插图清单

图2-1城市细粒度流量推断

图2-2 UrbanSTC整体架构

图2-3 空间超分辨率推理网络预训练

图2-4 外部信息融合

图2-5消融实验可视化

图2-6 P1数据集上硬采样和权重采样性能比较

图2-7不同通道数实验结果

图2-8阈值λ影响曲线图

图2-9 模型参数研究

附表清单

表2-1 P1不同比例训练数据结果

表2-2 P2不同比例训练数据结果

表2-3 P3不同比例训练数据结果

表2-4 P4不同比例训练数据结果

表2-5 BikeNYC不同比例训练数据结果

表2-6消融实验

表2-7端到端和两阶段对比

表2-8 TaxiBJ外部因素实验结果1

表2-9 TaxiBJ外部因素实验结果2

表2-10 P1数据集上评估模型的效率

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2-1(a)和2-1(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2-1(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。

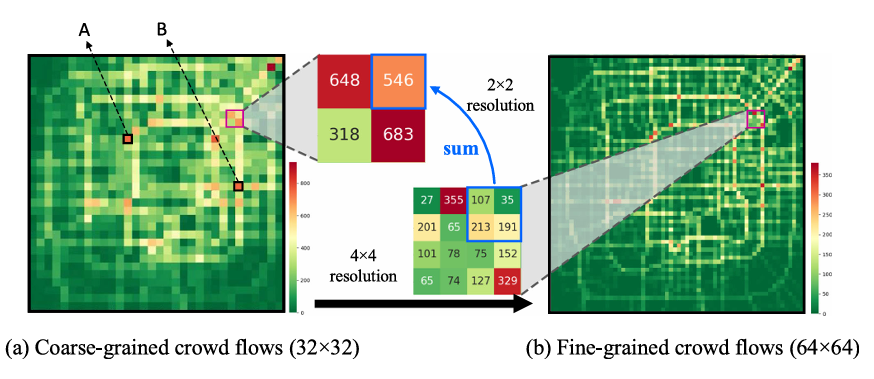


图2-1城市细粒度流量推断

(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。

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2-2所示。

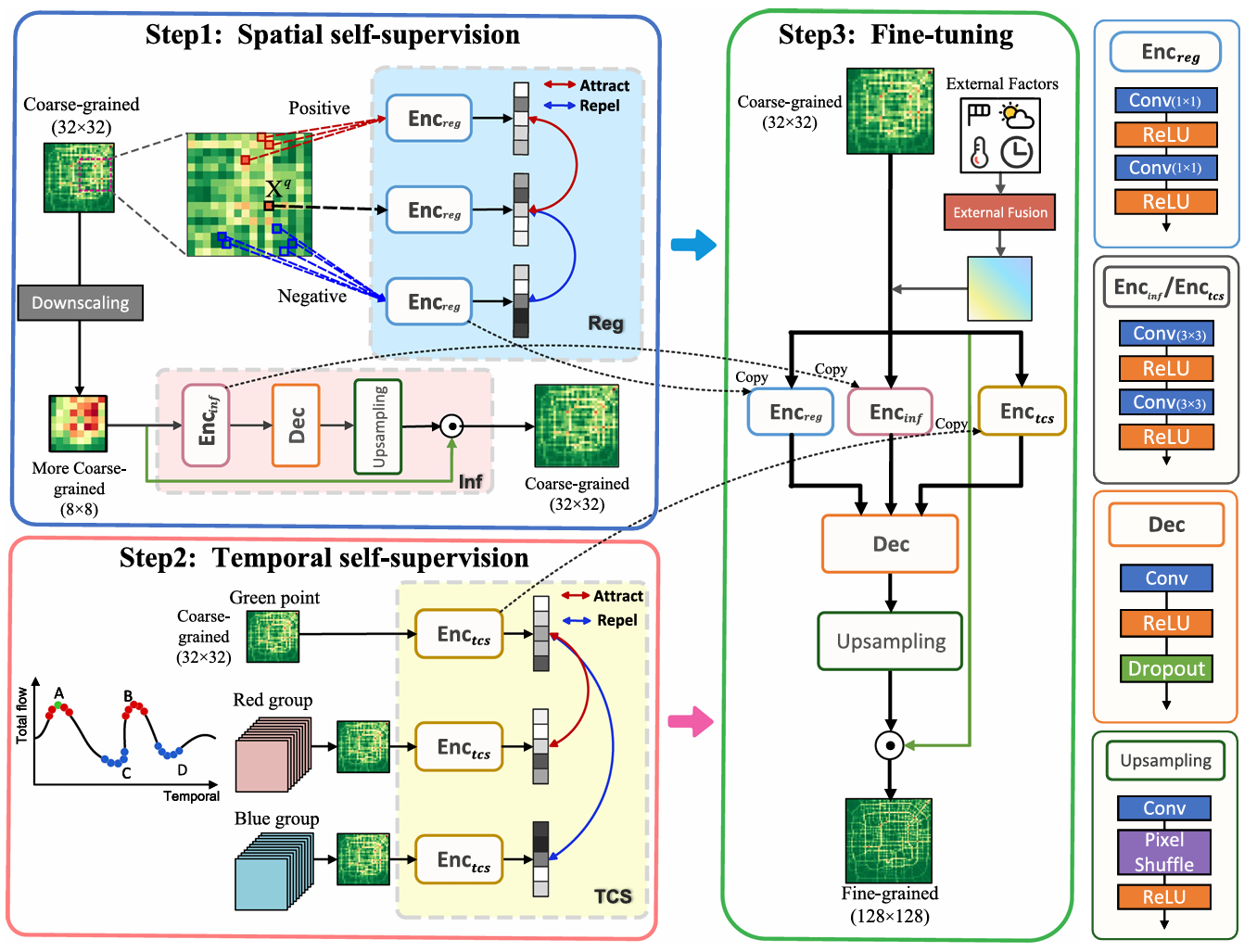


图2-2 UrbanSTC整体架构

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。

2.1.1.1 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2-1(a)所示，每个区域表示观测到的流量。

2.1.1.2超区域与子区域

图2-1(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]。

2.1.1.3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。

（1）

其并且。

2.1.1.4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2-2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：

(2)

（3）

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：

(4)

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2-3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。

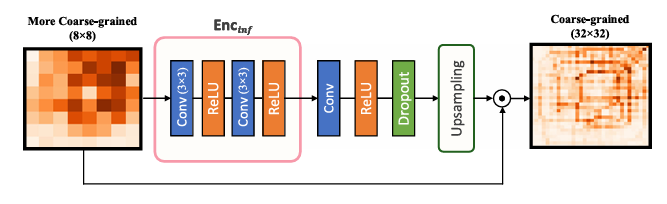


图2-3 空间超分辨率推理网络预训练

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2-3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

（5）

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2-2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:

（6）

其中是当前时刻的粗粒度流量图，是其他时刻流量图。如图2-2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； |

权重采样：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：

（7）

（8）

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：

（9）

（10）

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()。

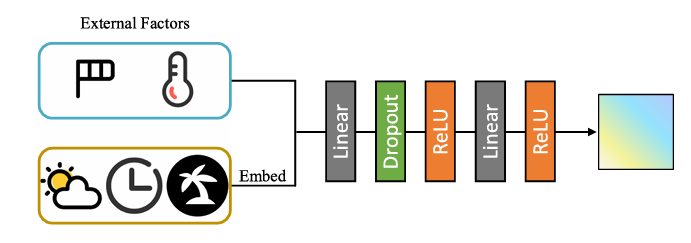


图2-4 外部信息融合

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2-4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2-2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：

(11)

(12)

(13)

(14)

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：

(15)

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：

(16)

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)。

(17)

(18)

(19)

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2-1到2-4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2-1 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2-2 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2-3 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2-4 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2-5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2-5 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2-6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2-6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。

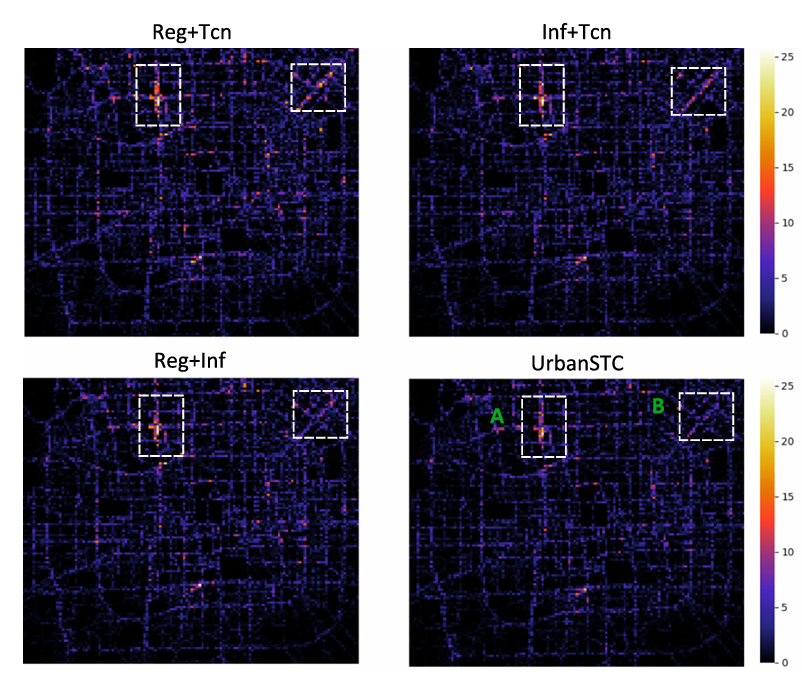
图2-5消融实验可视化

表2-6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：最好的结果为粗体显示。

为了更好地呈现消融实验结果，本研究在图2-5中展示了一些可视化对比。图2-5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2-7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2-7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

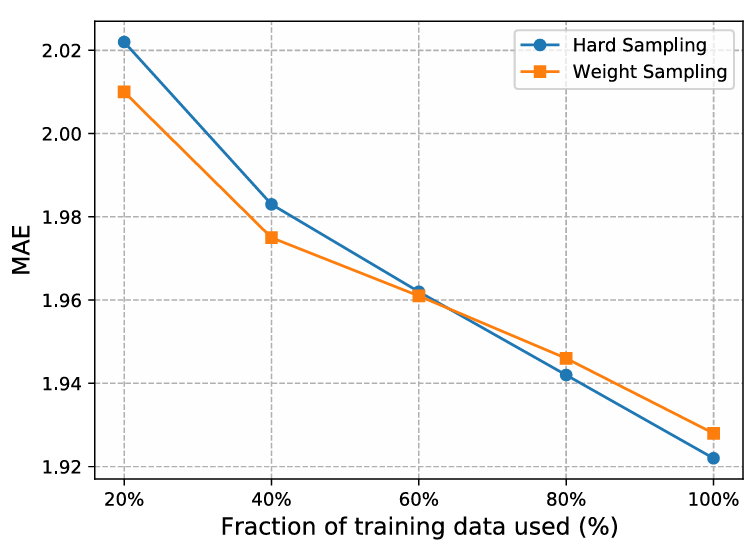
为了评估硬采样和权重采样的效果，本研究展示了在TaxiBJ-P1的实验结果。如图2-6所示，当使用的训练数据低于60%时，权重采样优于硬采样。这是因为权重采样方 法可以综合使用前Top-K个相关样本，而硬采样只选择最相似或者最不相似的数据。随 着训练数据量的增加，硬采样开始表现出比权重采样更好的性能。当训练数据较小时, 很难找到全局最相似的样本，但可以使用Top-K相似样本代替。随着训练数据量的增加 更容易找到最相似的样本，硬采样方法可以获得更好的结果。因此在不同的训练场景中 可以采用两种方法相结合。

图2-6 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2-8和2-9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2-8 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2-9 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

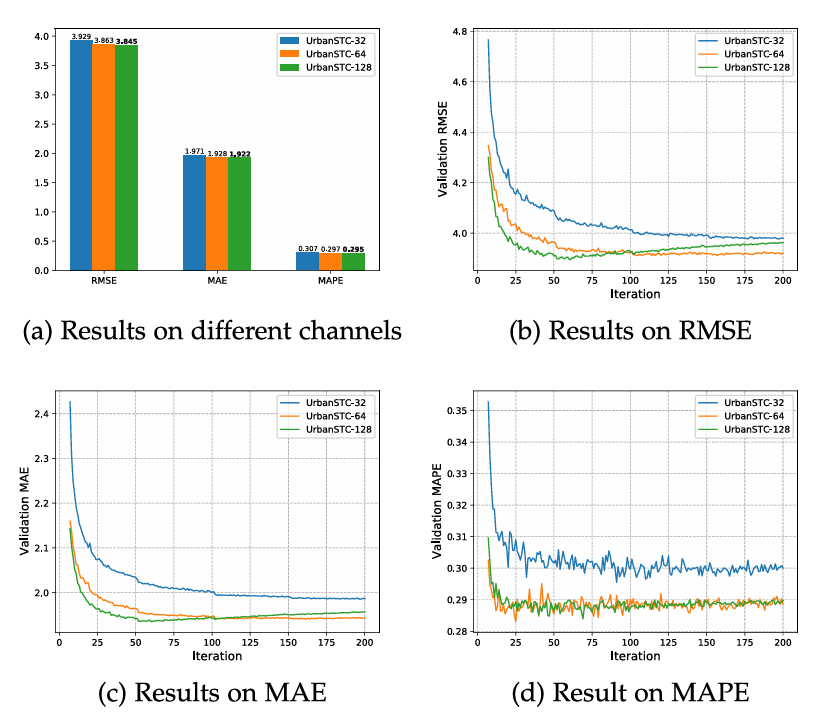
在本节中，本研究尝试探索UrbanSTC方法在各种环境中的学习能力。对比不同的通 道数(32、64、128)，如图2-7所示。图2-7(a)表明通道数越多，UrbanSTC具有更好的性能。此外，图2-7(b)(c)(d)表明更大的通道数量可以提高学习收敛速度。

图2-7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2-8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。

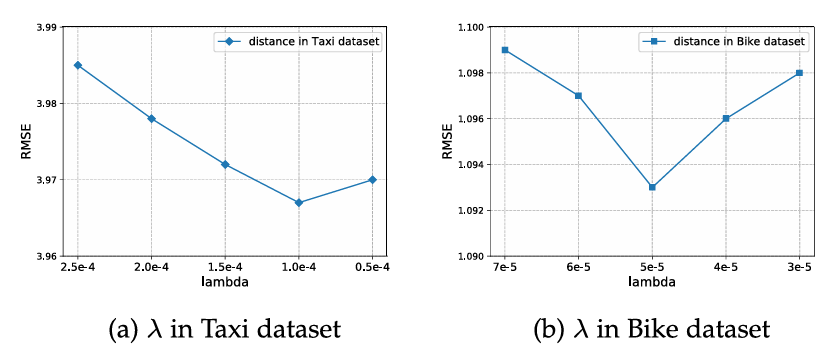


图2-8阈值λ影响曲线

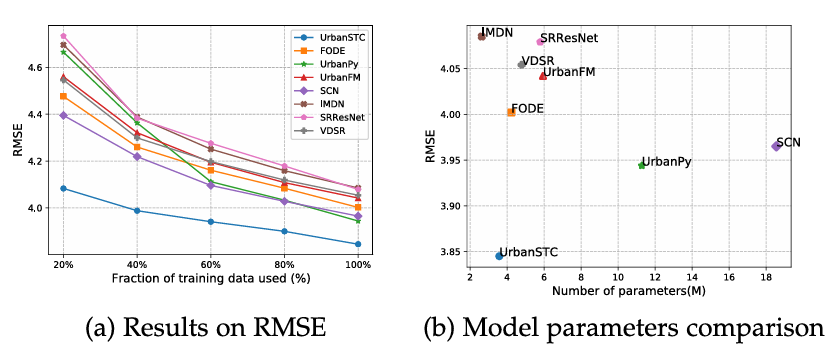
对于模型参数分析，图2-9(a)表示在不同比例的训练数据情况下UrbanSTC比其他 基线模型具有较好的结果。图2-9(b)表示传统的图像超分辨率方法，例如：IMDN、VDSR 和SRResNet,由于固有的结构性差异，不适用于FUFI问题。尽管SRResNet和UrbanFM 具有相似的结构，但UrbanFM中的M2-Normalization层有助于解决FUFI问题。UrbanPy 使用级联结构将原始任务分解为多个子任务来预测细粒度的城市流量，这会导致计算复 杂度的增加。FODE利用OED模块替换UrbanFM中的ResNet结构。因为上述模块可 以看成一个连续ODE算子的离散化，能够提高收敛速度，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2-9 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2-10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2-10 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2-10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2-3中的实验结果。

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。

参考文献

1. Yuxuan Liang, Kun Ouyang, Lin Jing, et al. Urbanftn: Inferring fine-grained urban flows[C]. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019, 3132-3142.
2. Fan Zhou, Liang Li, Ting Zhong, et al. Enhancing urban flow maps via neural ODEs[C]. Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2021, 1295-1302.
3. Yongshun Gong, Zhibin Li, Jian Zhang, et al. Online Spatio-temporal Crowd Flow Distribution Prediction for Complex Metro System[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, PP(99):1-1.
4. Kun Ouyang, Yuxuan Liang, Ye Liu, et al. Fine-Grained Urban Flow Inference[J]. arXiv prints, 2020: arXiv: 2002.0231 &
5. Sangkeon Lee, Heeseorain Kwon, Heeah Cho, et al. International case studies of smart cities: Anyang, Republic of Korea[J]. Institutions for Development Sector Fiscal and Municipal Management Division, 2016.
6. Qiang Zhou, Jingjing Gu, Xinjiang Lu, et al. Modeling heterogeneous relations across multiple modes for potential crowd flow prediction[J]. arXiv preprint arXiv:2101.06954, 2021.
7. Xi Yang, Suining He. Towards dynamic urban bike usage prediction for station network reconfiguration[J]. arXiv preprint arXiv:200&0731& 2020.

[8卩ingjing Gu, Qiang Zhou, Jingyuan Mng, et al. Exploiting Interpretable Patterns for Flow Prediction in Dockless Bike Sharing Systems]J]. IEEE Transactions on Knowledge and Data Engineering, 2020, PP(99):1-1.

1. Mingwei Lin, Chao Huang, Zeshui Xu. MULTIMOORA based MCDM model for site selection of car sharing station under picture fuzzy environment [J]. Sustainable cities and society, 2020, 53: 101873.
2. Yongshun Gong, Zhibin Li, Jian Zhang, 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.
3. Jianrui Cai, Shuhang Gu, Radu Timofte, et al. Ntire 2019 challenge on real image super­resolution: Methods and results[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019, 0-0.
4. Zhihao Wang, Jian Chen, Steven C.H.Hoi. Deep learning for image super-resolution: A survey[J].IEEE transactions on pattern analysis and machine intelligence, 2020, 43(10): 3365-3387.
5. Christian Ledig, Lucas Theis, Ferenc Huszar, et al. Photo-realistic single image super­resolution using a generative adversarial network[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, 4681-4690.
6. Rujia Shen, Jian Xu, Qing Bao, et al. Fine-Grained Urban Flow Prediction via a Spatio­Temporal Super-Resolution Scheme[C]. Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data. Springer, Cham, 2020, 360-375.
7. Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, et al. Neural ordinary differential equations[J]. Advances in neural information processing systems, 2018, 31.
8. Kamyar Nazeri, Harrish Thasarathan, Mehran Ebrahimi. Edge-informed single image super-resolution[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019, 0-0.
9. Wenzhe Shi, Jose Caballero, Ferenc Huszar, et al. Real-time single image and video super­resolution using an efficient sub-pixel convolutional neural network[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. 1874-1883.
10. M. W. Thornton, P. M. Atkinson, D. A. Holland. Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel- swapping[JJ. International Journal of Remote Sensing, 2006, 27(3): 473-491.
11. Robert G. Keys. Cubic convolution interpolation for digital image processing[J]. IEEE transactions on acoustics, speech, and signal processing, 1981, 29(6): 1153-1160.
12. Claude E. Duchon. Lanczos filtering in one and two dimensions [J]. Journal of Applied Meteorology and Climatology, 1979, 18(8): 1016-1022.
13. Shengyang Dai, Mei Han, Wei Xu, et al. Softcuts: a soft edge smoothness prior for color image super-resolution[J]. IEEE Transactions on Image Processing, 2009. 18(5): 969-981.
14. Jian Sun. Zongben Xu, Heung-Yeung Shum. Image super-resolution using gradient profile prior[C]. 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008,
15. Qing Yan, Yi Xu, Xiaokang Yang, et al. Single image superresolution based on gradient profile sharpness [J]. IEEE Transactions on Image Processing, 2015, 24(10): 3187・3202.
16. Hong Chang, Dit-Yan Yeung, Yimin Xiong. Super-resolution through neighbor embedding[C]. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. IEEE, 2004, 1:1-I.
17. Kaibing Zhang, Dacheng Tao, Xinbo Gao, et al. Coarse-to-fine learning for single-image super-resolution[J]. IEEE transactions on neural networks and learning systems, 2016, 28(5): 1109-1122.
18. Cheng Deng, Jie Xu, Kaibing Zhang, et at Similarity constraints-based structured output regression machine: An approach to image super-resolution[J]. IEEE transactions on neural networks and learning systems, 2015,27(12): 2472-2485.
19. Wenming Yang, Yapeng Tian, Fei Zhou, et al. Consistent coding scheme for single-image super-resolution via independent dictionaries [J]. IEEE Transactions on Multimedia, 2016, 18(3): 313-325.
20. Chao Dong, Chen Change Loy, Kaiming He, et al. Image super-resolution using deep convolutional networks [J]. IEEE transactions on pattern analysis and machine intelligence, 2015,38(2):295-307.
21. Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv: 1409.1556, 2014.
22. Jiwon Kim, Jung Kwon Lee, Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, 1646-1654.
23. Carl Doersch, Abhinav Gupta, Alexei A. Efros. Unsupervised visual representation learning by context prediction[C]. Proceedings of the IEEE international conference on computer vision. 2015, 1422-1430.
24. Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, et al. Context encoders: Feature learning by inpainting[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, 2536-2544.
25. Richard Zhang, Junyan Zhu, Phillip Isola, et al. Real-time user-guided image colorization with learned deep priors [J]. arXiv preprint arXiv: 1705.02999, 2017.
26. Spyros Gidaris, Praveer Singh, Nikos Komodakis. Unsupervised representation learning by predicting image rotations[J]. arXiv preprint arXiv: 1803.0772& 2018.
27. Tomas Mikolov, Kai Chen, Greg Corrado, et al. Efficient estimation of word representations in vector space[J].arXiv preprint arXiv:1301.3781, 2013.
28. Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv: 1810.04805, 2018.
29. Pierre Sermanet, Corey Lynch, Yevgen Chebotar, et al. Time-contrastive networks: Self­supervised learning from video[C]. 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, 1134-1141
30. Pierre Sermanet, Corey Lynch, Jasmine Hsu, et al. Time-contrastive networks: Self­supervised learning from multi-view observation[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2017, 486-487.
31. Xiaolong Wang, Abhinav Gupta. Unsupervised learning of visual representations using videos[C]. Proceedings of the IEEE international conference on computer vision. 2015, 2794-2802.
32. R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, et al. Learning deep representations by mutual information estimation and maximization[J]. arXiv preprint arXiv:180&06670,2018.
33. Aaron van den Oord, Yazhe Li, Oriol Vinyals. Representation learning with contrastive predictive coding[J], arXiv e-prints, 201& arXiv: 1807.03748.
34. Ting Chen, Simon Komblith, Mohammad Norouzi, et al. A simple framework for contrastive learning of visual representations[C]. International conference on machine learning. PMLR, 2020, 1597-1607.
35. Mu-Chen Chen, Yu Wei. Exploring time variants for short-term passenger flow[J]. Journal of Transport Geography, 2011, 19(4): 488-498.
36. Yu Wei, Mu-Chen Chen. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks[J]. Transportation Research Part C: Emerging Technologies, 2012,21(1): 148-162.
37. Sergey Ioffe, Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International conference on machine learning. PMLR, 2015,448-456.
38. Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, et al. End-to-end learning of visual representations from uncurated instructional videos[C]. Proceedings of the IEEE/CVF Conference on Computer Msion and Pattern Recognition・ 2020, 9879-9889.
39. Yongshun Gong, Zhibin Li, Jian Zhang, 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・
40. Florian Schroff Dmitry Kalenicheiiko, James Philbin. Facenet: A unified embedding for face recognition and clustering[C]. Proceedings of the IEEE conferenee on computer vision and pattern recognition. 2015, 815-823.
41. Thomas Vandal, Evan Kodra, Sangram Ganguly, et al. Deepsd: Generating high resolution climate change projections through single image super-resolution[C]. Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017, 1663-1672.
42. Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, et al. Deep laplacian pyramid networks for fest and accurate super-resolution[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, 624-632.
43. Zheng Hui, Xinbo Gao, Yunchu Yang, et al. Lightweight image super-resolution with information multi-distillation network[C]. Proceedings of the 27th acm international conference on multimedia. 2019, 2024-2032.
44. Yuchen Fan, Jiahui Yii, Ding Liu, et al. Scale-wise convolution for image restoration[C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2020,34(07): 10770-10777.
45. Diederik R Kingma, Jimmy Lei Ba. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
46. Omer Sumer, Tobias Dencker, Bjorn Ommer. Self-supervised learning of pose embeddings from spatiotemporal relations in videos[C]. Proceedings of the IEEE International Conference on Computer X^sion・ 2017, 4298-4307.
47. Jiayu Yang, Jose M. Alvarez, Miaomiao Liu. Self-supervised Learning of Depth Inference for Multi-view Stereo[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, 7526-7534.