

基于状态空间模型的高效语义感知轨迹表示学习方法

刘奕辰, 林彦, 周泽宇, 郭晟楠, 林友芳, 万怀宇

引用本文

刘奕辰, 林彦, 周泽宇, 郭晟楠, 林友芳, 万怀宇. 基于状态空间模型的高效语义感知轨迹表示学习方法[J]. 计算机科学, 2026, 53(4): 134-142.

LIU Yichen, LIN Yan, ZHOU Zeyu, GUO Shengnan, LIN Youfang, WAN Huaiyu. [Efficient Semantic-aware Trajectory Representation Learning Method via State Space Model](#) [J]. Computer Science, 2026, 53(4): 134-142.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于双重语义对比学习的无监督红外图像生成方法](#)

Unsupervised Infrared Image Generation Method Based on Dual Semantic Contrastive Learning
计算机科学, 2026, 53(4): 260-268. <https://doi.org/10.11896/jsjcx.250700172>

[基于对比学习的双通道源代码漏洞检测模型](#)

Dual-channel Source Code Vulnerability Detection Model Based on Contrastive Learning
计算机科学, 2026, 53(3): 424-432. <https://doi.org/10.11896/jsjcx.250200124>

[基于大型语言模型增广的少样本持续毒性检测](#)

Few-shot Continuous Toxicity Detection Based on Large Language Model Augmentation
计算机科学, 2026, 53(3): 321-330. <https://doi.org/10.11896/jsjcx.250600010>

[融合多视角习题表征与遗忘机制的深度知识追踪](#)

Multi-view Exercise Representation and Forgetting Mechanism for Deep Knowledge Tracing
计算机科学, 2026, 53(3): 107-114. <https://doi.org/10.11896/jsjcx.250700092>

[基于用户静动态兴趣和去噪隐式负反馈的新闻推荐算法](#)

News Recommendation Algorithm Based on User Static and Dynamic Interests and Denoised Implicit Negative Feedback
计算机科学, 2026, 53(2): 152-160. <https://doi.org/10.11896/jsjcx.241200177>

基于状态空间模型的高效语义感知轨迹表示学习方法

刘奕辰¹ 林彦² 周泽宇¹ 郭晟楠¹ 林友芳¹ 万怀宇¹

1 北京交通大学计算机科学与技术学院 北京 100044

2 奥尔堡大学计算机科学系 奥尔堡 9220

(liuyichen@bjtu.edu.cn)

摘要 车辆轨迹为各类交通服务应用提供了关键的运动信息。为了更好地利用车辆轨迹,有必要开发轨迹表示学习方法来准确且高效地提取包括运动行为和出行目的在内的出行语义,以支持精确的下游应用。然而,这一任务面临两大挑战:1)运动行为本质上是时空连续的,难以从离散轨迹点中有效提取;2)出行目的与车辆经过的区域和路段的功能相关,但这些功能无法从原始时空轨迹特征中直接获得,也难以从相关的复杂文本特征中提取。为了解决这些挑战,提出了一种高效语义感知轨迹表示学习方法 ESTRL。首先,引入了基于 Mamba 的轨迹编码器,使用高阶移动特征参数化轨迹状态空间模型,有效且高效地建模车辆的连续运动行为。其次,提出了出行目的感知预训练机制,通过对比学习将出行目的融入学习到的轨迹嵌入中,从而无需在嵌入计算过程中引入额外开销。在真实数据集上的大量实验表明,所提方法在效率和准确性方面均优于当前先进的基线模型。

关键词: 车辆轨迹;轨迹表示学习;状态空间模型;对比学习;自监督预训练

中图分类号 TP311

Efficient Semantic-aware Trajectory Representation Learning Method via State Space Model

LIU Yichen¹, LIN Yan², ZHOU Zeyu¹, GUO Shengnan¹, LIN Youfang¹ and WAN Huaiyu¹

1 School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China

2 Department of Computer Science, Aalborg University, Aalborg 9220, Denmark

Abstract Vehicle trajectories provide crucial movement information for various traffic service applications. To better utilize vehicle trajectories, it is essential to develop trajectory representation learning methods that can effectively and efficiently extract travel semantics, including movement behaviors and travel purposes, to support accurate downstream applications. However, this task presents two major challenges: 1) movement behaviors are inherently spatio-temporally continuous, making them difficult to extract effectively from discrete trajectory points; 2) travel purposes are related to the functionalities of areas and road segments traversed by vehicles, but these functionalities cannot be directly obtained from the raw spatio-temporal trajectory features, nor can they be extracted from the relevant complex textual features. To address these challenges, this paper proposes an efficient semantic-aware trajectory representation learning method called ESTRL. Firstly, a Mamba-based trajectory encoder is introduced. It uses high-order movement features to parameterize the trajectory state space model (Traj-SSM), which effectively and efficiently models continuous movement behaviors of vehicles. Secondly, a travel purpose-aware pre-training procedure is proposed. It integrates travel purposes into the learned trajectory embeddings through contrastive learning without introducing extra overhead to embedding calculation process. Extensive experiments on real-world datasets demonstrate that the proposed method outperforms state-of-the-art baseline models in both efficiency and accuracy.

Keywords Vehicle trajectories, Trajectory representation learning, State space model, Contrastive learning, Self-supervised pre-training

1 引言

车辆轨迹由〈地点,时间〉对序列组成,记录了车辆在行驶过程中的运动情况。随着智能手机和车载导航系统等位置记录设备的广泛使用,这些轨迹变得越来越容易获得。同时,智能交通系统的最新进展强调了车辆轨迹在揭示包括运动行为

和出行目的在内的出行语义方面的价值。这种出行语义支持一系列轨迹挖掘任务,包括轨迹预测^[1-2]、行程时间估计^[3-4]、轨迹异常检测^[5-6]、轨迹相似性度量^[7-8]和轨迹聚类^[9]。车辆轨迹日益增长的可用性和利用率推动了轨迹学习技术的发展,促进了现实世界交通应用的优化。

为了提高这些任务与应用的性能,需要开发能够有效地

从车辆轨迹中提取丰富出行语义的轨迹学习模型。此外,考虑到轨迹的诸多实际应用场景都需要精准且实时的决策支持,高效地提取出行语义对于减少计算负担和缩短响应时间也是必要的。然而,实现这些目标面临诸多阻碍。

首先,从离散的轨迹点中提取连续的运动行为是具有挑战性的。车辆的运动行为具有时空连续性,但车辆轨迹是由位置记录设备以一定的采样率记录的离散的轨迹点构成的,各轨迹点反映了车辆在特定时刻的位置信息。离散的轨迹点之间的相关性实际上并不能准确地表示连续的运动行为。常见的序列模型,包括循环神经网络(Recurrent Neural Networks, RNNs)^[10-11]和 Transformers^[12],只考虑了离散点之间的相关性,不能有效地从车辆轨迹中建模连续运动行为。现有的工作^[13-15]已经利用神经常微分方程(Neural Ordinary Differential Equations, Neural ODEs)^[16-17]或专门设计的 Transformers 来显式地从离散轨迹点中提取连续运动行为。尽管一定程度提升了性能,但这些解决方案带来了巨大的计算成本,损害了模型效率。具体而言,前者涉及求解积分方程,即使在现代硬件上速度也很慢,而后者涉及二次计算复杂度。因此,如何从离散轨迹点中高效建模连续的运动模式仍然是一个尚未解决的问题。

其次,从轨迹的原始时空特征或相关的文本特征中提取出行目的并不容易。例如,一条车辆轨迹从居民区出发,经过高速路和主干道,最后到达公园,体现了与娱乐或休闲活动相关的出行目的。也就是说,轨迹所经过的兴趣点(Points of Interest, POIs)和道路的功能承载了丰富的信息,能够揭示车辆的出行目的。这些功能往往存储于与 POI 和道路相关的文本特征中,而现有的轨迹学习方法大多仅使用车辆轨迹的时空特征编码轨迹嵌入,无法直接从中获得反映车辆出行目的的信息。近年来,语言模型(Language Models, LMs)^[18-20]的发展使得从文本描述中提取 POI 和道路的复杂功能成为可能。这一能力已被一些近期的轨迹学习研究(如文献[21])所探索。但由于 LMs 通常具有大量的模型参数,将其集成到标准的轨迹学习模型中会带来巨大的计算负担。因此,如何高效地结合 POI 和道路功能来提取出行目的仍然是一个有待探索的问题。

针对上述挑战,本文设计了一种高效语义感知轨迹表示学习(Efficient Semantic-aware Trajectory Representation Learning, ESTRL)方法。本文的主要贡献包括 3 个方面:

1)提出了一个基于 Mamba 的轨迹编码器,使用高阶移动特征参数化轨迹状态空间模型(Trajectory State Space Model, Traj-SSM),以线性时间复杂度将车辆轨迹映射为嵌入向量,有效且高效地提取轨迹的运动行为。

2)提出了出行目的感知预训练机制,通过对比学习将学习到的轨迹嵌入与用车辆轨迹的道路视图和 POI 视图表示的出行目的对齐,使经过预训练后的轨迹编码器可在嵌入过程中识别出行目的而无需额外计算开销,实现出行目的的高效提取。

3)在两个真实的车辆轨迹数据集上进行了广泛的实验,并比较了各种先进的车辆轨迹学习方法,验证了所提出方法的优越性。

2 相关工作

车辆轨迹学习方法从车辆轨迹中提取有价值的信息来执行各种任务。这些方法可以大致分为端到端的轨迹学习方法和预训练的轨迹嵌入方法。

2.1 端到端的轨迹学习方法

端到端的轨迹学习方法是为特定的任务量身定做的,通常使用任务特定的标签进行训练。根据任务类型的不同,端到端的方法大致可以分为轨迹预测方法^[2,22-24]、轨迹分类方法^[13,25-26]和轨迹相似性度量方法^[27-28]。在轨迹预测方面,文献[22-24]利用 RNNs 捕捉轨迹中的序列相关性,文献[2]则使用 Transformers 处理车辆轨迹。轨迹分类方法大多通过识别用户或驾驶员 ID 来将轨迹分类到各自的类标签中。轨迹相似性度量方面,文献[27-28]利用可学习的模型来高效地计算轨迹之间的相似性。其中,文献[28]使用了图神经网络(Graph Neural Network, GNN)编码轨迹中的各 POI 嵌入及其在空间网络上的邻域信息,以克服数据稀疏性问题。

虽然端到端的方法能够直接实现并具有一定的优势,但它们不能轻易地被重用到其他任务中。此时就需要为每个任务设计、训练和存储单独的模型,这在一定程度上会影响计算资源和存储效率。此外,端到端方法的有效性依赖于任务特定标签的丰富程度,这并不总是能得到保证的。

2.2 预训练的轨迹嵌入方法

为了解决端到端方法的局限性,可以在各种任务中使用预训练的轨迹嵌入方法,因此其最近得到了越来越多的关注。该方法涉及训练一个轨迹编码器,将车辆轨迹映射为嵌入向量,然后送入一个预测模块用于下游任务。文献[9]使用自编码框架^[29]将每个序列压缩成一个嵌入向量。文献[30]使用去噪自编码框架来增强其对轨迹噪声的抵抗能力。文献[31]利用自编码技术有效地提取编码在轨迹中的路网和时间信息。文献[32]结合对比学习^[33]来学习轨迹的嵌入向量。文献[34]引入了一种综合的轨迹嵌入学习方法,通过结合掩码语言模型(Masked Language Model, MLM)^[19]任务与 SimCLR^[35]来增强其学习能力。此外,该方法在空间建模上还引入了图编码器,通过路网结构进一步限制轨迹表示空间,同时将路网上的转移概率作为先验知识整合到轨迹编码。文献[15]利用最大熵编码^[36]来学习在不同任务中表现一致的轨迹嵌入。

3 预备知识

3.1 定义

定义 1(车辆轨迹) 车辆轨迹 \mathcal{T} 是一条带时间戳的位置序列,表示为 $\mathcal{T} = \langle p_1, p_2, \dots, p_n \rangle$,其中 n 为点的个数。每个点 $p_i = (lng_i, lat_i, t_i)$ 由经度 lng_i 、纬度 lat_i 和时间戳 t_i 组成,代表车辆在特定时刻的位置。

定义 2(路网) 将路网建模为一个有向图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 。其中 \mathcal{V} 是节点的集合,每个节点 $v_i \in \mathcal{V}$ 代表路段之间的交点或者路段的终点; \mathcal{E} 是边的集合,每条边 $e_i \in \mathcal{E}$ 表示连接两个节点的路段。一条边是由它的起点、终点以及包含道路名称和类型的文本描述定义的,记为 $e_i = (v_j, v_k, desc_i^{\text{Road}})$ 。

定义 3(兴趣点) 兴趣点(Point of Interest, POI)是具有特定文化、环境或经济重要性的位置。将一个 POI 表示为 $l_i = (lng_i, lat_i, desc_i^{POI})$, 其中 lng_i 和 lat_i 是 POI 的坐标, $desc_i^{POI}$ 是包含 POI 的名称、类型和地址的文本描述。

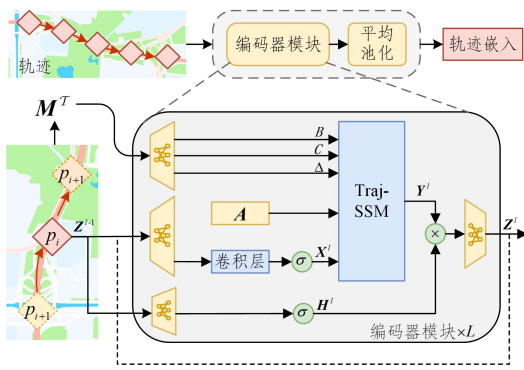
3.2 车辆轨迹表示学习

车辆轨迹学习旨在构建一个学习模型 f_θ , 其中 θ 是可学习参数的集合。给定车辆轨迹 \mathcal{T} , 模型计算其嵌入向量为 $e_{\mathcal{T}} = f_\theta(\mathcal{T})$ 。该嵌入向量 $e_{\mathcal{T}}$ 捕获了 \mathcal{T} 的出行语义, 可以通过添加预测模块用于后续应用。

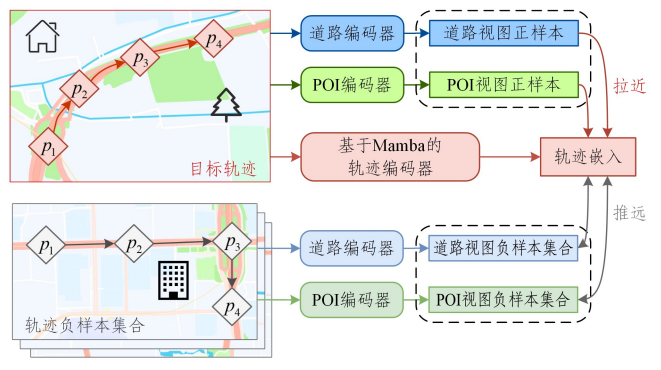
4 高效语义感知轨迹表示学习方法

4.1 总体框架

本文提出了一种新的高效语义感知轨迹表示学习方法 ESTRL, 模型框架如图 1 所示。该方法由两个主要部分组



(a) 基于 Mamba 的轨迹编码器结构



(b) 出行目的感知预训练机制

图 1 模型框架

Fig. 1 Framework of the proposed model

4.2 基于 Mamba 的轨迹编码器

为了有效提取运动行为, 本文提出了基于 Mamba 的轨迹编码器, 它主要由堆叠的编码器模块组成。每个模块首先利用轨迹的高阶移动特征对 Traj-SSM 进行参数化, 然后通过 Traj-SSM 捕获轨迹序列中的时空依赖关系, 实现车辆连续运动行为的有效建模。最后一层编码器模块的输出经平均池化得到车辆轨迹的嵌入向量。

4.2.1 运动行为特征提取

对于轨迹 $\mathcal{T} = \langle p_1, p_2, \dots, p_n \rangle$ 中的每个轨迹点 p_i , 使用一个线性层将其空间坐标 (lng_i, lat_i) 映射到嵌入空间 \mathbb{R}^E , 其中 E 表示嵌入维度。同时, 将其时间戳 t_i 转换为一个向量 $\mathbf{t}_i \in \mathbb{R}^4$, 该向量具有 4 个特征, 即一周中的天、一天中的小时、一小时中的分钟以及相对于 t_1 的时间差(分钟)。然后使用可学习的傅里叶编码层^[38]将这 4 个特征编码成 4 个嵌入向量, 接着将它们进行级联, 再通过线性层映射到嵌入空间 \mathbb{R}^E 。最后, 将空间隐向量和时间隐向量相加得到点的隐向量 \mathbf{z}_i , 如式(1)所示:

$$\mathbf{z}_i = \text{Linear}((\text{lng}_i, \text{lat}_i)) + \text{Linear}(\text{Cat}(\text{Fourier}(\mathbf{t}_i))) \quad (1)$$

其中, Cat 表示向量的级联操作, Fourier 表示傅里叶编码层。通过收集 \mathcal{T} 中每个点的隐向量, 得到其隐向量序列, 记为:

$$\mathbf{Z}^{\mathcal{T}} = \langle \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n \rangle \in \mathbb{R}^{n \times E} \quad (2)$$

接着, 根据轨迹点 p_i 与 p_{i-1} 的特征差异, 计算每个点的

成; 用于高效编码轨迹嵌入的基于 Mamba 的轨迹编码器和能够实现语义丰富的轨迹学习的出行目的感知预训练机制。

如图 1(a)所示, 基于 Mamba 的轨迹编码器主要由编码器模块组成, 它从轨迹中提取丰富的出行语义, 并将其融入轨迹的嵌入向量中。编码器模块受 Mamba2 结构^[37]的启发, 使用高阶移动特征参数化的 Traj-SSM 以线性时间复杂度建模轨迹序列中的时空依赖关系, 能有效且高效地从轨迹中提取连续的运动行为。

如图 1(b)所示, 出行目的感知预训练机制增强了基于 Mamba 的轨迹编码器在不增加显著计算开销的情况下提取出行目的的能力。该机制通过对比学习将学习到的轨迹嵌入与道路和 POI 编码器识别的出行目的对齐。基于 Mamba 的轨迹编码器经过该机制预训练后, 即可从轨迹中提取丰富的出行语义, 而无需在其嵌入过程中增加更多的计算或存储资源。

高阶移动特征, 包括速度 v_i 、加速度 acc_i 和运动角度 θ_i 。然后将这 3 个特征进行最大最小归一化, 再串联成一个向量, 记为 $\mathbf{m}_i = (v_i, acc_i, \theta_i)$ 。注意, 第一个点的向量 \mathbf{m}_1 设置为 \mathbf{m}_2 。通过计算 \mathcal{T} 中每个点的高阶移动特征, 得到其高阶移动特征序列, 记为:

$$\mathbf{M}^{\mathcal{T}} = \langle \mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n \rangle \in \mathbb{R}^{n \times 3} \quad (3)$$

4.2.2 编码器模块

本文使用受 Mamba2 结构启发的编码器模块作为基于 Mamba 的轨迹编码器的核心组件, 旨在利用提取到的上述特征, 通过参数化高阶移动特征来建模连续运动行为。

编码器模块采用多层结构, 第 l 层以隐向量序列 $\mathbf{Z}^{l-1} \in \mathbb{R}^{n \times E}$ 作为输入。该输入首先经过线性投影层处理, 然后经过卷积层和 SiLU 激活函数, 如式(4)所示:

$$\mathbf{X}^l = \sigma(\text{Conv}(\text{Linear}(\mathbf{Z}^{l-1}))) \quad (4)$$

其中, $\mathbf{X}^l \in \mathbb{R}^{n \times D}$, D 为模型维度; Conv 表示一维因果卷积; σ 表示 SiLU 激活函数。特别地, 第一层编码器模块的输入 \mathbf{Z}^0 是由式(2)计算得到的序列 $\mathbf{Z}^{\mathcal{T}}$ 。

另一方面, 高阶移动特征参数化的实现涉及到利用式(3)计算出高阶移动特征序列 $\mathbf{M}^{\mathcal{T}}$ 来构造参数矩阵, 然后用于每个编码器模块内的 Traj-SSM 中。形式上, 通过以下方式计算 3 个参数矩阵 $\mathbf{B}, \mathbf{C}, \mathbf{A}$:

$$\begin{aligned} \mathbf{B}, \mathbf{C}, \mathbf{A} &= \text{Linear}(\mathbf{M}^{\mathcal{T}}) \\ \mathbf{A} &= \tau(\mathbf{A} + \mathbf{b}_2) \end{aligned} \quad (5)$$

其中, τ 表示 Softplus 激活函数; \mathbf{b}_Δ 是 \mathbf{A} 的偏置参数; 矩阵 $\mathbf{B} \in \mathbb{R}^{n \times N}$ 和 $\mathbf{C} \in \mathbb{R}^{n \times N}$ 分别为 Traj-SSM 的输入和输出映射矩阵, N 为状态维数。通过使用高阶移动特征对它们进行参数化, 该模型能够精确控制运动行为的变化对输入特征处理及输出嵌入生成的影响。此外, $\mathbf{A} \in \mathbb{R}^{n \times H}$ 是时间尺度矩阵, 控制着 Traj-SSM 在轨迹序列上的演化速率, 其中 H 为 Traj-SSM 的头数。因此, \mathbf{A} 的参数化对于从离散的轨迹点中捕捉连续的运动行为非常重要。另一个参数 $\mathbf{A} \in \mathbb{R}^H$ 则作为隐状态映射矩阵被随机初始化, 它是编码器模块的一个可学习参数。

接着, 遵循 Mamba^[39] 中选择性状态空间模型 (Selective State Space Models) 的实现, 使用时间尺度矩阵 \mathbf{A} 将 \mathbf{A} 和 \mathbf{B} 离散化为 $\bar{\mathbf{A}}$ 和 $\bar{\mathbf{B}}$, 公式如下:

$$\begin{aligned} \bar{\mathbf{A}} &= \langle \exp(\mathbf{A}_1^\top \odot \mathbf{A}), \exp(\mathbf{A}_2^\top \odot \mathbf{A}), \dots, \exp(\mathbf{A}_n^\top \odot \mathbf{A}) \rangle \\ \bar{\mathbf{B}} &= \langle \mathbf{A}_1^\top \mathbf{B}_1, \mathbf{A}_2^\top \mathbf{B}_2, \dots, \mathbf{A}_n^\top \mathbf{B}_n \rangle \end{aligned} \quad (6)$$

其中, $\bar{\mathbf{A}} \in \mathbb{R}^{n \times H}$ 和 $\bar{\mathbf{B}} \in \mathbb{R}^{n \times H \times N}$ 分别采用零阶保持 (Zero-Order Hold, ZOH) 方法和欧拉方法进行离散化; \odot 表示 Hadamard 积; \mathbf{A}_i 和 \mathbf{B}_i 分别为 \mathbf{A} 和 \mathbf{B} 的第 i 行。该离散化过程使模型能够更准确地表示连续运动行为。随后, 即可利用离散化后的上述矩阵与由式(4)计算出的序列 \mathbf{X}^l 实现 Traj-SSM, 记为:

$$\mathbf{Y}^l = \text{Traj-SSM}(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C})(\mathbf{X}^l) \quad (7)$$

具体来说, Traj-SSM 沿用了 Mamba2^[37] 中多输入 SSM 的实现流程。输入 $\mathbf{X}^l \in \mathbb{R}^{n \times D}$ 首先变形为 $\mathbf{X}^l \in \mathbb{R}^{n \times H \times (D/H)}$ 来实现 H 个头的创建。接着, 类似于 RNNs, Traj-SSM 能以递归的形式进行计算。对于第 i 个时间步的第 j 个头, 当 $D/H=1$ 时, Traj-SSM 的递推公式如下:

$$\begin{aligned} \mathbf{h}_{ij} &= \bar{\mathbf{A}}_{ij} \mathbf{I} \mathbf{h}_{i-1,j} + \bar{\mathbf{B}}_{ij} \mathbf{x}_{ij} \\ \mathbf{y}_{ij} &= \mathbf{C}_{ij} \mathbf{h}_{ij} \end{aligned} \quad (8)$$

其中, $\mathbf{h}_{ij} \in \mathbb{R}^{N \times 1}$ 为隐状态; $\mathbf{I} \in \mathbb{R}^{N \times N}$ 为单位矩阵; $\mathbf{x}_{ij} \in \mathbb{R}$, $\mathbf{y}_{ij} \in \mathbb{R}$, $\bar{\mathbf{A}}_{ij} \in \mathbb{R}$, $\bar{\mathbf{B}}_{ij} \in \mathbb{R}^{N \times 1}$ 和 $\mathbf{C}_{ij} \in \mathbb{R}^{1 \times N}$ 分别代表 $\mathbf{X}^l[i, j, :]$, $\mathbf{Y}^l[i, j, :]$, $\bar{\mathbf{A}}[i, j, :]$, $\bar{\mathbf{B}}[i, j, :]$ 和 $\mathbf{C}[i, :]$ 。当 $D/H > 1$ 时, 可将 Traj-SSM 的第 j 个头的输入 $\mathbf{X}^l[:, j, :]$ 视为 D/H 个独立序列, 再对每个序列应用上述方程式进行计算。最后, 输出 $\mathbf{Y}^l \in \mathbb{R}^{n \times H \times (D/H)}$ 通过聚合各个头来重塑为与原始输入形状相同的 $\mathbf{Y}^l \in \mathbb{R}^{n \times D}$ 。为保证相对于轨迹长度 n 的线性计算复杂度, 模型在具体实现中使用了 Mamba2 提供的硬件高效算法。

除了 Traj-SSM 所在分支, 编码器模块的另一个分支由一个线性层和一个 SiLU 激活函数组成, 如式(9)所示:

$$\mathbf{H}^l = \sigma(\text{Linear}(\mathbf{Z}^{l-1})) \quad (9)$$

最后, 将两个分支的输出 \mathbf{Y}^l 和 \mathbf{H}^l 合并, 得到第 l 层编码器模块的输出:

$$\mathbf{Z}^l = \text{Linear}(\text{Norm}(\mathbf{Y}^l \odot \mathbf{H}^l)) \quad (10)$$

其中, \mathbf{Z}^l 与 \mathbf{Z}^{l-1} 形状相同, 可视作第 $l+1$ 层编码器模块的输入序列。归一化层 Norm 可以通过 LayerNorm, GroupNorm 或 RMSNorm 实现。

4.2.3 轨迹嵌入

本文通过堆叠 L 层编码器模块实现了基于 Mamba 的轨迹编码器。不同层之间不共享可学习参数。给定来自式(2)的序列 \mathbf{Z}^T 和来自式(3)的序列 \mathbf{M}^T , 得到最后一层编码器模块的输出序列 \mathbf{Z}^L , 然后将其送入平均池化层即可导出轨迹 \mathcal{T} 的嵌入向量 $\mathbf{z}_{\mathcal{T}} \in \mathbb{R}^E$ 。这样, 通过利用高阶移动特征参数化的 Traj-SSM 模块即可在保持线性时间复杂度的同时对连续运动行为进行建模, 实现轨迹嵌入向量的高效计算。

4.3 出行目的感知预训练程序

为了在不给基于 Mamba 的轨迹编码器增加额外计算负载的情况下提取出行目的, 本文提出了出行目的感知预训练机制对其进行预训练。首先, 通过使用道路和 POI 编码器对车辆轨迹遍历的道路和 POI 的文本信息进行建模, 从中提取出行目的。这两个编码器会将信息集成到道路和 POI 视图中。然后, 通过对比学习, 将基于 Mamba 的轨迹编码器的输出嵌入与这些视图进行对齐。

4.3.1 道路和 POI 视图

车辆轨迹的出行目的与其所遍历的道路和 POI 的功能密切相关。因此, 本文提出将轨迹的出行目的信息集成到其道路和 POI 视图中。

给定一条轨迹 $\mathcal{T} = \langle p_1, p_2, \dots, p_n \rangle$, 应用地图匹配算法 LMM (Leuven Map Matching)^[40] 将它的每个点映射到路网 \mathcal{G} 上, 得到由路段组成的地图匹配轨迹 $\mathcal{T}^{\text{mm}} = \langle e_1, e_2, \dots, e_n \rangle$, 其中 e_i 为点 p_i 对应的路段。然后, 计算每个路段 e_i 的嵌入向量 $\mathbf{z}_{e_i} \in \mathbb{R}^E$, 如式(11)所示:

$$\begin{aligned} \mathbf{z}_{e_i} &= \text{Linear}(\text{RoadIndexEmbed}(e_i)) + \\ &\quad \text{Linear}(\text{TextEmbed}(\text{desc}_i^{\text{Road}})) \end{aligned} \quad (11)$$

其中, RoadIndexEmbed 是一个索引获取嵌入层, 它将每个独特的路段映射成一个可学习的嵌入向量; TextEmbed 是一个预训练的文本嵌入模块, 用于将一行文本映射为一个嵌入向量, 具体实现上, 本文使用了 OpenAI¹⁾ 提供的 text-embedding-3-large 模型。最后, 通过对 \mathcal{T}^{mm} 中道路段的嵌入向量序列进行如下处理, 得到 \mathcal{T} 的道路视图 $\mathbf{z}_{\mathcal{T}}^{\text{Road}} \in \mathbb{R}^E$:

$$\mathbf{z}_{\mathcal{T}}^{\text{Road}} = \text{MeanPool}(\text{RoadEnc}(\langle \mathbf{z}_{e_1}, \mathbf{z}_{e_2}, \dots, \mathbf{z}_{e_n} \rangle)) \quad (12)$$

其中, RoadEnc 为道路编码器, 使用 2 层 Transformer 编码器来实现。

为计算 POI 视图, 首先根据每个轨迹点 p_i 与城市中各 POI 之间的地理距离, 找出距离该点最近的 POI l_i 。类似于式(11), 计算每个 POI l_i 的嵌入向量 $\mathbf{z}_{l_i} \in \mathbb{R}^E$, 如式(13)所示:

$$\begin{aligned} \mathbf{z}_{l_i} &= \text{Linear}(\text{POIIndexEmbed}(l_i)) + \\ &\quad \text{Linear}(\text{TextEmbed}(\text{desc}_i^{\text{POI}})) \end{aligned} \quad (13)$$

其中, POIIndexEmbed 是另一个索引获取嵌入层, 为每个唯一的 POI 分配一个可学习的嵌入向量; TextEmbed 与式(11)中使用的文本嵌入模块相同。最后, 通过对与 \mathcal{T} 相关的 POI 的嵌入向量序列进行如下处理, 计算出 \mathcal{T} 的 POI 视图 $\mathbf{z}_{\mathcal{T}}^{\text{POI}} \in \mathbb{R}^E$:

$$\mathbf{z}_{\mathcal{T}}^{\text{POI}} = \text{MeanPool}(\text{POIEnc}(\langle \mathbf{z}_{l_1}, \mathbf{z}_{l_2}, \dots, \mathbf{z}_{l_n} \rangle)) \quad (14)$$

¹⁾ <https://platform.openai.com/docs/guides/embeddings>

其中, POI_{enc} 是使用另一个 2 层 Transformer 编码器实现的 POI 编码器。

4.3.2 对比学习

在将轨迹的出行目的信息表示为道路和 POI 视图后, 使用对比学习将基于 Mamba 的轨迹编码器的输出嵌入向量与这两个视图进行对齐。

给定一组批量大小为 B 的轨迹集合 $\mathbb{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_B\}$, 通过基于 Mamba 的轨迹编码器得到它们的嵌入向量为 $\{\mathbf{z}_{\mathcal{T}_1}, \mathbf{z}_{\mathcal{T}_2}, \dots, \mathbf{z}_{\mathcal{T}_B}\}$ 。根据式 (12), 它们的道路视图可记为 $\{\mathbf{z}_{\mathcal{T}_1}^{Road}, \mathbf{z}_{\mathcal{T}_2}^{Road}, \dots, \mathbf{z}_{\mathcal{T}_B}^{Road}\}$; 根据式 (14), 它们的 POI 视图可记为 $\{\mathbf{z}_{\mathcal{T}_1}^{POI}, \mathbf{z}_{\mathcal{T}_2}^{POI}, \dots, \mathbf{z}_{\mathcal{T}_B}^{POI}\}$ 。然后使用点积计算 \mathcal{T}_i 与 \mathcal{T}_j 的道路和 POI 视图之间的相似度, 如式 (15) 所示:

$$s_{ij}^{Road} = \mathbf{z}_{\mathcal{T}_i} \cdot \mathbf{z}_{\mathcal{T}_j}^{Road}, s_{ij}^{POI} = \mathbf{z}_{\mathcal{T}_i} \cdot \mathbf{z}_{\mathcal{T}_j}^{POI} \quad (15)$$

接着, 对它们应用 InfoNCE 损失^[33], 如式 (16) 所示:

$$\mathcal{L}_{\mathbb{T}}^{Road} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(s_i^{Road}/\tau)}{\sum_{j=1}^B \exp(s_{ij}^{Road}/\tau)} \quad (16)$$

$$\mathcal{L}_{\mathbb{T}}^{POI} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(s_i^{POI}/\tau)}{\sum_{j=1}^B \exp(s_{ij}^{POI}/\tau)}$$

其中, τ 为温度参数, 作为一个对数参数化的乘法标量在训练过程中直接优化^[41]; $\mathcal{L}_{\mathbb{T}}^{Road}$ 和 $\mathcal{L}_{\mathbb{T}}^{POI}$ 可以看作最大化 \mathbb{T} 中每一条轨迹的轨迹嵌入与自身的道路和 POI 视图之间的相似度, 同时最小化该轨迹嵌入与其他轨迹的道路和 POI 视图之间的相似度。最后, 对比学习损失是上述两种损失的组合, 公式如下:

$$\mathcal{L}_{\mathbb{T}} = \frac{1}{2} (\mathcal{L}_{\mathbb{T}}^{Road} + \mathcal{L}_{\mathbb{T}}^{POI}) \quad (17)$$

经过预训练后, 基于 Mamba 的轨迹编码器的嵌入向量与道路和 POI 视图表示的出行目的进行对齐。此外, 该预训练机制在嵌入过程中不会给基于 Mamba 的轨迹编码器增加额外的计算需求, 从而保持其高效性。

5 实验及其结果分析

本文使用两个真实的车辆轨迹数据集来评估提出方法的有效性, 并将其性能与几种先进的基线方法进行比较。

5.1 数据集

本文使用的两个车辆轨迹数据集分别简称为成都和西安。它们由 Didi¹⁾ 发布, 包含中国成都和西安的出租车 GPS 轨迹。由于原始轨迹具有非常密集的采样间隔, 本文通过三跳重采样过程保留一部分轨迹点, 使得大多数轨迹的采样间隔不低于 6 秒。经过重采样后, 轨迹点少于 5 个或超过 120 个的轨迹被认为是异常并被排除。此外, 本文通过 AMap API²⁾ 检索两个数据集兴趣区域中的 POI 信息, 并从 OpenStreetMap³⁾ 中获取覆盖这两个数据集的路网拓扑结构和信息。利用现有的地图匹配算法 LMM^[40], 本文将所有的车辆轨迹映射到路网, 以生成地图匹配轨迹。这两个数据集经过上述预处理后的统计结果如表 1 所列。

表 1 数据集统计信息

Table 1 Dataset statistics

数据集	成都	西安
时间跨度	2018-09-30—2018-10-10	2018-09-29—2018-10-15
轨迹条数	140000	210000
轨迹点数	18832411	18267440
路段数	4315	3392
POI 数	12439	3900

5.2 基线方法

本文使用了几种先进的车辆轨迹学习方法作为基线模型进行比较。

1) t2vec^[30]: 通过使用去噪自编码器从低采样率轨迹中重建原始轨迹来预训练模型。

2) Trembr^[31]: 构建基于 RNN 的 seq2seq 模型, 以恢复输入轨迹的路段和时间。

3) CTLE^[42]: 预训练一个双向 Transformer, 具有两个 MLM 任务, 用于预测位置和小时。通过对轨迹的点嵌入应用平均池化来获得轨迹表示。

4) Toast^[43]: 使用上下文感知的 node2vec 模型来生成路段表示, 并使用基于 MLM 的任务和序列判别任务来训练模型。

5) TrajCL^[44]: 引入了一种基于双特征自注意力的编码器, 并使用 InfoNCE 损失以对比的方式训练模型。

6) LightPath^[45]: 构建稀疏路径编码器, 并用路径重建任务和跨视角与跨网络对比任务进行训练。

7) START^[34]: 包括一个时间感知的轨迹编码器和一个考虑路段之间转移关系的 GAT。该模型同时使用 MLM 任务和基于 SimCLR 损失的对比任务进行训练。

5.3 下游任务

为了评估 ESTRL 方法以及基线模型学习的轨迹嵌入的有效性, 本文将这些嵌入应用到 3 个具有代表性的下游任务中。

1) 目的地预测。该任务涉及预测轨迹的目的地。在计算一条轨迹 \mathcal{T} 的嵌入 $\mathbf{z}_{\mathcal{T}}$ 时, 省略了 \mathcal{T} 的最后 5 个点。然后, 一个全连接网络使用此嵌入来预测轨迹目的地的 GPS 坐标。使用均方误差 (Mean Squared Error, MSE), 通过比较预测坐标和真实坐标来监督预测, 地球表面最短距离的平均绝对误差 (Mean Absolute Error, MAE) 和均方根误差 (Root Mean Squared Error, RMSE) 作为评价指标。

2) 到达时间估计。该任务旨在预测一条轨迹到达目的地的时间。与目的地预测任务类似, 通过省略轨迹 \mathcal{T} 的最后 5 个点来计算其嵌入向量, 并使用全连接网络来预测行程时间。使用 MSE 对预测进行监督, MAE、RMSE 和平均绝对百分比误差 (Mean Absolute Percentage Error, MAPE) 作为评价指标。

3) 相似轨迹搜索。该任务旨在从一批候选轨迹中识别出与查询轨迹最相似的轨迹。轨迹之间的相似性通过它们嵌入之间的余弦相似度来计算。采用 Accuracy@N (Acc@N),

¹⁾ <https://gaia.didichuxing.com/>

²⁾ <https://lbs.amap.com/api/javascript-api-v2>

³⁾ <https://www.openstreetmap.org/>

Acc@5)和平均排名(Mean Rank)作为评价指标。由于大多数数据集没有用于该任务的标签数据,因此本文按照如下方式创建标签。从测试集中随机选取1000条轨迹,对每条轨迹 \mathcal{T} 收集它的奇数点组成查询 \mathcal{T}^q ,偶数点组成目标 \mathcal{T}^t 。对于每个查询,丢弃最接近查询的前10条轨迹,然后从测试集中的剩余轨迹中随机选择5000条轨迹作为数据库。为确定查询轨迹和其他轨迹之间的距离,遵循文献[46],将它们下采样到统一的长度并计算MSE。

对于相似轨迹搜索任务,轨迹学习方法的参数在预训练后是固定的。对于另外两个任务,既可以使用任务监督微调它们的参数,也可以固定它们的参数只更新预测器的参数。在实验中,本文将后一种设置记为无微调(Without Fine-tune, w/o ft)。

5.4 实验设置

本文将两个数据集均以8:1:1的比例划分为训练集、验证集和测试集,轨迹的出发时间按时间顺序排列。ESTRL方法在训练集上预训练30轮,在下游任务中基于验证集上的效果使用早停机制。最终的度量指标使用测试集计算。

表3 各方法在目的地预测与到达时间估计上的整体性能

Table 3 Overall performance of methods on destination prediction and arrival time estimation

数据集	成都					西安				
	目的地预测		到达时间估计			目的地预测		到达时间估计		
	RMSE ↓ /m	MAE ↓ /m	RMSE ↓ /s	MAE ↓ /s	MAPE ↓ /%	RMSE ↓ /m	MAE ↓ /m	RMSE ↓ /s	MAE ↓ /s	MAPE ↓ /%
t2vec	579.30	387.50	127.41	64.67	14.01	482.64	310.08	214.40	108.80	16.96
Trembr	505.62	376.88	124.32	63.42	13.60	473.97	301.45	209.12	107.02	16.40
CTLE	430.19	382.82	135.21	<u>55.41</u>	<u>11.18</u>	477.70	384.08	207.16	107.46	16.25
Toast	480.52	412.58	171.58	91.66	18.84	523.76	443.99	202.99	102.73	15.75
TrajCL	365.50	272.63	132.98	55.78	11.86	383.39	262.20	183.74	73.21	12.55
LightPath	553.27	360.86	123.00	58.04	12.83	598.20	348.61	169.01	74.08	10.50
START	<u>333.10</u>	<u>240.40</u>	<u>121.11</u>	58.97	13.49	<u>319.00</u>	<u>208.35</u>	<u>159.89</u>	<u>72.19</u>	<u>10.26</u>
ESTRL	161.28	118.84	46.37	18.08	4.40	263.16	182.39	86.08	30.80	4.15
t2vec(w/o ft)	2329.63	1868.49	138.30	79.74	18.71	2582.14	2235.27	207.11	117.86	16.01
Trembr(w/o ft)	1787.18	1419.58	159.60	110.36	29.50	2067.80	1749.76	435.04	337.35	47.11
CTLE(w/o ft)	3421.09	3041.49	135.59	63.45	13.99	3548.88	3320.46	272.88	176.16	31.84
Toast(w/o ft)	3434.84	3061.91	149.67	79.69	17.89	3549.65	3325.48	299.94	205.49	32.55
TrajCL(w/o ft)	<u>1059.81</u>	<u>865.48</u>	136.56	79.59	19.85	<u>1268.41</u>	<u>1054.21</u>	194.64	106.66	16.80
LightPath(w/o ft)	2365.87	1948.97	<u>129.48</u>	<u>56.82</u>	<u>12.71</u>	2177.37	1859.35	<u>186.02</u>	<u>77.33</u>	<u>10.41</u>
START(w/o ft)	1347.13	1111.77	144.54	79.78	19.72	1406.06	1173.62	213.22	120.74	20.01
ESTRL(w/o ft)	332.06	260.38	104.61	50.51	11.88	470.54	365.62	155.70	71.28	10.23

表4 各方法在相似轨迹搜索上的整体性能

Table 4 Overall performance of methods on similar trajectory search

数据集	成都			西安		
	Acc@1 ↑ /%	Acc@5 ↑ /%	Mean Rank ↓	Acc@1 ↑ /%	Acc@5 ↑ /%	Mean Rank ↓
t2vec	81.45	93.70	3.35	89.47	97.10	1.60
Trembr	83.98	89.88	4.66	88.00	93.00	3.48
CTLE	53.77	69.20	9.43	41.20	59.80	6.05
Toast	53.64	71.60	5.94	30.60	64.30	6.18
TrajCL	95.13	98.88	1.20	95.63	99.20	<u>1.09</u>
LightPath	74.27	86.10	27.27	79.63	91.70	13.88
START	<u>96.93</u>	<u>99.80</u>	<u>1.09</u>	<u>95.93</u>	<u>99.53</u>	1.14
ESTRL	98.07	99.90	1.04	99.77	99.97	1.01

对于目的地预测和到达时间估计任务,各方法要么使用固定参数进行预训练,要么使用任务监督进行微调。在两种情况下,本文方法均优于基线方法。这表明,ESTRL方法的预训练过程在不需要额外任务的特定监督的情况下,从轨迹中提取了丰富的出行语义。此外,基于Mamba的轨迹编码器

本文方法使用PyTorch^[47]实现。6个关键超参数及其最优值分别为 $B=128, L=4, N=128, H=4, E=256, D=256$ 。本文根据成都验证集上的目的地预测任务的MAE来决定超参数的取值。对于模型训练,本文使用初始学习率为0.001的Adam优化器。每组实验运行5次,并取指标的平均值。实验的具体环境配置如表2所列。

表2 实验软硬件环境

Table 2 Experimental hardware and software environment

名称	配置环境
CPU	Intel ^(R) Xeon ^(R) W-2155 CPU
GPU	TITAN RTX GPU
CUDA 版本	12.2
Python 版本	3.9.12
PyTorch 版本	2.1.2

5.5 性能比较

5.5.1 整体性能比较

表3和表4比较了不同方法在4.3节介绍的3个下游任务上的整体性能。可以看到,ESTRL方法在所有任务中都表现出一致的优越性能。

的设计使其能够在任务监督的情况下实现更优越的性能。对于相似轨迹搜索任务,预训练各方法并利用其输出嵌入进行相似度计算。本文方法在该任务中取得了最好的表现,进一步凸显了其预训练过程的有效性。

在各基线方法中,t2vec和TremBR忽略了关键的时空

特征,无法捕获出行目的,导致在下游任务上的性能较差。CTLE 和 Toast 由于缺乏必要的连续运动行为特征和无法提取出行目的而表现出局限性。值得注意的是,它们在相似轨迹搜索任务上的表现是次优的,因为它们没有直接学习轨迹级别的表示。TrajCL, START 和 LightPath 没有考虑 POI 的功能,而 START 和 LightPath 也由于对连续特征建模不充分而难以捕捉运动模式。因此,它们在目的地预测和到达时间估计任务中不能产生令人满意的结果。

5.5.2 模型效率分析

表 5 比较了不同方法在两个实验数据集上的效率。在模型规模和编码时间方面,ESTRL 方法表现出了较高的计算效率,其轻量化程度和嵌入速度与 TremBR 和 t2vec 等基于 RNN 的方法相当。与 START 和 LightPath 等基于 Transformer 的方法相比,本文方法在模型效率上具有显著优势。鉴于 ESTRL 方法在多种任务中的出色表现,它高效地实现了语义丰富的轨迹学习的设计目标。

表 5 各方法的效率

Table 5 Efficiency of methods

数据集	成都/西安		
	模型大小/ MBytes	训练时间/ (min/epoch)	编码时间/s
t2vec	1.641/1.415	2.783/5.937	4.445/9.705
Trembr	5.752/5.301	3.360/6.067	3.230/9.723
CTLE	3.756/3.756	4.533/14.354	14.581/33.863
Toast	4.008/3.557	4.400/10.650	14.540/33.863
TrajCL	4.382/3.932	7.699/14.567	10.253/23.877
LightPath	12.958/12.507	10.250/23.217	22.486/46.260
START	15.928/15.026	15.927/37.528	28.704/49.894
ESTRL	3.358/3.358	9.345/29.715	1.161/2.571

值得注意的是,本文方法在训练时间方面的表现并不突出。然而,由于预训练过程不会为嵌入编码过程增加额外的负担,且实际应用中模型的效率及性能更为关键,因此额外的训练时间可以被视为一种合理的权衡。

5.6 消融实验

本文在成都数据集上使用 w/o ft 设置下的目的地预测

任务对 ESTRL 方法的模块进行了分析,将完全(full)的 ESTRL 方法与以下模型变体进行比较。

- 1) w/o mb: 在编码器模块中将高阶移动特征参数化替换为普通的输入参数化。
- 2) w/o POI: 去除预训练机制中的 POI 视图。
- 3) w/o Road: 去除预训练机制中的道路视图。

表 6 展示了消融实验结果。可以观察到,移除高阶移动特征参数化对模型性能产生了负面影响,突出了该模块在提取运动行为方面的有效性。从出行目的感知预训练机制中移除 POI 或道路视图也会导致性能较差,表明两者都有助于出行目的信息的建模。

表 6 消融实验

Table 6 Ablation study

变体	RMSE ↓ /m	MAE ↓ /m
w/o mb	341.13	273.13
w/o POI	387.12	310.03
w/o Road	345.35	276.48
full	332.06	260.38

5.7 超参数分析

本文在成都数据集上使用 w/o ft 设置下的目的地预测任务对提出方法的关键超参数进行了敏感性分析。图 2 展示了 ESTRL 方法中的关键超参数的效果。根据这些实验数据,可以得出以下观察结果。

- 1) 批量大小 B 主要控制出行目的感知预训练机制中的轨迹负样本数量,最优值为 128。
- 2) 基于 Mamba 的轨迹编码器的层数 L 、状态维数 N 、嵌入维数 E 和模型维数 D 控制着模型的容量。其中 E 的效果最为突出,因为它直接控制了最终轨迹嵌入的维度。在权衡性能和效率后,它们的最优值分别为 $L=4$, $N=128$, $E=256$, $D=256$ 。
- 3) 头数 H 决定了基于 Mamba 的轨迹编码器中多输入 SSM 的复杂度,其最优值为 4。

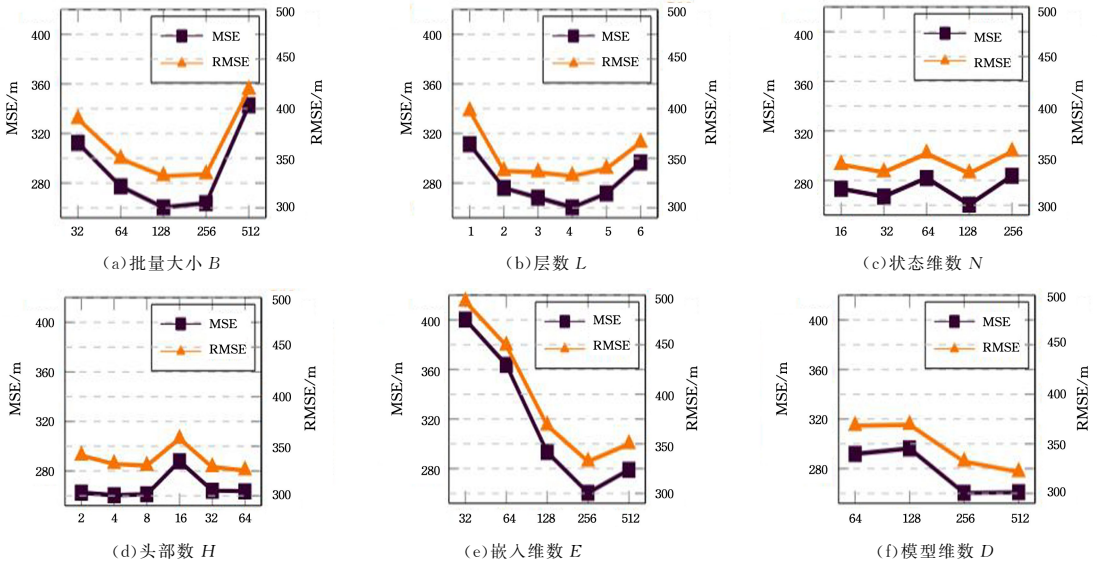


图 2 超参数有效性

Fig. 2 Effectiveness of hyper-parameters

结束语 本文提出了一种新的高效语义感知轨迹表示学习方法 ESTRL。该方法首先引入基于 Mamba 的轨迹编码器,使用高阶移动特征参数化的 Traj-SSM 捕获轨迹序列中的时空依赖关系,实现了有效且高效的连续运动行为提取。其次,提出了一种出行目的感知预训练机制,从轨迹中提取出行目的并保持计算效率。最后,在 2 个真实车辆轨迹数据集和 3 个代表性任务上的大量实验证明了提出方法的有效性。

ESTRL 实现了对车辆轨迹更精准高效的分析,为实时交通管理提供了更可靠的轨迹分析工具,未来可应用于自动驾驶车辆的协同轨迹预测或城市动态信号灯控制系统,以提升人类出行服务和城市交通管理水平。该方法目前受限于不同轨迹数据集中路段和 POI 的出行语义差异,因而难以实现模型的跨城市迁移和零样本实验。未来工作将致力于开发通用道路与 POI 嵌入表示,以增强跨城市迁移能力并提升模型可迁移性。

参 考 文 献

- [1] WU H, CHEN Z, SUN W, et al. Modeling trajectories with recurrent neural networks[C]// Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence. 2017: 3083-3090.
- [2] YAN B, ZHAO G, SONG L, et al. PreCLN: pretrained-based contrastive learning network for vehicle trajectory prediction [J]. World Wide Web, 2023, 26(4): 1853-1875.
- [3] LIN Y, WAN H, HU J, et al. Origin-destination travel time oracle for map-based services[J]. Proceedings of the ACM on Management of Data, 2023, 1(3): 1-27.
- [4] YUAN H, LI G, BAO Z, et al. Effective traveltime estimation: when historical trajectories over road networks matter [C]// Proceedings of the 2020 International Conference on Management of Data. ACM, 2020: 2135-2149.
- [5] HAN X, CHENG R, MA C, et al. DeepTEA: effective and efficient online time-dependent trajectory outlier detection[J]. Proceedings of the VLDB Endowment, 2022, 15(7): 1493-1505.
- [6] LIU Y, ZHAO K, CONG G, et al. Online anomalous trajectory detection with deep generative sequence modeling[C]// Proceedings of the 36th IEEE International Conference on Data Engineering. IEEE, 2020: 949-960.
- [7] HU D, CHEN L, FANG H, et al. Spatio-temporal trajectory similarity measures: a comprehensive survey and quantitative study[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(5): 2191-2212.
- [8] YAO D, HU H, DU L, et al. TrajGAT: a graph-based long-term dependency modeling approach for trajectory similarity computation [C]// KDD '22: the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2022: 2275-2285.
- [9] YAO D, ZHANG C, ZHU Z, et al. Trajectory clustering via deep representation learning [C]// 2017 International Joint Conference on Neural Networks. IEEE, 2017: 3880-3887.
- [10] CHUNG J, GÜLÇEHRE Ç, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [J]. arXiv:1412.3555, 2014.
- [11] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [12] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 5998-6008.
- [13] LIANG Y, OUYANG K, WANG Y, et al. TrajFormer: efficient trajectory classification with transformers [C]// Proceedings of the 31st ACM International Conference on Information & Knowledge Management. ACM, 2022: 1229-1237.
- [14] LIANG Y, OUYANG K, YAN H, et al. Modeling trajectories with neural ordinary differential equations [C]// Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence. ijcai.org, 2021: 1498-1504.
- [15] LIN Y, WAN H, GUO S, et al. Pre-training general trajectory embeddings with maximum multi-view entropy coding [J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(12): 9037-9050.
- [16] CHEN T Q, RUBANOVA Y, BETTENCOURT J, et al. Neural ordinary differential equations [C]// Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018. 2018: 6572-6583.
- [17] KIDGER P, MORRILL J, FOSTER J, et al. Neural controlled differential equations for irregular time series [C]// Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020. 2020.
- [18] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners [C]// Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020. 2020.
- [19] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACM, 2019: 4171-4186.
- [20] DU Z, QIAN Y, LIU X, et al. GLM: general language model pre-training with autoregressive blank infilling [C]// Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. ACM, 2022: 320-335.
- [21] ZHOU Z, LIN Y, WEN H, et al. PLM4Traj: cognizing movement patterns and travel purposes from trajectories with pre-trained language models [J]. arXiv:2405.12459, 2024.
- [22] FENG J, LI Y, ZHANG C, et al. DeepMove: predicting human mobility with attentional recurrent networks [C]// Proceedings of the 2018 World Wide Web Conference on World Wide Web. ACM, 2018: 1459-1468.
- [23] KONG D, WU F. HST-LSTM: a hierarchical spatial-temporal long-short term memory network for location prediction [C]// Proceedings of the Twenty-seventh International Joint Conference on Artificial Intelligence. ijcai.org, 2018: 2341-2347.
- [24] MIAO C, LUO Z, ZENG F, et al. Predicting human mobility via attentive convolutional network [C]// WSDM '20: the Thirteenth ACM International Conference on Web Search and Data Mining. ACM, 2020: 438-446.

- [25] CHEN W, LI S, HUANG C, et al. Mutual distillation learning network for trajectory-user linking [C] // Proceedings of the Thirty-first International Joint Conference on Artificial Intelligence. ijcai.org, 2022:1973-1979.
- [26] SANG Y, XIE Z, CHEN W, et al. TULRN: trajectory user linking on road networks[J]. World Wide Web, 2023, 26(4):1949-1965.
- [27] YAO D, CONG G, ZHANG C, et al. Computing trajectory similarity in linear time: a generic seed-guided neural metric learning approach[C] // 35th IEEE International Conference on Data Engineering. IEEE, 2019:1358-1369.
- [28] ZHOU S, HAN P, YAO D, et al. Spatial-temporal fusion graph framework for trajectory similarity computation[J]. World Wide Web, 2023, 26(4):1501-1523.
- [29] HINTON G E, SALAKHUTDINOV R R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313(5786):504-507.
- [30] LI X, ZHAO K, CONG G, et al. Deep representation learning for trajectory similarity computation[C] // 34th IEEE International Conference on Data Engineering. IEEE Computer Society, 2018: 617-628.
- [31] FU T Y, LEE W C. Trembr: exploring road networks for trajectory representation learning[J]. ACM Transactions on Intelligent Systems and Technology, 2020, 11(1):10:1-10:25.
- [32] ZHOU F, DAI Y, GAO Q, et al. Self-supervised human mobility learning for next location prediction and trajectory classification [J]. Knowledge-Based Systems, 2021, 228:107214.
- [33] OORD A VAN DEN, LI Y, VINYALS O. Representation learning with contrastive predictive coding [J]. arXiv:1807.03748, 2018.
- [34] JIANG J, PAN D, REN H, et al. Self-supervised trajectory representation learning with temporal regularities and travel semantics[C] // 39th IEEE International Conference on Data Engineering. IEEE, 2023:843-855.
- [35] CHEN T, KORNBLITH S, NOROUZI M, et al. A simple framework for contrastive learning of visual representations [C] // Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020:1597-1607.
- [36] LIU X, WANG Z, LI Y, et al. Self-supervised learning via maximum entropy coding[C] // Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022. 2022.
- [37] DAO T, GU A. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality [C] // Proceedings of the 41th International Conference on Machine Learning. PMLR, 2024:10041-10071.
- [38] TANCIK M, SRINIVASAN P P, MILDENHALL B, et al. Fourier features let networks learn high frequency functions in low dimensional domains[C] // Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020. 2020.
- [39] GU A, DAO T. Mamba: linear-time sequence modeling with selective state spaces[J]. arXiv:2312.00752, 2023.
- [40] MEERT W, VERBEKE M. HMM with non-emitting states for Map Matching [C] // European Conference on Data Analysis (ECDA). 2018.
- [41] RADFORD A, KIM J W, HALLACY C, et al. Learning transferable visual models from natural language supervision [C] // Proceedings of the 38th International Conference on Machine Learning. PMLR, 2021:8748-8763.
- [42] LIN Y, WAN H, GUO S, et al. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction [C] // Proceedings of the AAAI Conference on Artificial Intelligence. AAAI, 2021:4241-4248.
- [43] CHEN Y, LI X, CONG G, et al. Robust road network representation learning: when traffic patterns meet traveling semantics [C] // Cikm '21: the 30th ACM International Conference on Information and Knowledge Management. ACM, 2021:211-220.
- [44] CHANG Y, QI J, LIANG Y, et al. Contrastive trajectory similarity learning with dual-feature attention [C] // 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, 2023:2933-2945.
- [45] YANG S B, HU J, GUO C, et al. LightPath: lightweight and scalable path representation learning [C] // Proceedings of the 29th ACM Sigkdd Conference on Knowledge Discovery and Data Mining. ACM, 2023:2999-3010.
- [46] FANG Z, DU Y, ZHU X, et al. Spatio-temporal trajectory similarity learning in road networks [C] // Kdd '22: the 28th ACM Sigkdd Conference on Knowledge Discovery and Data Mining. ACM, 2022:347-356.
- [47] PASZKE A, GROSS S, MASSA F, et al. PyTorch: an imperative style, high-performance deep learning library [C] // Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019. 2019: 8024-8035.



LIU Yichen, born in 2001, master candidate. Her main research interests include deep learning, spatiotemporal data mining and representation learning.



WAN Huaiyu, born in 1981, Ph.D. professor, Ph. D supervisor, is a distinguished member of CCF (No. 17732D). His main research interests include spatiotemporal data mining, social network mining, information extraction and knowledge graph.