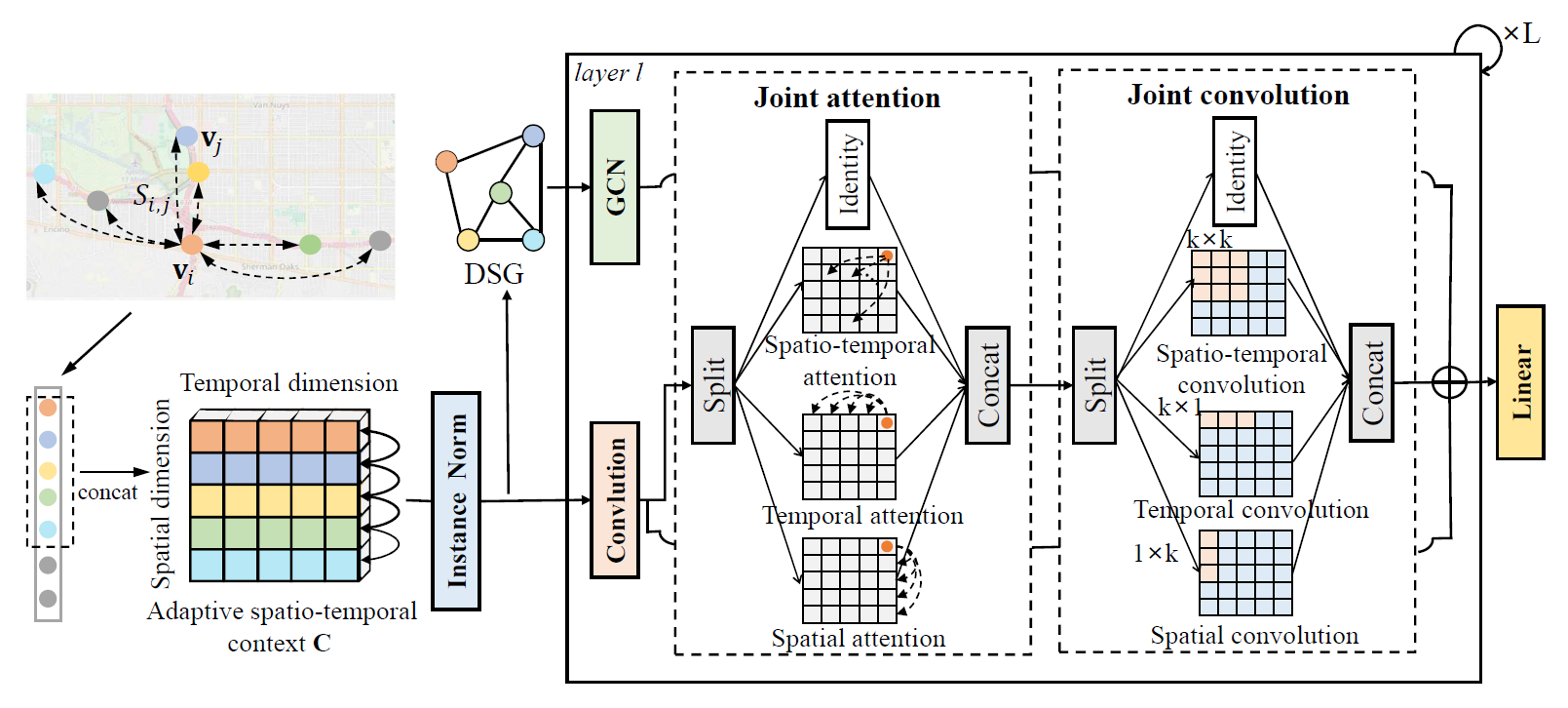


图 17 ASTCL模型框架图

|  |  |  |
| --- | --- | --- |
|  |  | (42) |

其中，、 和 是可学习参数。接着，通过自注意力操作获得时间步  上所有节点的注意力图。空间注意力图通过计算同一时间步内不同节点之间的成对关系，揭示了沿时间维度不变的空间属性。最终，空间自注意力的输出通过以下公式计算：

|  |  |  |
| --- | --- | --- |
|  |  | (43) |

对于，本方法应用时间自注意力机制来捕捉每个节点在时间维度上的相关性。对于则应用时空注意力机制来捕捉动态的时空相关性，全面理解交通模式在时间和空间上的演变规律。为了保持特征一致性并避免模型过度复杂化，对最后一个部分 应用恒等映射，直接保留其原始特征。随后将所有输出进行拼接，并通过通道重排操作重新排列学习到的特征，最终得到联合注意力模块的输出。

在联合注意力模块之后，为了建模局部的时空相关性，本研究提出一种联合卷积操作。首先采用与联合注意力模块类似的分割策略将划分为多个部分，并对每部分应用不同的卷积模块。这里使用三种不同的卷积核大小：、和 ，分别从空间、时间和时空视角提取特征。最后，联合卷积模块的输出通过残差连接与输入特征融合，其公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (44) |

其中，、 和 是不同卷积核的可学习参数； 表示卷积操作；是通道重排列；学习到的时空表示。

图神经网络在交通预测中被广泛应用于捕捉空间依赖性。构建交通图的常见方法是通过相似度或距离度量预定义一个连接图，以这种方式生成的图通常是时间不变的，并应用于所有时间戳，如图3所示。为了更好地捕捉交通动态，近期研究提出在每个时间戳构建动态图，如图 18所示。然而，这种方式通常忽略了不同时间间隔之间的相互作用，从而限制了模型的预测性能。

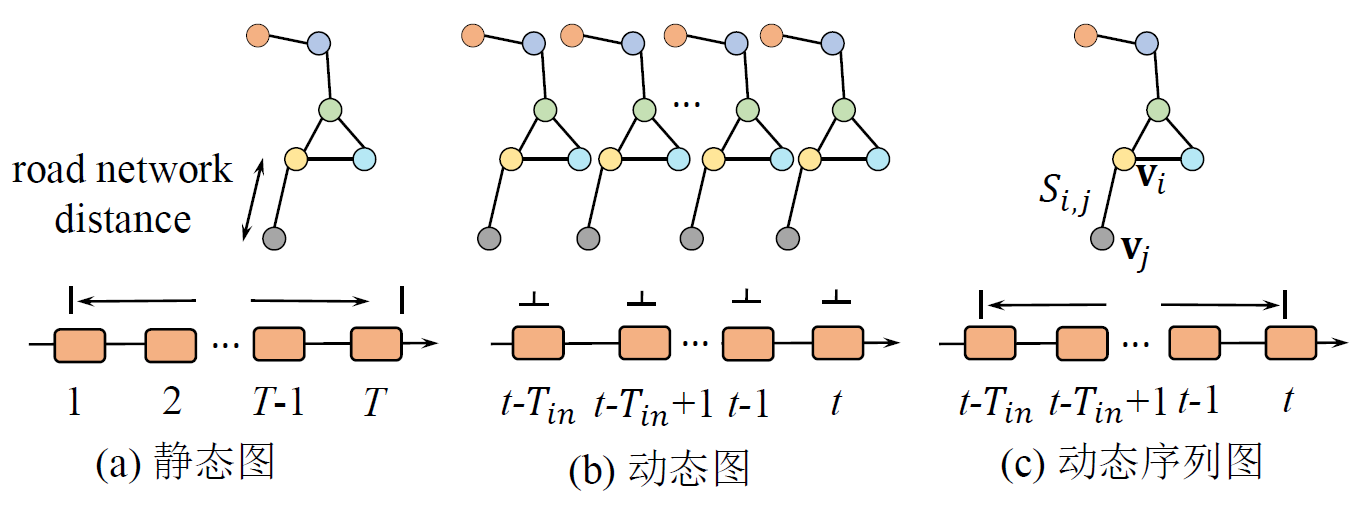


图 18不同图的构建方式

针对上述问题，本研究提出动态序列图（Dynamic Sequence Graph，DSG），旨在建模节点在整个输入序列中的复杂关系，动态捕捉空间和时间依赖性。DSG通过计算节点间的成对边权重，并设置阈值来限制图的密度，以确保边权重能够准确反映节点在空间和时间维度上的动态关系。DSG的构建在数据准备阶段完成，不会在训练过程中引入额外的计算开销。

基于DSG，本文为自适应时空上下文生成相应的邻接矩阵 并采用图卷积网络从图中提取特征。GNN通过聚合和转化邻域信息来平滑节点信息，给定跳数，第层的计算如下：

|  |  |  |
| --- | --- | --- |
|  |  | (45) |

其中，是经过图学习得到的特征，  是可学习的参数， 表示转移矩阵的幂级数，通过 来生成。

ASTCL的最后阶段是将来自动态序列图建模、联合时空学习和残差连接的特征结合起来，其公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (46) |

其中，是第 层的输出，综合了空间、时间和时空维度的特征信息。在通过堆叠的 个学习阶段后，可以得到高层时空特征 。为了生成多步交通预测，本研究应用一层MLP将时空特征映射为预测结果：

|  |  |  |
| --- | --- | --- |
|  |  | (47) |

其中，表示未来步预测结果。该模型的训练目标是最小化预测结果和真实交通信息 之间的平均绝对误差，其损失函数定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (48) |

其中， 是ASTCL中的可学习参数。

## 4.3 实验与分析

### 4.3.1 实验设置

本研究利用 PyTorch 框架实现了所提出的模型，并在 PEMS04 和 PEMS08 数据集来进行交通流量预测任务的评估，METR-LA 和 PEMS-BAY 数据来进行交通速度预测任务的评估。为了模拟快速适应的交通场景，本文将每个数据集中的第一天的交通数据作为所有基准模型的训练集。具体而言， PEMS04 和 PEMS08数据集的最后 40% 数据按 1:1 的比例划分为验证集和测试集；而 METR-LA 和 PEMS-BAY 数据集的最后 30% 数据按 1:2 的比例划分。本文利用过去一小时的交通数据（Tin = 12）预测未来三个时段的交通信息，分别为：15 分钟（Tout = 3）， 30 分钟（Tout = 6），和 60 分钟（Tout = 12）。通过参数网格搜索，构建自适应上下文时使用的节点数为 15，模型的层数和特征维度在最佳表现的设置分别为 2 和 32。

### 4.3.2 方法对比

表15和16分别展示了 ASTCL 和其他基线模型在交通流量预测和交通速度预测上的表现。可以看出，ASTCL 在四个数据集上均优于其他先进的交通预测模型，在所有评估指标上均取得了最佳性能。ASTCL 在单日训练数据情况下，能够快速适应不同交通的状况，在数据稀疏情况下表现出优秀的预测性能，展现出较强的鲁棒性和泛化能力。特别是在 PEMS04 和 PEMS08 这两个误差较大的数据集上，ASTCL 在大多数情况下实现了超过 30% 的性能提升。对于 METR-LA 和 PEMS-BAY 数据集，ASTCL 同样表现最佳，与次佳的模型相比，性能提升最高达 11.7%。从预测准确度上看， ASTCL 在捕捉和预测复杂交通模式方面的能力较为突出，同时保证了预测结果的鲁棒性和稳定性。

其次，在需要模型快速适应的单日交通预测任务当中，传统统计方法和深度学习模型均面临挑战。统计方法如 HA 和 VAR 虽能利用单日交通数据做出预测，但对异常值高度敏感，且难以适应交通模式的动态性和复杂性。另一方面，数据驱动的模型在需要快速适应的单日交通预测中可能会因为过拟合问题而导致性能下降。如表 15 所示，AGCRN 通过可学习的节点嵌入来生成自适应图，但在仅有一天数据的情况下，难以学习到有效的交通节点表征，而 GWNET 和STAEformer 也面临类似的问题。值得注意的是，在交通速度预测任务中，部分深度学习模型的表现甚至不如统计方法 VAR，这进一步突显了深度学习模型在单日交通预测任务上的局限性。相比之下，ASTCL 通过引入自适应时空上下文学习而丰富了时空语义，在预测性能和鲁棒性上取得了显著提升。

表 15 交通流量预测实验结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | PEMS04 | | | PEMS08 | | |
| 方法 | 15 分钟 | 30 分钟 | 60 分钟 | 15 分钟 | 30 分钟 | 60 分钟 |
|  | MAE RMSE MAPE | MAE RMSE MAPE | MAE RMSE MAPE | MAE RMSE MAPE | MAE RMSE MAPE | MAE RMSE MAPE |
| HA | 121.31 154.05 102.74 | 121.31 154.05 102.74 | 121.31 154.05 102.74 | 119.73 149.72 155.53 | 119.73 149.72 155.53 | 119.73 149.72 155.53 |
| VAR | 46.72 62.01 23.18 | 80.62 108.92 44.47 | 51.42 65.35 36.71 | 44.63 62.02 14.99 | 71.83 91.71 26.77 | 108.34 144.77 38.83 |
| ASTGCN | 41.08 58.85 31.78 | 42.34 60.67 32.00 | 46.90 67.30 38.42 | 23.40 35.27 17.95 | 25.08 38.05 18.94 | 31.16 45.91 22.88 |
| AGCRN | 160.25 213.83 100.67 | 160.74 214.41 100.22 | 160.87 214.58 100.12 | 160.33 207.24 108.17 | 159.48 206.33 108.76 | 159.24 206.13 108.83 |
| STSGCN | 41.96 64.12 30.23 | 44.04 66.43 31.65 | 50.22 73.91 34.93 | 35.13 53.83 24.60 | 36.56 55.74 25.23 | 42.05 62.93 27.57 |
| Bi-STAT | 33.28 51.20 21.19 | 37.30 58.01 23.63 | 47.15 72.65 30.25 | 22.86 33.10 16.34 | 24.76 36.45 17.92 | 32.13 46.79 23.09 |
| STID | 83.29 103.01 62.34 | 97.49 123.25 67.01 | 122.46 156.90 76.15 | 61.91 80.01 39.41 | 64.04 84.99 38.04 | 79.10 106.81 47.71 |
| FourierGNN | 29.37 44.12 24.50 | 37.65 55.26 31.85 | 48.47 70.89 39.57 | 24.01 35.60 16.45 | 30.27 44.82 20.29 | 43.56 62.56 26.83 |
| PDFormer | 32.00 44.65 33.26 | 79.41 115.50 121.85 | 96.94 131.52 155.95 | 65.35 91.46 108.86 | 93.03 121.78 176.17 | 101.42 129.15 197.72 |
| STAEformer | 82.15 107.78 58.66 | 92.23 119.36 71.45 | 96.54 125.44 74.51 | 38.40 51.44 30.76 | 44.24 58.84 35.11 | 52.38 69.45 35.88 |
| **ASTCL20** | 22.20 34.88 14.54 | 26.25 40.96 16.77 | 34.41 52.79 22.09 | 18.01 27.60 11.53 | 20.86 32.54 13.08 | 28.54 43.43 17.59 |
| **ASTCL50 ASTCL** | 21.67 34.19 14.20  **21.36 33.78 14.05** | 25.28 39.58 16.28  **24.90 39.01 16.17** | 33.55 51.41 21.99  **32.83 50.01 21.36** | 16.84 26.10 10.80  **16.77 25.95 11.15** | 19.74 30.87 12.61  **19.55 30.43 12.76** | 26.44 40.78 16.70  25.77 39.46 16.86 |

表 16 交通流量预测实验结果

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集 | METR-LA | | | | | | | | | PEMS-BAY | | | | | | | | |
| 方法 | 15 分钟 | | | 30 分钟 | | | 60 分钟 | | | 15 分钟 | | | 30 分钟 | | | 60 分钟 | | |
|  | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| HA | 11.30 | 15.86 | 30.99 | 11.30 | 15.86 | 30.99 | 11.30 | 15.86 | 30.99 | 6.42 | 9.12 | 14.71 | 6.42 | 9.12 | 14.71 | 6.42 | 9.12 | 14.71 |
| VAR | 7.74 | 12.50 | 23.02 | 8.10 | 11.25 | 19.15 | 7.15 | 10.79 | 18.18 | 4.24 | 6.86 | 7.55 | 4.33 | 6.39 | 7.39 | 5.32 | 8.10 | 8.61 |
| STGCN | 8.75 | 16.13 | 8.84 | 10.24 | 18.42 | 9.89 | 12.21 | 21.08 | 13.78 | 4.64 | 9.67 | 6.29 | 4.84 | 9.63 | 6.52 | 5.21 | 10.38 | 6.95 |
| DCRNN | 3.71 | 7.92 | 9.18 | 4.78 | 10.17 | 11.28 | 6.48 | 13.01 | 17.38 | 2.22 | 5.08 | 6.57 | 2.64 | 5.81 | 7.44 | 3.29 | 7.01 | 8.97 |
| GWNET | 3.24 | 6.09 | 8.44 | 3.83 | 7.58 | 11.01 | 4.90 | 9.51 | 15.49 | 1.76 | 3.73 | 4.52 | 2.24 | 4.92 | 5.71 | 2.97 | 6.78 | 7.53 |
| GMAN | 10.57 | 16.79 | 21.06 | 11.01 | 17.52 | 21.91 | 11.73 | 18.61 | 23.44 | 3.82 | 6.48 | 8.98 | 4.05 | 6.92 | 9.50 | 4.49 | 7.77 | 10.54 |
| MTGNN | 11.54 | 14.61 | 32.38 | 10.96 | 13.80 | 30.74 | 11.22 | 13.98 | 30.51 | 5.17 | 10.14 | 14.32 | 5.26 | 10.34 | 14.59 | 5.53 | 10.78 | 31.09 |
| STID | 5.48 | 10.07 | 19.08 | 5.84 | 10.51 | 20.30 | 6.26 | 11.38 | 22.21 | 2.94 | 6.05 | 8.18 | 4.07 | 9.03 | 11.96 | 4.80 | 10.28 | 13.84 |
| FourierGNN | 4.17 | 7.11 | 10.57 | 4.81 | 8.92 | 12.77 | 5.35 | 10.01 | 14.13 | 2.46 | 5.42 | 5.47 | 3.55 | 6.50 | 7.61 | 6.45 | 9.40 | 13.20 |
| STAEformer | 5.77 | 9.68 | 14.55 | 6.16 | 10.59 | 16.26 | 6.89 | 12.13 | 19.20 | 3.68 | 6.61 | 8.71 | 3.93 | 7.12 | 9.35 | 4.39 | 8.06 | 10.32 |
| **ASTCL20** | 2.90 | 5.84 | 7.67 | 3.52 | 7.26 | 9.84 | 4.65 | 9.36 | 13.60 | 1.62 | 3.57 | 3.42 | 2.16 | 4.98 | 4.85 | 2.99 | 6.93 | 7.03 |
| **ASTCL50** | 2.89 | 5.74 | 7.72 | 3.50 | 7.18 | 9.76 | 4.61 | 9.21 | 13.69 | 1.57 | 3.44 | 3.37 | 2.10 | 4.87 | 4.65 | 2.94 | 6.84 | 6.77 |
| **ASTCL** | **2.81** | **5.71** | **7.50** | **3.43** | **7.16** | **9.65** | **4.46** | **9.10** | **13.17** | **1.56** | **3.42** | **3.33** | **2.07** | **4.83** | **4.62** | **2.89** | **6.71** | **6.74** |

为了进一步分析 ASTCL 在复杂交通场景下的预测能力，本文在交通图传感器不足的情景下对模型性能进行了评估。这种情况一般发生在智慧城市早期发展阶段，由于交通传感器部署不足，许多城市地区会面临数据稀缺的问题。少样本交通预测可以有效评估模型在有限数据条件下的适应能力。为了模拟传感器稀疏的交通网络，本方法将在四个数据集中随机移除大部分的交通节点，模型需要在仅保留一小部分交通传感器数据的条件下进行训练，并预测整个网络的交通模式。为确保公平性，实验中的验证集和测试集以及其他设置与基准模型保持一致，这意味着模型不仅在未来的时间序列上进行评估，还需要在未见过的传感器上进行预测。这种实验设计显著增加了预测难度，并且在交通预测领域尚未得到充分研究。在表 15和16 中，分别用 ASTCL50和 ASTCL20表示仅保留 50%和20%可用交通传感器训练的模型。例如，在PEMS08数据集上，ASTCL20的训练集仅包含来自34个（占PEMS08的20%）交通传感器的单日交通数据。在实验结果表明，在少样本设置下，ASTCL20和ASTCL20相比基准模型仍表现出优越的预测性能。而相比于 ASTCL，ASTCL50和ASTCL20的平均性能仅分别下降了 2% 和 5%，这一结果进一步证明了 ASTCL 模型在少样本交通预测中的适应性和鲁棒性。

### 4.3.3 消融实验

为了验证 ASTCL 中各个组件的有效性，本文设计了如下消融实验：w/o STA： 去除联合时空注意力机制；w/o STC：去除联合时空卷积；w/oDSG：去除与动态序列图相关的模块； w/o IN：去除实例归一化。表17展示了在 PEMS04 和 PEMS08 数据集上预测时长 60 分钟的实验结果。结果表明所有变体的性能均低于完整的 ASTCL 模型，证明了每个组件对预测重要性。其中，去除联合时空注意力机制会导致模型性能显著下降，这说明长距离的时空依赖对预测起关键性作用。本文还探讨了联合注意力和联合卷积种各个结构的作用，联合时空学习模块利用三种操作来从不同角度建模交通网络：空间建模（S）、时间建模（T）和时空建模（ST）。如图 18所示，整体的联合注意力和联合卷积模块在所有变体中表现出了最佳性能。通过结合空间模块和时间模块，模型能够比单独使用任一模块获得更好的预测精度。综上所述，本方法所提出的联合时空学习能够同时建模复杂的时空相关性，并有效提升了预测准确率。

表 17 消融实验

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 方法 | PEMS04 | | | PEMS08 | | |
|  | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| w/o STA | 35.54 | 53.57 | 23.50 | 32.88 | 47.45 | 24.11 |
| w/o STC | 33.94 | 51.53 | 22.15 | 28.60 | 42.40 | 21.77 |
| w/o DSG | 33.71 | 51.31 | 21.72 | 26.54 | 39.86 | 17.27 |
| w/o IN | 33.92 | 51.89 | 22.30 | 27.54 | 40.94 | 21.20 |
| ASTCL | **32.83** | **50.01** | **21.36** | **25.77** | **39.46** | **16.86** |

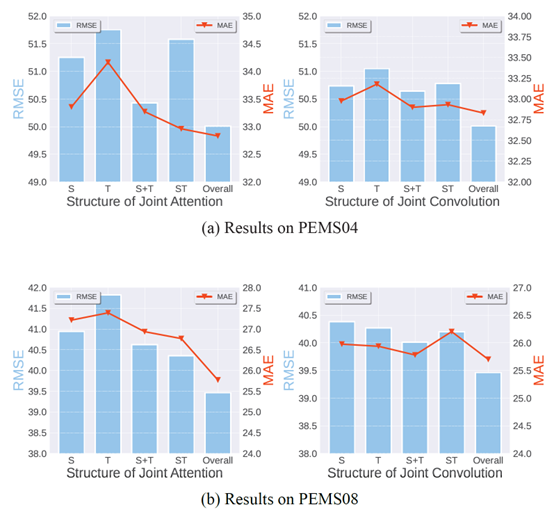


图 19 时空联合学习分析

### 4.3.4 参数分析

关于参数分析中模型层数和特征维度的实验结果如图 19所示。在 PEMS04 和 METR-LA 数据集上，采用2 层架构能够在保证高性能的同时，有效缓解过拟合并降低了计算成本。相比之下，对于 PEMS08 和 PEMS-BAY 数据集，较深的 5 层模型可以进一步降低 RMSE，表明这些数据集可能需要更复杂的模型结构来捕捉其内在特征。此外，关于特征维度的选择，较大的维度通常能够增强模型的表示能力，但同时会增加参数量，导致更高的计算成本和潜在的过拟合风险。基于实验结果，本文在 METR-LA 数据集中选择特征维度为 16，而在 PEMS04、 PEMS08 和 PEMS-BAY 数据集中选择 32。这种差异化的参数配置不仅反映了不同数据集的特性，也在性能与计算效率之间实现了平衡。

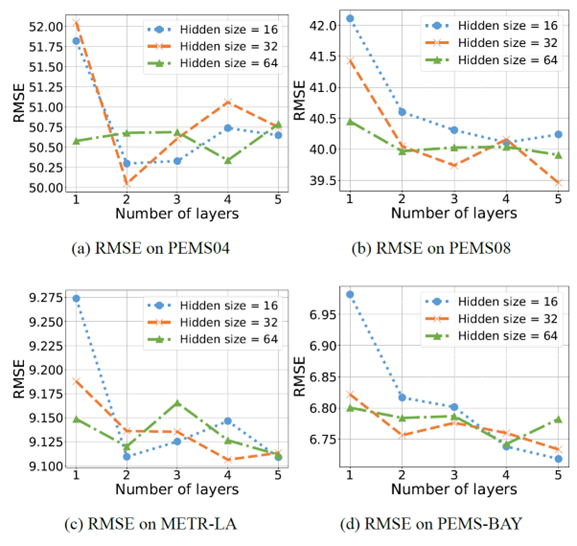


图 20 参数分析

### 4.3.5 鲁棒性分析

本节从跨城市和跨域预测两个实验角度来分析 ASTCL 在复杂交通预测场景中的鲁棒性。

表 18中展示了 ASTCL 在跨城市交通预测任务中的实验结果，从跨城市迁移的角度来论证 ASTCL 的鲁棒性。迁移实验的设计包括包括在源数据集上预训练模型，并将其直接迁移到目标数据集进行预测。ASTCL 与两种先进的迁移基模型 TransGTR 和 pFedCTP 进行了对比。根据其原始训练策略，首先在完整的源数据集上训练 TransGTR 和 pFedCTP，然后使用目标数据集中的一天数据进行微 调。相比之下， ASTCL 仅使用源数据集中的一天数据进行预训练，并直接在目标数据集上进行评估，无需微调或额外操作。这种训练模型的差异凸显了 ASTCL 的高效性，因为它避免了大量的预训练或微调过程。实验结果表明，ASTCL 在跨城市迁移预测任务优于 TransGTR 和 pFedCTP，并实现了最多 10% 的平均性能提升,这一结果充分证明了 ASTCL 在预测交通流量和交通速度方面的可迁移性和鲁棒性。特别值得注意的是，ASTCL 可以在任何交通数据集上进行训练，并直接应用于其他城市的类似预测任务，无需额外的训练过程，从而显著降低了计算成本。

表 18 跨城市预测实验结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 源数据 | 目标数据 | 方法 | 15 分钟 30 分钟 60 分钟 | | | | | | | | |
| MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| METR-LA | PEMS-BAY | TransGTR  pFedGTP  ASTCL | 2.25 1.78  **1.52** | 5.44 3.45  **3.31** | 6.87 3.89  **3.25** | 2.81 2.32  **2.05** | 6.50 4.77  **4.68** | 8.35 5.23  **4.59** | 3.46 3.21  **2.86** | 7.71 6.58  **6.51** | 10.12 7.38  **6.74** |
| PEMS-BAY | METR-LA | TransGTR  pFedGTP  ASTCL | 3.52 3.15  **2.88** | 7.34 5.78  **5.75** | 9.69 8.42  **7.62** | 4.35 3.93  **3.51** | 8.96 7.23  **7.22** | 12.65 11.23  **9.60** | 5.21 5.06  **4.62** | 10.10 9.41  **9.34** | 15.95 15.30  **13.16** |
| PEMS04 | PEMS08 | TransGTR  pFedGTP  ASTCL | 18.03 18.48  **17.15** | 26.83 27.81  **26.66** | \ 12.02  **11.97** | 20.77 21.19  **20.11** | 33.96 33.40  **33.28** | \ 14.29  **14.10** | 27.68 27.59  **27.50** | 41.93 42.29  **41.74** | \ 22.46  **22.30** |
| PEMS08 | PEMS04 | TransGTR  pFedGTP  ASTCL | 21.69 22.30  **21.67** | 34.51 35.03  **34.25** | 14.40  \  **13.77** | 25.74 26.25  **25.30** | 40.38 40.76  **39.56** | 17.50  \  **17.12** | **32.94**  34.57 32.98 | 50.49 51.97  **50.24** | 22.82  \  **22.44** |

表 19 跨域预测实验结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 源数据 | 目标数据 | 方法 | MAE | RMSE | MAPE |
| METR-LA | PEMS04 | ASTCL  pFedGTP FT-ASTCL | 47.92 35.22  **33.05** | 66.48 52.39  **51.69** | 40.24  \  **21.17** |
| PEMS08 | ASTCL  pFedGTP FT-ASTCL | 41.34 28.00  **26.66** | 57.51 41.45  **41.01** | 32.03 23.28  **16.43** |
| PEMS-BAY | PEMS04 | ASTCL  pFedGTP FT-ASTCL | 45.42 35.23  **32.87** | 62.99 52.12  **51.92** | 37.67  \  **21.24** |
| PEMS08 | ASTCL  pFedGTP FT-ASTCL | 38.31 27.68  **26.41** | 53.17 41.52  **40.93** | 30.93 23.58  **16.65** |

第二类实验旨在从跨域预测的角度来评估 ASTCL 在不同预测领域中的鲁棒性和泛化能力。对于跨城市预测而言，其主要考察模型在不同城市间迁移的能力，仍针对相同的预测变量；而跨领域评估则侧重于模型适应不同类型预测变量的能力。具体而言，本方法将在一种交通预测任务（例如，交通速度预测）上训练的模型迁移到另一种预测任务中（例如，交通流量预测）。由于不同预测变量之间固有的异质性，直接将预训练模型迁移到新的预测任务上通常会导致性能下降。为此，本文引入了 FT-ASTCL 变体，通过在目标变量数据集上进行额外的 20 轮微调，以提升预训练模型的性能。实验结果如表 19所示，通过简单的微调，FT-ASTCL 的性能提升至与在目标数据集上全量训练模型相当的水平。同时，ASTCL 在跨域场景中的预测表现要优于 pFedCTP，证明了 ASTCL 在不同交 通预测任务中的广泛适用性和良好表现，为其在实际应用中的灵活部署提供了有力支持。通过简单的微调，ASTCL 能够有效克服跨领域迁移中的性能下降问题，为多任务交通预测提供了可能的解决方案。

## 4.4 本章小结

本研究中针对交通预测的外部适应性提出 ASTCL 模型，旨在用单日的的交通数据学习到高效的时空表征以适应不同的交通预测环境。具体而言，ASTCL 通过构建目标节点的自适应时空上下文，并结合联合注意力机制与联合卷积操作，有效捕捉了交通数据中复杂的时空相关性。此外，为了建模交通模式的动态性，本研究还提出了一种动态序列图结构，进一步增强了模型对交通流变化的表征能力。在四个真实交通数据集上的综合实验表明，ASTCL 在预测性能上要优于现有方法，并在不同数据集上均表现出较强的鲁棒性和适应性。同时， ASTCL 不仅能够直接应用于多种交通数据集，还可以灵活预测多种交通属性，展现了较为广泛的适用性。

# 5 结论（结论和建议）

本研究聚焦于城市流量预测问题，尤其是在面对复杂的时空依赖关系和动态变化时，如何提升预测精度和稳定性。围绕缺失数据补全、轻量化训练策略和动态时空建模三个方向展开。

在本研究中，针对城市细粒度流量推断问题提出了一种名为 UrbanSTC 的时空对比自监督的方法。所提出的模型可以从城市流量中提取丰富的时空特征，能够从空间和时间相关性两个方面建立自监督辅助任务。对于空间相关性，区域对比和空间超分辨率推理网络为捕捉区域级别流量和流量推断模式之间的相似性做出了巨大的贡献。此外，本文设计了两种基于时间属性的采样策略。模型的整体结构遵循白监督训练模式:预训练&微调。通过精心设计的自监督辅助任务，使简单的网络可以从复杂的城市流量中学习高级表征能力。对两个真实世界的数据集进行了详细的实验来比较 UrbanSTC 和其他最先进方法之间的性能。结果不仅表明所提方法优于其他所有方法，而且在训练数据有限时依旧表现出较高性能。

针对缺失数据补全问题本研究提出了一种基于时空注意力学习的城市流量预测模型 UrbanSTA，解决了通过历史时刻的带有缺失值的流量图预测当前时刻的完整流量图问题。提出了一种基于时空分离的 Transformer 缺失重构策略的细粒度推断框架 STA-FIN,解决了在带有缺失值的粗粒度流量图的前提下推断细粒度流量图的问题。针对流量数据的邻近性和时间连续性特点设计了时空分离的注意力机制，同时考虑到其空间旋转性、时间周期性以及地区兴趣点因素分别提出了时空和兴趣点吸引约束。时空注意力机制和时空吸引约束全部具有高度的独立性、可自由移植到任何 Transformer 结构内。所提出的两个模型在智慧城市交通流量领域内具有较强的时空特征挖掘，在不依赖外部的数据集的情况下，依然能达到先进的水平。

本研究第三项工作提出了种自适应时空上文学习框架来探索由外部因素导致的时空模式变化问题。该框架的核心任务是提升模型对不同场景交通预测任务

的快速适应性，通过少样本的模型训练以获得稳定和高精度的预测效果。具体而言，本研究设计了一种节点级交通预测任务，并为每个交通节点构建自适应时空上下文，以丰富时空信息。这一设计使得模型能够更好地捕捉局部和全局的时空关系，从而提升预测的准确性。进一步地，本研究引人了时空联合学习模块，包括联合卷积和联合注意力机制，从局部与全局视角协同建模复杂的时空关系。联合卷积模块能够有效捕捉局部时空特征，而联合注意力机制则能够从全局视角挖掘交通节点之间的依赖关系。此外，本研究还根据各交通节点之间的特征相似性构建动态序列图，利用图神经网络来捕获各节点间的依赖关系。实验结果表明ASTCL,模型在稀疏图预测、跨城市预测和跨域预测等多个交通预测任务中均表现出色，具有快速适应不同交通场景的能力。

在未来的研究中，本研究将进一步提高城市流量预测的精度和可靠性。随着智能城市建设的不断推进，未来的城市流量预测可能面临更为复杂的场景，例如交通事故、突发事件等对流量的影响。如何在这些复杂场景下依然保持高精度的预测，将是未来研究的重要方向。此外，针对一些新兴智能城市的建设，如何在数据稀缺的情况下，依托少量的历史数据进行流量预测，也是一个亟待解决的挑战。未来，本研究将进一步提取复杂时空关系并更精确预测细粒度流量，促进智能交通系统的建设与优化，为智慧城市的发展提供强有力的支持。

# 参考文献

1. Zhang J, Wang F Y, Wang K, et al. Data-driven intelligent transportation systems: A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(4): 1624-1639.
2. Guo S, Lin Y, Feng N, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 922-929.
3. Lan S, Ma Y, Huang W, et al. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting[C]//International conference on machine learning. PMLR, 2022: 11906-11917.
4. Drouin A, Marcotte É, Chapados N. Tactis: Transformer-attentional copulas for time series[C]//International Conference on Machine Learning. PMLR, 2022: 5447-5493.
5. Gong Y, Li Z, Zhang J, et al. Online spatio-temporal crowd flow distribution prediction for complex metro system[J]. IEEE Transactions on knowledge and data engineering, 2020, 34(2): 865-880.
6. Guo K, Hu Y, Qian Z, et al. Dynamic graph convolution network for traffic forecasting based on latent network of Laplace matrix estimation[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(2): 1009-1018.
7. Jiang J, Han C, Zhao W X, et al. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(4): 4365-4373.
8. Wu Z, Pan S, Long G, et al. Graph wavenet for deep spatial-temporal graph modeling[J]. arxiv preprint arxiv:1906.00121, 2019.
9. Xie P, Ma M, Li T, et al. Spatio-temporal dynamic graph relation learning for urban metro flow prediction[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(10): 9973-9984.
10. Han L, Du B, Sun L, et al. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 547-555.
11. Li M, Zhu Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4189-4196.
12. 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. 2020, 34(01): 914-921.
13. Qu H, Gong Y, Chen M, et al. Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(8): 8008-8023.
14. Bai L, Yao L, Kanhere S, et al. Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting[J]. arXiv preprint arXiv:1905.10069, 2019.
15. 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. 2020, 34(01): 1234-1241.
16. Liang Y, Ouyang K, Sun J, et al. Fine-grained urban flow prediction[C]//Proceedings of the Web Conference 2021. 2021: 1833-1845.
17. Ashok A, Marcotte É, Zantedeschi V, et al. TACTis-2: Better, faster, simpler attentional copulas for multivariate time series[J]. arXiv preprint arXiv:2310.01327, 2023.
18. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
19. Ma Q, Zhang Z, Zhao X, et al. Rethinking sensors modeling: Hierarchical information enhanced traffic forecasting[C]//Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 1756-1765.
20. Bai L, Yao L, Li C, et al. Adaptive graph convolutional recurrent network for traffic forecasting[J]. Advances in neural information processing systems, 2020, 33: 17804-17815.
21. Zhang X, Huang C, Xu Y, et al. Traffic flow forecasting with spatial-temporal graph diffusion network[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(17): 15008-15015.
22. Guo K, Hu Y, Sun Y, et al. Hierarchical graph convolution network for traffic forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(1): 151-159.
23. Liang Y, Ouyang K, Jing L, et al. Urbanfm: Inferring fine-grained urban flows[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 3132-3142.
24. Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: Probabilistic forecasting with autoregressive recurrent networks[J]. International journal of forecasting, 2020, 36(3): 1181-1191.
25. Passalis N, Tefas A, Kanniainen J, et al. Deep adaptive input normalization for time series forecasting[J]. IEEE transactions on neural networks and learning systems, 2019, 31(9): 3760-3765.
26. Xie P, Li T, Liu J, et al. Urban flow prediction from spatiotemporal data using machine learning: A survey[J]. Information Fusion, 2020, 59: 1-12.
27. Tedjopurnomo D A, Bao Z, Zheng B, et al. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(4): 1544-1561.
28. Moorthy C K, Ratcliffe B G. Short term traffic forecasting using time series methods[J]. Transportation planning and technology, 1988, 12(1): 45-56.
29. Moreira-Matias L, Gama J, Ferreira M, et al. Predicting taxi–passenger demand using streaming data[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(3): 1393-1402.
30. Shekhar S, Williams B M. Adaptive seasonal time series models for forecasting short-term traffic flow[J]. Transportation Research Record, 2007, 2024(1): 116-125.
31. Zhang J, Zheng Y, Qi D, et al. DNN-based prediction model for spatio-temporal data[C]//Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems. 2016: 1-4.
32. Zhao L, Gao M, Wang Z. St-gsp: Spatial-temporal global semantic representation learning for urban flow prediction[C]//Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. 2022: 1443-1451.
33. Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1).
34. Yao H, Wu F, Ke J, et al. Deep multi-view spatial-temporal network for taxi demand prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).
35. Salinas D, Bohlke-Schneider M, Callot L, et al. High-dimensional multivariate forecasting with low-rank gaussian copula processes[J]. Advances in neural information processing systems, 2019, 32.
36. Pan Z, Wang Z, Wang W, et al. Matrix factorization for spatio-temporal neural networks with applications to urban flow prediction[C]//Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 2683-2691.
37. Papamakarios G, Pavlakou T, Murray I. Masked autoregressive flow for density estimation[J]. Advances in neural information processing systems, 2017, 30.
38. 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.
39. Rezende D, Mohamed S. Variational inference with normalizing flows[C]//International conference on machine learning. PMLR, 2015: 1530-1538.
40. Liu L, Zhen J, Li G, et al. Dynamic spatial-temporal representation learning for traffic flow prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(11): 7169-7183.
41. Ji J, Wang J, Huang C, et al. Spatio-temporal self-supervised learning for traffic flow prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(4): 4356-4364.
42. Zhang X, Huang C, Xu Y, et al. Traffic flow forecasting with spatial-temporal graph diffusion network[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(17): 15008-15015.
43. Yao H, Tang X, Wei H, et al. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 5668-5675.
44. Ke S, Pan Z, He T, et al. AutoSTG+: An automatic framework to discover the optimal network for spatio-temporal graph prediction[J]. Artificial Intelligence, 2023, 318: 103899.
45. Liu X, Liang Y, Huang C, et al. Do we really need graph neural networks for traffic forecasting?[J]. arXiv preprint arXiv:2301.12603, 2023.
46. Yao K, Liang J, Liang J, et al. Multi-view graph convolutional networks with attention mechanism[J]. Artificial Intelligence, 2022, 307: 103708.
47. He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-9738.
48. Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.
49. Rasul K, Sheikh A S, Schuster I, et al. Multivariate probabilistic time series forecasting via conditioned normalizing flows[J]. arXiv preprint arXiv:2002.06103, 2020.
50. Yang S, Xiao W, Zhang M, et al. Image data augmentation for deep learning: A survey[J]. arXiv preprint arXiv:2204.08610, 2022.
51. Wei J, Zou K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks[J]. arXiv preprint arXiv:1901.11196, 2019.
52. Zhu Y, Xu Y, Yu F, et al. Graph contrastive learning with adaptive augmentation[C]//Proceedings of the web conference 2021. 2021: 2069-2080.
53. Tabak E G, Turner C V. A family of nonparametric density estimation algorithms[J]. Communications on Pure and Applied Mathematics, 2013, 66(2): 145-164.
54. Zhang H, Lin S, Liu W, et al. Iterative graph self-distillation[J]. IEEE Transactions on Knowledge and Data Engineering, 2023.
55. Tashiro Y, Song J, Song Y, et al. Csdi: Conditional score-based diffusion models for probabilistic time series imputation[J]. Advances in Neural Information Processing Systems, 2021, 34: 24804-24816.
56. Liu X, Liang Y, Huang C, et al. When do contrastive learning signals help spatio-temporal graph forecasting?[C]//Proceedings of the 30th international conference on advances in geographic information systems. 2022: 1-12.
57. Zhang Y, Zhu H, Song Z, et al. COSTA: covariance-preserving feature augmentation for graph contrastive learning[C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022: 2524-2534.
58. Rasul K, Seward C, Schuster I, et al. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting[C]//International Conference on Machine Learning. PMLR, 2021: 8857-8868.
59. Zhou F, Li L, Zhong T, et al. Enhancing urban flow maps via neural odes[C]//Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI} 2020. 2020.
60. Dong C, Loy C C, He K, et al. Image super-resolution using deep convolutional networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 38(2): 295-307.
61. Chen Z, Zhang Y, Gu J, et al. Dual aggregation transformer for image super-resolution[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023: 12312-12321.
62. Wen H, Lin Y, Xia Y, et al. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models[C]//Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems. 2023: 1-12.
63. Rosenfeld A, Maksimov O, Kraus S. When security games hit traffic: A deployed optimal traffic enforcement system[J]. Artificial Intelligence, 2020, 289: 103381.
64. Oxford Analytica. Deadly stampede in Seoul will leave long legacy[J]. Emerald Expert Briefings, 2022 (oxan-es).
65. Gong Y, Li Z, Zhang J, et al. Potential passenger flow prediction: A novel study for urban transportation development[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 4020-4027.
66. Gong Y, Li Z, Zhang J, et al. Network-wide crowd flow prediction of sydney trains via customized online non-negative matrix factorization[C]//Proceedings of the 27th ACM international conference on information and knowledge management. 2018: 1243-1252.
67. Guo S, Lin Y, Li S, et al. Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10): 3913-3926.
68. Xu Z, Wang Y, Long M, et al. PredCNN: Predictive Learning with Cascade Convolutions[C]//IJCAI. 2018: 2940-2947.
69. Dai E, Chen J. Graph-augmented normalizing flows for anomaly detection of multiple time series[J]. arXiv preprint arXiv:2202.07857, 2022.
70. Guan S, He Z, Ma S, et al. Conditional normalizing flow for multivariate time series anomaly detection[J]. ISA transactions, 2023, 143: 231-243.
71. Liu J, Kumar A, Ba J, et al. Graph normalizing flows[J]. Advances in Neural Information Processing Systems, 2019, 32.
72. Wiese M, Knobloch R, Korn R. Copula & marginal flows: Disentangling the marginal from its joint[J]. arXiv preprint arXiv:1907.03361, 2019.
73. Li Z, Zhang J, Wu Q, et al. Sample adaptive multiple kernel learning for failure prediction of railway points[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 2848-2856.
74. Nelsen R B. An introduction to copulas[M]. Springer, 2006.
75. Zhang J, Zheng Y, Qi D, et al. Predicting citywide crowd flows using deep spatio-temporal residual networks[J]. Artificial Intelligence, 2018, 259: 147-166.
76. Zheng C, Fan X, Wang C, et al. Increase: Inductive graph representation learning for spatio-temporal kriging[C]//Proceedings of the ACM Web Conference 2023. 2023: 673-683.
77. Liu Z, Mao H, Wu C Y, et al. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11976-11986.
78. Feichtenhofer C, Fan H, Malik J, et al. Slowfast networks for video recognition[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6202-6211.
79. 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. 2021, 35(12): 11106-11115.
80. Wu H, Xu J, Wang J, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting[J]. Advances in neural information processing systems, 2021, 34: 22419-22430.
81. 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. 2019, 33(01): 3656-3663.
82. Xu J, De Mello S, Liu S, et al. Groupvit: Semantic segmentation emerges from text supervision[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 18134-18144.
83. Dosovitskiy A. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.
84. Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax[J]. arXiv preprint arXiv:1611.01144, 2016.
85. Maddison C J, Mnih A, Teh Y W. The concrete distribution: A continuous relaxation of discrete random variables[J]. arXiv preprint arXiv:1611.00712, 2016.
86. Scott S L, Varian H R. Bayesian variable selection for nowcasting economic time series[M]. National Bureau of Economic Research, 2013.
87. Woo G, Liu C, Sahoo D, et al. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting[J]. arXiv preprint arXiv:2202.01575, 2022.
88. Kingma D P. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
89. Wang J, Jiang J, Jiang W, et al. Libcity: An open library for traffic prediction[C]//Proceedings of the 29th international conference on advances in geographic information systems. 2021: 145-148.
90. Le Guen V, Thome N. Probabilistic time series forecasting with shape and temporal diversity[J]. Advances in neural information processing systems, 2020, 33: 4427-4440.
91. Zheng C, Fan X, Pan S, et al. Spatio-temporal joint graph convolutional networks for traffic forecasting[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 36(1): 372-385.
92. Lai W S, Huang J B, Ahuja N, et al. Deep laplacian pyramid networks for fast and accurate super-resolution[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 624-632.
93. Kim J, Lee J K, Lee K M. Accurate image super-resolution using very deep convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1646-1654.
94. Ledig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4681-4690.
95. Hui Z, Gao X, Yang Y, et al. Lightweight image super-resolution with information multi-distillation network[C]//Proceedings of the 27th acm international conference on multimedia. 2019: 2024-2032.
96. Zhou H, Li J, Zhang S, et al. Expanding the prediction capacity in long sequence time-series forecasting[J]. Artificial Intelligence, 2023, 318: 103886.
97. Johansen S. Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models[J]. Econometrica: journal of the Econometric Society, 1991: 1551-1580.
98. Williams B M, Durvasula P K, Brown D E. Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models[J]. Transportation Research Record, 1998, 1644(1): 132-141.
99. Box G E P, Jenkins G M, Reinsel G C, et al. Time series analysis: forecasting and control[M]. John Wiley & Sons, 2015.
100. Vaswani A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.